

doi 10.26089/NumMet.v??r???

УДК 519.852

О новом методе линейного программирования с использованием нейронных сетей

Н. А. Ольховский

Южно-Уральский государственный университет (национальный исследовательский университет),
Челябинск, Российская Федерация

ORCID: 0009-0008-9078-4799, e-mail: olkhovskii@susu.ru

Л. Б. Соколинский

Южно-Уральский государственный университет (национальный исследовательский университет),
Челябинск, Российская Федерация

ORCID: 0000-0001-9997-3918, e-mail: leonid.sokolinsky@susu.ru

Аннотация: В статье представлен новый итерационный метод линейного программирования, получивший название “метод поверхностного движения”. Данный метод строит на поверхности многогранника, ограничивающего допустимую область задачи линейного программирования, путь от начальной граничной точки до точки, в которой достигается оптимальное значение целевой функции. Вектор движения строится в направлении максимального увеличения/уменьшения значения целевой функции. Представлено формальное описание алгоритма, реализующего метод поверхностного движения. Доказана теорема сходимости. Описанный метод предполагает использование глубокой нейронной сети прямого распространения для определения направления движения по граням допустимого многогранника. Для этого строится многомерный локальный образ задачи линейного программирования в точке текущего приближения, который подается на вход нейронной сети. Множество размеченных прецедентов, необходимое для обучения нейронной сети может быть получено с помощью апекс-метода.

Ключевые слова: линейное программирование, метод поверхностного движения, итерационный метод, теорема сходимости, глубокая нейронная сеть.

Благодарности: Исследование выполнено при финансовой поддержке РНФ (проект № 23-21-00356).

Для цитирования: Ольховский Н.А., Соколинский Л.Б. О новом методе линейного программирования с использованием нейронных сетей // Вычислительные методы и программирование. 2023. 24, № ?. 1–22. doi 10.26089/NumMet.v??r???

A new method of linear programming using neural networks

N. A. Olkhovsky

South Ural State University (National Research University), Chelyabinsk, Russia

ORCID: 0009-0008-9078-4799, e-mail: olkhovskii@susu.ru

L. B. Sokolinsky

South Ural State University (National Research University), Chelyabinsk, Russia

ORCID: 0000-0001-9997-3918, e-mail: leonid.sokolinsky@susu.ru

Abstract: The article presents a new iterative method of linear programming, called the surface movement method. This method builds a path from the initial boundary point to the point at which the optimal value of the objective function is achieved on the surface of a polytope that restricts

the feasible region of linear programming problem. The movement vector defines the direction of the maximum increase/decrease in the value of the objective function. A formal description of the algorithm implementing the surface movement method is ed. The convergence theorem is proved. The described method involves the use of a feed forward deep neural network to determine the direction of movement along the edges of a feasible polytope. To do this, a multidimensional local image of the linear programming problem is constructed at the point of the current approximation, which is fed to the input of the neural network. The set of labeled precedents necessary for training a neural network can be obtained using the apex method.

Keywords: linear programming, surface motion method, iterative method, convergence theorem, deep neural network.

Acknowledgements: The research was supported by RSF (project No. 23-21-00356).

For citation: N.A. Olkhovsky and L.B. Sokolinsky, “A new method of linear programming using neural networks”, Numerical Methods and Programming, **24** (?), 1–22 (2023). doi 10.26089/NumMet.v??r???.

1. Введение. Быстрое развитие технологий обработки и хранения больших данных [1] привело к возникновению оптимизационных математических моделей в виде задач линейного программирования (ЛП) большой размерности [2]. Особый интерес представляют нестационарные задачи ЛП, связанные с оптимизацией нестационарных процессов [3]. В нестационарной задаче ЛП ограничения и целевая функция могут меняться динамически в ходе ее решения [4]. Следующие оптимизационные задачи могут быть сведены к нестационарным задачам ЛП: выбор оптимальных стратегий в роботтрейдинге [5, 6], оптимальное управление летательными аппаратами [7], оптимизация технологических процессов [8, 9, 10], логистические и транспортные задачи [11, 12, 13], планирование и управление производством продукции [14]. Отдельно можно упомянуть оптимизационные задачи, которые должны решаться в режиме реального времени [15]. В качестве примеров можно привести управление химическим производством, управление системой многоточечного впрыска топлива в ДВС, управление сотовыми сетями, автопилотирование, системы самонаведения ракет.

Простейший подход к решению нестационарных задач оптимизации заключается в том, что всякое изменение исходных данных воспринимается как отдельная новая задача [3]. Такой подход может быть приемлемым, когда изменения происходят относительно медленно, а оптимизационная задача решается относительно быстро. Однако для больших нестационарных оптимизационных задач решение, получаемое таким способом, оказывается далеким от оптимального в силу изменения исходных данных в процессе вычислений. В этом случае необходимо использовать алгоритмы, динамически корректирующие вычислительный процесс в соответствии с изменяющимися исходными данными. Тем самым, вычисления с измененными данными начинаются не с нуля, а используют информацию, полученную в прошлом. Такой подход применим для решения задач реального времени при условии, что алгоритм достаточно быстро отслеживает движение точки оптимума. Для больших задач ЛП последнее требование делает актуальной разработку масштабируемых методов и параллельных алгоритмов линейного программирования.

До настоящего времени одним из самых популярных способов решения задач линейного программирования является семейство алгоритмов, разработанных на основе симплекс-метода [16]. Симплекс-метод способен решать большие задачи ЛП, эффективно используя различные виды гиперразреженности [17]. Однако симплекс методу присущ ряд фундаментальных недостатков. Во-первых, на некоторых задачах симплекс-методу приходится обходить все вершины симплекса, что соответствует экспоненциальной временной сложности [18]. Во-вторых, при решении симплекс-методом очень больших задач ЛП, размерность которых превышает 50 000, часто наблюдается потеря точности [19], которую не удастся компенсировать применением даже таких мощных алгоритмов, как аффинное масштабирование или итеративное уточнение [20]. В-третьих, информационная структура алгоритмов, основанных на симплекс-методе, имеет ограниченный ресурс параллелизма, что делает невозможным их эффективное распараллеливание на больших вычислительных системах с распределенной памятью [21, 22]. Все перечисленное затрудняет использование симплекс-метода для решения нестационарных задач ЛП в режиме реального времени.

Другим популярным подходом к решению больших задач ЛП является класс алгоритмов, основанных на методе внутренних точек [23], предложенном Дикиным [24]. Эти алгоритмы способны решать задачи ЛП с миллионами переменных и ограничений [25]. Достоинством метода внутренних точек является то, что он самокорректируется и способен обеспечить высокую точность вычислений. Основными недостатками метода внутренних точек являются следующие. Во-первых, этот метод требует в качестве начального приближения некоторую внутреннюю точку допустимой области задачи ЛП. Нахождение такой точки можно свести к решению дополнительной задачи ЛП [26]. Другим способом нахождения внутренних точек является использование фейеровских приближений [27]. Во-вторых, метод внутренних точек плохо масштабируется в больших вычислительных системах кластерного типа. Известны некоторые частные случаи, когда удастся выполнить эффективное распараллеливание метода внутренних точек (см., например, [28]), но в общем случае построит эффективную параллельную реализацию этого метода для кластерных вычислительных систем не удастся. В-третьих, итерационный характер метода внутренних точек не позволяет заранее предсказать время вычислений для конкретной задачи ЛП. Указанные недостатки затрудняют применения метода внутренних точек для решения больших задач ЛП в режиме реального времени.

Новым перспективным подходом к решению оптимизационных задач, вызывающим большой интерес, являются искусственные нейронные сети [29], представляющие собой мощный универсальный инструмент, применимый практически во всех проблемных областях. Одним из первых применений искусственных нейронных сетей к решению задач ЛП была работа Хопфилда и Танка [30]. Нейронная сеть

Хопфилда-Танка состоит из двух полносвязных слоев и является рекуррентной. Число нейронов первого слоя совпадает с числом переменных задачи ЛП. Число нейронов во втором слое равно числу ограничений. Веса и смещения нейронной сети полностью определяются параметрами задачи линейного программирования. Выходной сигнал циклически подается на вход нейронной сети. Нейронная сеть работает до достижения состояния равновесия, когда выход становится равным входу. Это состояние соответствует минимум специальной энергетической функции и является решением задачи ЛП. Подход Хопфилда-Танка был развит в большом количестве работ (см., например, [31, 32, 33, 34, 35]). Главным недостатком этого подхода является то, что невозможно предсказать количество циклов работы нейронной сети, необходимое для достижения состояния равновесия. Это делает невозможным использование таких рекуррентных сетей для решения больших задач ЛП в режиме реального времени. Для этой цели более перспективными представляются глубокие нейронные сети прямого распространения. Архитектура и параметры таких сетей, как правило, не зависят от входных данных задачи. Решение получается за один проход с фиксированным временем работы сети, что обеспечивает возможность их применения для решения задач в режиме реального времени. Особый интерес представляют сверточные нейронные сети [36], ориентированные на распознавание и классификацию образов. В недавней работе [37] был предложен оригинальный метод построения образов многомерных задач ЛП, открывающий возможность использовать нейронные сети прямого распространения, включая сверточные, для их решения. Необходимо отметить, что глубокие нейронные сети требуют обучения на большом количестве размеченных прецедентов, которое может быть эффективно выполнено на графических процессорах [38]. В статье [39] предложен итерационный апекс-метод решения задач ЛП, позволяющий построить на поверхности допустимого многогранника путь в направлении максимального увеличения/уменьшения значения целевой функции, приводящий к точке оптимума. Апекс-метод принадлежит к классу проекционных методов, для которых характерна низкая линейная скорость сходимости, что делает их неприемлемыми для режима реального времени. Однако апекс-метод позволяет построить размеченное множество прецедентов, для обучения нейронных сетей прямого распространения.

В данной статье представлен новый итерационный метод решения задач ЛП, получивший название “метод поверхностного движения”. Указанный метод ориентирован на использование искусственных нейронных сетей прямого распространения, в том числе сверточных нейронных сетей. Статья организована следующим образом. В разделе 2 представлен теоретический базис, на котором основан метод поверхностного движения. Раздел 3 содержит описание метода поверхностного движения и доказательство теоремы сходимости. В разделе 4 суммируются полученные результаты и приводятся направления дальнейших исследований. Сводка обозначений, используемых в статье, приведена в разделе 5.

2. Теоретический базис. В этом разделе описывается теоретический фундамент, на котором базируется метод поверхностного движения.

Сформулируем задачу ЛП в следующем виде:

$$\bar{\mathbf{x}} = \arg \max_{\mathbf{x} \in \mathbb{R}^n} \{ \langle \mathbf{c}, \mathbf{x} \rangle \mid A\mathbf{x} \leq \mathbf{b} \}, \quad (1)$$

где $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$, $m > 1$, $\mathbf{c} \neq \mathbf{0}$. Здесь $\langle \cdot, \cdot \rangle$ обозначает скалярное произведение двух векторов. Мы предполагаем, что ограничение $\mathbf{x} \geq \mathbf{0}$ также включено в матричное неравенство $A\mathbf{x} \leq \mathbf{b}$ в форме

$$-\mathbf{x} \leq \mathbf{0}.$$

Обозначим через $\mathcal{P}$ множество индексов, нумерующих строки матрицы A :

$$\mathcal{P} = \{1, \dots, m\}. \quad (2)$$

Линейная целевая функция задачи (1) имеет вид

$$f(\mathbf{x}) = \langle \mathbf{c}, \mathbf{x} \rangle. \quad (3)$$

Вектор $\mathbf{c}$ в данном случае является градиентом целевой функции $f(\mathbf{x})$.

Пусть $\mathbf{a}_i \in \mathbb{R}^n$ обозначает вектор, представляющий i -тую строку матрицы A . Мы предполагаем, что $\mathbf{a}_i \neq \mathbf{0}$ для всех $i \in \mathcal{P}$. Обозначим через $\hat{H}_i$ замкнутое полупространство, определяемое неравенством $\langle \mathbf{a}_i, \mathbf{x} \rangle \leq b_i$, а через H_i — ограничивающую его гиперплоскость:

$$\hat{H}_i = \{ \mathbf{x} \in \mathbb{R}^n \mid \langle \mathbf{a}_i, \mathbf{x} \rangle \leq b_i \}; \quad (4)$$

$$H_i = \{\mathbf{x} \in \mathbb{R}^n | \langle \mathbf{a}_i, \mathbf{x} \rangle = b_i\}. \quad (5)$$

Определим допустимый многогранник

$$M = \bigcap_{i \in \mathcal{P}} \hat{H}_i, \quad (6)$$

представляющий множество допустимых точек задачи ЛП (1). Заметим, что M в этом случае будет замкнутым выпуклым множеством. Мы будем предполагать, что $M \neq \emptyset$, то есть задача ЛП (1) имеет решение. Обозначим через $\Gamma(M)$ множество граничных точек многогранника M ¹.

Утверждение 1. Пусть M — допустимый многогранник задачи ЛП (1), определяемый формулой (6). Тогда для любой точки $\mathbf{u} \in \Gamma(M)$ существует $\epsilon > 0$ такой, что для любой граничной точки $\mathbf{w}$, принадлежащей ϵ -окрестности $V_\epsilon(\mathbf{u})$ точки $\mathbf{u}$, найдется $i' \in \mathcal{P}$, для которого справедливо $\mathbf{u}, \mathbf{w} \in H_{i'}$:

$$\forall \mathbf{u} \in \Gamma(M) \exists \epsilon > 0 : \forall \mathbf{w} \in V_\epsilon(\mathbf{u}) \cap \Gamma(M) \exists i' \in \mathcal{P} : \mathbf{u}, \mathbf{w} \in H_{i'}. \quad (7)$$

Доказательство. Зафиксируем произвольную точку $\mathbf{u} \in \Gamma(M)$. Обозначим

$$\mathcal{P}_{\mathbf{u}} = \{i \in \mathcal{P} | \mathbf{u} \in H_i\}. \quad (8)$$

Другими словами, $\mathcal{P}_{\mathbf{u}}$ — множество индексов всех гиперплоскостей H_i , которым принадлежит точка $\mathbf{u}$. Положим

$$\mathcal{P}_{\setminus \mathbf{u}} = \mathcal{P} \setminus \mathcal{P}_{\mathbf{u}}, \quad (9)$$

то есть $\mathcal{P}_{\setminus \mathbf{u}}$ — множество индексов всех гиперплоскостей H_i , которым точка $\mathbf{u}$ не принадлежит. Определим

$$\delta = \min \{ \text{dist}(\mathbf{u}, H_i) | i \in \mathcal{P}_{\setminus \mathbf{u}} \}, \quad (10)$$

где $\text{dist}(\mathbf{u}, H_i)$ обозначает евклидово расстояние от точки $\mathbf{u}$ до гиперплоскости H_i ². По определению

$$\delta > 0. \quad (11)$$

Возьмем ϵ , удовлетворяющий условию

$$0 < \epsilon < \delta. \quad (12)$$

Тогда для любого $\mathbf{w} \in V_\epsilon(\mathbf{u}) \cap \Gamma(M)$ имеем

$$\forall i \in \mathcal{P}_{\setminus \mathbf{u}} : \mathbf{w} \notin H_i. \quad (13)$$

Поскольку $\mathbf{w}$ — граничная точка, то отсюда следует, что найдется $i' \in \mathcal{P}_{\mathbf{u}}$ такой, что

$$\mathbf{w} \in H_{i'}. \quad (14)$$

Заметим, что в силу (8) также имеем

$$\mathbf{u} \in H_{i'}. \quad (15)$$

Утверждение доказано. □

Определение 1. Целевой проекцией точки $\mathbf{z} \in \mathbb{R}^n$ на гиперплоскость H_i называется точка $\gamma_i(\mathbf{z}) \in \mathbb{R}^n \cup \{\infty\}$, определяемая формулой

$$\gamma_i(\mathbf{z}) = \begin{cases} L(\mathbf{z}) \cap H_i, & \text{если } \langle \mathbf{a}_i, \mathbf{c} \rangle \neq 0; \\ \infty, & \text{если } \langle \mathbf{a}_i, \mathbf{c} \rangle = 0, \end{cases} \quad (16)$$

где $L(\mathbf{z})$ — прямая, проходящую через точку $\mathbf{z}$ параллельно вектору $\mathbf{c}$:

$$L(\mathbf{z}) = \{\mathbf{y} \in \mathbb{R}^n | \mathbf{y} = \mathbf{z} + \lambda \mathbf{c}, \lambda \in \mathbb{R}\}. \quad (17)$$

¹Под граничной точкой множества $M \subset \mathbb{R}^n$ понимается точка в $\mathbb{R}^n$, для которой любая открытая ее окрестность в $\mathbb{R}^n$ имеет непустое пересечение как с множеством M , так и с его дополнением.

²В данном случае $\text{dist}(\mathbf{u}, H_i) = \frac{\langle \mathbf{a}_i, \mathbf{u} \rangle - b_i}{\|\mathbf{a}_i\|}$.

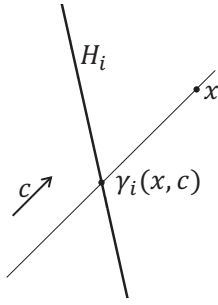


Рис. 1. Целевая проекция $\gamma_i(z)$ точки z на гиперплоскость H_i

Fig. 1. Objective projection $\gamma_i(z)$ of point z onto hyperplane H_i

Другими словами, если вектор c не параллелен гиперплоскости H_i , то целевой проекцией точки z на гиперплоскость H_i является точка пересечения этой гиперплоскости с прямой, проходящей через точку z параллельно вектору c (см. рис. 1). В случае, когда вектор c параллелен гиперплоскости H_i , целевая проекция полагается равной бесконечно удаленной точке ∞ .

Следующее утверждение предоставляет формулу для вычисления целевой проекции $\gamma_i(z)$ точки z на гиперплоскость H_i .

Утверждение 2. Пусть $\langle a_i, c \rangle \neq 0$. Тогда

$$\gamma_i(z) = z - \frac{\langle a_i, z \rangle - b_i}{\langle a_i, c \rangle} c. \quad (18)$$

Доказательство. В соответствии с (16) и (17) имеем

$$\gamma_i(z) = z + \lambda c \quad (19)$$

при некотором $\lambda \in \mathbb{R}$. С другой стороны, в соответствии с (5) имеем

$$\langle a_i, \gamma_i(z) \rangle = b_i. \quad (20)$$

Подставим правую часть формулы (19) в формулу (20) вместо $\gamma_i(z)$:

$$\langle a_i, z + \lambda c \rangle = b_i. \quad (21)$$

Отсюда

$$\lambda = -\frac{\langle a_i, z \rangle - b_i}{\langle a_i, c \rangle}. \quad (22)$$

Подставив правую часть формулы (22) вместо λ в формулу (19), получаем

$$\gamma_i(z) = z - \frac{\langle a_i, z \rangle - b_i}{\langle a_i, c \rangle} c. \quad (23)$$

Утверждение доказано. $\square$

Определение 2. Целевым смещением точки $z \in \mathbb{R}^n$ относительно гиперплоскости H_i называется скалярная величина $\beta_i(z)$, вычисляемая по формуле

$$\beta_i(z) = -\frac{\langle a_i, z \rangle - b_i}{\langle a_i, c \rangle} \|c\|. \quad (24)$$

Далее мы для краткости будем использовать термин “смещение”, подразумевая по этим “целевое смещение”. Обозначим

$$e_c = \frac{c}{\|c\|}. \quad (25)$$

Тогда формулу (18) можно переписать в виде

$$\gamma_i(z) = z + \beta_i(z) e_c, \quad (26)$$

что равносильно

$$\beta_i(z) e_c = \gamma_i(z) - z. \quad (27)$$

С учетом (25) отсюда следует

$$|\beta_i(z)| = \|\gamma_i(z) - z\|. \quad (28)$$

Таким образом, $|\beta_i(z)|$ является расстоянием от точки z до ее целевой проекции на гиперплоскость H_i .

Определение 3. Целевой гиперплоскостью $H_c(z)$, проходящей через точку z , будем называть гиперплоскость, задаваемую формулой

$$H_c(z) = \{x \in \mathbb{R}^n | \langle c, x \rangle = \langle c, z \rangle\}. \quad (29)$$

Справедливо следующее утверждение.

Утверждение 3. Зафиксируем произвольную точку $\mathbf{z} \in \mathbb{R}^n$. Тогда для любых точек $\mathbf{z}', \mathbf{z}'' \in H_c(\mathbf{z})$, $\mathbf{z}' \neq \mathbf{z}''$ справедливо

$$\langle \mathbf{c}, \gamma_i(\mathbf{z}') \rangle < \langle \mathbf{c}, \gamma_i(\mathbf{z}'') \rangle \Leftrightarrow \beta_i(\mathbf{z}') < \beta_i(\mathbf{z}'') \quad (30)$$

для всех $i \in \mathcal{P}$.

Доказательство. Поскольку $\mathbf{z}', \mathbf{z}'' \in H_c(\mathbf{z})$, а $\mathbf{c}$ является нормалью к гиперплоскости $H_c(\mathbf{z})$, справедливы следующие два равенства

$$\langle \mathbf{c}, \mathbf{z}' - \mathbf{z} \rangle = 0; \quad (31)$$

$$\langle \mathbf{c}, \mathbf{z}'' - \mathbf{z} \rangle = 0. \quad (32)$$

Следовательно

$$\langle \mathbf{c}, \mathbf{z}'' \rangle = \langle \mathbf{c}, \mathbf{z} \rangle = \langle \mathbf{c}, \mathbf{z}' \rangle. \quad (33)$$

Используя (26) и (25), отсюда получаем следующую цепочку эквивалентных неравенств.

$$\begin{aligned} \langle \mathbf{c}, \gamma_i(\mathbf{z}') \rangle < \langle \mathbf{c}, \gamma_i(\mathbf{z}'') \rangle &\Leftrightarrow \langle \mathbf{c}, \mathbf{z}' + \beta_i(\mathbf{z}')\mathbf{e}_c \rangle < \langle \mathbf{c}, \mathbf{z}'' + \beta_i(\mathbf{z}'')\mathbf{e}_c \rangle \\ &\Leftrightarrow \langle \mathbf{c}, \mathbf{z}' \rangle + \langle \mathbf{c}, \beta_i(\mathbf{z}')\mathbf{e}_c \rangle < \langle \mathbf{c}, \mathbf{z}'' \rangle + \langle \mathbf{c}, \beta_i(\mathbf{z}'')\mathbf{e}_c \rangle \\ &\Leftrightarrow \langle \mathbf{c}, \beta_i(\mathbf{z}')\mathbf{e}_c \rangle < \langle \mathbf{c}, \beta_i(\mathbf{z}'')\mathbf{e}_c \rangle \\ &\Leftrightarrow \langle \mathbf{c}, \beta_i(\mathbf{z}')\mathbf{c} / \|\mathbf{c}\| \rangle < \langle \mathbf{c}, \beta_i(\mathbf{z}'')\mathbf{c} / \|\mathbf{c}\| \rangle \\ &\Leftrightarrow \frac{\beta_i(\mathbf{z}')}{\|\mathbf{c}\|} \langle \mathbf{c}, \mathbf{c} \rangle < \frac{\beta_i(\mathbf{z}'')}{\|\mathbf{c}\|} \langle \mathbf{c}, \mathbf{c} \rangle \\ &\Leftrightarrow \beta_i(\mathbf{z}') < \beta_i(\mathbf{z}''). \end{aligned}$$

Утверждение доказано. □

Следуя [39] дадим определение рецессивного полупространства.

Определение 4. Полупространство $\hat{H}_i$ называется рецессивным, если

$$\forall \mathbf{x} \in H_i, \forall \lambda > 0 : \mathbf{x} + \lambda \mathbf{c} \notin \hat{H}_i. \quad (34)$$

Геометрический смысл этого определения состоит в том, что луч, исходящий в направлении вектора $\mathbf{c}$ из любой точки гиперплоскости, ограничивающей рецессивное полупространство, не имеет общих точек с этим полупространством, за исключением начальной. Известно [39], что следующее условие является необходимым и достаточным для того, чтобы полупространство $\hat{H}_i$ было рецессивным:

$$\langle \mathbf{a}_i, \mathbf{c} \rangle > 0. \quad (35)$$

Рецессивное полупространство обладает следующими свойствами.

Свойство 1. Пусть полупространство $\hat{H}_i$ является рецессивным. Тогда любая прямая, параллельная вектору $\mathbf{c}$, пересекает гиперплоскость H_i в единственной точке.

Данное свойство непосредственно вытекает из того факта, что гиперплоскость H_i , ограничивающая рецессивное полупространство $\hat{H}_i$ по определению не может быть параллельна вектору $\mathbf{c}$.

Свойство 2. Пусть полупространство $\hat{H}_i$ является рецессивным. Тогда

$$\mathbf{x} \in \hat{H}_i \Leftrightarrow \beta_i(\mathbf{x}) \geq 0. \quad (36)$$

Доказательство. Сначала предположим, что $\mathbf{x} \in \hat{H}_i$. Тогда в соответствии с (4) справедливо неравенство

$$\langle \mathbf{a}_i, \mathbf{x} \rangle - b_i \leq 0. \quad (37)$$

В силу (24) имеем

$$\beta_i(\mathbf{x}) = - \frac{\langle \mathbf{a}_i, \mathbf{x} \rangle - b_i}{\langle \mathbf{a}_i, \mathbf{c} \rangle} \|\mathbf{c}\|. \quad (38)$$

Принимая во внимание (35), получаем

$$\mathbf{x} \in \hat{H}_i \Rightarrow \beta_i(\mathbf{x}) \geq 0. \quad (39)$$

Теперь предположим, что $\beta_i(\mathbf{x}) \geq 0$. В соответствии с (38) это означает, что

$$\frac{\langle \mathbf{a}_i, \mathbf{x} \rangle - b_i}{\langle \mathbf{a}_i, \mathbf{c} \rangle} \|\mathbf{c}\| \leq 0. \quad (40)$$

Учитывая (35), отсюда получаем

$$\langle \mathbf{a}_i, \mathbf{x} \rangle - b_i \leq 0. \quad (41)$$

Согласно (4), отсюда следует, что

$$\mathbf{x} \in \hat{H}_i. \quad (42)$$

Таким образом

$$\beta_i(\mathbf{x}) \geq 0 \Rightarrow \mathbf{x} \in \hat{H}_i. \quad (43)$$

Свойство доказано. $\square$

Определим

$$\mathcal{I} = \{i \in \mathcal{P} \mid \langle \mathbf{a}_i, \mathbf{c} \rangle > 0\}, \quad (44)$$

то есть $\mathcal{I}$ представляет множество индексов, для которых полупространство $\hat{H}_i$ является рецессивным. Поскольку допустимый многогранник M представляет собой ограниченное множество, имеем

$$\mathcal{I} \neq \emptyset. \quad (45)$$

Положим

$$\hat{M} = \bigcap_{i \in \mathcal{I}} \hat{H}_i. \quad (46)$$

Очевидно, что $\hat{M}$ является выпуклым, замкнутым, неограниченным многогранником. Будем называть его рецессивным. Из (6) и (44) следует

$$M \subset \hat{M}. \quad (47)$$

Обозначим через $\Gamma(\hat{M})$ множество граничных точек рецессивного многогранника $\hat{M}$. Согласно утверждению 3 в [39] имеем

$$\bar{\mathbf{x}} \in \Gamma(\hat{M}), \quad (48)$$

то есть решение задачи ЛП (1) лежит на границе рецессивного многогранника $\hat{M}$.

Утверждение 4. Пусть $\hat{M}$ — рецессивный многогранник, определяемый формулой (46). Тогда для любой точки $\mathbf{u} \in \Gamma(\hat{M})$ существует $\epsilon > 0$ такой, что для любой граничной точки $\mathbf{w}$, принадлежащей ϵ -окрестности $V_\epsilon(\mathbf{u})$ точки $\mathbf{u}$, найдется $i' \in \mathcal{I}$, для которого справедливо $\mathbf{u}, \mathbf{w} \in H_{i'}$:

$$\forall \mathbf{u} \in \Gamma(\hat{M}) \exists \epsilon > 0 : \forall \mathbf{w} \in V_\epsilon(\mathbf{u}) \cap \Gamma(\hat{M}) \exists i' \in \mathcal{I} : \mathbf{u}, \mathbf{w} \in H_{i'}. \quad (49)$$

Доказательство. Доказательство идентично доказательству утверждения 1. $\square$

Определение 5. Целевой проекцией точки $\mathbf{z} \in \mathbb{R}^n$ на границу $\Gamma(\hat{M})$ рецессивного многогранника $\hat{M}$ называется точка $\hat{\gamma}(\mathbf{z})$, вычисляемая по формуле

$$\hat{\gamma}(\mathbf{z}) = L(\mathbf{z}) \cap \Gamma(\hat{M}), \quad (50)$$

где $L(\mathbf{z})$ — прямая, проходящую через точку $\mathbf{z}$ параллельно вектору $\mathbf{c}$:

$$L(\mathbf{z}) = \{\mathbf{y} \in \mathbb{R}^n \mid \mathbf{y} = \mathbf{z} + \lambda \mathbf{c}, \lambda \in \mathbb{R}\}. \quad (51)$$

Скалярную величину $\hat{\beta}(\mathbf{z}) \in \mathbb{R}$, удовлетворяющую уравнению

$$\hat{\gamma}(\mathbf{z}) = \mathbf{z} + \hat{\beta}(\mathbf{z})\mathbf{c} \quad (52)$$

будем называть целевым смещением (или кратко — смещением) точки $\mathbf{z}$ относительно границы рецессивного многогранника $\hat{M}$.

Заметим, что корректность этого определения базируется на свойстве 1. Следующее утверждение представляет формулу для вычисления целевой проекции на границу рецессивного многогранника.

Утверждение 5. Пусть задана произвольная точка $\mathbf{z} \in \mathbb{R}^n$. Положим

$$i' = \arg \min \{ \beta_i(\mathbf{z}) \mid i \in \mathcal{I} \}. \quad (53)$$

Тогда

$$\hat{\gamma}(\mathbf{z}) = \gamma_{i'}(\mathbf{z}). \quad (54)$$

Другими словами, целевая проекция точки $\mathbf{z}$ на границу рецессивного многогранника $\hat{M}$ совпадает с проекцией этой точки на гиперплоскость $H_{i'}$, имеющей минимальное смещение относительно $\mathbf{z}$.

Доказательство. Зафиксируем произвольную точку $\mathbf{z} \in \mathbb{R}^n$. В соответствии с определением 5 построим прямую, параллельную вектору $\mathbf{c}$, которая проходит через точку $\mathbf{z}$:

$$L = \{ \mathbf{y} \in \mathbb{R}^n \mid \mathbf{y} = \mathbf{z} + \lambda \mathbf{c}, \lambda \in \mathbb{R} \}. \quad (55)$$

Имеем

$$\hat{\gamma}(\mathbf{z}) = L \cap \Gamma(\hat{M}). \quad (56)$$

Согласно свойству 1 для любого $i \in \mathcal{I}$ прямая L пересекает H_i в точности в одной точке. Обозначим

$$Y = \bigcup_{i \in \mathcal{I}} \{ \mathbf{y}_i \}, \quad (57)$$

где $\mathbf{y}_i = L \cap H_i$. То есть Y — множество точек, в которых прямая L пересекает границы рецессивных полупространств. По определению 1 имеем

$$\forall i \in \mathcal{I} : \gamma_i(\mathbf{z}) = \mathbf{y}_i, \quad (58)$$

и

$$\forall i \in \mathcal{I} : \mathbf{y}_i \in H_i. \quad (59)$$

В силу (46) также имеем

$$\hat{\gamma}(\mathbf{z}) \in Y. \quad (60)$$

Это означает, что найдется $i' \in \mathcal{I}$ такой, что

$$\hat{\gamma}(\mathbf{z}) = \gamma_{i'}(\mathbf{z}). \quad (61)$$

Покажем, что

$$i' \in \arg \min \{ \beta_i(\mathbf{z}) \mid i \in \mathcal{I} \}. \quad (62)$$

Предположим противное, то есть

$$\beta_{i'}(\mathbf{z}) > \min \{ \beta_i(\mathbf{z}) \mid i \in \mathcal{I} \}. \quad (63)$$

Тогда существует $i'' \in \mathcal{I}$ такой, что

$$\beta_{i''}(\mathbf{z}) < \beta_{i'}(\mathbf{z}) \quad (64)$$

В силу (26) и (58) имеем

$$\mathbf{y}_{i'} = \mathbf{z} + \beta_{i'}(\mathbf{z}) \mathbf{e}_{\mathbf{c}}; \quad (65)$$

$$\mathbf{y}_{i''} = \mathbf{z} + \beta_{i''}(\mathbf{z}) \mathbf{e}_{\mathbf{c}}. \quad (66)$$

Отсюда

$$\mathbf{y}_{i'} = \mathbf{y}_{i''} + (\beta_{i'}(\mathbf{z}) - \beta_{i''}(\mathbf{z})) \mathbf{e}_{\mathbf{c}}, \quad (67)$$

что в силу (64) и (25) равносильно

$$\mathbf{y}_{i'} = \mathbf{y}_{i''} + \frac{|\beta_{i''}(\mathbf{z}) - \beta_{i'}(\mathbf{z})|}{\|\mathbf{c}\|} \mathbf{c}. \quad (68)$$

В соответствии с (34) и (59), из (68) следует

$$\mathbf{y}_{i'} \notin \hat{H}_{i''}. \quad (69)$$

Это означает, что

$$\mathbf{y}_{i'} \notin \hat{M}. \quad (70)$$

Принимая во внимание (58), отсюда следует

$$\hat{\gamma}(\mathbf{z}) \notin \hat{M}. \quad (71)$$

Получили противоречие с определением 5. Утверждение доказано. $\square$

Следующее утверждение предоставляет формулу для вычисления смещения точки относительно границы рецессивного многогранника.

Утверждение 6. Пусть задана произвольная точка $\mathbf{z} \in \mathbb{R}^n$. Тогда

$$\hat{\beta}(\mathbf{z}) = \frac{\langle \mathbf{c}, \hat{\gamma}(\mathbf{z}) - \mathbf{z} \rangle}{\|\mathbf{c}\|^2}. \quad (72)$$

Доказательство. В соответствии с (52) точки $\hat{\gamma}(\mathbf{z})$ и $\mathbf{z}$ находятся на нормали к гиперплоскости $H_c(\mathbf{z})$. Поэтому точка $\mathbf{z} \in H_c(\mathbf{z})$ является ортогональной проекцией точки $\hat{\gamma}(\mathbf{z})$ на гиперплоскость $H_c(\mathbf{z})$, и с учетом (29) может быть вычислена по известной формуле

$$\mathbf{z} = \hat{\gamma}(\mathbf{z}) - \frac{\langle \mathbf{c}, \hat{\gamma}(\mathbf{z}) - \mathbf{z} \rangle}{\|\mathbf{c}\|^2} \mathbf{c}. \quad (73)$$

Перепишем это в виде

$$\hat{\gamma}(\mathbf{z}) = \mathbf{z} + \frac{\langle \mathbf{c}, \hat{\gamma}(\mathbf{z}) - \mathbf{z} \rangle}{\|\mathbf{c}\|^2} \mathbf{c}. \quad (74)$$

Сопоставляя это с (52), получаем

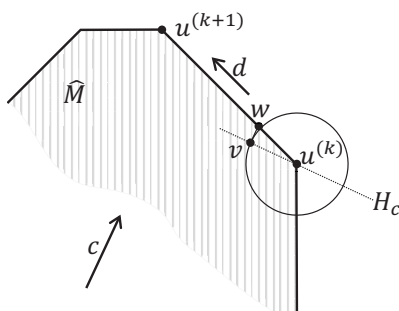
$$\hat{\beta}(\mathbf{z}) = \frac{\langle \mathbf{c}, \hat{\gamma}(\mathbf{z}) - \mathbf{z} \rangle}{\|\mathbf{c}\|^2}. \quad (75)$$

Утверждение доказано. $\square$

3. Метод поверхностного движения. Метод поверхностного движения строит на поверхности допустимого многогранника путь из произвольной граничной точки $\mathbf{u}^{(0)} \in \hat{M} \cap \Gamma(M)$ до точки $\bar{\mathbf{x}}$, являющейся решением задачи ЛП (1). Перемещение по поверхности рецессивного многогранника происходит в направлении наибольшего увеличения целевой функции. Реализация метода поверхностного движения приведена в виде алгоритма 1. Прокомментируем шаги этого алгоритма. На шаге 1 вводится начальное приближение $\mathbf{u}^{(0)}$. Это может быть произвольная граничная точка рецессивного многогранника $\hat{M}$, удовлетворяющая условию

$$\mathbf{u}^{(0)} \in \hat{M} \cap \Gamma(M). \quad (76)$$

Для получения хорошего начального приближения может применяться алгоритм, реализующий стадию Quest апекс-метода [39]. На шаге 2 счетчик итераций k устанавливается в значение 0. На шаге 3 строится n -мерный диск D , являющийся пересечением целевой гиперплоскости $H_c(\mathbf{u}^{(0)})$, проходящей через точку $\mathbf{u}^{(0)}$, и n -мерного шара $V_r(\mathbf{u}^{(0)})$ малого радиуса r с центром в точке $\mathbf{u}^{(0)}$. На шаге 4 вычисляется точка $\mathbf{v} \in D$ с максимальным смещением относительно границы рецессивного многогранника $\hat{M}$. Смещение $\hat{\beta}(\mathbf{z})$ вычисляется с помощью формулы (72). Целевая проекция $\hat{\gamma}(\mathbf{z})$, используемая в формуле (72), вычисляется с помощью формул (53) и (54). Смещение $\beta_i(\mathbf{z})$, используемое в формуле (53), вычисляется с помощью формулы (24).



Шаг 5 вычисляет точку $\mathbf{w}$, являющуюся целевой проекцией точки $\mathbf{v}$ на границу рецессивного многогранника. Шаги 6–15 реализуют основной аппроксимирующий цикл метода поверхностного движения, геометрическая интерпретация которого приведена на рис. 2. Этот цикл выполняется пока справедливо условие

$$\langle \mathbf{c}, \mathbf{w} - \mathbf{u}^{(k)} \rangle > \epsilon_f, \quad (77)$$

Рис. 2. Итерация основного цикла в методе поверхностного движения

Алгоритм 1. Метод поверхностного движения

Algorithm 1. Surface movement method

Require $\hat{H}_i = \{\mathbf{x} \in \mathbb{R}^n | \langle \mathbf{a}_i, \mathbf{x} \rangle \leq b_i\}$, $\hat{M} = \bigcap_{i \in \mathcal{I}} \hat{H}_i$, $H_c(\mathbf{z}) = \{\mathbf{x} \in \mathbb{R}^n | \langle \mathbf{c}, \mathbf{x} - \mathbf{z} \rangle = 0\}$

- 1: **input** $\mathbf{u}^{(0)}$; **assert** $\mathbf{u}^{(0)} \in \hat{M} \cap \Gamma(M)$
- 2: $k := 0$
- 3: $D := H_c(\mathbf{u}^{(0)}) \cap V_r(\mathbf{u}^{(0)})$
- 4: $\mathbf{v} := \arg \max \{\hat{\beta}(\mathbf{x}) \mid \mathbf{x} \in D\}$
- 5: $\mathbf{w} := \hat{\gamma}(\mathbf{v})$
- 6: **while** $\langle \mathbf{c}, \mathbf{w} - \mathbf{u}^{(k)} \rangle > \epsilon_f$ **do**
- 7: **assert** $\exists i \in \mathcal{I} : \mathbf{w}, \mathbf{u}^{(k)} \in H_i$ // Если не выполняется, уменьшить r
- 8: $\mathbf{d} := \mathbf{w} - \mathbf{u}^{(k)}$
- 9: $L = \{\mathbf{u}^{(k)} + \lambda \mathbf{d} \mid \lambda \in \mathbb{R}_{>0}\}$
- 10: $\mathbf{u}^{(k+1)} := \arg \max \{\|\mathbf{x} - \mathbf{u}^{(k)}\| \mid \mathbf{x} \in L \cap \Gamma(M)\}$
- 11: $k := k + 1$
- 12: $D := H_c(\mathbf{u}^{(k)}) \cap V_r(\mathbf{u}^{(k)})$
- 13: $\mathbf{v} := \arg \max \{\hat{\beta}(\mathbf{x}) \mid \mathbf{x} \in D\}$
- 14: $\mathbf{w} := \hat{\gamma}(\mathbf{v})$
- 15: **end while**
- 16: **output** $\mathbf{u}^{(k)}$
- 17: **stop**

где ϵ_f — малый положительный параметр. Шаг 7 проверяет, что существует рецессивное полупространство $\hat{H}_i$ такое, что граничные точки $\mathbf{w}$ и $\mathbf{u}^{(k)}$ лежат на гиперплоскости H_i ограничивающей данное полупространство. Это необходимо для того, чтобы перемещение происходило по поверхности рецессивного многогранника, а не через его внутреннюю область. Если указанное требование не выполняется, необходимо уменьшить радиус r n -мерного шара $V_r(\mathbf{u}^{(k)})$. Подходящий r найдется в силу утверждения 4. На шаге 8 формируется вектор $\mathbf{d}$, определяющий направление движения:

$$\mathbf{d} = \mathbf{w} - \mathbf{u}^{(k)}. \quad (78)$$

Шаг 9 строит луч L с началом в точке $\mathbf{u}^{(k)}$, сонаправленный с вектором $\mathbf{d}$. На шаге 10 определяется следующее приближение $\mathbf{u}^{(k+1)}$ как точку на луче L , лежащую на границе допустимого многогранника M и максимально удаленную от точки $\mathbf{u}^{(k)}$. По построению из (77) следует

$$\langle \mathbf{c}, \mathbf{u}^{(k)} \rangle < \langle \mathbf{c}, \mathbf{u}^{(k+1)} \rangle. \quad (79)$$

Шаг 11 увеличивает счетчик итераций k на единицу. Шаг 12 строит новый гипердиск D радиуса r с центром в $\mathbf{u}^{(k)}$:

$$D := H_c(\mathbf{u}^{(k)}) \cap V_r(\mathbf{u}^{(k)}). \quad (80)$$

Шаг 13 находит на гипердиске точку $\mathbf{v}$ с максимальным смещением. На шаге 14 вычисляется точка $\mathbf{w}$, являющаяся целевой проекцией точки $\mathbf{v}$ на границу рецессивного многогранника. На шаге 15 происходит переход на начало основного цикла **while**. Шаг 16 выводит в качестве результата последнее приближение $\mathbf{u}^{(k)}$. Шаг 17 завершает работу алгоритма. Отметим, что по построению алгоритма 1 для любого k имеет место

$$\mathbf{u}^{(k)} \in M \cap \Gamma(\hat{M}), \quad (81)$$

то есть все точки последовательности $\{\mathbf{u}^{(k)}\}$, генерируемой алгоритмом 1, одновременно лежат и на границе допустимого многогранника M , и на границе рецессивного многогранника $\hat{M}$.

Следующая теорема гарантирует сходимость алгоритма 1 к точному решению задачи ЛП (1).

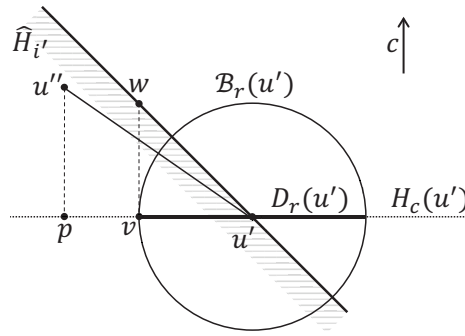


Рис. 3. Иллюстрация к доказательству теоремы 1

Fig. 3. Illustration to proof of theorem 1

Теорема 1. Пусть допустимый многогранник M задачи ЛП (1) является ограниченным непустым множеством. Тогда последовательность $\{\mathbf{u}^{(k)}\}$, генерируемая алгоритмом 1, сходится за конечное число итераций $K \geq 0$ к точному решению $\bar{\mathbf{x}}$ задачи ЛП (1) при $\epsilon_f \rightarrow 0$.

Доказательство. Сначала покажем, что количество итераций K , выполняемых в основном цикле алгоритма 1, конечно. Предположим противное, то есть алгоритм 1 генерирует бесконечную последовательность точек $\mathbf{u}^{(k)}$. Но тогда, в силу (79), мы получаем бесконечную монотонно возрастающую числовую последовательность

$$\langle \mathbf{c}, \mathbf{u}^{(0)} \rangle < \langle \mathbf{c}, \mathbf{u}^{(1)} \rangle < \langle \mathbf{c}, \mathbf{u}^{(2)} \rangle < \dots \quad (82)$$

Так как по условию теоремы допустимый многогранник M является непустым ограниченным множеством, решение $\bar{\mathbf{x}}$ задачи ЛП (1) существует. В силу (81) для всех $k = 0, 1, 2, \dots$ имеем

$$\langle \mathbf{c}, \mathbf{u}^{(k)} \rangle \leq \langle \mathbf{c}, \bar{\mathbf{x}} \rangle. \quad (83)$$

Это означает, что последовательность (82) является ограниченной сверху. По теореме Вейерштрасса монотонно возрастающая ограниченная сверху числовая последовательность имеет конечный предел, равный ее супремуму. То есть существует $K' \in \mathbb{N}$ такой, что

$$\forall k > K' : \langle \mathbf{c}, \mathbf{u}^{(k+1)} \rangle - \langle \mathbf{c}, \mathbf{u}^{(k)} \rangle < \epsilon_f. \quad (84)$$

Отсюда следует

$$\forall k > K' : \langle \mathbf{c}, \mathbf{w} \rangle - \langle \mathbf{c}, \mathbf{u}^{(k)} \rangle < \epsilon_f, \quad (85)$$

что равносильно

$$\forall k > K' : \langle \mathbf{c}, \mathbf{w} - \mathbf{u}^{(k)} \rangle < \epsilon_f. \quad (86)$$

Получили противоречие с условием (77) выполнения цикла, используем на шаге 6 алгоритма 1. Таким образом, количество итераций K , выполняемых в основном цикле алгоритма 1, конечно.

Пусть точка $\mathbf{u}' = \mathbf{u}^{(K)}$ является конечной точкой последовательности, генерируемой алгоритмом 1. В силу (81) имеем

$$\mathbf{u}' \in \Gamma(\hat{M}). \quad (87)$$

Покажем, что $\mathbf{u}'$ является решением задачи ЛП (1) при $\epsilon_f \rightarrow 0$. Рис. 3 иллюстрирует последующую часть доказательства.

Предположим противное, то есть существует точка $\mathbf{u}''$ такая, что

$$\mathbf{u}'' \in M, \quad (88)$$

и

$$\langle \mathbf{c}, \mathbf{u}'' \rangle > \langle \mathbf{c}, \mathbf{u}' \rangle. \quad (89)$$

Последнее равносильно

$$\langle \mathbf{c}, \mathbf{u}'' - \mathbf{u}' \rangle > 0. \quad (90)$$

Исходя из определения 3, вычислим ортогональную проекцию $\mathbf{p}$ точки $\mathbf{u}''$ на целевую гиперплоскость $H_c(\mathbf{u}')$, проходящую через точку $\mathbf{u}'$:

$$\mathbf{p} = \mathbf{u}'' - \frac{\langle \mathbf{c}, \mathbf{u}'' - \mathbf{u}' \rangle}{\|\mathbf{c}\|^2} \mathbf{c}. \quad (91)$$

Заметим, что

$$\|\mathbf{p} - \mathbf{u}'\| \neq 0, \quad (92)$$

так как в противном случае, в соответствии с определением 4 и формулой (87), точка $\mathbf{u}''$ не может принадлежать рецессивному многограннику $\hat{M}$, что в силу (47) противоречит предположению (88).

Выберем $r \in \mathbb{R}$, удовлетворяющий условию

$$r > 0, \quad (93)$$

для которого существует $i' \in \mathcal{I}$ такой, что

$$\hat{\gamma}(\mathbf{v}), \mathbf{u}' \in H_{i'} \cap \Gamma(M), \quad (94)$$

где

$$\mathbf{v} = \mathbf{u}' + \frac{r}{\|\mathbf{p} - \mathbf{u}'\|} (\mathbf{p} - \mathbf{u}'). \quad (95)$$

Это возможно в силу утверждения 1. Тогда в соответствии с утверждением 5 имеем

$$\hat{\gamma}(\mathbf{v}) = \gamma_{i'}(\mathbf{v}). \quad (96)$$

Обозначим

$$\mathbf{w} = \gamma_{i'}(\mathbf{v}). \quad (97)$$

Согласно утверждению 2

$$\mathbf{w} = \mathbf{v} - \frac{\langle \mathbf{a}_{i'}, \mathbf{v} \rangle - b_{i'}}{\langle \mathbf{a}_{i'}, \mathbf{c} \rangle} \mathbf{c}. \quad (98)$$

Поскольку $\mathbf{u}' \in H_{i'}$, из (5) следует

$$\langle \mathbf{a}_{i'}, \mathbf{u}' \rangle = b_{i'}. \quad (99)$$

Поэтому (98) можно переписать в виде

$$\mathbf{w} = \mathbf{v} - \frac{\langle \mathbf{a}_{i'}, \mathbf{v} \rangle - \langle \mathbf{a}_{i'}, \mathbf{u}' \rangle}{\langle \mathbf{a}_{i'}, \mathbf{c} \rangle} \mathbf{c}. \quad (100)$$

Подставив вместо $\mathbf{v}$ правую часть формулы (95), отсюда получаем

$$\mathbf{w} = \mathbf{u}' + \frac{r}{\|\mathbf{p} - \mathbf{u}'\|} (\mathbf{p} - \mathbf{u}') - \frac{\left\langle \mathbf{a}_{i'}, \mathbf{u}' + \frac{r}{\|\mathbf{p} - \mathbf{u}'\|} (\mathbf{p} - \mathbf{u}') \right\rangle - \langle \mathbf{a}_{i'}, \mathbf{u}' \rangle}{\langle \mathbf{a}_{i'}, \mathbf{c} \rangle} \mathbf{c}, \quad (101)$$

что равносильно

$$\mathbf{w} = \mathbf{u}' + \frac{r}{\|\mathbf{p} - \mathbf{u}'\|} (\mathbf{p} - \mathbf{u}') - \frac{\left\langle \mathbf{a}_{i'}, \frac{r}{\|\mathbf{p} - \mathbf{u}'\|} (\mathbf{p} - \mathbf{u}') \right\rangle}{\langle \mathbf{a}_{i'}, \mathbf{c} \rangle} \mathbf{c}. \quad (102)$$

В соответствии с (4) имеем

$$\hat{H}_{i'} = \{\mathbf{x} \in \mathbb{R}^n | \langle \mathbf{a}_{i'}, \mathbf{x} \rangle \leq b_{i'}\}. \quad (103)$$

Используя (99), формулу (103) можно переписать в виде

$$\hat{H}_{i'} = \{\mathbf{x} \in \mathbb{R}^n | \langle \mathbf{a}_{i'}, \mathbf{x} \rangle \leq \langle \mathbf{a}_{i'}, \mathbf{u}' \rangle\}. \quad (104)$$

Из (88) следует $\mathbf{u}'' \in \hat{H}_{i'}$. Сопоставляя это с (104) получаем

$$\langle \mathbf{a}_{i'}, \mathbf{u}'' \rangle \leq \langle \mathbf{a}_{i'}, \mathbf{u}' \rangle, \quad (105)$$

что равносильно

$$\langle \mathbf{a}_{i'}, \mathbf{u}' - \mathbf{u}'' \rangle \geq 0. \quad (106)$$

Поскольку полупространство $\hat{H}_{i'}$ является рецессивным, в соответствии с утверждением 1 в [39] справедливо

$$\langle \mathbf{a}_{i'}, \mathbf{c} \rangle > 0. \quad (107)$$

В силу (102) и (91) имеем

$$\begin{aligned} \langle \mathbf{c}, \mathbf{w} - \mathbf{u}' \rangle &= \left\langle \mathbf{c}, \frac{r}{\|\mathbf{p} - \mathbf{u}'\|} (\mathbf{p} - \mathbf{u}') - \frac{\langle \mathbf{a}_{i'}, \frac{r}{\|\mathbf{p} - \mathbf{u}'\|} (\mathbf{p} - \mathbf{u}') \rangle}{\langle \mathbf{a}_{i'}, \mathbf{c} \rangle} \mathbf{c} \right\rangle = \frac{r}{\|\mathbf{p} - \mathbf{u}'\|} \left(\langle \mathbf{c}, \mathbf{p} - \mathbf{u}' \rangle - \frac{\langle \mathbf{a}_{i'}, \mathbf{p} - \mathbf{u}' \rangle}{\langle \mathbf{a}_{i'}, \mathbf{c} \rangle} \|\mathbf{c}\| \right) \\ &= \frac{r}{\|\mathbf{p} - \mathbf{u}'\|} \left(\left\langle \mathbf{c}, \mathbf{u}'' - \frac{\langle \mathbf{c}, \mathbf{u}'' - \mathbf{u}' \rangle}{\|\mathbf{c}\|^2} \mathbf{c} - \mathbf{u}' \right\rangle - \frac{\langle \mathbf{a}_{i'}, \mathbf{u}'' - \frac{\langle \mathbf{c}, \mathbf{u}'' - \mathbf{u}' \rangle}{\|\mathbf{c}\|^2} \mathbf{c} - \mathbf{u}' \rangle}{\langle \mathbf{a}_{i'}, \mathbf{c} \rangle} \|\mathbf{c}\| \right) \\ &= \frac{r}{\|\mathbf{p} - \mathbf{u}'\|} \left(- \frac{\langle \mathbf{a}_{i'}, \mathbf{u}'' - \frac{\langle \mathbf{c}, \mathbf{u}'' - \mathbf{u}' \rangle}{\|\mathbf{c}\|^2} \mathbf{c} - \mathbf{u}' \rangle}{\langle \mathbf{a}_{i'}, \mathbf{c} \rangle} \|\mathbf{c}\| \right) = \frac{r \|\mathbf{c}\|}{\|\mathbf{p} - \mathbf{u}'\| \langle \mathbf{a}_{i'}, \mathbf{c} \rangle} \left(- \left\langle \mathbf{a}_{i'}, \mathbf{u}'' - \frac{\langle \mathbf{c}, \mathbf{u}'' - \mathbf{u}' \rangle}{\|\mathbf{c}\|^2} \mathbf{c} - \mathbf{u}' \right\rangle \right) \\ &= \frac{r \|\mathbf{c}\|}{\|\mathbf{p} - \mathbf{u}'\| \langle \mathbf{a}_{i'}, \mathbf{c} \rangle} \left(\langle \mathbf{a}_{i'}, \mathbf{u}' - \mathbf{u}'' \rangle + \frac{\langle \mathbf{c}, \mathbf{u}'' - \mathbf{u}' \rangle \langle \mathbf{a}_{i'}, \mathbf{c} \rangle}{\|\mathbf{c}\|^2} \right) \end{aligned}$$

В соответствии с (93), (92), (107), (106) и (90) отсюда следует

$$\langle \mathbf{c}, \mathbf{w} - \mathbf{u}' \rangle > 0. \quad (108)$$

Это означает, что существует $\epsilon_f > 0$ такой, что

$$\langle \mathbf{c}, \mathbf{w} - \mathbf{u}' \rangle > \epsilon_f. \quad (109)$$

Вспомнив, что $\mathbf{u}' = \mathbf{u}^{(K)}$, перепишем последнее неравенство в виде

$$\langle \mathbf{c}, \mathbf{w} - \mathbf{u}^{(K)} \rangle > \epsilon_f. \quad (110)$$

Но тогда, согласно циклу **while**, представленному шагами 6–15 алгоритма 1, точка $\mathbf{u}^{(K)}$ не может быть конечной точкой последовательности $\{\mathbf{u}^{(k)}\}$, генерируемой алгоритмом 1. Получили противоречие. Таким образом точка $\mathbf{u}' = \mathbf{u}^{(K)}$ является решением задачи ЛП (1) при $\epsilon_f \rightarrow 0$. *Теорема доказана.* $\square$

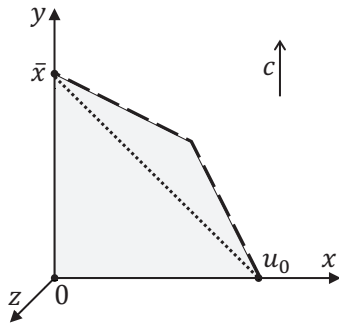


Рис. 4. Два пути от $\mathbf{u}_0$ к решению $\bar{\mathbf{x}}$: оптимальный целевой путь обозначен пунктиром; путь минимальной длины показан точками

Fig. 4. Two paths from $\mathbf{u}_0$ to solution $\bar{\mathbf{x}}$: optimal objective path is indicated by dashed line; path of minimum length is shown by dotted line

Алгоритм 1 строит на поверхности допустимого многогранника оптимальный целевой путь к решению задачи ЛП³. Это непосредственно следует из построения алгоритма и утверждения 3. Заметим, что оптимальный целевой путь не обязательно является путем наименьшей длины в смысле евклидовой метрики. Следующий простой пример в пространстве $\mathbb{R}^3$, проиллюстрированный на рис. 4, подтверждает этот факт. Система ограничений:

$$\begin{cases} x + 2y \leq 2, \\ 2x + y \leq 2, \\ x \geq 0, \\ y \geq 0, \\ z = 0; \end{cases} \quad (111)$$

целевая функция: $f(x, y, z) = y$. Также заметим, что шаг 13 алгоритма 1 должен найти точку гипердиска D , имеющую максимальное смещение. Нам неизвестен алгоритм, позволяющий получить численное решение этой задачи. Однако мы видим следующий путь решения,

³В случае формулировки задачи ЛП в виде (1) оптимальный целевой путь — это путь в направлении максимального увеличения значения целевой функции.

Алгоритм 2. Метод ЛП с использованием DNN

Algorithm 2. LP method using DNN

Require $\hat{H}_i = \{\mathbf{x} \in \mathbb{R}^n | \langle \mathbf{a}_i, \mathbf{x} \rangle \leq b_i\}$, $\hat{M} = \bigcap_{i \in \mathcal{I}} \hat{H}_i$, $H_c(\mathbf{z}) = \{\mathbf{x} \in \mathbb{R}^n | \langle \mathbf{c}, \mathbf{x} - \mathbf{z} \rangle = 0\}$

- 1: **input** $\mathbf{u}^{(0)}$; **assert** $\mathbf{u}^{(0)} \in \hat{M} \cap \Gamma(M)$
- 2: $k := 0$
- 3: $D := \mathfrak{G}(\mathbf{u}^{(0)})$
- 4: $\mathbf{d} := \text{DNN}(D)$
- 5: **while** $\mathbf{d} \neq \mathbf{0}$ **do**
- 6: $L = \{\mathbf{u}^{(k)} + \lambda \mathbf{d} \mid \lambda \in \mathbb{R}_{>0}\}$
- 7: $\mathbf{u}^{(k+1)} := \arg \max\{\|\mathbf{x} - \mathbf{u}^{(k)}\| \mid \mathbf{x} \in L \cap \Gamma(M)\}$
- 8: $k := k + 1$
- 9: $D := \mathfrak{G}(\mathbf{u}^{(k)})$
- 10: $\mathbf{d} := \text{DNN}(D)$
- 11: **end while**
- 12: **output** $\mathbf{u}^{(k)}$
- 13: **stop**

предполагающий использование искусственной нейронной сети. Используя технику, описанную в статье [37], мы заменяем гипердиск D на регулярное множество точек, называемых рецептивным полем. Каждой точке рецептивного поля мы сопоставляем ее смещение относительно границы многогранника. Таким образом мы получаем матрицу размерности $(n - 1)$, представляющей собой локальный образ задачи ЛП. Этот образ подается на вход предварительно обученной нейронной сети прямого распространения, которая определяет вектор $\mathbf{d}$, указывающий направление движения на поверхности допустимого многогранника в сторону максимального увеличения значения целевой функции. Обозначим через $\mathfrak{G}(\mathbf{u})$ функцию, строящую рецептивное поле с центром в точке $\mathbf{u}$ и вычисляющую локальный образ задачи ЛП в этой точке. Обозначим через DNN глубокую нейронную сеть прямого распространения, на вход которой подается локальный образ задачи ЛП, а на выходе получается вектор $\mathbf{d}$, задающий направление поверхностного движения. Тогда алгоритм 1 может быть преобразован в алгоритм 2, допускающий реализацию на практике. Множество размеченных прецедентов, необходимое для обучения DNN, может быть получено с помощью апекс-метода [39], строящего путь, близкий к оптимальному целевому пути.

4. Заключение. В статье описан новый итерационный метод решения задачи линейного программирования (ЛП), получивший название “метод поверхностного движения”. Указанный метод строит на поверхности многогранника, ограничивающего допустимую область задачи ЛП, путь от начальной точки до точки решения задачи ЛП. Вектор движения всегда выбирается в направлении максимального увеличения/уменьшения значения целевой функции. Получившийся путь называется оптимальным целевым путем.

Метод поверхностного движения предполагает использование глубокой нейронной сети прямого распространения для определения направления движения по граням допустимого многогранника. Для этого строится многомерный локальный образ задачи ЛП в точке текущего приближения, который подается на вход нейронной сети. Множество размеченных прецедентов, необходимое для обучения нейронной сети может быть получено с помощью апекс-метода.

Для построения теоретического фундамента метода поверхностного движения введено понятие целевой проекции — косоугольной проекции в направлении, параллельном вектору градиента целевой функции. Определена скалярная величина, называемая смещением. Модуль смещения равен расстоянию от точки до ее целевой проекции. Знак смещения определяется положением точки внутри или вне допустимого многогранника. Получена формула вычисления смещения точки относительно границы допустимого многогранника. Показано, что большему целевому смещению соответствует большее значение целевой функции. Приведено формализованное описание метода поверхностного движения в виде алгоритма. Доказана основная теорема сходимости метода поверхностного движения к решению задачи ЛП за конечное число итераций. Приведен вариант алгоритма поверхностного движения, использующий функцию по-

строения локального многомерного образа задачи ЛП и глубокую нейронную сеть.

В качестве направлений дальнейших исследований можно указать следующие.

1. Оценка временной сложности алгоритма 2.
2. Разработка и обучение сети DNN, способной вычислять вектор движения в направлении максимального увеличения значения целевой функции для многомерных задач ЛП.
3. Исследование зависимости точности работы сети DNN от плотности рецептивного поля.
4. Разработка программного комплекса для кластерной вычислительной системы, реализующего алгоритм 2 путем синтеза суперкомпьютерных и нейросетевых технологий.

5. Обозначения.

$\mathbb{R}^n$	вещественное евклидово пространство
$\ \cdot\ $	евклидова норма
$\langle \cdot, \cdot \rangle$	скалярное произведение двух векторов
$[\cdot, \cdot]$	конкатенация двух векторов
$f(\mathbf{x})$	линейная целевая функция
$\mathbf{c}$	градиент целевой функции $f(\mathbf{x})$
$\mathbf{e}_c$	единичный вектор, сонаправленный с вектором $\mathbf{c}$
$\tilde{\mathbf{x}}$	решение задачи ЛП
$\mathbf{a}_i$	i -тая строка матрицы A
H_i	гиперплоскость, определяемая формулой $\langle \mathbf{a}_i, \mathbf{x} \rangle = b_i$
$\hat{H}_i$	полупространство, определяемое формулой $\langle \mathbf{a}_i, \mathbf{x} \rangle \leq b_i$
$\mathcal{P}$	множество индексов строк матрицы A
M	допустимый многогранник, определяемый формулой $M = \bigcap_{i \in \mathcal{P}} \hat{H}_i$
$\Gamma(M)$	множество граничных точек допустимого многогранника M
$\mathcal{I}$	множество индексов, для которых полупространство $\hat{H}_i$ является рецессивным
$\hat{M}$	рецессивный многогранник, определяемый формулой $\hat{M} = \bigcap_{i \in \mathcal{I}} \hat{H}_i$
$\Gamma(\hat{M})$	множество граничных точек рецессивного многогранника $\hat{M}$
$\gamma_i(\mathbf{z})$	целевая проекция точки $\mathbf{z}$ на гиперплоскость H_i
$\beta_i(\mathbf{z})$	целевое смещение точки $\mathbf{z}$ относительно гиперплоскости H_i : $ \beta_i(\mathbf{z}) = \ \gamma_i(\mathbf{z}) - \mathbf{z}\ $
$\hat{\gamma}(\mathbf{z})$	целевая проекция точки $\mathbf{z}$ на границу $\Gamma(\hat{M})$ рецессивного многогранника $\hat{M}$
$\hat{\beta}(\mathbf{z})$	целевое смещение точки $\mathbf{z}$ относительно границы $\Gamma(\hat{M})$: $ \hat{\beta}(\mathbf{z}) = \ \hat{\gamma}(\mathbf{z}) - \mathbf{z}\ $
$V_r(\mathbf{x})$	гипершар радиуса r с центром в точке $\mathbf{x}$

Список литературы

1. *Hartung T.* Making Big Sense from Big Data // *Frontiers in Big Data*. 2018. Vol. 1. P. 5. doi [10.3389/fdata.2018.00005](https://doi.org/10.3389/fdata.2018.00005).
2. *Соколинская И.М., Соколинский Л.Б.* О решении задачи линейного программирования в эпоху больших данных // *Параллельные вычислительные технологии (ПаВТ'2017)*. Короткие статьи и описания плакатов. Челябинск: Издательский центр ЮУрГУ, 2017. С. 471–484. <http://omega.sp.susu.ru/pavt2017/short/014.pdf>.
3. *Branke J.* Optimization in Dynamic Environments // *Evolutionary Optimization in Dynamic Environments. Genetic Algorithms and Evolutionary Computation*, vol. 3. Boston, MA: Springer, 2002. P. 13–29. doi [10.1007/978-1-4615-0911-0_2](https://doi.org/10.1007/978-1-4615-0911-0_2).
4. *Ерёмин И.И., Мазуров В.Д.* Нестационарные процессы математического программирования. М.: Наука. Главная редакция физико-математической литературы, 1979. 288 с.
5. *Brogaard J., Hendershott T., Riordan R.* High-Frequency Trading and Price Discovery // *Review of Financial Studies*. 2014. Vol. 27, no. 8. P. 2267–2306. doi [10.1093/rfs/hhu032](https://doi.org/10.1093/rfs/hhu032).
6. *Deng S., Huang X., Wang J., et al.* A Decision Support System for Trading in Apple Futures Market Using Predictions Fusion // *IEEE Access*. 2021. Vol. 9. P. 1271–1285. doi [10.1109/ACCESS.2020.3047138](https://doi.org/10.1109/ACCESS.2020.3047138).
7. *Seregin G.* Lecture notes on regularity theory for the Navier-Stokes equations. Singapore: World Scientific Publishing Company, 2014. 268 p. doi [10.1142/9314](https://doi.org/10.1142/9314).

8. *Demin D.A.* Synthesis of optimal control of technological processes based on a multialternative parametric description of the final state // Eastern-European Journal of Enterprise Technologies. 2017. Vol. 3, 4(87). P. 51–63. doi [10.15587/1729-4061.2017.105294](https://doi.org/10.15587/1729-4061.2017.105294).
9. *Kazarinov L.S., Shnayder D.A., Kolesnikova O.V.* Heat load control in steam boilers // 2017 International Conference on Industrial Engineering, Applications and Manufacturing, ICIEAM 2017 - Proceedings. IEEE, 2017. doi [10.1109/ICIEAM.2017.8076177](https://doi.org/10.1109/ICIEAM.2017.8076177).
10. *Zagoskina E.V., Barbasova T.A., Shnaider D.A.* Intelligent Control System of Blast-furnace Melting Efficiency // SIBIRCON 2019 - International Multi-Conference on Engineering, Computer and Information Sciences, Proceedings. IEEE, 2019. P. 710–713. doi [10.1109/SIBIRCON48586.2019.8958221](https://doi.org/10.1109/SIBIRCON48586.2019.8958221).
11. *Fleming J., Yan X., Allison C., et al.* Real-time predictive eco-driving assistance considering road geometry and long-range radar measurements // IET Intelligent Transport Systems. 2021. Vol. 15, no. 4. P. 573–583. doi [10.1049/ITR2.12047](https://doi.org/10.1049/ITR2.12047).
12. *Scholl M., Minnerup K., Reiter C., et al.* Optimization of a thermal management system for battery electric vehicles // 14th International Conference on Ecological Vehicles and Renewable Energies, EVER 2019. IEEE, 2019. doi [10.1109/EVER.2019.8813657](https://doi.org/10.1109/EVER.2019.8813657).
13. *Meisel S.* Dynamic Vehicle Routing // Anticipatory Optimization for Dynamic Decision Making. Operations Research/Computer Science Interfaces Series, vol. 51. New York, NY: Springer, 2011. P. 77–96. doi [10.1007/978-1-4614-0505-4_6](https://doi.org/10.1007/978-1-4614-0505-4_6).
14. *Kiran D.* Production Planning and Control: A Comprehensive Approach. Elsevier Inc., 2019. 582 p. doi [10.1016/C2018-0-03856-6](https://doi.org/10.1016/C2018-0-03856-6).
15. *Mall R.* Real-Time Systems: Theory and Practice. Delhi, India: Pearson Education, 2007. 242 p.
16. *Dantzig G.B.* Linear programming and extensions. Princeton, N.J.: Princeton university press, 1998. 656 p.
17. *Hall J., McKinnon K.* Hyper-sparsity in the revised simplex method and how to exploit it // Computational Optimization and Applications. 2005. Vol. 32, no. 3. P. 259–283. doi [10.1007/s10589-005-4802-0](https://doi.org/10.1007/s10589-005-4802-0).
18. *Klee V., Minty G.* How good is the simplex algorithm? // Inequalities - III. Proceedings of the Third Symposium on Inequalities Held at the University of California, Los Angeles, Sept. 1-9, 1969 / ed. by O. Shisha. New York-London: Academic Press, 1972. P. 159–175.
19. *Bartels R., Stoer J., Zenger C.* A Realization of the Simplex Method Based on Triangular Decompositions // Handbook for Automatic Computation. Volume II: Linear Algebra. Berlin, Heidelberg: Springer, 1971. P. 152–190. doi [10.1007/978-3-642-86940-2_11](https://doi.org/10.1007/978-3-642-86940-2_11).
20. *Tolla P.* A Survey of Some Linear Programming Methods // Concepts of Combinatorial Optimization / ed. by V.T. Paschos. 2nd ed. Hoboken, NJ, USA: John Wiley, Sons, 2014. Chap. 7. P. 157–188. doi [10.1002/9781119005216.ch7](https://doi.org/10.1002/9781119005216.ch7).
21. *Hall J.* Towards a practical parallelisation of the simplex method // Computational Management Science. 2010. Vol. 7, no. 2. P. 139–170. doi [10.1007/s10287-008-0080-5](https://doi.org/10.1007/s10287-008-0080-5).
22. *Mamalis B., Pantziou G.* Advances in the Parallelization of the Simplex Method // Algorithms, Probability, Networks, and Games. Lecture Notes in Computer Science, vol. 9295 / ed. by C. Zaroliagis, G. Pantziou, S. Kontogiannis. Cham: Springer, 2015. P. 281–307. doi [10.1007/978-3-319-24024-4_17](https://doi.org/10.1007/978-3-319-24024-4_17).
23. *Зоркальцев В.И., Мокрый И.В.* Алгоритмы внутренних точек в линейной оптимизации // Сибирский журнал индустриальной математики. 2018. Т. 21, 1 (73). С. 11–20. doi [10.17377/sibjim.2018.21.102](https://doi.org/10.17377/sibjim.2018.21.102).
24. *Дикин И.И.* Итеративное решение задач линейного и квадратичного программирования // Доклады Академии наук СССР. 1967. Т. 174, № 4. С. 747–748. <https://www.mathnet.ru/rus/dan33112>.
25. *Gondzio J.* Interior point methods 25 years later // European Journal of Operational Research. 2012. Vol. 218, no. 3. P. 587–601. doi [10.1016/j.ejor.2011.09.017](https://doi.org/10.1016/j.ejor.2011.09.017).
26. *Roos C., Terlaky T., Vial J.-P.* Interior Point Methods for Linear Optimization. New York: Springer, 2005. 500 p. doi [10.1007/b100325](https://doi.org/10.1007/b100325).
27. *Sokolinskaya I.M.* Parallel Method of Pseudoprojection for Linear Inequalities // Parallel Computational Technologies. PCT 2018. Communications in Computer and Information Science, vol. 910 / ed. by L. Sokolinsky, M. Zymbler. Cham: Springer, 2018. P. 216–231. doi [10.1007/978-3-319-99673-8_16](https://doi.org/10.1007/978-3-319-99673-8_16).
28. *Gondzio J., Grothey A.* Direct Solution of Linear Systems of Size 109 Arising in Optimization with Interior Point Methods // Parallel Processing and Applied Mathematics. PPAM 2005. Lecture Notes in Computer Science, vol. 3911. 3911 LNCS / ed. by R. Wyrzykowski, J. Dongarra, N. Meyer, J. Wasniewski. Berlin, Heidelberg: Springer, 2006. P. 513–525. doi [10.1007/11752578_62](https://doi.org/10.1007/11752578_62).
29. *Prieto A., Prieto B., Ortigosa E.M., et al.* Neural networks: An overview of early research, current frameworks and new challenges // Neurocomputing. 2016. Vol. 214. P. 242–268. doi [10.1016/j.neucom.2016.06.014](https://doi.org/10.1016/j.neucom.2016.06.014).

30. *Tank D.W., Hopfield J.J.* Simple ‘neural’ optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit // IEEE transactions on circuits and systems. 1986. Vol. CAS–33, no. 5. P. 533–541. doi [10.1109/TCS.1986.1085953](#).
31. *Kennedy M.P., Chua L.O.* Unifying the Tank and Hopfield Linear Programming Circuit and the Canonical Nonlinear Programming Circuit of Chua and Lin // IEEE Transactions on Circuits and Systems. 1987. Vol. 34, no. 2. P. 210–214. doi [10.1109/TCS.1987.1086095](#).
32. *Rodriguez-Vazquez A., Dominguez-Castro R., Rueda A., et al.* Nonlinear Switched-Capacitor “Neural” Networks for Optimization Problems // IEEE Transactions on Circuits and Systems. 1990. Vol. 37, no. 3. P. 384–398. doi [10.1109/31.52732](#).
33. *Zak S.H., Upatising V.* Solving Linear Programming Problems with Neural Networks: A Comparative Study // IEEE Transactions on Neural Networks. 1995. Vol. 6, no. 1. P. 94–104. doi [10.1109/72.363446](#).
34. *Malek A., Yari A.* Primal-dual solution for the linear programming problems using neural networks // Applied Mathematics and Computation. 2005. Vol. 167, no. 1. P. 198–211. doi [10.1016/J.AMC.2004.06.081](#).
35. *Liu X., Zhou M.* A one-layer recurrent neural network for non-smooth convex optimization subject to linear inequality constraints // Chaos, Solitons and Fractals. 2016. Vol. 87. P. 39–46. doi [10.1016/j.chaos.2016.03.009](#).
36. *LeCun Y., Bengio Y., Hinton G.* Deep learning // Nature. 2015. Vol. 521, no. 7553. P. 436–444. doi [10.1038/nature14539](#).
37. *Ольховский Н.А., Соколинский Л.Б.* Визуальное представление многомерных задач линейного программирования // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2022. Т. 11, № 1. С. 31–56. doi [10.14529/cms220103](#).
38. *Raina R., Madhavan A., Ng A.Y.* Large-scale deep unsupervised learning using graphics processors // Proceedings of the 26th Annual International Conference on Machine Learning (ICML '09). New York, NY, USA: ACM Press, 2009. P. 873–880. doi [10.1145/1553374.1553486](#).
39. *Соколинский Л.Б., Соколинская И.М.* О новой версии апекс-метода для решения задач линейного программирования // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2023. Т. 12, № 2. С. 5–46. doi [10.14529/cms230201](#).

Поступила в редакцию
16 июня 2021 г.

Принята к публикации
13 октября 2021 г.

Информация об авторах

Николай Александрович Ольховский — аспирант; Южно-Уральский государственный университет (национальный исследовательский университет), пр. им. В.И. Ленина, д. 76, 454080, Челябинск, Российская Федерация.

Леонид Борисович Соколинский — д.ф.-м.н., зав. каф. системного программирования; Южно-Уральский государственный университет (национальный исследовательский университет), пр. им. В.И. Ленина, д. 76, 454080, Челябинск, Российская Федерация.

References

40. *Hartung T.* Making Big Sense From Big Data. *Frontiers in Big Data*. 2018. Vol. 1. P. 5. doi [10.3389/fdata.2018.00005](https://doi.org/10.3389/fdata.2018.00005).
41. *Sokolinskaya I.M., Sokolinsky L.B.* On the Solution of Linear Programming Problems in the Age of Big Data. *Parallel Computational Technologies. PCT 2017. Communications in Computer and Information Science*, vol. 753. / ed. by L. Sokolinsky, M. Zymbler. Cham: Springer, 2017. P. 86–100. doi [10.1007/978-3-319-67035-5_7](https://doi.org/10.1007/978-3-319-67035-5_7).
42. *Branke J.* Optimization in Dynamic Environments. *Evolutionary Optimization in Dynamic Environments. Genetic Algorithms and Evolutionary Computation*, vol. 3. Boston, MA: Springer, 2002. P. 13–29. doi [10.1007/978-1-4615-0911-0_2](https://doi.org/10.1007/978-1-4615-0911-0_2).
43. *Eremín I.I., Mazurov V.D.* Nonstationary processes of mathematical programming (Nestacionarnye processy matematicheskogo programmirovaniya). Moscow: Nauka. Glavnaya redakciya fiziko-matematicheskoy literatury, 1979. 288 p. (in Russian).
44. *Brogaard J., Hendershott T., Riordan R.* High-Frequency Trading and Price Discovery. *Review of Financial Studies*. 2014. Vol. 27, no. 8. P. 2267–2306. doi [10.1093/rfs/hhu032](https://doi.org/10.1093/rfs/hhu032).
45. *Deng S., Huang X., Wang J., et al.* A Decision Support System for Trading in Apple Futures Market Using Predictions Fusion. *IEEE Access*. 2021. Vol. 9. P. 1271–1285. doi [10.1109/ACCESS.2020.3047138](https://doi.org/10.1109/ACCESS.2020.3047138).
46. *Seregin G.* Lecture notes on regularity theory for the Navier-Stokes equations. Singapore: World Scientific Publishing Company, 2014. 268 p. doi [10.1142/9314](https://doi.org/10.1142/9314).
47. *Demin D.A.* Synthesis of optimal control of technological processes based on a multialternative parametric description of the final state. *Eastern-European Journal of Enterprise Technologies*. 2017. Vol. 3, 4(87). P. 51–63. doi [10.15587/1729-4061.2017.105294](https://doi.org/10.15587/1729-4061.2017.105294).
48. *Kazarinov L.S., Shnayder D.A., Kolesnikova O.V.* Heat load control in steam boilers. 2017 International Conference on Industrial Engineering, Applications and Manufacturing, ICIEAM 2017 - Proceedings. IEEE, 2017. doi [10.1109/ICIEAM.2017.8076177](https://doi.org/10.1109/ICIEAM.2017.8076177).
49. *Zagoskina E.V., Barbasova T.A., Shnaider D.A.* Intelligent Control System of Blast-furnace Melting Efficiency. *SIBIRCON 2019 - International Multi-Conference on Engineering, Computer and Information Sciences, Proceedings. IEEE*, 2019. P. 710–713. doi [10.1109/SIBIRCON48586.2019.8958221](https://doi.org/10.1109/SIBIRCON48586.2019.8958221).
50. *Fleming J., Yan X., Allison C., et al.* Real-time predictive eco-driving assistance considering road geometry and long-range radar measurements. *IET Intelligent Transport Systems*. 2021. Vol. 15, no. 4. P. 573–583. doi [10.1049/ITR2.12047](https://doi.org/10.1049/ITR2.12047).
51. *Scholl M., Minnerup K., Reiter C., et al.* Optimization of a thermal management system for battery electric vehicles. 14th International Conference on Ecological Vehicles and Renewable Energies, EVER 2019. IEEE, 2019. doi [10.1109/EVER.2019.8813657](https://doi.org/10.1109/EVER.2019.8813657).
52. *Meisel S.* Dynamic Vehicle Routing. *Anticipatory Optimization for Dynamic Decision Making. Operations Research/Computer Science Interfaces Series*, vol. 51. New York, NY: Springer, 2011. P. 77–96. doi [10.1007/978-1-4614-0505-4_6](https://doi.org/10.1007/978-1-4614-0505-4_6).
53. *Cheng A.M.K.* Real-Time Scheduling and Schedulability Analysis. *Real-Time Systems: Scheduling, Analysis, and Verification*. John Wiley, Sons, 2002. P. 41–85. doi [10.1002/0471224626.CH3](https://doi.org/10.1002/0471224626.CH3).
54. *Kopetz H.* Real-Time Scheduling. *Real-Time Systems. Real-Time Systems Series*. Boston, MA: Springer, 2011. P. 239–258. doi [10.1007/978-1-4419-8237-7_10](https://doi.org/10.1007/978-1-4419-8237-7_10).
55. *Mall R.* Real-Time Systems: Theory and Practice. Delhi, India: Pearson Education, 2007. 242 p.
56. *Dantzig G.B.* Linear programming and extensions. Princeton, N.J.: Princeton university press, 1998. 656 p.
57. *Hall J., McKinnon K.* Hyper-sparsity in the revised simplex method and how to exploit it. *Computational Optimization and Applications*. 2005. Vol. 32, no. 3. P. 259–283. doi [10.1007/s10589-005-4802-0](https://doi.org/10.1007/s10589-005-4802-0).
58. *Klee V., Minty G.* How good is the simplex algorithm? Inequalities - III. Proceedings of the Third Symposium on Inequalities Held at the University of California, Los Angeles, Sept. 1-9, 1969 / ed. by O. Shisha. New York-London: Academic Press, 1972. P. 159–175.
59. *Bartels R., Stoer J., Zenger C.* A Realization of the Simplex Method Based on Triangular Decompositions. *Handbook for Automatic Computation. Volume II: Linear Algebra*. Berlin, Heidelberg: Springer, 1971. P. 152–190. doi [10.1007/978-3-642-86940-2_11](https://doi.org/10.1007/978-3-642-86940-2_11).
60. *Tolla P.* A Survey of Some Linear Programming Methods. *Concepts of Combinatorial Optimization* / ed. by V.T. Paschos. 2nd ed. Hoboken, NJ, USA: John Wiley, Sons, 2014. Chap. 7. P. 157–188. doi [10.1002/9781119005216.ch7](https://doi.org/10.1002/9781119005216.ch7).
61. *Hall J.* Towards a practical parallelisation of the simplex method. *Computational Management Science*. 2010. Vol. 7, no. 2. P. 139–170. doi [10.1007/s10287-008-0080-5](https://doi.org/10.1007/s10287-008-0080-5).

62. *Mamalis B., Pantziou G.* Advances in the Parallelization of the Simplex Method. Algorithms, Probability, Networks, and Games. Lecture Notes in Computer Science, vol. 9295 / ed. by C. Zaroliagis, G. Pantziou, S. Kontogiannis. Cham: Springer, 2015. P. 281–307. doi [10.1007/978-3-319-24024-4_17](https://doi.org/10.1007/978-3-319-24024-4_17).
63. *Zorkaltsev V., Mokryi I.* Interior point algorithms in linear optimization. Journal of applied and industrial mathematics. 2018. Vol. 12, no. 1. P. 191–199. doi [10.1134/S1990478918010179](https://doi.org/10.1134/S1990478918010179).
64. *Dikin I.* Iterative solution of problems of linear and quadratic programming. Soviet Mathematics. Doklady. 1967. Vol. 8. P. 674–675.
65. *Gondzio J.* Interior point methods 25 years later. European Journal of Operational Research. 2012. Vol. 218, no. 3. P. 587–601. doi [10.1016/j.ejor.2011.09.017](https://doi.org/10.1016/j.ejor.2011.09.017).
66. *Roos C., Terlaky T., Vial J.-P.* Interior Point Methods for Linear Optimization. New York: Springer, 2005. 500 p. doi [10.1007/b100325](https://doi.org/10.1007/b100325).
67. *Sokolinskaya I.* Parallel Method of Pseudoprojection for Linear Inequalities. Parallel Computational Technologies. PCT 2018. Communications in Computer and Information Science, vol. 910 / ed. by L. Sokolinsky, M. Zymbler. Cham: Springer, 2018. P. 216–231. doi [10.1007/978-3-319-99673-8_16](https://doi.org/10.1007/978-3-319-99673-8_16).
68. *Gondzio J., Grothey A.* Direct Solution of Linear Systems of Size 109 Arising in Optimization with Interior Point Methods. Parallel Processing and Applied Mathematics. PPAM 2005. Lecture Notes in Computer Science, vol. 3911. 3911 LNCS / ed. by R. Wyrzykowski, J. Dongarra, N. Meyer, J. Wasniewski. Berlin, Heidelberg: Springer, 2006. P. 513–525. doi [10.1007/11752578_62](https://doi.org/10.1007/11752578_62).
69. *Prieto A., Prieto B., Ortigosa E.M., et al.* Neural networks: An overview of early research, current frameworks and new challenges. Neurocomputing. 2016. Vol. 214. P. 242–268. doi [10.1016/j.neucom.2016.06.014](https://doi.org/10.1016/j.neucom.2016.06.014).
70. *Tank D.W., Hopfield J.J.* Simple ‘neural’ optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit. IEEE transactions on circuits and systems. 1986. Vol. CAS–33, no. 5. P. 533–541. doi [10.1109/TCS.1986.1085953](https://doi.org/10.1109/TCS.1986.1085953).
71. *Kennedy M.P., Chua L.O.* Unifying the Tank and Hopfield Linear Programming Circuit and the Canonical Nonlinear Programming Circuit of Chua and Lin. IEEE Transactions on Circuits and Systems. 1987. Vol. 34, no. 2. P. 210–214. doi [10.1109/TCS.1987.1086095](https://doi.org/10.1109/TCS.1987.1086095).
72. *Rodriguez-Vazquez A., Dominguez-Castro R., Rueda A., et al.* Nonlinear Switched-Capacitor “Neural” Networks for Optimization Problems. IEEE Transactions on Circuits and Systems. 1990. Vol. 37, no. 3. P. 384–398. doi [10.1109/31.52732](https://doi.org/10.1109/31.52732).
73. *Zak S.H., Upatizing V.* Solving Linear Programming Problems with Neural Networks: A Comparative Study. IEEE Transactions on Neural Networks. 1995. Vol. 6, no. 1. P. 94–104. doi [10.1109/72.363446](https://doi.org/10.1109/72.363446).
74. *Malek A., Yari A.* Primal-dual solution for the linear programming problems using neural networks. Applied Mathematics and Computation. 2005. Vol. 167, no. 1. P. 198–211. doi [10.1016/J.AMC.2004.06.081](https://doi.org/10.1016/J.AMC.2004.06.081).
75. *Liu X., Zhou M.* A one-layer recurrent neural network for non-smooth convex optimization subject to linear inequality constraints. Chaos, Solitons and Fractals. 2016. Vol. 87. P. 39–46. doi [10.1016/j.chaos.2016.03.009](https://doi.org/10.1016/j.chaos.2016.03.009).
76. *LeCun Y., Bengio Y., Hinton G.* Deep learning. Nature. 2015. Vol. 521, no. 7553. P. 436–444. doi [10.1038/nature14539](https://doi.org/10.1038/nature14539).
77. *Olkhovskiy N., Sokolinsky L.* Visualizing Multidimensional Linear Programming Problems. Parallel Computational Technologies. PCT 2022. Communications in Computer and Information Science, vol. 1618 / ed. by L. Sokolinsky, M. Zymbler. Cham: Springer, 2022. P. 172–196. doi [10.1007/978-3-031-11623-0_13](https://doi.org/10.1007/978-3-031-11623-0_13).
78. *Raina R., Madhavan A., Ng A.Y.* Large-scale deep unsupervised learning using graphics processors. Proceedings of the 26th Annual International Conference on Machine Learning (ICML '09). New York, NY, USA: ACM Press, 2009. P. 873–880. doi [10.1145/1553374.1553486](https://doi.org/10.1145/1553374.1553486).
79. *Sokolinsky L.B., Sokolinskaya I.M.* Apex Method: A New Scalable Iterative Method for Linear Programming. Mathematics. 2023. Vol. 11, no. 7. P. 1654. doi [10.3390/MATH11071654](https://doi.org/10.3390/MATH11071654).

Received
June 16, 2021

Accepted for publication
October 13, 2021

Information about the authors

Nikolay A. Olkhovskiy — PhD student; South Ural State University (National Research University), pr. Lenina 76, 454080, Chelyabinsk, Russia.

Leoind B. Sokolinsky — Doctor of sciences (phys. and math.), Head of Comp. Sc. Dept.; South Ural State University (National Research University), pr. Lenina 76, 454080, Chelyabinsk, Russia.

Главному редактору научного журнала
“Вычислительные методы и
программирование”
чл.-корр. РАН, проф. Вл. В. Воеводину

СОПРОВОДИТЕЛЬНОЕ ПИСЬМО

Просим опубликовать в журнале “Вычислительные методы и программирование” статью “О новом методе линейного программирования с использованием нейронных сетей”.
Сообщаем следующую информацию о каждом авторе статьи.

1. Фамилия, имя, отчество полностью: Ольховский Николай Александрович (Olkhovsky Nikolay A.).

2. Основное место работы: Южно-Уральский государственный университет (национальный исследовательский университет) (South Ural State University (national research university)) <http://www.susu.ru/>.

3. Должность по основному месту работы: аспирант (PhD student).

4. Ученая степень и ученое звание: нет (no).

5. Телефон мобильный: +7 9043070939.

6. E-mail: olkhovskiina@susu.ru.

7. ORCID: 0009-0008-9078-4799.

8. Номер и название рубрики ГРНТИ: 27.47.19 исследование операций; код OECD: 1.02.

9. Ссылка на профиль автора в elibrary.ru: https://elibrary.ru/author_items.asp?authorid=1200111.

1. Фамилия, имя, отчество полностью: Соколинский Леонид Борисович (Sokolinsky Leonid B.).

2. Основное место работы: Южно-Уральский государственный университет (национальный исследовательский университет) (South Ural State University (national research university)) <http://www.susu.ru/>.

3. Должность по основному месту работы: заведующий кафедрой системного программирования (Head of Computer Science Department).

4. Ученая степень и ученое звание: доктор физико-математических наук, профессор (Doctor of Sciences, Professor).

5. Телефон мобильный: +7 9128912525.

6. E-mail: leonid.sokolinsky@susu.ru.

7. ORCID: 0000-0001-9997-3918.

8. Номер и название рубрики ГРНТИ: 27.47.19 исследование операций; код OECD: 1.02.

9. Ссылка на профиль автора в elibrary.ru: https://elibrary.ru/author_items.asp

?authorid=99314.

Текущую переписку по вопросам публикации статьи следует вести с Соколинским Л.Б.

Авторы согласны с правилами подготовки статьи к публикации, включая рецензирование, научное и литературное редактирование и доведение статьи до редакторских стандартов, принятых в рамках журнала.

Авторы ознакомлены с Публикационной этикой журнала и согласны с ее положениями.

Авторы также согласны с передачей журналу своего права на издание и распространение статьи в электронной и бумажной версиях, в том числе на размещение библиографической информации о статье в Российском индексе научного цитирования (РИНЦ) и в других базах научного цитирования и на размещение полных текстов статей в Научной электронной библиотеке (elibrary.ru) и портале <http://www.mathnet.ru> для свободного доступа всем пользователям Интернет независимо от их категории и местоположения.

23.05.2023

_____ Н.А. Ольховский

_____ Л.Б. Соколинский