

О НЕКОТОРЫХ СОВРЕМЕННЫХ НАПРАВЛЕНИЯХ В МАТЕМАТИЧЕСКОМ ПРОГРАММИРОВАНИИ

Е. Г. ГОЛЬШТЕЙН, С. М. МОВШОВИЧ

(Москва)

Математическое программирование — дисциплина прикладной математики. Поэтому, естественно, что в наибольшей степени ее развитие стимулировали различного рода практические приложения, в частности, приложения к экономике. Однако определенное влияние оказывают на математическое программирование, особенно в последнее время, и некоторые чисто математические проблемы, связанные в основном с задачами неклассического вариационного исчисления. Особое внимание в связи с практическими задачами уделялось методам решения линейных задач большого размера, общих и специальных задач выпуклого программирования, многоэкстремальных задач, задач стохастического и дискретного программирования. Разрабатывались теория и методы решения задач математического программирования в функциональных пространствах.

В настоящей статье мы не стремились охватить все эти направления, а в пределах каждого направления — все или даже основные полученные в последнее время результаты. Рассмотрены лишь три важных для экономических приложений вычислительных аспекта математического программирования: 1) методы решения линейных задач большого размера; 2) методы анализа многоэкстремальных задач; 3) методы целочисленного линейного программирования.

Заметим, что подбор материала в какой-то степени отражает интересы авторов и их мнение о перспективности указанных исследований. Одна из целей статьи — побудить к дальнейшим исследованиям в затронутых здесь направлениях математического программирования.

1. МЕТОДЫ РЕШЕНИЯ ЛИНЕЙНЫХ ЗАДАЧ БОЛЬШОГО РАЗМЕРА

В настоящее время общие конечные методы линейного программирования можно считать достаточно хорошо отработанными как в теоретическом, так и в алгоритмическом плане.

Это относится в особенности к методу последовательного улучшения плана (симплексному методу), послужившему основой для ряда эффективных машинных программ. Поэтому, если размеры задачи линейного программирования и объем оперативной памяти ЭВМ допускают эффективное использование общих методов, не имеет смысла тратить усилия на построение более экономных алгоритмов, базирующихся на учете специальной структуры задачи. Однако многие практические линейные задачи экономического происхождения не удовлетворяют этому требованию и в то же время имеют достаточно простую структуру. Вот почему получили большое распространение так называемые специальные методы линейного программирования, разработанные для численного анализа таких линейных задач, как транспортные, распределительные, трехиндексные и др. Недостатком

подобных методов является прежде всего их узкая направленность. Достаточно, например, изменить или добавить хотя бы одно ограничение задачи, как она перестает быть транспортной или распределительной, и соответствующий специальный метод оказывается неприменимым. С расширением круга приложений моделей линейного программирования настоятельной необходимостью стало создание методов, которые, с одной стороны, имели бы достоинства специальных (достаточно полный учет специфики задачи), а с другой — были бы в значительной степени лишены основного недостатка последних (узкая направленность). Первый шаг в разработке методов, занимающих промежуточное положение между общими и специальными, был сделан Данцигом и Вулфом [1], предложившими свой теперь уже широко известный принцип разложения. Метод Данцига — Вулфа применим к достаточно широкому классу линейных задач вида

$$(c, x) \rightarrow \max, \quad (1)$$

$$Ax = b, \quad (2)$$

$$A^{(1)}x = b^{(1)}, \quad (3)$$

$$x \geq 0. \quad (4)$$

Здесь A и $A^{(1)}$ — матрицы с m строками, n столбцами и m_1 строками, n столбцами соответственно; $b = (b_1, b_2, \dots, b_m)^T$, $b^{(1)} = (b_1^{(1)}, \dots, b_{m_1}^{(1)})^T$, $x = (x_1, x_2, \dots, x_n)^T$ — векторы-столбцы, $c = (c_1, c_2, \dots, c_n)$ — вектор-

строка; $(c, x) = \sum_{j=1}^n c_j x_j$ — скалярное произведение векторов c и x . Предпо-

лагается, что матрица $A^{(1)}$ имеет специальную структуру, существенно упрощающую анализ линейной задачи с ограничениями (3), (4). Например, $A^{(1)}$ может иметь блочно-диагональный вид:

$$A^{(1)} = \begin{pmatrix} A_1^{(1)} & 0 & \dots & 0 \\ 0 & A_2^{(1)} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & A_t^{(1)} & \dots \end{pmatrix},$$

где матрица $A_s^{(1)}$ состоит из m_s строк и n_s столбцов ($s = 1, 2, \dots, t$). В этом случае (рассмотренном Данцигом и Вулфом) линейная задача с условиями (3), (4) распадается, очевидно, на t задач меньших размеров.

Матрица $A^{(1)}$ может также представлять собой матрицу условий некоторой специальной задачи, например транспортной, и, следовательно, к задаче с ограничениями (3) и (4) применим соответствующий специальный метод, например метод потенциалов. Принцип разложения Данцига — Вулфа дает возможность получить решение задачи (1) — (4) путем решения ряда линейных задач с ограничениями (3) и (4). В дальнейшем методы такого типа получили название блочных. К сожалению, метод разложения, несмотря на свою логическую стройность и прозрачность, во многих случаях приводит к крайне медленно сходящемуся вычислительному процессу. В связи с этим он не получил достаточно широкого распространения в вычислительной практике. Последующие исследования, связанные с блочными методами, были направлены главным образом на получение более эффективных вычислительных алгоритмов блочного типа. Весьма плодотвор-

ным здесь оказался двойственный подход к анализу задач блочной структуры, позволивший развить достаточно общие принципы формирования методов блочного программирования. Мы осветим здесь лишь общие контуры этого подхода, имеющего весьма прозрачный геометрический смысл (подробное изложение см. в [2]).

Введем в рассмотрение функцию

$$f(\lambda) = \sup_{x \in G} (c_\lambda, x) + (b, \lambda), \quad (5)$$

где G — многогранное множество точек x , удовлетворяющих условиям (3), (4); $c_\lambda = c - \lambda A$, $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$. Будем для простоты считать, что G — ограниченное множество, т. е. выпуклый многогранник. Тогда в любой точке λ функция $f(\lambda)$ принимает конечное значение. Легко показать, что $f(\lambda)$ — кусочно-линейная, выпуклая вниз функция. Решение задачи (1) — (4) сводится к минимизации $f(\lambda)$, причем точка λ^* , в которой $f(\lambda)$ достигает минимума, является вектором оценок условий (2). Обозначим через R множество точек $\bar{\lambda} = (\lambda_0, \lambda)$, расположенное в $(m+1)$ -мерном пространстве, для которых $\lambda_0 \geq f(\lambda)$. Переход к пространству точек $\bar{\lambda}$ дает возможность, в частности, отчетливо представить себе геометрию различных блочных методов. Обратимся, например, к методу разложения, считая для сохранения геометрической наглядности, что $m = 2$, т. е., что $\lambda = (\lambda_1, \lambda_2)$ — точка плоскости.

Проведем через произвольные три грани поверхности $\lambda_0 = f(\lambda)$ плоскости Π_1, Π_2, Π_3 и точку их пересечения обозначим через $\lambda' = (\lambda'_0, \lambda')$. При этом грани следует выбрать таким образом, чтобы трехгранный конус, образованный плоскостями Π_i ($i = 1, 2, 3$), содержал множество R , а горизонтальная плоскость, проходящая через вершину конуса, была для него опорной.

Если $\lambda'_0 = f(\lambda')$, то $\lambda' = \lambda^*$ — искомая точка минимума. В противном случае ($\lambda'_0 < f(\lambda^*)$) выбирается плоскость, проходящая через некоторую грань множества R , содержащую точку $(f(\lambda'), \lambda')$, и заменяется одна из плоскостей Π_i . Новая тройка плоскостей определяет новую точку пересечения $\lambda'' = (\lambda''_0, \lambda'')$, причем в невырожденном случае $\lambda''_0 > \lambda'_0$. Процесс продолжается до тех пор, пока очередная точка пересечения окажется на поверхности $\lambda_0 = f(\lambda)$.

Как видим, в процессе минимизации функции $f(\lambda)$ методом разложения промежуточные точки расположены под поверхностью $\lambda_0 = f(\lambda)$. Методы, основанные на непосредственной минимизации функции $f(\lambda)$, связаны с движением по самой этой поверхности. Общая схема непосредственной минимизации функции $f(\lambda)$ состоит в следующем.

Пусть λ' — произвольная исходная точка. Направление l' назовем подходящим для точки λ' , если $f(\lambda' + \theta' l') < f(\lambda')$ при некотором $\theta' > 0$, т. е. некоторый сдвиг вдоль l' приводит к убыванию функции $f(\lambda)$. Допустим, что нам известен способ T , который позволяет выбирать подходящее направление, либо фиксировать, что множество этих направлений пусто. Используя способ T , выберем подходящее направление l' (если множество подходящих направлений оказывается пустым, то, очевидно, λ' — искомая точка минимума функции $f(\lambda)$). Далее перейдем от точки λ' к точке $\lambda'' = \lambda' + \theta' l'$. Выбор числа $\theta' > 0$ (длины шага) производится при помощи некоторого способа E . Итак, процесс минимизации функции $f(\lambda)$ складывается из ряда итераций, каждая из которых состоит из двух этапов. Первый этап — выбор подходящего направления — связан со способом T ; второй — вычисление длины шага вдоль выбранного направления — осуществляется при помощи способа E . Способы T и E могут меняться от итерации к итерации. Варьируя способы выбора подходящего направления и длины шага,

можно на основе описанной схемы получать различные блочные методы. Если, например, исходная точка λ' — проекция некоторой вершины множества R ; l' выбирается среди проекций ребер R , выходящих из вершины $\lambda' = (f(\lambda'), \lambda')$; θ' вычисляется из условия перехода в соседнюю с λ' вершину R , то получаем блочный аналог метода уточнения оценок (двойственного симплексного метода). Заметим, что метод разложения — блочный аналог метода улучшения плана. На основе данной схемы можно получить также блочные аналоги остальных конечных методов линейного программирования.

Каждый из этих методов базируется на использовании в течение всего вычислительного процесса фиксированных способов T и E . Отмеченное обстоятельство является, видимо, главной причиной, порождающей их основной недостаток — медленную сходимость вычислительного процесса. Поэтому представляется естественным увязать выбор способов T и E с ходом самого процесса решения. В настоящее время в ЦЭМИ АН СССР ведется работа по составлению машинной программы, основанной на гибкой схеме минимизации функции $f(\lambda)$. Выбраны две пары способов T и E , одна из которых эффективна в начале процесса, а другая лучше приспособлена для использования в достаточной близости от искомого минимума. Момент переключения с одного режима работы на другой будет определяться в процессе решения самой машиной. Вопросы, связанные с построением гибких схем блочного программирования, еще недостаточно разработаны. Исследования в этом направлении должны быть продолжены.

Кроме описанного выше подхода, основанного на идеях разложения, для решения задач линейного программирования вида (1) — (4) разработаны также специальные алгоритмы [3—5], основанные на методе последовательного улучшения плана. В этих алгоритмах специфика задачи используется для более экономного (в сравнении с общей схемой метода улучшения плана) хранения и преобразования на каждой итерации всей необходимой информации. Естественно, что в данном случае в отличие от алгоритмов, рассмотренных выше, движение к оптимуму осуществляется по опорным планам исходной задачи.

Остановимся теперь еще на одной группе методов решения задач линейного программирования большого размера, обладающих специальной структурой. Речь пойдет о методах, основанных на бесконечных итеративных процессах решения матричных игр.

Известная связь между парами двойственных задач линейного программирования и матричными играми позволяет использовать любой метод решения игр для анализа линейных задач. Важное достоинство итеративных алгоритмов теории игр типа методов Брауна и фон-Неймана состоит в том, что их применение к задачам линейного программирования связано с хранением и преобразованием исходных данных задачи, записанных в компактной форме (фиксируются лишь ненулевые элементы таблицы, отвечающей данной задаче, и информация о их расположении в таблице).

Отмеченное свойство итеративных игровых методов позволяет надеяться на возможность использования их для эффективного анализа задач больших размеров со слабо заполненными матрицами (в частности, для задач типа (1) — (4)).

Препятствием этому до сих пор была крайне медленная сходимость итеративных процедур. Однако ряд исследований последних лет, направленных на ускорение сходимости игровых алгоритмов (на некоторых из них мы вкратце остановимся ниже), дают основание для оптимистических прогнозов в пользу бесконечных итеративных процедур.

Напомним прежде всего сущность итеративного метода решения матричных игр, предложенного Брауном, предполагая, что читателю знакомы

элементарные понятия теории матричных игр. Предположим, что необходимо найти оптимальные смешанные стратегии игры, определяемой матрицей B . Процесс вычисления этих стратегий методом Брауна складывается из ряда однотипных итераций. Если считать, что $k-1$ итераций уже проделано и в результате получены смешанные стратегии x_{k-1} и y_{k-1} 1-го и 2-го игроков, то течение k -й итерации происходит следующим образом. 1-й игрок выбирает одну из своих чистых стратегий, скажем, $\tilde{x}_k$, обеспечивающих ему максимальный выигрыш в предположении, что 2-й игрок применяет смешанную стратегию y_{k-1} . Новая смешанная стратегия x_k 1-го игрока определяется формулой:

$$x_k = \frac{1}{k} \tilde{x}_k + \frac{k-1}{k} x_{k-1}.$$

Затем 2-й игрок выбирает свою чистую стратегию $\tilde{y}_k$ таким образом, чтобы обеспечить минимальный выигрыш партнеру, считая, что последний использует смешанную стратегию x_k . Новая смешанная стратегия y_k этого игрока вычисляется по аналогичной формуле:

$$y_k = \frac{1}{k} \tilde{y}_k + \frac{k-1}{k} y_{k-1}.$$

Формулы, определяющие процесс решения игры методом Брауна, удобно записать так:

$$x_k = \lambda_k \tilde{x}_k + (1 - \lambda_k) x_{k-1}, \quad y_k = \mu_k \tilde{y}_k + (1 - \mu_k) y_{k-1},$$

где $\lambda_k = \mu_k = 1/k$.

Этот эквивалентный вид формул, управляющих процессом, сразу же наводит на мысль о возможности следующей модификации метода Брауна. Естественно предположить, что медленная сходимость алгоритма Брауна связана с тем, что программа изменения чисел λ_k и μ_k задана заранее и совершенно не связана с ходом вычислительного процесса. А что, если не фиксировать заранее последовательности чисел $\{\lambda_k\}$, $\{\mu_k\}$, а вычислять их на каждой итерации?

В [6] было отмечено, что в процессе решения игры методом Брауна наилучшие чистые стратегии $\tilde{x}_k$ и $\tilde{y}_k$ остаются неизменными на протяжении ряда итераций. Это позволяет «склеить» некоторую последовательность итераций и на каждой новой итерации соответственно вычислить значения λ_k и μ_k . Для случая выпуклых игр при некоторых дополнительных предположениях (которым, к сожалению, не удовлетворяют матричные игры), в [7] показано, что для сходимости процессов, аналогичных процессу Брауна, достаточно, чтобы числа $\{\lambda_k\}$, $\{\mu_k\}$ удовлетворяли условиям $\lambda_k \rightarrow 0$,

$\mu_k \rightarrow 0$ при $k \rightarrow \infty$ и $\sum_{k=1}^N \lambda_k \rightarrow \infty$, $\sum_{k=1}^N \mu_k \rightarrow \infty$ при $N \rightarrow \infty$. Некоторый класс

процессов решения матричных игр исследовался в [8].

Эксперименты, проведенные в ЦЭМИ АН СССР (см. [9]) по нескольким модификациям метода Брауна, основанным на ряде способов вычисления текущих значений λ_k и μ_k , показали, что таким образом можно существенно ускорить сходимость процесса. Отмеченные модификации метода Брауна подобно самому методу Брауна на каждом шаге приводят к изменению только одной компоненты текущей смешанной стратегии (векторы $\tilde{x}_k$ и $\tilde{y}_k$ содержат лишь по одной единице, остальные их компоненты — нули). Представляется, что выбор в качестве $\tilde{x}_k$ и $\tilde{y}_k$ смешанных стратегий

игроков, связанный с анализом игры, которая порождается некоторой подматрицей матрицы B сравнительно небольших размеров, может еще более ускорить сходимость итеративной процедуры. Заметим, что решение вспомогательных игр следует осуществлять при помощи мало трудоемких приближенных приемов.

Приведем еще одно соображение, реализация которого, как нам кажется, может повысить эффективность итеративных методов игр. Как показывает вычислительный опыт, большинство компонент текущих смешанных стратегий, отвечающих нулевым компонентам оптимальных стратегий, довольно быстро становится сравнительно малым. Представляется целесообразным, обнаружив это, положить их равными нулю, а остальные компоненты попытаться определить путем решения системы линейных (вообще говоря, несовместных) уравнений одним из известных итеративных методов. В процессе решения системы следует следить за знаками неизвестных. Как только выявится неизвестное, имеющее тенденцию стать отрицательным, необходимо вернуться к исходной итеративной процедуре решения игры. Есть основания предполагать, что наметенный симбиоз методов решения игр и несовместных систем линейных уравнений позволит построить еще более эффективные игровые алгоритмы.

Возможности итеративных алгоритмов еще далеко не исчерпаны. Поэтому дальнейшие исследования, направленные на совершенствование итеративных методов анализа задач линейного программирования, представляются нам весьма актуальными и перспективными.

2. СТАТИСТИЧЕСКИЕ МЕТОДЫ АНАЛИЗА МНОГОЭКСТРЕМАЛЬНЫХ ЗАДАЧ

К сожалению, не все задачи математического программирования, возникающие в народнохозяйственной практике, в планировании и управлении производством, укладываются в схему линейного или даже выпуклого программирования. В ряде важных практических задач либо функционал, измеряющий эффективность возможных вариантов решения, либо само множество возможных вариантов решения оказываются невыпуклыми. Естественно, круг возможных постановок таких задач весьма обширен. В общем случае задача состоит в том, что требуется найти максимум (если он существует) произвольного функционала на произвольном же множестве в n -мерном евклидовом пространстве. Однако о методах решения задачи в такой общей постановке и ее свойствах ничего содержательного, по-видимому, сказать нельзя. Мы остановимся здесь только на двух частных типах подобных задач. К первому типу относятся задачи, в которых множество допустимых планов — выпуклый многогранник, а минимизируемый функционал является выпуклой вверх функцией. Такие задачи, как правило, обладают локальными экстремумами. Поэтому они относятся к числу многоэкстремальных задач. Ко второму типу относятся задачи целочисленного линейного программирования, в которых функционал линеен, а множество планов дискретно или распадается на несвязное множество выпуклых многогранников (частично целочисленная задача).

В практике планирования многоэкстремальные задачи возникают, когда издержки производства на единицу выпускаемой продукции достаточно быстро падают с ростом объема выпуска. Хорошо известным важным примером подобной задачи является задача размещения производства. Приведем ее математическую формулировку в простейшем случае:

$$f(x) = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + \sum_{i=1}^m f_i(x_i) \rightarrow \min, \quad (6)$$

$$\sum_{i=1}^m x_{ij} = b_j, \quad j = 1, \dots, n, \quad (7)$$

$$\sum_{j=1}^n x_{ij} \leq x_i, \quad i = 1, \dots, m, \quad (8)$$

$$0 \leq p_i \leq x_i \leq q_i, \quad i = 1, \dots, m, \quad x_{ij} \geq 0, \quad (9)$$

$f_i(x)$ — выпуклые вверх функции, например:

$$f_i(x) = \begin{cases} d_i + c_i x, & x > 0, \\ 0 & x = 0, \end{cases} \quad (10)$$

где $d_i > 0$, $i = 1, \dots, m$.

При описании этой задачи естественно использовать экономическую терминологию, связанную с размещением производства. Заданы объемы потребностей $\{b_j\}$ в некоторой продукции в пунктах потребления $j = 1, \dots, n$. Для удовлетворения этих потребностей необходимо организовать производство продукции в m пунктах производства, причем в i -м пункте производства объем производства может быть выбран в пределах $[p_i, q_i]$. Вектор $X = (x_1, \dots, x_m, x_{11}, \dots, x_{mn})$, удовлетворяющий условиям (7) — (9), называется планом. Величина x_{ij} — поставка из i -го пункта производства в j -й пункт потребления. Если в плане X величина $x_i > 0$, то i -й пункт производства называется функционирующим.

Хотя многоэкстремальные задачи привлекли внимание исследователей не намного позднее, чем задачи выпуклого программирования, успехи в их анализе существенно скромнее. Трудность решения задачи объясняется следующим. Все известные методы математического программирования поразному реализуют одну идею. В некоторой точке (плане прямой, или двойственной, или расширенной задачи) определяется подходящее направление. Вдоль этого направления делается шаг определенной длины, после чего процесс повторяется из вновь построенной точки. Отсутствие подходящего направления в точке гарантирует ее оптимальность. В этом процессе переход к новой точке производится на основе информации о поведении целевой функции в достаточно малой окрестности текущей точки (локальной информации). Если в малой окрестности данной точки нет точек, в которых значение функционала меньше чем в данной, т. е. нет подходящего направления, процесс минимизации прекращается. Совершенно ясно при этом, что данная точка является оптимальной в своей малой окрестности, т. е. локально-оптимальной.

В выпуклых задачах математического программирования гарантируется, что локально оптимальная точка оптимальна на всем множестве планов, т. е. является и глобальным оптимумом. Иначе обстоит дело в задачах минимизации выпуклой вверх функции. В такой задаче может быть много локально-оптимальных планов, в которых значение целевой функции существенно больше, чем в глобальном минимуме. Легко привести пример, в котором все вершины многогранника условий будут локальными минимумами.

Таким образом, в многоэкстремальных задачах отсутствие подходящего направления гарантирует лишь локальную оптимальность соответствующей точки, поэтому использование описанного процесса минимизации, начиная с некоторой точки, приведет нас в какой-либо локальный минимум, быть может, очень плохой.

Для улучшения качества решения необходимо повторить процедуру построения локального минимума многократно, начиная с различных исходных планов. Эффективным способом организации подобной процедуры является метод случайного поиска.

Прежде всего следует отметить, что в задачах минимизации выпуклого вверх функционала на выпуклом многограннике все локальные минимумы достигаются в вершинах. Поэтому в процессе решения можно рассматривать лишь множество вершин, определяя на этом множестве понятия плана, окрестности, локального минимума.

Процесс минимизации выглядит так. Каждая итерация начинается с того, что в соответствии с некоторым распределением вероятностей, заданным на конечном множестве вершин, выбирается исходный план. Затем, отправляясь от этого плана, каким-либо методом строится локально-оптимальный план. Построением локально-оптимального плана итерация завершается. После этого процесс повторяется. Различные реализации описанного метода можно получать, варьируя распределение вероятностей, определение окрестности и метод построения локального минимума.

Поскольку в процессе минимизации можно ограничиться лишь рассмотрением множества вершин многогранника условий, то окрестность любого плана (опорного) задается некоторым конечным подмножеством вершин. Различные определения окрестности получают, по-разному определяя понятие соседства вершин и считая, что окрестность вершины образуется всеми с ней соседними. Наиболее естественным является обычное понятие соседства, когда соседними считаются вершины многогранника, лежащие на одном и том же его ребре. Применительно к условиям (7) — (9) при $p_i = 0$ и отсутствии ограничений на x_i сверху это означает, что в соседних планах различны ровно две поставки. (В опорных планах каждый пункт потребления снабжается ровно одним пунктом производства.) В этой задаче оказывается полезным и другое определение соседства: планы называются соседними, если различны поставки из ровно двух пунктов производства. Ясно, что так определенная окрестность содержит определенную выше.

План называется локально-минимальным, если соответствующее ему значение целевой функции не превосходит значений целевой функции на всех соседних с ним планах. Это определяет пути построения локально-оптимального плана.

Чтобы убедиться в том, что план локально-оптимальный, необходимо просмотреть все соседние с ним планы. Если при этом обнаруживается лучший план, то начинается просмотр его окрестности и т. д., пока после конечного числа шагов не будет достигнут локально-оптимальный план.

Возможность и удобство реализации различных распределений вероятностей на множестве вершин существенно зависит от вида ограничений многоэкстремальной задачи.

Рассмотрим различные возможные способы выбора распределения вероятностей на примере задачи с условиями (7) — (9) при $p_i = 0$ и отсутствии ограничений на x_i сверху.

Простейшим является равномерное распределение вероятностей на множестве вершин. Если это распределение не меняется в процессе итераций, такой поиск называется слепым, поскольку получаемая в процессе счета информация не используется.

Распределение вероятностей может меняться в процессе счета, адаптируясь к решаемой задаче. Одна из возможностей состоит в последовательном изменении распределения вероятностей от итерации к итерации. Можно, например, увеличивать вероятность выбора на данной итерации планов, у которых не функционируют те пункты производства, которые функционировали в локально-оптимальном плане, построенном на предшествующей

итерации. Еще один подход состоит в том, что заранее сужается число функционирующих пунктов производства. Для этого на каждой итерации из всего множества m пунктов производства случайным образом выбирается некоторое подмножество, затем с помощью описанных выше способов решается задача меньшей размерности. Такой процесс повторяется многократно.

Если проводится N итераций и вероятность выбора любой вершины на каждой итерации ограничена снизу положительным числом, то при $N \rightarrow \infty$ наилучшее найденное значение локального минимума f_N сходится по вероятности к значению глобального минимума. Выбор распределения вероятностей и адаптация в процессе решения могут значительно повысить скорость сходимости. Влияние выбора распределения вероятностей на эффективность решения многоэкстремальных задач исследовалась в ЦЭМИ АН СССР ([10]) на примере задачи (6) — (10), в которой все $p_i = 0$ и нет ограничений на x_i сверху. Исследования проводились для описанных выше способов выбора исходного плана.

Естественно, что при росте времени счета значения найденных различными способами приближений к глобальному минимуму будут различаться все меньше и меньше. Поэтому эффективность различных методов разумно сравнить между собой по средним наименьшим значениям целевой функции, полученным при фиксированном и не слишком большом времени счета при решении большого количества однотипных задач. Проводившиеся исследования показали, что при подходящей адаптации распределения вероятностей в процессе счета эффективность алгоритма увеличивается в среднем на 5—10% в сравнении со слепым поиском, причем тем сильнее, чем сильнее задача отличается от линейной.

Примерно такие же результаты были получены для третьего способа построения исходных планов (подробнее см. [10]). В настоящее время на основе проведенных исследований в ЦЭМИ АН СССР разрабатывается программа, реализующая метод случайного поиска применительно к задаче (6) — (9) и некоторым ее обобщениям.

Различные комбинации случайного поиска с процессом построения локальных экстремумов описаны в работах [11—14]. В работе [15] указана статистическая процедура решения одного класса многоэкстремальных задач путем последовательного сужения области определения. В работе [16] предложены статистические методы минимизации функций при наличии ошибок в измерениях.

3. МЕТОДЫ ЦЕЛОЧИСЛЕННОГО ПРОГРАММИРОВАНИЯ

Решение задач целочисленного линейного программирования также связано с большими вычислительными трудностями.

В связи с большой практической важностью этих задач им уделяется много внимания. Однако до сих пор нельзя сказать, что имеются эффективные методы решения для достаточно широкого класса задач.

В народнохозяйственном планировании и управлении целочисленные задачи возникают всякий раз, когда управляющему органу приходится выбирать один из нескольких имеющихся вариантов решения. Стремление применить методы математического программирования к решению вопросов о строительстве крупных объектов, о выборе числа агрегатов на каждом объекте, к задачам теории расписаний, раскроя и т. д., порождает задачи, в которых все или некоторые переменные могут принимать только целые значения.

К целочисленному линейному программированию можно свести многие невыпуклые задачи, в частности задачу (6) — (10).

В ряде целочисленных задач весьма большие трудности возникают даже при попытке построения хотя бы одного плана. В практических задачах экономики обычно какой-либо план построить нетрудно. Однако, как правило, нет возможности определить, является ли данный план оптимальным. Если даже известно, что данный план не оптимален, не существует, как правило, регулярного способа последовательного перехода к лучшим планам, приводящего в конце концов к оптимальному.

Для решения целочисленных задач применяются различные приближенные методы, методы улучшенного перебора вариантов, методы ветвей и границ, методы отсечения, методы случайного поиска. Ниже мы кратко остановимся лишь на двух последних методах.

Одним из наиболее развитых методов решения целочисленных задач является метод отсечения. Этот метод основан на следующей прозрачной идее.

Рассмотрим задачу целочисленного линейного программирования

$$f(x) = (c, x) \rightarrow \max, \quad (11)$$

$$Ax \leq b, \quad (12)$$

$$0 \leq x \leq q. \quad (13)$$

$$x_i - \text{целые числа, } i = 1, \dots, n \quad (14)$$

Прежде всего решается задача (11) — (13). Если компоненты ее решения x^0 — целые, то тем самым решена задача (11) — (14).

Если решение x^0 задачи (11) — (13) не удовлетворяет условиям (14), то строится дополнительное линейное условие, которому, во-первых, точка x^0 не удовлетворяет, т. е. x^0 перестает быть планом, во-вторых, удовлетворяют все целочисленные планы задачи (11) — (13).

Дополнительное ограничение добавляется к условиям (12), после чего задача решается снова, и т. д. Идея подобного метода была впервые высказана Данцигом, Фулкерсоном и Джонсоном в [17]. Однако только в 1958 г. Гомори ([18]) предложил способ построения дополнительных линейных ограничений, при котором достижение оптимального плана (11) — (14) (если он существует) гарантируется за конечное число шагов. Если такого плана нет, это также выясняется за конечное число шагов.

Хотя с помощью предложенных Гомори алгоритмов можно в принципе решить любую задачу целочисленного линейного программирования, встречаются даже небольшие задачи, в которых число дополнительных линейных ограничений, построенных по методу Гомори, превосходит число целочисленных планов задачи (11) — (13). Таким образом, в некоторых случаях метод Гомори менее эффективен, чем полный перебор. Поэтому продолжают исследования с целью обнаружения методов построения более сильных дополнительных условий для различных классов задач.

В частности, в ЦЭМИ АН СССР исследуются способы эффективного построения дополнительных ограничений для некоторых целочисленных задач размещения производства, для целочисленных задач с «булевыми» переменными (в условиях (13) все $q_j = 1$). Остановимся подробнее на последней задаче.

Вместо решения задачи (11) — (14) строятся целочисленные планы задач с условиями (12), (13) и

$$f(x) \geq t \quad (15)$$

для некоторой последовательности значений t . Пусть m и M соответственно нижняя и верхняя границы значений функции $f(x)$ на множестве (12), (13). Для того чтобы получить приближенное решение исходной задачи,

отличающееся от оптимального не более чем на ε , необходимо решить не более $2 + \lceil \log_2 (M - m) / \varepsilon \rceil$ задач построения целочисленного решения системы неравенств (12), (13), (15).

Такая оценка видна из следующего плана решения. Прежде всего решаем задачу для $t = (M + m) / 2$. Если решения нет, переходим к значению $t = (M + m) / 4$. Если решение найдено, переходим к $t = 3/4(M + m)$ и т. д. Таким образом, задача (11) — (14) сведена к решению последовательности задач отыскания целочисленного решения системы неравенств (11), (12), (14). Каждую такую задачу будем называть t -задачей.

Решение t -задачи ищется путем решения задачи лексикографической максимизации вектора

$$x = (x_1, x_2, \dots, x_n) \rightarrow \max \quad (16)$$

при условиях (12) — (14) и (15).

Вектор x лексикографически максимален, если для любого допустимого вектора y первая ненулевая компонента вектора $x - y$ положительна. Для построения лексикографического максимума достаточно максимизировать целевую функцию $f'(x) = x_1$. Если этот максимум равен x_1^0 , то далее решается задача максимизации $f''(x) = x_2$ при дополнительном условии $x_1 = x_1^0$ и т. д.

Решение задачи (12) — (16) осуществляется методом отсечения, т. е. решается задача (12), (13), (15), (16). Если все компоненты вектора x целые, то задача лексикографической максимизации, а вместе с ней и соответствующая t -задача решены.

Пусть теперь после k -й итерации решения задачи (12), (13), (15), (16) первые $v - 1$ -компоненты вектора $x^{(k)}$ целые, а $0 < x_v^{(k)} < 1$. Тогда строится дополнительное линейное ограничение

$$\sum_{j=1}^{v-1} [(1 - x_j)x_j^{(k)} + x_j(1 - x^{(k)})] + 1 - x_v \geq 1. \quad (17)$$

Легко проверить, что план $x^{(k)}$ не удовлетворяет этому ограничению, а любой план задачи (12) — (16) ему удовлетворяет. Ограничение (17) добавляется к ограничениям (12), после чего снова решается задача (12), (13), (15), (16) ($(k + 1)$ -я итерация). После конечного числа шагов либо будет построено решение задачи лексикографической максимизации, либо показано, что она не имеет решения. Описанный алгоритм (подробнее см. [19]) решает задачу лексикографической максимизации не более чем за 2^n итераций (отметим, что для метода отсечения Гомори никакие оценки сверху для числа итераций неизвестны). Этот алгоритм, называемый B -алгоритмом, может применяться и для частично целочисленных задач линейного программирования.

Проводившийся вычислительный эксперимент показал, что B -алгоритм в среднем эффективнее, чем циклический алгоритм Гомори. При эксперименте решались задачи небольшого размера (от 2×8 до 9×30), все коэффициенты которых были целыми, равномерно распределенными на отрезке $[0, 9]$. В качестве характеристики эффективности рассматривалось среднее число дополнительных линейных ограничений, добавляемых в процессе решения задачи каждого размера. Для циклического алгоритма это число в 5—20 раз больше, чем для B -алгоритма.

Рост эффективности объясняется тем, что очень часто дополнительное ограничение типа (17) отсекает больший кусок многогранника, чем ограничения, построенные по методу Гомори. Однако легко построить примеры, когда ни один из отсекаемых кусков не содержит другого.

Несмотря на то, что B -алгоритм и некоторые другие методы, реализующие идею отсечения, заметно эффективнее исходных алгоритмов Гомори, скорость сходимости, обеспечиваемая этими методами, недостаточна для решения задач даже среднего размера на современных ЭВМ. Поэтому в последнее время уделяется большое внимание различным комбинаторным методам решения целочисленных задач и методам случайного поиска.

Один из перспективных, на наш взгляд, методов случайного поиска, разработанный с участием сотрудников ЦЭМИ АН СССР, описан в [20]. Рассматривается задача (11) — (14), в которой все переменные x_j принимают значения либо 0, либо 1. Так же как и в B -алгоритме, вместо исходной решается последовательность t -задач. Однако каждая t -задача решается следующим методом.

В качестве начального плана x^0 выбирается произвольная последовательность нулей и единиц. Пусть на k -м шаге построен вектор $x^{(k)}$. Вычисляются величины

$$\Delta_0 = \max \left(\frac{t - (c, x^{(k)})}{t}, 0 \right), \quad \Delta_i = \max \left(\frac{(a_i, x^{(k)}) - b_i}{b_i}, 0 \right),$$

$$i = 1, \dots, m,$$

после чего определяется вероятность $p = \min(\lambda, \Delta_i, i = 0, 1, \dots, m)$, где постоянная λ выбирается в интервале $(0, 1)$. В проводившихся расчетах было принято $\lambda = 0,5$.

Для построения плана $x^{(k+1)}$ для всех j полагают

$$x_j^{(k+1)} = \begin{cases} x_j^{(k)} & \text{с вероятностью } 1 - p, \\ 0, & \text{если } x_j^{(k)} = 1 \\ 1, & \text{если } x_j^{(k)} = 0 \end{cases} \text{ с вероятностью } p.$$

Процесс заканчивается либо построением плана t -задачи, либо после некоторого наперед выбранного числа итераций. Дальнейший процесс происходит в точности так же, как и в B -алгоритме.

Проводившиеся авторами эксперименты на простейших задачах показали, что предлагаемый ими метод существенно эффективнее полного перебора.

ЛИТЕРАТУРА

1. G. B. Dantzig, F. Wolfe. Decomposition principle for linear programs. *Opns. Res.*, 1960, v. 8, N 1.
2. Е. Г. Гольштейн. Методы блочного программирования. Экономика и матем. методы, 1966, т. 2, вып. 1.
3. К. В. Ким. Об использовании специфики условий задачи в методе улучшения плана. Экономика и матем. методы, 1965, т. 1, вып. 1.
4. Р. А. Звягина. Задача линейного программирования с блочно-диагональными матрицами. Сб. Оптимальное планирование, II. Новосибирск, 1964.
5. Э. А. Мухачева. Транспортная задача на сети с дополнительными ограничениями. Экономика и матем. методы, 1965, т. 1, вып. 4.
6. А. В. Амвросенко. Ускорение сходимости метода Брауна решения матричных игр. Там же.
7. В. А. Волконский. Оптимальное планирование в условиях большой размерности (итеративные методы и принцип декомпозиции). Там же, вып. 2.
8. Лихтеров. Об одном классе процессов для решения матричных игр. Изв. АН СССР, Техническая кибернетика, 1965, № 5.
9. Э. П. Борисова, И. В. Магарик. О двух модификациях метода Брауна решения матричных игр. Экономика и матем. методы, 1966, т. 2, вып. 5.
10. Б. И. Алейников, С. М. Мовшович. Некоторые алгоритмы решения многоэкстремальных задач размещения и их экспериментальное исследование. Сб. Исследования по математическому программированию (в печати).

11. А. Д. Запкин, Р. М. Петухов. Решение задачи размещения промышленных предприятий методом случайного поиска. Сб. Применение математики в экономических исследованиях, т. 3, М., 1965.
12. Л. С. Гурин, В. П. Лобач. Комбинация метода Монте-Карло с методом скорейшего спуска при решении некоторых экстремальных задач. Ж. вычислит. матем. и матем. физики, 1962, т. 2, № 3.
13. Р. З. Хасмьинский. Применение случайного шума в задачах оптимизации и опознания. Проблемы передачи и информации, 1965, № 3.
14. Д. Б. Юдин, Э. М. Хазен. Некоторые математические аспекты статистических методов поиска. Автоматика и вычислит. техника, Рига, 1966, № 13.
15. И. Б. Моцкус. Об одном способе распределения случайных испытаний при решении многоэкстремальных задач. Ж. вычисл. матем. и матем. физики, 1964, т. 4, № 2.
16. Л. А. Растргин. Случайный поиск в задачах оптимизации многопараметрических систем. Рига, Зинатне, 1965.
17. G. B. Dantzig, D. R. Fulkerson, S. Johnson. Solution of largescale travelling salesman problem. Opns. Res., 1954, v. 2, N 3.
18. R. E. Gomory. Outline of an algorithm for integer solutions to linear programs. Bull. Amer. Math. Soc., 1958, v. 64, N 5.
19. Ю. Ю. Финкельштейн. Алгоритм для решения задач целочисленного линейного программирования с булевыми переменными. Экономика и матем. методы, 1965, т. 1, вып. 5.
20. И. И. Пятецкий-Шапиро, В. А. Волконский, Л. В. Левина, А. О. Поманский. Об одном итеративном методе решения задач целочисленного программирования. Докл. АН СССР, 1966, т. 169, № 6.

Поступила в редакцию
1 IV 1967

НОВЫЕ ТЕНДЕНЦИИ В МАКРОЭКОНОМИЧЕСКОМ МОДЕЛИРОВАНИИ ЗА РУБЕЖОМ (ОБЗОР) *

С. М. ВИШНЕВ, Г. Г. ПИРОГОВ

(Москва)

1. ПРИЧИНЫ НЕУДОВЛЕТВОРЕННОСТИ ПРЕЖНИМИ МОДЕЛЯМИ

Пятидесятые годы были периодом большой активности зарубежных экономистов в области построения математических макро- и микроэкономических моделей. Однако по мере накопления опыта все явственнее выступало расхождение между возлагаемыми на эти модели надеждами и получаемыми результатами.

Переход от теоретических разработок к выработке практических мероприятий экономической политики обнаружил ограниченную применимость простых моделей, построенных с использованием принципов мультипликатора и акселератора. Эти модели предназначались для анализа в явном виде динамических свойств систем на основе понятия статического или динамического равновесия. Однако такой анализ требует достаточной простоты системы, и главным образом линейности. Между тем простота моделей не позволяет использовать их для прогнозирования и выработки конкретной экономической политики, а усложнения, и особенно введение нелинейности, делают невозможным анализ системы в явном виде. С другой стороны, при попытках проанализировать при помощи подобных моделей конкретные экономические ситуации исследователи натолкнулись на чрезвычайную неустойчивость решений при малых изменениях структурных коэффициентов [1—4].

Ненадежность экономических прогнозов на основе прежних моделей побуждает к созданию более адекватных моделей, так как практическое значение прогнозов, в особенности среднесрочных и дальних, резко возросло. Прогноз в настоящее время стал одной из актуальнейших задач экономистов и статистиков, как впрочем и социологов, инженеров и т. д.; уже вошло в употребление и название новой отрасли науки «футурология» (соответственно появились и специалисты «футурологи»). Футурология требует коренного улучшения методики экономического прогнозирования на всех уровнях **.

Приведем пример. Комитет экономических советников президента США ежегодно делает экономический прогноз на 1—2 года вперед. Несмотря на короткий срок прогноза, средняя погрешность в оценке валового нацио-

* В данной статье рассматриваются работы, появившиеся в странах капиталистической системы. О работах в СССР рассказывалось в обзоре Н. Е. Кобринского и А. М. Матлина в № 2 нашего журнала за 1967 г. Специализированные обзоры по различным проблемам моделирования будут даваться в последующих номерах.

** Большое внимание уделяется сверхдальним прогнозам (30—50 лет и более), но здесь речь идет не о математических моделях, а о статистической экстраполяции и экспертных оценках, ввиду чего эти исследования, представляющие несомненный интерес, здесь не рассматриваются.