

**СИНТЕЗ СЕКУЩИХ И ОТДЕЛЯЮЩИХ ПЛОСКОСТЕЙ
В ОДНОМ МЕТОДЕ НЕГЛАДКОЙ ОПТИМИЗАЦИИ¹**

Аннотация. Предложен алгоритм решения задач недифференцируемой оптимизации семейства методов отделяющих плоскостей с дополнительными отсечениями, порождаемыми решением вспомогательной задачи метода секущих плоскостей. Доказана сходимость данного алгоритма, приведены результаты вычислительных экспериментов при решении транспортных задач. Задачи транспортного типа с ограничениями на потоки сводятся к задачам проекции достаточно удаленной точки на допустимое множество.

Ключевые слова: выпуклая оптимизация, методы отделяющих плоскостей.

ВВЕДЕНИЕ

Пусть $x = (x^1, \dots, x^n)$ — вектор n -мерного евклидова пространства \mathcal{R}^n с обычным скалярным произведением xu . В \mathcal{R}^n рассматривается задача безусловной выпуклой недифференцируемой оптимизации (НДО): $\min_{x \in \mathcal{R}^n} f(x)$, где $f(x)$ — выпуклая недифференцируемая функция. Будем считать, что данная задача является разрешимой.

Проблемы такого рода возникают в различных областях науки и техники, например при решении задач механики сплошных сред с учетом трения [1], теории управления [2], экономики [3–5] и др. Кроме того, прогресс в области разработки и реализации методов НДО даст возможность построить более эффективные способы решения оптимизационных задач большой размерности.

Данная статья посвящена дальнейшему исследованию эффективного метода [21] решения задач многомерной выпуклой НДО без ограничений, который не требует дополнительной информации о внутренней структуре оптимизируемой функции и является представителем так называемой black-box оптимизации. Предполагается, что вся доступная информация о целевой функции $f(x)$ задачи предоставляется субградиентным оракулом — в произвольной точке $\bar{x} \in \mathcal{R}^n$ можно определить только значение функции $f(\bar{x})$ и субградиент $g \in \partial f(\bar{x})$, произвольно выбранный из субдифференциала $\partial f(\bar{x})$ функции $f(x)$.

Схемы оракульного типа для минимизации гладких и негладких функций имеют принципиальные различия. Оракулы дифференцируемых функций позволяют построить сходящиеся релаксационные минимизирующие последовательности (см., например, [6]). Для негладких функций такая возможность неприменима в принципе, поскольку произвольно выбранный субградиент не определяет релаксационного направления [7]. По сути методы выпуклой НДО используют только свойства отделимости, что значительно уменьшает их скорость сходимости.

В [8] были установлены нижние границы оценки сложности методов, пользующихся только оракулом, для различных классов задач оптимизации. Оказалось, что для рассматриваемого класса задач выпуклой НДО не существует метода, который сходится быстрее, чем со скоростью порядка $O(q^{k/n})$, где $q < 1$ — абсолютная константа, k — число итераций метода. Ни в одном из методов оцен-

¹Работа проводится в рамках ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2014–2020 годы», соглашение 14.604.21.0052 от 30.06.2014 г. с МОН. Уникальный идентификатор проекта RFMEFI60414X0052.

ка скорости сходимости, равномерная по размерности пространства n , не может быть лучше, чем оценка скорости порядка $O(k^{-1/2})$. Доказано, что такими скоростями сходимости обладают метод центров тяжести [10, 11] и субградиентный метод [32]. Таким образом, два конкретно первых метода, разработанные для решения задач негладкой минимизации, оказались неулучшаемыми по своим скоростным характеристикам. Метод центров тяжести на практике не является применимым, поскольку операция нахождения центра тяжести выпуклого множества в многомерном пространстве — весьма сложная задача. Субградиентный метод, впервые предложенный Н.З. Шором [32], имеет простейшую вычислительную схему

$$x_{k+1} = x_k - \lambda_k g_k, \quad g_k \in \partial f(x_k), \quad k = 0, 1, \dots,$$

и сходится при весьма приемлемых условиях. Его практическая вычислительная эффективность зависит от способов управления шаговыми множителями λ . Наиболее удачным правилом выбора шагового множителя λ признано правило Б.Т. Поляка [12], однако при условии заранее известного значения оптимума. Среди других способов регулировки шага можно отметить также способ из [13]. Субградиентный метод с такой регулировкой шага на известных тестах продемонстрировал практически линейную скорость сходимости. Однако следует отметить, что данное утверждение о скорости сходимости методов НДО является верным равномерно по размерности пространства переменных. Для задач умеренной размерности можно разработать более эффективные схемы [9].

Следующим этапом развития методов НДО стало появление так называемых bundle-методов (от англ. bundle — пучок) [14–16], представителем которых является метод уровней [17], разработанный в 1995 г. Из последних публикаций в области bundle-методов можно отметить идею расщепления на подпространства гладкости–негладкости, названную VU-алгоритмом [18]; в области методов НДО, не относящихся к оракульному типу, отметим специальную технику сглаживания с последующим применением градиентных схем гладкой минимизации [19] для негладких функций с определенной структурой [20].

Предлагаемый в данной статье проекционный алгоритм SPACLIP решения задач минимизации негладких функций является результатом дальнейшего развития и усовершенствования методов отделяющих плоскостей [21–23], имеющих ряд важных теоретических и вычислительных особенностей.

В вычислительных экспериментах алгоритм SPACLIP применен для решения транспортной задачи, которая не только является одной из наиболее распространенных в экономических приложениях, но и имеет определенную символическую ценность, так как развитие НДО началось именно с задач этого типа [37].

В матричной постановке данной задачи, рассматриваемой в настоящей статье, на объемы поставок наложены ограничения сверху и снизу, что затрудняет применение метода потенциалов и симплекс-метода для решения таких задач, особенно задач большой размерности. Определенный интерес представляет сведение транспортной задачи к задаче нахождения проекции на сдвиг допустимого множества.

ПРОЕКЦИОННЫЙ АЛГОРИТМ SPACLIP

Методы отделяющих плоскостей [21] основываются на идее замены исходной задачи минимизации на задачу вычисления сопряженной функции Фенхеля–Моро в нуле

$$f(x^*) = \min_{x \in \mathbb{R}^n} f(x) = -\sup_x \{0 \cdot x - f(x)\} = -f^*(0), \quad x^* \in \partial f^*(0), \quad (1)$$

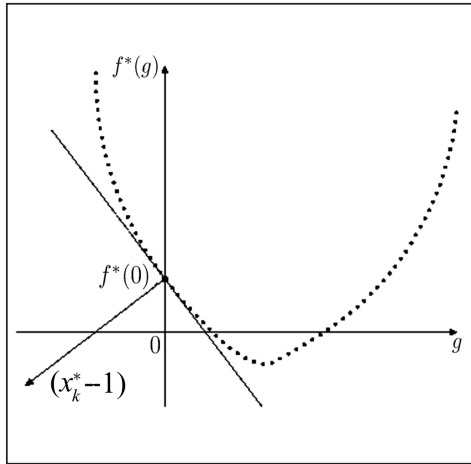


Рис. 1. Графическая интерпретация задачи (1)

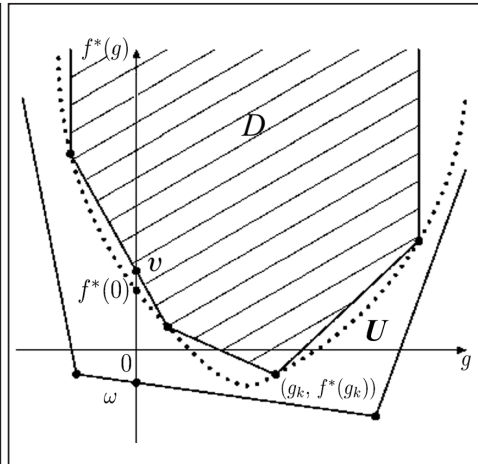


Рис. 2. Иллюстрация стандартного метода отделяющих плоскостей

где $f^*(g) = \sup \{gx - f(x)\}$ — функция, сопряженная (по Фенхелю–Моро) к $f(x)$. Задачу (1) можно интерпретировать как задачу нахождения точки пересечения графика сопряженной функции с вертикальной прямой $0 \times \mathfrak{R}_+$ (рис. 1). Будем считать, что $f(0) = 0$ и начало координат пространства прямых переменных решением задачи минимизации не является, т.е. $f(x^*) < 0$. На рис. 1 точечной линией показан график сопряженной функции $f^*(g)$. Опорный вектор касательной гиперплоскости в точке $(0, f^*(0))$ к надграфику сопряженной функции с точностью до нормировки определяет решение задачи (1).

В методе отделяющих плоскостей (МОП) надграфик сопряженной функции f^* аппроксимируется внутренне и внешне выпуклыми полиэдральными множествами D и U (рис. 2). Уточняя на каждой итерации эти аппроксимации в окрестности вертикальной прямой $0 \times \mathfrak{R}_+$, получаем сходящиеся нижние w и верхние v оценки для $f^*(0)$.

Множества D и U модифицируются добавлением либо отсечений (для U), либо новых точек $(\varepsilon, g(\varepsilon))$, лежащих на графике f^* .

Базовый вариант МОП не гарантирует монотонности, особенно при подходе к экстремуму. Для улучшения свойства монотонности метода введем дополнительное отсечение по верхней оценке v значения $f^*(0)$, полученной из внутренней аппроксимации надграфика (см. рис. 2):

$$v = \min_{(0, \varepsilon) \in D} \varepsilon \geq \min_{(0, \varepsilon) \in \text{epi } f^*} \varepsilon = f^*(0) \geq \min_{(0, \varepsilon) \in U} \varepsilon.$$

Такое отсечение априори более точно локализует потенциальные точки надграфика f^* , которые будут в дальнейшем добавляться при уточнении его аппроксимации.

В вычислительном плане значение v может быть получено как решение задачи линейного программирования и представляет собой оценку, определяемую по методу секущих плоскостей Келли [36]:

$$v = \min_{\substack{(0, \tau) \in \text{co}((g_k, f^*(g_k)), \\ k=1, 2, \dots, m) + 0 \times \mathfrak{R}_+}} \tau = \min_{\substack{\tau = \sum_{k=1}^m \lambda_k f^*(g_k), \\ 0 = \sum_{k=1}^m \lambda_k g_k, \\ \lambda \in \Delta_m}} \tau = \min_{\substack{\tau = \sum_{k=1}^m \lambda_k f^*(g_k), \\ 0 = \sum_{k=1}^m \lambda_k g_k, \\ \lambda \in \Delta_m}} \sum_{k=1}^m \lambda_k f^*(g_k) =$$

$$\begin{aligned}
&= \max_x \min_{\lambda \in \Delta_m} \sum_{k=1}^m \lambda_k (f^*(g_k) - xg_k) = \max_x \min_k \{f^*(g_k) - xg_k\} = \\
&= \max_x \min_k \{x_k g_k - f(x_k) - xg_k\} = - \min_x \max_k \{f(x_k) + (x - x_k)g_k\}, \quad (2)
\end{aligned}$$

где $\lambda = (\lambda_1, \dots, \lambda_m)$, Δ_m — стандартный симплекс, $\Delta_m = \{\lambda_k \geq 0, k = 1, \dots, m; \sum_{k=1}^m \lambda_k = 1\}$.

Добавление следующей точки к аппроксимации $\text{epi } f^*$ при таком отсечении заключается в решении задачи построения опорной гиперплоскости к усеченному надграфику f^* :

$$\sup_{\substack{(g, \varepsilon) \in \text{epi } f^* \\ \varepsilon \leq v}} \{gx - \varepsilon\}. \quad (3)$$

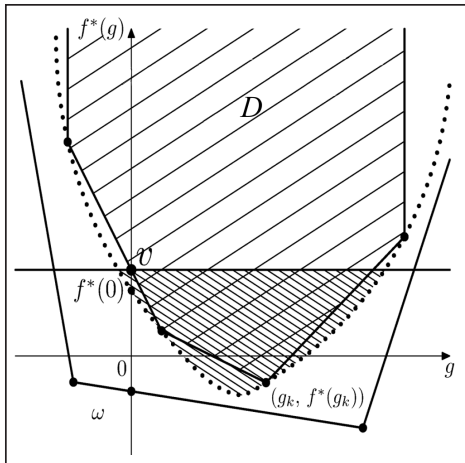


Рис. 3. Иллюстрация алгоритма SPACLIP-MOP с дополнительными отсечениями

Данная задача отличается от аналогичной задачи уточнения аппроксимации надграфика сопряженной функции в стандартном МОП наличием дополнительного ограничения $\varepsilon \leq v$. В этом плане алгоритм подобен методу уровней [17], однако отличается тем, что построение аппроксимаций происходит в расширенном пространстве \mathcal{R}^{n+1} . Рис. 3 иллюстрирует идею такого дополнительного отсечения.

Задачу (3), как и в стандартном МОП, легко перевести в пространство прямых переменных $x \in \mathcal{R}^n$, однако при этом появляется вспомогательная задача одномерной минимизации с достаточно неожиданной целевой функцией. Действительно, вводя для дополнительного ограничения двойственную переменную λ , получаем

$$\begin{aligned}
&\sup_{(g, \varepsilon) \in \text{epi } f^*; \varepsilon \leq v} \{gx - \varepsilon\} = \sup_g \inf_{\lambda \geq 0} \{gx - f^*(g) + \lambda(v - f^*(g))\} = \\
&= \inf_{\lambda \geq 0} \left\{ \lambda v + \sup_g \{gx - (\lambda + 1)f^*(g)\} \right\} = \inf_{\lambda \geq 0} \left\{ \lambda v + (1 + \lambda) f\left(\frac{x}{1 + \lambda}\right) \right\} = \\
&= -v + \inf_{\lambda \geq 1} \{\lambda(v + f(\lambda^{-1}x))\} = -v + \inf_{\lambda \geq 1} \varphi(\lambda, x). \quad (4)
\end{aligned}$$

Легко показать, что функция $\varphi(\lambda, x) = \lambda(f(\lambda^{-1}x) + v)$ выпукла по совокупности переменных (λ, x) . Оценку сверху для $\varphi(\alpha\lambda_1 + (1 - \alpha)\lambda_2, \alpha x + (1 - \alpha)y)$, где $0 \leq \alpha \leq 1$, можно получить, используя неравенство Йенсена для f :

$$\begin{aligned}
\varphi(\alpha\lambda_1 + (1 - \alpha)\lambda_2, \alpha x + (1 - \alpha)y) &= (\alpha\lambda_1 + (1 - \alpha)\lambda_2) \left(f\left(\frac{\alpha x + (1 - \alpha)y}{\alpha\lambda_1 + (1 - \alpha)\lambda_2}\right) + v \right) = \\
&= (\alpha\lambda_1 + (1 - \alpha)\lambda_2) \left(f\left(\frac{\alpha\lambda_1}{\alpha\lambda_1 + (1 - \alpha)\lambda_2} \cdot \frac{x}{\lambda_1} + \frac{(1 - \alpha)\lambda_2}{\alpha\lambda_1 + (1 - \alpha)\lambda_2} \cdot \frac{y}{\lambda_2}\right) + v \right) \leq \\
&\leq \alpha\lambda_1 \left(f\left(\frac{x}{\lambda_1}\right) + v \right) + (1 - \alpha)\lambda_2 \left(f\left(\frac{y}{\lambda_2}\right) + v \right) = \alpha\varphi(\lambda_1, x) + (1 - \alpha)\varphi(\lambda_2, y).
\end{aligned}$$

Таким образом, функция $\varphi(\lambda, x)$ выпукла по совокупности переменных (λ, x) , а следовательно, и по переменной λ . ■

Решать задачу одномерной НДО (4) предлагается с помощью быстрого алгоритма одномерного поиска [27, 28], поскольку в благоприятных случаях он имеет сверхлинейную или даже квадратичную скорость сходимости. Для алгоритма SPACCLIP была создана отдельная реализация быстрого алгоритма одномерного поиска.

В результате алгоритм метода отделяющих плоскостей с отсечениями (SPACCLIP) имеет следующий вид.

Шаг 0. Инициализация. Установить счетчик итераций $k=0$, определить начальную точку минимизирующей последовательности x_0 .

Шаг 1. Найти $\inf_{0 \in U_k(\omega)} \omega = \omega_k$, где U_k — k -я внешняя аппроксимация надграфика сопряженной функции f^* . Данную задачу можно решить рекуррентно:

$$\begin{aligned} \omega_k &= \inf_{0 \in U_k(\omega)} \omega = \inf_{0 \in U_{k-1}(\omega) \cap (g, \omega) | g x_{k-1} - \omega \leq f(x_{k-1})} \omega = \\ &= \max \left\{ \inf_{0 \in U_{k-1}(\omega)} \omega, \inf_{\omega \geq -f(x_{k-1})} \omega \right\} = \max \{ \omega_{k-1}, -f(x_{k-1}) \}, k \geq 1. \end{aligned} \quad (5)$$

При этом можно считать, что $\omega_0 = -\infty$. Фактически $-\omega_k$ является рекордом функции f .

Шаг 2. Найти вектор $\bar{z}^k = (z^k, \xi_k)$ — проекцию точки $(0, \omega_k)$ на полиэдр D_k внутренней аппроксимации надграфика сопряженной функции

$$\bar{z}^k = P_{D_k}((0, \omega_k)),$$

где $P_X(a)$ — решение задачи проекции точки a на множество X .

Для решения данной задачи используется метод подходящих аффинных подпространств [26]. Данный конечный метод решает задачу нахождения вектора минимальной длины в полиэдре конечномерного евклидова пространства и обладает «лучше, чем линейной» глобальной скоростью сходимости.

Шаг 3. Вычислить очередное приближение к решению задачи $\min_{x \in \mathfrak{R}^n} f(x)$:

$$x_k = -z^k / \xi_k.$$

При такой нормировке последняя координата в $\bar{x}_k = -\bar{z}^k / \xi_k = (x_k, -1)$ будет равна -1 , что и требуется (см. рис. 1).

Шаг 4. Определить v — уровень отсечения верхней части надграфика $\text{epi } f^*$, для чего необходимо решить задачу линейного программирования (2). Заметим, что такое отсечение не мешает решению задачи проекции на D_k шага 2 и, следовательно, построению приближенного решения x_k .

Если задача (2) не имеет решения, перейти к шагу 7.

Шаг 5. Решить задачу одномерной НДО (4). Пусть λ_k — найденное на k -й итерации решение данной задачи.

Шаг 6. Модифицировать приближение x_k по формуле $x_k = \lambda_k^{-1} x_k$.

В алгоритме SPACCLIP, в отличие от МОП, при добавлении очередной точки в аппроксимацию $\text{epi } f^*$ субградиент оптимизируемой функции вычисляется не в тестируемой точке x , а в масштабируемой относительно x точке $\lambda_k^{-1} x$.

Шаг 7. Добавить пару $(g_k \in \partial f(x_k), f^*(g_k))$ к полиэдру D_k .

Шаг 8. Если выполняется какое-либо из условий прекращения работы алгоритма, то завершить работу. Иначе увеличить на единицу счетчик итераций k и перейти к шагу 1.

В следующих разделах приведены доказательства сходимости данного алгоритма и результаты вычислительных экспериментов.

СХОДИМОСТЬ МЕТОДА

Сходимость проекционного алгоритма отделяющих плоскостей с отсечениями утверждается следующей теоремой.

Теорема 1. Пусть $f(x)$ — конечная выпуклая функция, $f(0) = 0$, $\omega_* = -\min f(x) < \Omega < \infty$. Тогда $\lim_{k \rightarrow \infty} \omega_k = \omega_*$.

Доказательство. По индукции можно доказать, что $\omega_k \leq f^*(0)$ для любого k .

Действительно, это неравенство по определению выполняется для $k=0$. Согласно (5) $\omega_k = \max\{\omega_{k-1}, -f(x_{k-1})\}$. Делая индуктивный переход и учитывая, что $-f(x_{k-1}) = 0 \cdot x_{k-1} - f(x_{k-1}) \leq \sup_x \{0 \cdot x - f(x)\} = f^*(0)$, получаем

$\omega_k \leq \max\{f^*(0), f^*(0)\} = f^*(0)$, что и требовалось доказать.

Поскольку $\omega_k \leq f^*(0)$, а $\bar{z}_k + (0, \omega_k) \in \text{co}\{D_k\} = D_k$, то для доказательства утверждения теоремы достаточно показать, что $\|\bar{z}_k\| \rightarrow 0$ при $k \rightarrow \infty$, так как это означает, что $-f(x_k) \rightarrow f^*(0)$.

Для доказательства монотонного убывания нормы векторов $\|\bar{z}_k\|$ рассмотрим несколько случаев.

1. Задача (2) на шаге 4 не имеет решения для всех k . Тогда можно считать, что $v \equiv \infty$ и МОП с дополнительными отсечениями преобразуется в стандартный МОП, сходимость которого доказана в [25].

2. Задача (2) на шаге 4 имеет решение. Обозначим \bar{x}_k решение задачи (3), $\bar{x}_k = \lambda_k^{-1} x_k$, $\lambda_k \geq 1$. Тогда вектор \bar{z}_k представим в виде $\bar{z}_k = -r_k(\lambda_k^{-1} x_k, -1)$.

В зависимости от того, меняется ли на k -й итерации текущий рекорд целевой функции ω_k , возможны два варианта.

Вариант 1. $\omega_k = \omega_{k-1}$. Тогда проекция выполняется из одной и той же точки: $\|\bar{z}_k\|^2 = \min_{z+(0, \omega_k) \in \text{co}D_k} \|z\|^2 = \min_{z+(0, \omega_{k-1}) \in \text{co}D'_{k-1}} \|z\|^2$. Здесь D'_k — политоп, полученный на k -й итерации после отсечения (3). При этом справедливо следующее неравенство:

$$\|\bar{z}_k\|^2 \leq \min_{\lambda \in [0, 1]} \|\bar{z}_{k-1} + \lambda(g_k, f^*(g_k)) - \bar{z}_{k-1}\|^2.$$

Решение задачи $\min_{\lambda \in [0, 1]} \|\bar{z}_{k-1} + \lambda(g_k, f^*(g_k)) - \bar{z}_{k-1}\|^2$ представляет проекцию минимума одномерной квадратичной функции на интервал $[0, 1]$:

$$\lambda^* = \min \{(\bar{z}_{k-1} - (g_k, f^*(g_k))) \bar{z}_{k-1} / \|(g_k, f^*(g_k)) - \bar{z}_{k-1}\|^2, 1\}.$$

Тогда для любого

$$\lambda \leq \frac{(\bar{z}_{k-1} - (g_k, f^*(g_k))) \bar{z}_{k-1}}{\|(g_k, f^*(g_k)) - \bar{z}_{k-1}\|^2} \quad (6)$$

выполняется неравенство $\|\bar{z}_k\|^2 \leq \|\bar{z}_{k-1} + \lambda((g_k, f^*(g_k)) - \bar{z}_{k-1})\|^2$.

После возведения в степень получим

$$\|\bar{z}_k\|^2 \leq \|\bar{z}_{k-1}\|^2 - 2\lambda((\bar{z}_{k-1} - (g_k, f^*(g_k))) \bar{z}_{k-1} - \frac{\lambda}{2} \|(g_k, f^*(g_k)) - \bar{z}_{k-1}\|^2). \quad (7)$$

Согласно (6)

$$\begin{aligned} & (\bar{z}_{k-1} - (g_k, f^*(g_k))) \bar{z}_{k-1} - \frac{\lambda}{2} \|(g_k, f^*(g_k)) - \bar{z}_{k-1}\|^2 \geq \\ & \geq \frac{\lambda}{2} \|(g_k, f^*(g_k)) - \bar{z}_{k-1}\|^2 > 0 \text{ при } \lambda \neq 0. \end{aligned}$$

Для доказательства монотонного убывания для варианта 1 нормы векторов $\|\bar{z}_k\|$ достаточно в (7) учесть, что вычитаемое число в правой части положительно по последнему неравенству: $\|\bar{z}_k\|^2 < \|\bar{z}_{k-1}\|^2$.

Вариант 2. $\omega_k = -f(x_{k-1}) > \omega_{k-1}$. Тогда

$$\|\bar{z}_k\|^2 = \min_{z+(0, \omega_k) \in \text{co}D_k} \|z\|^2 \leq \|\bar{z}_\lambda\|^2, \quad (8)$$

где $\bar{z}_\lambda = \frac{\Omega - \omega_k}{\Omega - \omega_{k-1}} \bar{z}_{k-1} < \bar{z}_{k-1}$.

Легко увидеть, что при определенном таким образом \bar{z}_λ последнее неравенство в (8) выполняется. Поэтому $\|\bar{z}_k\|^2 < \|\bar{z}_{k-1}\|^2$.

Из монотонности нормы векторов $\|\bar{z}_k\|$ следует существование предела $\lim_{k \rightarrow \infty} \|\bar{z}_k\| = \rho$.

Для доказательства того, что $\rho = 0$, предположим противное. Пусть $\|\bar{z}_k\| \geq \tau r_k$ для некоторого $\tau > 0$.

Выполняется следующее равенство:

$$\lim_{k \rightarrow \infty} [((g_{k+1}, f^*(g_{k+1})) - (0, \omega_k)) \bar{z}_k - \|\bar{z}_k\|^2] = 0.$$

Предположим противное. Пусть для некоторой подпоследовательности имеем $((g_{k'+1}, f^*(g_{k'+1})) - (0, \omega_{k'})) \bar{z}_{k'} \leq \|\bar{z}_{k'}\|^2 - \gamma$, $\gamma > 0$. Тогда

$$\begin{aligned} \|\bar{z}_{k'+1}\|^2 &= \min_{z+(0, \omega_{k'}) \in \text{co}\{(g_i, f^*(g_i)), i=0, 1, \dots, k'+1\}, (0, \Omega)} \|\bar{z}\|^2 \leq \\ &\leq \min_{z+(0, \omega_{k'}) = \lambda \tilde{g}_{k'+1} + (1-\lambda)(g_{k'+1}, f^*(g_{k'+1}))} \|\bar{z}\|^2, \lambda \in [0, 1]. \end{aligned}$$

Учитывая, что $\tilde{g}_{k'} = (0, \omega_{k'}) + \bar{z}_{k'}$, получаем

$$\|\bar{z}_{k'+1}\|^2 \leq \min_{\lambda \in [0, 1]} \|\lambda \bar{z}_{k'} + (1-\lambda)((g_{k'+1}, f^*(g_{k'+1})) - (0, \omega_{k'}))\|^2. \quad (9)$$

После возведения в квадрат правой части неравенства (9) получим

$$\begin{aligned} \|\bar{z}_{k'+1}\|^2 &\leq \min_{\lambda \in [0, 1]} \{\lambda^2 \|\bar{z}_{k'}\|^2 + 2\lambda(1-\lambda) \bar{z}_{k'}((g_{k'+1}, f^*(g_{k'+1})) - (0, \omega_{k'})) + \\ &+ (1-\lambda)^2 \|(g_{k'+1}, f^*(g_{k'+1})) - (0, \omega_{k'})\|^2\}. \end{aligned} \quad (10)$$

Продолжим преобразования в правой части неравенства (10):

$$\begin{aligned} \|\bar{z}_{k'+1}\|^2 &\leq \min_{\lambda \in [0, 1]} \{\lambda^2 \|\bar{z}_{k'}\|^2 + 2\lambda(1-\lambda) \|\bar{z}_{k'}\|^2 - 2\gamma \lambda(1-\lambda) + \\ &+ (1-\lambda)^2 \|(g_{k'+1}, f^*(g_{k'+1})) - (0, \omega_{k'})\|^2\} = \\ &= \min_{\lambda \in [0, 1]} \{(2\lambda - \lambda^2) \|\bar{z}_{k'}\|^2 - 2\gamma \lambda(1-\lambda) + (1-\lambda)^2 \|(g_{k'+1}, f^*(g_{k'+1})) - (0, \omega_{k'})\|^2\}. \end{aligned} \quad (11)$$

Из (11) получена оценка

$$\begin{aligned} & \|\bar{z}_{k'+1}\|^2 \leq \\ \leq & \min_{\lambda \in [0,1]} \{ \|\bar{z}_{k'}\|^2 - 2\gamma \lambda(1-\lambda) + ((1-\lambda)^2 \| (g_{k'+1}, f^*(g_{k'+1})) - (0, \omega_{k'}) \|^2 - \|\bar{z}_{k'}\|^2) \} \leq \\ & \leq \|\bar{z}_{k'}\|^2 - 2\gamma \lambda(1-\lambda) + (1-\lambda)^2 \delta^2 \text{ для любого } \lambda \in [0,1]. \end{aligned}$$

Подставив в данное неравенство выражение $\lambda = (\delta^2 + \gamma) / (\delta^2 + 2\gamma) > 0$, получим

$$\|\bar{z}_{k'+1}\|^2 \leq \|\bar{z}_{k'}\|^2 - \gamma^2 / (\delta^2 + 2\gamma). \quad (12)$$

Переходя в (12) к пределу при $k' \rightarrow \infty$, получим противоречие. Поэтому $\lim_{k \rightarrow \infty} \lfloor ((g_{k+1}, f^*(g_{k+1})) - (0, \omega_k)) \bar{z}_k - \|\bar{z}_k\|^2 \rfloor = 0$.

Поскольку $0 \leq (\bar{z}_k + (0, \omega_k)) \bar{z}_k - (g_{k+1}, f^*(g_{k+1})) \bar{z}_k \rightarrow 0$ при $k \rightarrow \infty$, для достаточно больших k выполняется неравенство $(\bar{z}_k + (0, \omega_k)) \bar{z}_k - (g_{k+1}, f^*(g_{k+1})) \bar{z}_k \leq r_k^2 \varepsilon^2$ для любого $\varepsilon > 0$. Следовательно,

$$\begin{aligned} (0, \omega_{k+1}) \bar{z}_k & \geq (g_{k+1}, f^*(g_{k+1})) \bar{z}_k \geq (0, \omega_k) \bar{z}_k + \|\bar{z}_k\|^2 - r_k \varepsilon \geq \\ & \geq (0, \omega_k) \bar{z}_k + r_k^2 \tau^2 - r_k^2 \varepsilon^2 \geq (0, \omega_k) \bar{z}_k + r_k^2 \tau^2 / 2 \end{aligned}$$

для $\varepsilon \leq \tau / \sqrt{2}$, т.е. $r_k(0, \omega_{k+1}) \geq r_k(0, \omega_k) + r_k^2 \tau^2 / 2$.

Далее, $f^*(0) \geq (0, \omega_{k+1}) \geq (0, \omega_k) + r_k \tau^2 \geq (0, \omega_k) + \delta$, где $\delta \geq r_k \tau^2 \geq 0$, что невозможно при $k \rightarrow \infty$. Данное противоречие доказывает равенство $\lim_{k \rightarrow \infty} \|\bar{z}_k\| = 0$. Теорема доказана. ■

ПРИМЕНЕНИЕ АЛГОРИТМА SPASLIP ДЛЯ РЕШЕНИЯ ЗАДАЧ ТРАНСПОРТНОГО ТИПА

Рассмотрим матричную транспортную задачу с ограничениями на объемы поставок в стандартной постановке [30, 33]. Пусть m_1 — количество потребителей, n_1 — количество поставщиков, A_i ($i=1, \dots, n_1$) — запасы продукта у поставщиков, B_j ($j=1, \dots, m_1$) — потребности в продукте у потребителей, c_{ij} — цена поставки единицы продукта от i -го поставщика j -му потребителю, x_{ij}^{low} и x_{ij}^{up} — нижние и верхние границы на объемы поставок x_{ij} . Считаем задачу

$$\text{сбалансированной: } \sum_{i=1}^{n_1} A_i = \sum_{j=1}^{m_1} B_j.$$

Минимизация транспортных затрат с установлением балансов по поставщикам и потребителям и двусторонними границами на переменные приводит к классической задаче линейного программирования:

$$\min_{x_{ij} \in X_{ij}} \sum_{i=1}^{n_1} \sum_{j=1}^{m_1} c_{ij} x_{ij}, \quad (13)$$

где множество X_{ij} задает ограничения

$$\sum_{j=1}^{m_1} x_{ij} = A_i, i=1, \dots, n_1; \quad \sum_{i=1}^{n_1} x_{ij} = B_j, j=1, \dots, m_1; \quad (14)$$

$$x_{ij}^{low} \leq x_{ij} \leq x_{ij}^{up}, i=1, \dots, n_1; j=1, \dots, m_1, \quad (15)$$

или в матрично-векторном виде $\min_{x \in X} cx, X = \{x | Ax = b, x^{low} \leq x \leq x^{up}, b = (A_i, B_j)^T\}$.

Вместо (13)–(15) рассмотрим эквивалентную квадратичную задачу при тех же ограничениях:

$$\min_{x_{ij} \in X} \sum_{i=1}^{n_1} \sum_{j=1}^{m_1} \varepsilon \left(x_{ij} + \frac{c_{ij}}{2\varepsilon} \right)^2 = \min_{x \in X} \left\| x + \frac{c}{\varepsilon} \right\|^2 = \left\| P_X \left(-\frac{c}{\varepsilon} \right) + \frac{c}{\varepsilon} \right\|^2, \quad (16)$$

где $P_X(a)$ — результат решения задачи проекции точки a на множество X .

Развивая результаты работы [30], можно показать, что в действительности существует такое $\bar{\varepsilon} > 0$, что для всех $0 < \varepsilon < \bar{\varepsilon}$ задача линейного программирования (13)–(15) и задача проектирования (16) эквивалентны, и $P_X(-c\varepsilon^{-1})$ решает транспортную задачу (13)–(15).

Для формулировки этого результата введем некоторые дополнительные определения. Пусть $\hat{x} \in X$ — некоторая фиксированная точка. Обозначим $K_{\hat{x}} = \{z | \hat{x} + \lambda z \in X, \lambda > 0\}$ конус допустимых направлений в точке \hat{x} . В случае полиэдрального множества X конус $K_{\hat{x}}$ замкнут. Обозначим $K_{\hat{x}}^*$ конус, двойственный конусу $K_{\hat{x}}$: $K_{\hat{x}}^* = \{u | u z \leq 0, z \in K_{\hat{x}}\}$.

Заметим, что если $y = \hat{x} + u, u \in K_{\hat{x}}^*$, то для любого $x \in X$ выполняется неравенство $(y - \hat{x})(x - \hat{x}) = u(x - \hat{x}) \leq 0$, поскольку $x - \hat{x} \in K_{\hat{x}}$. Следовательно, $\hat{x} = P_X(y)$, т.е. \hat{x} является проекцией y на X .

Во введенных обозначениях данный результат может быть сформулирован как следующее утверждение.

Теорема 2. Пусть транспортная задача (13)–(15) имеет единственное решение \hat{x} . Тогда существует $\bar{\varepsilon} > 0$ такое, что

$$\left\| P_X \left(-\frac{c}{\varepsilon} \right) + \frac{c}{\varepsilon} \right\|^2 = \min_{x \in X} \left\| x + \frac{c}{\varepsilon} \right\|^2 = \left\| \hat{x} + \frac{c}{\varepsilon} \right\|^2 \text{ для всех } 0 < \varepsilon \leq \bar{\varepsilon}.$$

Доказательство. Определим для \hat{x} соответствующие конусы $K_{\hat{x}}$ и $K_{\hat{x}}^*$. В силу единственности решения имеем $-c(x - \hat{x}) < 0$ для любого $x \in X$. Следовательно, $-c \in \text{int } K_{\hat{x}}^*$, что доказывает и непустоту внутренности $\text{int } K_{\hat{x}}^*$. Тогда $-\frac{c}{\varepsilon} = \hat{x} - \frac{c}{\varepsilon} - \hat{x} = \hat{x} + \frac{1}{\varepsilon}(-\varepsilon\hat{x} - c) = \hat{x} + \frac{1}{\varepsilon}z_\varepsilon$, где $z_\varepsilon = -c - \varepsilon\hat{x} \in \text{int } K_{\hat{x}}^*$ для достаточно малых ε . Соответственно $\frac{1}{\varepsilon}z_\varepsilon \in K_{\hat{x}}^*$ и $\hat{x} = P_X\left(\hat{x} + \frac{1}{\varepsilon}z_\varepsilon\right) = P_X\left(-\frac{c}{\varepsilon}\right)$, что и требовалось доказать. ■

Приведенная теорема позволяет в условиях единственности решения задачи линейного программирования свести ее к специальной задаче квадратичного программирования с простейшей квадратичной формой, что намного упрощает решение.

ВЫЧИСЛИТЕЛЬНЫЕ ЭКСПЕРИМЕНТЫ

Транспортная задача с тремя поставщиками. Проекционный алгоритм SPACLIP был реализован на свободно распространяемой системе матрично-векторных вычислений Octave [35]. Синтаксис Octave весьма близок к MATLAB, и эта система является удобным инструментом для разработки пилотных версий вычислительных алгоритмов и быстрого прототипирования.

После проверки работоспособности реализованного алгоритма на ряде задач НДО данный метод был применен для решения оптимизационных задач транспортной логистики с помощью приемов, описанных в предыдущем разделе.

Вначале была решена транспортная задача с тремя поставщиками и четырьмя потребителями (соответственно $n_1 = 3$, $m_1 = 4$, а вектор x для этой задачи

Таблица 1. Результаты решения транспортной задачи с тремя поставщиками с помощью МОП и SPACLIP

Метод	Компоненты вектора x												Количество итераций	Транспортные затраты (13)	ε
	x_{11}	x_{12}	x_{13}	x_{14}	x_{21}	x_{22}	x_{23}	x_{24}	x_{31}	x_{32}	x_{33}	x_{34}			
МОП	0	0	61	136	150	27	0	0	0	97	90	0	75	1532	10^{-4}
SPACLIP	0	0	56	141	146	28	0	0	0	101	87	0	63	1527	10^{-4}
МОП	0	0	57	136	145	28	0	0	0	104	83	0	60	1506	10^{-5}
SPACLIP	0	0	55	145	146	24	0	0	0	99	90	0	54	1514	10^{-5}
МОП	0	0	54	148	148	24	0	0	0	101	83	0	46	1514	10^{-6}
SPACLIP	0	0	96	167	123	61	0	0	0	166	154	0	53	2024	10^{-6}
МОП	0	0	65	154	166	19	0	0	0	106	95	0	41	1628	10^{-7}
SPACLIP	0	0	30	59	81	54	0	0	0	108	200	0	100	1557	10^{-7}
МОП	0	0	61	152	138	62	0	0	0	103	83	0	35	1680	10^{-8}
SPACLIP	0	0	200	100	100	0	0	0	0	0	200	0	38	1401	10^{-8}

имеет размерность $3 \times 4 = 12$), приведенная в [30]. Заданная матрица тарифов $\{c_{ij}\}_{i,j=1}^{3,4}$ преобразуется в вектор $c = (7, 8, 1, 2, 4, 5, 9, 8, 9, 2, 3, 6)$. По условию запасы продукта составляют $A_i = [200, 180, 190]$, а потребности в продукте — $B_j = [150, 130, 150, 140]$. Нижняя граница x_{ij}^{low} была установлена равной нулю, верхняя граница $x_{ij}^{up} = 200$, $i = 1, \dots, n_1$, $j = 1, \dots, m_1$.

Результаты решения данной транспортной задачи приведены в табл. 1. При решении задачи оба метода завершали свою работу по условию близости к значению аргумента. Условие близости к нулю нормы градиента оптимального решения выполнено не было. Значение ε из (16) совпадает с заданной точностью для выбранных методов негладкой оптимизации — МОП и SPACLIP — и приведено в последнем столбце таблицы. Ухудшение результатов при $\varepsilon = 10^{-8}$ связано с ограничениями реализации языка Octave на разрядность. В целом оба метода справились с решением данной модельной задачи, правильно вычислив оптимальное распределение объемов продукта. При сравнении данных результатов с результатами решения задачи из [30] оказалось, что методы семейства отделяющих плоскостей более чувствительны к близости верхней границы x_{ij}^{up} к решению, чем r -алгоритм Шора [32]. Именно поэтому верхняя граница была установлена равной 200, при меньшем значении алгоритмы семейства отделяющих плоскостей в качестве оптимального решения выдавали вектор-комбинацию из заданных чисел-ограничений. У r -алгоритма таких проблем не возникло. Отметим, что в таблице приведены округленные до целого значения компоненты оптимального решения x , поэтому указанные затраты на транспортировку продукции могут незначительно отличаться от произведения $c \cdot x$.

Транспортные задачи малых и больших размерностей. После решения модельной задачи было проведено массовое тестирование проекционного алгоритма на серии транспортных задач малой и большой размерности со случайно сгенерированными данными.

Для каждой из этих задач $n_1 = m_1 = n$; вектор тарифов c генерируется случайным образом как вектор размерности n^2 , состоящий из случайных чисел, принадлежащих отрезку $[1, 101]$. Затем (также случайным образом) генерировался вектор x_{opt} размерности n^2 — допустимое решение задачи (с компонентами из диапазона $[1, 1001]$). Тогда вектор b , равный $[A_i, B_j]^T$, вычисляется по фор-

муле $b = A \cdot x_{\text{opt}}$. Нижние и верхние границы на объемы поставок задаются умножением допустимого решения на 0.1 и 20 соответственно. Значение ε из (16) приравнивается к заданной пользователем требуемой точности решения задачи. В качестве начального приближения возьмем случайный вектор размерности n^2 с компонентами из диапазона $[0, 10]$. Далее проводится тестирование на тысячах однотипных задач и результаты обрабатываются по методике из [34] с построением профилей производительности.

Профилем производительности (performance profile [34]) ρ_s для метода решения оптимизационной задачи называется функция распределения какого-либо измеримого показателя производительности. Вычисление профилей производительности позволяет визуализировать различия по эффективности нескольких оптимизационных методов.

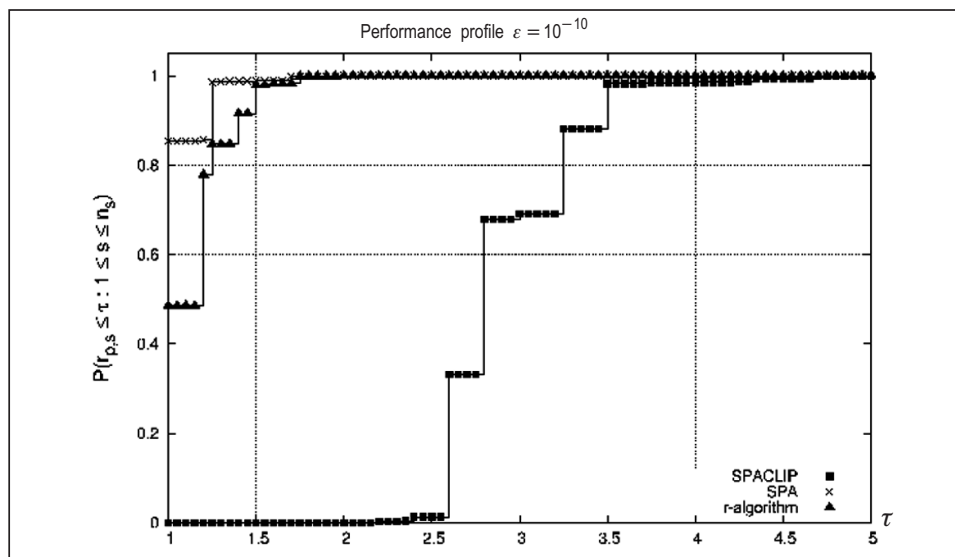


Рис. 4. Профили производительности для стандартного МОП (SPA), алгоритма SPACLIP и r -алгоритма, транспортная задача размерности 100 переменных

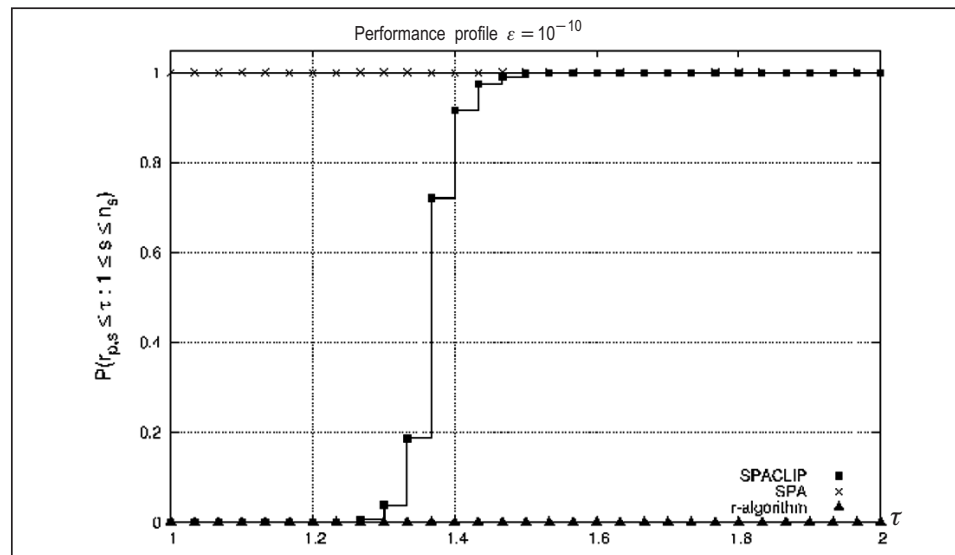


Рис. 5. Профили производительности для стандартного МОП (SPA), алгоритма SPACLIP и r -алгоритма, транспортная задача размерности 10000 переменных

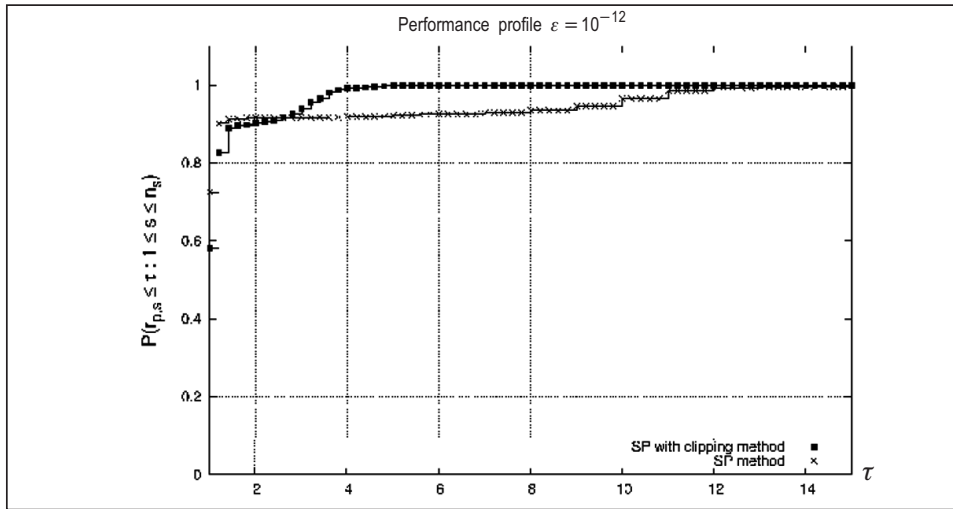


Рис. 6. Профили производительности для стандартного МОП и алгоритма SPACLIP, транспортная задача размерности 10000 переменных

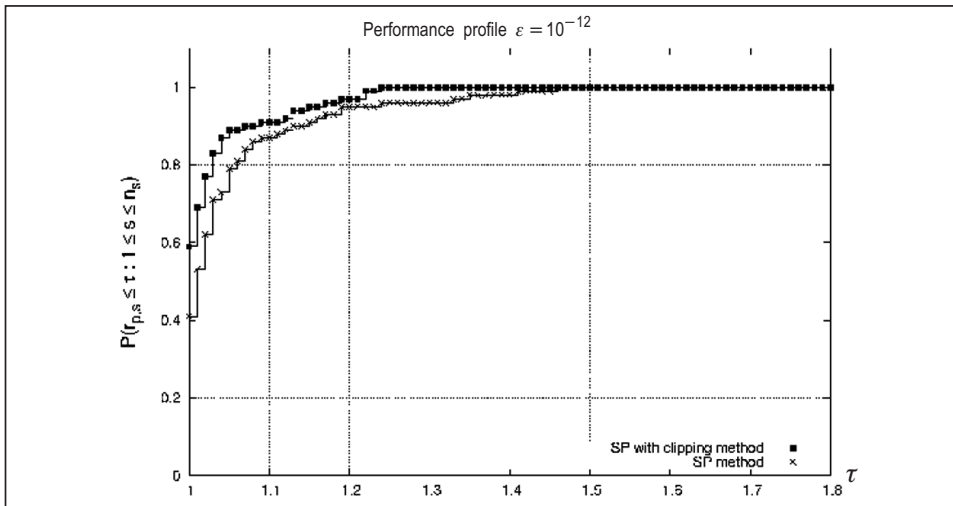


Рис. 7. Профили производительности для стандартного МОП и МОП с дополнительными отсечениями, транспортная задача размерности 40000 переменных

Функция распределения находится следующим образом:

$$\rho_s(\tau) = \frac{1}{n_p} |p \in P : r_{p,s} \leq \tau|; r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s} : s \in S\}}.$$

Здесь S — множество сравниваемых методов, P — множество решаемых с помощью этих методов задач. Количество элементов в P обозначено через n_p , n_s — количество элементов в S . В данном случае $n_s = 3$ (сравниваются три метода, в том числе r -алгоритм) или $n_s = 2$ (сравниваются стандартный МОП и SPACLIP), $n_p = 5000$. В качестве измеримого показателя производительности $t_{p,s}$ оценивалось затраченное на решение задачи процессорное время.

Тестирование проводилось на компьютере под управлением ОС Linux, дистрибутив openSUSE 12.3 Dartmouth, процессор AMD Athlon 64 3500+, 2 Гб оперативной памяти; версия интерпретатора Octave 3.6.4. Результаты тестирования приведены на рис. 4–7. Были решены задачи со 100 переменными ($n=10$), 10000 переменными ($n=100$), 40000 переменными ($n=200$). Для каждой размерности было решено 5000 транспортных задач.

Анализ полученных профилей производительности показывает, что при решении транспортных задач малых размерностей r -алгоритм работает быстрее, чем проекционный алгоритм SPACLIP. Стандартный МОП для задач таких размерностей также опережает проекционный алгоритм SPACLIP. Это связано с отсутствием необходимости решать задачу линейного программирования (2). Но чем больше становится размерность задачи и требуемая точность решения, тем больше преимущество SPACLIP по сравнению с другими методами.

ЗАКЛЮЧЕНИЕ

В настоящей статье предложен алгоритм SPACLIP решения задач выпуклой НДО семейства методов отделяющих плоскостей с дополнительными отсеками, порождаемыми решением вспомогательной задачи метода секущих плоскостей, и доказана его сходимости. В качестве практического применения алгоритма рассматривается решение транспортных задач большой размерности. В условиях единственности решения транспортной задачи доказана эквивалентность задач проекции достаточно удаленной точки на допустимое множество и задач транспортного типа. Вычислительные эксперименты демонстрируют довольно высокую эффективность SPACLIP при решении задач проекции.

СПИСОК ЛИТЕРАТУРЫ

1. Nann R.V., Woo G. Introduction to the theory and solution method for variational inequalities. — Changwon: Changwon National University Press, 2002. — 117 p.
2. Clarke F.N., Ledyaev Y.S., Stern R.J. and Wolenski P.R. Nonsmooth analysis and control theory. — New York: Springer, 1998. — 278 p.
3. Сергиенко И.В., Михалевич М.В., Стецюк П.И., Кошлай Л.Б. Модели и информационные технологии для поддержки принятия решений при проведении структурно-технологических преобразований // Кибернетика и системный анализ. — 2009. — № 2. — С. 26–49.
4. Сергиенко И.В. Методы оптимизации и системного анализа для задач транsvычислительной сложности. — Киев: Академперіодика, 2010. — 296 с. (укр.)
5. Шор Н.З., Стеценко С.И. Квадратичные экстремальные задачи и недифференцируемая оптимизация. — Киев: Наук. думка, 1989. — 208 с.
6. Nocedal J., Wright S. Numerical optimization. Series: Springer series in operations research and financial engineering. XXII. 2nd ed. — New York: Springer, 2006. — 664 p.
7. Поляк Б.Т. Введение в оптимизацию. — 2-е изд. — М.: ЛЕНАНД, 2014. — 392 с.
8. Немировский А.С., Юдин Д.Б. Сложность задач и эффективность методов оптимизации. — М.: Наука, 1979. — 383 с.
9. Нестеров Ю.Е. Введение в выпуклую оптимизацию. — М.: МЦНМО, 2010. — 280 с.
10. Левин А.Ю. Об одном алгоритме минимизации выпуклых функций // Доклады АН СССР. — 1965. — **160**, № 6. — С. 1244–1247.
11. Newman D.J. Location of maximum on unimodal surfaces // Journal of ACM. — 1965. — **12**. — P. 395–398.
12. Поляк Б.Т. Минимизация негладких функционалов // Журнал вычислительной математики и математической физики. — 1969. — **9**, № 3. — С. 509–521.
13. Nurminski E.A. Envelope stepsize control for iterative algorithms based on Fejer processes with attractants // Optimization Methods and Software. — 2010. — **25**, N 1. — P. 97–108.
14. Kiwiel K.C. An aggregate subgradient method for nonsmooth convex minimization // Mathematical Programming. — 1983. — N 27. — P. 320–341.
15. Lemarechal C. An extension of Davidon methods to nondifferentiable problems // Nondifferentiable Optimization (M.L. Balinski, P. Wolfe, eds.). Mathematical Programming Study. — 1975. — N 3. — P. 95–109.

16. Wolfe P. A method of conjugate subgradients for minimizing nondifferentiable functions // *Nondifferentiable Optimization* (M.L. Balinski, P. Wolfe, eds.). *Mathematical Programming Study*. — 1975. — N 3. — P. 145–173.
17. Lemarechal C., Nemirovskii A. and Nesterov Yu. New variants of bundle methods // *Mathematical Programming*. — 1995. — N 69. — P. 111–148.
18. Mifflin R., Sagastizabal C. A VU-algorithm for convex minimization // *Mathematical Programming*. — 2005. — N 104. — P. 583–608.
19. Nesterov Yu. *Introductory lectures on convex optimization: A basic course*. — Boston: Kluwer Academic Publishers, 2004. — 236 p.
20. Nesterov Yu. Smooth minimization of non-smooth functions // *Mathematical Programming*. — 2005. — N 103. — P. 127–152.
21. Nurminski E.A. Separating plane algorithms for convex optimization // *Mathematical Programming*. — 1997. — N 76. — P. 373–391.
22. Нурминский Е.А. Метод отделяющих плоскостей с ограниченной памятью для решения задач выпуклой негладкой оптимизации // *Вычислительные методы и программирование*. — 2006. — 7. — С. 133–137.
23. Vorontsova E.A. A projective separating plane method with additional clipping for non-smooth optimization // *WSEAS Transactions on Mathematics*. — 2014. — 13. — P. 115–121.
24. Демьянов В.Ф., Васильев Л.В. *Недифференцируемая оптимизация*. — М.: Наука, 1981. — 384 с.
25. Нурминский Е.А. *Численные методы выпуклой оптимизации*. — М.: Наука, 1991. — 168 с.
26. Нурминский Е.А. О сходимости метода подходящих аффинных подпространств для решения задачи о наименьшем расстоянии до симплекса // *Журнал вычислительной математики и математической физики*. — 2005. — 45, вып. 11. — С. 1996–2004.
27. Nurminski E. A quadratically convergent line-search algorithm for piecewise smooth convex optimization // *Optimization Methods and Software*. — 1995. — N 6. — P. 59–80.
28. Воронцова Е.А. Быстро сходящийся алгоритм линейного поиска в недифференцируемой оптимизации // *Информатика и системы управления*. — 2012. — № 2. — С. 39–48.
29. Стецюк П.И., Нурминский Е.А. Негладкий штраф и субградиентные алгоритмы для решения задачи проекции на политоп // *Кибернетика и системный анализ*. — 2010. — № 1. — С. 59–63.
30. Стецюк П.И., Нурминский Е.А., Соломон Д.И. Транспортная задача и ортогональное проектирование на линейные многообразия // *Материалы V международной научной конференции «Транспортные системы и логистика» (Кишинев, Молдова, 11–13 декабря 2013 г.)*. — Кишинев: Эврика, 2013. — С. 251–263.
31. Пшеничный Б.Н. *Выпуклый анализ и экстремальные задачи*. — М.: Наука, Главная редакция физико-математической литературы, 1980. — 320 с.
32. Шор Н.З. *Методы недифференцируемой оптимизации и сложные экстремальные задачи*. — Кишинев: Эврика, 2008. — 270 с.
33. Юдин Д.Б., Гольштейн Е.Г. *Задачи и методы линейного программирования: Задачи транспортного типа*. — 3-е изд. — М.: Книжный дом «Либроком», 2010. — 184 с.
34. Dolan E., More J. Benchmarking optimization software with performance profiles // *Mathematical Programming*. — 2002. — N 91. — P. 201–213.
35. Octave Page [Электронный ресурс]. — Режим доступа: <http://www.gnu.org/software/octave/>.
36. Kelley J.E. The cutting plane method for solving convex programs // *Journal of the SIAM*. — 1960. — 8 (4). — P. 703–712.
37. Шор Н.З. Применение метода градиентного спуска для решения сетевой транспортной задачи // *Материалы научного семинара по теоретическим и прикладным вопросам кибернетики и исследований операций: Научный совет по кибернетике АН УССР*. — 1962. — Вып. 1. — Киев. — С. 9–17.

Поступила 27.10.2014