

УДК 519.712

РАСПРЕДЕЛЕНИЕ РЕСУРСОВ В КОНТЕКСТЕ МУЛЬТИАГЕНТНЫХ СИСТЕМ

Н.В. Мальковский

Санкт-Петербургский государственный университет
Россия, 199034, Санкт-Петербург, Университетская наб., 7-9
E-mail: malkovskynv@gmail.com

О.Н. Граничин

Санкт-Петербургский государственный университет
Россия, 199034, Санкт-Петербург, Университетская наб., 7-9
E-mail: Oleg_granichin@mail.ru

К.С. Амелин

Санкт-Петербургский государственный университет
Россия, 199034, Санкт-Петербург, Университетская наб., 7-9
E-mail: konstantinamelin@gmail.com

Ключевые слова: распределение ресурсов, оптимизация, мультиагентные системы

Аннотация: В статье предлагается в общих чертах описание различных вариантов задачи распределения ресурсов в контексте мультиагентных систем, областей ее применимости и существующих методов решения этой задачи. Приведена постановка задачи, охватывающая как классические задачи распределения ресурсов, так и практические задачи распределения ресурсов в больших информационных системах.

1. Введение

Можно с уверенностью утверждать, что задача распределения ресурсов является одной из фундаментальных задач для человечества: от эффективного распределения своего собственного времени конкретным индивидуумом до распределения общественных усилий между различными видами деятельности.

Во второй половине XX-ого века развитие способов решения этой задачи (как и многих других) получило мощный импульс за счет развития вычислительных устройств. В статье анализируются основные разновидности задачи распределения ресурсов, существующие модели и методы решения, развиваются более ранние постановки и идеи из [1]

2. Постановка задачи

2.1. Стационарная задача распределения ресурсов

Пусть есть n заказов и m типов ресурсов. Рассмотрим один из способов формализации задачи о распределении ресурсов между заказами с целью получения максимальной выгоды.

Обозначим r_j — объем (количество) j -го ресурса, $j = 1, \dots, m$.

Будем считать, что для каждого заказа $i = 1, \dots, n$, задана функция $c_i : \mathbb{R}_+^m \rightarrow \mathbb{R}$, задающая возможную прибыль при использовании $x_{i,1}, x_{i,2}, \dots, x_{i,m}$ ресурсов соответствующего типа.

С математической точки зрения задача распределения ресурсов заключается в нахождении такой матрицы распределения ресурсов $X \in \mathbb{R}^{nm}$, которая максимизирует функционал

$$(1) \quad C(X) = \sum_{i=1}^n c_i(x_{i,1}, x_{i,2}, \dots, x_{i,m}) \rightarrow \max$$

при условии выполнения ограничений на общее количество ресурсов

$$(2) \quad \sum_{i=1}^n x_{i,j} \leq r_j, \quad j = 1, \dots, m,$$

и их неотрицательности

$$(3) \quad x_{i,j} \geq 0, \quad i = 1, \dots, n, \quad j = 1, \dots, m.$$

Дополнительным условием может быть требование целочисленности $x_{i,j} \in \mathbb{Z}_+$, $i = 1, \dots, n$, $j = 1, \dots, m$. Ограничения на общее количество ресурсов и их неотрицательность иногда удобнее записать в общем виде

$$(4) \quad F(X) \leq 0,$$

где $F : \mathbb{R}^{mn} \rightarrow \mathbb{R}^k$ — некоторая вектор-функция. Для условий (2) и (3) $k = n(1 + m)$ и вектор-функция $F(X)$ линейная. Если X представить в виде вектора $(x_{1,1}, x_{2,1}, \dots, x_{n,1}, x_{2,1}, \dots, x_{n,m})^T$ размерности mn , то $F(X) = AX - R$, и условие (4) принимает вид

$$(5) \quad AX - R \leq 0,$$

где матрица A размерности $(m(1 + n) \times mn)$, состоящая из элементов $a_{i,j}$, имеет вид

$$(6) \quad a_{i,j} = 1, \quad i = 1, \dots, m, \quad j = (i - 1)n + 1, \dots, in,$$

$$(7) \quad a_{i,i-m} = -1, \quad i = m + 1, \dots, m(1 + n),$$

$$(8) \quad a_{i,i-m} = 0, \quad \text{в остальных случаях,}$$

а вектор R размерности $(m(1 + n))$ состоит из r_j в первых m строках и нулей в остальных.

Важный частный случай — выпуклые “задачи”, когда функции $C(X)$ и $F(X)$ выпуклы.

Среди выпуклых задач наиболее хорошо изучен линейный случай задачи (1), когда функции прибыли $c_i(\cdot)$ линейно зависят от $x_{i,j}$, $i = 1, 2, \dots, n$, $j = 1, 2, \dots, m$. Функционал (1) при этом принимает вид

$$(9) \quad \sum_{i=1}^n \sum_{j=1}^m c_{i,j} x_{i,j} \rightarrow \max.$$

В разных относительно тривиальных частных случаях можно получить точное аналитическое решение различными методами анализа, дискретной математики и смежных дисциплин. Классическими примерами таких задач служат *задача о назначениях* (или же задача нахождения максимального взвешенного паросочетания) и *транспортная задача*.

Задача о назначениях формулируется довольно просто: дана квадратная матрица C размером $n \times n$ с элементами из \mathbb{R} , нужно найти такую перестановку $\mathbf{p} = \{p_1, p_2, \dots, p_n\}$ порядка n : $p_i \in \mathbb{N}$, $p_i \leq n$, чтобы сумма

$$\sum_{i=1}^n c_{i,p_i}$$

была максимальной. Таким образом, если мы положим

- $m = n$,
- $r_j = 1$, $j = 1, 2, \dots, n$,

и добавим условие

- $x_{i,j} \in \mathbb{Z}_+$, $i = 1, \dots, m$, $j = 1, \dots, n$

то получим сведение к стационарной задаче (9) распределения ресурсов. Задача о назначениях полиномиально разрешима, традиционное решение (венгерский алгоритм [2]) имеет асимптотическую сложность $O(n^3)$.

Транспортную задачу можно рассматривать, как некоторое обобщение задачи о назначениях. Задан типовой ресурс, который добывается в n шахтах, и в котором заинтересованы m фабрик, желающих получить ресурс в количестве z_i , $j = 1, 2, \dots, m$. Задана матрица стоимостей перевозок единичного объема ресурса F размером $m \times n$ с элементами из \mathbb{R} . Задача заключается в минимизации затрат на транспортировку ресурсов, т. е. нужно найти такую матрицу распределения ресурсов X , которая бы минимизировала функционал (9) с дополнительным ограничением удовлетворенности всех заказов:

$$(10) \quad \sum_{j=1}^m x_{i,j} = z_i, \quad i = 1, 2, \dots, n,$$

и выполнения условий ограниченности и неотрицательности ресурсов (2), (3).

Условия (10), (2) и (3) могут быть сведены к общему виду (5) добавлением n строк в начало матрицы A

$$(11) \quad a_{i,j} = 1, \quad i = 1, \dots, n, \quad j = i, i+n, \dots, in(m-1),$$

$$(12) \quad a_{i,j} = 0, \quad i = 1, \dots, n, \quad \text{для остальных } j,$$

и z_1, z_2, \dots, z_n в начало вектора R .

Как и задача о назначениях, транспортная задача так же является полиномиально разрешимой.

Более сложным примером является задача о составлении расписания использования аудиторий учебного заведения. Ресурсами в этом случае являются аудитории вместе со временным интервалом проведения занятий. Соответствующие временные интервалы традиционно заранее урегулированы и дискретизированы. Заказами являются занятия, которые Формулировка требований к расписанию может варьироваться от многих факторов в зависимости от политики учебного заведения. Наиболее распространенными из них являются такие факторы, как

- общие физические ограничения (один человек не может находиться на разных занятиях одновременно, программирование должно проходить в компьютерных классах и пр.),
- равномерное (или же наоборот) распределение нагрузки по неделе;
- отдельные пожелания от различных индивидов (например, преподавателю жизненно необходимо, чтобы все его занятия проходили в один день).

В достаточно простых случаях эту задачу можно свести к решению проблемы о раскраски графа (причем существуют различные идеи сведения, приводящие задаче *вершинной* раскраске графа, и приводящие к *реберной* раскраске графа). При решении достаточно распространены генетические методы, рандомизированные алгоритмы, линейное программирование, нейронные сети (подробная классификация и описание применимости различных методов можно найти в [3]).

2.2. Помехи и неопределенности

На практике работоспособность традиционных алгоритмов решения задач о распределении ресурсов между заказами существенно снижается из-за невозможности учесть влияние различных неопределенностей (очень сложно учесть “неточность” реальных систем). Эти неточности могут появляться по разным причинам: от ошибок округления или случайных помех при измерении интенсивности сигнала до заведомо известных контролируемых но случайных факторов. В рамках таких неопределенностей ставится задача оптимальности “в среднем”. Неопределенности могут иметь и более сложную природу, которая не позволяет надеяться на возможности их повторяемости и предсказуемости в каком-либо смысле.

В математическую модель неопределенности δ могут “включаться” как в функционал качества

$$(13) \quad C(X, \delta) \rightarrow \max$$

так и в набор условий-ограничений

$$(14) \quad F(X, \delta) \leq 0.$$

При этом различают статистические и минимаксные постановки задач, в которых вместо (13) рассматривается соответственно либо усредненный функционал качества, либо минимаксный.

2.3. Нестационарная задача распределения ресурсов

Добавим в рассмотрение фактор времени t , которое можно считать как дискретным, так и непрерывным. Рассмотренный ранее стационарный случай соответствует одному моменту времени.

В нестационарном случае появляются важные особенности ресурсов — ресурсы могут быть *расходуемыми* со временем (их количество ограничено на всей протяженности задачи: топливо, пища и пр.) и *репродуктивными*, количество которых ограничено на единицу времени (процессорное время, пропускная способность канала). Стоит отметить, что ресурс может иметь и оба этих свойства одновременно. Обозначим \mathbb{M}_c и \mathbb{M}_r наборы репродуктивных и расходуемых ресурсов соответственно. Пусть набор $\mathbf{r} = \{r_j, j \in \mathbb{M}_c\}$ задает ограниченность репродуктивных ресурсов в момент времени t . Для расходуемых ресурсов дополнительно введем набор $\phi = \{\phi_j, j \in \mathbb{M}_c\}$, показывающий общее количество этих ресурсов. В стационарном случае нет разницы между репродуктивными и расходуемыми ресурсами и наборы \mathbf{r} и ϕ совпадают. В общем случае набор ϕ общего количества расходуемых ресурсов может зависеть от времени $\phi(t)$, включая в рассмотрение как возможность восполнения ресурсов, так и возможность их потери.

У заказов появляются новые характеристики: время поступления в систему t_i^{in} , критичное время принятия/отклонения t_i^d , время выполнения заказа t_i^{out} , $i = 1, 2, \dots, n$.

Функция прибыли от заказа i , $i = 1, 2, \dots, n$, также становится нестационарной, появляется зависимость не только от используемых ресурсов, но и от времени выполнения заказа.

Обозначим $X(t)$ набор распределения ресурсов между заказами в момент времени t . Для нестационарного случая при конечном временном горизонте: $0 \leq t \leq T < \infty$, задачу о распределении ресурсов между заказами можно переформулировать в следующем виде:

$$(15) \quad C(X(\cdot)) = \sum_{t=0}^T \sum_{i=1}^n c_i(t, x_{i,1}, x_{i,2}, \dots, x_{i,m}) \rightarrow \max$$

в дискретном времени, или

$$(16) \quad C(X(\cdot)) = \int_{t=0}^T \sum_{i=1}^n c_i(t, x_{i,1}, x_{i,2}, \dots, x_{i,m}) \rightarrow \max$$

в непрерывном времени, при выполнении условий неотрицательности:

$$(17) \quad x_{i,j}(t) \geq 0, \quad i = 1, \dots, n, \quad j = 1, \dots, m, \quad 0 \leq t \leq T,$$

ограничений на общее количество репродуктивных ресурсов:

$$(18) \quad \sum_{i=1}^n x_{i,j}(t) \leq r_j, \quad j \in \mathbb{M}_c, \quad 0 \leq t \leq T,$$

и дополнительных ограничениях для расходуемых ресурсов:

$$(19) \quad \sum_{i=1}^n \sum_0^t x_{i,j}(s) ds \leq \phi_j(t), \quad j \in \mathbb{M}_r, \quad 0 \leq t \leq T,$$

в дискретном времени, или

$$(20) \quad \sum_{i=1}^n \int_0^t x_{i,j}(s) ds \leq \phi_j(t), \quad j \in \mathbb{M}_r, \quad 0 \leq t \leq T,$$

в непрерывном времени.

Во многих нестационарных случаях возникают сложности, если задачи/ресурсы поступают в режиме реального времени, т. е. параметры задачи заранее не известны и могут меняться с течением времени.

Простой пример нестационарной задачи распределения ресурсов: Есть некоторое помещение под сдачу на аренду и n потенциальных арендаторов. Про каждого арендатора известно, что он собирается снимать помещение на протяжении интервала времени $[L_i, R_i]$ и готов заплатить за это D_i условных единиц. Таким образом нужно выбрать из всех потенциальных арендаторов несколько из них, таким образом, чтобы их запросы не пересекались по времени, и прибыль была максимальной.

Как нестационарная задача распределения ресурсов эта проблема формулируется следующим образом:

- $m = 1$,
- $r_1 = 1$, при этом $x_{i,1}(t)$ может принимать только значения 0 или 1,
- $C(X) = \sum_{i=1}^n (x_{i,1}|_{[L_i, R_i]} \equiv 1) D_i \rightarrow \max$.

Эта задача может быть очень просто решена с помощью динамического программирования с асимптотической сложностью порядка $\mathcal{O}(n \ln n)$.

Примером действительно сложной и актуальной задача, подпадающей под случай нестационарной задачи распределения ресурсов, является задача распределения процессорного времени между процессами. Ресурсом (репродуктивным) в данном случае является процессорное время одного или нескольких вычислительных устройств, которое нужно распределять между задачами в реальном времени. Необходимость решения задачи в режиме реального времени сильно усложняет как саму задачу, так и ее формализацию: как правило, функция прибыли заранее неизвестна и распределение процессорного времени зависит от поступающих задач и обычно производится на основе упрощенных факторов — приоритетов задач. Наиболее разумный критерий прибыли процесса — скорость выполнения им задач. Для операционных систем реального времени важно, чтобы некоторые задачи (в основном системные) решались быстро и не откладывались.

2.4. Распределение ресурсов в мультиагентных системах

В случае распределения ресурсов в мультиагентных системах надо учитывать их децентрализованность: далеко не всегда каждый агент такой системы может взаимодействовать с другими. В нашем случае, соответственно, не каждому заказу может быть доступен каждый из ресурсов, что формально можно задать следующим ограничением: пусть $D \in \mathbb{M}(n, m)$ — матрица смежности графа связей, $D_{ij} \in \{0, 1\}$, тогда в дополнении к (2) имеем ограничение

$$x_{i,j} \leq D_{ij} r_j.$$

В нестационарном случае, топология сети может меняться, поэтому в нестационарном случае $D_{i,j}$ зависит от времени, и, соответственно, введенное ограничение следует понимать как

$$x_{i,j}(t) \leq D_{ij}(t)r_j, \forall t.$$

Еще одной особенностью децентрализованного подхода является тот факт, что, вообще говоря, агенты непосредственно отвечают только за себя и не имеют полной информации о системе, из-за чего меняется сама суть задачи: вместо того, чтобы находить распределение, мы строим поведение агента таким образом чтобы получить это желаемое распределение.

3. Методы решения

В этой части статьи будут в общих чертах описаны наиболее распространенные модели и методы, использующиеся для решения задач, близких в постановке к задаче распределения ресурсов.

3.1. Задачи нахождения максимума функционала

В самом общем случае, нужно найти максимум некоторого выпуклого функционала

$$(21) \quad C(x) \rightarrow \max, C : \mathbb{R}^n \rightarrow \mathbb{R}$$

Подавляющее большинство численных методов основаны на итеративной процедуре построения последовательности оценок θ_n , сходящейся к искомому экстремуму функционала. В целом, такие процедуры можно описать как выбор некоторых функций $\alpha_n(\cdot)$

$$(22) \quad \theta_n = \theta_{n-1} + \alpha_n(C, \theta_{n-1}, \theta_{n-2}, \dots, \theta_0),$$

или даже

$$(23) \quad \theta_n = \theta_{n-1} + \alpha_n(C, \theta_{n-1})$$

Традиционными алгоритмами такого типа являются *метод Ньютона*, имеющий вид:

$$\theta_n = \theta_{n-1} - |\nabla^2(\theta_{n-1})|^{-1} \nabla C(\theta_{n-1}),$$

и *градиентный спуск*:

$$\theta_n = \theta_{n-1} + \alpha_n \nabla C(\theta_{n-1}).$$

Несмотря на свою теоретическую эффективность, метод Ньютона редко применяется в таком из-за необходимости вычисления обратной матрицы второй производной. В методе градиентного спуска этот множитель заменятся либо на постоянный скаляр, либо на некоторую последовательность скаляров.

Так же стоит отметить, что градиентный метод требует вычислений производной функции C . На практике, обычно, такой возможности нет, в связи с чем приходится использовать численные методы для оценки производной, что сильно влияет на сходимость метода, особенно в случае, когда измерения доступны с некоторыми

неконтролируемыми помехами В некоторых задачах на практике вместо градиента удается взять вектор, который только “в среднем” совпадает с градиентом. Такие методы принято называть *псевдоградиентными*. Подробное описание различных методов решения можно найти, например, в [5]. Рандомизированные версии алгоритмов изучены в [6].

Методы решения задач выпуклой оптимизации описаны в [7].

Для максимизации линейных функционалов при линейных ограничениях используются методы линейного программирования [8], методы решения линейных матричных неравенств и задач полуопределенного программирования [9, 10]. В задачах с выпуклыми ограничениями при неопределенностях (14) в последнее время активно применяется сценарный подход [11].

Во всех перечисленных задачах явно или неявно предполагается условие выпуклости или же хотя бы отсутствие нескольких локальных экстремумов у функционала. Если это условие не выполняется, то могут быть применимы метод отжига [12] и генетические алгоритмы [13]. В целом, метод отжига является модификацией метода случайного поиска, который локально с некоторой вероятностью может принять менее оптимальное состояние. Генетические алгоритмы же скорее применяются к задачам имеющим немного другую структуру, однако, сравнивая с методом случайного поиска, их можно характеризовать как алгоритмы, получающие на каждом шаге набор приближений.

3.2. Задачи планирования

Традиционная теория планирования рассматривает общую задачу распределения работ по вычислительным устройствам. Достаточно обширный класс задач планирования входит в описанную задачу распределения ресурсов. Единственным видом ресурсов в таких задачах являются вычислительные ресурсы. Далее приведен пример типичной задачи планирования:

Есть m вычислительных устройств и n заказов. Про каждый заказ известны: T_i — время поступления заказа, V_i — плата за выполнение заказа и D_i — дедлайн заказа. Если заказ выполнен в период между временем поступления и дедлайном, то мы получаем соответствующую выгоду (предполагается, что мы не можем начинать выполнять заказ до его поступления). Так же известно, что вычислитель i может выполнить заказ j за время $\tau_{i,j}$ (заказ должен быть выполнен на одном из вычислителей). Таким образом, формально, нужно максимизировать следующий функционал:

$$c_i = \prod_{j=1}^m \left(\left[\int_{T_i}^{D_i} x_{i,j}(t) dt \right] \geq \tau_{j,i} \right) * V_i$$

с ограничением на процессорное время:

$$\sum_{i=1}^n x_{i,j}(t) \leq 1, \forall j, t$$

Опять же, стоит отметить, что часто такие задачи нужно решать в режиме реального времени, т. е. о заказе мы узнаем только, когда он поступает (однако, для каждого конкретного момента времени решение этой задачи может дать оптимальную стратегию действия “на данный момент”). Подробная классификация и методы решения таких задач даны в [4].

В [14] показано применение мультиагентных технологий для решения классической задачи о восьми ферзях. Задача состоит в том, чтобы расставить 8 ферзей на стандартной шахматной доске 8×8 таким образом, чтобы ни один ферзь “не бил” другого. Такую задачу можно рассматривать как задачу распределения ресурсов: ресурсами являются клетки шахматной доски, имеется 8 идентичных заказов, каждый из которых получает условную единицу выгоды только в том случае, если ему предоставляют все клетки, находящиеся на одной диагонали/вертикали/горизонтالي с некоторой клеткой (i, j) .

Хороший пример задачи распределения ресурсов в мультиагентной системе продемонстрирован в [15, 16], где в нестационарной постановке решается задача балансировки загрузки между вычислительными устройствами с помощью применения алгоритма локального голосования типа стохастической аппроксимации. Работоспособность алгоритма демонстрируется моделированием выполнения 10^6 заказов на 1024 серверах при использовании только 2048 связей.

4. Распределение ресурсов для БПЛА

В этом разделе будут описаны общие задачи, связанные с распределением ресурсов для беспилотных летательных аппаратов (БПЛА).

В общем случае предполагается, что перед нами стоит некоторая прикладная задача, для решения которой будет использоваться группа БПЛА. Такая группа образует некоторую распределенную систему (далее мы будем считать, что это все-таки мультиагентная система), в связи с чем сразу возникают такие классические задачи, как распределение информации между БПЛА, целостность системы и прочие. Сразу стоит отметить, что группа БПЛА - типичный пример системы с изменяющейся топологией сети (относительно возможности взаимодействия), так как далеко не всегда БПЛА могут взаимодействовать на дальних расстояниях (или же, к примеру, передача сигнала на большое расстояние возможно, но требует больших затрат). Стоит отметить, что, вообще говоря, вероятность выхода из строя БПЛА из-за внешних причин относительно высока, в силу чего необходимо принимать это во внимание (т.е. обеспечение целостности системы при выходе из строя одного или нескольких БПЛА является актуальной задачей).

Таким образом мы получаем следующие наиболее важные аспекты, присущие задачам управления группой БПЛА:

- Распределенность
- Переменная топология сети
- Возможно сильное неконтролируемое внешнее воздействие
- Относительно высокая вероятность отказа БПЛА

Прежде всего стоит выделить “системные задачи” - задачи, присутствующие в том или ином виде во всех возможных прикладных задачах управления групп БПЛА:

- Навигация - в неё входят задача передвижения по заданному маршруту и задача предотвращения взаимного столкновения.

- Обеспечение целостности системы - опять же, как было сказано ранее, БПЛА могут выходить из строя, и другие БПЛА должны уметь это определять и перераспределять задачи учитывая выход из строя одного или нескольких из беспилотников.
- Коммуникация - как централизованная, так и децентрализованная. В некоторых случаях система может быть отдана сама себе в распоряжение, после передачи ей задач. В некоторых же случаях нужно поддерживать возможность связи с диспетчером.

Фактически, уже можно провести некоторую аналогию между этими задачами и задачами распределения ресурсов: предотвращение столкновений - распределение ресурса "пространства". Обеспечения целостности системы ведет к таким задачам, как распределение данных между БПЛА.

В качестве актуальных практических задач можно привести следующие:

- Облет территории - территория, которую нужно облететь и получить её фотографии. Выбирается количество БПЛА для облета из расчета их энерговооруженности. Территория делится на количество БПЛА. Высота полета 100 метров и захватываемая территория на снимке 200*200. Таким образом ставится задача облета списка точек с координатами куда надо прилететь и сделать фотографии. При этом у БПЛА ограничено время полета (например, из-за объема топливного бака или емкости аккумулятора). Таким образом, нужно распределить области между БПЛА так, чтобы все БПЛА уложились в это ограничение (или даже минимизировать время последнего прибывшего).
- Патрулирование - задача очень похожа на предыдущую, однако облет, при необходимости, должен происходить регулярно, а так же возможно реагирование в реальном времени (например оповещение диспетчера в случае обнаружения аномалий на полученных снимках).
- Отслеживание - задача заключается в отслеживании перемещения некоторого объекта, при этом задача ставится не для отдельного БПЛА, а для всей группы, т.е. необходимо, чтобы в каждый момент времени объект находился в зоне видимости хотя бы одного из БПЛА.

5. Заключение

В заключение хотелось бы сказать пару слов об общем положении дел на данный момент. В современных тенденциях развития вычислительной техники децентрализованные подходы организации вычислительных систем становятся все более актуальными. Повышение производительности процессора за счет повышения тактовой частоты уже практически себя исчерпало, что делает актуальным повышение производительности за счет увеличения количества вычислителей, в связи с чем связано много проблем при организации их работы. Таким образом, интересными становятся подходы, изначально подразумевающие децентрализованность вычислительной системы. Общим видом таких подходов является мультиагентные системы

(далее МАС), которые в настоящее время активно развиваются. Однако стоит отметить, что из-за изначальной децентрализованности можно отметить общую тенденцию неоптимальности таких подходов из-за отсутствия всей информации о системе у отдельных агентов, что обычно компенсируется отсутствием необходимости в централизованном вычислении. Задача распределения ресурсов в таких системах сродни задаче достижения консенсуса [17].

Работа выполнена при частичной финансовой поддержке РФФИ (проект 13-07-00250-а).

Список литературы

1. Граничина Н.О. Мультиагентная система для распределения заказов // Управление большими системами: сборник трудов. 2010. № 30-1. С. 549-566.
2. Kuhn H.W. The Hungarian method for the assignment problem // Naval Research Logistics Quarterly. 1955. Vol. 2, No. 1-2. P. 83-97.
3. Schaerf A. A survey of automated timetabling // Artificial Intelligence Review. 1999. Vol. 13, No. 2. P. 87-127.
4. Pinedo M. Scheduling: Theory, Algorithms, and Systems. Springer, 2012.
5. Поляк Б.Т. Введение в оптимизацию. Наука. Гл.ред. физ.-мат. лит., 1983.
6. Граничин О.Н. Рандомизированные алгоритмы стохастической аппроксимации при произвольных помехах // Автоматика и телемеханика. 2002. № 2. С. 44-55.
7. Boyd S.P., Vandenberghe L. Convex optimization. Cambridge university press, 2004.
8. Юдин Д.Б., Гольштейн Е.Г. Линейное программирование. Теория, методы и приложения. М.: Наука, гл. ред. физ.-мат. лит. 1969.
9. Поляк Б.Т., Щербаков П.С. Рандомизированный метод решения задач полуопределенного программирования // Стохастическая оптимизация в информатике. 2006. Т. 2. С. 38-70.
10. Nesterov Y., Nemirovskii A.S., Ye Y. Interior-point polynomial algorithms in convex programming. Philadelphia : Society for Industrial and Applied Mathematics, 1994. Vol. 13.
11. Calafiore G.C., Campi M.C. The scenario approach to robust control design // IEEE Trans. Automat. Control. Vol. 51, No. 5. May 2006. P. 742-753.
12. Лопатин А. Метод отжига // Стохастическая оптимизация в информатике. 2005. Т. 1. С. 133-149.
13. Mitchell M. An Introduction to Genetic Algorithms (Complex Adaptive Systems). 1998.
14. Симонова Е.В., Скобелев П.О. Мультиагентная система решения задачи о расстановке восьми ферзей. Методические указания. ПГУТИ, 2010. 33 с.
15. Амелина Н.О. Применение протокола локального голосования для децентрализованной балансировки загрузки сети с переменной топологией и помехами в измерениях // Вестник Санкт-Петербургского университета. Серия 1: Математика. Механика. Астрономия. 2013. №3. С. 12-20.
16. Amelina N., Granichin O., Kornivets A. Local voting protocol in decentralized load balancing problem with switched topology, noise, and delays // In: Proc. of the 52nd IEEE Conference on Decision and Control. Firenze, Italy, December 10-13, 2013. P. 4613-4618.
17. Olfati-Saber R., Fax J. A., Murray R.M. Consensus and cooperation in networked multi-agent systems // Proceedings of the IEEE. 2007. Vol. 95, No. 1. P. 215-233.