ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Июль—декабрь 1999. Серия 2. Том 6, № 2, 32–41

УДК 517.854.3

ПАКЕТ ПРОГРАММ ЦЕЛОЧИСЛЕННОГО ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ*)

Г. И. Забиняко

Рассматривается пакет программ для решения задач целочисленного и частично целочисленного линейного программирования. Пакет может выполняться на компьютерах IBM PC в среде DOS и на Silicon Graphics в среде IRIX; язык программирования — Фортран-77. Решение задач основано на методе ветвей и границ с односторонним ветвлением. Приведены результаты решения тестовых задач с анализом особенностей вычислительного процесса.

1. Общие сведения о пакете

Предлагаемый пакет программ предназначен для решения задач следующего вида: найти максимум функции $f(x) = (c, x)$ при ограничениях

$$Ax = b, \quad \alpha \leqslant x \leqslant \beta, \quad x_j \text{ — целое}, \ j \in J.$$

Здесь A — матрица размера $m \times n$; J — список номеров целочисленных переменных; $c, x, \alpha, \beta \in R^n$, $b \in R^m$. Пакет рассчитан на задачи, имеющие размерности

$$m \leqslant 32000, \quad |J| \leqslant 32000, \quad n \leqslant 2^{31} - 1.$$

Нецелочисленные переменные x_j, $j \notin J$, могут быть ограниченными только снизу или сверху либо свободными. Любые переменные x_j можно фиксировать, полагая $\alpha_j = \beta_j$. Общие ограничения можно задавать в виде неравенств, не приводя их к равенствам.

Пакет может выполняться на компьютерах IBM PC в среде DOS и на Silicon Graphics в среде IRIX; язык программирования — Фортран-77. Программное обеспечение содержит процедуры численного решения задач и программы обработки данных, которые обеспечивают ввод, логический контроль данных, составление словарей и справочных таблиц, перевод из внешнего MPS-формата [3] во внутренний формат пакета

*) Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (код проекта 99–07–90422).

и формирование выходных форм. Версии пакета для IBM PC и Silicon Graphics отличаются вспомогательными программами вследствие различий в организации памяти. Для пакета IBM PC разработаны процедуры отображения хода решения задачи на дисплей (на Silicon Graphics такой возможности нет).

Входные данные задачи представляются в MPS-формате. Для матриц используется столбцовый разреженный формат, который определяется по аналогии со строчным разреженным форматом [4]. Последний также можно использовать для обращения к программам пакета.

В оценочных задачах линейного программирования (ЛП) матрица B^{-1}, обратная к базисной матрице B, представляется в мультипликативной форме. Столбцы мультипликаторов располагаются в восьмибайтовом поле. Первое восьмибайтовое слово каждого мультипликатора состоит из четырехбайтовых целых переменных I^- и I^+ — указателей на предшествующий и последующий данному мультипликаторы. После I^- и I^+ записывается вещественная восьмибайтовая переменная G — величина, обратная ведущему элементу. Далее следуют двухбайтовые целые: iv — номер ведущего элемента, k — число ненулевых элементов столбца без ведущего элемента, $i_1, \dots, i_k$ — номера ненулевых компонент. Затем размещаются восьмибайтовые вещественные числа $d_{i_1}, \dots, d_{i_k}$ — ненулевые элементы столбца.

Настройка пакета осуществляется с помощью числовых параметров, которые указывают частоту перестроения обратных матриц, задают точность выполнения условий по исходным и двойственным переменным в задачах ЛП, определяют ограничения на выбор ведущих элементов и т. п. Пользователь задает имена двух файлов, в один из которых помещаются статистические данные о задаче и процессе ее решения. Второй файл содержит информацию, необходимую для возобновления решения задачи после того, как она была снята со счета или завершилась по выполнению заданного числа итераций.

2. Процедуры симплекс-метода

Решение оценочных задач ЛП производится с помощью подпрограмм, рассмотренных в [5]. В них реализован ряд алгоритмических приемов, направленных на обеспечение численной устойчивости. Ниже приводятся краткие характеристики основных процедур ЛП.

2.1. Построение начального базиса

В пакете используются два алгоритма построения начального базиса для системы ограничений $Ax = b$, $\alpha \leqslant x \leqslant \beta$. Пусть $b_i \geqslant 0$ при всех i. Стандартный способ формирования начального базиса состоит во введении фиктивных неотрицательных переменных x_{n+i}, $i = 1, \ldots, m$, и присоединении к A единичной матрицы I порядка m. В результате получаем начальный базис $x_{n+i} = b_i$, $i = 1, \ldots, m$, с единичной базисной матрицей.

В рассматриваемых алгоритмах делается попытка уменьшить число фиктивных переменных в начальном базисе за счет проблемных. В первом алгоритме (обозначим его через $B1$) начальная базисная матрица выбирается нижнетреугольной с учетом перестановок строк и столбцов. Для того чтобы j-й столбец матрицы A можно было включить в базис с i-й ведущей строкой, необходимо выполнение условий:

1) $|a_{ij}| \geqslant \delta$ и $|a_{ij}| \geqslant u \cdot \max\limits_{k \in K} |a_{kj}|$, где δ и u — заданные положительные числа, $0 < u \leqslant 1$; K — множество строк, которые еще не использовались для назначения ведущих элементов;

2) значение переменной x_j должно удовлетворять неравенствам $\alpha_j \leqslant x_j \leqslant \beta_j$.

Если не удается найти столбец матрицы A с ведущей строкой i, то вводится фиктивная переменная x_{n+i}.

Во втором алгоритме (обозначим его через $B2$) при включении в базис j-го столбца с ведущей строкой i не требуется выполнения неравенств $\alpha_j \leqslant x_j \leqslant \beta_j$. Здесь в начальном базисе используется следующий вектор b'. Если для проблемной переменной x_j выполняются неравенства $\alpha_j \leqslant x_j \leqslant \beta_j$, то $b'_i = 0$. В противном случае переменная x_j полагается равной α_j или β_j, а компонента b'_i выбирается так, чтобы обеспечить равенство в i-м уравнении. В результате начальный базис представляется системой уравнений $Bx + b'x_0 = b$, где матрица B состоит из $m-1$ столбца и $x_0 = 1$. Базисная матрица $B' = [B|b']$ может отличаться от нижнетреугольной столбцом b'. Решение задачи ЛП с выбором начального базиса по алгоритму $B2$ начинается с устранения недопустимости базисных переменных: $x_j < \alpha_j$ или $x_j > \beta_j$. Для фиктивных переменных и x_0 допустимыми считаются только нулевые значения.

2.2. Корректировка матриц, обратных к базисным

Пусть на очередной итерации симплекс-метода p-й столбец базисной матрицы B заменяется на столбец $a_{.q}$ матрицы A. Базисной становится матрица $\bar{B} = B + (a_{.q} - Be_p)e_p^T$, где e_p — единичный p-й орт в R^m. Матрица $\bar{B}^{-1}$ вычисляется по методу Форреста — Томлина [6], дополненному операциями контроля численной устойчивости. Для вновь создаваемого мультипликатора U_k^{-1} контролируется малость модуля ведущего элемента. Максимальная величина модулей элементов U_k^{-1} сравнивается с максимумом модулей элементов в $\bar{B}$. В результате этих проверок может быть принято решение о внеочередном обращении к процедуре перестроения обратной матрицы.

До обращения к процедуре корректировки, после определения вектора $s = \pm B^{-1}a_{.q}$ и номера ведущей строки p, вычисляется величина $s_{\max} = \max\limits_{i=1,\dots,m} |s_i|$. Если $\frac{s_{\max}}{|s_p|} > M$, где M достаточно большое положительное число, то производится внеочередное обращение к процедуре перестроения.

2.3. Перестроение матриц, обратных к базисным

В алгоритме перестроения обратной матрицы [8] делается попытка минимизировать заполнение ненулевыми элементами мультипликативного представления матрицы $B^{-1} = U^{-1}L^{-1}$, где каждая из матриц U^{-1} и L^{-1} — произведение соответствующих мультипликаторов. Это эвристический метод приведения матрицы B с помощью перестановок строк и столбцов к виду, по возможности близкому к нижнетреугольной форме.

Столбцы, которые после перестановок не имеют ненулевых элементов над главной диагональю, называют *нормальными*, остальные — *столбцами-выступами*. При выборе ведущих элементов нормальные столбцы порождают только L^{-1}-мультипликаторы, а столбцы-выступы — L^{-1}- и U^{-1}-мультипликаторы. Мультипликатор нормального столбца полностью определяется значениями элементов соответствующего столбца исходной матрицы A. Поэтому в поле мультипликаторов запоминаются только номер столбца из A и номер ведущей строки.

В пакете программ реализован алгоритм, в котором наряду с эвристиками [8] применяются операции контроля численной устойчивости при выборе ведущих элементов [1].

2.4. Вычисление шага

Пусть на очередной k-й итерации симплекс-метода определено направление $s = \pm B^{-1}a_{.q}$ возрастания целевой функции. Процедура вычисления шага λ вдоль s основана на алгоритме из [7]. Вначале определяется максимальная величина λ_1 такая, что $\alpha_j - \delta_k \leqslant x_j + \lambda_1 s_j \leqslant \beta_j + \delta_k$, где x_j — базисные переменные, δ_k — малая положительная величина, которая увеличивается на каждой итерации симплекс-метода. При выборе ведущей строки p делается попытка максимизировать величину $|s_p|$, исходя из соотношений $\lambda_2 \leqslant \lambda_1$ и $\alpha_j \leqslant x_j + \lambda_2 s_j \leqslant \beta_j$. В результате $\lambda = \max\{\lambda_2, \tau/|s_p|\}$, где τ — заданная положительная величина, $\tau \ll \delta_0$. На $(k+1)$-й итерации $\delta_{k+1} = \delta_k + \tau$.

После перестроения обратной базисной матрицы может оказаться, что некоторые из базисных переменных x_j не удовлетворяют двусторонним ограничениям. Восстановление соотношений $\alpha_j \leqslant x_j \leqslant \beta_j$ осуществляется с помощью решения задачи ЛП, в целевой функции которой $c_j = 0$

при $\alpha_j \leqslant x_j \leqslant \beta_j$, $c_j = 1$ при $x_j < \alpha_j$ и $c_j = -1$ для $x_j > \beta_j$. Начальное значение λ_1 длины шага вдоль направления s находится как

$$\lambda_1 = \arg\min_{\mu>0} \sum_j \Big(\max\{0,\ \alpha_j - x_j - \mu s_j\} + \max\{0,\ x_j + \mu s_j - \beta_j\} \Big).$$

Выбор ведущей строки p и шага λ производится аналогично рассмотренному выше варианту.

3. Алгоритм целочисленного линейного программирования

В пакете используется алгоритм метода ветвей и границ с односторонним ветвлением. Выбор переменной ветвления основан на анализе штрафных оценок [2]. Пусть $j \in J$. Базисную переменную x_j представим в виде $x_j = \lfloor x_j \rfloor + v_j$, где $\lfloor x_j \rfloor$ — целая часть x_j. Штраф за увеличение x_j на величину $1 - v_j$ обозначим через P_j^+, а за уменьшение x_j на величину v_j — через P_j^-.

Алгоритм.

Шаг 0. Положить $i = 0$, $k = 0$, значение рекорда $r^0 = -\infty$, $\alpha^0 = \alpha$ и $\beta^0 = \beta$.

Шаг 1. Решить текущую задачу ЛП. Пусть x^i ее оптимальное решение и $f^i = f(x^i)$ максимальное значение целевой функции.

а) Если $f^i \leqslant r^i$ или система ограничений несовместна, то присвоить $r^{i+1} = r^i$, $i = i + 1$ и перейти на шаг 3.

б) Пусть $f^i > r^i$. Если решение x^i оценочной задачи не целочисленное, то перейти на шаг 2. Если вектор x^i целочисленный, то присвоить $r^{i+1} = f^i$, $i = i + 1$. Если $f^i = f^0$, то перейти на шаг 5. В противном случае перейти на шаг 3.

Шаг 2. С помощью алгоритма из [2] для всех базисных индексов $j \in J$ вычислить штрафы P_j^+ и P_j^-. Среди них найти минимальный штраф $P_{\min}$.

а) Если $f^i - P_{\min} \leqslant r^i$, то перейти на шаг 3.

б) Если $f^i - P_{\min} > r^i$, то произвести ветвление по базисной переменной x_j, которой соответствует максимальный штраф P_j^+ или P_j^-. Из двух величин P_j^+ и P_j^- выбрать меньшую. Пусть $P_j^- \leqslant P_j^+$. Задать параметр

$$l = \begin{cases} 1, & \text{если } f^i - P_j^+ \leqslant r^i, \\ 0 & \text{в противном случае.} \end{cases}$$

Увеличить $k = k + 1$ и во вспомогательном массиве h осуществить присвоения:

$$h(1,k) = j, \quad h(2,k) = \beta_j^i, \quad h(3,k) = l, \quad h(4,k) = f^i, \quad h(5,k) = P_j^+.$$

Изменить верхнюю границу переменной x_j, положив ее равной $\lfloor x_j^i \rfloor$. Перейти на шаг 1. (Если $P_j^- > P_j^+$, то производятся присвоения:

$$h(1,k) = -j, \quad h(2,k) = \alpha_j^i, \quad h(3,k) = l, \quad h(4,k) = f^i, \quad h(5,k) = P_j^-.$$

Параметр l определяется аналогично приведенному выше. Нижняя граница для переменной x_j полагается равной $\lfloor x_j^i \rfloor + 1$.)

Шаг 3. Если $k = 0$, то перейти на шаг 5. Пусть $k > 0$. Если $h(3,k) = 1$ или $h(4,k) - h(5,k) \leqslant r^i$, то перейти на шаг 4.

Используя массив h, сформировать задачу, альтернативную образованной на шаге 2, б. Присвоить $h(3,k) = 1$ и перейти на шаг 1.

Шаг 4. Установить двусторонние ограничения на переменную x_j, используя значения $h(1,k)$ и $h(2,k)$. Присвоить $k = k - 1$ и перейти на шаг 3.

Шаг 5. Остановиться.

Конец

При вычислении штрафов может встретиться вырожденный случай, когда максимальное значение среди всех P_j^- и P_j^+ равно 0. Тогда для ветвления выбирается переменная x_j, которая имеет значение, наиболее уклоняющееся от целочисленного.

В схемах с односторонним ветвлением можно экономно организовать переход к очередной оценочной задаче. Выделим два случая, которые возникают при формировании задач:

1) переменной x_j предписывается фиксированное значение $x_j = \alpha_j^i = \beta_j^i$;
2) переменная x_j должна изменяться в пределах $\alpha_j^i \leqslant x_j \leqslant \beta_j^i$, где $\alpha_j^i < \beta_j^i$.

В первом случае в базис предшествующей задачи с базисной матрицей B вводится фиктивная переменная x_ν вместо переменной x_j. Для этого вычисляется вектор $z^T = e_p^T B^{-1}$, где e_p — p-й орт в R^m, а p — номер переменной x_j в базисе. Находится номер $k = \arg \max\limits_{i=1,\dots,m} |z_i|$ и полагается $\nu = n + k$. Переменной x_ν ставится в соответствие k-й орт из R^m. Затем обычным образом корректируется матрица B^{-1} с учетом замены в базисе переменной x_j на x_ν и вычисляются значения базисных переменных в новом базисе.

Во втором случае нет необходимости в корректировке матрицы B^{-1}. Решение очередной оценочной задачи всегда начинается с проверки выполнения ограничений $\alpha_j^i \leqslant x_j \leqslant \beta_j^i$ для базисных переменных.

С помощью управляющих параметров пакета можно модифицировать схему решения задачи. Например, для сокращения перебора пользователь может задать оценку $\tilde{f}$ значения целевой функции. Если оптимум f^i оценочной задачи оказывается меньше $\tilde{f}$, то она далее не применяется в качестве родительской. Для получения приближенных решений предусмотрен входной параметр δ_f. Процесс решения задачи завершается, если очередной рекорд r^i удовлетворяет неравенству $\frac{f^0-r^i}{|f^0|} \leqslant \delta_f$.

В задачах, содержащих общие ограничения только типа неравенств, можно применить локальный поиск. После решения исходной задачи ЛП без учета целочисленности переменных оптимальные значения x_j заменяются на $\lfloor x_j + 0,5 \rfloor$ для $j \in J$. Полученная точка x является начальной для поиска допустимого решения. После нахождения допустимой точки, удовлетворяющей требованиям целочисленности, покоординатным поиском строится локально оптимальное решение.

4. Решение тестовых задач

Для испытания пакета программ использовались задачи, полученные на электронных носителях в MPS-формате. В табл. 1 приведены их характеристики.

Таблица 1

N	Название задачи	m	m_e	n	n_i	n_b	nz
1	*air*01	23	23	771	771	771	4215
2	*air*02	50	50	6774	6774	6774	61555
3	*air*03	124	124	10757	10757	10757	91028
4	*air*05	310	310	6200	6200	6200	32740
5	*boeing*1	351	98	473	150	16	3574
6	*boeing*2	166	23	162	50	–	1215
7	*cracpb*1	143	90	573	572	572	4159
8	*dsbmio*	1182	343	1928	183	160	7417
9	*gen*	780	150	870	150	144	2592
10	*modglob*	291	95	422	98	98	968

Примечание: N — номер задачи; m — число ограничений, среди которых m_e ограничений являются равенствами; n — число переменных, среди которых n_i переменных являются целочисленными и n_b переменных — булевыми; nz — общее число ненулевых элементов в матрице A.

В задачах 1–4 требуется найти оптимальное расписание полетов (G. Astfalk). Задачи 5 и 6 относятся к авиастроению (N. Gould). Область приложений в задаче 7 неизвестна (H. Growder, E. A. Boyd). В задаче 8 оптимизируется добыча газа, его глубокая переработка и распределение (J. J. H. Forrest). Задача 9 состоит в определении расписания работы технических устройств (L. A. Wolsey, M. W. P. Savelsbergh). Задача 10 относится к проектированию системы отопления (L. A. Wolsey, M. W. P. Savelsbergh).

В табл. 2 приведены статистические данные по решению задач.

Таблица 2

N	it	it_0	it_s	k	k_s	k_i	$P1$	$P2$	t
Начальный базис $B1$									
1	75	42	75	1	2	1	1	–	0,84
2	798	262	564	5	7	1	4	–	26,2
3	1319	843	1319	1	2	1	1	–	95,7
4*	10^6	10076	431822	69	102	2	2	–	20412,0
5*	10^6	879	664568	185	43879	36	22452	351	3237,0
6	180014	206	104713	48	11683	7	4754	53	358,9
7	2520	766	2520	56	57	1	–	–	11,28
8	29107	1519	29107	46	140	11	32	–	616,3
9	113226	2152	55178	34	670	6	79	–	1088,0
10*	10^6	170	990149	96	30613	115	11169	145	1983,0
Начальный базис $B2$									
1	76	60	76	1	2	1	1	–	0,68
2	51449	230	51262	42	345	2	28	1	698,8
3	1750	1076	1750	1	2	1	1	–	43,38
4*	10^6	8176	386851	68	69	1	–	–	22243,0
5*	10^6	1069	707371	189	58126	14	29562	342	3534,0
6	285923	159	208334	50	17836	6	6951	62	561,9
7	3854	678	3854	57	77	2	3	–	19,36
8	14097	2949	14097	48	503	15	43	–	4783,0
9	148803	1945	71621	38	1009	11	92	–	1390,0
10*	10^6	168	170932	89	39656	65	16971	41	2150,0

Примечание: N — номер задачи; it — общее число итераций; it_0 — номер итерации, на которой получено решение задачи ЛП без учета целочисленности; it_s — номер итерации, на которой получено оптимальное целочисленное решение; k — максимальная глубина дерева вариантов; k_s — общее число сформированных задач ЛП; k_i — количество найденных допустимых целочисленных решений; $P1$ — число выполнений условия $f^i - P_j^+ \leqslant r^i$ или $f^i - P_j^- \leqslant r^i$; $P2$ — число выполнений условия $f^i - P_{\min} \leqslant r^i$; t — время решения задачи в секундах на Silicon Graphics (тактовая частота 75 МГц). Звездочкой отмечены задачи, решение которых прекратилось после выполнения заданного числа итераций.

Различия в табл. 2 для разных начальных базисов объясняются в значительной степени неединственностью решений исходных задач ЛП без учета целочисленности переменных, что видно на примере задач 7 и 8, в которых начальная оценка f^0 равна оптимуму. В задаче 7 оптимальное решение без учета целочисленности имеет нулевые относительные оценки для 430 небазисных переменных.

При решении задачи 8 встречаются плохо обусловленные базисные матрицы. Некоторый стандартный набор значений управляющих параметров, который использовался при решении задач, не позволил получить ни одного решения, удовлетворяющего требованиям целочисленности. Решение было получено при специальном выборе управляющих параметров, обеспечивших более высокую точность в оценочных задачах.

Таблица 3

N	m	Заполненность A	Заполненность B^{-1}			
			Начальный базис $B1$		Начальный базис $B2$	
		$\frac{nz}{m\times n}$	$\frac{nz_1}{m\times m}$	$\frac{nz_0}{m\times m}$	$\frac{nz_1}{m\times m}$	$\frac{nz_0}{m\times m}$
1	23	0,24	1,70	0,20	2,05	0,21
2	50	0,18	1,39	0,21	1,22	0,23
3	124	0,07	0,37	0,09	0,50	0,11
4	310	0,02	0,28	0,15	0,29	0,15
5	351	0,02	0,06	0,02	0,06	0,02
6	166	0,04	0,15	0,05	0,15	0,04
7	143	0,05	0,50	0,21	0,46	0,20
8	1182	0,003	0,01	0,005	0,01	0,007
9	780	0,004	0,04	0,01	0,04	0,01
10	291	0,008	0,04	0,007	0,04	0,007

Примечание: N — номер задачи; m — число строк в матрице A; nz — число ненулевых элементов в A; nz_1 — максимальное количество восьмибайтовых слов, занятых мультипликаторами; nz_0 — то же после перестроения матрицы B^{-1}.

Кроме приведенных задач рассматривались еще две: $air04$ и $air06$, имеющие большую размерность, чем $air05$. Решение этих задач не удалось получить с приемлемыми затратами машинного времени. Их особенность состоит в вырожденности базисных решений. В оптимальном решении без требований целочисленности и на промежуточных итерациях несколько сот базисных переменных имеют нулевые значения (в этих задачах $\alpha_j = 0$ при всех j).

В табл. 3 приведены данные, характеризующие максимальное заполнение ненулевыми элементами мультипликативного представления матрицы B^{-1} в течение всего процесса решения, при корректировках на итерациях и после ее перестроения. Перестроение производилось после выполнения 50 корректировок обратных матриц. В процессе решения некоторых задач осуществлялись внеочередные вызовы процедуры перестроения вследствие ошибок вычислений. Максимальное число таких вызовов, равное 171, наблюдалось при решении задачи 8 с начальным базисом $B2$.

ЛИТЕРАТУРА

1. **Забиняко Г. И.** Алгоритм повторения симплекс-метода // Тр. ИВМиМГ. Сер. Системное моделирование. Новосибирск, 1998. Вып. 5 (23). С. 59–73.
2. **Ковалев М. М.** Дискретная оптимизация (целочисленное программирование). Минск: Изд-во Белорус. ун-та, 1977.
3. **Муртаф Б.** Современное линейное программирование: Теория и практика. М.: Мир, 1984.
4. **Писсанецки С.** Технология разреженных матриц. М.: Мир, 1988.
5. **Программы линейного программирования ЛП-ВЦ** / Забиняко Г. И., Котельников Е. А., Кобкова Т. М., Рожин В. Е. Отчет ВЦ СО РАН. Новосибирск, 1995. ГР № 01.9.30001317. Инв. № 02.9.50003757.
6. **Forrest J. J. H., Tomlin J. A.** Updated triangular factors of the basis to maintain sparsity in the product form simplex method // Math. Programming. 1972. V. 2, N 3. P. 263–278.
7. **Gill P. E., Murray W., Saunders M. A., Wright M. A.** A practical anti-cycling procedure for linearly constrained optimization // Math. Programming. Ser. B. 1989. V. 45, N 3. P. 437–474.
8. **Hellerman E., Rarick D.** Reinversion with the preassigned pivot procedure // Math. Programming. 1971. V. 1, N 2. P. 195–216.

Адрес автора:

ИВМиМГ СО РАН,
пр. Академика Лаврентьева, 6,
630090 Новосибирск, Россия.
E-mail: zabin@rav.sscc.ru

Статья поступила
5 апреля 1999 г.