НОВЫЙ АЛГОРИТМ ДВОЙСТВЕННОЙ ДЕКОМПОЗИЦИИ С ПРИЛОЖЕНИЕМ К ЗАДАЧАМ ТРАНСПОРТНОГО ТИПА*

© 1994 Гольштейн Е.Г., Немировский А.С., Соколов Н.А.

(Москва)

Описан новый декомпозиционный алгоритм двойственного типа, обладающий практической линейной скоростью сходимости. Алгоритм использован для решения трехиндексной транспортной задачи с аксиальными суммами. Приведены результаты численного эксперимента, подтверждающие эффективность алгоритма.

1. Рассмотрим задачу линейного программирования (ЛП)

$$(c, x) \rightarrow \max, \quad Ax = b, \quad A^1x = b^1, \quad x \geq 0, \quad (1)$$

где $c, x \in E_n, b \in E_m, b_1 \in E_{m_1}, A$ и A^1 – матрицы соответствующих размеров, E_l – l -мерное евклидово пространство. Предполагается, что система уравнений $A^1x = b^1$ обладает спецификой, облегчающей решение задачи ЛП вида

$$(c', x) \rightarrow \max, \quad A^1x = b^1, \quad x \geq 0 \quad \forall c' \in E_n. \quad (2)$$

Например, матрица A^1 может иметь блочную структуру (как в [1], где впервые был предложен декомпозиционный подход), либо являться матрицей условий специальной задачи ЛП, для которой существуют эффективные методы решения.

Общая схема двойственной декомпозиции задачи (1) состоит в следующем.

Введем функцию $f: E_m \rightarrow E_1$, положив

$$f(y) = \max_{x \in G} (c - y^T A, x) + (y, b), \quad (3)$$

где

$$G = \{x \in E_n: A^1x = b^1, \quad x \geq 0\}. \quad (4)$$

Очевидно, f – выпуклая, кусочно-линейная функция. В дальнейшем предполагается, что многогранное множество G непусто и ограничено, т.е. является непустым многогранником. В этом случае функция f определена конечными значениями во всех точках $y \in E_m$.

Решение задачи (1) сводится к минимизации функции f в том смысле, что если $\bar{y} \in \text{Arg min}_{y \in E_m} f(y)$, то среди оптимальных планов (2) при $c' = c - \bar{y}^T A$ содержится решение исходной задачи (1) (т.е. тот, который удовлетворяет системе уравнений $Ax = b$).

Из определения (3) функции f вытекает, что при любом $y \in E_m$ ее субдифференциал ∂f выражается формулой: $\partial f(y) = \{z = b - Ax: x \in G_y^*\}$, где G_y^* – множество оптимальных планов (2) при $c' = c - y^T A$. Поэтому для вычисления $f(y)$ и некоторого субградиента

*Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (код проекта 93-012-499),

$f'(y) \in \partial f(y)$ достаточно найти одно из решений задачи (2) при $c' = c - y^T A$: если x_y — найденное решение, то

$$f(y) = (c, x_y) + (y, b - Ax_y), \quad f'(x) = b - Ax_y. \quad (5)$$

Из (5) следует, что любой сходящийся алгоритм минимизации первого порядка выпуклой (но не обязательно дифференцируемой) функции приводит к итеративной процедуре решения задачи (1), на каждом шаге которой необходимо решать задачу (2) при некотором c' . Такого рода процедуры принято называть декомпозиционными (или блочными) методами, причем, учитывая тесную связь этих методов с процессом минимизации двойственной функции f , подчеркивать в названии их двойственный характер.

Заметим, что при реализации двойственного блочного метода (ДБМ) необходимо, в частности, иметь удовлетворительный способ решения так называемой проблемы возврата к исходной задаче, т.е. уметь достаточно экономно отыскивать приближенное решение (1) по найденному приближенному решению задачи минимизации функции f .

Для построения двойственных декомпозиционных схем ранее использовались методы возможных направлений [2], субградиентные методы и так называемый r -алгоритм [3]. В настоящей работе мы строим ДБМ на базе нового алгоритма минимизации выпуклой функции, предложенного в [4] и получившего название "метод уровней". Выбор этого метода для реализации двойственной декомпозиции связан, во-первых, с его эффективностью (в смысле числа вычислений $f(y)$ и $f'(y)$), т.е. числа решенных задач типа (2)) и возможностью решить проблему возврата к исходной задаче без дополнительных вычислений.

2. Остановимся вкратце на методе уровней, лежащем в основе предлагаемого ниже двойственного декомпозиционного подхода.

Пусть $f: Q \rightarrow E_1$ — выпуклая липшицева функция, определенная на выпуклом многограннике $Q \subset E_n$. Требуется решить с определенной точностью по функционалу экстремальную задачу

$$f(y) \rightarrow \min, \quad y \in Q. \quad (6)$$

Метод уровней представляет собой итеративную процедуру решения задачи (6) первого порядка, на каждом шаге которой выбирается очередная точка $y \in Q$. После $(k - 1)$ шагов метода мы располагаем k точками $y^i \in Q$, $i = 1, \dots, k$ (y^1 выбирается произвольно). Предполагается, что выбрав точку y^i мы можем вычислить значение $f(y^i)$ функции f и один из ее субградиентов $l^i \in \partial f(y^i)$ в y^i . Информация о f , которой мы располагаем после $(k - 1)$ шагов метода, позволяет построить ее аппроксимацию f_k вида

$$f_k(y) = \max_{1 \leq i \leq k} [f(y^i) + (l^i, y - y^i)]. \quad (7)$$

Из (7) и выпуклости f вытекают соотношения

$$f_k(y) \leq f(y), \quad y \in Q; \quad f_k(y^i) = f(y^i), \quad i = 1, \dots, k. \quad (8)$$

Очередной k -й шаг метода уровней начинается с решения задачи

$$f_k(y) \rightarrow \min, \quad y \in Q, \quad (9)$$

которая, очевидно, представляет собой задачу ЛП.

$$\text{Пусть } f_k^* = \min_{y \in Q} f_k(y), \quad f^* = \min_{y \in Q} f(y), \quad \bar{y}^k = \arg \min_{1 \leq i \leq k} f(y^i), \quad \Delta_k = f(\bar{y}^k) - f_k^*.$$

Из (8) с очевидностью вытекает следующая оценка сверху для отклонения значения минимизируемой функции f в точке $\bar{y}^k$ от ее минимального значения f^*

$$f(\bar{y}^k) - f^* \leq \Delta_k. \quad (10)$$

Если $\Delta_k \leq \varepsilon$, где ε – заданная точность решения задачи (6), то в качестве ее приближенного решения выбирается точка $\bar{y}^k$. В противном случае переходят к построению очередной точки $y^{k+1} \in Q$.

В качестве точки y^{k+1} принимается проекция точки y^k на многогранник $\left\{ y \in Q: f_k \leq \frac{1}{2}(f(y^k) + f_k^*) \right\}$, представляющий собой множество уровня аппроксимации f_k (отсюда и название метода). Отыскание точки y^{k+1} сводится к решению задачи квадратичного программирования (КП)

$$|y - y^k|^2 \rightarrow \min, \quad f(y^i) + (l^i, y - y^i) \leq t_k, \quad i = 1, \dots, k, \quad y \in Q, \quad (11)$$

где $t_k = \frac{1}{2}(f(y^k) + f_k^*)$.

Итак, каждый шаг метода уровней сводится к решению одной задачи ЛП (9) и одной – КП (11).

Для метода уровней в [4] получена следующая оценка скорости сходимости

$$f(\bar{y}^k) - f^* \leq 2.3(\bar{d}\bar{l})k^{-\frac{1}{2}}, \quad k = 1, \dots, \quad (12)$$

где $\bar{d}$ – диаметр многогранника Q , а $\bar{l}$ – постоянная Липшица функции f .

Оценка (12) интересна тем, что гарантирует оптимальность метода уровней при невысоких требованиях к точности в смысле информационной трудоемкости алгоритмов (соответствующие определения можно найти в [5]). Естественно, что метод уровней не представлял бы большого практического интереса, если бы следовал оценке (12). Однако, как показали многочисленные вычислительные эксперименты, проведенные в ЦЭМИ, Технионе (г. Хайфа, Израиль) и INRIA (Франция), метод уровней сходится со скоростью геометрической прогрессии, знаменатель которой зависит от m – размерности пространства, содержащего Q , таким же образом, как и в оптимальном методе при высоких требованиях к точности. Во всех расчетах соблюдалось неравенство

$$f(\bar{y}^k) - f^* \leq \bar{d}\bar{l} \exp\left(-\frac{k}{\mu m}\right), \quad k = 1, \dots, \quad (13)$$

где константа μ не превышает единицы.

Согласно (13), метод уровней ведет себя как оптимальный, если его трудоемкость оценивать числом точек, в которых вычисляется значение функции и ее субградиент. Подобная оценка трудоемкости вполне оправдана в случаях, когда получение необходимой информации о функции и ее субградиентах требует значительных вычислительных усилий. Именно так обстоит дело при применении метода к декомпозиции, когда определение $f(y^i)$ и $l^i \in \partial f(y^i)$ сводится к решению экстремальной задачи, зависящей от y^i .

3. Вернемся к исходной задаче (1). Чтобы использовать метод уровней для ее двойственной декомпозиции, перейдем от (1) к семейству задач

$$P(R): (c, x) - R|Ax - b|_1 \rightarrow \max, \quad x \in G, \quad (14)$$

где $R > 0$ – параметр штрафа, $|y|_1 = \sum_{i=1}^m |y_i|$, а многогранник G , как и прежде, задается соотношением (4).

Если положить $F(x, y) = (c, x - y^T A) + (b, y)$, $Q(R) = \{y \in E_m: |y_i| \leq R, i = 1, \dots, m\}$ и вспомнить определение (3) функции f , то нетрудно проверить, что

$$\min_{y \in Q(R)} F(x, y) = (c, x) - R|Ax - b|_1, \quad \max_{x \in G} F(x, y) = f(y).$$

Поэтому задача, двойственная к (14), имеет вид

$$\tilde{P}(R): f(y) \rightarrow \min, \quad y \in Q(R), \quad (15)$$

где f определяется формулой (3).

В силу классической теоремы Дж.ф. Неймана (14) и (15) связаны соотношением двойственности; их общее экстремальное значение обозначим через $V(R)$. При $R \rightarrow \infty$ задачи (14) и (15) переходят соответственно в (1) и двойственную к ней

$$f(y) \rightarrow \min, \quad y \in E_m. \quad (16)$$

Задачи (1) и (16) всегда связаны соотношением двойственности, их общее экстремальное значение $V = V(\infty)$ в случае совместности ограничений $Ax = b$, $A^1x = b^1$, $x \geq 0$, конечно, а при несовместности этих ограничений равно $-\infty$. В дальнейшем предполагается, что задача (1) разрешима, т.е. $V = V(\infty)$ – конечно.

Применим метод уровней к (15), выбрав предварительно некоторое $R > 0$.

После $(k - 1)$ итераций имеем точки $y^i \in Q(R)$, $i = 1, \dots, k$, $x^i \in G$, $i = 1, \dots, k - 1$ (y^1 выбирается произвольно). Следующий, k -й, шаг метода начинается с вычисления точки x^k , в качестве которой берется одно из решений задачи (2) при $c' = c - (y^k)^T A$. Как отмечалось, $f(y^i) = (c - (y^i)^T A, x) + (b, y^i)$, а вектор $l^i = b - Ax^i \in \partial f(y^i)$. Поэтому $f(y^i) + (l^i, y - y^i) = (c, x^i) + (b - Ax^i, y)$. Следовательно, соотношение (7) для аппроксимации f_k функции f можно записать в виде

$$f_k(y) = \max_{1 \leq i \leq k} [(c, x^i) + (b - Ax^i, y)]. \quad (17)$$

Обозначим через $G(k)$ выпуклую оболочку точек $x^i \in G$, $i = 1, \dots, k$. Очевидно, (17) можно переписать в эквивалентной форме

$$f_k(y) = \max_{x \in G(k)} [(c, x) + (b - Ax, y)],$$

откуда следует, что задачи

$$\tilde{P}(R, k): f_k(y) \rightarrow \min, \quad y \in Q(R), \quad (18)$$

и

$$P(R, k): (c, x) - R \|Ax - b\|_1 \rightarrow \max, \quad x \in G(k), \quad (19)$$

взаимно двойственны и имеют по теореме Дж.ф. Неймана общее экстремальное значение $V(R, k)$.

Продолжая k -ю итерацию, находим оптимальный план задачи $\tilde{P}(R, k)$, которая в силу (17) является задачей ЛП

$$V \rightarrow \min, \quad (c, x^i) + (b - Ax^i, y) \leq V, \quad i = 1, \dots, k, \quad |y_i| \leq R, \quad i = 1, \dots, m. \quad (20)$$

Одновременно с ее решением определяется оптимальный план $\underline{x}^k$ задачи (19), ей двойственной, причем, как легко видеть

$$\underline{x}^k = \sum_{i=1}^k \gamma_i^k x_i, \quad (21)$$

где γ_i^k , $i = 1, \dots, k$, – первые k компонент оптимального плана задачи, двойственной к (20). Поскольку для решения последней используется двойственный симплекс-метод, в сумме из правой части (21) всегда содержится не более, чем $(m + 1)$ слагаемых.

Пусть, как и прежде, $\bar{y}^k = \arg \min_{1 \leq i \leq m} f(y^i)$. Тогда, учитывая соотношения $(c, \underline{x}^k) - R \|A \underline{x}^k - b\|_1 = V(R, k) \leq V(R) \leq f(\bar{y}^k)$, имеем

$$V(R) - [(c, \underline{x}^k) - R \|A \underline{x}^k - b\|_1] \leq \Delta_k, \quad (22)$$

где $\Delta_k = f(\bar{y}^k) - (c, \underline{x}^k) + R \|A \underline{x}^k - b\|_1$.

Если число Δ_k не превосходит заранее выбранную точность $\varepsilon > 0$; то вычислительный процесс завершается. В противном случае решается задача КП (11) и y^i пополняются новой точкой $y^{k+1} \in Q(R)$. Этим завершается k -й шаг алгоритма.

4. Сделаем несколько замечаний относительно вычислительной реализации алгоритма двойственной декомпозиции.

Задачи $\tilde{P}(R, i)$ и $\tilde{P}(R, i + 1)$, возникающие на двух соседних итерациях блочного метода, различаются лишь одним дополнительным ограничением. Поэтому для решения $\tilde{P}(R, k)$ целесообразно использовать двойственный симплекс-метод, т.е. симплекс-метод, примененный к $P(R, k)$. Задача проектирования (11) решается двойственным методом КП, предложенным в [6].

Перед тем как затронуть вопрос о выборе параметра R , докажем одно утверждение. Начнем с определения. Пусть $\varepsilon \geq 0$.

Точка $x \in G$ называется ε -решением задачи (1), если

$$(c, x) \geq V - \varepsilon, \quad \rho(x) = \max_{1 \leq i \leq m} |(Ax - b)_i| \leq \varepsilon,$$

где $(Ax - b)_i$ — i -я координата вектора $Ax - b$.

Положим

$$L = \max_{x \in G} (c, x), \quad l = \min_{x \in G} (c, x).$$

Теорема. Пусть задача (1) разрешима, $\varepsilon > 0$ и $R \geq \frac{L-l}{\varepsilon}$. Если после k итераций двойственного блочного метода оказывается выполненным неравенство $\Delta_k \leq \varepsilon$, то $\underline{x}^k$ — ε -решение задачи (1).

Доказательство. Из определения L и неравенства $\rho(x) \leq |Ax - b|_1$ имеем при любом $R > 0$

$$L - R\rho(\underline{x}^k) \geq (c, \underline{x}^k) - R|A\underline{x}^k - b|_1. \quad (23)$$

Поскольку задача (1) разрешима, то существует точка $x^0 \in G$, для которой $Ax^0 = b$, откуда, учитывая определение l , получаем: $(c, \underline{x}^k) - R|A\underline{x}^k - b|_1 \geq (c, x^0) \geq l$. Последнее неравенство вместе с (23) и требованием к R в условии теоремы дает

$$\rho(\underline{x}^k) \leq \frac{L-l}{R} \leq \varepsilon. \quad (24)$$

Далее, из (22) и требования к Δ_k в условии теоремы вытекает, что

$$(c, \underline{x}^k) \geq (c, \underline{x}^k) - R|A\underline{x}^k - b|_1 \geq V(R) - \varepsilon \geq (c, x^*) - \varepsilon, \quad (25)$$

где x^* — оптимальный план задачи (1).

Неравенства (24) и (25) означают, что $\underline{x}^k$ — ε -решение задачи (1). Теорема доказана.

Выбрать параметр R можно двумя способами. Первый основан на утверждении теоремы. Однако найденное таким образом R может быть весьма велико, что грозит замедлить вычислительный процесс. В этом случае следует постепенно наращивать R ,

пользуясь, например, формулой $R = R_0 2^d$, где $d = 0, 1, \dots$, а $R_0 < \frac{L-l}{\varepsilon}$.

Пусть $\underline{x}^k(d)$ — результат решения задачи $P(R_0 2^d)$ с точностью ε (в том смысле, что $\Delta_k \leq \varepsilon$). Если $|A\underline{x}^k(d) - b|_1 \leq \varepsilon$, то вектор $\underline{x}^k(d)$ является, как нетрудно видеть, ε -решением задачи (1). В противном случае d заменяется на $d + 1$, и процесс решения продолжается. Очевидно, придется наращивать параметр R не более, чем

$$\left[\ln \frac{L-l}{\varepsilon R_0} / \ln 2 \right] + 1 \text{ раз.}$$

5. Изложенный декомпозиционный подход имеет смысл использовать, например, для класса транспортных задач (в матричной постановке и на сети) с m дополнительными линейными ограничениями общего вида. Для этих задач существуют чрезвычайно эффективные программные средства, которые позволяют реализовать ДБМ достаточно экономно.

В качестве примера нами была выбрана трехиндексная транспортная задача с аксиальными суммами

$$\begin{aligned} \sum_{i_1=1}^p \sum_{i_2=1}^q \sum_{i_3=1}^r c_{i_1 i_2 i_3} x_{i_1 i_2 i_3} &\rightarrow \max, \\ \sum_{i_2=1}^q \sum_{i_3=1}^r x_{i_1 i_2 i_3} &= \alpha_{i_1}, \quad \sum_{i_1=1}^p \sum_{i_3=1}^r x_{i_1 i_2 i_3} = \beta_{i_2}, \\ \sum_{i_1=1}^p \sum_{i_2=1}^q x_{i_1 i_2 i_3} &= \gamma_{i_3}, \quad x_{i_1 i_2 i_3} \geq 0, \quad i_1 = 1, \dots, p, \quad i_2 = 1, \dots, q, \quad i_3 = 1, \dots, r, \end{aligned} \quad (26)$$

которая имеет pqr переменных и $p + q + r$ ограничений-равенств (из которых $p + q + r - 2$ линейно независимы). Будем для определенности считать $p \leq q \leq r$. При применении к (26) двойственного декомпозиционного подхода роль связывающих ограничений $Ax = b$ играет система уравнений

$$\sum_{i_2=1}^q \sum_{i_3=1}^r x_{i_1 i_2 i_3} = \alpha_{i_1}, \quad 1 \leq i_1 \leq p.$$

Задача (2) в данном случае имеет вид

$$\begin{aligned} \sum_{i_1=1}^p \sum_{i_2=1}^q \sum_{i_3=1}^r c'_{i_1 i_2 i_3} x_{i_1 i_2 i_3} &\rightarrow \max, \\ \sum_{i_1=1}^p \sum_{i_3=1}^r x_{i_1 i_2 i_3} &= \beta_{i_2}, \quad \sum_{i_1=1}^p \sum_{i_2=1}^q x_{i_1 i_2 i_3} = \gamma_{i_3}, \quad x_{i_1 i_2 i_3} \geq 0, \\ i_1 &= 1, \dots, p, \quad i_2 = 1, \dots, q, \quad i_3 = 1, \dots, r, \end{aligned} \quad (27)$$

и легко сводится к обычной транспортной задаче

$$\begin{aligned} \sum_{i=1}^q \sum_{j=1}^r c_{ij} x_{ij} &\rightarrow \max, \quad \sum_{j=1}^r x_{ij} = \beta_i, \quad \sum_{i=1}^q x_{ij} = \gamma_j, \\ x_{ij} &\geq 0, \quad i = 1, \dots, q, \quad j = 1, \dots, r, \end{aligned}$$

если положить

$$x_{ij} = \sum_{i_1=1}^p x_{i_1 i j}, \quad c_{ij} = \max_{1 \leq i_1 \leq p} c_{i_1 i j}.$$

Заметим, что при двусторонних ограничениях на переменные задачи (26) аналог (27) также сводится к двухиндексной транспортной задаче на максимум, но теперь уже с вогнутой сепарабельной кусочно-линейной целевой функцией (подобные задачи также допускают весьма эффективные методы решения).

Практически ДБМ был опробован на широком наборе задач типа (26). Расчеты проводились на ПК IBM 486 (с сопроцессором, тактовая частота 33 МГц, объем оперативной памяти 4 Мб) с помощью программы, написанной на языке FORTRAN (транслятор NDP FORTRAN, версия 1.4). В качестве решателя задач ЛП и КП была использована программа [7], основанная на методе, изложенном в [6]. Для решения транспортных задач применялась программа MEXTR [8]. Вся информация хранилась в оперативной памяти. Размеры задач: $5 \leq p \leq 40$, p кратно 5, $p \leq q \leq r \leq 100$, q, r

Таблица 1

q	r	k	k_p/k	k_q/k	π	ρ	T	time
20	20	123,1	18,56	11,89	28,59	0,34	5,81	31,36
20	30	123,7	18,75	11,77	29,74	0,35	8,03	33,80
20	40	136,1	18,59	12,17	30,35	0,38	9,77	41,66
20	50	108,0	18,22	10,65	26,28	0,30	12,72	30,79
20	60	124,9	19,52	11,31	29,99	0,35	13,72	40,61
20	70	122,7	19,12	11,45	28,49	0,34	14,99	41,44
20	80	124,5	18,73	11,35	28,89	0,35	16,88	43,20
20	90	122,2	19,36	11,13	29,05	0,34	18,17	45,02
20	100	126,8	19,71	11,19	30,58	0,35	18,51	50,49
30	30	126,4	19,40	11,89	28,59	0,35	10,86	38,03
30	40	128,1	19,16	11,69	30,38	0,36	13,21	42,56
30	50	135,2	19,65	12,12	31,81	0,38	15,13	49,18
30	60	121,0	19,08	11,28	28,88	0,34	17,72	45,48
30	70	118,3	18,39	10,68	28,32	0,33	19,44	44,77
30	80	122,1	19,02	11,42	29,53	0,34	20,56	51,01
30	90	126,7	19,62	11,76	31,36	0,36	21,55	58,21
30	100	118,8	18,83	11,20	28,45	0,33	23,63	54,35
40	40	126,1	19,33	11,98	29,28	0,35	16,15	45,22
40	50	123,1	18,80	11,47	28,67	0,34	19,06	47,03
40	60	117,5	19,03	10,93	29,25	0,33	20,89	48,31
40	70	127,3	18,62	11,82	30,56	0,36	22,05	57,18
40	80	122,6	19,09	11,48	31,63	0,34	23,33	59,82
40	90	115,8	19,42	11,45	27,83	0,32	25,44	60,29
40	100	139,9	19,99	12,68	32,20	0,39	25,68	80,42
50	50	134,6	20,08	11,97	32,19	0,38	20,72	61,19
50	60	113,1	18,81	11,16	27,76	0,32	23,52	52,18
50	70	140,5	20,82	12,77	33,81	0,39	23,18	76,76
50	80	134,1	19,57	12,50	29,04	0,38	25,77	74,63
50	90	139,1	19,78	12,16	32,10	0,39	27,14	85,80
50	100	148,4	20,58	13,13	34,39	0,42	27,26	100,39
60	60	117,1	19,07	11,19	29,28	0,33	25,34	60,11
60	70	135,8	19,62	12,71	33,10	0,38	25,93	80,08
60	80	113,6	18,90	11,37	29,46	0,32	28,17	69,23
60	90	123,3	19,20	11,68	29,44	0,35	29,85	82,41
60	100	144,5	19,46	12,83	34,57	0,40	29,66	106,69
70	70	127,6	19,75	12,17	31,50	0,36	27,73	82,19
70	80	126,9	19,95	12,20	30,18	0,36	29,24	88,41
70	90	135,7	20,76	12,49	31,28	0,38	30,91	105,17
70	100	122,9	18,97	11,77	29,77	0,35	31,93	98,76
80	80	134,5	19,20	12,23	32,09	0,38	30,86	102,33
80	90	124,1	19,72	11,84	31,30	0,35	32,98	103,64
80	100	138,8	20,33	12,70	34,28	0,39	32,15	128,43
90	90	129,4	19,96	12,09	31,70	0,36	34,30	119,72
90	100	143,8	20,77	12,91	33,14	0,40	33,41	142,93
100	100	126,3	19,45	12,55	30,90	0,35	35,05	133,33

кратны 10, $pqr \leq 324000$ (ограничение программы). Для каждого набора (p, q, r) решалась серия из 15 тестовых задач (с относительной точностью $\epsilon = 10^{-5}$), в которых числа $\alpha_i, \beta_i, \gamma_i, c_{i_1 i_2 i_3}$ генерировались с помощью датчика случайных чисел так, что $0 \leq \alpha_i, \beta_i, \gamma_i \leq 10^4, 0 \leq c_{i_1 i_2 i_3} \leq 1$.

Основное отличие применяемого в декомпозиционном алгоритме метода уровней от описанного заключается в том, что используется вычислительный прием "обновление": когда величина Δ_k уменьшается в два раза, то вместо всех ограничений-неравенств,

p	var	k	k/p	k_l/k	k_l/kp	k_q/k	k_q/kp	π	π/p	μ
5	55	32,2	6,44	5,64	1,13	4,18	0,84	5,88	1,18	0,35
10	55	61,4	6,14	9,95	0,99	6,81	0,68	12,97	1,30	0,34
15	45	89,4	5,96	14,54	0,97	9,14	0,61	20,61	1,37	0,33
20	45	127,4	6,37	19,39	0,97	11,85	0,59	30,44	1,52	0,36
25	36	160,3	6,41	24,73	0,99	14,16	0,57	38,90	1,56	0,36
30	36	210,9	7,02	30,59	1,02	18,20	0,61	51,31	1,71	0,39
35	27	254,3	7,27	37,05	1,06	19,91	0,57	62,59	1,77	0,41
40	26	288,7	7,22	42,54	1,06	26,77	0,67	69,42	1,74	0,40

определенных точками $x^1, \dots, x^k$ (к моменту проведения k -й итерации их накопилось бы k штук), задающих модель, оставляются только те ограничения, которые входят в оптимальный базис задачи (20). При этом, естественно, теряется часть накопленной информации о функции f , однако вспомогательные задачи становятся меньшего размера и суммарный выигрыш во времени значителен.

Проведенные вычислительные эксперименты показали высокую эффективность и надежность ДБМ при решении всех тестовых задач. Поскольку результаты вычислений в зависимости от q и r оказались примерно одинаковы для различных p , то приведем их лишь для $p = 20$ (табл. 1). Все результаты в ней усреднены по серии из 15 задач, $R = 10^3$.

Смысл столбцов q и r табл. 1 очевиден. В остальных ее столбцах помещены следующие величины: k – среднее число итераций метода уровней для получения относительной точности ε ; k_l/k – среднее число итераций линейного решателя на одну итерацию метода; k_q/k – среднее число итераций квадратичного решателя на одну итерацию метода; π – среднее число ограничений при решении вспомогательных линейных задач (без обновлений оно было бы равно k); μ – приближенное значение коэффициента μ из (13), определяемое соотношением

$$\mu = \frac{k}{p \ln \frac{L_k - l_k}{\Delta_k}}, \quad L_k = \max_{1 \leq i \leq k} f(y^i), \quad l_k = \min_{1 \leq i \leq k} f(y^i);$$

T – средний процент временных затрат на решение транспортных задач (для нахождения значений $f(y^k)$ и $f'(y^k)$) по отношению к общему времени; $time$ – среднее время (сек.) на нахождение приближенного решения с точностью ε .

Из приведенных результатов можно сделать следующие выводы.

Число итераций k метода уровней мало меняется в зависимости от параметров q и r . Незначительна зависимость от q и r среднего числа итераций линейного и квадратичного решателя на одну итерацию метода. Мало изменяются в зависимости от размеров внутренней транспортной задачи и числа π и μ . Как следовало ожидать, с ростом размеров транспортной задачи растет и доля временных затрат на их решение, а также общее время решения задачи.

В табл. 2 для $R = 10^3$ прослеживается зависимость приведенных в табл. 1 величин от параметра p , причем их усреднение проводилось по всем решенным задачам при фиксированном параметре p (количество решенных серий задач — наборов (q, r) — помещено в табл. 2 в столбец var).

Из анализа этой таблицы можно сделать выводы о том, что в пределах приведенных значений параметра p : для средних значений величин k , k_l/k , k_q/k характерна почти линейная зависимость от него (это особенно хорошо видно при просмотре столбцов 4, 6 и 8, где k , k_l/k , k_q/k поделены на p); заметна тенденция роста числа π с увеличением p ; то же относится в целом и к μ .

Следует отметить, что общее количество итераций k_l и k_q может быть значительно уменьшено, если рационально воспользоваться информацией о решаемых вспомогательных линейных и квадратичных задачах. Используемая нами программа [7] не нацелена на решение серии близких задач. Между тем, в каждой новой линейной задаче к старым ограничениям добавляется только одно дополнительное ограничение. Близкими являются и последовательные задачи проектирования. Таким образом, имея надежную модификацию программы [7], приспособленную к решению серии близких задач, можно рассчитывать на ощутимое сокращение чисел k_l и k_q .

В заключении опишем вычислительные эксперименты по определению оптимального значения параметра R на примере задачи с $p = 20$, $q = 80$, $r = 100$. Для нее $L = 90125$, $l = 113$, $f^* = 90046$. Следовательно, $R = 10^5$ гарантирует получение решения с точностью $\varepsilon = 10^{-5}$. Однако подобную точность можно обеспечить не только для $R = 10^3$, но даже для $R = 0,0015$. При этом число итераций k метода уровней сокращается примерно в два раза. Отсюда вытекает, что величина R мало влияет на эффективность вычислительного процесса, тем не менее, целесообразно начинать расчет с малого значения R , постепенно увеличивая его, как указано в конце п. 4.

ЛИТЕРАТУРА

1. Dantzig G., Wolfe Ph. Decomposition Algorithm for Linear Program // *Econometrica*. 1961. V. 29, N 4.
2. Гольштейн Е.Г., Юдин Д.Б. Новые направления в линейном программировании. М.: Сов. радио, 1966.
3. Шор Н.З. Методы минимизации недифференцируемых функций и их приложения. Киев: Наукова думка, 1979.
4. Lemarechal C., Nemirovskii A., Nesterov Yu. New Variants of Bundle Methods // Research Report N 1508, Institut Nationale de Recherches en Informatique et en Automatique (INRIA). Paris, 1991.
5. Немировский А.С., Юдин Д.Б. Сложность задач и эффективность методов оптимизации. М.: Наука, 1979.
6. Goldfarb D., Idnani A. A Numerically Stable Dual Method for Solving Strictly Convex Quadratic Problems // *Math. Program*. 1983. V. 27. N 1.
7. Powell M.J.D. A FORTRAN Subroutine for Convex Programming // Report NA17, University of Cambridge. Cambridge, 1983.
8. Программа МЕХТР решения матричной транспортной задачи / Пакет анализа оптимизационных моделей ППП ПАОЭМ ЕС, задачи транспортного типа. Калинин: НПО Центр-ПрограммСистем, 1982.

Поступила в редакцию
18 II 1993