Модел. и анализ информ. систем. Т. **20**, № **2** (2013) 34–53

УДК 519.7 + 519.8 + 004.8

Мультиагентная задача о роботах в пространстве: сложностно́й, информационный и криптографический аспекты

Бернштейн А.Ю.[1], Шилов Н.В.[2]

[1]*Новосибирский государственный университет*
630090, г. Новосибирск, ул. Пирогова, д. 2
[2]*Институт систем информатики им. А.П. Ершова*
630090, г. Новосибирск, проспект Лаврентьева, 6

e-mail: bahtoh@gmail.com; shilov@iis.nsk.su

получена 18 июня 2012

Ключевые слова: мультиагентные (многоагентные) системы и алгоритмы, геометрическая задача о назначениях, анонимность, масштабируемость, свойства безопасности и прогресса, верификация алгоритмов.

В статье изучается следующая мультиагентная алгоритмическая *задача о роботах в пространстве* (Robot in Space — RinS): *В пространстве есть* $n \geq 2$ *автономных робота, которым необходимо самостоятельно договориться о выборе индивидуальных укрытий так, чтобы прямолинейные маршруты к этим укрытиям не пересекались.* Эта задача имеет отношение к *задаче о назначениях* в теории графов, *задаче построения выпуклой оболочки* в комбинаторной геометрии и *задаче планирования перемещений* в искусственном интеллекте. Предлагаемый в статье мультиагентный алгоритм (протокол) основан на централизованном локальном алгоритме Э.В. Дейкстры. Наш алгоритм обладает свойствами анонимности и масштабируемости, мы доказываем его корректность и верхнюю оценку сложности. Кроме того, мы исследуем два коммуникационных аспекта задачи о роботах в пространстве — *информационный* и *криптографический*. В статье показано, что (1) не существует протокола, решающего задачу RinS, передающего ограниченное число битов, но (2) существует протокол для решения этой задачи, который позволяет роботам не раскрывать информацию о своём положении относительно укрытий. Настоящая статья является продолжением исследований, представленных в статье Е.В. Бодина, Н.О. Гараниной и Н.В. Шилова *"Задача о роботах на Марсе (мультиагентный подход к задаче Дейкстры)"* опубликованной в № 2 за 2011 г. журнала *"Моделирование и анализ информационных систем"*.

[1]Работа поддержана грантом РФФИ 10-01-00532-а на 2010–2012 гг.
[2]Работа выполнена в рамках проекта СО РАН 15/10 на 2012–2014 гг.

1. Введение

Распределённая система — это группа децентрализованных взаимодействующих исполнителей [6]. *Распределённый алгоритм* — это протокол (алгоритм) работы и взаимодействия исполнителей в распределённой системе, превращающий децентрализованную группу в коллектив, совместно решающий некоторую задачу [5]. *Масштабируемый* алгоритм (или протокол) — это алгоритм для исполнителя в распределённой системе (протокол распределённой системы соответственно), формулировка которого не фиксирует число (множество) исполнителей в системе. Например, альтернирующий битовый протокол ABP (Alternating Bit Protocol) не является масштабируемым, т.к. он сразу фиксирует (число) участников, а протокол передачи гипертекста HTTP (Hyper Text Treansfer Protocol) является масштабируемым. *Параметрический* алгоритм (или протокол) — это алгоритм для исполнителя в распределённой системе (протокол распределённой системы соответственно), в котором множество исполнителей в системе использовано как параметр-константа.

Мультиагентная система — это распределённая система, состоящая из агентов [19], имеющих (возможно) разные роли в системе. *Агент* — это автономный (воспринимающий мир разделённым на *себя* и *среду*, включающую всё остальное), реагирующий (способный взаимодействовать со средой и отвечать на воздействия среды) объект (в объектно-ориентированном смысле), внутреннее состояние которого можно охарактеризовать в терминах *мнений* или *веры* (Believes), *целей* (Desires) и *намерений* (Intentions) агента. *Мультиагентный алгоритм* — это распределённый алгоритм для мультиагентной системы. Будем говорить, что протокол является *ролевым*, если он одинаков для всех агентов с одинаковой ролью (т.е. не зависит от переименования агентов внутри группы с одной ролью). Будем говорить, что протокол является *анонимным*, если во время переговоров он не позволяет агенту узнать, с кем именно из агентов-напарников он имеет дело (но позволяет определить роль, например). Таким образом, мы трактуем анонимность в духе *теории выбора* (см., например, теорему И. Мэя в гл. 6 из [4]), и в духе работ [11, 12].

Вера агента является совокупностью его представлений о себе и о среде, которые могут быть неполными, несогласованными и вообще неверными (не соответствующими действительности). Согласно Платону, знания — это представления о реальности, соответствующие реальности. Поэтому *знания* агента — это его вера, которая соответствует действительности. Цели агента — это его долгосрочные задачи и обязанности, которые также могут быть несогласованными. Намерения агента используются для краткосрочного планирования. Агент является реагирующим, т.е. он может изменить свою веру, цели или намерения после взаимодействия с другими агентами/средой; однако изменение его внутреннего состояния зависит только от него, но не от окружения (агент *автономен*). Агенты описанного вида обычно называются BDI-агентами [19].

2. Задача о роботах в пространстве

2.1. Постановка задачи

Рассмотрим в качестве первого примера следующую задачу, которую будем называть задачей о **роботах в пространстве**:

> *В евклидовом пространстве $\mathbb{R}^k$ ($k \geq 2$) находятся $n > 1$ автономных роботов и столько же укрытий, все роботы и укрытия считаем точками $\mathbb{R}^k$ в общем положении*[3]*. Местоположение всех укрытий фиксировано и известно каждому роботу. Каждому роботу изначально назначено индивидуальное укрытие, о котором робот знает. Каждый робот знает о существовании всех остальных роботов, но не знает их координаты, и знает, что каждому роботу изначально назначено и известно индивидуальное укрытие. В некоторый момент времени все роботы останавливаются, фиксируют свои текущие позиции; затем они все должны выбрать укрытия, чтобы двинуться к ним по прямолинейному маршруту. Ясно, что роботам нельзя сталкиваться. Роботы могут каждый с каждым попарно (P2P — "peer to peer") вести переговоры об укрытиях и обмениваться укрытиями. Задача: разработать анонимный мультиагентный алгоритм переговоров, гарантирующий, что каждый робот когда-нибудь будет точно знать, что его маршрут к укрытию, которое он выберет в результате переговоров, не пересекается с маршрутами остальных роботов.*

В дальнейшем мы будем обозначать эту задачу RinS — *Robot in Space*. Она "выросла" из ранее рассмотренной задачи о роботах на Марсе (*Mars Robot Puzzle* — MRP) [2]: MRP является частным случаем RinS на плоскости. Обе задачи RinS и MRP — это мультиагентная интерпретация т.н. *задачи Дейкстры* [17]:

> *Разработать алгоритм, соединяющий попарно данные n чёрных и n белых точек в общем положении на плоскости таким образом, чтобы отрезки, связывающие точки, не пересекались.*

Эту задачу Дейкстры можно решать как частный "геометрический" случай задачи о назначениях из теории графов или как частный случай комбинаторной геометрической задачи о выпуклой оболочке [2], тогда как RinS и MRP может рассматриваться как частный случай задачи планирования перемещений [8, 15].

С точки зрения классификации задач, которые решают мультиагентные системы, задача RinS является одновременно *задачей поддержания состояния* и *задачей достижения состояния*. Такая классификация задач в теории мультиагентных систем определяется следующим образом [19]:

- Задачи поддержания состояния возникают, когда агенту (или агентам) требуется избежать определенных нежелательных состояний системы и оставаться всё время в *безопасных* состояниях. Например, в задаче RinS роботам надо всё время оставаться в состоянии или покоя, или движения по маршрутам без пересечений.

[3] То есть никакие три из этих точек не лежат на одной прямой.

- Задачи достижения состояния возникают, когда агенту (или агентам) требуется достичь определённого *прогресса*, т.е. целевого состояния (или состояний). Например, в задаче RinS всем роботам надо достичь состояния знания, что их маршруты к выбранным укрытиям не пересекаются.

2.2. Мультиагентный протокол для RinS

Наш мультиагентный протокол для RinS подразумевает, что все роботы исполняют один и тот же алгоритм (играют одну роль). Этот алгоритм состоит из алгоритма переговоров и обмена и алгоритма-планировщика общения между роботами. Будем говорить, что алгоритм-планировщик общения между роботами удовлетворяет условию *справедливости*, если в любой момент времени для любого робота, желающего контактировать в этот момент с каким-либо другим роботом-партнёром, обязательно наступит в будущем момент времени[4], когда партнёр будет готов к общению с данным роботом. Мы не будем обсуждать варианты алгоритма-планировщика, а рекомендуем обратиться для этого к работе [2], в которой описан справедливый, масштабируемый и параметрический алгоритм-планировщик, основанный на приоритетах. С обзором по проблеме справедливости в мультиагентных системах можно познакомиться по статье [13] или диссертации [14].

Как уже было сказано выше, RinS — обобщение задачи RAM [2] с двумерного случая $\mathbb{R}^2$ на многомерный $\mathbb{R}^k$, $k \geq 2$. Поэтому в качестве алгоритма для переговоров и обменов примем алгоритм SMEx (Safe Mars Explorer), разработанный для MRP [2]. Мы не будем воспроизводить псевдокод этого алгоритма, а только неформально опишем его суть.

Целью каждого робота является выбор индивидуального укрытия так, чтобы избежать столкновения на пути к нему. Эта цель достигается посредством кооперативного поиска бесконфликтного/безопасного (т. е. без пересечений) множества прямолинейных маршрутов к выбранным укрытиям. В каждый момент времени в намерениях каждого робота содержится его текущее укрытие-назначение. Вера каждого робота представлена значениями двух целочисленных счётчиков: NC для *числа собственных конфликтов* и CF для *числа бесконфликтных роботов*[5]; NC представляет индивидуальную верхнюю оценку числа роботов, с которыми данный робот может иметь конфликт (Number of Conflicts), а соответственно, CF — индивидуальная нижняя оценка числа роботов, которые вообще конфликтов не имеют (Coflict-Free). Кроме того, пусть у каждого робота есть несколько переменных для вещественных чисел, он умеет складывать, вычитать, сравнивать вещественные значения и вычислять расстояния (от своей позиции на плоскости до укрытий).

Алгоритм SMEx делит время работы робота на раунды — интервалы, в течение которых он должен ровно один раз пообщаться со всеми другим роботами. Мы различаем два типа раундов: основной и контрольный[6].

Каждый робот во время основного раунда последовательно общается со всеми другими роботами и подсчитывает в своей переменной NC, сколько ещё конфликтов может быть с другими роботами, т. е. “формирует” мнение (веру) о том, есть ли

[4]Будущий по отношению к текущему моменту времени.

[5]В [2] для счётчика CF использован идентификатор CFR.

[6]В [2] *основной раунд* был назван *основным процессом*, а *контрольный — пост-процессом*.

у него конфликты с другими роботами. Суть сеанса общения между роботом и его партнёром во время основного процесса состоит в следующем[7]:

```
send <NC> to partner; receive <NC партнёра> from partner;
send <текущее назначение> to partner;
receive <назначение партнёра> from partner;
s11:= расстояние до своего текущего назначения;
s12:= расстояние до текущего назначения партнёра;
send <s11,s12> to partner; receive <s22,s21> from partner;
if s11 + s22 > s12 + s21
then{ обменяться назначениями с партнёром; NC:= (n - 1); CF:= 0 }
else if NC > 0 then { NC:= (NC - 1); CFR:= 0 } ......
```

Во время контрольного раунда каждый робот, который верит, что у него конфликтов нет (т. е. его индивидуальный $NC = 0$), последовательно общается со всеми другими роботами и накапливает в своём счётчике CF, сколько других роботов тоже считают себя бесконфликтными (т. е. их собственный $NC = 0$).

Основной и контрольный раунды могут чередоваться (так как NC и CF реинициализируются в случае обнаружения конфликтов или роботов с конфликтами). Однако два последних раунда работы каждого робота — обязательно контрольные, когда значение его NC все время равно 0. В течение первого из этих двух раундов робот уведомляет всех других роботов, что его $NC = 0$, получает аналогичные уведомления от всех остальных роботов, и монотонно наращивает значение своего CF от 0 до $(n - 1)$. В течение второго из этих двух раундов уведомляет всех других роботов, что его $NC = 0$, получает аналогичные уведомления от всех остальных роботов и монотонно наращивает значение своего CF от $(n-1)$ до конечного значения $2\times(n-1)$. Условием завершения работы робота является $(NC = 0 \& CF = 2\times(n-1))$. Робот может начать движение к выбранному укрытию в любой момент только *по завершении двух последних контрольных раундов*.

Остаётся только отметить, что (как следует из представленного описания алгоритма SMEx), описанный мультиагентный алгоритм для задачи RinS на основе алгоритма SMEx является масштабируемым и параметрическим, но не является анонимным (т.к. каждый робот во время перговоров знает, с кем ведёт переговоры).

3. Корректность алгоритма

Т.к. мультиагентный алгоритм — это протокол распределённой системы, то понятие корректности для мультиагентных алгоритмов наследуется от понятия корректности протоколов распределённых систем. *Корректность* протоколов распеределённых систем обычно характеризуют в терминах *безопасности* и *прогресса* [6]. Свойство *безопасности* — это утверждение о том, что система при соблюдении протокола никогда не попадёт ни в какое *плохое* состояние, и система будет всё время находиться только в *безопасных* состояниях. Свойство *прогресса* — это утверждение о том, что какое-либо *целевое* состояние обязательно будет достигнуто системой при

[7]Использован псевдокод, близкий к псевдокоду из [5].

соблюдении протокола. Как видно из определений, свойство безопасности в распределённых системах соответствует задаче поддержания состояний в мультиагентных системах, а свойство прогресса — задаче достижения состояний. Таким образом, в контексте RinS

- свойство безопасности состоит в том, что все роботы находятся в состоянии или покоя, или движения по маршрутам без пересечений,
- свойство прогресса состоит в том, что всем роботам надо достичь состояния знания, что их маршруты к выбранным укрытиям не пересекаются.

В следующих двух утверждениях 1 и 2 предполагается следующее условие: *мультиагентная система состоит из $n > 1$ роботов и справедливого планировщика, все роботы исполняют алгоритм SMEx и в самом начале они имеют взаимнооднозначно назначенные начальные укрытия, все роботы и укрытия находятся в общем положении.*

Утверждение 1. *Система обязательно остановится (т. е. все роботы завершат исполнение алгоритма).*

Доказательство. Предположим противное: пусть есть роботы, которые будут "работать вечно". Следовательно, они будут исполнять команду обмена назначениями бесконечное число раз (также как и реинициализацию NC и CF). Рассмотрим следующую общую сумму расстояний от роботов до их текущих (на данный момент) укрытий. Так как никакие три объекта не лежат на одной прямой, то каждый обмен уменьшает значение этой общей суммы. В то же время имеется всего $n!$ возможных значений этой сумм и, следовательно, эта сумма не может уменьшаться бесконечное число раз. Противоречие. ■

Утверждение 2. *По завершении работы любого робота системы не будет пересекающихся путей роботов к выбранным ими укрытиям.*

Доказательство. Если робот завершил свою работу, то два последних его раунда — контрольные. В течение первого из этих двух раундов робот уведомляет всех других роботов, что его $NC = 0$, получает аналогичные уведомления от всех остальных роботов и монотонно наращивает значение своего CF от 0 до $(n-1)$. Но то, что робот получил 0 в качестве значения NC партнёра, означает, что этот партнёр выполняет свой собственный пост-процессный раунд. Следовательно, на этом раунде на момент переговоров робота с любым другим роботом их маршруты не пересекались.

В течение последнего из этих контрольных раундов робот уведомляет всех других роботов, что его $NC = 0$, получает аналогичные уведомления от всех остальных роботов и монотонно наращивает значение своего CFR от $(n-1)$ до конечного значения $2(n-1)$. Но то, что он получил 0 в качестве значения NC партнёра, означает, что этот партнёр выполняет свой собственный контрольный раунд, который тоже должен быть вторым контрольным раундом. Следовательно, в этом раунде все роботы подтверждают (послав 0 как значение их NC), что они не меняли своего назначения. ■

Следующая теорема является следствием из утверждений 1 и 2.

Теорема 1. *Пусть мультиагентная система состоит из $n > 1$ роботов и справедливого планировщика, все роботы исполняют алгоритм SMEx и в самом начале они имеют взаимно-однозначно назначенные начальные укрытия, все роботы и укрытия находятся в общем положении. Тогда эта мультиагентная система корректна, т.е. обладает следующими свойствами безопасности и прогресса:*

- *во время работы алгоритма все роботы системы находятся в состоянии или покоя,или движения по маршрутам без пересечений,*
- *каждый робот спустя конечное время завершает исполнение алгоритма и перед началом движения в конце алгоритма знает, что его маршрут к выбранному укрытию не пересекается с маршрутами других роботов.*

Стоит заметить, что утверждение о знании во втором пункте этой теоремы основано на том, что каждый робот в конце алгоритма перед началом движения верит, что ни у него ($NC = 0$), ни у других роботов ($CF = 2 \times (n - 1)$) нет конфликтов, и эта вера соответствует действительности, как это утверждается в первом пункте теоремы.

4. Новые аспекты задачи

Первый вопрос, оставшийся открытым в предыдущей части (и явно сформулированный в предшествующей работе [2]) — это вопрос о сложности (количестве сеансов общения между роботами) описанного выше мультиагентного протокола; этот вопрос составляет содержание *сложностно́го* аспекта задач MRP и RinS. Второй вопрос, требующий дополнительного исследования, — это вопрос о сложности передачи отдельных сообщений, которые необходимы для работы описанного протокола; в этом состоит *информационный аспект* обеих задач. *Криптографический аспект* обеих задач состоит в следующем: в описанном выше протоколе агенты сообщали друг другу достаточно много сведений о себе, что может позволить каждому отдельному роботу при определённых условиях вычислить положения всех других роботов (и решить задачу сразу за всех роботов); поэтому возникает третий вопрос о возможности для роботов сокрытия сведений о своём положении от других роботов.

Введём некоторую терминологию и обозначения для удобства дальнейшего обсуждения сложностно́го, информационного и криптографического аспектов задач MRP и RinS. *Протоколом распределения укрытий* между $n > 1$ участниками будем называть любой протокол, удовлетворяющий следующим условиям:

1. В протоколе участвуют $n > 1$ агентов-роботов R_1, ..., R_n.
2. Каждый робот R_i, $1 \le i \le n$, знает свои координаты $(x_i, y_i, \dots) \in \mathbb{R}^k$, неизвестные остальным роботам.
3. Имеются также $n > 1$ векторов-констант, известных всем роботам, $(x^1, y^1, \dots)$, ... $(x^n, y^n, \dots) \in \mathbb{R}^k$ — координат укрытий.
4. Все роботы и укрытия в совокупности находятся в общем положении, а векторы $(x_1, y_1, \dots)$, ... $(x_n, y_n, \dots)$ $(x^1, y^1, \dots)$, ... $(x^n, y^n, \dots)$ — входные данные протокола.

5. Есть $n > 1$ глобальных переменных S_1,... S_n, открытых для чтения всем роботам, но модифицировать значение переменной S_i может только робот R_i, $1 \leq i \leq n$.

6. В переменных S_1,... S_n хранятся текущие назначения укрытий для роботов, и изначально они инициализированы *разными* укрытиями.

7. Между каждыми двумя роботами существует канал связи, по которому они могут обмениваться сообщениями.

8. В случае, если того требует протокол, любые два робота R_i и R_j, $1 \leq i, j \leq n$, могут обменяться укрытиями, то есть обменяться значениями переменных S_i и S_j.

9. Протокол завершает работу за конечное время, а по его завершении любые два отрезка $[R_i, S_i]$ и $[R_j, S_j]$, $1 \leq i < j \leq n$, не пересекаются.

По сути, любой протокол назначения укрытий — это протокол, решающий задачу RinS. Сужением протокола на множество S будем называть протокол, отличающийся только тем, что входные данные для него должны принадлежать множеству S.

Назовём протокол, участниками которого являются два робота, позволяющий этим роботам проверить, нужно ли им обмениваться укрытиями, *щелчком*, если он удовлетворяет следующим двум условиям:

- Если текущие пути роботов пересекаются, то они должны совершить обмен.

- Если обмен укрытиями совершился, то суммарная длина путей строго уменьшилась.

Эти два свойства позволяют на основе любого протокола щелчка построить протокол распределения укрытий, если использовать этот щелчок как основной процесс в алгоритме SMEx. Два крайних случая щелчка — это *простой щелчок*, при котором обмен укрытиями выполняется тогда и только тогда, когда пути пересекаются, и *щелчок со сравнением*, при котором обмен укрытиями выполняется тогда и только тогда, когда сумма длин путей уменьшается[8]. В рамках данного исследования для нас не важно, как именно устроен щелчок, но существенно то, что при исполнении протокола распределения роботы сообщают друг другу что-либо о своём положении только во время щелчков.

Теорема 2. *Любой протокол назначения укрытий, основанный на протоколе щелчка, не может совершить более* $\frac{L-l}{\Delta} \times n$ *щелчков, где* L *и* l *— максимальное и минимальное расстояние между роботом и укрытием в системе, а* Δ *— минимальный декремент суммы длин путей при щелчке. В частности, в мультиагентном протоколе, основанном на алгоритме SMEx и справедливом планировщике контактов, не может быть более* $(2 + \frac{L-l}{\Delta} \times n)$ *раундов.*

[8]Алгоритм SMEx использует щелчок со сравнением.

Доказательство. $L \times n$ — верхняя граница суммарной длины маршрутов, $l \times n$ — нижняя граница суммарной длины маршрутов. Так как при щелчке суммарная длина убывает как минимум на величину Δ (которая не равна 0, т.к. никакие три объекта не лежат на одной прямой), то щелчков не может быть больше $\frac{L-l}{\Delta} \times n$. А так как на каждом раунде (кроме двух последних контрольных) протокола, основанного на алгоритме SMEx и справедливом планировщике контактов, обязательно происходит хотя бы один щелчок хотя бы с одной парой роботов, поэтому в этом протоколе не может быть более $(2 + \frac{L-l}{\Delta} \times n)$ раундов. ■

5. Исследование информационного аспекта

В рамках исследования информационного аспекта задачи RinS в качестве входных данных будут рассматриваться действительные числа. При этом нас не будет интересовать то, как роботы хранят их в памяти. Кроме того, мы будем считать, что роботы могут мгновенно выполнять с ними (с абсолютной точностью) произвольные операции: складывать, вычитать, умножать, логарифмировать и так далее. Таким образом, единственной проблемной частью протоколов становится передача роботами результатов своих вычислений друг другу.

5.1. Характеристики протоколов

Основная величина, которая будет нас интересовать, — это общее количество бит, переданное участниками друг другу в ходе исполнения протокола. Будем называть протокол *финитным*, если при любых допустимых значениях входных данных его работа завершается после передачи конечного количества бит. Будем называть протокол *ограниченным*, если существует такая константа N, что при любых допустимых значениях входных данных его работа завершается после передачи не более N бит.

Ещё один способ ослабить свойство ограниченности мы получим, если вместо количества переданных бит будем рассматривать только количество переданных сообщений. Будем называть сообщением любую конечную битовую последовательность, переданную агентом своему напарнику “за один сеанс связи” (т.е. когда агент не получает сообщений от напарника). Будем называть протокол *ограниченным по сообщениям*, если существует такая константа N, что при любых допустимых значениях входных данных его работа завершается после передачи не более N сообщений. Поскольку каждое сообщение содержит по меньшей мере один бит, любой ограниченный протокол является также и ограниченным по сообщениям. Будем говорить, что протокол *имеет ограничивающую функцию* f, если количество сообщений, передаваемое участниками за время его работы на допустимых значениях входных данных, не превосходит значения этой функции f на тех же данных.

Всякий ограниченный протокол автоматически является ограниченным по сообщениям (т.к. любое сообщение имеет хотя бы один бит). Всякий ограниченный по сообщениям протокол автоматически имеет ограничивающую функцию-константу (в качестве значения которой можно принять константу, участвующую в определении ограниченности). Всякий имеющий ограничивающую функцию протокол автоматически является финитным.

5.2. Вспомогательные двусторонние протоколы

Пусть M — произвольное множество (вещественных) чисел. *Протокол равенства* на множестве M — это любой двусторонний протокол, удовлетворяющий следующим условиям: изначально участник (сторона) A знает число $x_A \in M$, неизвестное участнику B, а участник (сторона) B знает число $x_B \in M$, неизвестное участнику B, но в результате работы протокол позволяет за конечное время выяснить обоим участникам, верно ли, что $x_A = x_B$. *Протокол сравнения* на множестве M — это любой двусторонний протокол, удовлетворяющий следующим условиям: изначально участник (сторона) A знает число $x_A \in M$, неизвестное участнику B, а участник (сторона) B знает число $x_B \in M$, неизвестное участнику A, причём всем известно[9], что числа $x_A \neq x_B$; однако в результате работы протокол позволяет за конечное время выяснить обоим участникам, какое из чисел x_A или x_B больше.

Следующие два утверждения 3 и 4 могут показаться очевидными, т.к. проверка на равенство или сравнение подразумевает передачу последовательно знаков десятичного разложения чисел, причём нельзя сказать заранее, сколько знаков предстоит передать. Однако они требуют формального обоснования.

Утверждение 3. *Не существует финитного протокола равенства на более чем счётном множестве.*

Доказательство. Предположим противное. Пусть существует финитный протокол проверки равенства на некотором более чем счётном множестве M. Для любых $y, z \in M$ обозначим через $d(y, z)$ "диалог" — конечную последовательность битов, индексированных именами отправителей A или B, которые были посланы друг другу во время проверки равенства значений $x_A = y$ и $x_B = z$. Если C — это любая из сторон A или B, а $\overline{C}$ — её напарник B или A соответственно, то C делает вывод об истинности $x_{\overline{C}} = x_C$, исходя из диалога $d(x_C, x_{\overline{C}})$.

Заметим, что множество всех диалогов счётно. Поэтому найдутся такие числа $y, z \in M$, что $d(y, y) = d(z, z)$, но $y \neq z$. Обозначим посредством d диалог $d(y, y) = d(z, z)$. Рассмотрим работу протокола для входных данных $x_A = y$ и $x_B = z$. Докажем индукцией по длине префикса, что $d(y, z) = d$. Это утверждение верно для префиксов длины 0. Предположим, что префиксы длины $k \geq 0$ у $d(y, z)$ и d совпадают. Пусть следующее $(k + 1)$-е сообщение отправляет участник $C \in \{A, B\}$. Это сообщение однозначно определяется числом x_C и префиксом длины k диалога $d(y, z)$. Но (по предположению индукции) этот префикс совпадает с префиксом длины k диалога d. Следовательно, должны совпадать и $(k + 1)$-е сообщения в диалагах $d(y, z)$ и d, что завершает доказательство шага индукции. Но так как $d(y, z) = d(y, y) = d(z, z) = d$, то обе стороны должны будут сделать вывод, что $y = z$. Полученное противоречие завершает доказательство.■

Утверждение 4. *Не существует ни ограниченного протокола равенства, ни ограниченного протокола сравнения на бесконечном множестве.*

[9]Здесь фраза "всем известно" использована очень неформально. Точнее следует сказать, здесь речь идёт о так называемом *общем знании* (common knowledge) [3, 9]: каждый знает факт, о котором идёт речь, и знает, что все его знают, и знает, что все знают, что каждый знает этот факт, и т.д.

Доказательство невозможности существования ограниченного протокола равенства можно провести от противного аналогично доказательству предыдущего утверждения 3: теперь множество всех возможных диалогов фиксированной длины — конечно (а не счётно, как в утверждении 3), поэтому для получения противоречия достаточно просто бесконечного (вместо не более чем счётного, как в утверждении 3) множества входных данных для протокола.

Однако доказательство невозможности ограниченного протокола сравнения отличается от доказательства утверждения 3. Опять предположим противное. Пусть существует ограниченный протокол сравнения на некотором бесконечном множестве M, а N — верхняя граница числа бит, передаваемых в соответствии с этим протоколом. Для неравных чисел $y, z \in M$ обозначим через $d(y, z)$ диалог, который получается при исполнении данного протокола с начальными данными $x_A = y$ и $x_B = z$. Для любого $y \in M$ пусть $D(y) = \{d(y, z) : z \in M,\ z \neq y\}$. Поскольку диалогов, содержащих не более N бит, конечное число, то множество всех множеств таких диалогов тоже конечно. В частности, существует лишь конечное множество различных $D(y)$, $y \in M$. Следовательно, в бесконечном множестве M найдутся такие три числа $y_1 < z < y_2$, что $D(y_1) = D(y_2)$.

Покажем индукцией по длине префикса, что $d(y_1, z) = d(y_2, z)$. База индукции для префиксов длины 0 очевидна. Предположим, что префикс d длины $k \geq 0$ является общим у $d_{y_1,z}$ и $d_{y_2,z}$.

- Если "авторство" $(k+1)$-го бита принадлежит B, то значение $(k+1)$-го бита в диалогах $d_{y_1,z}$ и $d_{y_2,z}$ однозначно определяется префиксом d и значением переменной $x_B = z$ и, следовательно, совпадает.

- В случае, если "авторство" принадлежит A, то значение $(k+1)$-го бита b в диалоге $d_{y_1,z}$ однозначно определяется префиксом d и значением переменной $x_A = y_1$. Так как $D(y_1) = D(y_2)$, то существует такое число $z' \in M$, что $d(y_1, z) = d(y_2, z')$. В частности, префикс длины $(k+1)$ в диалогах $d(y_1, z)$ и $d(y_2, z')$ совпадает, и, следовательно, $(k+1)$-ый бит b в диалоге $d_{y_2,z'}$ однозначно определяется префиксом d и значением переменной $x_A = y_2$. Но в таком случае $(k+1)$-й бит в диалоге $d_{y_2,z}$ тоже b, т.к. это значение однозначно определяется префиксом d и значением переменной $x_A = y_2$.

Таким образом, совпадение диалогов $d(y_1, z)$ и $d(y_2, z)$ доказано.

Вывод, который делает агент B о сравнении значений y_1 и z, y_2 и z, зависит только от значения $x_B = z$ и (соответственно) диалогов $d(y_1, z)$ и $d(y_2, z)$. Но так как $d(y_1, z) = d(y_2, z)$, то эти выводы должны совпадать, в то время как $y_1 < z < y_2$. Полученное противоречие завершает доказательство. ■

Следующее утверждение является следствием предыдущего.

Утверждение 5. *Не существует ограниченного по сообщениям протокола сравнения на более чем счётном множестве.*

Доказательство. Предположим противное: пусть существует ограниченный по сообщениям протокол сравнения π на некотором более чем счётном множестве M. Пусть N — верхняя граница числа сообщений в этом протоколе. Для произвольных точек $y, z \in M$ пусть $d(y, z)$ — это диалог между агентами A и B, работающими по

этому протоколу, в случае, когда $x_A = y$ и $x_B = z$, а $d_i(y, z)$, где $i \in [1..N]$, — i-е сообщение в этом диалоге. В силу того, что каждое сообщение — это конечная последовательность битов, множество M можно представить в виде $M = \bigcup_{k_1,\dots k_N>0} M_{k_1\dots k_N}$, где $M_{k_1\dots k_N} = \{y, z \in M : \forall i \in [1..N] : |d_i(y, z)| = k_i\}$, $k_1, \dots k_N > 0$. Так как M более чем счётно, то найдутся такие $k_1, \dots k_N > 0$, что $M_{k_1\dots k_N}$ бесконечно. Но тогда сужение протокола π на это множество $M_{k_1\dots k_N}$ является ограниченным протоколом сравнеия на бесконечном множестве (с границей $\sum_{1\le i\le N} k_i$ для числа передаваемых бит). Пришли к противоречию с утверждением 4. ■

5.3. Информационный аспект: результаты

Теорема 3. *Для любого $n \ge 2$ существует финитный протокол распределения укрытий между n участниками.*

Доказательство. Достаточно модифицировать алгоритм SMEx следующим образом: робот отправляет напарнику и получает от напарника по переговорам не точные значения расстояний до укрытий, а поэтапно — с точностью до одного знака после запятой, потом — до второго знака и т.д., пока не сможет вычислить наверняка, уменьшится ли суммарная длина путей после обмена. ■

Следующие теоремы 4 и 5 являются нашими главными результатами в рамках информационной постановки задачи.

Теорема 4. *Пусть множество точек $M \subseteq \mathbb{R}^k$, $k \ge 2$, содержит бесконечное подмножество $S \subseteq M$ точек, лежащих в общем положении, такое, что существует непрерывная замкнутая плоская кривая, содержащая S и ограничивающая выпуклое плоское множество. Тогда сужение любого протокола распределения укрытий между $n \ge 2$ участниками на множество M не может быть ограниченным. Кроме того, если множество S более чем счётно, то сужение любого протокола распределения укрытий между $n \ge 2$ участниками на множество M не может быть ограниченным по сообщениям.*

Доказательство.

Предположим противное: пусть существует некоторый протокол распределения укрытий для множества M. Рассмотрим плоскость, в которой лежит подмножество S и непрерывная замкнутая кривая L, содержащая S и ограничивающая выпуклое множество на этой плоскости. Покажем, как, используя сужение протокола распределения укрытий на множество S, построить сужение протокола сравнения на множество той же мощности, что и S. Выберем ориентацию кривой L и рассмотрим произвольные точки X_1, X_2, ... X_{2n-2} из множества S, перечисленные в соответствии с ориентацией L. Пусть $\uparrow \{X_1, \dots, X_{2n-2}\}$ — это множество точек $\{X \in S \ : X_1, \dots, X_{2n-2} \in \overset{\frown}{L}\ (X_1, X)\}$, где $\overset{\frown}{L}\ (A, B)$ обозначает дугу кривой L от точки A к B в соответствии с ориентацией L. Можно показать, что точки X_1, X_2, ... X_{2n-2} можно выбрать так, что множество $\uparrow \{X_1, \dots, X_{2n-2}\}$ равномощно множеству S. Обозначим $S_1 := X_1$, ... $S_n := X_n$, $R_n := X_{n+1}$, ... $R_3 := X_{2n-2}$. Выберем $R_1, R_2 \in \uparrow \{X_1, \dots, X_{2n-2}\}$ и выполним протокол распределения укрытий с роботами, находящимися в точках R_1, ... R_N, и укрытиями в точках S_1, ... S_n. Тогда в силу того, что кривая L ограничивает выпуклое множество, в результате работы

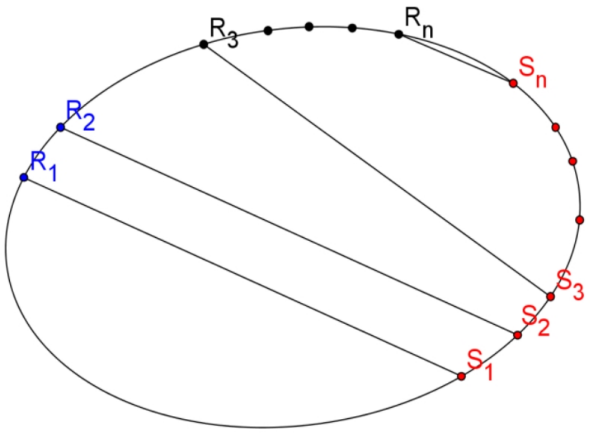

Рис. 1. Иллюстрация к доказательству теоремы 4

протокола роботу R_i будет поставлено в соответствие укрытие S_i для всех $3 \le i \le n$ (см. рис. 1). При этом роботу R_1 будет поставлено в соответствие укрытие S_1 тогда и только тогда, когда R_2 находится на кривой L между R_1 и R_3 (см. рис. 1). Введя на кривой L координаты, можно убедиться, что работа такого протокола эквивалентна сужению протокола сравнения на множество координат точек из $\uparrow\{X_1, \ldots, X_{2n-2}\}$. Поскольку это множество имеет ту же мощность, что и множество S, то (в силу утверждения 4) протокол распределения укрытий не может быть ограниченным. Если, кроме того, множество S более чем счётно, то (в силу утверждения 4) протокол распределения укрытий не может быть ограниченным по сообщениям. ■

Теорема 5. *Пусть π — произвольный протокол распределения укрытий между $n > 1$ роботами, входные данные которого берутся из открытого подмножества $D \subseteq \mathbb{R}^k$, $k \ge 2$, а $f : D^n \to \mathbb{R}$ — произвольная непрерывная функция. Тогда π не может иметь f в качестве ограничивающей функции.*

Доказательство. Предположим противное: пусть f является ограничивающей функцией для числа сообщений во время работы протокола π. Как обычно, обозначим через f^- “обратную функцию” для f. (Разумеется, в общем случае f^- не функция, а отношение.) Возьмём такое $N > 0$, что $f^-(N) \neq \emptyset$. Тогда, в силу непрерывности f, множество $f^-((0..(N+1)))$ — открытое подмножество $D^n \subseteq (\mathbb{R}^k)^n$ и, следовательно, содержит некоторый (открытый) шар $B \subseteq D^n$. Не теряя общности можно считать, что B — шар с центром в начале координат[10]. Пусть $M \subseteq D \subseteq \mathbb{R}^k$ — (открытый) шар такой, что $M^n \subseteq B$. Такое множество M удовлетворяет условиям теоремы 4 (т.к. является шаром), но по построению M функция f на M^n ограничена $(N+1)$, что означает ограниченность сужения протокола π на множество M. Получили противоречие с утверждением теоремы 4. ■

Последняя теорема 5 может быть прокомментирована следующим образом (аналогично утверждениям 3 и 4). Протокол, основанный на щелчке, подразумевает сравнение чисел, для чего надо организовать передачу последовательно знаков десятичного разложения этих чисел. Но для любого натурального числа N в любой

[10] Достаточно применить параллельный перенос.

окрестности любого вещественного числа есть неравные числа, десятичное разложение которых совпадает на префиксе длины N. Следовательно, в любой окрестности этого вещественного числа ограничивающая функция может принимать любые значения. Однако такой комментарий о соответствии теоремы 5 "здравому смыслу" не является доказательством, т.к. основан на допущениях об использовании щелчков и методе сравнения чисел.

6. Исследование криптографического аспекта

В рамках исследования криптографического аспекта задачи RinS мы будем считать, что входные данные (координаты роботов и укрытий) берутся из некоторого конечного множества точек в пространстве $\mathbb{R}^k$, $k \geq 2$, все координаты которых — числа, представленные конечным числом разрядов какой-либо (фиксированной) позиционной системы счисления. Заметим, что в этом случае смысл протокола распределения укрытий состоит в вычислении некоторой функции координат роботов и укрытий, значением которой является искомое распределение. В силу *теоремы о конфиденциальных вычислениях* (Oblivious Transfersecure multi-party computation) [7, 10], существует способ вычисления этой функции, при котором участники не получат никаких дополнительных сведений о входных данных друг друга. К сожалению, прямое описание этого способа чрезмерно громоздко. Поэтому мы несколько упростим себе задачу: будем считать, что участники пользуются каким-то протоколом распределения укрытий, основанным на щелчках. Наша цель, таким образом, состоит в построении протокола щелчка, при котором участники бы не раскрывали друг другу никаких дополнительных сведений о своих координатах. Участников мы будем считать "получестными" (честными, но любопытными). Это означает, что они строго следуют протоколу, хотя и пытаются из сообщений друг друга извлечь как можно больше дополнительной информации. Такой подход полностью согласуется с интерпретацией участников как агентов, которые строго следуют заложенной в них программе и, кроме того, заинтересованы в правильности работы протокола.

6.1. Протокол раздельного вычисления логической функции

Известны различные протоколы конфиденциального вычисления логических (булевозначных) функций [7, 10]. Мы будем использовать следующий двусторонний протокол.

Пусть требуется вычислить некоторую логическую функцию f, часть аргументов которой известны участнику A, а часть — участнику B. При этом участники не хотят раскрывать друг другу сведений о своих входных данных. Предположим, что функция f записана с использованием только операций конъюнкции &, отрицания $\neg$ и исключающего *или* $\oplus$. Представим функцию f в виде логической схемы, на входы которой подаются аргументы функции, а на выходе получается её значение. Будем последовательно вычислять значения в узлах схемы, но не полностью, а *раздельно*: это значит, что для каждого узла i у участников получатся биты a_i и b_i такие, что значение узла равно $a_i \oplus b_i$, причём бит a_i известен только участнику A, а бит b_i — только участнику B. Для этого участники делают следующее. Для каждого

своего бита p участник создаёт два случайных бита a и b такие, что $a \oplus b = p$ и посылает один из этих битов своему напарнику. Таким образом оказываются разделены входные данные.

Теперь осталось описать, как вычислять результаты логических операций. С отрицанием и исключающим *или* всё просто: чтобы вычислить результат отрицания, надо взять отрицание соответствующего бита первого участника, а чтобы вычислить результат исключающего *или*, каждый участник должен вычислить исключающее *или* для своей части значения. Сложность возникает при вычислении конъюнкции. Пусть на входе имеются биты a_1, b_1, a_2 и b_2, а требуется получить такие биты a и b, что $a \oplus b = (a_1 \oplus b_1)\&(a_2 \oplus b_2)$. Поскольку исключающее *или* и конъюнкция — это сложение и умножение по модулю 2 соответственно, то имеем $(a + b) = (a_1 \times a_2 + a_1 \times b_2 + b_1 \times a_2 + b_1 \times b_2) \mod (2)$. Поскольку $a_1 \times a_2$ известно участнику A, а $b_1 \times b_2$ — участнику B, им теперь достаточно вычислить биты c и d такие, что $(c+d) = (a_1 \times b_2 + b_1 \times a_2) \mod (2)$; тогда можно положить $a := (a_1 \times a_2 + c) \mod (2)$ и $b := (b_1 \times b_2 + d) \mod (2)$. Биты c и d вычисляются с помощью следующего протокола. Участник A генерирует случайный бит x. Затем он полагает $s_{00} := x$, $s_{01} := (x + a_1) \mod (2)$, $s_{10} := (x + a_2) \mod (2)$, $s_{01} := (x + a_1 + a_2) \mod (2)$. С помощью протокола передачи данных "*вслепую 1 из 4*" (она же *забывчивая передача*, *oblivious transfer*) [7, 10] участник B узнаёт значение бита $s_{b_1 b_2}$. Теперь можно положить $c := x$ и $d := s_{b_1 b_2}$.

После того, как выходные биты оказываются вычислены разделённо, участникам A и B остаётся только обменяться своими частями результата вычислений.

6.2. Криптографический аспект: результаты

Утверждение 6. *Пусть $S \subset \mathbb{R}^k$ — произвольное конечное множество точек, все координаты которых — числа с фиксированным числом разрядов в некоторой фиксированной позиционной системе счисления. Тогда*

1. *существует сужение протокола простого щелчка на множество S, в котором агенты не сообщают друг другу свои координаты;*

2. *существует сужение протокола щелчка со сравнениями на множество S, в котором агенты не сообщают друг другу свои расстояния до укрытий.*

Доказательство. Рассмотрим подробно протокол простого щелчка. (Протокол щелчка со сравнениями рассматривается аналогично.) Заметим, что пересечение двух прямолинейных маршрутов $[R_1, S_1]$ и $[R_2, S_2]$ эквивалентно конъюнкции трёх условий:

- оба отрезка лежат в одной плоскости,
- точки R_1 и S_1 разделены прямой $l(R_2, S_2)$,
- точки R_2 и S_2 разделены прямой $l(R_1, S_1)$.

Все эти три условия легко выражаются средствами аналитической геометрии в виде равенств и неравенств между (фиксированными) арифметическими выражениями от координат роботов R_i, R_j и укрытий S_i, S_j. В силу конечности множества

S и конечности представления всех координат всех точек мы можем считать эти выражения фиксированными булевскими функциями от бинарного представления координат роботов и укрытий, т.е. просто известной булевской функцией. К этой функции мы можем применить протокол раздельного вычисления, описанный выше. ■

Замечание: В доказательстве утверждения 6 описан очень неэффективный метод определения булевской функции для раздельного вычисления. Более эффективный метод описан в работе [1].

7. Заключение

В данной статье мы представили и доказали корректность масштабируемого параметрического мультиагентного алгоритма для задачи о роботах в пространстве RinS. Этот алгоритм принадлежит к классу *волновых* распределённых алгоритмов [5], поскольку удовлетворяет следующим трём свойствам.

Завершение: Алгоритм завершается в силу убывания суммы расстояний от роботов до выбранных укрытий.

Решение: В алгоритме событием принятия решения является осознание того, что можно двигаться к укрытиям.

Зависимость: Поскольку каждый из роботов общается с остальными и это общение влияет на дальнейшие шаги, то все эти события в жизни каждого робота влияют на событие решения.

Нами была получена и обоснована верхняя оценка сложности этого алгоритма, зависящая от геометрии (взаимного расположения) роботов и укрытий.

В рамках информационной постановки задачи была доказана теорема, утверждающая, что для широкого класса множеств координат роботов и укрытий количество передаваемых сообщений не может быть ограничено никакой непрерывной функцией координат. В рамках криптографической постановки задачи были построены протоколы, позволяющие роботам проверить, пересекаются ли их текущие пути, без раскрытия дополнительной информации о своём положении.

Однако для нас по-прежнему остаётся открытым вопрос о сложности как функции от числа роботов и укрытий. Этот вопрос впервые был сформулирован в [2] для задачи про роботов на Марсе MRP. В цитируемой статье была предложена оценка числа раундов для каждого робота $O(T(n))$, где $T(n)$ — сложность эвристического алгоритма Дейкстры [17]. Но, к сожалению, нам неизвестна верхняя граница для $T(n)$, меньшая $O(n!)$.

Другой вопрос, требующий дополнительного исследования — это вопрос о существовании *анонимного* протокола для задачи о роботах в пространстве RinS, когда во время переговоров роботы (даже косвенно) не раскрывают друг другу свои “имена”. Алгоритм SMEx подразумевает, что роботы во время общения знают имена друг друга; это необходимо, чтобы исключить повторное общение пары роботов во время раунда.

Но главный интерес для нас представляет следующее обобщение задачи о роботах в пространстве:

> *Дискретный ресурс заранее разделён на фиксированное число частей, каждая из которых неделима далее; есть также некоторое число агентов, каждый из которых претендует ровно на одну часть ресурса; задача состоит в определении протокола попарных переговоров между агентами (основанного на индивидуальных мнениях агентов), благодаря которому каждый агент в некоторый момент времени будет знать, какая часть ресурса принадлежит ему, что никто из агентов не конкурирует с ним из-за этой части в этот момент и что такое распределение ресурса удовлетворяет его (агента) представлениям о рациональности.*

Эту задачу мы будем называть *мультиагентной задачей распределения дискретных ресурсов* (MADAP — Multi-Agent Discrete Assignment Problem).

Задача о роботах в пространстве RinS является частным случаем (вариантом) MADAP, в которой роль ресурсов выполняют укрытия, а рациональность — отсутствие столкновений. Другой частный случай MADAP — следующая задача о *рациональных агентах у разрезанного пирога* (Rational Agents Near Cut Cake — RANCC), описанная и исследованная в работе [18]:

> *На базарной площади $m > 0$ покупателей и $n \geq m$ продавцов. Продавцы ведут себя пассивно, каждый из них выставил на продажу единичный неделимый товар ("кусок пирога") по индивидуальной цене для разных покупателей, но ему всё равно, кому продать свой товар (или вообще не продать). А вот каждый покупатель является рациональным агентом, которому нужно приобрести ровно один товар, причём ему заранее известны предложения цены от всех продавцов ему лично. Покупатели могут самостоятельно выбирать или менять выбор продавца (flip), вести переговоры в парах, однако во всех своих действиях каждый покупатель всегда соблюдает свою выгоду. Покупатель может совершить сделку с каким-либо из продавцов только тогда, когда он знает, что никто другой никогда не будет претендовать на сделку с этим продавцом. Задача: разработать мультиагентный алгоритм для покупателей, который позволит каждому покупателю рано или поздно совершить покупку.*

Для этой задачи в работе [18] был разработан и обоснован масштабируемый параметрический мультиагентный алгоритм, близкий к алгоритму SMEx, использованному в настоящей статье. Фактически разница состоит только в том, как агенты разрешают конфликт: в алгоритме SMEx агенты обмениваются укрытиями как только убывает сумма расстояний, а в алгоритме из работы [18] — они "разыгрывают" кусок пирога, из-за которого произошёл конфликт. Здесь "разыгрывают" означает вычисление равновесия по Нэшу в смешанных стратегиях в некой игре с полной информацией.

Следующий пример варианта мультиагентной задачи распределения дискретных ресурсов — это *задача о конкуренции за процессоры*:

Есть ресурсный центр, состоящий из $n > 1$ "процессоров" (возможно, различных), пула заданий, ожидающих выделения процессора (пул пополняется из потока входящих заданий), и монитора загрузки процессоров и состояния пула. Монитор проставляет вновь поступившим заданиям тэг времени поступления, назначает/выделяет свободный процессор, если задание из пула затребовало этот процессор, и удаляет из пула задания, которым был выделен процессор. Каждое задание является агентом. Оно знает индивидуальный "список полезности" процессоров (представленный частичным порядком на подмножестве процессоров, способных его выполнить) и время ожидания: пока время ожидания не истекло, задание может запросить только наиболее полезный для него свободный процессор, но после истечения времени ожидания задание запрашивает первый из свободных процессоров из списка полезности; однако задание может запросить выделить ему процессор только тогда, когда знает, что никакая другая задача не требует этот процессор в данный момент времени. Задача: разработать мультиагентный алгоритм, который гарантирует, что любое задание из потока обязательно получит процессор для выполнения.

В ближайшее время мы планируем представить вариант масштабируемого (но не параметрического) мультиагентного алгоритма для задачи о конкуренции за процессоры. Этот алгоритм также близок к алгоритму SMEx, но отличается тем, что каждый агент-задание знает только о заданиях, которые попали в пул раньше него, но ещё не получили процессора.

Как видно из сказанного выше о специальных случаях мультиагентной задачи распределения дискретных ресурсов MADAP, все они могут быть решены с использованием вариантов алгоритма SMEx. Поэтому естественно возникает вопрос о применимости этого алгоритма для решения самой задачи MADAP.

Список литературы

1. Бернштейн А.Ю. Информационный и криптографический аспекты задачи о роботах на Марсе // Ершовская конференция по информатике 2011: Труды третьего семинара "Знания и Онтология *ELSEWHERE* 2011". Новосибирск: Прайс-Курьер, 2011. С. 35–46 (Bernstein A.Yu. Informatsyonny i kriptografichesky aspekty zadachi o robotah na Marse // Ershovskaya konferentsiya po informatike 2011: Trudy tretego seminara Znaniya i Ontologiya *ELSEWHERE*. Novosibirsk: Prais-Kurer, 2011. P. 35–46 [in Russian]).

2. Бодин Е.В., Гаранина Н.О., Шилов Н.В. Задача о роботах на Марсе (мультиагентный подход к задаче Дейкстры) // Моделирование и анализ информационных систем. 2011. Т. 18, №2. С. 111–126 (Bodin E.V., Garanina N.O., Shilov N.V. Mars Robot Puzzle (a Multiagent Approach to the Dijkstra Problem) // Modeling and analysis of information systems. 2011. V. 18, №2. P. 111–126 [in Russian]).

3. Гаранина Н.О., Шилов Н.В. Верификация комбинированных логик знаний, действий и времени в моделях // Методы и модели современного программирования. Системная информатика, вып. 10. Новосибирск: изд-во Сибирского Отделения РАН, 2006. С. 114–173 (Garanina N.O., Shilov N.V. Verifikatsiya kombinirovannyh logic znany, deistvy

i vremeny v modelyah // Metody i modeli sovremennogo programmirovaniay. Sistemnaya informatica. № 10. Novosibirsk: izdatelstvo Sibirskogo Otdelenya RAN, 2006. P. 114–173 [in Russian]).

4. Мюллер Д. Общественный выбор III. М.: Гос. ун-т Высшая школа экономики, Институт "Экономическая школа", 2007 (English trans.: Mueller D.C. Public Choice III. Cambridge: Cambridge University Press, 2003).

5. Тель Ж. Введение в распределенные алгоритмы. М.: МЦНМО, 2009 (English trans.: Tel G. Introduction to Distributed Algorithms. Cambridge: Cambridge University Press, 2003).

6. Таненбаум Э., ван Стеен М. Распределенные системы. Принципы и парадигмы. СПб.: Питер, 2003 (English trans.: Andrew S. Tanenbaum A.S., van Steen M. Distributed Systems: Principles and Paradigms. New Jersey: Pearson Prentice Hall, 2007).

7. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. М.: Издательство ТРИУМФ, 2002 (English trans.: Schneier B. Applied Cryptography: Protocols, Algorithms, and Source Code in C. New York: John Wiley & Sons, 1996).

8. Яковлев К.С. Алгоритмы планирования перемещений на плоскости: Слайды презентации, 2010. Доступна на `www.raai.org/news/pii/ppt/yakovlev.ppt`. (Yakovlev K.S. Algorithmy planyrovanya peremescheny na ploskosti: Presentation, 2010. Available at www.raai.org/news/pii/ppt/yakovlev.ppt [in Russian]).

9. Fagin R., Halpern J.Y., Moses Y., Vardi M.Y. Reasoning about Knowledge. MIT Press, 1995.

10. Goldreich O. Foundations of Cryptography — A Primer // Foundations and Trends in Theoretical Computer Science. 2005. V. 1, № 1. P. 1–116.

11. Halpern J., O'Neill K. Anonymity and Information Hiding in Multiagent Systems // Journal of Computer Security. 2005. V. 13, № 3. P. 483–514.

12. Hughes D., Shmatikov V. Information Hiding, Anonymity and Privacy: a Modular Approach // Journal of Computer Security. 2004. V. 12, № 1. P. 3–36.

13. de Jong S., Tuyls K., Verbeeck K. Fairness in Multiagent Systems // The Knowledge Engineering Review. 2008. V. 23, № 2. P. 153–180.

14. de Jong S. Fairness in Multi-Agent Systems. PhD Thesis. Maastricht University, 2009. Доступна на `http://dl.dropbox.com/u/1505034/website/optima/thesis(17x24).pdf`.

15. LaValle S.M. Planning Algorithms. Cambridge University Press, 2006.

16. Manna Z., Pnueli A. The Temporal Logic of Reactive and Concurrent Systems: Specification. Springer, 1992.

17. Shilov N.V., Shilova S.O. Etude on theme of Dijkstra // ACM SIGACT News. 2004. V. 35, № 3. P. 102–108.

18. Shilov N.V., Garanina N.O. Rational Agents at the Marketplace // Proceedings of Workshop on Concurrency, Specification and Programming CS&P'2011 (Pultusk, Poland, September 28-30, 2011). Bialystok University of Technology, 2011. P. 465–476.

19. Wooldridge M. An Introduction to Multiagent Systems. Jhon Willey & Sons, 2002.

"Robots in Space" Multiagent Problem: Complexity, Information and Cryptographic Aspects

Bernstein A.Yu., Shilov N.V.

Novosibirsk State University, Pirogova Str., 2, Novosibirsk, 630090, Russia
A.P. Ershov Institute of Informatics Systems SB RAS
prospect Akademika Lavrentjeva, 6, Novosibirsk, 630090, Russia

Keywords: multiagent systems and algorithms, location assignment problem, anonymity, scalability, safety and progress properties, algorithm verification.

We study a multiagent algorithmic problem that we call *Robot in Space* (RinS): *There are $n \geq 2$ autonomous robots, that need to agree without outside interference on distribution of shelters, so that straight pathes to the shelters will not intersect.* The problem is closely related to *the assignment problem* in Graph Theory, to *the convex hull problem* in Combinatorial Geometry, or to *the path-planning problem* in Artificial Intelligence. Our algorithm grew up from a local search solution of the problem suggested by E.W. Dijkstra. We present a multiagent anonymous and scalable algorithm (protocol) solving the problem, give an upper bound for the algorithm, prove (manually) its correctness, and examine two communication aspects of the RinS problem — the informational and cryptographic. We proved that (1) there is no protocol that solves the RinS, which transfers a bounded number of bits, and (2) suggested the protocol that allows robots to check whether their paths intersect, without revealing additional information about their relative positions (with respect to shelters). The present paper continues the research presented in *Mars Robot Puzzle (a Multiagent Approach to the Dijkstra Problem)* (by E.V. Bodin, N.O. Garanina, and N.V. Shilov), published in *Modeling and analysis of information systems*, 18(2), 2011.

Сведения об авторах:
Бернштейн Антон Юрьевич,
Новосибирский государственный университет,
студент механико-математического факультета;
Шилов Николай Вячеславович,
Институт систем информатики им. А.П. Ершова СО РАН,
канд. физ.-мат. наук, старший научный сотрудник,
доцент Новосибирского государственного университета
и Новосибирского государственного технического университета.