

МОДИФИЦИРОВАННЫЙ МЕТОД СИМПЛЕКСНЫХ ПОГРУЖЕНИЙ ДЛЯ РЕШЕНИЯ ЗАДАЧ ВЫПУКЛОЙ ОПТИМИЗАЦИИ С БОЛЬШИМ ЧИСЛОМ ОГРАНИЧЕНИЙ

А. В. Колосницын¹

Рассматривается метод симплексных погружений, адаптированный для решения задач выпуклой оптимизации с большим числом ограничений. Разработаны две модификации, позволяющие ускорять работу метода. Первая из них использует более экономичный способ расчета невязок ограничений, что позволяет существенно сокращать время работы алгоритма в случае большой размерности задачи. Вторая модификация основана на возможности метода определять неактивные ограничения задачи. Представлены результаты вычислительных экспериментов с использованием модифицированных версий метода симплексных погружений при решении тестовых задач квадратичной и выпуклой недифференцируемой оптимизации.

Ключевые слова: метод симплексных погружений, выпуклая недифференцируемая оптимизация, идентификация неактивных ограничений.

1. Введение. Впервые метод симплексных погружений был представлен в работах [1–3] для решения задачи выпуклой оптимизации. В [1] предложен аналог известного метода эллипсоидов [4, 5] с использованием n -мерных симплексов вместо эллипсоидов для аппроксимации допустимой области задачи. Идеи, изложенные в [1], были в дальнейшем развиты в работах [6, 7].

В основе метода лежит известная идея построения отсекающей плоскости через центр некоторого выпуклого множества. Еще в 1965 г. независимо друг от друга А. Ю. Левин [8] и Д. Ньюман [9] предложили метод, в котором отсекающие плоскости проводились через центр тяжести некоторого выпуклого множества. При этом данное множество разделяется на две части, а отношение объема любой из этих частей к объему исходного множества не превосходит величину, равную $1 - 1/e$, где e — основание натурального логарифма.

Несмотря на то что метод центров тяжести является оптимальным (т.е. неуплучшаемым среди всех методов негладкой выпуклой оптимизации, использующих только значение субградиента [4, 10]), он представляет лишь теоретический интерес, поскольку нахождение центра тяжести выпуклого множества является NP -трудной задачей [11]. Тем не менее, идея данного метода послужила основой для ряда новых методов оптимизации, использующих различные аппроксимирующие допустимую область множества, центр тяжести которых вычисляется достаточно просто. К этому классу методов можно отнести методы описанных и вписанных эллипсоидов, метод объемного центра, метод аналитических центров [12, 13] и метод симплексных погружений, рассматриваемый в настоящей статье.

Дальнейшие исследования метода симплексных погружений, опубликованные в [7], позволили получить оценку слабой скорости сходимости. Оказалось, что скорость сходимости метода, согласно такой оценке, зависит только от числа отсеченных вершин симплекса. Мы представим данную оценку более подробно в отдельном разделе статьи.

Для метода симплексных погружений разработан ряд модификаций, позволяющих увеличить скорость сходимости. В работе [14] изложены идеи введения нескольких отсекающих плоскостей в центре симплекса, что позволяет отсекать большее число его вершин. Данный подход получил развитие в работе [15] при решении специального класса задач выпуклой недифференцируемой оптимизации. Представленная в [16] модификация метода симплексных погружений использует технику построения симплекса минимального объема, который описывается вокруг заданного набора точек. Такая процедура во многом аналогична известному способу нахождения эллипсоида минимального объема, содержащего заданное множество точек [12]. И наконец, данные о вычислительной эффективности модификаций метода симплексных погружений с использованием сдвига отсекающей плоскости представлены в [17]. Тестирование в работе [17] осуществлялось на наборе задач выпуклой недифференцируемой оптимизации [18, 19].

¹ Институт систем энергетики им. Л. А. Мелентьева СО РАН, ул. Лермонтова, 130, 664033, Иркутск; мл. науч. сотр., e-mail: ankolos25@mail.ru

В настоящей статье мы предлагаем эффективную версию метода симплексных погружений, адаптированную для решения задач выпуклой оптимизации с большим числом ограничений. Эта версия метода протестирована на наборе задач выпуклой оптимизации как с квадратичными, так и с недифференцируемыми ограничениями.

В статье приняты стандартные обозначения: $\|\cdot\|$ — евклидова норма; $\|x\|_1 = \sum_{i=1}^n |x_i|$; $(x, y) = \sum_{i=1}^n x_i y_i$ для векторов $x \in \mathbb{R}^n$ и $y \in \mathbb{R}^n$.

2. Идея метода симплексных погружений и основные определения. Введем ряд необходимых определений и обозначений.

Определение 1. Симплексом $S \subset \mathbb{R}^n$ с вершинами x^0, x^1, \dots, x^n называется множество

$$S = \left\{ x \in \mathbb{R}^n : x = \sum_{i=0}^n \sigma_i x^i, \sigma_i \geq 0, \sum_{i=0}^n \sigma_i = 1 \right\}.$$

Центр симплекса S находится по формуле $\theta = \frac{1}{n+1} (x^0 + \dots + x^n)$. Объем симплекса S определяется как $V(S) = \frac{1}{n!} |\det(X)|$, где X — матрица размеров $n \times n$, столбцы которой состоят из векторов $x^j - x^0$, $j = 1, \dots, n$. Далее мы будем предполагать, что векторы $x^j - x^0$, $j = 1, \dots, n$, линейно независимы; следовательно, $\det(X) \neq 0$ и внутренность $\text{int}(S) \neq \emptyset$.

Определение 2. Для любого вектора $g \in \mathbb{R}^n$ определим

- отсекающую плоскость вида $L = \{x : (g, x - \theta) = 0\}$;
- полупространство $G = \{x : (g, x - \theta) \leq 0\}$;
- усеченный симплекс $S_G = S \cap G$.

Определение 3. Вершина x^i симплекса S не отсекается плоскостью L , если $\alpha_i = (g, x^i - \theta) < 0$, и отсекается, если $\alpha_i \geq 0$.

Обозначим через ν количество неотсеченных (сохраняемых) вершин симплекса S при проведении отсекающей плоскости L . В силу предположения имеем $\text{int}(S) \neq \emptyset$, $\nu > 0$. Число вершин усеченного симплекса S_G при этом равно $\nu + \nu(n - \nu + 1)$.

Введем $p : \alpha_p = \min_{0 \leq i \leq n} \alpha_i$. Так как $\alpha_i = (g, x^i - \theta)$, $i = 0, \dots, n$, то

$$\sum_{i=0}^n \alpha_i = (n+1) \left(g, \frac{1}{n+1} \sum_{i=0}^n x^i - \theta \right) = (n+1)(g, \theta - \theta) = 0. \tag{1}$$

Пусть α_p — минимальное из чисел $\alpha_0, \alpha_1, \dots, \alpha_n$. Величина α_p не может быть положительной: это противоречит равенству (1) и неравенствам $\alpha_p \leq \alpha_i$ для всех i . Аналогично, если $\alpha_p = 0$, то $\alpha_i = 0 \forall i$. В этом случае все вершины симплекса и, следовательно, сам симплекс лежат в плоскости $\{x \in \mathbb{R}^n : (g, x - \theta) = 0\}$. Тогда можно перейти в пространство меньшей размерности и продолжить алгоритм. Такая ситуация считается исключительной и в данной статье не рассматривается. Остается случай $\alpha_p < 0$, что и предполагается в алгоритме.

Вершину x^p будем называть опорной. Таких вершин может быть несколько, тогда в качестве опорной выбирается любая из них. В [7] предполагается, что $p = 0$.

Приведем описание основной идеи метода симплексных погружений применительно к решению следующей задачи оптимизации:

$$f_0(x) \rightarrow \min, \quad f_i(x) \leq 0, \quad i = 1, \dots, m, \quad \underline{x} \leq x \leq \bar{x}. \tag{2}$$

Здесь $x \in \mathbb{R}^n$; $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 0, \dots, m$, — выпуклые и не обязательно гладкие функции; $\underline{x}, \bar{x} \in \mathbb{R}^n$; $\underline{x} \leq \bar{x}$. Обозначим через Q допустимую область задачи (2) и будем предполагать, что решение x^* данной задачи существует.

Метод симплексных погружений работает следующим образом. Строится симплекс S_0 , содержащий оптимальное решение x^* . Определяется центр построенного симплекса θ_0 , через который проводится отсекающая плоскость вида $L_0 = \{x \in \mathbb{R}^n : (g_0, x - \theta_0) = 0\}$, где $g_0 \in \mathbb{R}^n$ — градиент, а в случае решения задачи недифференцируемой оптимизации — субградиент функции f_s в точке θ_0 . Индекс s определяется из алгоритма метода и может соответствовать как целевой, так и одной из функций ограничений. Плоскость L_0 делит симплекс на две части, одна из которых содержит оптимальное решение. Часть симплекса, содержащая решение, погружается в новый симплекс S_1 минимально возможного объема. Для

полученного симплекса S_1 вновь ищется центр и проделываются все описанные выше процедуры. Такой процесс продолжается до тех пор, пока не сработает критерий останова, в качестве которого можно выбрать получение достаточно малого объема симплекса или достаточно малой длины максимального ребра симплекса.

3. Симплекс минимального объема и оценка скорости сходимости. Важным принципом работы метода симплексных погружений является процедура построения нового симплекса минимального объема, содержащего заданный усеченный симплекс. Пусть имеется некоторый симплекс S с центром в точке θ и отсекающей плоскостью $L = \{x \in \mathbb{R}^n : (g, x - \theta) = 0\}$. Определим усеченный симплекс $S_G = \{x \in S : (g, x - \theta) \leq 0\}$ и симплекс $S(\tau)$:

$$S(\tau) = \left\{ x \in \mathbb{R}^n : x = x^p + \sum_{i=0, i \neq p}^n \sigma_i [\tau_i (x^i - x^p)], \sum_{i=0, i \neq p}^n \sigma_i \leq 1, \sigma_i \geq 0, i = 0, \dots, n, i \neq p \right\},$$

где $\tau \in R_+^n = \{\tau \in \mathbb{R}^n : \tau \geq 0\}$ — вектор параметров, который определяет сжатие или растяжение направляющих векторов $(x^i - x^p)$ ребер симплекса S исходя из условия $S(\tau) \supset S_G$. Следующая теорема позволяет найти параметры τ и построить симплекс минимального объема $S(\tau)$ [7].

Теорема 1. Симплекс минимального объема $S(\tau^*)$, содержащий заданный усеченный симплекс S_G , определяется параметрами

$$\tau_i^* = 1 + \alpha_i t^*, \quad i = 0, 1, \dots, p-1, p+1, \dots, n,$$

где t^* — решение задачи выпуклой одномерной минимизации

$$\min_{0 \leq t \leq -1/\alpha_p} \prod_{i=0, i \neq p}^n (1 + \alpha_i t)^{-1}, \quad \alpha_p = \min_{0 \leq i \leq n} \alpha_i. \quad (3)$$

Задачу (3) можно переписать в эквивалентной и более удобной форме. Введем обозначения

$$\beta_i = -\frac{\alpha_i}{\alpha_p}, \quad i = 0, \dots, n, \quad q(\beta, t) = \prod_{i=0, i \neq p}^n (1 + \beta_i t)^{-1}.$$

Тогда вместо задачи (3) решается следующая задача

$$q^* = \min_{0 \leq t \leq 1} \prod_{i=0, i \neq p}^n (1 + \beta_i t)^{-1} = \min_{0 \leq t \leq 1} q(\beta, t). \quad (4)$$

Еще одна важная особенность метода связана с оценкой скорости сокращения объемов симплексов, которую мы приведем в виде следующей теоремы [7].

Теорема 2. Пусть $S \subset \mathbb{R}^n$ — n -мерный симплекс, θ — его центр, S_G — усеченный симплекс, соответствующий единичному вектору нормали g отсекающей плоскости. Тогда усеченный симплекс S_G всегда можно погрузить в симплекс $S^* = S(\tau^*)$, такой, что для объемов $V(S)$ и $V(S^*)$ симплексов S и S^* будет выполняться неравенство

$$q^* = \frac{V(S^*)}{V(S)} \leq \begin{cases} 1/2, & \nu = 1, \\ \left(\frac{\nu}{\nu+1}\right)^\nu \left(\frac{\nu}{\nu-1}\right)^{\nu-1}, & 2 \leq \nu \leq n, \end{cases} \quad (5)$$

где ν — число сохраненных при отсечении вершин симплекса.

Поскольку сумма чисел отсеченных и неотсеченных вершин симплекса постоянна и равна $n+1$, то можно сказать, что оценка скорости сокращения объемов симплексов (5) зависит от числа отсеченных вершин симплекса, причем при отсечении n вершин симплекса мы получаем аналог метода дихотомии. Такая интерпретация скорости сходимости более содержательна, так как чем больше вершин отсекается, тем выше скорость сходимости.

4. Алгоритм метода симплексных погружений. Приведем подробное описание алгоритма метода симплексных погружений при решении задачи (2). В основу представленного ниже алгоритма положена версия из [7] с использованием иного критерия останова. Везде ниже обозначение x_k^i соответствует вектору с номером i на итерации k .

Шаг 0. Выбрать начальный симплекс $S_0 = co\{x_0^0, x_0^1, \dots, x_0^n\}$, имеющий объем v_0 , $S_0 \supset x^*$. Координаты начального симплекса определить таким образом:

$$\begin{aligned} x_{0j}^0 &= \underline{x}_j, \quad j = 1, \dots, n, \\ x_{0j}^i &= \sum_{j=1}^n \bar{x}_j - \sum_{j=1, i \neq j}^n \underline{x}_j, \quad i = j, \\ x_{0j}^i &= \underline{x}_j, \quad i \neq j, \quad i = 1, \dots, n. \end{aligned}$$

Задать параметры $\varepsilon_1, \varepsilon_2$, определяющие точность решения. Итерация $k, k = 0, 1, 2, \dots$, осуществляется следующим образом.

Шаг к.1. Найти центр симплекса: $\theta_k = \frac{1}{n+1} (x_k^0 + \dots + x_k^n)$.

Шаг к.2. Найти длину максимального ребра симплекса S_k : $\rho_k = \max_{0 \leq i, j \leq n} \|x_k^i - x_k^j\|, i \neq j$, где $x_k^i, i = 0, \dots, n$, — i -я вершина симплекса S_k .

Шаг к.3.1. Если $\rho_k \leq \varepsilon_1$, то завершить алгоритм; θ_k — ε_1 -оптимальное решение задачи (2). Иначе перейти на Шаг к.3.2.

Шаг к.3.2. Если $v_k \leq \varepsilon_2$, завершить алгоритм; θ_k — ε_2 -оптимальное решение задачи (2). Иначе перейти на Шаг к.4.

Шаг к.4. Вычислить максимальную невязку ограничений $r_k = \max_{1 \leq i \leq m} \{0, f_i(\theta_k)\} = f_{s_k}(\theta_k)$.

Шаг к.5. Если $r_k = 0$, то определить нормаль отсекающей плоскости g_k как субградиент функции f_0 в точке θ_k . Если $r_k \neq 0$, то задать нормаль отсекающей плоскости как субградиент функции f_{s_k} в точке θ_k .

Шаг к.6. Найти параметры $\alpha_{ik} = (g_k, x_k^i - \theta_k), i = 0, \dots, n$. Определить индекс p из условия $\alpha_p = \min_{0 \leq i \leq n} \alpha_{ik}$. Найти величины $\beta_{ik} = -\alpha_{ik}/\alpha_p, i = 0, \dots, n, i \neq p$.

Шаг к.7. Найти параметр $t_k \in [0, 1]$ и q_k^* из решения задачи (4) при $\beta_i = \beta_{ik}$. Рассчитать коэффициенты растяжения $\tau_k = (\tau_{1k}, \dots, \tau_{p-1k}, \tau_{p+1k}, \dots, \tau_{nk})$ ребер симплекса S_k , исходящих из вершины x_k^p : $\tau_{ik} = (1 + \beta_{ik}t_k)^{-1}$.

Шаг к.8. Определить вершины симплекса S_{k+1} , которые вычисляются по правилу

$$x_{k+1}^i = x_k^p + \tau_{ik}(x_k^i - x_k^p), \quad i = 0, \dots, p-1, p+1, \dots, n, \quad x_{k+1}^p = x_k^p.$$

Шаг к.9. Определить объем симплекса S_{k+1} по формуле $v_{k+1} = q_k^* v_k$. Положить $k \rightarrow k+1$ и перейти на Шаг к.1.

Комментарий к алгоритму. На Шаге к.3 используется критерий останова по длине максимального ребра симплекса. Как только данная величина становится меньше заданного значения ε_1 , алгоритм останавливается. Такой критерий обеспечивает сильную сходимость метода (т.е. сходимость по аргументу). Кроме того, проверяется критерий останова по величине объема симплекса (слабая сходимость), что обеспечивает конечное число итераций алгоритма в силу доказанной в [7] сходимости метода при использовании данного критерия.

5. Модифицированные версии метода симплексных погружений. Очевидно, что при решении задач вида (2) с большим числом ограничений (мы рассматриваем число ограничений порядка $m \geq 10n$) важно сократить время работы алгоритма, которое тратится на их обработку. Мы предлагаем две модификации, позволяющие достичь существенной экономии как времени работы алгоритма, так и количества итераций на поиск решения.

Модификация 1. Ускорение вычисления невязок ограничений. Обратим внимание на Шаг к.4 алгоритма, представленного в предыдущем разделе. Перебор всех ограничений с целью вычисления максимальной невязки — это достаточно трудоемкая операция. Однако, как показали многочисленные расчеты, можно достичь весьма значительной экономии времени на данном шаге, ограничившись вычислениями лишь до первой положительной невязки. Мы находим такой номер невязки s_k :

$$f_i(\theta_k) \leq 0, \quad i = 1, \dots, s_k - 1, \quad f_{s_k}(\theta_k) > 0, \quad s_k \leq m.$$

Модификация 2. Идентификация неактивных ограничений. Одной из уникальных особенностей метода симплексных погружений является возможность идентифицировать неактивные ограничения даже для задач недифференцируемой выпуклой оптимизации. Данная процедура позволяет исключать такие ограничения из работы алгоритма для экономии времени при расчете невязок ограничений.

Описание рассматриваемой модификации приведено в [17]. Дадим для нее более детальное исследование. Пусть на k -й итерации метода построен симплекс S_k с вершинами x_k^j , $j = 0, \dots, n$. Кроме того, имеется список ограничений $f_i(x) \leq 0$, $i = 1, \dots, m$, задающих допустимую область. Чтобы определить, какие из данных ограничений являются неактивными, необходимо проверить следующее условие:

$$f_i(x_k^j) \leq 0, \quad i = 1, \dots, m, \quad j = 0, \dots, n. \quad (6)$$

Если условие (6) выполняется для i -го ограничения и для каждой вершины x_k^j , $j = 0, \dots, n$, то в силу выпуклости функции f_i это ограничение является неактивным и исключается из работы алгоритма.

Опишем использование второй модификации в схеме алгоритма.

Шаг 0. Дополнительно ввести параметры $C \in [50, 100]$, $c \in [5, 10]$, $M = 0$. Перейти на Шаг к.1.

После Шага к.3.2 перейти на Шаг к.4.0.

Шаг к.4.0. Если $M=0$ и k ратно $C \cdot n$, то выполнить проверку условия (6) и исключить неактивные ограничения. Положить $M = 1$.

Если $M=1$ и k кратно $c \cdot n$, то выполнить проверку условия (6) и исключить неактивные ограничения. Перейти на Шаг к.4.

Первая проверка неактивных ограничений осуществляется через Cn итераций, а последующие — через каждые cn итераций, где константы C и c определяются на Шаге 0 алгоритма. Такая эвристическая настройка модификации позволяет сократить время работы алгоритма, несмотря на достаточно трудоемкую процедуру выявления неактивных ограничений. Диапазон выбора констант C и c определялся исходя из анализа динамики исключения неактивных ограничений для разных размерностей задач. Соответствующие результаты представлены в разделах 6 и 7 на рис. 1 и 2.

6. Результаты вычислительных экспериментов. Разработанные модификации метода симплексных погружений были протестированы на двух классах задач выпуклой оптимизации (ВО).

1. Задача ВО с квадратичными ограничениями:

$$\sum_{j=1}^n \frac{1}{2} j (x_j - j)^2 \rightarrow \min_x, \quad a_i x + \gamma_i \|x - x^*\|^2 \leq b_i, \quad i = 1, \dots, m, \quad \underline{x} \leq x \leq \bar{x}. \quad (7)$$

2. Задача ВО с недифференцируемыми ограничениями:

$$\sum_{j=1}^n |x_j - j| \rightarrow \min_x, \quad a_i x + \gamma_i \|x - x^*\|_1 \leq b_i, \quad i = 1, \dots, m, \quad \underline{x} \leq x \leq \bar{x}. \quad (8)$$

Решением в задачах (7) и (8) является вектор $x^* = (1, 2, \dots, n)$. Элементы матрицы $A = \{a_{ij}\}$, $i = 1, \dots, m$, $j = 1, \dots, n$, генерировались случайным образом, а вектор $b \in \mathbb{R}^m$ формировался в виде $b_i = a_i x^* + \delta_i$, где δ_i , $i = 1, \dots, m$, — положительные константы.

Сгенерированные задачи вида (7) и (8) решались методом симплексных погружений, алгоритм которого был написан в среде Matlab. Под базовым алгоритмом будем понимать версию из [7] со следующими критериями останова: $\rho_k \leq \varepsilon_1$, $v_k \leq \varepsilon_2$, где ρ_k — величина максимального ребра симплекса на k -й итерации, v_k — объем симплекса на k -й итерации, ε_1 , ε_2 — заданные константы. Все тестируемые в статье модификации тоже снабжались упомянутыми критериями останова. Для расчетов использовался персональный компьютер с 8-ядерным процессором AMD FX-8350 (4 ГГц) и оперативной памятью 8 Гб. Для более представительного сравнения мы приведем и результаты решения задач (7) и (8) методом эллипсоидов [12], который, как и метод симплексных погружений, не требует решения вспомогательных задач оптимизации при поиске центра множества. Константа ε_3 вводится для критерия останова в методе эллипсоидов.

Введем обозначения: n — число переменных; m — число ограничений; iter — количество итераций; time — время работы алгоритма в секундах; Max — базовый метод симплексных погружений с построением отсекающей плоскости по максимальной невязке ограничения; First — метод симплексных погружений с построением отсекающей плоскости по первой положительной невязке ограничения; MaxAct — метод Max с идентификацией неактивных ограничений; FirstAct — метод First с идентификацией неактивных ограничений; Els — метод эллипсоидов.

Таблица 1
 Результаты работы модифицированных методов симплексных погружений,
 $\varepsilon_1 = 10^{-6}$, $\varepsilon_2 = 10^{-6}$

| $n \times m$ | Max | | First | | MaxAct | | FirstAct | |
|--------------|-------|-------|-------|-------|--------|-------|----------|-------|
| | iter | time | iter | time | iter | time | iter | time |
| 25 × 1000 | 1287 | 1.4 | 1399 | 1.8 | 1392 | 0.9 | 1399 | 1.9 |
| 50 × 1000 | 4321 | 7.2 | 3870 | 7.3 | 4209 | 5.3 | 3870 | 5.7 |
| 75 × 1000 | 7834 | 18.9 | 6693 | 17.6 | 7568 | 14.3 | 6693 | 13.2 |
| 100 × 1000 | 11471 | 33.2 | 10325 | 31.7 | 11703 | 26.7 | 10325 | 24.9 |
| 150 × 1500 | 22141 | 214.8 | 18502 | 177.4 | 21935 | 167.0 | 18502 | 136.7 |
| 200 × 2000 | 33628 | 568.1 | 27157 | 423.7 | 33552 | 433.6 | 27157 | 331.3 |

Таблица 2
 Результаты работы метода
 эллипсоидов, $\varepsilon_3 = 10^{-6}$

| $n \times m$ | Els | |
|--------------|---------|--------|
| | iter | time |
| 25 × 1000 | 20444 | 5.0 |
| 50 × 1000 | 78404 | 30.7 |
| 75 × 1000 | 173095 | 101.7 |
| 100 × 1000 | 302205 | 228.6 |
| 150 × 1500 | 661533 | 2377.2 |
| 200 × 2000 | 1149357 | 6755.9 |

Таблица 3
 Результаты работы модифицированных методов симплексных погружений,
 $\varepsilon_1 = 10^{-6}$, $\varepsilon_2 = 10^{-6}$

| $n \times m$ | Max | | First | | MaxAct | | FirstAct | |
|--------------|--------|--------|--------|--------|--------|-------|----------|-------|
| | iter | time | iter | time | iter | time | iter | time |
| 25 × 1000 | 5540 | 3.2 | 4431 | 4.2 | 5540 | 2.3 | 5351 | 2.7 |
| 50 × 1000 | 18638 | 17.9 | 14979 | 20.0 | 18638 | 12.9 | 16481 | 12.1 |
| 75 × 1000 | 36696 | 48.0 | 29280 | 49.9 | 36696 | 33.7 | 32121 | 31.5 |
| 100 × 1000 | 59270 | 107.6 | 46460 | 103.6 | 59270 | 79.1 | 51326 | 71.6 |
| 150 × 1500 | 117716 | 534.2 | 92778 | 462.1 | 117716 | 353.1 | 100773 | 266.3 |
| 200 × 2000 | 190302 | 1405.1 | 148120 | 1258.7 | 190302 | 995.5 | 162205 | 702.0 |

Таблица 4
 Результаты работы метода
 эллипсоидов, $\varepsilon_3 = 10^{-6}$

| $n \times m$ | Els | |
|--------------|---------|--------|
| | iter | time |
| 25 × 1000 | 20381 | 4.6 |
| 50 × 1000 | 78474 | 30.1 |
| 75 × 1000 | 172444 | 96.7 |
| 100 × 1000 | 300712 | 218.6 |
| 150 × 1500 | 657670 | 1488.4 |
| 200 × 2000 | 1144307 | 4343.8 |

В табл. 1 представлены результаты расчетов задачи (7) для четырех вариантов метода симплексных погружений, в табл. 2 приведены результаты работы метода эллипсоидов. В табл. 3 представлены результаты расчетов задачи (8) для четырех вариантов метода симплексных погружений, в табл. 4 приведены результаты работы метода эллипсоидов.

На рис. 1 изображены графики сокращения времени на одну итерацию и динамика исключения неак-

тивных ограничений для модификаций MaxAct и FirstAct при решении тестовых задач с квадратичными ограничениями. Рис. 2 содержит графики с аналогичными характеристиками для модификации MaxAct и FirstAct при решении тестовых задач с недифференцируемыми ограничениями. Оба типа задач решались при $n = 10^2$, $m = 10^3$, $\varepsilon_1 = 10^{-6}$, $\varepsilon_2 = 10^{-6}$. На рис. 3 показана динамика сокращения длины максимального ребра симплекса для модификации FirstAct при разном количестве переменных.

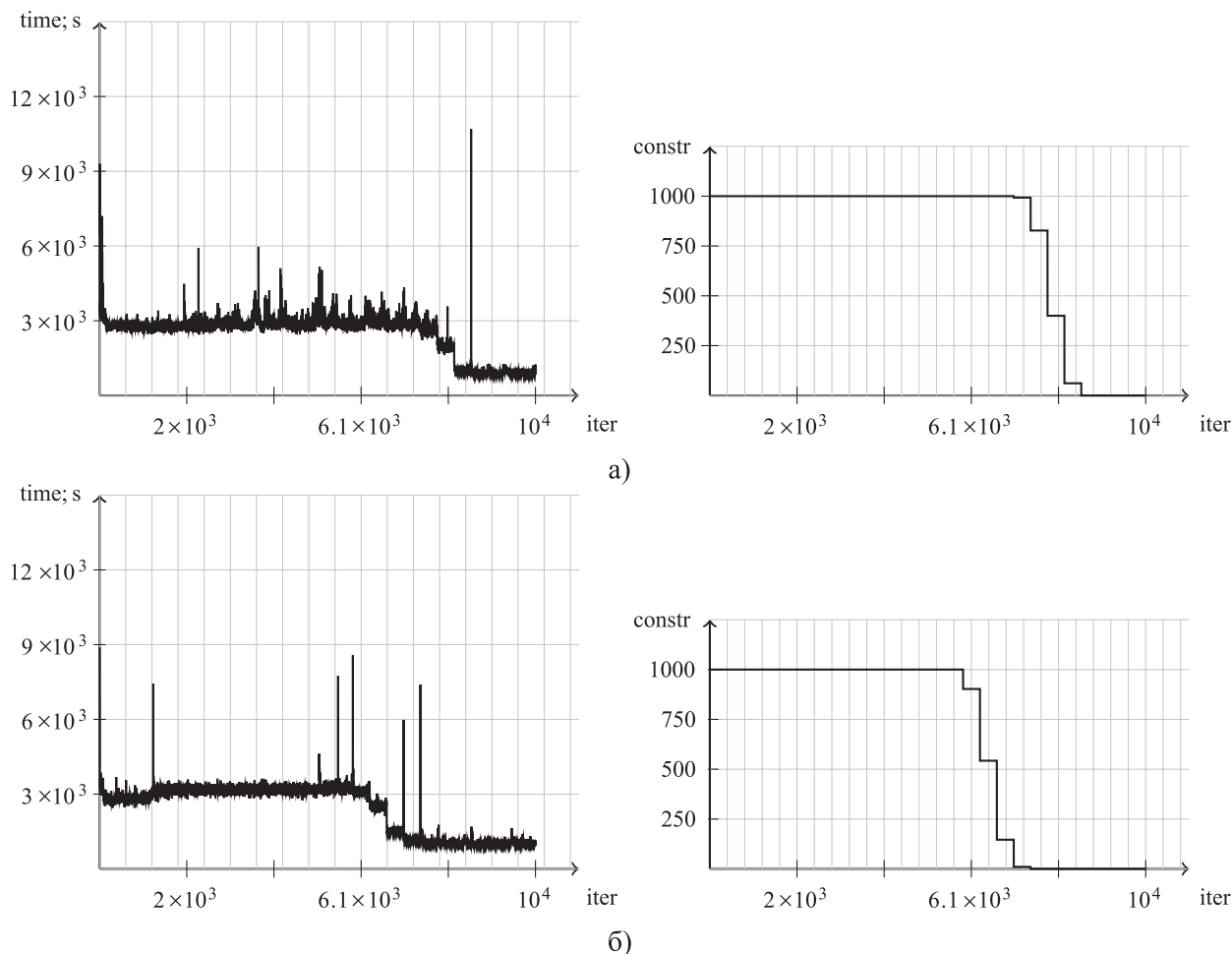


Рис. 1. Время на одну итерацию и динамика исключения неактивных ограничений для модификаций MaxAct (а) и FirstAct (б) при решении задач с квадратичными ограничениями

7. Обсуждение результатов. Наиболее эффективной модификацией для решения задач выпуклой квадратичной оптимизации (7) в большинстве тестов оказалась модификация FirstAct, которая уже начиная с 50 переменных лидировала по количеству итераций наряду с модификацией First, а с 75 переменных начала показывать лучшую экономию времени на поиск решения. Для самой большой размерности тестовых задач (200×2000) удалось сократить количество итераций на 20%, а время на поиск решения — на 42%. Экономия рассчитана относительно базовой версии метода симплексных погружений (версия Max). Метод эллипсоидов в среднем затратил на решение задач такой размерности в 42 раза больше итераций и в 20 раз больше времени по сравнению с модификацией FirstAct.

Для задач выпуклой недифференцируемой оптимизации (8) наилучшие показатели по числу итераций оказались у модификации First, обеспечивая в среднем 20% экономии относительно базовой версии метода симплексных погружений. А вот лучшие результаты по времени на поиск решения показала модификация FirstAct. Максимальная экономия времени по сравнению с базовой версией составила 50%, что можно объяснить исключением из алгоритма вычисления невязок неактивных ограничений и экономией времени на осуществление одной итерации. Метод эллипсоидов для задач размерности 200×2000 затратил в 7 раз больше итераций и в 6 раз больше времени на поиск решения по сравнению с модификацией FirstAct.

Из рис. 1 и 2 видно, что сокращение времени на одну итерацию совпадает с весьма быстрым исключением неактивных ограничений. Это обеспечивает экономию времени на поиск решения. Отметим, что

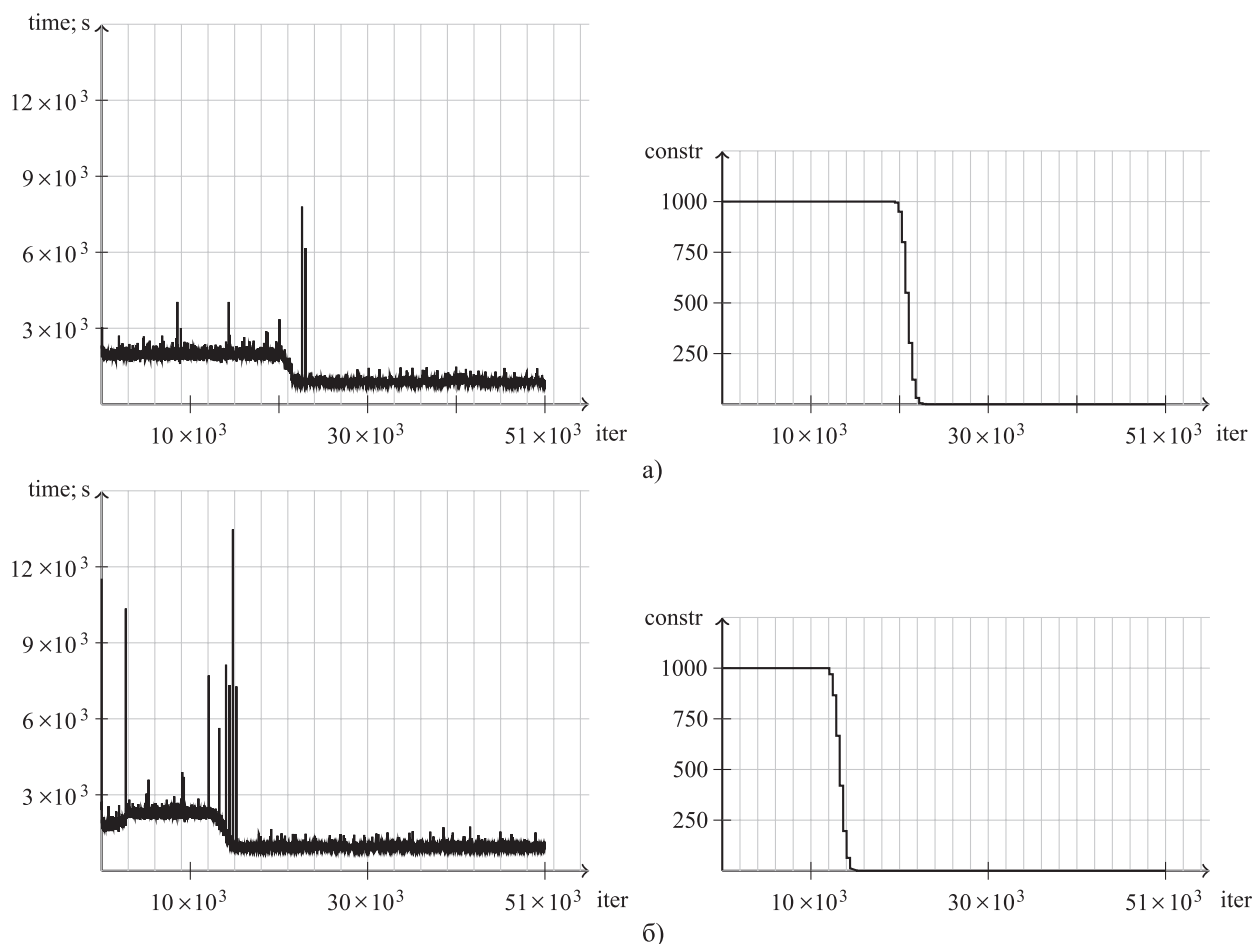


Рис. 2. Время на одну итерацию и динамика исключения неактивных ограничений для модификаций MaxAct (а) и FirstAct (б) при решении задач с недифференцируемыми ограничениями

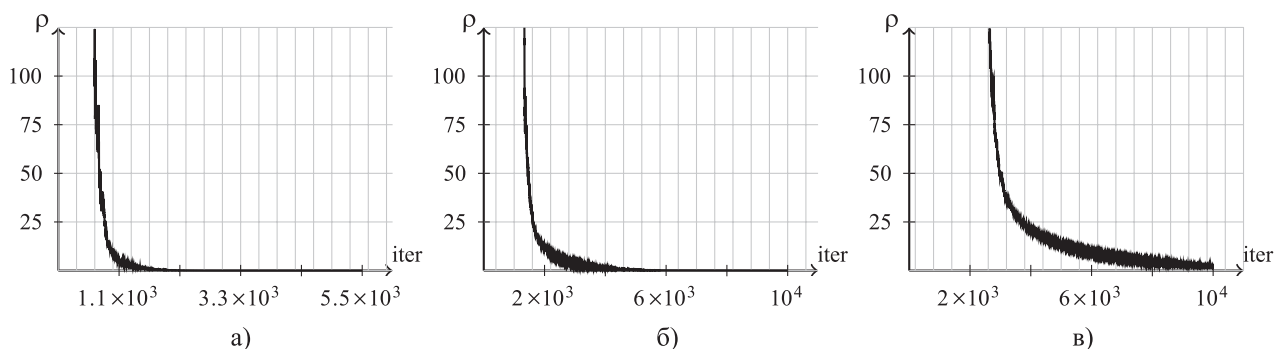


Рис. 3. Динамика сокращения длины максимального ребра (ρ) симплекса для модификаций FirstAct при $n = 25$ (а), $n = 50$ (б), $n = 100$ (в)

наличие нескольких пиков на графике времени связано лишь с внутренними вычислительными процессами среды Matlab. На рис. 3 можно выделить крутой наклон графика сокращения длины максимального ребра, что говорит о достаточно быстрой локализации решения задачи.

Отметим, что рассматриваемые модификации не претендуют на роль универсальных подходов для решения разнообразных задач оптимизации. Однако для определенных задач квадратичной и недифференцируемой оптимизации вида (7) и (8) исследованные модификации демонстрируют весьма заметную экономию времени и числа итераций.

8. Заключение. Метод симплексных погружений позволяет осуществлять гибкую настройку своего

алгоритма для решения достаточно обширного набора задач. В настоящей статье представлены две модификации алгоритма: в первой вместо вычисления максимальной невязки ограничений выполняется поиск первой положительной невязки, вторая модификация позволяет исключать неактивные ограничения из процесса работы алгоритма. Кроме того, алгоритм дополнительно снабжается критерием останова, который обеспечивает сильную сходимость метода. Использование представленных модификаций позволяет получить лучшие результаты расчетов в сравнении с базовой версией метода и методом эллипсоидов. При решении задач выпуклой квадратичной и недифференцируемой оптимизации с большим числом ограничений удалось добиться экономии как числа итераций, так и времени работы алгоритма.

Исследование выполнено при финансовой поддержке РФФИ (проект № 18-07-01432).

СПИСОК ЛИТЕРАТУРЫ

1. Александров И.А., Анциферов Е.Г., Булатов В.П. К центрированным методам отсечений в выпуклом программировании // Тезисы докл. конф. “Методы математического программирования и их программное обеспечение”. Свердловск: Институт математики и механики АН СССР, 1981. 10–11.
2. Булатов В.П., Шепотько И.О. Метод центров тяжести ортогональных симплексов для решения задачи выпуклого программирования // Методы оптимизации и их приложения. Иркутск: Институт систем энергетики АН СССР, 1982. 79–86.
3. Yamnitsky B., Levin L.A. An old linear programming algorithm runs in polynomial time // Proc. of the 23rd Annual IEEE Symposium on Foundations of Computer Science, Vol. 1. New York: IEEE Press, 1982. 327–338.
4. Юдин Д.Б., Немировский А.С. Информационная сложность и эффективные методы решения выпуклых экстремальных задач // Экономика и мат. методы. 1976. **12**, № 2. 357–369.
5. Шор Н.З. Метод отсечения с растяжением пространства для решения задач выпуклого программирования // Кибернетика. 1977. № 1. 94–95.
6. Александров И.А., Анциферов Е.Г., Булатов В.П. Методы центрированных сечений в выпуклом программировании. Препринт СЭИ АН СССР. Иркутск, 1983.
7. Анциферов Е.Г., Булатов В.П. Алгоритм симплексных погружений в выпуклом программировании // Ж. выч. мат. и мат. физ. 1987. **27**, № 3. 377–384.
8. Левин А.Ю. Об одном алгоритме минимизации выпуклых функций // Докл. АН СССР. 1965. **160**, № 6. 1244–1247.
9. Newman D.J. Location of the maximum on unimodal surfaces // Journal of the ACM. 1965. **12**, N 3. 395–398.
10. Поляк Б.Т. Введение в оптимизацию. М.: Наука, 1983.
11. Rademacher L.A. Approximating the centroid is hard // Proc. Symposium on Computational Geometry (SCG’07). New York: ACM Press, 2007. 302–305.
12. Nesterov Yu.E. Introductory lectures on convex programming: a basic course. Boston: Kluwer, 2004.
13. Goffin J.-L., Vial J.P. Convex nondifferentiable optimization: a survey focused on the analytic center cutting plane method // Optimization Methods and Software. 2002. **17**, N 5. 805–867.
14. Апекина Е.В., Хамисов О.В. Модифицированный метод симплексных погружений с одновременным введением нескольких секущих плоскостей // Изв. вузов. Серия Матем. 1997. № 12. 16–24.
15. Колосницын А.В. Применение модифицированного метода симплексных погружений для решения специального класса задач выпуклой недифференцируемой оптимизации // Известия Иркутского государственного университета. Серия Матем. 2015. **11**. 54–68.
16. Kolosnitsyn A. V. Modified simplex imbeddings method in convex non-differentiable optimization // CEUR Workshop Proc. 2016. **1623**. 226–233.
17. Колосницын А.В. Вычислительная эффективность метода симплексных погружений в задачах выпуклой недифференцируемой оптимизации // Ж. вычисл. матем. и матем. физ. 2018. **58**, № 2. 228–236.
18. Нурминский Е.А. Численные методы выпуклой оптимизации. М.: Наука, 1991.
19. Bagirov A., Karmitsa N., Makela M.M. Introduction to nonsmooth optimization. Theory, Practice and Software. Cham: Springer, 2014.

Поступила в редакцию
01.08.2019

A Modified Simplex Embedding Method for Solving Convex Optimization Problems with a Large Amount of Constraints

A. V. Kolosnitsyn¹

¹ Melentiev Energy Systems Institute, Siberian Branch of Russian Academy of Sciences; ulitsa Lermontova 130, Irkutsk, 664033, Russia; Junior Scientist, e-mail: ankolos25@mail.ru

Received August 1, 2019

Abstract: A simplex embedding method adapted for solving convex optimization problems with a large amount of constraints is considered. Two modifications of the method are proposed for better performance. First of them uses a more economical approach to the residual computation for constraints, which allows one to significantly reduce the execution time of the algorithm in the case of a large amount of constraints. One of the important peculiarities of the simplex embedding method is its ability to find inactive constraints. This property of the method is used as the basis for its second modification. The numerical results obtained when solving a number of quadratic and convex nondifferentiable optimization problems show the efficiency of the proposed modifications.

Keywords: simplex embedding method, convex nondifferentiable optimization, identification of inactive constraints.

References

1. I. A. Aleksandrov, E. G. Antsiferov, and V. P. Bulatov, "On the Centered Cutting Methods in Convex Programming," in *Proc. Conf. on Methods of Mathematical Programming and Software Sverdlovsk, USSR, April 14–17, 1981* (Inst. Mat. Mekh. Akad. Nauk SSSR, Sverdlovsk, 1981), pp. 10–11.
2. V. P. Bulatov and I. O. Shepot'ko, "The Gravity Center Method of Orthogonal Simplexes for Solving Convex Programming Problems," in *Optimization Methods and Their Applications* (Energy Systems Inst. Akad. Nauk SSSR, Irkutsk, 1982), pp. 79–86.
3. B. Yamnitsky and L. A. Levin, "An Old Linear Programming Algorithm Runs in Polynomial Time," in *Proc. 23rd Annual IEEE Symposium on Foundations of Computer Science, Chicago, USA, November 3–5, 1982* (IEEE Press, New York, 1982), Vol. 1, pp. 327–338.
4. D. B. Yudin and A. S. Nemirovskii, "Informational Complexity and Effective Methods for the Solution of Convex Extremal Problems," *Ekonomika Mat. Metody* **12** (2), 357–369 (1976).
5. N. Z. Shor, "A Cutting Method with Space Dilation for Solving Convex Programming Problems," *Kibernetika*, No. 1, 94–95 (1977).
6. I. A. Aleksandrov, E. G. Antsiferov, and V. P. Bulatov, *Centered Cutting Methods in Convex Programming*, Preprint (Energy Systems Inst. Akad. Nauk SSSR, Irkutsk, 1983).
7. E. G. Antsiferov and V. P. Bulatov, "An Algorithm of Simplex Imbeddings in Convex Programming," *Zh. Vychisl. Mat. Mat. Fiz.* **27** (3), 377–384 (1987) [*USSR Comput. Math. Math. Phys.* **27** (2), 36–41 (1987)].
8. A. Yu. Levin, "A Minimization Algorithm for Convex Functions," *Dokl. Akad. Nauk SSSR* **160** (6), 1244–1247 (1965).
9. D. J. Newman, "Location of the Maximum on Unimodal Surfaces," *J. ACM* **12** (3), 395–398 (1965).
10. B. T. Polyak, *Introduction to Optimization* (Nauka, Moscow, 1983; Optimization Software, New York, 1987).
11. L. A. Rademacher, "Approximating the Centroid is Hard," in *Proc. 23rd Annual Symp. on Computational Geometry, Gyeongju, South Korea, June 6–8, 2007* (ACM Press, New York, 2007), pp. 302–305.
12. Yu. E. Nesterov, *Introductory Lectures on Convex Programming: A Basic Course* (Kluwer, Boston, 2004).
13. J.-L. Goffin and J. P. Vial, "Convex Nondifferentiable Optimization: A Survey Focused on the Analytic Center Cutting Plane Method," *Optim. Methods Softw.* **17** (5), 805–867 (2002).
14. Y. V. Apekina and O. V. Khamisov, "A Modified Simplex Immersions Method with Simultaneous Introduction of Several Intersecting Planes," *Izv. Vyssh. Uchebn. Zaved., Mat., No. 12*, 16–24 (1997) [*Russ. Math.* **41** (12), 14–22 (1997)].
15. A. V. Kolosnitsyn, "Using of Modified Simplex Imbeddings Method for Solving Special Class of Convex Non-Differentiable Optimization Problems," *Vestn. Irkutsk Gos. Univ., Ser. Mat.* **11**, 54–68 (2015).
16. A. Kolosnitsyn, "Modified Simplex Imbeddings Method in Convex Non-differentiable Optimization," in *Proc. 9th Int. Conf. on Discrete Optimization and Operations Research, Vladivostok, Russia, September 19–23, 2016* CEUR Workshop Proc. **1623**, 226–233 (2016).
17. A. V. Kolosnitsyn, "Computational Efficiency of the Simplex Embedding Method in Convex Nondifferentiable Optimization," *Zh. Vychisl. Mat. Mat. Fiz.* **58** (2), 228–236 (2018) [*Comput. Math. Math. Phys.* **58** (2), 215–222 (2018)].
18. E. A. Nurminskii, *Numerical Methods of Convex Optimization* (Nauka, Moscow, 1991) [in Russian].
19. A. Bagirov, N. Karmitsa, and M. M. Makela, *Introduction to Nonsmooth Optimization. Theory, Practice and Software* (Springer, Cham, 2014).