

# ОБ АГЕНТНОЙ ИНТЕРПРЕТАЦИИ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

**Д.В. Булгакова**

*Физический институт имени П.Н. Лебедева РАН*  
Россия, 119333, Москва, Ленинский пр-кт, 53  
E-mail: [bulgakovadari@gmail.com](mailto:bulgakovadari@gmail.com)

**Ключевые слова:** линейное программирование, мультиагентные системы, аукцион, задача о распределении ресурсов.

**Аннотация:** В современной промышленности зачастую возникают задачи, решение которых требует больших вычислительных мощностей. Например, это может быть задача распределения ресурсов между заводами для выгодного совместного производства. Как правило, такие задачи можно сформулировать в терминах линейного программирования, что подразумевает использование системы линейных уравнений и функции, которую необходимо максимизировать. На данный момент известно несколько методов решения как общих, так и целочисленных задач линейного программирования. Однако, не все они подходят для практического применения. В данной работе представлен алгоритм для решения такой задачи линейного программирования, как задача распределения ресурсов между агентами системы. При этом в работе задача линейного программирования формулируется в терминах мультиагентных систем, и решается с их помощью. Данный подход позволяет более эффективно по сравнению с существующими алгоритмами производить перестроение системы при динамическом изменении ее коэффициентов, добавлении и удалении переменных в системе, а так же при изменении целевой функции системы.

## 1. Введение

На протяжении долгого времени во многих производственных областях возникают задачи, решение которых требует больших вычислительных мощностей. И по мере развития промышленности задачи все усложняются: в современном мире требуется динамическое и быстрое решение, а количество неизвестных в системах при этом неуклонно растет.

Задачи при этом ставятся самые разные, от распределения задач между спутниками, которые в любой момент могут выйти из строя, до распределения ресурсов между заводами для выгодного общего производства. Как правило, такие задачи можно сформулировать в терминах линейного программирования, что подразумевает появление системы линейных уравнений и функции, которую необходимо максимизировать.

Естественно, на данный момент известны различные решения для задач линейного программирования, при чем как для общих, так и для целочисленных. Однако, не все эти решения подходят для практического применения. В производственных задачах системы, как правило, имеют большую размерность, и чтобы повысить скорость работы алгоритмов, решающих эти системы, используют параллельные вычисления. Но, например, параллельную вариацию метода внутренней точки для решения общих задач линейного программирования можно использовать только для систем, имеющих определенную структуру [2, 3]. Широко известный симплекс-метод имеет несколько параллельных вариаций [1], но и он не вполне подходит для решения практических задач: с течением времени в таких задачах могут появляться новые переменные и изменяться коэффициенты, что требует полного перестроения симплекс-таблицы и нахождения решения с нуля.

Таким образом, выделим основные сложности практических задач, которые можно сформулировать в терминах линейного программирования:

- Большая размерность системы
- Динамическое добавление/удаление переменных
- Динамическое изменение коэффициентов системы
- Динамическое изменение целевой функции

Для преодоления этих сложностей в данной докладе будет рассмотрен подход, основанный на использовании мультиагентных систем. Задача линейного программирования будет переформулирована в терминах мультиагентных систем и решена с помощью вариации алгоритма восходящего аукциона, который показывает высокую эффективность для распределенного поиска решения в таких системах. Также, благодаря особенностям данного подхода, алгоритм сможет оптимально реагировать на динамические изменения в системе.

## 2. Постановка задачи

Рассматривается следующая задача распределения ресурсов для производства различных продуктов:

$$\begin{aligned} c^T x &\rightarrow \max \\ Ax &\leq b \end{aligned}$$

- $N$  — количество конечных продуктов,  $N \in \mathbb{N}$
- $M$  — количество ресурсов,  $M \in \mathbb{N}$
- $x = (x_1, \dots, x_N)^T$  — вектор продуктов,  $\forall i x_i \in \mathbb{R}, x_i \geq 0$
- $c = (c_1, \dots, c_N)^T$  — вектор цен на продукты,  $\forall i c_i \in \mathbb{R}, c_i \geq 0$
- $b = (bx_1, \dots, b_M)^T$  — запасы ресурсов в экономике,  $\forall j b_j \in \mathbb{R}, b_j \geq 0$
- $A = \{a_{ij}\}_{i=1, \dots, M; j=1, \dots, N} \forall i, j a_{ij} \in \mathbb{R}, a_{ij} \geq 0$  — матрица потребностей ресурсов для производства продуктов

В данной формулировке задача о распределении ресурсов является задачей линейного программирования.

Опишем данную задачу в терминах мультиагентных систем.

Будем считать, что в системе  $N$  агентов, каждый из которых может произвести один из  $N$  продуктов. У каждого из агентов при этом имеется определенный набор ресурсов, соответствующий столбцу в матрице  $A$ . Цель задачи заключается в распределении ресурсов по агентам таким образом, чтобы получить максимальную полезность, что означает максимизацию функции  $c^T x$ .

### 3. Решение задачи и описание алгоритма

Основная идея алгоритма заключается в проведении торгов за ресурсы между агентами системы. Благодаря аукциону с определенными правилами происходит перераспределение ресурсов между агентами, что во многих случаях гарантирует нахождение оптимального решения задачи.

В алгоритме используется понятие буфера – вектора ресурсов, доступных каждому из агентов на данной итерации алгоритма. До начала алгоритма в буфере содержатся все ресурсы экономики системы.

Алгоритм состоит из следующих шагов:

- 1) Начальное перераспределение ресурсов
  - Расчет потенциального выпуска для каждого агента.
  - Выбор агента с максимальной потенциальной прибылью (что является произведением потенциального выпуска на цену продукта), передача ему необходимого количества ресурсов из буфера.
  - Формирование очереди потенциальных продавцов, в которой первым будет агент с максимальным потенциальным выпуском.
- 2) Итеративный аукцион
  - Каждый из агентов-покупателей формирует предложение агенту-продавцу.
  - Агент-продавец вычисляет потенциальную прибыль для каждого агента-покупателя, исходя из их предложений.
  - Совершается та сделка между агентом-продавцом и агентом-покупателем, в которой агент-продавец получит наибольшую выгоду. Если выгодное для агента-продавца предложение отсутствует, то сделка не совершается.

Рассмотрим каждый шаг более подробно.

На нулевом шаге подготовки алгоритма к работе происходит распределение ресурсов между агентами. Идея заключается в том, чтобы выбрать агента, способного принести наибольшую прибыль при использовании максимального количества доступных ресурсов из буфера, отдать ему часть требуемых ресурсов из буфера, и сформировать из агентов очередь потенциальных продавцов в зависимости от их возможной прибыли. При этом первым в очереди продавцов будет выбранный агент.

Одной итерацией алгоритма считается проведение торгов одним агентом из очереди потенциальных продавцов.

Идея торгов заключается в том, что каждый из агентов-покупателей формирует предложение о покупке определенной части ресурсов агента-продавца. При этом агент-продавец вычисляет возможную прибыль от продажи ресурсов каждому из агентов-покупателей и, если она положительна, продает часть своих ресурсов агенту-покупателю, при продаже ресурсов которому получит наибольшую выгоду.

Заметим, что вычисление потенциального выпуска агентами-покупателями можно проводить параллельно. Более того, эта задача оптимизации одномерная, так как максимум ищется по одной переменной, а значит, имеет невысокую сложность вычисления.

Алгоритм завершается после того, как каждый из  $N$  агентов проведет аукцион. Так как каждый аукцион по сути проводится «жадно», а последовательность продавцов определена по максимальной возможной прибыли, то есть *прибыльности* агентов, то после проведения всех аукционов будет найдено решение задачи, близкое к оптимальному. В случае, если для решения задачи достаточно только парных контрактов, то есть совершения только парных сделок между агентами, найденное решение будет оптимальным.

#### 4. Добавление агента в систему, удаление агента из системы

При добавлении нового агента в системе происходит перестроение. Идея перестроения заключается в том, чтобы определить прибыльность нового агента и поместить его в очередь потенциальных продавцов. При этом все ресурсы, распределенные между агентами, оказавшимися в очереди потенциальных продавцов после нового агента, передаются новому агенту. После этого алгоритм запускает итеративный аукцион.

Благодаря такому способу перестроения системы сокращается время на подготовительный шаг распределения ресурсов и итеративное перераспределение ресурсов между агентами, находящимися в очереди потенциальных продавцов до нового агента.

Чтобы корректно удалить агента из системы, необходимо перераспределить ресурсы, которыми он обладал в текущем состоянии системы, и заново запустить итерационный аукцион. Перераспределение происходит путем передачи всех имеющихся ресурсов удаляемого агента агенту, который в очереди потенциальных продавцов находился перед ним, или, в случае, если удаляемый агент первый в очереди потенциальных продавцов, ресурсы передадутся агенту, который был после удаляемого в очереди потенциальных продавцов. Такой способ перераспределения будет корректным, так как агенты, находящиеся в очереди потенциальных продавцов после агента, получившего ресурсы от удаляемого агента, смогут провести торги за имеющиеся у этого агента ресурсы. Если же сбрасывать ресурсы удаляемого агента в буфер, то пришлось бы проводить полное перераспределение ресурсов, что более трудоемко.

Данный способ перераспределения ресурсов при удалении агента позволяет не проводить более вычислительно сложное начальное перераспределение ресурсов и не решать задачу заново.

#### 5. Недостаточность парных контрактов

Как уже было замечено, алгоритм не всегда находит оптимальное решение. Идея алгоритма основана на заключении *парных контрактов* — то есть, совершении сделок только с двумя агентами одновременно, что дает возможность параллельно решать набор одномерных задач вместо одной многомерной. Но в некоторых задачах парных контрактов может быть недостаточно.

Недостаточность парных контрактов появляется тогда, когда суммарная потенциальная прибыль нескольких агентов превышает потенциальную прибыль агента-продавца, чья потенциальная прибыль, однако, выше потенциальной прибыли каждого из этих агентов по отдельности, и который, таким образом, стоит в очереди потенциальных продавцов впереди них. Проблема алгоритма заключается в том, что при объявлении торгов этим агентом-продавцом не состоится сделка ни с одним из этих агентов. Чтобы разрешить эту проблему, сделка должна была происходить между всеми этими агентами одновременно, что означает необходимость наличия  $N$ -контрактов.

## Список литературы

1. Панюков А.В., Горбик В.В. Параллельные реализации симплекс-метода для безошибочного решения задач линейного программирования // Вестн. ЮУрГУ. Сер. Математическое моделирование и программирование. 2011. № 9. С. 107-118.
2. RezaPeyghamiac M., Fathi Hafshejaniac S., Shirvanib L. Complexity of interior-point methods for linear optimization based on a new trigonometric kernel function // Journal of Computational and Applied Mathematics. 2014. Vol. 255. P. 74-85.
3. Durazzi C., Ruggiero V., Zanghirati G. Parallel Interior-Point Method for Linear and Quadratic Programs with Special Structure // Journal of Optimization Theory and Applications. 2001. Vol. 110, P. 289-313.
4. Shoham Y., Leyton-Brown K. Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations. New York: Cambridge University Press, 2008.
5. Rzevski G., Skobelev P. Managing Complexity. WIT Press. 2014.
6. Easley D., Kleinberg J. Networks, Crowds, and Markets: Reasoning about a Highly Connected World. New York: Cambridge University Press, 2010.