

УДК 519.852

**МЕТОД ПОСЛЕДОВАТЕЛЬНОЙ АКТИВАЦИИ ОГРАНИЧЕНИЙ
В ЛИНЕЙНОМ ПРОГРАММИРОВАНИИ**

В. С. Колосов

г. Москва, Россия

Представлен алгоритм решения задачи линейного программирования посредством процедуры последовательной активации ограничений (включения в расчёт одного за другим) с удержанием состояния оптимальности на сгенерированной последовательности вложенных многогранников. Вследствие сжатия области допустимых решений при критерии оптимальности «max» целевая функция на каждом шаге убывает (движение к максимуму сверху), в противоположность росту в других методах (снизу). Компьютерные эксперименты демонстрируют преимущества программной реализации этого алгоритма перед опцией симплекс-метода программы `linprog` библиотеки MATLAB в скорости и полноте выводимой информации.

Ключевые слова: *линейное программирование, активация ограничения, MATLAB, компьютерный эксперимент.*

DOI 10.17223/20710410/41/11

**METHOD FOR SEQUENTIAL ACTIVATION OF LIMITATIONS
IN LINEAR PROGRAMMING**

V. S. Kolosov

*Moscow, Russia***E-mail:** vs.kolosov@yandex.ru

In this paper, we present an algorithm for solving a problem in linear programming by means of procedure of sequential activation of limitations (inclusions in calculation are made one after another) with retention of the optimality status in the generated sequence of nested polyhedrons. Due to compression of the admissible solutions area, the objective function decreases on each step at the “max” criterion of an optimality (the movement to a maximum “from above”) contrary to its growth in other methods (“from below”). Geometrically, the motion to the maximum starts from the trivially determined starting point and continues along the broken line outside of the polytope of admissible solutions. In the theoretical justification of the algorithm, the signs for the incompatibility of the system of conditions in the problem, for the uniqueness and nonuniqueness of the solution, and for its unboundedness are formulated and proved. Computer experiments demonstrate the advantages of the program implementation of this algorithm in speed and completeness of the output information over the simplex-method option of the MATLAB’s library `linprog` program.

Keywords: *linear programming, activation of limitation, MATLAB, computer experiment.*

Введение

Число существенно различных методов условной линейной оптимизации невелико. Еще в 1939 г. Л. В. Канторович сформулировал ряд задач линейного программирования и для их решения предложил метод разрешающих множителей [1]. В 1947 г. Дж. Данцигом был создан симплекс-метод, нашедший отражение в отечественной литературе в изложении автора в переводе [2]. Разработанные в то время методы последовательного улучшения плана, сокращения оценок и сокращения невязок классифицируются по типу решаемых ими задач — прямой, двойственной или сразу обеих [3]. В середине 1960-х годов И. И. Дикин в [4] предложил решение задачи линейного программирования методом внутренних точек. Поиск новых подходов к оптимизации дал эффективный, но вычислительно неэффективный метод эллипсоидов Л. Г. Хачияна [5], принципиально важным достоинством которого явилось выражение времени счёта полиномом от размерности задачи. Полиномиальность варианта метода внутренних точек Н. Кармаркара от 1984 г. в совокупности с его вычислительной эффективностью послужила причиной его патентования в США. Следствием изысканий в данном направлении было множество публикаций, в их числе [6–8]. Метод внутренних точек глубоко проработан и как эффективный оптимизатор задействован в ряде программных продуктов, в том числе в MATLAB.

К недавним разработкам относятся метод оператора-проектора [9] и скелетный алгоритм [10]. Суть первого заключается в проектировании произвольной точки на гиперплоскости ограничений вплоть до попадания в область допустимых решений и удержании последующих перемещений в заданных границах согласно критерию оптимальности. Скелетный алгоритм не требует обращения матриц и подменяет исходную задачу рядом вспомогательных задач пониженной размерности. В процессе тестовых расчётов установлено быстрое продвижение к оптимуму в начале вычислительного процесса и его резкое замедление по мере приближения к цели. Метод внутренних точек является приближённым и находит решение со сколь угодно малой заданной погрешностью, но полиномиален; ни один из известных точных алгоритмов таким важным свойством не обладает, за некоторыми исключениями. Так, при условии неотрицательности всех параметров задачи и совместности системы ограничений метод экспоненциальной аппроксимации [11] обещает находить точный оптимум в задаче с n переменными и $m \leq n$ равенствами всего за $(n - m)$ шагов. Алгоритм выявляет и исключает переменные с гарантированно нулевыми оптимальными значениями, и решение исходной задачи завершается решением системы уравнений с квадратной матрицей m -го порядка. Перспективы ослабления жёстких требований этого метода к исходным данным не просматриваются.

Перечисленные выше методы объединяют два общих свойства:

- многогранник допустимых решений подразумевается существующим в своем целостном виде, т. е. предполагается одновременная действенность всех ограничений системы условий;
- на каждом шаге поиск следующей точки производится с умыслом улучшения значения целевой функции в пределах области допустимых решений.

Недавние публикации свидетельствуют, что, несмотря на снижение интереса к проблематике линейного программирования, поиск новых путей в этой сфере продолжается. В [12] автор завершил разработку точного алгоритма решения задачи линейного программирования с использованием параметризации лимитов системы условий, запрограммировал его на языке Fortran-4 и успешно апробировал на ЕС-ЭВМ. С учётом

современных возможностей программных и аппаратных средств данный метод им основательно доработан и реализован средствами MATLAB.

Далее изложена доказательная база этого по сути геометрического метода решения общей задачи линейного программирования, отличительная черта которого состоит в следующем. Многогранник допустимых решений не существует изначально, а формируется путём последовательного включения в расчёт (активации с помощью параметризации лимитов) ограничений с удержанием оптимальности целевой функции на генерируемой последовательности вложенных многогранников. Начальным является гиперпараллелограмм, задаваемый ограничениями на переменные с возможно неограниченными лимитами, на котором оптимум определяется элементарно. Затем в его систему условий добавляется первое нетривиальное ограничение, быстрым перебором находится оптимум на построенном многограннике (вложенном в предыдущий) и т. д. Активируя неравенства, алгоритм последовательно усекает образующиеся многогранники до окончательной конфигурации, постепенно сужая область допустимых решений, вычисляя и фиксируя на каждой итерации алгебраические и геометрические характеристики текущего оптимума для последующих применений.

Метод оперирует с неравенствами и, исключая переменные через посредство равенств, понижает размерность задачи, но может работать напрямую и с равенствами, легко разбираясь с ситуациями вырожденности и маскировки равенств совокупностью неравенств. В результате сжатия области допустимых решений в задачах с критерием оптимальности «max» целевая функция на каждом шаге убывает или, по крайней мере, не возрастает (к максимуму сверху), в противоположность её традиционному увеличению в других методах (снизу). Геометрически происходит перемещение в оптимум вдоль некоторой ломаной прямой из тривиально определяемой точки, которая для многогранника допустимых решений может оказаться как граничной, так и внешней. В процессе активации ограничений выявляется несовместность системы условий с указанием виновных в этом ограничений, вычисляются координаты оптимальной вершины и смежных с ней в случае неединственности, а также определяются двойственные переменные как оценки чувствительности.

Разработанный алгоритм реализован автором в виде программы `mpao` в среде MATLAB и апробирован на обширном экспериментальном материале. Показана простота процедуры ввода исходных данных и запуска программы. Установлено преимущество по быстродействию и широте выводимой информации программы `mpao` по сравнению с опцией симплекс-метода программы `linprog` библиотеки MATLAB на примерах задач с размерностями до 1000×1000 .

1. Теоретическое обоснование алгоритма

В евклидовом пространстве E^n решается задача линейного программирования в самой общей постановке с разделением ограничений на «многофакторные», содержащие несколько переменных, и «координатные» (однофакторные), устанавливающие диапазоны возможных изменений собственно переменных при допущении отрицательности и неограниченности всех фигурирующих здесь величин включая сами переменные:

$$L(x) = cx \rightarrow \max, \quad c, x \in E^n,$$

$$\begin{cases} -\infty \leq \underline{b}^{(i)} \leq a_i x \leq \bar{b}^{(i)} \leq +\infty, a_i = (a_i^{(1)} \dots a_i^{(n)}), i=1, \dots, m \text{ (многофакторные)}, \\ -\infty \leq \underline{d}^{(j)} \leq x^{(j)} \leq \bar{d}^{(j)} \leq +\infty, j = 1, \dots, n \text{ (координатные)}. \end{cases} \quad (1)$$

Здесь сам вектор нумеруется нижним индексом, а его координаты — верхним в скобках. Многофакторные и координатные ограничения, высекая в E^n многогранник до-

пустимых решений (МДР) M , геометрически равнозначны. Первым отвечают гиперплоскости с задаваемой нормалью a_i ориентацией в пространстве, а вторым — ортогональные осям координат. Равенства реализуются уравниванием значений нижней и верхней границ разрешённых изменений и в отдельную группу не выделяются.

Равенствам отвечает размещение МДР в пересечении соответствующих гиперплоскостей, что открывает принципиальную возможность понижения размерности задачи за счёт фактического исключения (в отличие от симплекс-метода) части переменных. Это обстоятельство порождает две алгоритмические ветви. Следуя первой ветви, из равенства $a_i^{(1)}x^{(1)} + \dots + a_i^{(n)}x^{(n)} = b^{(i)}$ выражается переменная с ненулевым коэффициентом; пусть это $x^{(n)} = \alpha_i^{(1)}x^{(1)} + \dots + \alpha_i^{(n-1)}x^{(n-1)} + \alpha_i^{(n)}$. При её исключении из остальных многофакторных ограничений и целевой функции координатное ограничение $\underline{d}^{(n)} \leq x^{(n)} \leq \bar{d}^{(n)}$ преобразуется в многофакторное неравенство $\beta_i^{(n)} \leq \alpha_i^{(1)}x^{(1)} + \dots + \alpha_i^{(n-1)}x^{(n-1)} \leq \beta_i^{(n)}$ с меньшим числом переменных. В итоге размерность пространства решений понижается, но только за счет переменных при сохранении количества многофакторных ограничений. Исключением равенств из системы условий заканчивается подготовительный шаг алгоритма. С этого момента МДР считается полноразмерным ($\dim M = n$), т. е. задаётся только неравенствами. Особенности другой ветви алгоритма без исключения равенств из системы условий будут разъяснены позже.

Совокупности неравенств в E^n отвечает выпуклый многогранник M . Её любая подсистема ранга n представляет собой неограниченный многогранник $\dot{M}(x) \supseteq M$ с единственной вершиной в точке \dot{x} , являющейся решением соответствующей системы линейных равенств; пусть это $\{a_i x = b^{(i)} : i = 1, \dots, n\}$. В параметрическом представлении

$$\dot{M}(x) = \left\{ x = \dot{x} + \sum_{j=1}^n \beta_j v_j(\dot{x}), \beta_j \geq 0, j = 1, \dots, n \right\}, \quad (2)$$

где $v_j(\dot{x})$ — направляющие векторы рёбер $R_j(\dot{x}) = \{x = \dot{x} + tv_j(\dot{x}), t \geq 0\}$ многогранника M , выходящих из вершины \dot{x} и образованных пересечением $n - 1$ гиперплоскостей $G_i = \{x : a_i x = b^{(i)}\}$. Тогда $R_j(\dot{x}) = \bigcap_{i=1, i \neq j}^n G_i$ и $\{\dot{x}\} = R_j(\dot{x}) \cap G_j$. Алгебраически $R_j(\dot{x}) = \{x : a_i x = b^{(i)}, i = 1, \dots, n, i \neq j; a_j x \leq b^{(j)}\}$, где оно индексируется номером того единственного ограничения, которое в его образовании не участвует. Возьмём произвольную точку $x_j \in R_j(\dot{x})$ и построим вектор $v_j(\dot{x}) = x_j - \dot{x}$, который, будучи направляющим вектором, удовлетворяет однородной системе ранга $n - 1$:

$$\{a_i v_j(\dot{x}) = a_i x_j - a_i \dot{x} = b^{(i)} - b^{(i)} = 0, i = 1, \dots, j - 1, j + 1, \dots, n.$$

Этот вектор ортогонален нормальям $n - 1$ гиперплоскостей и может быть найден с помощью алгоритма Грама — Шмидта, который линейно независимую совокупность векторов $u_1, \dots, u_{n-1} \in E^n$ рекуррентными соотношениями ортогонального проектирования

$$u_i^0 = u_i, u_i^k = u_i^{k-1} - (u_i^{k-1} \tilde{u}_k^{k-1}) \tilde{u}_k^{k-1}, \tilde{u}_i^{i-1} = u_i^{i-1} / \|u_i^{i-1}\|, k < i < n \quad (3)$$

(k — «порядок» проекции) преобразует в ортонормированную систему $\{\tilde{u}_i^{i-1}, i = 1, \dots, n - 1 : \tilde{u}_i^{i-1} \tilde{u}_j^{j-1} = \delta_j^i\}$, δ_j^i — символ Кронекера. Очевидно, что при линейной независимости $\{u_0, u_1, \dots, u_{n-1}\}$ решением системы $\{u_i v = 0 : i = 1, \dots, n - 1\}$ будет вектор $v(u_0, u_1, \dots, u_{n-1}) = -u_0 + \sum_{i=1}^{n-1} (u_0 \tilde{u}_i^{i-1}) \tilde{u}_i^{i-1}$, где векторы \tilde{u}_i^{i-1} вычислены проектором Грама — Шмидта (3). Легко проверяются такие свойства построенного вектора, как

$u_0v < 0$ и $u_iv = 0$, $i \in \{1, \dots, n-1\}$. Поскольку вершина \dot{x} образована пересечением n гиперплоскостей, то, принимая поочередно в качестве u_0 их нормали, можно найти все направляющие векторы. В такой схеме для каждого вектора строится своя матрица ортогонализации. Поэтому более эффективно работает приём с одноразовым обращением матрицы координат нормалей и принятием в качестве направляющих векторов её столбцов, взятых с обратным знаком. С ростом размерности задачи метод ортогонализации быстро уступает обращению матрицы, которое представляет собой механизм вычисления направляющих векторов, выходящих из вершины многогранника.

Рассматриваемый метод опирается на общеизвестный факт достижения линейной функцией экстремума на выпуклом замкнутом многограннике по крайней мере в одной из его вершин. Метод реализует идею формирования из имеющихся ограничений начального многогранника M_0 , с гарантией содержащего оптимум задачи x^* , нахождения на нём оптимальной вершины x_0^* и построения последовательности оптимальных вершин $x_0^* \rightarrow x_1^* \rightarrow \dots \rightarrow x_m^* = x^*$ на промежуточных вложенных многогранниках $M_0 \supseteq M_1 \supseteq \dots \supseteq M_m = M$. В этой цепочке текущий M_k получен усечением предыдущего M_{k-1} следующим k -м ограничением.

В качестве M_0 разумно принять ортогональный параллелепипед, образованный координатными ограничениями. В случае неопределённости какой-либо границы её значение сообразно конкретной ситуации полагается равным $\pm\infty$. Если $M \neq \emptyset$, то, безусловно, $x^* \in M_0$. В силу структурной примитивности M_0 максимум на нём линейной функции L достигается в вершине $x_0^* = (x_0^{(1)*}, \dots, x_0^{(n)*})$, координаты которой

определяются простым правилом:
$$x_0^{(j)*} = \begin{cases} \bar{d}^{(j)}, & \text{если } c^{(j)} \geq 0, \\ \underline{d}^{(j)}, & \text{если } c^{(j)} < 0, \end{cases} \quad j = 1, \dots, n.$$

Определение 1. Полупространство (неравенство) $P = \{x : ax \leq b\}$ назовём значимым для точки $\tilde{x} \in E^n$, если $\tilde{x} \notin P$ ($a\tilde{x} > b$), и незначимым в противном случае ($a\tilde{x} \leq b$). Значимость и незначимость гиперплоскости (равенства) $G = \{x : ax = b\}$ определяются через неравенство $a\tilde{x} \neq b$ и равенство $a\tilde{x} = b$.

В результате усечения M_0 первым многофакторным двусторонним неравенством образуется многогранник $M_1 = M_0 \cap \underline{P}_1 \cap \overline{P}_1$, где $\underline{P}_1, \overline{P}_1$ — встречные полупространства с непустым пересечением: $\underline{P}_1 = \{x : \underline{b}^{(1)} \leq a_1x\}$, $\overline{P}_1 = \{x : a_1x \leq \overline{b}^{(1)}\}$. Поскольку $M_1 \subseteq M_0$, то $\max_{x \in M_1} L(x) \leq \max_{x \in M_0} L(x)$. Характер последующих действий определяется значимостью данного ограничения для x_0^* , т.е. расположением величины $a_1x_0^*$ относительно отрезка $[\underline{b}^{(1)}, \overline{b}^{(1)}]$. Неравенство $\underline{b}^{(1)} \leq a_1x_0^* \leq \overline{b}^{(1)}$ означает, что $x_0^* \in M_1$, и потому максимум L на M_1 достигается в той же самой вершине, т.е. $x_1^* = x_0^*$. Таким образом, данное ограничение для x_0^* незначимо и можно перейти к построению многогранника M_2 , усекая M_1 следующим ограничением. В противном случае проверяется $M_1 \neq \emptyset$, вычисляется x_1^* и т.д.

Рассмотрим k -й шаг алгоритма, к началу которого уже построен многогранник $M_{k-1} = M_0 \cap \underline{P}_1 \cap \overline{P}_1 \cap \dots \cap \underline{P}_{k-1} \cap \overline{P}_{k-1}$; пусть $M_{k-1} \neq \emptyset$. Если $x_{k-1}^* \in M_k$, то $x_k^* = x_{k-1}^*$ и текущий шаг на этом завершается. Остаётся разобраться с ситуацией $x_{k-1}^* \notin M_k$. Здесь двустороннее неравенство $\underline{b}^{(k)} \leq a_kx \leq \overline{b}^{(k)}$ для вершины x_{k-1}^* может быть не выполнено лишь с одной стороны и либо $a_kx_{k-1}^* < \underline{b}^{(k)}$, либо $a_kx_{k-1}^* > \overline{b}^{(k)}$. Принципиального различия между типами этих двух неравенств нет, так как один сводится к другому умножением неравенства на -1 .

В случае значимости ограничения $a_k x_{k-1}^* \notin [\underline{b}^{(k)}, \bar{b}^{(k)}]$ производятся основные расчёты. Пусть для определённости $x_{k-1}^* \notin \bar{P}_k$, граница которого $\bar{G}_k = \{x : a_k x = \bar{b}^{(k)}\}$. Параметризуем статическое полупространство $\bar{P}_k = \{x : a_k x \leq \bar{b}^{(k)}\}$, образовав динамическую структуру $\tilde{P}_k(\lambda) = \{x : a_k x \leq \lambda\}$, $\tilde{P}_k(\bar{b}^{(k)}) = \bar{P}_k$, $\tilde{G}_k(\lambda) = \{x : a_k x = \lambda\}$, $\tilde{G}_k(\bar{b}^{(k)}) = \bar{G}_k$.

Определение 2. Активацией значимого ограничения $a_k x \leq \bar{b}^{(k)}$ называется изменение параметра λ от $+\infty$ до $\bar{b}^{(k)}$, в результате чего движение гиперплоскости $\tilde{G}_k(\lambda)$ влечёт трансформацию $x_{k-1}^* \rightarrow x_k^*$.

Активация ограничения — динамическая процедура его приведения в активное состояние в случае значимости. Под активностью ограничения $P = \{x : ax \leq b\}$ в точке \dot{x} традиционно понимается её принадлежность граничной гиперплоскости, т.е. $\dot{x} \in G = \{x : ax = b\}$ или $a\dot{x} = b$. Вычисление x_0^* составляет содержание «нулевого» шага алгоритма, на котором активированы только все координатные ограничения.

Обозначим: $\{\dot{x}_j : j = 1, \dots, r\}$ — концевые точки рёбер $R_j(x_{k-1}^*)$ с направляющими векторами $v_j(x_{k-1}^*)$, являющиеся вершинами M_{k-1} , смежными с x_{k-1}^* ; $\tilde{P}_k(\lambda) = \{x : a_k x \leq \lambda\}$ — активируемое динамическое полупространство; $\lambda^* = a_k x_{k-1}^*$; $\dot{\lambda}_j = a_k \dot{x}_j$; $\tilde{M}_k(\lambda) = M_{k-1} \cap \tilde{P}_k(\lambda)$.

Сигналом к активации многофакторного ограничения \bar{P}_k является его значимость, и на k -м шаге этому отвечает решение задачи оптимизации $L(\tilde{x}_k^*(\lambda)) = \max_{\tilde{x} \in \tilde{M}_k(\lambda)} L(x)$,

при том что $\tilde{M}_k(\lambda) = M_{k-1} \cap \tilde{P}_k(\lambda)$, $+\infty = \lambda \rightarrow \bar{b}^{(k)}$. По построению $\tilde{M}_k(\lambda)$ при всех λ из указанного диапазона относительно M_{k-1} не расширяется и $\tilde{M}_k(\lambda) \subseteq M_{k-1}$, $\tilde{M}_k(+\infty) = M_{k-1}$, $\tilde{M}_k(\bar{b}^{(k)}) = M_k$, $\tilde{x}_k^*(\bar{b}^{(k)}) = x_k^*$. Полупространство $\tilde{P}_k(\lambda)$ при $\lambda \geq \lambda^* = a_k x_{k-1}^*$ для x_{k-1}^* является незначимым. Его граница $\tilde{G}_k(\lambda)$ при $\lambda = \lambda^*$ проходит через x_{k-1}^* , а при $\lambda < \lambda^*$ отвечает за трансформацию $x_{k-1}^* \rightarrow \tilde{x}_k^*(\lambda)$.

Теорема 1. Если $a_k v_j(x_{k-1}^*) \geq 0$ для всех $j \in \{1, \dots, r\}$, то для всех $\lambda < \lambda^*$ система условий задачи (1) несовместна ($M = \emptyset$).

Доказательство. Для любого $\lambda \leq \lambda^*$ по построению $\tilde{M}_k(\lambda) \subseteq M_{k-1}$, и в соответствии с представлением (2) для любого $\tilde{x} \in \tilde{M}_k(\lambda)$ с учётом $\beta_j \geq 0$ для всех $j \in \{1, \dots, r\}$ находим $a_k \tilde{x} = a_k x_{k-1}^* + \sum_{j=1}^r \beta_j a_k v_j(x_{k-1}^*) \geq a_k x_{k-1}^* = \lambda^*$. Тогда $\tilde{x} \notin \tilde{M}_k(\lambda)$ при $\lambda < \lambda^*$, т.е. $\tilde{M}_k(\lambda) = \emptyset$, а вследствие $M \subseteq \tilde{M}_k(\lambda)$ и $M = \emptyset$. ■

Замечание 1. Здесь сформирован механизм выявления противоречивости системы условий задачи в процессе активации ограничений.

Теорема 2. Пусть \dot{x} — вершина выпуклого многогранника $M \neq \emptyset$ с направляющими векторами всех выходящих из неё рёбер $v_1(\dot{x}), \dots, v_r(\dot{x})$. Максимум $L = cx$ на M достигается в \dot{x} , если $cv_j(\dot{x}) \leq 0$ для всех j ; эта вершина единственна в случае $cv_j(\dot{x}) < 0$ для всех j . Верно и обратное.

Доказательство. Достаточность следует из (2). Из включения $M \subseteq \dot{M}(\dot{x})$ получаем, что $cx = c\dot{x} + \sum_{j=1}^r \beta_j cv_j(\dot{x})$ для всех $x \in M$. Учитывая существование хотя бы одного $\beta_k > 0$ при $x \neq \dot{x}$ в случае $cv_k(\dot{x}) \leq 0$, получаем $cx \leq c\dot{x}$, а при строгом неравенстве $cv_k(\dot{x}) < 0$ таким же строгим будет неравенство $cx < c\dot{x}$.

Необходимость докажем от противного. Пусть $c\dot{x} > cx$ для всех $x \in M$, но существует $v_k(\dot{x})$, для которого $cv_k(\dot{x}) > 0$. Так как $x(t_0) = \dot{x} + t_0v_k(\dot{x}) \in M$ при некотором малом $t_0 > 0$, то $cx(t_0) = c\dot{x} + t_0cv_k(\dot{x}) > c\dot{x}$, что противоречит оптимальности \dot{x} . Далее, предположив единственность оптимума и существование $v_k(\dot{x})$, для которого $cv_k(\dot{x}) = 0$, получаем противоречие со сделанным предположением, поскольку $cx(t_0) = c\dot{x}$ при $x(t_0) \neq \dot{x}$. ■

Замечание 2. Так фиксируется неединственность решения.

Теорема 3. Если $M_k \neq \emptyset$, то для любого $\lambda \in \{\lambda^* - \varepsilon, \lambda^*\}$, где $\lambda^* = a_kx_{k-1}^*$, при достаточно малом $\varepsilon > 0$ имеет место $a_k\tilde{x}_k^*(\lambda) = \lambda$, чему геометрически соответствует $\tilde{x}_k^*(\lambda) \in \tilde{G}_k(\lambda)$.

Доказательство. Поскольку $\tilde{x}_k^*(\lambda) \in \tilde{M}_k(\lambda)$, то $a_k\tilde{x}_k^*(\lambda) \leq \lambda$. При допущении противного ($\tilde{x}_k^*(\lambda) \notin \tilde{G}_k(\lambda)$) это неравенство может быть только строгим ($a_k\tilde{x}_k^*(\lambda) < \lambda$). Из $\tilde{M}_k(\lambda) \subseteq M_{k-1}$ следует $\tilde{x}_k^*(\lambda) \in M_{k-1}$ и в силу выпуклости $M_{k-1} \supset [\tilde{x}_k^*(\lambda), x_{k-1}^*]$. По условию $\lambda < \lambda^*$ и, следовательно, концевые точки отрезка $\tilde{x}_k^*(\lambda)$ и x_{k-1}^* лежат строго по разные стороны гиперплоскости $\tilde{G}_k(\lambda)$, а значит, существует $x_\lambda \in (\tilde{x}_k^*(\lambda), x_{k-1}^*)$, что $x_\lambda \in \tilde{G}_k(\lambda)$. В параметрическом представлении $(\tilde{x}_k^*(\lambda), x_{k-1}^*) = \{x(t) = \tilde{x}_k^*(\lambda) + t(x_{k-1}^* - \tilde{x}_k^*(\lambda)) : t \in (0, 1)\}$; тогда существует $t_\lambda \in (0, 1)$, при котором $x_\lambda = x(t_\lambda) = \tilde{x}_k^*(\lambda) + t_\lambda(x_{k-1}^* - \tilde{x}_k^*(\lambda))$. Отсюда находим $x_{k-1}^* = \tilde{x}_k^*(\lambda) + t_0(x_\lambda - \tilde{x}_k^*(\lambda))$, $t_0 = 1/t_\lambda > 1$. Сделанное допущение $\tilde{x}_k^*(\lambda) \notin \tilde{G}_k(\lambda)$ влечёт строгое неравенство $c\tilde{x}_k^*(\lambda) > cx$ для всех $x \in \tilde{G}_k(\lambda)$, в том числе $c\tilde{x}_k^*(\lambda) > cx_\lambda$ или $c(x_\lambda - \tilde{x}_k^*(\lambda)) < 0$. Посчитав $cx_{k-1}^* = c(\tilde{x}_k^*(\lambda) + t_0(x_\lambda - \tilde{x}_k^*(\lambda))) = c\tilde{x}_k^*(\lambda) + t_0c(x_\lambda - \tilde{x}_k^*(\lambda)) < c\tilde{x}_k^*(\lambda)$, приходим к противоречию с оптимальностью x_{k-1}^* на $M_{k-1} \ni \tilde{x}_k^*(\lambda)$. ■

Замечание 3. Таким образом, если $M_k \neq \emptyset$, то либо сохраняется оптимальность предыдущей вершины и $x_k^* = x_{k-1}^*$, либо оптимальная вершина удовлетворяет активируемому ограничению как равенству с одной из его границ, что означает её принадлежность гиперплоскости $\underline{G}_k = \{x : a_kx = \underline{b}^{(k)}\}$ или $\overline{G}_k = \{x : a_kx = \overline{b}^{(k)}\}$.

Теорема 4. Если многогранник $M_k \neq \emptyset$ и активируемое ограничение $\overline{P}_k = \{x : a_kx \leq \overline{b}^{(k)}\}$ значимо для x_{k-1}^* , т. е. $J = \{j : a_kv_j(x_{k-1}^*) < 0\} \neq \emptyset$ и $a_kx_{k-1}^* > \overline{b}^{(k)}$, то в малой окрестности $U_\varepsilon(\lambda^*) = \{\lambda : 0 < \lambda^* - \lambda < \varepsilon\}$ вершина $\tilde{x}_k^*(\lambda)$, доставляющая целевой функции $L(x)$ максимум на $\tilde{M}_k(\lambda)$, может быть представлена в виде

$$\tilde{x}_k^*(\lambda) = x_{k-1}^* + \frac{\lambda - \lambda^*}{a_kv_{j_0}(x_{k-1}^*)}v_{j_0}(x_{k-1}^*), \quad \lambda \in U_\varepsilon(\lambda^*), \quad (4)$$

где индекс j_0 определяется решающим правилом

$$\frac{cv_{j_0}(x_{k-1}^*)}{a_kv_{j_0}(x_{k-1}^*)} = \min_{j \in J} \frac{cv_j(x_{k-1}^*)}{a_kv_j(x_{k-1}^*)}. \quad (5)$$

Доказательство. В малой окрестности x_{k-1}^* при $\lambda \leq \lambda^* = a_kx_{k-1}^*$ вершинами $\tilde{M}_k(\lambda)$ являются точки пересечения рёбер $R_j(x_{k-1}^*)$ многогранника M_{k-1} с гиперплоскостью $\tilde{G}_k(\lambda) = \{x : a_kx = \lambda\}$, и только они. Поэтому в согласии с теоремой 3 остаётся выбрать ту из них, в которой L достигает максимума. По построению $\tilde{x}_{k,j}(\lambda) = x_{k-1}^* + \frac{\lambda - \lambda^*}{a_kv_j(x_{k-1}^*)}v_j(x_{k-1}^*)$ является вершиной $\tilde{M}_k(\lambda)$, поскольку она лежит на ребре с направляющим вектором $v_j(x_{k-1}^*)$ и в то же время оказывается на гиперплос-

кости $\tilde{G}_k(\lambda)$, что нетрудно проверить прямым расчётом:

$$a_k \tilde{x}_{k,j}(\lambda) = a_k \left(x_{k-1}^* + (\lambda - \lambda^*) \frac{v_j(x_{k-1}^*)}{a_k v_j(x_{k-1}^*)} \right) = a_k x_{k-1}^* + \lambda - \lambda^* = \lambda.$$

Целевая функция в этих вершинах принимает следующие значения

$$c \tilde{x}_{k,j}(\lambda) = c x_{k-1}^* + (\lambda - \lambda^*) \frac{c v_j(x_{k-1}^*)}{a_k v_j(x_{k-1}^*)}, \quad j \in J.$$

Здесь второе слагаемое неположительно, так как, согласно теореме 2, $c v_j(x_{k-1}^*) \leq 0$ и по условию $\lambda - \lambda^* \leq 0$, $a_k v_j(x_{k-1}^*) < 0$. Поэтому для всех $\lambda \in U_\varepsilon(\lambda^*)$ максимум L достигается на ребре с направляющим вектором $v_{j_0}(x_{k-1}^*)$, для которого величина отношения $(c v_j(x_{k-1}^*)) / (a_k v_j(x_{k-1}^*))$ минимальна, т. е. $\tilde{x}_{k,j_0}(\lambda) = \tilde{x}_k^*(\lambda)$. ■

Замечание 4. Полученный результат позволяет придать двойственным оценкам следующий смысл. Найдём двойственную оценку активируемого ограничения в вершине $\tilde{x}_k^*(\lambda)$ прямым расчётом с учётом её аналитического представления (4):

$$\frac{dL(\tilde{x}_k^*(\lambda))}{d\lambda} = \frac{d}{d\lambda} c \left(x_{k-1}^* + \frac{\lambda - \lambda^*}{a_k v_{j_0}(x_{k-1}^*)} v_{j_0}(x_{k-1}^*) \right) = \frac{c v_{j_0}(x_{k-1}^*)}{a_k v_{j_0}(x_{k-1}^*)}. \quad (6)$$

Таким образом, двойственная оценка активируемого ограничения по сути является решающим правилом (4), и наоборот, в чём и проявляется связь прямой и двойственной задач. Теперь формулы (4), (5) можно интерпретировать так: на текущей стадии активации значимого ограничения оптимальными будут точки ребра, вычисляющего двойственную оценку этого ограничения [13]. Правило (5) позволяет после нахождения оптимума вычислять двойственные переменные посредством последовательной псевдоактивации (назначения в качестве активируемого) каждого базового ограничения.

Далее для удобства принята сквозная нумерация ограничений: сначала координатные $(1, \dots, n)$, а затем многофакторные $(n + 1, \dots, n + m)$.

Определение 3. Назовём базой вершины \dot{x} совокупность номеров всех активных ограничений $B(\dot{x}) = \{i_k : a_{i_k} \dot{x} = b_{i_k}, k = 1, \dots, r \geq n\}$, т. е. тех, которым она удовлетворяет как равенствам (принадлежит гиперплоскостям), а сами ограничения (гиперплоскости) — базовыми. Величина $q = r - n + 1$ называется кратностью вершины. Однократная вершина — простая.

Активация ограничения в случае его значимости наряду с изменением оптимума влечёт соответствующую трансформацию базы. Локальный результат теоремы 4 допускает следующее расширение.

Теорема 5. База вершины $\tilde{x}_k^*(\lambda) = \tilde{x}_{k,j_0}(\lambda)$ сохраняется в пределах интервала $(\dot{\lambda}_{j_0}, \lambda^*)$, левая граница которого отвечает концевой точке \dot{x}_{j_0} ребра $R_{j_0}(x_{k-1}^*)$, т. е. $\dot{\lambda}_{j_0} = a_k \dot{x}_{j_0}$. Вершина \dot{x}_{j_0} , как смежная с x_{k-1}^* , вычисляется из расчёта максимального сдвига вдоль вектора $v_{j_0}(x_{k-1}^*)$, допускаемого остальными ограничениями многогранника M_{k-1} , помимо базовых, а также противоположной гранью j_0 -го базового условия.

Доказательство. Очевидно, что кандидатами на пересечение с ребром $R_{j_0}(x_{k-1}^*)$ являются все активированные к настоящему моменту небазовые ограничения. В дополнение к этому необходимо учитывать также гарантированное пересечение данного ребра с противоположной гранью базового ограничения с номером j_0 , поскольку из всех базовых гиперплоскостей направляющий вектор $v_{j_0}(x_{k-1}^*)$ не ортогонален только

ей. Сообразно этим обстоятельствам параметры принятого для кандидата обозначения $\tilde{a}_s x = \tilde{b}^{(s)}$ конкретизируются по следующей схеме:

$$\left\{ \begin{array}{l} \text{для координатных ограничений} \\ v_{j_0}^{(s)}(x_{k-1}^*) > 0 \Rightarrow \tilde{a}_s = e_s, \quad \tilde{b}^{(s)} = \bar{d}^{(s)}, \\ v_{j_0}^{(s)}(x_{k-1}^*) < 0 \Rightarrow \tilde{a}_s = -e_s, \quad \tilde{b}^{(s)} = -\underline{d}^{(s)}, \\ s = 1, \dots, n, \quad s \notin B(x_{k-1}^*) \setminus \{j_0\}, \end{array} \right. \quad \left| \quad \begin{array}{l} \text{для многофакторных ограничений} \\ a_r v_{j_0}(x_{k-1}^*) > 0 \Rightarrow \tilde{a}_s = a_r, \quad \tilde{b}^{(s)} = \bar{b}^{(r)}, \\ a_r v_{j_0}(x_{k-1}^*) < 0 \Rightarrow \tilde{a}_s = -a_r, \quad \tilde{b}^{(s)} = -\underline{b}^{(r)}, \\ r = 1, \dots, k, \quad s = n + r \notin B(x_{k-1}^*) \setminus \{n + j_0\}. \end{array} \right.$$

Здесь e_s — единичный базисный вектор: $e_s^{(j)} = \delta_s^j$. Первая группа включает все небазовые координатные ограничения, а вторая — активированные к настоящему моменту небазовые многофакторные ограничения. Базовое ограничение с номером j_0 учитывается в той группе, к которой оно принадлежит.

Величина сдвига вдоль ребра $R_{j_0}(x_{k-1}^*)$ до пересечения с гиперплоскостью $\tilde{G}_s = \{x : \tilde{a}_s x = \tilde{b}^{(s)}\}$ в точке $x_{j_0,s}$ определяется соотношениями

$$x_{j_0,s} = x_{k-1}^* + t_{j_0,s} v_{j_0}(x_{k-1}^*), \quad \tilde{a}_s (x_{k-1}^* + t_{j_0,s} v_{j_0}(x_{k-1}^*)) = \tilde{b}^{(s)}, \quad t_{j_0,s} = \frac{\tilde{b}^{(s)} - \tilde{a}_s x_{k-1}^*}{\tilde{a}_s v_{j_0}(x_{k-1}^*)}.$$

Концевой точке ребра \dot{x}_{j_0} , сопряжённой с x_{k-1}^* по ребру $R_{j_0}(x_{k-1}^*)$, отвечает максимальный возможный сдвиг в пределах многогранника M_{k-1} :

$$t_{j_0,s_0} = \min_{s \in S} t_{j_0,s}, \quad S = \{1, \dots, n + k - 1\} \setminus B(x_{k-1}^*), \quad \dot{x}_{j_0} = x_{k-1}^* + t_{j_0,s_0} v_{j_0}(x_{k-1}^*)$$

и $\dot{\lambda}_{j_0} = a_k \dot{x}_{j_0}$ — левый конец интервала стабильности базы $B(\tilde{x}_k^*(\lambda))$. ■

Таким образом, $\tilde{x}_k^*(\lambda)$ — оптимум L на $\tilde{M}_k(\lambda)$ для всех $\lambda \in [\dot{\lambda}_{j_0}, \lambda^*]$ и характер последующих действий определяется одной из двух ситуаций:

- 1) Если $\bar{b}^{(k)} \geq \dot{\lambda}_{j_0}$, то $x_k^* = \tilde{x}_k^*(\bar{b}^{(k)}) = x_{k-1}^* + \frac{\bar{b}^{(k)} - \lambda^*}{a_k v_{j_0}(x_{k-1}^*)} v_{j_0}(x_{k-1}^*)$. Преобразование $x_{k-1}^* \rightarrow x_k^*$ влечёт трансформацию базы $B(x_{k-1}^*) \rightarrow B(x_k^*) = (B(x_{k-1}^*) \setminus \{j_0\}) \cup \{k\}$, состоящую в замещении элемента j_0 на номер k , чем и завершается текущий шаг алгоритма.
- 2) В случае $\bar{b}^{(k)} < \dot{\lambda}_{j_0}$ активация ограничения продолжается относительно \dot{x}_{j_0} с базой $B(\dot{x}_{j_0}) = (B(x_{k-1}^*) \setminus \{j_0\}) \cup \{s_0\}$ вплоть до занятия гиперплоскостью $\tilde{G}_k(\lambda)$ её геометрически окончательного положения.

На этом заканчивается построение вычислительных конструкций метода последовательной активации ограничений (МПАО), название которого происходит от характера производимых действий. В процессе активации ограничений теорема 1 выявляет несовместность системы условий задачи. Формулы теорем 4 и 5 вычисляют координаты оптимума и двойственные переменные. Неограниченность некоторых координат x^* означает неограниченность решения задачи по соответствующим переменным. В случае неединственности оптимума в интересах послеоптимизационного анализа теорема 2 позволяет перечислить смежные вершины в виде множества

$$\dot{X}^*(M) = \{x_m^*, \quad \dot{x}_{m,j} = x_m^* + t_j v_j \in M : cv_j(x_m^*) = 0, \quad j = 1, \dots, n^*\}.$$

Каждая итерация заканчивается замещением в базе одного элемента другим, чему соответствует замена только одной строки матрицы нормалей активных ограничений.

Поэтому при нахождении направляющих векторов посредством инвертирования матриц использование факторизации существенно повышает быстродействие алгоритма.

Следуя первой ветви алгоритма с исключением переменных через равенства, по окончании решения задачи оптимальные значения исключенных переменных восстанавливаются по найденным на предварительном шаге их выражениям через действующие переменные.

Отличие ветви алгоритма без исключения равенств из системы условий состоит в том, что активация многофакторных ограничений производится в следующем порядке: сначала равенства как безусловно активные, затем неравенства. Порядок обрабатываемых матриц равен размерности пространства решений и, естественно, больше по сравнению с первой ветвью алгоритма. Зато объём других вычислений несколько уменьшается, поскольку по мере поступления равенств в базу промежуточной оптимальной вершины при активации ограничения в расчёт принимаются только те рёбра, которые удерживают последующие перемещения в гиперплоскостях равенств.

Подозрение на возможность заикливания вычислений в МПАО возникает лишь относительно ситуации прохождения кратных вершин (определение 3), которая в результате накопления ошибок округления может как разрешиться сама собой, так и ложно возникнуть. В [3] в борьбе с заикливанием предлагается делать ставку на регулярность метода. Следуя этой рекомендации, в случае неединственности оптимального ребра неоднозначность выбора в (5) разрешается по принципу наименьшего номера.

Процедура активации ограничения обладает регулярностью в том смысле, что в соответствии с теоремой 4 движение допускается вдоль направляющего вектора, образующего с нормальным вектором активируемого ограничения исключительно тупой угол. Это обстоятельство обеспечивает безаварийное прохождение вершин кратности больше 1, хотя МПАО настроен на простые вершины. Убедиться в этом помогает умозрительный пример 2-кратной вершины с применением к нему приёма «расклеивания» [14], суть которого состоит в представлении такой вершины в виде двух близко расположенных простых вершин за счёт малого изменения свободного члена со смещением «лишней» гиперплоскости вдоль её нормали во избежание сужения множества допустимых решений. В результате алгоритм если все-таки и пройдёт по фиктивному ребру, то всё равно выйдет на ненулевое оптимальное ребро. Однако тот же результат достижим при виртуальном расклеивании кратной вершины. Величина смещения вдоль фиктивного ребра равна 0, но, несмотря на сохранение координат промежуточной оптимальной вершины, произойдёт преобразование её базы, что означает фактический переход в структурно другую вершину. Повторение элементного состава базы означало бы возврат в ранее пройденную вершину, что возможно лишь при движении под запрещённым острым углом в обратном направлении. Это свидетельствует о безразличии МПАО к кратности вершин и невозможности заикливания по этой причине. МПАО нечувствителен к вырожденности вершин в традиционной терминологии, легко распознаёт и разрешает случаи дублирования равенств и их маскировки совокупностью неравенств, что подтверждается экспериментальными расчётами.

В отличие от метода внутренней точки (МВТ), являющегося приближённым, МПАО — точный, подобно симплекс-методу (СМ). Однако теоретическая точность метода ещё не признак его безусловного превосходства, так как злую шутку может сыграть накопленная ошибка округления. В то же время приближённый метод может оказаться и точнее, и быстрее, что подтверждается практикой применения МВТ. Вычислительная эффективность МПАО определяется количеством шагов при активации

многофакторных ограничений. Её примитивная оценка сверху на базе подсчёта числа проходимых алгоритмом вершин промежуточных многогранников является ожидаемо экспоненциальной, и возможность её снижения вызывает законное сомнение.

2. Анализ результатов вычислительных экспериментов

МПАО реализован в MATLAB 7 R2011b программой `mpao`, которая решает задачу линейного программирования с двусторонними ограничениями, допуская неопределённость границ и отрицательность переменных. Условия-равенства оформляются равенством верхней и нижней границ допустимых изменений, в отдельную группу не выделяются и могут перемежаться неравенствами. Исходные данные передаются в программу массивом определённой структуры или текстовым файлом `trafaret` с вводом с клавиатуры только его номера при запуске программы. Данные из трафаретного файла считываются программой как строчные, что обеспечивает отсутствие жёстких требований к формату их расположения в поле ввода и визуализацию исходных данных сообразно вкусам пользователя. В случаях несовместности, неограниченности, неединственности решения выдаются соответствующие сообщения. всплывающие меню с активными кнопками обеспечивают управление функциональными возможностями программы в части её запуска, визуализации исходных данных и полученных результатов, вычисления альтернативных решений в случае неединственности.

Рассмотрим результаты применения программ `mpao` и `linprog` библиотеки MATLAB в опциях крупномасштабного алгоритма LSA (MBT) и среднемасштабного алгоритма MSA (CM) на простом примере

$$L = -x^{(1)} + x^{(2)} \rightarrow \min, \quad \begin{cases} -x^{(1)} + x^{(2)} \geq 1, \\ -2x^{(1)} + x^{(2)} \leq 2, \\ 3x^{(1)} + x^{(2)} \leq 3 \end{cases}$$

с допустимыми решениями в треугольнике с вершинами $A(0,5, 1,5)$, $B(0,2, 2,4)$, $C(-1, 0)$ и оптимумом на ребре AC .

В программе `linprog` не предусмотрено выявление неединственности решения, хотя уже сам факт возникновения такой ситуации в практических задачах может быть весьма весомым. Опция LSA вышла на внутреннюю точку $x^* = (-0,2743, 0,7257)$ ребра AC , а MSA — на вершину $x^* = (-1, 0)$ и там остановилась.

Программа `mpao` нашла обе оптимальные вершины $x_1^* = (0,5, 1,5)$ и $x_2^* = (-1, 0)$, т. е. всё ребро AC . Вычисленные двойственные переменные в обоих решениях имеют одинаковые соответствующие значения $(1, 0, 0)$, что объясняется следующим обстоятельством. В первом решении активными являются первое и третье ограничения, а во втором — первое и второе. Превратим активные ограничения в свободные и объединим их правые части $\lambda^{(k)}$ в вектор $\lambda = (\lambda^{(1)}, \lambda^{(2)}, \lambda^{(3)})$. При условии сохранения базы оптимальной вершины (теорема 5) прямыми выкладками получаем $L(\tilde{x}_1^*(\lambda)) = L(\tilde{x}_2^*(\lambda)) = \lambda^{(1)}$. Обозначив оптимум двойственных переменных $y^* = (y^{(1)*}, y^{(2)*}, y^{(3)*})$, следуя формуле (6) и [14], находим

$$y^{(1)*} = \frac{\partial L(\tilde{x}_1^*(\lambda))}{\partial \lambda^{(1)}} = \frac{\partial L(\tilde{x}_2^*(\lambda))}{\partial \lambda^{(1)}} = 1, \quad y^{(k)*} = \frac{\partial L(\tilde{x}_1^*(\lambda))}{\partial \lambda^{(k)}} = \frac{\partial L(\tilde{x}_2^*(\lambda))}{\partial \lambda^{(k)}} = 0, \quad k = 2, 3,$$

т. е. у двойственной задачи оптимум $y^* = (1, 0, 0)$ единственен, что и подтверждается её непосредственным решением.

Замена первого ограничения на $-x^{(1)} + x^{(2)} \geq 3$ превращает систему условий в несовместную. Реакция `linprog`: решение не найдено. Диагностика `mpao`: совместность нарушает ограничение 3 (правостороннее). Другой порядок активации ограничений может изменить номер нарушителя.

Изъятие первого неравенства влечёт неограниченность решения. Сообщение `linprog`: решение не ограничено, $x^* = (2,46 \cdot 10^{31}, -7,39 \cdot 10^{31})$, $L = -9,85 \cdot 10^{31}$. Ответ `mpao`: решение неограничено по переменным $x^{(1)}$ и $x^{(2)}$, $L = -\infty$.

Масштабная апробация программы `mpao` производилась на базе задач, созданных с помощью специальной модификации `mpao`, для которой исходными данными являются количество переменных и ограничений, а также наличие или отсутствие требования совместности системы условий. Используя генератор случайных чисел, эта программа выдаёт коэффициенты целевой функции и ограничений, а также лимиты ограничений и переменных. В таблице приведено время счёта для задач с размерностями в диапазоне $10 \times 10 \div 1000 \times 1000$ для программ `mpao` и `linprog` на одноядерном ПК с частотой процессора 1,73 ГГц и ОЗУ 1 Гб. Столь маломощный компьютер был выбран умышленно для обеспечения осязаемости времени счёта на задачах малой размерности.

Размер $m \times n$	mpao			LSA			MSA		
	мин.	сред.	макс.	мин.	сред.	макс.	мин.	сред.	макс.
10 × 10	0,00	0,00	0,08	0,03	0,04	0,67	0,01	0,025	0,562
20 × 20	0,00	0,01	0,05	0,03	0,05	0,09	0,06	0,074	0,219
30 × 30	0,03	0,04	0,12	0,11	0,13	0,78	0,16	0,185	0,719
40 × 40	0,09	0,11	0,12	0,12	0,13	0,17	0,34	0,369	0,454
50 × 50	0,13	0,14	0,2	0,12	0,14	0,20	0,73	0,758	0,875
80 × 80	0,73	0,76	0,89	0,40	0,44	0,56	2,89	3,067	3,485
100 × 100	1,44	1,47	1,52	0,76	0,79	0,86	7,03	7,333	7,593
200 × 200	16,4	16,6	16,9	6,09	6,21	6,31	116	116,8	118,3
300 × 300	60,2	60,6	60,9	6,75	6,98	7,61	310	310,5	311,1
300 × 500	327	328	329	21,8	21,9	22,4	617	618,3	622,4
500 × 300	98,9	101	101	17,1	17,3	17,8	874	874,7	875,3
500 × 500	437	442	446	36,4	36,7	37,0	2346	2351	2356
500 × 1000	3393	3410	3433	447	471	516	5902	5904	5906
1000 × 500	805	851	942	45,5	46,5	47,7	10895	11113	11505
1000 × 1000	5854	5862	5869	328	333	345	46091	48961	51223

Примечание. Указаны минимальное (мин.), максимальное (макс.) и усреднённое (сред.) время счёта в секундах по 1000 прогонов для каждой из малоразмерных задач с постепенным снижением числа прогонов до трёх по мере роста размерности. Усреднённое время рассчитано как среднеарифметическое по числу прогонов.

В экспериментальных расчётах последовательность активации ограничений заметно влияла на количество итераций, и это обстоятельство логично использовать с максимальной выгодой. Пока что для повышения эффективности алгоритма порядок активации устанавливается исходя из просто реализуемых и по возможности малообременительных правил. Одним из них является выбор кандидата на активацию с максимальной значимостью. Под мерой значимости может пониматься, например, возможность наибольшего снижения максимума целевой функции при активации ограничения, что подразумевает выявление кандидата на активацию на каждом шаге алгоритма. Активацию ограничений можно производить также в порядке убывания угла между нормальными целевой функции и ограничений. Детали такой стратегии устанавливаются один раз изначально, основанием для этого служит следующее соображение. В E^2 одна из нормалей базовой гиперплоскости оптимальной вершины составляет с

нормалью целевой функции минимальный угол. Хотя уже в E^3 это вовсе не обязательно, результаты экспериментальных расчётов показывают, что такая последовательность активации, не будучи наилучшей, сокращает количество итераций на десятки процентов и потому вполне оправдана. В случае нахождения оптимальной стратегии активации ограничений традиционным неопровержимым аргументом против принципиальной возможности понижения экспоненциальной оценки числа итераций МПАО будет контрпример.

Анализ полученных результатов свидетельствует, что, в противоположность MSA, быстродействие *mpao* и LSA в большей мере зависит от числа переменных, нежели от количества ограничений. Это обстоятельство обеспечивает первой ветви алгоритма (с исключением части переменных через равенства) вычислительное преимущество по сравнению со второй ветвью (решение задачи непосредственно с равенствами). На малых размерностях до 50×50 уверенно лидирует *mpao*, а далее при стабильном удержании ею второго места быстро растущее превосходство переходит к опции LSA, которая демонстрирует слабую зависимость времени счёта от размерности задач. На размерности 1000×1000 MSA многократно уступает LSA и *mpao*. На ПК с двухъядерным процессором с частотой 2 ГГц и ОЗУ 2 Гб вычисления ускорились примерно на 10% с сохранением соотношения величин этого показателя для всех трёх программ.

В условиях наличия высокопроизводительных компьютеров быстродействие далеко не всегда является основным критерием эффективности алгоритма и соответствующего программного продукта, на первое место могут выдвигаться их функциональные и сервисные возможности.

Заключение

Сопоставление СМ и МПАО приводит к следующим выводам. Оба метода — точные и конечные. Решая задачу на максимум, СМ на каждом шаге пытается увеличить текущее значение целевой функции, продвигаясь вдоль одной из координатных осей в границах МДР, т.е. максимум достигается снизу. Для превращения неравенств в равенства и нахождения начальной точки вводятся искусственные переменные, что формально повышает размерность задачи.

МПАО, напротив, предпочитает сокращать количество переменных путём их исключения через посредство равенств, поскольку его вычислительная эффективность обратно пропорциональна числу неизвестных. Последовательно активируя ограничения, алгоритм усекает начальный гиперпараллелепипед до заданного МДР, сужая область допустимых решений и сохраняя состояние оптимальности. В результате этого максимальное значение целевой функции на каждом шаге уменьшается или, по крайней мере, не увеличивается и приближение к максимуму выглядит как спуск сверху. Начальная точка маршрута определяется тривиально, и последующее движение к оптимальной вершине происходит по траектории, представляющей собой ломаную прямую, вершинами которой являются оптимальные вершины промежуточных многогранников. Эта траектория пролегает вне МДР; если в ходе активации ограничений не выявлена их несовместность, то его первой же захваченной алгоритмом точкой как раз и будет искомая оптимальная вершина. В связи с этим напрашиваются такие параллели: симплекс метод — метод внутренней точки, метод последовательной активации ограничений — метод внешней точки. В отличие от геометрического метода, преодолевающего задачи максимум с тремя переменными с визуализацией МДР, МПАО, базируясь на геометрических представлениях, решает задачи любой размерности. МПАО активно использует органичную связь прямой и двойственной задач. Это проявляется в том,

что решающее правило выбора оптимального ребра (5) вычисляет двойственную оценку ограничения на текущей стадии его активации. После нахождения оптимума задачи посредством псевдоактивации каждого из базовых ограничений решающее правило находит оптимальное решение двойственной задачи. Последнее действие вычислительно малообременительно, поскольку к его началу известны направляющие векторы рёбер многогранника допустимых решений, выходящих из оптимальной вершины, и активные в ней ограничения. Таким образом, в МПАО двойственность играет решающую роль на всех стадиях вычислительного процесса.

Анализ результатов экспериментальных расчётов свидетельствует о теоретической и практической состоятельности МПАО и превосходстве в быстродействии программы *mpao* над опцией симплекс-метода программы *linprog*, растущим с размерностью задач. Потенциал программы позволяет: констатировать неограниченность целевой функции и указывать ответственные за это переменные; выявлять ограничения, вызывающие несовместность системы условий; в отличие от *linprog*, в случае неединственности описывать структуру множества решений в виде смежных оптимальных вершин; определять чувствительность оптимума относительно активных ограничений посредством двойственных оценок; подобно прожектору, высвечивать структуру МДР в окрестности оптимума, что в случае неполной достоверности исходных данных имеет практическое значение. Резервы повышения быстродействия МПАО ещё не исчерпаны, но уже сейчас всего лишь с экспериментальной программой *mpao* он уверенно занимает среднее положение между СМ и МВТ и по ряду параметров более информативен. Стремительное продвижение к оптимуму МВТ вне конкуренции и точным методом пока недостижимо.

ЛИТЕРАТУРА

1. Канторович Л. В. Математические методы в организации и планировании производства. Л.: Изв. Ленингр. гос. ун-та, 1939. 67 с.
2. Данциг Дж. Линейное программирование, его применение и обобщения. М.: Прогресс, 1966. 600 с.
3. Юдин Д. Б., Гольштейн Е. Г. Линейное программирование: теория, методы и приложения. М.: Наука, 1969. 424 с.
4. Дикин И. И. Итеративное решение задач линейного и квадратичного программирования // Докл. АН СССР. 1967. Т. 174. С. 747–748.
5. Хачиян Л. Г. Полиномиальный алгоритм в линейном программировании // Докл. АН СССР. 1979. Т. 244. С. 1093–1096.
6. Зоркальцев В. И. Двойственные алгоритмы внутренних точек // Изв. вузов. Математика. 2011. № 4. С. 33–53.
7. Зоркальцев В. И., Медвежонков Д. С. Численные эксперименты с вариантами алгоритмов внутренних точек на нелинейных задачах потокораспределения // Управление большими системами. Ин-т систем энергетики им. Л. А. Мелентьева СО РАН, 2013. Вып. 46. С. 68–87.
8. Медвежонков Д. С. Экспериментальные исследования алгоритмов внутренних точек на нелинейных задачах потокораспределения // Вестник Бурятского гос. ун-та. 2013. № 9. С. 12–16.
9. Вылегжанин О. Н., Шкатова Г. И. Решение задачи линейного программирования с использованием оператора-проектора // Изв. Томского политехн. ун-та. 2009. Т. 314. № 5. С. 37–40.

10. *Бакшиян Б. Ц., Горьянов А. В.* Скелетный алгоритм решения задачи линейного программирования и его применение для решения задач оценивания // Вестник МАИ. 2008. Т. 15. № 2. С. 5–16.
11. *Гордуновский В. М.* Метод экспоненциальной аппроксимации для линейного программирования. М.: Эдитус, 2013. 32 с.
12. *Колосов В. С.* Конечный параметрический метод решения задачи линейного программирования // Науч. труды МЛТИ. 1978. Вып. 103. С. 197–198.
13. *Колосов В. С.* Роль двойственных оценок в параметрическом методе решения задачи линейного программирования // Науч. труды МЛТИ. 1986. Вып. 183. С. 51–54.
14. *Карманов В. Г.* Математическое программирование. М.: ФИЗМАТЛИТ, 2004. 264 с.

REFERENCES

1. *Kantorovich L. V.* Matematicheskie metody v organizacii i planirovanii proizvodstva [Mathematical Methods in the Organization and Planning of Production]. Leningrad, News of Leningrad State University, 1939. 67 p. (in Russian)
2. *Dantzig J.* Linejnoe programmirovaniye, ego primeneniye i obobshcheniya [Linear Programming, its Application and Generalizations]. Moscow, Progress Publ., 1966. 600 p. (in Russian)
3. *Yudin D. B. and Golstein E. G.* Linejnoe programmirovaniye: teoriya, metody i prilozheniya [Linear Programming: Theory, Methods and Applications]. Moscow, Nauka Publ., 1969. 424 p. (in Russian)
4. *Dikin I. I.* Iterativnoye resheniye zadach lineynogo i kvadratichnogo programmirovaniya [Iterative solution of problems of linear and quadratic programming]. Report AS USSR, 1967, vol. 174, pp. 747–748. (in Russian)
5. *Khachiyan L. G.* Polinomial'nyy algoritm v lineynom programmirovanii [Polynomial algorithm in linear programming]. Report AS USSR, 1979, vol. 244, pp. 1093–1096. (in Russian)
6. *Zorkaltsev V. I.* Dvoystvennyye algoritmy vnutrennikh toчек [Dual algorithms of interior-points]. Iz. VUZ, Mathematics, 2011, no. 4 pp. 33–53. (in Russian)
7. *Zorkaltsev V. I. and Medvejonkov D. S.* Chislennyye eksperimenty s variantami algoritmov vnutrennikh toчек na nelineynykh zadachakh potokoraspredeleniya [Computational experiments with variants of interior-point algorithms for nonlinear flow distribution problems]. Institute of Energy Systems of SB RAS, 2013, no. 46, pp. 68–87. (in Russian)
8. *Medvejonkov D. S.* Eksperimental'nyye issledovaniya algoritmov vnutrennikh toчек na nelineynykh zadachakh potokoraspredeleniya [Experimental researches of interior-point algorithms for solution of flow distribution nonlinear problems]. Bulletin of the Buryat State University, 2013, no. 9, pp. 12–16. (in Russian)
9. *Vylegzhanin O. N. and Skatova G. I.* Resheniye zadachi lineynogo programmirovaniya s ispol'zovaniem operatora-proektora [The solution of the linear programming problem using the operator-projector]. News Tomsk Polytechnic University, 2009, vol. 314, no. 5, pp. 37–40. (in Russian)
10. *Bakhshiyan B. Ts. and Gurianov A. V.* Skeletnyy algoritm resheniya zadachi lineynogo programmirovaniya i ego primeneniye dlya resheniya zadach otsenivaniya [The skeletal algorithm for solving the linear programming problem and its application for solving estimation problems]. Bulletin MAI, 2008, vol. 15, no. 2, pp. 5–16. (in Russian)
11. *Gordunovski V. M.* Metod ehksponencial'noj approksimacii dlya lineynogo programmirovaniya [The Method of Exponential Approximation for Linear Programming]. Moscow, Editus Publ., 2013. 32 p. (in Russian)

12. *Kolosov V. S.* Konechnyy parametricheskiy metod resheniya zadachi lineynogo programmirovaniya [The finite parametric method for solving linear programming problem]. Scientific Works MLTI, 1978, vol. 103, pp. 197–198. (in Russian)
13. *Kolosov V. S.* Rol' dvoystvennykh otsenok v parametricheskom metode resheniya zadachi lineynogo programmirovaniya [The role of dual estimates in the parametric method for solving the linear programming problem]. Scientific Works MLTI, 1986, vol. 183, pp. 51–54. (in Russian)
14. *Karmanov V. G.* Matematicheskoe programmirovaniye [Mathematical Programming]. Moscow, FIZMATLIT Publ., 2004. 264 p.