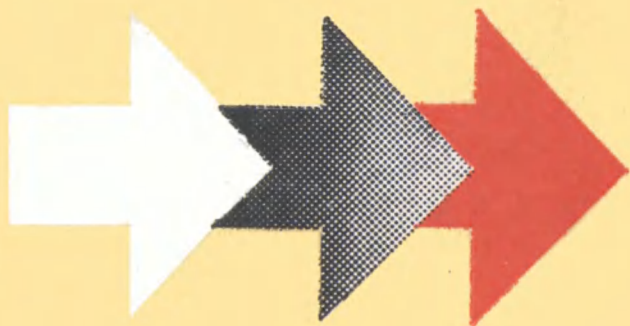


А.А.Павлов

Алгоритмическое обеспечение

Алгоритмы
решения задач
линейного,
сепарабельного,
нелинейного
дискретного
программирования
с блочной
структурой
ограничений

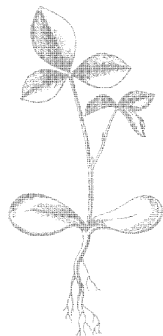
СЛОЖНЫХ СИСТЕМ УПРАВЛЕНИЯ



А.А.Павлов

Алгоритмическое
обеспечение
**СЛОЖНЫХ
СИСТЕМ
УПРАВЛЕНИЯ**

Киев
Головное издательство
издательского объединения
„Выща школа”
1989



Scan AAW

ББК 32.973–01

П12

УДК 681.5

Алгоритмическое обеспечение сложных систем управления/А. А. Павлов. – К.: Выща шк. Головное изд-во, 1989. – 166 с. – ISBN 5–11–000002–6.

В монографии описаны алгоритмы решения задач линейного, сепарабельного, нелинейного дискретного программирования с блочной структурой ограничений, имеющие псевдополиномиальную оценку числа вычислений. Описаны классы прикладных задач дискретного программирования. Изложены новые возможности использования моделей линейного целочисленного программирования в задачах проектирования сложных систем управления.

Предназначена для научных и инженерно-технических работников, преподавателей, аспирантов, студентов.

Ил. 1. Табл. 5. Библиогр.: 19 назв.

Рецензент: д-р техн. наук, проф. С. Д. Бушув (Киевский инженерно-строительный институт)

Редакция литературы по информатике и автоматике
Редактор Л. И. Мубаракшина

п 160202000–025 133–89
M211(04)–89

ISBN 5–11–000002–6

© Издательское объединение
„Выща школа”, 1989

Введение	5
Глава 1. Линейное целочисленное программирование	7
1. Задача линейного целочисленного программирования с фиксированным числом ограничений.	7
2. Сведение задачи линейного целочисленного программирования к задаче „о ранце“	26
3. Исследование задачи „о ранце“	42
4. Алгоритм решения задачи линейного целочисленного программирования, использующий информацию, содержащуюся в эквивалентной задаче „о ранце“	60
Глава 2. Сепарабельное целочисленное программирование	94
1. Оптимизация сложных систем.	94
2. Алгоритм решения одного класса задач сепарабельного целочисленного программирования с частично блочной структурой ограничений, имеющий полиномиальную оценку сложности относительно числа переменных задачи	96
3. Декомпозиционные алгоритмы задачи блочного линейного целочисленного программирования общего вида.	107
4. Алгоритмы решения одного класса задач целочисленного сепарабельного программирования с „лестничной“ структурой матрицы ограничений и полиномиальной оценкой сложности относительно числа переменных	114
5. Алгоритмы решения одного класса задач целочисленного сепарабельного программирования с частично заполненной матрицей ограничений, имеющих полиномиальную оценку сложности относительно числа переменных	129
6. Алгоритм решения одного класса задач нелинейного целочисленного программирования с частично-блочной структурой ограничений, имеющей полиномиальную оценку сложности относительно числа переменных.	137
7. Использование алгоритма последовательного анализа и отсева вариантов без пошагового конструирования решения.	140

Глава 3. Целочисленные модели в сложных системах управления	143
1. Анализ эффективности решения NP -полных задач	143
2. Области применения дискретных моделей в сложных системах управления	152
3. Распределение ресурсов в задаче планирования выполнения частично-упорядоченного множества работ	159
4. Управление дискретными динамическими системами	162
<i>Список использованной литературы</i>	166

В ускорении научно-технического прогресса одна из основных задач состоит во внедрении вычислительной техники в различные области управления народным хозяйством, в создании новых высокопроизводительных средств обработки информации и систем управления.

Системы управления относятся к классу больших или сложных систем, имеющих свойства:

- функциональные связи в них описываются набором большого числа уравнений и неравенств (алгебраических, дифференциальных, линейных, нелинейных) с относительно большим числом непрерывных и дискретных переменных;

- структура функциональных связей имеет специфику, указывающую на возможность более полного расчленения системы уравнений и неравенств на подсистемы меньшей размерности.

Основными проблемами теории оптимизации сложных систем управления оказываются проблемы создания методов решения задач математического программирования большой размерности, использующих, в частности, специфику структуры связей и критериальных функций. Возможность решения той или иной задачи зависит, как известно, от возможностей используемой вычислительной техники. При этом, несмотря на резкое повышение производительности современных ЭВМ, требование повышения вычислительной эффективности к создаваемым алгоритмам остается в силе и даже становится более острым вследствие удорожания ЭВМ, а также необходимости проводить вычисления в реальном масштабе времени (проблемы функционирования систем управления). Следовательно, практический интерес к разработке новых методов, позволяющих ускорить решение указанных задач, ни в коей мере не уменьшается.

Настоящая монография является обобщением последних научных результатов, полученных автором в области создания эффективных алгоритмов решения задач целочисленного программирования и их использования в сложных системах управления.

В первой главе исследуется задача линейного целочисленного программирования (ЗЛЦП). Показано, что задача групповой минимизации, построенная по ЗЛЦП с неотрицательными переменными, фиксированным числом ограничений и не экспоненциально большими по модулю коэффициентами, может быть решена алгоритмом X_u с числом операций, ограниченных полиномом от

числа переменных задач. ЗЛЦП с фиксированным числом связанных ограничений (ограничений, в состав которых входят не менее двух переменных) и неэкспоненциально большими по модулю коэффициентами может быть сведена к ЗЛЦП с одним связным ограничением, которая методом динамического программирования может быть решена с вычислительной сложностью, ограниченной полиномом фиксированной степени от числа переменных. Предложена эффективная вычислительная схема решения исходной ЗЛЦП, использующая информацию, содержащуюся в эквивалентной ЗЛЦП с одним связным ограничением (определение направления эффективного ветвления, формулировка эффективных правил отсечения, оценка отклонения по показателю качества допустимых решений от оптимального). В разработке вычислительной схемы принимали участие Б. Б. Мисюра, О. В. Щербатенко.

Во второй главе приведены классы задач сепарабельного целочисленного программирования с частично-блочной структурой, „лестничной” структурой матрицы ограничений, частично-заполненной матрицей ограничений специального вида, для которых построены специальные схемы последовательного конструирования вариантов с полиномиальной относительно числа переменных сложностью.

В третьей главе приведен краткий обзор теории NP -полноты, показано, что все рассмотренные в первой и второй главе задачи комбинаторной оптимизации принадлежат классу P , проанализирована вычислительная эффективность предложенных алгоритмов их решения, сформулирован ряд новых сложных систем управления, формальное описание которых сводится к рассмотренным в монографии моделям.

Глава 1. ЛИНЕЙНОЕ ЦЕЛОЧИСЛЕННОЕ ПРОГРАММИРОВАНИЕ

1. ЗАДАЧА ЛИНЕЙНОГО ЦЕЛОЧИСЛЕННОГО ПРОГРАММИРОВАНИЯ С ФИКСИРОВАННЫМ ЧИСЛОМ ОГРАНИЧЕНИЙ

Рассмотрим задачу линейного целочисленного программирования (ЗЛЦП) общего вида. Для получения ее оптимального решения используем задачу групповой минимизации [18]. Покажем, что точное решение ЗЛЦП общего вида с неотрицательными переменными и фиксированным числом ограничений (при увеличении числа неотрицательных переменных число ограничений не меняется) может быть получено по специальным образом сконструированной задаче групповой минимизации. Пусть выполняются следующие условия: величины коэффициентов матрицы и столбца ограничений исходной ЗЛЦП по модулю ограничены полиномом фиксированной степени от числа переменных n , полиномом фиксированной степени от n ограничена величина положительного числа k , где k определяет гиперплоскость $\sum_{i=1}^n x_i = k$, ограничивающую область допустимых целочисленных значений исходной ЗЛЦП. Тогда точный алгоритм решения задачи групповой минимизации, а следовательно, и исходной ЗЛЦП имеет верхнюю оценку числа операций, представляющую собой полином фиксированной степени от n . Этот результат получен на основании двух утверждений, изложенных в [8]. Для полноты и связности подачи материала доказательство этих утверждений приведем полностью.

Утверждение 1. Пусть дана ЗЛЦП

$$\begin{aligned} & \max_x \sum_{j=1}^n c_j x_j, \\ & \sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i=1, \dots, m; \quad m \gg n, \quad (1.1) \\ & b_i \geq 0, \quad i=1, \dots, m; \quad \forall_j x_j \equiv 0 \pmod{1}, \end{aligned}$$

где a_{ij}, b_i – целые числа.

Допустим, что решена задача линейного программирования (ЗЛП), полученная из ЗЛЦП (1.1) снятием требования целочисленности переменных. Рассмотрим случай, когда оптимальное решение данной ЗЛП невырождено. Определим такие n ограничений из системы ограничений (1.1), которым оптимальная точка удовлетворяет как строгим равенствам. Множество таких ограничений обозначим через J . Сформулируем ЗЛЦП первого уровня:

$$\begin{aligned} \max_{\hat{x}} \sum_{j=1}^n C_j x_j, \\ \sum_{j=1}^n a_{ij} x_j \leq b_i, \quad \hat{i} \in J, \end{aligned} \quad (1.2)$$

где x_j – целые числа.

Тогда решения ЗЛЦП (1.1) и (1.2), полученные в результате применения асимптотического алгоритма, совпадают.

Доказательство. Запишем ограничения ЗЛЦП (1.1) в канонической форме следующим образом:

$$\begin{pmatrix} E & d \\ 0 & A \end{pmatrix} \begin{pmatrix} y \\ x \end{pmatrix} + \begin{pmatrix} 0 \\ \bar{E} \end{pmatrix} z = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}, \quad (1.3)$$

где $A = (a_{ij})$, $i \in J$, $j = 1, \dots, n$; единичная матрица E имеет размерность $(m-n) \times (m-n)$; единичная матрица \bar{E} имеет размерность $n \times n$; y, z – векторы свободных переменных задачи. Ограничения (1.3) могут отличаться от ограничений (1.2) порядком строк. Базис оптимальной вершины ЗЛП, полученной из ЗЛЦП (1.1), равен

$$B = \begin{pmatrix} E & d \\ 0 & A \end{pmatrix} \quad (1.4)$$

Тогда по формуле Фробениуса получаем

$$B^{-1} = \begin{pmatrix} E & -dA^{-1} \\ 0 & A^{-1} \end{pmatrix}.$$

Как видно из матрицы B (1.4), групповые соотношения задач групповой минимизации, соответствующие ЗЛЦП (1.1) и (1.2), одинаковы. Действительно, пусть P и Q такие унимодулярные матрицы, что

$$PBQ = S,$$

где S – нормальная форма Смита.

Запишем матрицу P в виде

$$P = \begin{pmatrix} E & 0 \\ 0 & \tilde{P} \end{pmatrix},$$

где \tilde{P} – унимодулярная матрица размерности $(n \times n)$ такая, что

$$\tilde{P}A\tilde{Q} = \tilde{S} = \begin{pmatrix} \varepsilon_1 & & 0 \\ & \dots & \\ 0 & & \varepsilon_n \end{pmatrix}, \quad \prod_{i=1}^n \varepsilon_i = \det A.$$

Матрица Q имеет следующую структуру:

$$Q = \begin{pmatrix} E & -d\tilde{Q} \\ 0 & \tilde{Q} \end{pmatrix}.$$

Таким образом, матрицу S представим в виде

$$S = \left(\underbrace{1 \dots 1}_{m-n} \quad 0 \right) \quad (1.5)$$

Справедливость формулы (1.5) можно проверить непосредственным перемножением матриц PBQ .

Матричное равенство (1.3) запишем в несколько другой форме:

$$Bx' + Nz = b,$$

$$x' = \begin{pmatrix} y \\ x \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix},$$

или

$$PBQQ^{-1}x' + PNz = Pb, \quad (1.6)$$

$$Sy' + PNz = Pb; \quad y' = Q^{-1}x'.$$

Следовательно, y образов столбцов матрицы PN принадлежит фактор-группе $\{A'\}/\{B\}$, где A' – матрица линейных ограничений задачи (1.1) в канонической форме, первые $m-n$ компонент равны 0. Первые $m-n$ компонент образа столбца Pb также равны 0.

Равенство, аналогичное (1.6) для задачи (1.2) имеет вид:

$$\tilde{S}y'' + \tilde{P}\tilde{E}z = \tilde{P}b_2.$$

Остальные n компонент образов столбцов матрицы PN и столбца Pb совпадают с компонентами образов столбцов матрицы $\tilde{P}\tilde{E}$, а также столбца $\tilde{P}b_2$ соответственно. Поскольку нулевые компоненты элементов фактор-группы $\{A'\}/\{B\}$ на задачу групповой минимизации не влияют, образы фактор-групп $\{A'\}/\{B\}$ и $\{\tilde{A}'\}/\{\tilde{A}\}$ совпадают, где \tilde{A}' – каноническая матрица ограничений ЗЛЦП (1.2).

Рассмотрим векторы коэффициентов целевых функций обеих задач групповой минимизации.

Вектор коэффициентов ЗЛЦП (1.2) представим так:

$$c^T A^{-1}.$$

Для задачи ЛЦП (1.1)

$$(0^T, c^T)(B^{-1}) \begin{pmatrix} 0 \\ \tilde{E} \end{pmatrix}$$

или

$$(0^T, c^T) \begin{pmatrix} E & -dA^{-1} \\ 0 & A^{-1} \end{pmatrix} \begin{pmatrix} 0 \\ \tilde{E} \end{pmatrix} = c^T A^{-1} \tilde{E} = c^T A^{-1}.$$

Таким образом, обе задачи групповой минимизации определяют одно и то же значение вектора свободных переменных Z .

Отсюда обе задачи дадут одно и то же значение вектора x . Действительно,

$$\begin{pmatrix} y \\ x \end{pmatrix} = \begin{pmatrix} E & -dA^{-1} \\ 0 & A^{-1} \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} - \begin{pmatrix} E & -dA^{-1} \\ 0 & A^{-1} \end{pmatrix} \begin{pmatrix} 0 \\ \bar{E} \end{pmatrix} (z) = \begin{pmatrix} b_1 - dA^{-1}b_2 + dA^{-1}z \\ A^{-1}b_2 - A^{-1}z \end{pmatrix}.$$

Решив ЗЛЦП (1.2), получим

$$x = A^{-1}b_2 - A^{-1}z.$$

Таким образом доказано, что при использовании в асимптотическом алгоритме фактор-группы $\{A^i\}/\{B\}$ у ЗЛЦП (1.1) и построенной по ней ЗЛЦП (1.2) одно и то же решение. Утверждение 1 считается окончательно доказанным, если показать, что тот же результат получится при использовании в асимптотическом алгоритме группы G , порожденной дробными частями матрицы.

$$\begin{pmatrix} B^{-1} \\ \bar{E} \end{pmatrix} \begin{pmatrix} 0 \\ \bar{E} \end{pmatrix} \text{ и } (A^{-1})(E)$$

соответственно. Это следует из следующей цепочки матричных равенств:

$$\begin{aligned} B^{-1} &= \begin{pmatrix} E & -dA^{-1} \\ 0 & A^{-1} \end{pmatrix}; \\ \begin{pmatrix} E & -dA^{-1} \\ 0 & A^{-1} \end{pmatrix} \begin{pmatrix} 0 \\ \bar{E} \end{pmatrix} z &\equiv \begin{pmatrix} E & -dA^{-1} \\ 0 & A^{-1} \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \pmod{1}; \\ \begin{pmatrix} -dA^{-1}z \\ A^{-1}z \end{pmatrix} &\equiv \begin{pmatrix} b_1 - dA^{-1}b_2 \\ A^{-1}b_2 \end{pmatrix} \pmod{1}. \end{aligned}$$

Если в процессе решения задачи групповой минимизации получим, что

$$A^{-1}z \equiv A^{-1}b_2 \pmod{1},$$

то

$$-dA^{-1}z \equiv b_1 - dA^{-1}b_2 \pmod{1}.$$

Утверждение 1 доказано.

Следствие. Если в ЗЛЦП вида

$$\begin{aligned} \max C^T x; \\ Ax \leq b, \quad \forall x_j \equiv 0 \pmod{1} \end{aligned} \quad (1.7)$$

при выполнении условия $C^T A^{-1} > 0$ матрица A является квадратной, то решение задачи групповой минимизации, построенной по ЗЛЦП (1.7), дает ее оптимальное решение. ЗЛЦП (1.1) может очевидным образом быть сведена к ЗЛЦП с неотрицательными переменными. Действительно, для этого каждую закононеопределенную переменную нужно заменить разностью двух неотрицательных,

а каждую отрицательную переменную заменить соответствующим образом на обратную.

Утверждение 2. Рассмотрим ЗЛЦП вида

$$\max \sum_{j=1}^n c_j x_j, \quad (1.8)$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = \overline{1, m}, \quad \forall x_i \geq 0, \quad x_i \equiv 0 \pmod{1},$$

где a_{ij}, b_i – целые числа.

Тогда ЗЛЦП (1.8) может быть сведена к ЗЛЦП с квадратной матрицей ограничений неравенств (1.7), оптимальное решение которой совпадает с оптимальным решением ЗЛЦП (1.8).

Доказательство. Рассмотрим ЗЛЦП (1.8).

Систему ограничений ЗЛЦП (1.8) дополним вспомогательным ограничением

$$\sum_{j=1}^n x_j \leq k,$$

где k – некоторая константа, определяемая практическим содержанием решаемой задачи.

Рассмотрим ЗЛЦП с квадратной матрицей ограничений неравенств:

$$\begin{aligned} \max Z(x) = \max \sum_{j=1}^n c_j x_j + M_0 x_{n+1} + M \sum_{j=n+2}^{n+m+1} x_j; \\ x_j \geq 0, \quad j = \overline{1, n}; \\ \sum_{j=1}^n x_j + p_0 x_{n+1} \leq k; \end{aligned} \quad (1.9)$$

$$\begin{aligned} \sum_{j=1}^n a_{ij} x_j - f_i x_{n+1} + p_i x_{n+1+i} \leq b_i, \quad i = \overline{1, m}; \\ x_j \equiv 0 \pmod{1}, \quad j = \overline{1, n+m+1}, \end{aligned}$$

где $x_{n+i}, i = \overline{1, m+1}$ – дополнительные переменные, а коэффициенты при них выбирают из условий:

$$p_0 > k, \quad p_i > b_i - s_i k, \quad i = 1, \dots, m; \quad f_i > b_i - s_i (k + p_0);$$

$$i = 1, \dots, m; \quad M > (U-L)k, \quad M_0 > U p_0;$$

$$U = \max(\max_{1 \leq j \leq n} c_j, 0);$$

$$L = \min(\min_{1 \leq j \leq n} c_j, 0);$$

$$s_i = \min(\min_{1 \leq j \leq n} a_{ij}, 0), \quad i = 1, \dots, m.$$

Покажем, что для любого допустимого целочисленного решения $x = (x_j)_{j=1}^{n+m+1}$ ЗЛЦП (1.9), при котором $x_{n+1} = 0$ справедливы условия $x_{n+1+i} \leq 0, i = 1, \dots, m$.

Действительно, это утверждение вытекает из цепочки неравенств, справедливой для любого $i, i = \overline{1, \dots, m}$,

$$x_{n+1+i} \leq \frac{b_i - \sum_{j=1}^n a_{ij} x_j}{p_i} \leq \frac{b_i - \min_{x \in D_1} \sum_{j=1}^n a_{ij} x_j}{p_i} = \frac{b_i - s_i k}{p_i} < 1,$$

где область D_1 представляется следующим образом:

$$x_i \geq 0, i = \overline{1, n}; \sum_{j=1}^n x_j \leq k.$$

Аналогично можно показать, что для любого допустимого целочисленного решения $x = (x_j)_{1}^{n+m+1}$ ЗЛЦП (1.9), при котором $x_{n+1} \neq 0$, справедливы условия

$$x_{n+1+i} < 0, i = \overline{1, m}.$$

Действительно, в силу выбора коэффициента p_0 при любом допустимом целочисленном решении ЗЛЦП (1.9) $x_{n+1} \leq 0$. Для любого $i (i = \overline{1, m})$ справедлива цепочка неравенств:

$$x_{n+1+i} \leq \frac{b_i - \sum_{j=1}^n a_{ij} x_j + f_i x_{n+1}}{p_i} < \frac{b_i - s_i (k - p_0 x_{n+1}) + f_i x_{n+1}}{p_i}.$$

Из данной цепочки неравенств, а также из условий

$$f_i > b_i - s_i (k + p_0); x_{n+1} < 0$$

следует, что $x_{n+1+i} < 0$ для $i = \overline{1, m}$.

Из приведенных выше результатов вытекает, что для всякого решения $x^* = (x_j^*)_{1}^{n+m+1}$, удовлетворяющего системе ограничений ЗЛЦП (1.9), при котором

$$\sum_{j=1}^n c_j x_j^* + M_0 x_{n+1}^* + M \sum_{i=1}^m x_{n+1+i}^* \geq Lk,$$

справедливы условия

$$x_j^* = 0, j = \overline{n+1, n+m+1}.$$

Докажем справедливость данного утверждения методом от противного, рассмотрев два случая.

Случай 1. $x_{n+1} = 0$. Пусть некоторые из $x_{n+1+i} (i = \overline{1, m})$ не равны нулю. Множество индексов таких переменных обозначим через J . В данном случае максимально возможное значение целевой функции ЗЛЦП (1.9) удовлетворяет условию

$$z(x^*) \leq Uk + M \sum_{(n+1+i) \in J} x_{n+1+i}^*,$$

но $x_{n+1+i}^* \leq 0$ для $(n+1+i)$. Поэтому справедливы неравенства

$$Uk + M \sum_{(n+1+i) \in J} x_{n+1+i}^* < Uk + (U-L)k \sum_{(n+1+i) \in J} x_{n+1+i}^* \leq Lk.$$

Из приведенной цепочки неравенств следует справедливость рассматриваемого утверждения в случае 1.

ной, то оптимальное решение задачи групповой минимизации, построенной по ЗЛЦП (1.9), совпадает с ее оптимальным решением.

Запишем ЗЛЦП (1.9) в канонической форме:

$$\begin{aligned} \max c^T x; \\ Ax + Ny = b; \\ y_j \geq 0; \quad j = \overline{1, n+m+1}, \end{aligned}$$

где y – целочисленный вектор, $N = E$. Выразим x через y

$$x = A^{-1}b - A^{-1}Ny,$$

получим следующую задачу:

$$\begin{aligned} \max_{x, y} (c^T A^{-1}b - c^T A^{-1}Ny); \\ x + A^{-1}Ny = A^{-1}b, \end{aligned}$$

где x, y – целые числа, $y \geq 0$.

Дробные части вектор-столбцов матрицы $A^{-1}N$ являются элементами абелевой группы G относительно операции алгебраического сложения по модулю 1 порядка

$$D = |\det A| = \prod_{i=0}^m p_i. \quad (1.16)$$

Данная группа изоморфна фактор-модулю [18]

$$\{A'\} / \{A\}, \quad \text{где } A' = (A, N).$$

Для построения задачи групповой минимизации необходимо определить образы дробных частей вектор-столбцов матрицы $A^{-1}N$, для чего по матрице A построим нормальную форму Смита $S = PAQ$

$$S = \begin{pmatrix} \varepsilon_1 & & 0 \\ & \varepsilon_2 & \\ 0 & & \varepsilon_{n+m+1} \end{pmatrix}; \quad \det S = \det A = \prod_{i=1}^{n+m+1} \varepsilon_i,$$

где P и Q – унимодулярные целочисленные матрицы. В данном случае $S = PA$, так как A – нижняя треугольная матрица. В результате получим

$$S = \begin{pmatrix} -1 & \dots & -1 & & 0 \\ & & & p_0 & \\ 0 & & & p_1 & \dots & p_m \end{pmatrix}$$

$$\det |S| = \det |A| = \prod_{i=0}^m p_i.$$

Поскольку увеличение коэффициентов p_i ($i = \overline{1, m}$) и p_0 при-

водит к увеличению порядка группы G (1.16) и, как будет показано ниже, к увеличению числа операций в алгоритме групповой минимизации, то ρ_i ($i = \overline{1, m}$) и ρ_0 следует выбирать наименьшими из допустимых. Таким образом, дополним (1.10) условием

$$\rho_0 = k+1; \rho_i = b_i - S_i k + 1, \quad i = \overline{1, m}. \quad (1.17)$$

Рассмотрим сведение матрицы A к нормальной форме Смита. В силу частного вида коэффициентов матрицы A матрица P является произведением матриц вида

$$T_{ij} = \left(\begin{array}{cccc} 1 & \cdot & & 0 \\ 0 & \cdot & \cdot & \\ \vdots & & & \\ 0 & \dots & 0 & 1 \cdot \\ \vdots & & & \\ 0 & a_{ij} & \cdot & 0 \\ \vdots & \vdots & \cdot & \\ 0 & \cdot & \cdot & \cdot \\ & & & 0 \cdot \cdot \cdot \cdot 0 \cdot 1 \end{array} \right)_{n+1+i}, \quad i = \overline{1, m}; j = \overline{1, n},$$

коэффициент a_{ij} стоит в $n+1+i$ -й строке и j -м столбце, другие элементы равны 0, кроме единиц на главной диагонали,

$$T'_j = \left(\begin{array}{cccc} 1 & & & 0 \\ 0 & 1 & \cdot & \\ \vdots & & & \\ 0 & \dots & 0 & 1 \\ \vdots & & & \\ 0 & \cdot & \cdot & \cdot \\ & & & 0 \cdot 1 \end{array} \right), \quad j = \overline{1, n},$$

единицы в матрицах T'_j стоят на главной диагонали и в $n+1$ -й строке в j -м столбце

$$T''_i = \left(\begin{array}{cccc} & & n+1 & \\ 1 & \cdot & \cdot & 0 \\ 0 & \cdot & \cdot & \\ \cdot & \cdot & 1 & \\ \cdot & \cdot & 0 & \\ \cdot & \cdot & \cdot & \\ \cdot & \cdot & \cdot & \\ 0 & \dots & 0 & \cdot \cdot \cdot \cdot 0 \cdot 1 \end{array} \right)_{n+1+i}, \quad i = \overline{1, m},$$

натуральное число n_i в матрице T''_i стоит в $n+1$ -м столбце

в $n+1+i$ -й строке. Числа $n_i, i = \overline{1, m}$ выбираются из условий (1.10), (1.11) и (1.18)

$$f_i = n_i p_0 . \tag{1.18}$$

Суммируя условия (1.10), (1.11), (1.17), (1.18), получим

$$\begin{aligned} n_i(k+1) &> b_i - s_i(k+(k+1)) ; \\ n_i &= [(b_i - s_i(2k+1))/(k+1)] + 1 , \\ i &= \overline{1, m} . \end{aligned} \tag{1.19}$$

Для полного сведения матрицы A к матрице S необходимо $(m+1)n+m$ преобразований типа T [9].

Поскольку преобразование P применяется также к матрице N и вектор-столбцу b , то

$$PN = \begin{pmatrix} 0 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \vdots & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & 0 & 1 \\ 1 & \cdot & \cdot & \cdot & \cdot & 1 & 1 \\ a_{n+n_1} & \cdot & \cdot & \cdot & a_{1n+n_1} & n_1 & 1 \\ \vdots & \cdot & \cdot & \cdot & \cdot & \vdots & \cdot \\ a_{m1+n_m} & \cdot & \cdot & \cdot & a_{mn+n_m} & n_m & \cdot \cdot \cdot 1 \end{pmatrix} \tag{1.20}$$

$$Pb = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ k \\ b_1 + kn_1 \\ \vdots \\ b_m + kn_m \end{pmatrix} \tag{1.21}$$

Образы $f(l_j), j = \overline{1, n+m+1}$ вектор-столбцов матрицы PN получим по следующему правилу:

$$f(l_j) = \begin{pmatrix} f(l_{1j}) \\ \vdots \\ f(l_{n+m+1,j}) \end{pmatrix} = l_j \text{ mod } \begin{pmatrix} -1 \\ \vdots \\ -1 \\ p_0 \\ p_1 \\ \vdots \\ p_m \end{pmatrix} ,$$

где l_j - j -й столбец матрицы PN . Запишем компоненты вектор-образов $f(l_j) (j = \overline{1, n+m+1})$:

$$f(l_{ij}) \equiv 0, i = \overline{1, n}, j = \overline{1, n+m+1};$$

$$f(l_{n+1,j}) = 1, j = \overline{1, n+1};$$

$$f(l_{n+1+i,j}) = (a_{ij} + n_i) \bmod p_i, \quad i = \overline{1, m}, \quad j = \overline{1, n};$$

$$f(l_{n+1+i, n+1}) = n_i \bmod p_i, \quad i = \overline{1, m};$$

$$f(l_{n+1,j}) \equiv 0, \quad j = \overline{n+2, n+1+m};$$

$$f(l_{n+1+i,j}) \equiv 1 \text{ при } n+1+i=j, \quad i = \overline{1, m}, \quad j = \overline{n+2, n+m+1};$$

$$f(l_{n+1+i,j}) \equiv 0 \text{ при } n+1+i \neq j, \quad i = \overline{1, m}, \quad j = \overline{n+2, n+m+1}.$$

Заметим, что при работе с фактор-группой $\{A'\}/\{A\}$ помнить все компоненты вектор-образов не обязательно в силу того, что $f(l_{ij}) \equiv 0$ при $i = \overline{1, n}, j = \overline{1, n+m+1}$. Достаточно помнить последнее $m+1$ компонент каждого образа. При этом общее количество хранимой информации равно $(m+1)D$ чисел, где D – порядок фактор-группы $\{A'\}/\{A\}$.

Проводя аналогичные рассуждения для вектор-столбца Pb , получим:

$$f(Pb) = Pb \bmod \begin{pmatrix} -1 \\ \vdots \\ -1 \\ \mu_0 \\ p_1 \\ \vdots \\ p_m \end{pmatrix} \cdot \begin{pmatrix} 0 \\ \vdots \\ 0 \\ k \bmod p_0 \\ (b_1 + kn_1) \bmod p_1 \\ \vdots \\ (b_m + kn_m) \bmod p_m \end{pmatrix}.$$

Обозначим через η множество образов. Очевидно, $|\eta| \leq n + m + 1$, так как два и более столбцов матрицы PN могут отобразиться в один и тот же элемент фактор-группы $\{A'\}/\{A\}$.

Определим коэффициенты в целевой функции задачи групповой минимизации по правилу

$$c^*(g) = \min_{j \in J} c_j^*, \quad \{J = \{j\} / f(l_j) = g\},$$

где вектор C^* находится из условия

$$C^* = C^T A^{-1}$$

или, что то же самое, из уравнения

$$C^* A = C^T.$$

Распишем это матричное уравнение

$$\begin{aligned} -C_1^* &+ C_{n+1}^* + a_{11} C_{n+2}^* + \dots + a_{m1} C_{n+m+1}^* = C_1; \\ -C_2^* &+ C_{n+1}^* + a_{12} C_{n+2}^* + \dots + a_{m2} C_{n+m+1}^* = C_2; \\ \dots & \\ -C_n^* &+ C_{n+1}^* + a_{1n} C_{n+2}^* + \dots + a_{mn} C_{n+m+1}^* = C_n; \\ p_0 C_{n+1}^* &- f_1 C_{n+2}^* - \dots - f_m C_{n+m+1}^* = M_0; \\ & p_1 C_{n+2}^* \dots = M; \\ & \dots p_m C_{n+m+1}^* = M, \end{aligned}$$

откуда

$$\begin{aligned}
 c_{n+m+1}^* &= M/p_m; \\
 c_{n+2}^* &= M/p_1; \\
 c_{n+1}^* &= (M_0 + \sum_{i=1}^m f_i c_{n+1+i}^*)/p_0; \\
 c_n^* &= c_{n+1}^* + \sum_{i=1}^m a_{in} c_{n+1+i}^* - c_n; \\
 c_1^* &= c_{n+1}^* + \sum_{i=1}^m a_{i1} c_{n+1+i}^* - c_1.
 \end{aligned} \tag{1.22}$$

Используя относительную свободу в выборе коэффициентов M и M_0 по (1.12), (1.13), выберем M и M_0 так, чтобы вектор-столбец c^* был целочисленным.

M выберем по любому из следующих двух правил:

1. $M > (U - L)k$ и $M = l \text{ НОК}(p_1, p_2, \dots, p_m)$, где l - натуральное число.

2. Можно выбрать вместо одного числа M числа $M_i, i = \overline{1, m}$ такие, что $M_i > (U - L)k$ и $M_i = l_i p_i$, где l_i - натуральные числа. В этом случае

$$c_{n+1+i}^* = [(U-L)k/p_i] + 1, \quad i = \overline{1, m}. \tag{1.23}$$

M_0 выбираем из условия

$$M_0 > U p_0 \text{ и } c_{n+1}^* = (M_0 + \sum_{i=1}^m f_i c_{n+1+i}^*)/p_0 = 0 \text{ mod } 1,$$

откуда

$$\begin{aligned}
 c_{n+1}^* &> (U p_0 + \sum_{i=1}^m f_i c_{n+1+i}^*)/p_0 = \\
 &= U + \sum_{i=1}^m \frac{f_i}{p_0} c_{n+1+i}^* = U + \sum_{i=1}^m \eta_i c_{n+1+i}^*,
 \end{aligned}$$

где η_i определены выше.

Поскольку $(U + \sum_{i=1}^m \eta_i c_{n+1+i}^*) = 0 \text{ mod } 1$, то примем

$$c_{n+1}^* = U + \sum_{i=1}^m \eta_i c_{n+1+i}^* + 1 \tag{1.24}$$

Заметим, что $c^* > 0$ [8].

Таким образом, построена задача групповой минимизации

$$\begin{aligned}
 \min_{g \in \eta} \sum c^*(g) t(g); \\
 \sum_{g \in \eta} t(g) = g_b.
 \end{aligned} \tag{1.25}$$

Задача (1.25) решается алгоритмом групповой минимизации Ху [18]. Постановка (1.25) охватывает случай $g \in \eta \in G \setminus \bar{0}$, где $\bar{0}$ - нулевой элемент фактор-группы $\{A^i\}/\{A\}^*$, в которой отображаются целочисленные столбцы матрицы $A^{-1}N$. Для элементов $g \notin \eta$ можно взять в качестве его стоимости $c^*(g')$ сколь угодно большое число, так что g' в оптимальном решении исполь-

зоваться не будет. Заметим, что группа $\{A'\}/\{A\}$ является прямой суммой $m + 1$ циклических групп.

Элемент g_b является одним из элементов $\{A'\}/\{A\}$. Поэтому решение задачи (1.25) определяет представление группового элемента g_b через другие элементы с наименьшей стоимостью.

Поставленную задачу будем решать по этапам. Информация, рассматриваемая на каждом этапе, называется *текущей*. Пусть $C^*(g_\alpha)$ – наименьшая (текущая) стоимость представления g_α через другие элементы, известная к данному текущему моменту. Пара, образованная из $C^*(g_\alpha)$ и чисел, указывающих (кодирующих) представление g_α в виде комбинации со стоимостью $C^*(g_\alpha)$ называется *пометкой группового элемента g_α* . Пометка является временной, если она может быть изменена в процессе вычислений, т. е. если найдено представление для g_α в виде комбинации элементов из $G \setminus \bar{O}$ со стоимостью, меньшей текущей стоимости $C^*(g_\alpha)$. В противном случае пометка постоянная.

Вычисления начинаются с пометок $[C^*(g_\alpha), \alpha]$. Эти пометки – временные, так как нет уверенности, что стоимость $C^*(g_\alpha)$ представления $g_\alpha = g_\alpha$ элемента g_α наименьшая. Первую компоненту пометки примем за ее стоимость. Положим, что пометка элемента g_α меньше пометки элемента g_γ , если стоимость пометки элемента g_α меньше стоимости пометки элемента g_γ .

Заметим, что так как рассматриваемая группа является суммой $m + 1$ циклических групп, то

$$g_i + g_j = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ g_{i0} \\ g_{i1} \\ \vdots \\ g_{im} \end{pmatrix} + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ g_{j0} \\ g_{j1} \\ \vdots \\ g_{jm} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ g_{(i0+j0)} \bmod p_0 \\ g_{(i1+j1)} \bmod p_1 \\ \vdots \\ g_{(im+jm)} \bmod p_m \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ g_{\alpha 0} \\ g_{\alpha 1} \\ \vdots \\ g_{\alpha m} \end{pmatrix} = g_\alpha,$$

поэтому $C^*(g_i + g_j) = C^*(g_\alpha)$.

Алгоритм решения задачи (1.25) состоит в следующем.

1. Сравнить стоимости всех временных пометок, объявить пометку с минимальной стоимостью постоянной и выделить ее. Если этой пометкой является $[C(g_b), g_b]$, то перейти на шаг 3. Если несколько пометок имеют минимальную стоимость, то в качестве постоянной выделяется первая из них. Перейти на шаг 2.

2. Пусть $g_{i_1}, g_{i_2}, \dots, g_{i_r}$ – элементы группы с постоянными пометками и g_{i_r} – последний групповой элемент, получивший постоянную пометку. Найти суммы $C^*(g_{i_1}) + C^*(g_{i_r})$, $C^*(g_{i_2}) + C^*(g_{i_r})$, \dots , $C^*(g_{i_r}) + C^*(g_{i_r})$ и сравнить их соответственно с

$$c^*(g_{i_1} + g_{i_r}), c^*(g_{i_2} + g_{i_r}), \dots, c^*(g_{i_r} + g_{i_r}).$$

Если $C^*(g_{ij}) + C^*(g_{ir}) \geq C^*(g_{ij} + g_{ir})$, то пометки не менять. Если же $C^*(g_{ij}) + C^*(g_{ir}) < C^*(g_{ij} + g_{ir})$, то заменить пометку группового элемента $(g_{ij} + g_{ir})$ на $[C^*(g_{ij}) + C^*(g_{ir}), q]$, где q – вторая компонента текущей пометки элемента g_{ij} . Перейти на шаг 1.

3. Установить $t(g_i) = 0, i = \overline{1, D}, j = b$. Перейти на шаг 4.

4. Вычислить $t(g_j) = t(g_j) + 1$. Если $g_j = q_j$, то конец, в противном случае

$$g_j = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ g_{j0} \\ g_{j1} \\ \vdots \\ g_{jm} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ g_{j0} \\ g_{j1} \\ \vdots \\ g_{jm} \end{pmatrix} - \begin{pmatrix} 0 \\ \vdots \\ 0 \\ g_{j0} \\ g_{j1} \\ \vdots \\ g_{jm} \end{pmatrix} \text{ mod } \begin{pmatrix} 0 \\ \vdots \\ p_0 \\ p_1 \\ \vdots \end{pmatrix}$$

и вновь выполнить шаг 4.

В результате получим вектор $t(g)$ решения задачи групповой минимизации. Число операций в данном алгоритме в крайнем случае равно $2D^2$.

Соотношение между x – пространством исходной ЗЛЦП и t – пространством задачи групповой минимизации (1.25) является естественным. Запишем соотношение между y и t :

$$y_j = 0, \text{ если } \begin{cases} t(g) = y & \text{если } f(l_j) = g; \\ f(l_i) = f(l_j) \neq \bar{0} & \text{и } c_i^* < c_j^* \text{ или} \\ f(l_i) = f(l_j) \neq \bar{0} & \text{и } c_i^* = c_j^*, i < j, \\ f(l_j) = \bar{0}, \end{cases}$$

где $\bar{0}$ – нулевой элемент фактор-группы $\{A'\}/\{A\}$. Определив вектор y , можно вычислить компоненты x по формуле [8]:

$$x = A^{-1}b - A^{-1}y.$$

В силу частного вида матрицы A получим

$$\begin{aligned} x_j &= y_j, \quad j = \overline{1, n}; \\ x_{n+1} &= ((k - y_{n+1}) - \sum_{j=1}^n x_j) / p_0; \\ x_{n+1+i} &= ((b_i - y_{n+1+i}) - \sum_{j=1}^n a_{ij} x_j + f_i x_{n+1}) / p_i. \end{aligned} \quad (1.26)$$

Далее сформулируем утверждение 3.

Утверждение 3. Рассмотрим ЗЛЦП общего вида с неотрицательными переменными и фиксированным числом ограничений. Пусть величины коэффициентов матрицы и столбца ограниче-

ний по модулю ограничены полиномом фиксированной степени от n -числа переменных ЗЛЦП. Кроме того, k также является полиномом фиксированной степени от n , где k определяет гиперплоскость $\sum_{i=1}^n x_i = k$, ограничивающую область допустимых целочисленных значений ЗЛЦП. Тогда решение задачи групповой минимизации, построенной по ЗЛЦП (1.9), дает точное решение исходной ЗЛЦП и имеет верхнюю оценку числа элементарных операций, представляющую собой полином фиксированной степени от n -числа переменных исходной ЗЛЦП.

Доказательство. Как было показано ранее, ЗЛЦП (1.9), сведенная к задаче групповой минимизации, асимптотическим алгоритмом решается точно и имеет оптимальное целочисленное решение, совпадающее с оптимальным целочисленным решением исходной ЗЛЦП. Рассмотрим верхнюю оценку числа операций построения задачи групповой минимизации и асимптотического алгоритма Гомори. Как показано в [9], для построения матрицы S необходимо число операций, линейное относительно n при фиксированном m . Следовательно, на основании приведенного выше алгоритма построения задачи групповой минимизации для ее построения необходимо число итераций, ограниченное полиномом первой степени от n при фиксированном m . Верхняя оценка числа операций асимптотического алгоритма Гомори равна $2(\det|S|)^2$ или в данном случае $2(\prod_{i=0}^m p_i)^2$. В силу неравенств (1.10) при выполнении условий утверждения числа p_i могут быть такими, что их величина будет ограничена полиномами фиксированной степени от n ; при фиксированной величине ограниченного m число $2(\prod_{i=0}^m p_i)^2$ становится полиномом фиксированной степени от n числа переменных задачи. Анализ формулы (1.10) показывает, что по модулю ограничены полиномом фиксированной степени от n только отрицательные коэффициенты матрицы ограничений.

Пример. Запишем исходную ЗЛЦП:

$$\begin{aligned} & \max (x_1 + x_2); \\ & 2x_1 + 2x_2 \leq 3; \\ & x_1, x_2 \geq 0; \quad x_1, x_2 = 0 \pmod{1}. \end{aligned}$$

Зададим $k=2$.

По формулам (1.17) вычислим p_0 и p_i , $i = \overline{1, m}$:

$$p_0 = k+1 = 3; \quad p_1 = b_1 - s_1 k + 1 = 4.$$

S_1 вычисляется по (1.14): $S_1 = 0$.

По формуле (1.19) вычислим r_{1i} , $i = \overline{1, m}$:

$$r_{11} = [(b_1 - s_1(2k+1)) / (k+1)] + 1 = 2.$$

По формуле (1.18) вычислим f_i , $i = \overline{1, m}$:

$$f_1 = r_{11} p_1 = 6.$$

U и L вычислим по (1.14)

$$U = 1; \quad L = 0.$$

Ограничения расширенной ЗЛЦП 1-го уровня (1.9) примут вид:

$$\begin{aligned} -x_1 &\leq 0; \\ -x_2 &\leq 0; \\ x_1 + x_2 + 3x_3 &\leq 2; \\ 2x_1 + 2x_2 - 6x_3 + 4x_4 &\leq 3. \end{aligned}$$

Запишем матрицы A (1.15); PN (1.20); Pb (1.21):

$$A = \begin{pmatrix} -1 & -1 & & \\ 1 & 1 & 3 & \\ 2 & 2 & -6 & 4 \end{pmatrix}.$$

$$PN = \begin{pmatrix} 1 & 1 & & \\ 1 & 1 & 1 & \\ 4 & 4 & 2 & 1 \end{pmatrix}.$$

$$Pb = \begin{pmatrix} 0 \\ 0 \\ 2 \\ 7 \end{pmatrix}.$$

Образы $f(l_j)$ примут вид:

$$f(l_1) = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}; \quad f(l_2) = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}; \quad f(l_3) = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 2 \end{pmatrix}; \quad f(l_4) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

Образ вектор-столбца Pb равен:

$$f(Pb) = \begin{pmatrix} 0 \\ 0 \\ 2 \\ 3 \end{pmatrix}.$$

Вычислим вектор C^* по формулам (1.22), (1.23) и (1.24):

$$C_4^* = [(U-L)k/p_1] + 1 = 1;$$

$$C_3^* = U + p_1 C_4^* + 1 = 4;$$

$$C_2^* = C_3^* + a_{12} C_4^* - C_2 = 5;$$

$$C_1^* = C_3^* + a_{11} C_4^* - C_1 = 5.$$

Заметим, что первый и второй столбцы отобразились в один элемент группы G .

Примем $f(l_1) = g$.

Порядок группы равен (1.16):

$$D = p_0 p_1 = 12.$$

Задача групповой минимизации (1.25) приняла вид:

$$\min \left(5 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + 4 \begin{pmatrix} 1 \\ 2 \end{pmatrix} + 1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right);$$

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 \\ 2 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}.$$

Первые два нуля в каждом элементе группы здесь и далее опущены:

$$\eta = \left(\begin{pmatrix} 1 \\ 0 \end{pmatrix}; \begin{pmatrix} 1 \\ 2 \end{pmatrix}; \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right).$$

Для $c^*(g) \notin \Gamma$ примем $c^*(g) = \infty$.

Заполним ряд из $D - 1$ столбцов, соответствующих $D - 1$ элементу группы. Все пометки – временные.

5	∞	1	∞	∞	∞	4	∞	∞	∞	∞
1	2	0	1	2	0	1	2	0	1	2
0	0	1*	1	1	2	2	2	3	3	3

1. Сравниваем стоимости всех временных пометок и выделяем $[1, \begin{pmatrix} 0 \\ 1 \end{pmatrix}]$ как постоянную пометку посредством введения знака *.

2. Сравниваем $c^*(\begin{pmatrix} 0 \\ 1 \end{pmatrix}) + c^*(\begin{pmatrix} 0 \\ 1 \end{pmatrix})$ с $c^*(\begin{pmatrix} 0 \\ 2 \end{pmatrix})$ так как $[1, \begin{pmatrix} 0 \\ 1 \end{pmatrix}]$ – единственная постоянная пометка. Имеем $1 + 1 = 2 < \infty$.

Заменяем $[\infty, \begin{pmatrix} 0 \\ 2 \end{pmatrix}]$, соответствующую элементу группы $(\begin{pmatrix} 0 \\ 2 \end{pmatrix})$, на $[2, \begin{pmatrix} 0 \\ 1 \end{pmatrix}]$, как показано ниже.

5	∞	1	∞	∞	2	4	∞	∞	∞	∞
1	2	0	1	2	0	1	2	0	1	2
0	0	1*	1	1	1*	2	2	3	3	3

1. Сравниваем стоимости всех временных пометок и выделяем $[2, \begin{pmatrix} 0 \\ 1 \end{pmatrix}]$ как постоянную. Эта пометка соответствует элементу группы $(\begin{pmatrix} 0 \\ 2 \end{pmatrix})$.

2. Поскольку $(\begin{pmatrix} 0 \\ 2 \end{pmatrix})$ – последний элемент группы, получивший постоянную пометку, сравниваем

$$c^*(\begin{pmatrix} 0 \\ 1 \end{pmatrix}) + c^*(\begin{pmatrix} 0 \\ 2 \end{pmatrix}) = 1 + 2 = 3 < \infty = c^*(\begin{pmatrix} 0 \\ 3 \end{pmatrix})$$

и заменяем $[\infty, \begin{pmatrix} 0 \\ 3 \end{pmatrix}]$ на $[3, \begin{pmatrix} 0 \\ 1 \end{pmatrix}]$;

$$c^*(\begin{pmatrix} 0 \\ 2 \end{pmatrix}) + c^*(\begin{pmatrix} 0 \\ 2 \end{pmatrix}) = 2 + 2 = 4 > 0 = c^*(\begin{pmatrix} 0 \\ 0 \end{pmatrix}) = c(\bar{0}),$$

замен нет.

В результате получается

5	∞	1	∞	∞	2	4	∞	3	∞	∞
1	2	0	1	2	0	1	2	0	1	2
0	0	1*	1	1	1*	2	2	1	3	3

В дальнейшем данная таблица будет последовательно принимать вид:

5	∞	1	∞	∞	2	4	∞	3	∞	∞
1	2	0	1	2	0	1	2	0	1	2
0	0	1*	1	1	1*	2	2	1*	3	3

5	8	1	7	∞	2	4	∞	3	∞	∞
1	1	0	0	2	0	1	2	0	1	2
0	2	1*	3	1	1*	2*	2	1*	3	3

5	8	1	6	∞	2	4	9	3	5	∞
1	1	0	0	2	0	1	1	0	0	2
0*	2	1*	1	1	1*	2*	2	1*	1	3

5	8	1	6	9	2	4	9	3	5	10
1	1	0	0	1	0	1	1	0	0	1
0*	2	1*	1	2	1*	2*	2	1*	1*	0

5	8	1	6	9	2	4	9	3	5	10
1	1	0	0	1	0	1	1	0	0	1
0*	2	1*	1*	2	1*	2*	2	1*	1*	0

5	8	1	6	9	2	4	9	3	5	10
1	1	0	0	1	0	1	1	0	0	1
0*	2*	1*	1*	2	1*	2*	2	1*	1*	0

5	8	1	6	9	2	4	9	3	5	10
1	1	0	0	1	0	1	1	0	0	1
0*	2*	1*	1*	2*	1*	2*	2	1*	1*	0

5	8	1	6	9	2	4	9	3	5	10
1	1	0	0	1	0	1	1	0	0	1
0*	2*	1*	1*	2*	1*	2*	2*	1*	1*	0

5	8	1	6	9	2	4	9	3	5	10
1	1	0	0	1	0	1	1	0	0	1
0*	2*	1*	1*	2*	1*	2*	2*	1*	1*	0

Элемент группы $(\begin{smallmatrix} 2 \\ 3 \end{smallmatrix})$ с пометкой $[10, (\begin{smallmatrix} 1 \\ 0 \end{smallmatrix})]$ вошел в множество постоянных пометок. Выразим элемент группы $g_b = (\begin{smallmatrix} 2 \\ 3 \end{smallmatrix})$ через $g_i, i \in \eta$ с минимальной стоимостью.

$$(\begin{smallmatrix} 2 \\ 3 \end{smallmatrix}) = (\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}) + (\begin{smallmatrix} 1 \\ 3 \end{smallmatrix}); (\begin{smallmatrix} 1 \\ 3 \end{smallmatrix}) = (\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}) + (\begin{smallmatrix} 1 \\ 2 \end{smallmatrix}).$$

Значит, $(\begin{smallmatrix} 2 \\ 3 \end{smallmatrix}) = (\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}) + (\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}) + (\begin{smallmatrix} 1 \\ 2 \end{smallmatrix})$. Следовательно, $t(\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}) = 1$; $t(\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}) = 1$;

$t(\frac{1}{2}) = 1$, откуда $y_1 = 1, y_2 = 0, y_3 = 1, y_4 = 1$. По формулам (1.26) определим x :

$$\begin{aligned}x_1 &= y_1 = 1; \\x_2 &= y_2 = 0; \\x_3 &= ((k - y_3) - (x_1 + x_2)) / p_0 = ((2 - 1) - 1) / 3 = 0; \\x_4 &= ((b_1 - y_4) - (2x_1 + 2x_2) + 6x_3) / p_1 = \\&= ((3 - 1) - (2 \cdot 1 + 2 \cdot 0) + 6 \cdot 0) / 4 = 0.\end{aligned}$$

2. СВЕДЕНИЕ ЗАДАЧИ ЛИНЕЙНОГО ЦЕЛОЧИСЛЕННОГО ПРОГРАММИРОВАНИЯ К ЗАДАЧЕ „О РАНЦЕ”

ЗЛЦП представим в виде двух моделей:

$$\max \sum_{j=1}^n c_j x_j; \quad (1.27)$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = \overline{1, m}; \quad \forall x_j \geq 0, \quad x_j \equiv 0 \pmod{1}; \quad (1.28)$$

$$\max \sum_{j=1}^n c_j x_j; \quad (1.29)$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = \overline{1, m}; \quad \forall x_j \geq 0, \quad x_j \equiv 0 \pmod{1}; \quad (1.30)$$

$$\forall x_j \in D_j, \quad a_{ij}, b_i - \text{целые числа.} \quad (1.31)$$

Ограничения (1.28) либо (1.30) являются связными, т. е. в каждое из них с ненулевыми коэффициентами входит более чем одна переменная. Легко видеть, что ЗЛЦП общего вида может быть представлена либо моделью (1.27), (1.28), либо (1.29)–(1.31). Действительно, пусть в исходной ЗЛЦП были закононеопределенные переменные, на которые наложены дополнительно ограничения вида $-a_j \leq x_j \leq b_j$, тогда каждая из них заменится разностью неотрицательных $x_j = y_j - z_j$ и на переменные y_j, z_j накладываются ограничения вида (1.31):

$$0 \leq y_j \leq b_j; \quad 0 \leq z_j \leq a_j.$$

Если в исходной ЗЛЦП переменные являются знакоопределенными, то очевидным образом они преобразуются в неотрицательные, а множества D_j в этом случае могут быть произвольной совокупностью целых чисел (неотрицательных).

Сведем ЗЛЦП вида (1.27), (1.28) либо (1.29)–(1.31) с ЗЛЦП с одним связным ограничением. Аналитические выкладки проведем последовательно сначала для ЗЛЦП (1.27), (1.28), а затем, используя полученные результаты, для ЗЛЦП (1.29)–(1.31). Используя результаты предыдущего параграфа, ЗЛЦП (1.27), (1.28) можно преобразовать в эквивалентную ЗЛЦП с квадратной матрицей ограничений неравенств:

$$\max C^T x = \max \sum_{j=1}^n c_j x_j + M_0 x_{n+1} + M \sum_{j=n+2}^{n+m+1} x_j; \quad (1.32)$$

$$x_j \geq 0, \quad j = \overline{1, n}; \quad (1.33)$$

$$\sum_{j=1}^n x_j + p_0 x_{n+1} \leq k; \quad (1.34)$$

$$\max \sum_{j=1}^n c_j x_j ; \quad (1.41)$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i ; \quad \forall x_j \equiv 0 \pmod{1},$$

a_{ij}, b_i – целые числа.

Предположим, что существует такая i -я строка матрицы ограничений ЗЛЦП (1.41), при изменении которой определитель новой матрицы ограничений может быть равным единице. Запишем разложение по i -й строке

$$\det A = \sum_{j=1}^n a_{ij} p_{ij},$$

где p_{ij} – алгебраическое дополнение элемента.

Таким образом, необходимо добиться того, чтобы

$$\sum_{j=1}^n \hat{a}_{ij} p_{ij} = 1, \quad (1.42)$$

где \hat{a}_{ij} – измененные и пока неизвестные элементы i -й строки матрицы A . Как известно, необходимое и достаточное условие разрешимости в целых числах линейного уравнения (1.42), где p_{ij} – целые числа, состоит в том, что НОД чисел $p_{ij}, i = \overline{1, n}$ был равен единице. Это и является условием, определяющим класс задач, сводящихся к задаче „о ранце“: должна существовать такая строка матрицы ограничений A , алгебраические дополнения к элементам которой имеют НОД, равный единице.

Введем обозначение

$$D = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ \hat{a}_{i1} & & \hat{a}_{in} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix}.$$

Матрица D – унимодулярная целочисленная. Введем замену переменных

$$Dx = y. \quad (1.43)$$

Из свойств унимодулярных целочисленных матриц вытекает, что вектор x является целочисленным тогда и только тогда, когда является целочисленным вектор y . С учетом замены переменных (1.43) ЗЛЦП (1.41) можно переписать

$$\begin{aligned} \max \sum_{i=1}^n \hat{c}_i y_i ; \\ y_1 \leq b_1 ; \\ \vdots \\ y_{i-1} \leq b_{i-1} ; \end{aligned} \quad (1.44)$$

$$y_i \leq b_i + \sum_{j=1}^n (\hat{a}_{ij} - a_{ij}) \sum_{k=1}^n d_{jk}^{-1} y_k;$$

$$y_{i+1} \leq b_{i+1};$$

$$\vdots$$

$$y_n \leq b_n,$$

где вектор-строка C^* определяется из условия

$$\hat{C}^T = C^T D^{-1}.$$

При введении замены переменных $Z_p = b_p - y_p > 0$ в ограничениях вида $y_p \leq b_p$ ЗЛЦП (1.44) сводим к задаче „о ранце“.

$$\max \sum_{i=1}^n \hat{C}_i Z_i + \hat{C}_i Y_i; \quad (1.45)$$

$$Z_i \geq 0, \dots, Z_{i-1} \geq 0, Y_i \leq b_i + \sum_{j=1}^n (\hat{a}_{ij} - a_{ij}) \left(\sum_{\substack{k=1 \\ k \neq i}}^n d_{jk}^{-1} (b_k - Z_k) + d_{ji}^{-1} Y_i \right), Z_{i+1} \geq 0, \dots, Z_n \geq 0.$$

Для получения ЗЛЦП (1.45) необходимо найти частное решение диофантова уравнения вида (1.42).

Существующие схемы поиска общего решения линейного диофантова уравнения достаточно сложны с точки зрения их реализации на ЭВМ. Однако эти же методы решения позволяют построить эффективные вычислительные схемы для нахождения частного решения. Ниже предлагается одна из таких схем.

В основе ее лежит метод наименьшего коэффициента, который состоит в последовательном исключении одной из неизвестных. Пусть дано уравнение

$$c x_0 + c_1 x_1 + \dots + c_k x_k = d. \quad (1.46)$$

Найдем в данном уравнении ненулевой коэффициент с наименьшим абсолютным значением. Для простоты можно принять, что $c > 0$. Если $c = 1$ то общее решение диофантова уравнения (1.46) очевидно: $x_i, i = 1, k$ — произвольные целые, а $x_0 = d - c_1 x_1 - \dots - c_k x_k$.

Если $c > 1$ и $c_1 \bmod c = \dots = c_k \bmod c = 0$, то необходимо иметь $d \bmod c = 0$, так как в противном случае целочисленного решения не существует; дальше делим обе части уравнения (1.46) на c и получаем общее решение так же, как и в случае $c = 1$. И, наконец, если $c > 1$ и не все $c_1 \bmod c, \dots, c_k \bmod c$ нулевые, вводим новую переменную

$$L(c/c) x_0 + L(c_1/c) x_1 + \dots + L(c_k/c) x_k = t$$

и заменяем исходное уравнение (1.46) уравнением

$$ct + (c_1 \bmod c) x_1 + \dots + (c_k \bmod c) x_k = d.$$

Этот процесс должен закончиться, поскольку в результате выпол-

нения каждого шага уменьшается величина наименьшего ненулевого коэффициента в уравнении. Таково краткое изложение метода наименьшего коэффициента.

Кроме того, перед началом решения линейного диофантова уравнения необходимо знать, разрешимо ли оно в целых числах.

Известно, что необходимое и достаточное условие разрешимости в целых числах линейного уравнения [16]

$$\sum_{j=1}^n a_j x_j = b, \quad (1.47)$$

где $a_j (j = \overline{1, n})$ и b – целые числа, состоит в том, что наибольший общий делитель (a_1, \dots, a_n) чисел a_1, \dots, a_n делит b .

Теперь перейдем к описанию вычислительной схемы, которую удобно представить в виде таблиц. Пусть дано уравнение (1.47). Сначала составляем таблицу прямого прохода, которая имеет $2n$ столбцов. Первые n столбцов соответствуют переменным x_1, \dots, x_n следующие n столбцов соответствуют переменным t_1, \dots, t_n которые вводятся на очередной итерации метода наименьшего коэффициента. Строка такой таблицы представляет собой коэффициенты диофантова уравнения на очередной итерации. Первая строка заполняется следующим образом: в первые n столбцов заносятся коэффициенты уравнения перед соответствующими переменными, следующие n столбцов заполняют нулями, так как ни одна новая переменная еще не введена.

Рассмотрим, как получается $(k+1)$ -я строка из k -й. Находится в k -й строке ненулевой коэффициент с наименьшим абсолютным значением. В $(k+1)$ -й строке под этим коэффициентом ставится 0, так как соответствующая этому коэффициенту переменная выводится из уравнения на очередной итерации. Но в то же время этот коэффициент ставится в $(k+1)$ -й строке в том столбце, который соответствует вводимой на очередной итерации переменной. Все остальные коэффициенты $(k+1)$ -й строки получаются как остатки целочисленного деления коэффициентов тех же столбцов k -й строки на ненулевой коэффициент k -й строки с наименьшим абсолютным значением. Такая процедура построения строк таблицы продолжается до тех пор, пока в очередной строке не встретится единичный коэффициент.

Пример. Пусть дано уравнение

$$101x_1 + 7x_2 + 13x_3 = 1.$$

Согласно необходимому и достаточному условию разрешимости в целых числах линейного уравнения данное уравнение разрешимо.

Таблица, построенная по описанной выше методике, имеет вид:

Таблица 1

x_1	x_2	x_3	t_1	t_2	t_3
101	7	13	0	0	0
3	0	6	0	7	0
0	0	0	3	1	0

Как только получена строка с единичным коэффициентом (одним или более), полагаем все переменные, кроме одной, имеющей единичный коэффициент, с ненулевыми коэффициентами в этой строке равным нулю. Тогда, оставшаяся одна переменная с единичным коэффициентом равна правой части диофантова уравнения \bar{b} .

Теперь составляем таблицу обратного прохода, которая также имеет $2n$ столбцов (табл. 2), причем таких же, как и таблица прямого прохода. Однако построение таблицы начинается с последней строки и далее строки надстраиваются сверху. Последняя строка заполняется следующим образом: в столбцах, соответствующих всем переменным, кроме одной, имеющей единичный коэффициент, с ненулевыми коэффициентами в последней строке табл. 1, ставим нули; в столбце, соответствующем оставшейся одной переменной с единичным коэффициентом в последней строке табл. 1, ставим \bar{b} ; в остальных столбцах ставим символ *, если они соответствуют переменным x_i , и 0, если они соответствуют переменным t_j .

Рассмотрим, как получается $(k-1)$ -я строка из k -й. Во-первых, следует заметить, что все элементы k -й строки, имеющие числовые значения, без изменения переносятся в $(k-1)$ -ю строку. Затем поступают следующим образом. Умножают скалярно k -ю строку табл. 2 на $(k-1)$ -ю строку табл. 1, рассматривая символ * как 0. Вычитают полученное скалярное произведение из \bar{b} (правой части диофантова уравнения). Потом находят в $(k-1)$ -й строке табл. 1 ненулевой коэффициент с наименьшим абсолютным значением среди тех столбцов, в которых в k -й строке табл. 2 стоит символ *. В столбце, соответствующем этому коэффициенту, в $(k-1)$ -й строке табл. 2 записывают число, равное отношению разности \bar{b} и полученного ранее скалярного произведения к этому коэффициенту с наименьшим абсолютным значением. Во всех остальных столбцах $(k-1)$ -й строки табл. 2, которым соответствуют ненулевые элементы $(k-1)$ -й строки табл. 1 и символ * в k -й строке табл. 2, записывают нули. В оставшиеся незаполненными столбцы $(k-1)$ -й строки табл. 2 передается без изменения символ * из k -й строки.

Такая процедура построения строк таблицы продолжается до тех пор, пока в очередной строке не останется символов *. Тогда первые n элементов этой строки представляют собой частное решение исходного диофантова уравнения.

Таблица обратного прохода для нашего примера имеет вид:

Таблица 2

x_1	x_2	x_3	t_1	t_2	t_3
- 2	29	0	0	1	0
- 2	*	0	0	1	0
*	*	*	0	1	0

В утолщенную рамку выделено решение диофантова уравнения.

При выполнении прямого прохода может возникнуть ситуация, когда из уравнения выводится переменная, ранее введенная на какой-то итерации (в том случае, если на текущей итерации этой переменной соответствует ненулевой коэффициент с наименьшим абсолютным значением). В этом случае вместо нее должна быть введена в уравнение другая новая переменная и может оказаться, что в таблицах столбцов будет больше, чем $2n$. Однако вычисления на итерации прямого прохода при этом не изменяются. Обратный проход будет также выполняться по вышеописанной схеме.

Очевидно, предложенная здесь вычислительная схема легко реализуется на ЭВМ. Несложно также убедиться в том, что временная сложность схемы увеличивается с ростом размерности уравнения не быстрее кубического полинома от n . Таким образом, предложенная вычислительная схема может оказаться очень полезной при решении различных задач целочисленной оптимизации.

Вернемся к ЗЛЦП (1.32)–(1.35).

В [12] было показано, что с помощью соответствующего выбора коэффициентов $p_0, \dots, p_n, f_1, \dots, f_m$ удовлетворяющих (1.36), можно добиться, чтобы матрица A (1.40) такую строку содержала. Этот результат основан на следующих леммах.

Лемма 1.

При разложении $\det A$ по любой строке с номером $i = \overline{n+1, m+n+1}$ первые n адьюнктов равны нулю.

Доказательство.

Первые n строк матрицы A содержат по одному ненулевому (диагональному) элементу $C_{jj}, j = \overline{1, n}$, где

$$A = \{C_{ij}\}_{m+n+1}^{m+n+1}$$

Поэтому при вычеркивании любого столбца с номером $k = \overline{1, n}$ вычеркивается и элемент C_{kk} , а так как k -я строка при этом не вычеркивается в силу того, что $k < n+1$, то в полученном миноре имеется нулевая строка, что доказывает лемму.

Лемма 2.

Имеет место равенство:

$$\begin{vmatrix} -1 & 0 & \dots & 0 & 0 \\ \alpha_1 & d_{11} & \dots & d_{1,n-1} & d_{1n} \\ \alpha_2 & d_{21} & \dots & d_{2,n-1} & d_{2n} \\ \vdots & \vdots & & \vdots & \vdots \\ \alpha_n & d_{n1} & \dots & d_{n,n-1} & d_{nn} \end{vmatrix} = - \begin{vmatrix} d_{11} & \dots & d_{1,n-1} & d_{1n} \\ d_{21} & \dots & d_{2,n-1} & d_{2n} \\ \vdots & & \vdots & \vdots \\ d_{n1} & \dots & d_{n,n-1} & d_{nn} \end{vmatrix}$$

где числа α_i произвольные, $n > 0$ – целое число. Доказательство леммы тривиально.

Рассмотрим разложение $\det A$ по строке с номером $n+1$. В силу леммы 1 первые n адьюнктов в таком разложении равны нулю. Докажем, что $(n+1)$ -й адьюнкт разложения $A_{n+1, n+1}$ удовлетворяет равенству

$$A_{n+1, n+1} = (-1)^n \prod_{j=1}^m p_j. \quad (1.48)$$

В дальнейшем через A_{il} будем обозначать алгебраическое дополнение элемента C_{il} определителя матрицы A . Докажем также, что

$$A_{n+1, n+k} = (-1)^n f_{k-1} \prod_{\substack{j=1 \\ j \neq k-1}}^m p_j, \quad k = \overline{2, m+1}. \quad (1.49)$$

$A_{n+1, n+k}$, $k = \overline{1, m+1}$ получается из $\det A$ вычеркиванием $(n+1)$ -й строки и $(n+k)$ -го столбца и домножением полученного минора на $(-1)^{n+1+n+k} = (-1)^{k+1}$, т. е.

$$A_{n+1, n+k} = (-1)^{k+1} M_{n+1, n+k},$$

где M_{il} – минор определителя матрицы A , полученной вычеркиванием i -й строки и l -го столбца.

Применяя n раз лемму 2 к $M_{n+1, n+k}$, $k = \overline{1, m+1}$, получим

$$M_{n+1, n+k} = (-1)^n N_{1k}, \quad k = \overline{1, m+1},$$

где N_{1k} , $k = \overline{1, m+1}$, – миноры определителя

$$\begin{vmatrix} p_0 & 0 & \dots & 0 \\ -f_1 & p_1 & 0 & \dots & 0 \\ \vdots & 0 & & \vdots & \\ \vdots & \vdots & & \vdots & \\ -f_m & 0 & \dots & 0 & p_m \end{vmatrix},$$

откуда

$$A_{n+1, n+k} = (-1)^{k+1} (-1)^n N_{1k} = (-1)^{n+k+1} N_{1k}, \quad (1.50)$$

$$k = \overline{1, m+1}.$$

Из (1.50) вытекает справедливость (1.48).

Покажем, что

$$N_{1k} = (-1)^{k-1} f_{k-1} \prod_{\substack{j=1 \\ j \neq k-1}}^m p_j, \quad k = \overline{2, m+1}. \quad (1.51)$$

Из (1.51) в силу (1.50) вытекает справедливость (1.49). Соотношение (1.51) докажем методом математической индукции.

1. Базис индукции ($m = 2$)

Рассмотрим определитель

$$\begin{vmatrix} p_0 & 0 & 0 \\ -f_1 & p_1 & 0 \\ -f_2 & 0 & p_2 \end{vmatrix}.$$

Очевидно, $N_{12} = -f_1 p_2$, $N_{13} = f_2 p_1$. Это соответствует формуле (1.51).

2. Шаг индукции

Предположим, что соотношение верно для $m = l$, т. е.

$$N_{1k} = (-1)^{k-1} f_{k-1} \prod_{\substack{j=1 \\ j \neq k-1}}^l p_j, \quad k = \overline{2, l+1}. \quad (1.52)$$

Покажем, что оно верно и для $m = l + 1$. С этой целью рассмотрим определитель

$$\begin{vmatrix} p_0 & & & & \\ -f_1 & p_1 & & & 0 \\ \vdots & 0 & \ddots & & \\ -f_l & 0 & \dots & 0 & p_l \\ -f_{l+1} & 0 & \dots & 0 & p_{l+1} \end{vmatrix}.$$

Разложив его по последнему столбцу, получим

$$\begin{vmatrix} p_0 & & & & \\ -f_1 & p_1 & & & 0 \\ \vdots & 0 & \ddots & & \\ -f_l & 0 & \dots & 0 & p_l \\ -f_{l+1} & 0 & \dots & 0 & p_{l+1} \end{vmatrix} = p_{l+1} \begin{vmatrix} p_0 & & & & \\ -f_1 & p_1 & & & 0 \\ \vdots & 0 & \ddots & & \\ -f_l & 0 & \dots & 0 & p_l \end{vmatrix}. \quad (1.53)$$

Легко видеть, что миноры N_{1k} , $k = \overline{2, l+1}$ определителей, стоящих в обеих частях равенства (1.53), также удовлетворяют этому равенству.

Миноры определителя, стоящего в правой части равенства (1.53), вычисляются по соотношению (1.52) по индуктивному предположению. Отсюда

$$N_{1k} = p_{l+1} (-1)^{k-1} f_{k-1} \prod_{\substack{j=1 \\ j \neq k-1}}^l p_j =$$

$$= (-1)^{k-1} f_{k-1} \prod_{\substack{j=1 \\ j \neq k-1}}^{l+1} p_j, \quad k = \overline{2, l+1} \quad \text{для} \quad m = l+1. \quad (1.54)$$

Теперь рассмотрим определитель

$$\begin{vmatrix} -f_1 & p_1 & & & 0 \\ \vdots & \vdots & & & \\ -f_l & 0 & \dots & 0 & p_l \\ -f_{l+1} & 0 & \dots & 0 & 0 \end{vmatrix}.$$

Разложив $N_{l, l+2}$ по последней строке, запишем

$$N_{l, l+2} = (-1)^{l+1} f_{l+1} \begin{vmatrix} p_1 & 0 \\ 0 & \dots & p_l \end{vmatrix} = (-1)^{l+1} f_{l+1} \prod_{\substack{j=1 \\ j \neq l+1}}^{l+1} p_j. \quad (1.55)$$

Из (1.54) и (1.55) получим

$$N_{lk} = (-1)^{k-1} f_{k-1} \prod_{\substack{j=1 \\ j \neq k-1}}^{l+1} p_j, \quad k = \overline{2, l+2} \quad \text{для} \quad m = l+1.$$

Таким образом, доказана справедливость соотношения (1.49). Справедливость равенств (1.48), (1.49) позволяет сформулировать следующую теорему.

Теорема. Если числа $p_i, f_i, i = \overline{1, m}$ выбрать простыми, произвольными, отличными друг от друга, то числа

$$\{ |A_{n+1, i}| \}, \quad i = \overline{1, m+n+1},$$

являются взаимно простыми, т. е. их НОД равен 1.

Доказательство.

Действительно, $\text{НОД} \{ |A_{n+1, n+k}| \}, k = \overline{1, m}$ равен p_m , а $\text{НОД} \{ p_m; |A_{n+1, n+m+1}| \}$ равен 1. Следовательно, числа $\{ |A_{n+1, i}| \}, i = \overline{1, n+m+1}$ взаимно просты.

Поскольку ограничения (1.36) позволяют выбрать числа p_i, f_i , удовлетворяющие теореме, то тем самым доказано, что ЗЛЦП (1.27), (1.28) сводится к задаче с одним связным ограничением.

Решив диофантово уравнение

$$\sum_{i=1}^{m+1} \hat{a}_i \lambda_{n+1, n+i} = 1, \quad (1.56)$$

где $\lambda_{n+1, n+i}$ – алгебраическое дополнение соответствующих элементов $n+1$ -й строки матрицы A (1.40) и согласно вышеприведенным результатам равны:

$$\lambda_{n+1, n+1} = (-1)^n \prod_{j=1}^m p_j, \quad (1.57)$$

$$\lambda_{n+1, n+i} = (-1)^n f_{i-1} \prod_{\substack{j=1 \\ j \neq i-1}}^m p_j, \quad i = \overline{2, m+1}, \quad (1.58)$$

получим новые элементы строки $n+1$ матрицы A_1 , для которых определитель $\det A_1 = 1$.

Тогда

$$\alpha_{11}^{-1} = \frac{p_1 \cdot p_2 \cdot \dots \cdot p_m}{\hat{a}_1 \prod_{i=1}^m p_i + \hat{a}_2 f_{i=2} \prod_{i=2}^m p_i + \dots + \hat{a}_{m+1} f_m \prod_{i=1}^{m-1} p_i}.$$

Умножив числитель и знаменатель на $(-1)^n$ заметим, что знаменатель равен 1 в силу (1.56)–(1.58), и, следовательно,

$$\alpha_{11}^{-1} = (-1)^n \prod_{i=1}^m p_i. \quad (1.62)$$

Остальные элементы столбца имеют вид

$$\alpha_{i1}^{-1} = \frac{f_{i-1}}{p_{i-1}} \alpha_{11}^{-1} = (-1)^n f_{i-1} \prod_{\substack{j=1 \\ j \neq i-1}}^m p_j, \quad i = \overline{2, m+1}. \quad (1.63)$$

Заметим, что $\alpha_{i1}^{-1} (i = \overline{1, m+1})$ являются коэффициентами диофантова уравнения (1.56).

При вычислении остальных элементов α^{-1} используется выражение $\alpha^{-1} \alpha = E$. Запишем произведение i -й строки $\alpha^{-1} (i = \overline{1, m+1})$ на j -й столбец $\alpha (j = \overline{2, m+1})$:

для $i \neq j$

$$\alpha_{i1}^{-1} \hat{a}_j + \alpha_{ij}^{-1} p_{j-1} = 0 \Rightarrow \alpha_{ij}^{-1} = -\frac{\hat{a}_j}{p_{j-1}} \alpha_{i1}^{-1}, \quad (1.64)$$

для $i = j$

$$\alpha_{i1}^{-1} \hat{a}_j + \alpha_{ij}^{-1} p_{j-1} = 1 \Rightarrow \alpha_{ij}^{-1} = \frac{1}{p_{j-1}} (1 - \alpha_{i1}^{-1} \hat{a}_j). \quad (1.65)$$

Формулы (1.64), (1.65) перепишем так:

$$\alpha_{ij}^{-1} = (-1)^{n+1} \frac{\hat{a}_j}{p_{j-1}} \prod_{k=1}^m p_k, \quad j = \overline{2, m+1}; \quad (1.66)$$

$$\alpha_{ij}^{-1} = (-1)^{n+1} \hat{a}_j f_{i-1} \frac{1}{p_{j-1}} \prod_{\substack{k=1 \\ k \neq i-1}}^m p_k, \quad i = \overline{2, m+1}, j = \overline{2, m+1}, i \neq j; \quad (1.67)$$

$$\alpha_{ii}^{-1} = \frac{1}{p_{i-1}} (1 - (-1)^n \hat{a}_i f_{i-1} \prod_{\substack{k=1 \\ k \neq i-1}}^m p_k), \quad i = \overline{2, m+1}. \quad (1.68)$$

Найдем выражение для $t = \alpha^{-1} \beta$

$$t_{ij} = \left(\sum_{k=2}^{m+1} \alpha_{ik}^{-1} a_{k-1, j} \right) + \alpha_{i1}^{-1}, \quad i = \overline{1, m+1}, j = \overline{1, n} \quad (1.69)$$

или

$$t_{ij} = (-1)^n \left(1 - \sum_{k=2}^{m+1} \frac{\hat{a}_k}{p_{k-1}} a_{k-1, j} \right) \prod_{i=1}^m p_i, \quad j = \overline{1, n}; \quad (1.70)$$

$$t_{ij} = (-1)^{n+1} f_{i-1} \prod_{\substack{l=1 \\ l \neq i-1}}^m p_l \left(\sum_{\substack{k=2 \\ k \neq i}}^{m+1} \frac{a_k}{p_{k-1}} a_{k-1, j} \right) + \frac{a_{i-1, j}}{p_{i-1}} (1 - (-1)^n \hat{a}_i).$$

$$f_{i-1} \prod_{\substack{k=1 \\ k \neq i-1}}^m p_k + (-1)^n f_{i-1} \prod_{\substack{k=1 \\ k \neq i-1}}^m p_k, \quad i = \overline{2, m+1}, j = \overline{1, n}. \quad (1.71)$$

Вернемся к матрице D

$$D = AA_1^{-1} = \begin{pmatrix} -E & 0 \\ \beta & \alpha^* \end{pmatrix} \cdot \begin{pmatrix} -E & 0 \\ t & \alpha^{-1} \end{pmatrix} = \begin{pmatrix} E & 0 \\ \gamma & \eta \end{pmatrix}, \quad (1.72)$$

где $\gamma = (\gamma_{ij}), i = \overline{1, m+1}, j = \overline{1, n}, \eta = (\eta_{ij}), i = \overline{1, m+1}, j = \overline{1, m+1}. \quad (1.73)$

Тогда $\gamma_{ij} = p_0 t_{ij}^{-1}, j = \overline{1, n}; \quad (1.74)$

$$\eta_{ij} = p_0 \alpha_{ij}^{-1}, \quad j = \overline{1, m+1}.$$

$$\eta_{ij} = -f_{i-1} \alpha_{ij}^{-1} + p_{i-1} \alpha_{ij}^{-1} = (-1)^n f_{i-1} \hat{a}_j \frac{1}{p_{j-1}} \prod_{k=1}^m p_k +$$

$$+ (-1)^{n+1} p_{i-1} f_{i-1} \hat{a}_j \frac{1}{p_{j-1}} \prod_{\substack{k=1 \\ k \neq i-1}}^m p_k = 0, \quad i = \overline{2, m+1}, j = \overline{1, m+1}, j \neq i; \quad (1.75)$$

$$\eta_{ii} = (-1)^n f_{i-1} \frac{\hat{a}_i}{p_{i-1}} \prod_{k=1}^m p_k + p_{i-1} \frac{1}{p_{i-1}} (1 - (-1)^n \hat{a}_i f_{i-1} \cdot$$

$$\cdot \prod_{\substack{k=1 \\ k \neq i-1}}^m p_k) = 1, \quad i = \overline{2, m+1}; \quad (1.76)$$

$$\gamma_{ij} = -a_{i-1, j} + (-f_{i-1}) t_{ij} + p_{i-1} t_{ij} = -a_{i-1, j} + (-f_{i-1}) \cdot (-1)^n \prod_{k=1}^m p_k +$$

$$+ f_{i-1} (-1)^n \prod_{l=1}^m p_l \sum_{k=2}^{m+1} \frac{\hat{a}_k}{p_{k-1}} a_{k-1, j} + p_{i-1} (-1)^{n+1} f_{i-1} \prod_{\substack{l=1 \\ l \neq i-1}}^m p_l \sum_{k=2}^{m+1} \frac{\hat{a}_k}{p_{k-1}}.$$

$$a_{k-1, j} + p_{i-1} \frac{a_{i-1, j}}{p_{i-1}} - \frac{a_{i-1, j}}{p_{i-1}} p_{i-1} (-1)^n \hat{a}_i f_{i-1} \prod_{\substack{k=1 \\ k \neq i-1}}^m p_k + p_{i-1} (-1)^n f_{i-1} \prod_{k=1}^m p_k = 0,$$

$$i = \overline{2, m+1}, j = \overline{1, n}. \quad (1.77)$$

Следовательно, ЗЛЦП (1.60) можно переписать в виде

$$\max \sum_{j=1}^{m+n+1} \hat{C}_j Y_j;$$

$$Y_j \leq 0, \quad j = \overline{1, n}; \quad (1.78)$$

$$\sum_{j=1}^n (p_0 t_{ij}^{-1}) Y_j + p_0 \sum_{j=n+1}^{n+m+1} \alpha_{1, j-n}^{-1} Y_j \leq k;$$

где $\hat{C}^T = C^T A_1^{-1}, Y_{n+2} \leq b_1, \dots, Y_{n+m+1} \leq b_m,$

Компоненты вектор-строки $\hat{C}^T = (\hat{C}_1, \dots, \hat{C}_{n+m+1})$ удовлетворяют условиям:

$$\hat{C}_i = -C_i + M_0 t_{i1} + M \sum_{j=2}^{m+1} t_{ji}, \quad i = \overline{1, n};$$

$$\hat{C}_{n+i} = M_0 \alpha_{i1}^{-1} + M \sum_{j=2}^{m+1} \alpha_{ji}^{-1}, \quad i = \overline{1, m+1}. \quad (1.79)$$

ЗЛЦП (1.78) введением замены переменных

$$Z_j = \begin{cases} -Y_j, & j = \overline{1, n}, \\ Y_j, & j = n+1, \\ b_i - Y_j, & j = \overline{n+2, n+m+1}, i = j-n-1. \end{cases} \quad (1.80)$$

сводим к задаче о „ранце“

$$\begin{aligned} & \max \sum_{j=1}^{n+m+1} \hat{c}_j z_j ; \\ & \sum_{j=1}^n (1-p_0 t_{1j}) z_j + p_0 \alpha_{11}^{-1} z_{n+1} - p_0 \sum_{j=n+2}^{n+m+1} \alpha_{1,(j-n)}^{-1} z_j \leq \\ & \leq k - p_0 \sum_{j=1}^m \alpha_{1,(j+1)}^{-1} b_j, \quad z_j \geq 0, \quad j = \overline{1, m+n+1}, j \neq n+1, \end{aligned} \quad (1.81)$$

где

$$\hat{c}_j = -\hat{c}_j, \quad j = \overline{1, n}, \quad \hat{c}_{n+1} = \hat{c}_{n+1}, \quad \hat{c}_{n+1+j} = -\hat{c}_{n+1+j}, \quad j = \overline{1, m}. \quad (1.82)$$

Имеем место следующие утверждения.

Утверждение 1. Задачи (1.32)–(1.35) и (1.81) эквивалентны в том смысле, что оптимальному решению Z_{opt} ЭЛЦП (1.81) соответствует оптимальное решение ЗЛЦП (1.32)–(1.35), вычисляемое по формуле

$$y_{j, opt} = \begin{cases} -z_{j, opt}, & j = \overline{1, n}; \\ z_{j, opt}, & j = n+1; \\ b_i - z_{j, opt}, & j = n+2, n+m+1, i = j-n-1; \end{cases} \quad (1.83)$$

$$x_{opt} = A_1^{-1} y_{opt}. \quad (1.84)$$

Справедливость утверждения очевидна.

Утверждение 2. Любому допустимому решению Z ЗЛЦП (1.81) со значением показателя качества, удовлетворяющего условию

$$\sum_{j=1}^{n+m+1} \hat{c}_j z_j \geq Lk - \sum_{j=1}^m \hat{c}_{n+1+j} b_j, \quad (1.85)$$

по формулам (1.83)–(1.84) соответствует допустимое решение ЗЛЦП (1.27), (1.28) и, наоборот, любому допустимому решению ЗЛЦП (1.27), (1.28), удовлетворяющему условию $\sum_{i=1}^r x_i \leq k$, соответствует допустимое решение задачи „о ранце“ (1.81) для которого выполняется (1.85).

Доказательство. Рассмотрим произвольное допустимое решение ЗЛЦП (1.81), удовлетворяющее условию (1.85). Поскольку функционалы ЗЛЦП (1.32)–(1.35) и (1.85) связаны зависимостями: $c_1^T x = \hat{c}_1^T y = \hat{c}_1^T z + \sum_{j=1}^m \hat{c}_{n+1+j} b_j$, то для целочисленного решения ЗЛЦП (1.32)–(1.35), вычисленного по формулам (1.83), (1.84), выполняется соотношение

$$c_1^T x \geq Lk.$$

Следовательно, $x_{n+1} = 0, \dots, x_{n+m+1} = 0$, т. е. полученное допустимое решение ЗЛЦП (1.32)–(1.35) является допустимым решением ЗЛЦП (1.27), (1.28).

Рассмотрим произвольное допустимое решение ЗЛЦП (1.27), (1.28), удовлетворяющее условию

$$\sum_{i=1}^n x_i \leq k,$$

Следовательно, $C^T x \geq Lk$. Равенство возможно только в случае

$$\forall x_i = 0, i \neq j; x_j = k, c_j = \min\{c_i\} = L.$$

Таким образом, рассматриваемому допустимому решению ЗЛЦП (1.27), (1.28) соответствует допустимое решение ЗЛЦП (1.32)–(1.35) вида $x_1, \dots, x_n, \underbrace{0, \dots, 0}_{m+1}$, которому соответствует согласно

$$y = A_1 x$$

и формуле (1.80) допустимое решение ЗЛЦП (1.81), удовлетворяющее условию (1.85).

Следствие. Если оптимальное решение ЗЛЦП (1.27), (1.28) удовлетворяет условию $\sum_{i=1}^n x_i \leq k$, то оптимальное решение ЗЛЦП (1.32)–(1.35), вычисленное по формулам (1.83), (1.84), является оптимальным решением ЗЛЦП (1.27), (1.28), а

$$Z_{j \text{ opt}} = \bar{x}_{j \text{ opt}}, j = \overline{1, n},$$

где $\bar{x}_j, j = \overline{1, n}$ – переменные ЗЛЦП (1.27), (1.28).

Доказательство. Первая часть следствия очевидна. Покажем, что $Z_j = \bar{x}_j, j = \overline{1, n}, \bar{x}_j, j = \overline{1, n}$ – переменные ЗЛЦП (1.27), (1.28). Действительно, в силу структуры матрицы A_1 (1.59) $y_j = -x_j, j = \overline{1, n}$, а в силу (1.80) –

$$Z_j = -y_j = x_j, j = \overline{1, n}.$$

Таким образом, ЗЛЦП (1.27), (1.28) может быть сведена к ЗЛЦП (1.81) с одним связным ограничением и неотрицательными переменными. Покажем, что ЗЛЦП (1.29)–(1.31) также может быть сведена к ЗЛЦП с одним связным ограничением.

Утверждение 3. При дополнительном условии $\sum_{i=1}^n x_i \leq k$ ЗЛЦП (1.29)–(1.31) сводится к ЗЛЦП с одним связным ограничением вида

$$\max \sum_{j=1}^{n+m+1} \hat{C}_j Z_j, \quad (1.86)$$

$$\sum_{j=1}^n (1 - p_0 t_{1j}) Z_j + p_0 \alpha_{11}^{-1} Z_{n+1} - p_0 \sum_{j=n+2}^{n+m+1} \alpha_{1(j-n)}^{-1} Z_j \leq$$

$$\leq k - p_0 \sum_{j=1}^m \alpha_{1(j+1)}^{-1} b_j, \quad (1.87)$$

$$Z_j \in D_j, j = \overline{1, n}. \quad (1.88)$$

При этом ЗЛЦП (1.86)–(1.88) отличается от ЗЛЦП (1.81) только ограничениями (1.88). Кроме того, $Z_j = \bar{x}_j, j = \overline{1, n}$, где $\bar{x}_j, j = \overline{1, n}$ – переменные ЗЛЦП (1.29)–(1.31).

Доказательство. Как следует из утверждений 1, 2 любому допустимому решению ЗЛЦП (1.29), (1.30) соответствует допустимое решение ЗЛЦП (1.86), (1.87), удовлетворяющее условию (1.85), и наоборот. При этом $Z_j = x_j, j = \overline{1, n}$, где $x_j, j = \overline{1, n}$ – переменные ЗЛЦП (1.29), (1.30).

Если при этом переменные $Z_j, j = \overline{1, n}$ удовлетворяют условию (1.88), то между допустимыми решениями ЗЛЦП (1.29)–(1.31) и допустимыми решениями ЗЛЦП (1.86)–(1.88), удовлетворяющими условию (1.85), также устанавливается взаимнооднозначное соответствие. В силу следствия утверждения 2 первые n компонент оптимального решения ЗЛЦП (1.86)–(1.88) являются оптимальным решением ЗЛЦП (1.29)–(1.31).

В заключение рассмотрим сведения к задаче „о ранце” ЗЛЦП:

$$\begin{aligned} & \max \sum_{j=1}^n C_j x_j ; \\ & \sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = \overline{1, m}, \quad \forall x_j \geq 0, \quad x_j \equiv 0 \pmod{1}; \quad (1.89) \\ & \sum_{j=1}^n a_{ij} x_j = b_{m+i}, \quad i = \overline{1, m-m_1}. \end{aligned}$$

В случае необходимости к ограничениям (1.89) могут быть добавлены такие $\forall x_j \in D_j$.

Аналог ЗЛЦП (1.32)–(1.35) для задачи (1.89) имеет вид

$$\begin{aligned} & \max \sum_{j=1}^n C_j x_j + M_0 x_{n+1} + M \sum_{j=n+2}^{n+m+1} x_j, \quad x_j \geq 0, \quad j = \overline{1, n}; \\ & \sum_{j=1}^n x_j + \rho_0 x_{n+1} \leq k; \\ & \sum_{j=1}^n a_{ij} x_j - f_i x_{n+1} + \rho_i x_{n+1+i} \leq b_i, \quad i = \overline{1, m_1}; \quad (1.90) \\ & \sum_{j=1}^n a_{ij} x_j - f_i x_{n+1} + \rho_i x_{n+1+i} = b_i, \quad i = \overline{m_1+1, m}, \\ & \quad \quad \quad x_j \equiv 0 \pmod{1}, \quad j = \overline{1, n+m+1}. \end{aligned}$$

Аналог ЗЛЦП (1.78) для ЗЛЦП (1.90):

$$\begin{aligned} & \max \sum_{j=1}^{n+1+m} \hat{C}_j y_j, \quad y_j \leq 0, \quad j = \overline{1, n}; \\ & \sum_{j=1}^n (\rho_0 t_{ij} - 1) y_j + \rho_0 \sum_{j=n+1}^{n+m+1} \alpha_{1(j-n)}^{-1} y_j \leq k, \quad (1.91) \\ & y_{n+i} \leq b_i, \quad i = \overline{1, m_1}, \quad y_{n+i} = b_i, \quad i = \overline{m_1+1, m}. \end{aligned}$$

Очевидным образом задача (1.91) преобразуется к виду

$$\begin{aligned} & \max \sum_{j=1}^{n+1+m} \hat{C}_j y_j, \quad y_j \leq 0, \quad j = \overline{1, n}; \\ & \sum_{j=1}^n (\rho_0 t_{ij} - 1) y_j + \rho_0 \sum_{j=n+1}^{n+1+m} \alpha_{1(j-n)}^{-1} y_j \leq \\ & \leq k - \rho_0 \sum_{j=n+1+m} \alpha_{1(j-n)}^{-1} b_{j-n-m_1}. \quad (1.92) \end{aligned}$$

К задаче (1.92) применимы с точностью до обозначения все исследования, проведенные в гл. 1, пп. 2, 3, 4 для задачи (1.78).

3. ИССЛЕДОВАНИЕ ЗАДАЧИ „О РАНЦЕ“

В п. 2 показано, что ЗЛЦП:

$$\max \sum_{j=1}^n c_j x_j ; \quad (1.93)$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = \overline{1, m}; \quad (1.94)$$

$\forall x_j \geq 0, x_j = 0 \pmod{1}, \forall x_j \in D_j, a_{ij}, b_i$ – целые числа (1.95)

может быть сведена к следующей ЗЛЦП с одним связным ограничением:

$$\max \sum_{j=1}^{n+m+1} \hat{c}_j z_j ; \quad (1.96)$$

$$\sum_{j=1}^n (1-p_0 t_{1j}) z_j + p_0 \alpha_{11}^{-1} z_{n+1} - p_0 \sum_{j=n+2}^{n+m+1} \alpha_{1(j-n)}^{-1} z_j \leq$$

$$\leq k - p_0 \sum_{j=1}^m \alpha_{1(j+1)}^{-1} b_j ; \quad \forall z_j \geq 0, j \neq n+1 ; \quad (1.97)$$

$$z_j \in D_j, \quad j = \overline{1, n}, \quad (1.98)$$

где p_0 удовлетворяет условию (1.36), k определяется вспомогательным неравенством $\sum_{i=1}^n x_i \leq k$,

$$\hat{c}_j = -\hat{c}_j, \quad j = \overline{1, n}; \quad \hat{c}_{n+1} = \hat{c}_{n+1}, \quad \hat{c}_{n+1+j} = -\hat{c}_{n+1+j}, \quad j = \overline{1, m}; \quad (1.99)$$

$$\hat{c}_i = -c_i + M_0 t_{1i} + M \sum_{j=2}^{m+1} t_{ji}, \quad i = \overline{1, n}; \quad \hat{c}_{n+i} = M_0 \alpha_{1i}^{-1} +$$

$$+ M \sum_{j=2}^{m+1} \alpha_{ji}^{-1}, \quad i = \overline{1, m+1};$$

M, M_0 удовлетворяют условиям (1.37).

$$\alpha_{11}^{-1} = (-1)^n \prod_{j=1}^m p_j ; \quad \alpha_{1j}^{-1} = (-1)^{n+1} \frac{\hat{a}_j}{p_{j-1}} \prod_{k=1}^m p_k, \quad j = \overline{2, m+1}; \quad (1.101)$$

$$t_{1j} = (-1)^n \left(1 - \sum_{k=2}^{m+1} \frac{\hat{a}_k}{p_{k-1}} a_{k-1,j} \right) \prod_{j=1}^m p_j, \quad j = \overline{1, n}, \quad (1.102)$$

где $p_i, i = \overline{1, m}$ удовлетворяют условию (1.36), $\hat{a}_1, \dots, \hat{a}_{m+1}$ и являются произвольным частным решением диофантова уравнения вида

$$\sum_{i=1}^{m+1} \hat{a}_i \lambda_{n+1, n+i} = 1 ; \quad (1.103)$$

где
$$\lambda_{n+1, n+i} = (-1)^n \prod_{j=1}^m p_j ; \quad (1.104)$$

$$\lambda_{n+1, n+i} = (-1)^n f_{i-1} \prod_{\substack{j=1 \\ j \neq i-1}}^m p_j, \quad i = \overline{2, m+1}. \quad (1.105)$$

Если ЗЛЦП (1.93)–(1.95) не содержит ограничения (1.95), то эквивалентная ЗЛЦП (1.96)–(1.98) не содержит ограничения (1.98). Первые n компонент z_j совпадают с x_j .

Основным научным результатом данного параграфа является доказательство утверждения о том, что ЗЛЦП (1.96)–(1.98) при фиксированном числе связанных ограничений (1.94) и ограниченности по модулю величин k, a_{ij}, b_i полиномами фиксированной степени от числа переменных ЗЛЦП (1.93)–(1.95) n может быть точно решена алгоритмом, число элементарных операций которого ограничено полиномом фиксированной степени от n (утверждение 5).

Для удобства дальнейшего изложения сделаем следующую переиндексацию переменных:

$$\begin{aligned} Z_i &= \omega_i, \quad i = \overline{1, n}; \quad Z_{n+1+i} = \omega_{n+i}, \quad i = \overline{1, m}; \\ Z_{n+1} &= \omega_{n+m+1}, \quad \hat{C}_i = d_i, \quad i = \overline{1, n}; \\ \hat{C}_{n+1+i} &= d_{n+i}, \quad i = \overline{1, m}; \quad \hat{C}_{n+1} = d_{n+m+1}. \end{aligned} \quad (1.106)$$

Тогда задача (1.96)–(1.97) переписывается так:

$$\begin{aligned} \max \sum_{j=1}^{n+m+1} d_j \omega_j; \\ \sum_{j=1}^n (1-p_0 t_{ij}) \omega_j - p_0 \sum_{j=n+2}^{n+m+1} \alpha_{1,j-n}^{-1} \omega_{j-1} + p_0 \alpha_{11}^{-1} \omega_{n+m+1} \leq \\ \leq k - p_0 \sum_{j=1}^m \alpha_{1,j+1}^{-1} b_j \Rightarrow \sum_{j=1}^{n+m+1} a_j \omega_j \leq b, \omega_j \geq 0, j = \overline{1, n+m}. \end{aligned} \quad (1.107)$$

Неравенство (1.85) примет вид

$$\sum_{j=1}^{n+m+1} d_j \omega_j \geq Lk + \sum_{j=1}^m d_{n+j} b_j. \quad (1.108)$$

Утверждение 1. Все допустимые целочисленные решения задачи (1.107), для которых выполняется (1.108), удовлетворяет следующим неравенствам:

$$\begin{aligned} 0 \leq \sum_{j=1}^n \omega_j \leq k, \quad 0 \leq \omega_j \leq k, \quad j = \overline{1, n}; \\ 0 \leq \omega_{n+i} \leq b_i + \max_j |a_{ij}^{(-)}| k, \quad i = \overline{1, m}, \quad 0 \leq \omega_{n+m+1} \leq k, \end{aligned} \quad (1.109)$$

где $a_{ij}^{(-)}$ — те из коэффициентов a_{ij} , которые удовлетворяют неравенству $a_{ij} < 0$.

Доказательство. В силу утверждения 2 (п. 2) всем допустимым целочисленным решениям задачи (1.107), для которых выполняется (1.108), соответствуют целочисленные допустимые решения задачи (1.32)–(1.35), удовлетворяющие условию

$$x_{n+1} = 0, \dots, x_{n+m+1} = 0.$$

В силу равенства $A_f x = y$ и ограничений (1.78) получаем цепочку неравенств

$$\begin{aligned} -k \leq \sum_{i=1}^n y_i \leq 0, \quad 0 \leq y_{n+1} \leq k, \quad y_{n+1+i} \leq b_i, \\ -\max_j |a_{ij}^{(-)}| k \leq y_{n+1+i} \leq \max_j |a_{ij}^{(+)}| k, \quad i = \overline{1, m+1}, \end{aligned}$$

где $a_{ij}^{(+)}$ – те из коэффициентов a_{ij} , которые удовлетворяют неравенству $a_{ij} \geq 0$. В силу (1.83)

$$0 \leq \sum_{j=1}^n z_j \leq k, \quad 0 \leq z_{n+1} \leq k, \quad 0 \leq z_{n+i} \leq b_i + m \max_j |a_{ij}^{(-)}| k, \quad i = \overline{1, m}.$$

Воспользовавшись равенствами (1.106), завершаем доказательство утверждения 1.

Утверждение 2. Задача (1.107) сводится к задаче „о ранце“, у которой коэффициенты $d_j, a_j, j = \overline{1, n+m+1}$ отличны от нуля и имеют совпадающие знаки.

Доказательство. 1. $a_{n+m+1} = (-1)^n \prod_{i=0}^m p_i$. Пусть $a_{n+m+1} \leq 0$.

Если $d_{n+m+1} > 0$, то задача (1.107) имеет неограниченное оптимальное решение, что невозможно.

Пусть $d_{n+m+1} = 0$. Если существует хотя бы один коэффициент $d_j > 0, j = \overline{1, n+m}$, то задача (1.107) имеет неограниченное оптимальное решение. Если все $d_j \leq 0, j = \overline{1, n+m}$, то одним из оптимальных решений задачи (1.107) является тривиальное $\omega_j \equiv 0, j = \overline{1, n+m+1}$.

Пусть $d_{n+m+1} > 0$. Если $d_{n+m+1} < 0$, то в силу законоопределенности переменной ω_{n+m+1} задача (1.107) имеет неограниченное решение. Если $d_{n+m+1} = 0$, то либо задача (1.107) имеет неограниченное оптимальное решение, что невозможно, либо одно из оптимальных решений тривиальное $\omega_j = 0, j = \overline{1, n+m+1}$. Следовательно, либо $d_{n+m+1} > 0, a_{n+m+1} > 0$, либо $d_{n+m+1} < 0, a_{n+m+1} < 0$. В противном случае найдено оптимальное решение задачи (1.107).

2. $\exists d_j < 0, a_j \geq 0, j = \overline{1, n+m}$.

В этом случае в оптимальном решении $\omega_j \equiv 0$, так как $\omega_j \geq 0$. Если $d_j = 0$, то существует оптимальное решение, в котором $\omega_j = 0$.

$$\exists d_j > 0, a_j < 0, j = \overline{1, n+m+1}.$$

В этом случае в силу того что $\omega_j \geq 0$, задача (1.107) имеет неограниченное оптимальное решение. Если $a_j = 0, d_j > 0$, задача (1.107), по-прежнему, неограничена. Если $d_j = 0, a_j < 0$, то оптимальное решение задачи (1.107) неограниченное, если существует $d_k > 0, k = \overline{1, n+m}$ либо $d_{n+m+1} \neq 0$. В противном случае оптимальным решением задачи (1.107) является тривиальное $\omega_j \equiv 0, j = \overline{1, n+m+1}$.

Таким образом, доказано, что либо $d_j < 0, a_j < 0$, либо $d_j > 0, a_j > 0, j = \overline{1, n+m+1}$.

Утверждение 3. Задача (1.107) может быть сведена к задаче „о ранце“ с положительными коэффициентами функционала и ограничения, а также неотрицательными ограниченными переменными.

Доказательство. На основании утверждения 1 любое целочисленное допустимое решение задачи (1.93), (1.94) удовлетворяет ограничениям

$$0 \leq \sum_{j=1}^n \omega_j \leq k, \quad 0 \leq \omega_{n+i} \leq b_i + \max_i |a_{ij}^{(-)}| k, \\ i = \overline{1, m}, \quad 0 \leq \omega_{n+m+1} \leq k.$$

Откуда $0 \leq \omega_j \leq k, j = \overline{1, n}$.

Пусть на основании утверждения 2 J – множество индексов, для каждого из которых $a_j < 0, d_j < 0$.

Тогда делаем замену переменных

$$\forall j \in J \quad \omega_j = k - \vartheta_j, \quad 0 \leq \vartheta_j \leq k, \quad j = \overline{1, n}, \quad j = n+m+1;$$

$$\omega_{n+j} = b_j + \max_i |a_{ji}^{(-)}| k - \vartheta_{n+j};$$

$$0 \leq \vartheta_{n+j} \leq b_j + \max_i |a_{ji}^{(-)}| k, \quad j = \overline{1, m}.$$

Тогда задача (1.107) переписывается в виде

$$\max \sum_{j=1}^{n+m+1} |d_j| \vartheta_j; \\ \sum_{j=1}^{n+m+1} |a_j| \vartheta_j \leq b - k \sum_{j \in J} a_j - \sum_{n+j \in J} a_{n+j} (b_j + \max_i |a_{ji}^{(-)}| k); \quad (1.110)$$

$$0 \leq \vartheta_j \leq k, \quad j = \overline{1, n}; \quad j = n+m+1;$$

$$0 \leq \vartheta_{n+j} \leq b_j + \max_i |a_{ji}^{(-)}| k, \quad j = \overline{1, m}. \quad (1.111)$$

Соотношение между функционалами задач (1.107) и (1.110):

$$\sum_{j=1}^{n+m+1} d_j \omega_j = \sum_{j=1}^{n+m+1} |d_j| \vartheta_j + k \sum_{j \in J} d_j + \\ + \sum_{\substack{n+j \in J \\ n+j \in \{\overline{n+1, n+m}\}}} d_{n+j} (b_j + \max_i |a_{ij}^{(-)}| k).$$

Если к задаче „о ранце” сводится ЗЛЦП (1.93)–(1.95), то к задаче (1.109) необходимо добавить ограничения

$$k - \vartheta_j \in D_j, \quad j = \overline{1, n}. \quad (1.112)$$

Утверждение 4. Существует частное решение диофантова уравнения (1.103), удовлетворяющее условию

$$|\hat{a}_i| \leq p_{i-1} - 1, \quad i = \overline{2, m+1}. \quad (1.113)$$

Доказательство. В [16] показано, что общее решение линейного диофантова уравнения $a_1 x_1 + \dots + a_n x_n = b$ с целочисленными коэффициентами имеет целочисленное решение, где b должно делиться на наибольший общий делитель d_n чисел a_1, \dots, a_n . Таким образом, $b = d_n t_n$, где t_n – произвольное целое число. Показано, что общее решение уравнения $a_1 x_1 + \dots + a_n x_n = d_n t_n, n \geq 2$, имеет вид

$$\begin{aligned} x_1 &= t_n x_1^{(n)} + t_{n-1} \frac{a_n}{d_n} x_1^{(n-1)} + t_{n-2} \frac{a_{n-1}}{d_{n-1}} x_1^{(n-2)} + \dots + t_1 \frac{a_2}{d_2}; \\ x_2 &= t_n x_2^{(n)} + t_{n-1} \frac{a_n}{d_n} x_2^{(n-1)} + t_{n-2} \frac{a_{n-1}}{d_{n-1}} x_2^{(n-2)} + \dots - t_1 \frac{d_1}{d_2}; \end{aligned} \quad (1.114)$$

$$x_{n-1} = t_n x_{n-1}^{(n)} + t_{n-1} \frac{a_n}{d_n} x_{n-1}^{(n-1)} - t_{n-2} \frac{d_{n-2}}{d_{n-1}};$$

$$x_n = t_n x_n^{(n)} - t_{n-1} \frac{d_{n-1}}{d_n},$$

где t_1, \dots, t_n — произвольные целые числа; $d_i = a_1, a_i$ — наибольший общий делитель чисел a_1, \dots, a_i , $i = \overline{2, n}$, $x_i^{(j)}$, $i, j = \overline{1, n}$ и представляет собой целочисленные решения уравнений:

$$\begin{aligned} d_2 &= a_1 x_1^{(2)} + a_2 x_2^{(2)}; \\ d_3 &= a_1 x_1^{(3)} + a_2 x_2^{(3)} + a_3 x_3^{(3)}; \end{aligned} \quad (1.115)$$

$$\dots$$

$$d_n = a_1 x_1^{(n)} + a_2 x_2^{(n)} + \dots + a_n x_n^{(n)}.$$

В нашем случае $n = m + 1$, $a_j = (-1)^n \prod_{j=1}^m p_j$, $a_i = (-1)^n \rho_i \prod_{j=1}^m p_j$, $i = \overline{2, m+1}$, так как в каждом уравнении (1.115) d_i является наибольшим общим делителем чисел a_1, \dots, a_i , то каждое из уравнений системы (1.115) имеет вид

$$a_{1i} x_1 + a_{2i} x_2 + \dots + a_{ii} x_i = 1$$

и, следовательно, его частное решение может быть получено с помощью алгоритма, изложенного в параграфе 2.

Пусть $x_1^{(2)}, x_2^{(2)}, x_1^{(3)}, \dots, x_3^{(3)}, \dots, x_1^{(n)}, \dots, x_n^{(n)}$ — набор целых чисел, удовлетворяющих системе (1.115). Найдем числа d_1, \dots, d_{m+1} :

$$d_1 = (-1)^n \prod_{j=1}^m p_j, \quad d_2 = (-1)^n \prod_{j=2}^m p_j, \quad d_3 = (-1)^n \prod_{j=3}^m p_j, \dots$$

$$\dots, \quad d_m = (-1)^n \rho_m, \quad d_{m+1} = (-1)^n.$$

Откуда

$$\frac{d_m}{d_{m+1}} = \rho_m, \quad \frac{d_{m-1}}{d_m} = \rho_{m-1}, \quad \frac{d_{m-2}}{d_{m-1}} = \rho_{m-2}, \dots,$$

$$\frac{d_2}{d_3} = \rho_2, \quad \frac{d_1}{d_2} = \rho_1.$$

Рассмотрим последнее уравнение системы (1.114): $t_{m+1} = 1, \frac{d_m}{d_{m+1}} = \rho_m, t_m$ — произвольное целое число. Выбираем t_m таким, чтобы x_{m+1} по модулю принимало наименьшее значение. Это можно сделать всегда, и при этом

$$|x_{m+1}| \leq \rho_m - 1.$$

Допустим, полученному значению x_{m+1} соответствует t_m^{\min} . Рассмотрим предпоследнее уравнение системы (1.114):

$$x_m = t_{m+1} x_m^{(m+1)} - t_m^{\min} \frac{d_{m+1}}{d_{m+1}} x_m^{(m)} - t_{m-1} \frac{d_{m-1}}{d_m},$$

где t_{m-1} — произвольное целое число.

В правой части последнего равенства все числа, кроме t_{m-1} известны, а $\frac{d_{m-1}}{d_m} = p_{m-1}$ следовательно, всегда можно выбрать такое целое число t_{m-1} , при котором

$$|x_m| \leq p_{m-1} - 1.$$

Аналогичным образом последовательно определяются x_{m-1}, \dots, x_2 , удовлетворяющие условию $|x_i| \leq p_{i-1} - 1, i = 2, m-1$. Утверждение доказано.

Утверждение 5. Рассмотрим ЗЛЦП общего вида, заданные в виде (1.93), (1.94) либо (1.93)–(1.95). Пусть фиксировано число связанных ограничений m и по модулю величины $\forall (a_{ij}, b_j)$, k ограничены полиномами фиксированной степени от n — числа переменных задачи (k определяется дополнительным ограничением $\sum_{i=1}^n x_i \leq k$). Тогда эквивалентные ЗЛЦП с одним

связным ограничением (1.81) или (1.110)–(1.112) соответственно могут быть решены с помощью задачи групповой минимизации (ЗЛЦП (1.81)) либо методом динамического программирования (ЗЛЦП (1.110)–(1.112)) с числом элементарных операций, ограниченным полиномом фиксированной степени от n .

Доказательство. Рассмотрим условия (1.36), которым должны удовлетворять различные простые числа $p_0, p_1, \dots, p_m, f_1, \dots, f_m$. Известно [4], что существуют положительные постоянные числа α и β такие, что при всех $n \geq 2$ выполняются неравенства

$$\alpha n \ln n < p_n < \beta n \ln n, \quad (1.116)$$

где p_n есть n -е простое число.

Пусть полиномы, ограничивающие по модулю величины $k, p_0, \dots, p_m, f_1, \dots, f_m$, являются полиномами фиксированной степени от n с целыми положительными коэффициентами. Тогда при произвольном k и фиксированном n каждый такой полином является натуральным числом.

В качестве числа p_0 выберем простое число вида

$$p_0 = p_0(n^{t_1}), \leq \beta 0(n^{t_1}) \ln(0(n^{t_1})), \quad (1.117)$$

где $k \leq 0(n^{t_1}), 0(n^{t_1})$ — полином фиксированной степени t_1 от n .

Найдем максимум

$$S = \max \{ p_0(n^{t_1}) + 1, [b_i - s_i(k + p_0(n^{t_1}))] + 1, i = \overline{1, m} \}.$$

Поскольку числа b_i, S_i, k ограничены по условию утверждения 5 полиномами фиксированной степени, а число $\rho_{0(n^t_1)} \leq \beta \cdot 0(n^t_1) \cdot 0(n^t_1)$, так как $\ln n \leq n$ то S ограничено полиномом фиксированной степени от n , который обозначим $0(n^t_2)$. Различные простые числа $p_1, \dots, p_m, f_1, \dots, f_m$ необходимо выбрать из условия

$$\forall \rho_i \geq 0(n^t_2); f_i \geq 0(n^t_2). \quad (1.118)$$

Рассмотрим верхнюю оценку разности между $(n+1)$ -м и n -м простым числом. В силу (1.116)

$$\rho_{n+1} - \rho_n \leq \beta(n+1) \ln(n+1) - \alpha n \ln n \leq \beta(n+1)^2 - \alpha n^2. \quad (1.119)$$

Необходимо получить $2m$ различных простых чисел, удовлетворяющих условию (1.118), т.е. получить числа

$$\rho_{0(n^t_2)+1}, \rho_{0(n^t_2)+2}, \dots, \rho_{0(n^t_2)+2m}. \quad (1.120)$$

При фиксированном m в силу (1.118) все числа (1.120) ограничены полиномами фиксированной степени от n . Действительно,

$$\rho_{0(n^t_2)+i} \leq \beta [0(n^t_2) + i]^2, \quad i = \overline{1, 2m}. \quad (1.121)$$

Рассмотрим ЗЛЦП (1.107), эквивалентную (1.81). Ее структура ограничений

$$\begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & \alpha_1 & \alpha_2 & \dots & \alpha_{m+n} & \rho_0 & \alpha_{11}^{-1} & \\ & & & & & & & \omega_1 \geq 0 \\ & & & & & & & \vdots \\ & & & & & & & \vdots \\ & & & & & & & \vdots \\ & & & & & & & \vdots \\ & & & & & & & \vdots \\ & & & & & & & \vdots \\ & & & & & & & \omega_{n+m} \geq 0 \\ & & & & & & & \vdots \\ & & & & & & & \vdots \\ & & & & & & & \vdots \\ & & & & & & & \omega_{n+m+1} \leq k - \rho_0 \sum_{j=1}^m \alpha_{1(j+1)} b_j, \\ & & & & & & & \end{bmatrix} \quad (1.122)$$

$$\rho_0 \alpha_{11}^{-1} = (-1)^n \prod_{i=0}^m \rho_i.$$

Следовательно, ЗЛЦП (1.107) является частным случаем ЗЛЦП вида (1.32)–(1.35). Поэтому, используя результаты п. 1 гл. 1, можно утверждать, что ЗЛЦП (1.107) за число элементарных операций, линейных относительно n , сводится к задаче групповой минимизации, которая может быть решена алгоритмом Ху с числом элементарных операций, не превышающим

$$2 \left(\prod_{i=0}^m \rho_i \right)^2. \quad (1.123)$$

Оценка (1.123) ограничена полиномом фиксированной степени относительно n – числа переменных задачи, поскольку числа $\rho_0, \rho_1, \dots, \rho_m$ могут быть выбраны различными, простыми, удовлетворяющими условию (1.36), но ограниченными по модулю полиномами фиксированной степени от n . Таким образом, первая часть утверждения 5 доказана.

Замечание 1. Реальный выбор простых различных чисел $p_0, p_1, \dots, p_m, f_1, \dots, f_m$, удовлетворяющих условию (1.36), всегда приводит к числам, значительно меньшим по величине, чем те, которые использовались в доказательстве первой части утверждения 5.

Замечание 2. ЗЛЦП (1.107) решать с помощью метода групповой минимизации Ху нецелесообразно. С вычислительной точки зрения удобней сводить к задаче групповой минимизации ЗЛЦП (1.32)–(1.35). В этом случае на выбор чисел p_i, f_i накладываются условия (1.36), что приводит к меньшему числу операций в асимптотическом алгоритме Ху.

Докажем вторую часть утверждения 5. Для этого рассмотрим связанное ограничение ЗЛЦП (1.110)–(1.112). В силу (1.101), (1.102)

$$|a_j| = |1 - p_0(-1)^n (1 - \sum_{k=2}^{m+1} \frac{\hat{a}_k}{p_{k-1}} a_{k-1,j}) \prod_{i=1}^m p_i|, \quad j = \overline{1, n}; \quad (1.124)$$

$$|a_{n+j}| = |(-1)^{n+1} p_0 \frac{\hat{a}_{j+1}}{p_j} \prod_{k=1}^m p_k|, \quad j = \overline{1, m}; \quad |a_{n+m+1}| = \prod_{i=0}^m p_i; \quad (1.125)$$

$$|a_j| = -a_j, \quad \forall j \in J. \quad (1.126)$$

Легко видеть, что в выражении (1.124)–(1.125) входят решения диофантова уравнения (1.103) \hat{a}_j при $j \geq 2$. Воспользуемся утверждением 4 и в качестве частного решения диофантова уравнения (1.103) возьмем то, для которого $|\hat{a}_i| \leq p_{i-1} - 1, i = \overline{2, m}$. Было показано, что величины чисел p_0, p_1, \dots, p_m ограничены по модулю полиномами фиксированной степени относительно n . Тогда, в силу условий утверждения 5, правая часть связанного ограничения ЗЛЦП (1.110)–(1.112)

$$b - k \sum_{\substack{j \in J \\ j \in \{\overline{1, n}, n+m+1\}}} a_j - \sum_{\substack{n+j \in J \\ n+j \in \{\overline{n+1, n+m}\}}} a_{n+j} (b_j + \max_i |a_{ji}^{(-)}| k)$$

является полиномом фиксированной степени от n .

Воспользуемся результатом, изложенным в [6]. Пусть дана задача сепарабельного целочисленного программирования вида

$$\max \sum_{j=1}^n f_j(x_j);$$

$$\sum_{j=1}^n a_j x_j \leq b, \quad b, a_j \geq 0, \quad j = \overline{1, n}, \quad x_j \geq 0, \quad x_j \equiv 0 \pmod{1}, \quad (1.127)$$

$b, a_j, j = \overline{1, n}$ – целые числа.

Тогда количество операций для решения задачи (1.127) методом динамического программирования ограничено величиной

$$\frac{(b+1)[(n-1)(b+2)+2]}{2}.$$

Рассмотрим ЗЛЦП вида

$$\max \sum_{j=1}^{n+m+1} |d_j| v_j;$$

$$\sum_{j=1}^{n+m+1} |a_j| v_j \leq b - k \sum_{\substack{j \in J \\ j \in \{\overline{1, n}, n+m+1\}}} a_j - \sum_{\substack{n+j \in J \\ n+j \in \{\overline{n+1, n+m}\}}} a_{n+j} (b_j + \max_i |a_{ji}^{(-)}| k), \quad (1.128)$$

$$v_j \geq 0, \quad j = \overline{1, n+m+1},$$

которая решается методом динамического программирования с числом операций не более чем

$$\frac{(b_1+1)[(n_1-1)(b_1-2)+2]}{2}, \quad (1.129)$$

где $b_1 = b - k \sum_{\substack{j \in J \\ j \in \{1, n, n+m+1\}}} a_j - \sum_{\substack{n+j \in J \\ n+j \in \{n+1, n+m\}}} a_{n+j} (b_j + m \max_i |a_{ji}^{(-)}| k)$,

$$n_1 = n + m + 1. \quad (1.130)$$

ЗЛЦП (1.110)–(1.112) отличается от ЗЛЦП (1.128) дополнительными ограничениями (1.111)–(1.112). Очевидно, использование этих ограничений в методе динамического программирования при решении ЗЛЦП (1.110)–(1.112) может только уменьшить верхнюю оценку числа элементарных операций (1.129). Показано, что величина b_1 ограничена полиномом фиксированной степени от n – числа переменных исходной ЗЛЦП. Следовательно, оценка (1.129) ограничена полиномом фиксированной степени от n .

Очевидно, величина памяти, необходимой для решения ЗЛЦП (1.110)–(1.112) методом динамического программирования, ограничена также полиномом фиксированной степени от n . Утверждение 5 доказано.

Замечание. ЗЛЦП (1.110)–(1.112) нецелесообразно решать методом динамического программирования, так как при большом m (числа связанных ограничений исходной ЗЛЦП) в силу формул (1.101), (1.102) величина b_1 может быть значительной, что может привести к нереализуемому объему вычислений и необходимой памяти. В п. 4 гл. 1 приводятся эффективные алгоритмы решения ЗЛЦП (1.110)–(1.112), число операций которых не связано с величинами a_i, b_1 .

Утверждение 6. ЗЛЦП (1.107) может быть сведена к задаче „о ранце” с положительными коэффициентами функционала и ограничения, а также неотрицательными ограниченными переменными. При этом коэффициенты ограничения не превышают

$$\prod_{j=0}^m p_j, \quad \text{а коэффициенты функционала не превышают величины}$$

$$\left| M_0 \alpha_{11}^{-1} + M \sum_{j=2}^{m+1} \alpha_{j1}^{-1} \right|.$$

Доказательство. Обозначим через J_1 множество индексов, для каждого из которых $a_j < 0$ либо $a_j > \prod_{j=0}^m p_j = |a_{n+m+1}|$, $j \neq n + m + 1$. Пусть при этом \bar{J}_1 обозначает количество индексов в множестве J_1 .

Проведем последовательно \bar{J}_1 унимодулярных преобразований вида:

$$\begin{bmatrix} \omega_{j_1} \\ \vdots \\ \omega_{n+m+1} \end{bmatrix} = \begin{bmatrix} 1 & & & \\ 0 & \ddots & & 0 \\ & j_1 & \dots & \\ & 0 & \ddots & 0 \\ & 0 & \dots & \lambda_{j_1} & 1 \end{bmatrix} \cdot \begin{bmatrix} u_{j_1}^1 \\ \vdots \\ u_{n+m+1}^1 \end{bmatrix}, \quad j_1 \in J_1; \quad (1.131)$$

$$\begin{bmatrix} u_{j_2}^1 \\ \vdots \\ u_{n+m+1}^1 \end{bmatrix} = \begin{bmatrix} 1 & & & \\ 0 & \ddots & & 0 \\ & j_2 & \dots & \\ & 0 & \ddots & 0 \\ & 0 & \dots & \lambda_{j_2} & 1 \end{bmatrix} \cdot \begin{bmatrix} u_{j_2}^2 \\ \vdots \\ u_{n+m+1}^2 \end{bmatrix}, \quad j_2 \in J_1; \quad (1.132)$$

$$\begin{bmatrix} u_{j_{\bar{j}_1}}^{\bar{j}_1-1} \\ \vdots \\ u_{n+m+1}^{\bar{j}_1-1} \end{bmatrix} = \begin{bmatrix} 1 & & & \\ 0 & \ddots & & 0 \\ & j_{\bar{j}_1} & \dots & \\ & 0 & \ddots & 0 \\ & 0 & \dots & \lambda_{j_{\bar{j}_1}} & 1 \end{bmatrix} \cdot \begin{bmatrix} u_{j_{\bar{j}_1}}^{\bar{j}_1} \\ \vdots \\ u_{n+m+1}^{\bar{j}_1} \end{bmatrix}, \quad j_{\bar{j}_1} \in J_1. \quad (1.133)$$

Числа λ_{j_i} , $i = \overline{1, \bar{j}_1}$ являются целыми. Линейные преобразования вида (1.131)–(1.133) обладают тем свойством, что они целые векторы переводят в целые и дробные в дробные. В соответствии с равенствами (1.131)–(1.133) ни одному вектору u^k , в состав которого входят дробные компоненты, не могут соответствовать векторы u^{k-1} , u^{k+1} с целочисленными компонентами.

Анализируя равенства (1.131)–(1.133), имеем:

$$\omega_i = u_i^1, \quad i = \overline{1, n+m}; \quad \omega_{n+m+1} = \lambda_{j_1} u_{j_1}^1 + u_{n+m+1}^1;$$

$$u_i^2 = u_i^1, \quad i = \overline{1, n+m}; \quad u_{n+m+1}^2 = \lambda_{j_2} u_{j_2}^2 + u_{n+m+1}^2$$

$$\text{или } \omega_{n+m+1} = \lambda_{j_1} u_{j_1}^1 + \lambda_{j_2} u_{j_2}^2 + u_{n+m+1}^2.$$

Окончательно

$$u_i^{\bar{j}_1} = \omega_i, \quad i = \overline{1, n+m}; \quad \omega_{n+m+1} = \sum_{i=1}^{\bar{j}_1} \lambda_{j_i} \omega_{j_i} + u_{n+m+1}^{\bar{j}_1}. \quad (1.134)$$

Обозначим через J_2 множество индексов, дополняющих J_1 до множества $\{1, 2, \dots, n+m\}$. Тогда (1.107) сводится к задаче

$$\max \left\{ \sum_{j \in J_2} d_j \omega_j + \sum_{j \in J_1} (d_j + \lambda_j d_{n+m+1}) \omega_j + d_{n+m+1} u_{n+m+1}^{\bar{j}_1}; \quad (1.135) \right.$$

$$\left. \sum_{j \in J_2} a_j \omega_j + \sum_{j \in J_1} (a_j + \lambda_j a_{n+m+1}) \omega_j + a_{n+m+1} u_{n+m+1}^{\bar{j}_1} \leq b, \right.$$

$$\left. \omega_j \geq 0, \quad j = \overline{1, n+m}. \right.$$

Как показано выше,

$$|a_{m+n+1}| = \prod_{j=0}^m p_j,$$

$$|d_{n+m+1}| = |M_0 \alpha_{11}^{-1} + M \sum_{j=2}^{m+1} \alpha_{j1}^{-1}|.$$

Выбираем целые $\lambda_j, j \in J_1$

$$0 \leq a_j + \lambda_j a_{n+m+1} \leq \prod_{j=0}^m p_j.$$

Это можно сделать всегда.

Покажем, что при этом

$$0 \leq d_j + \lambda_j d_{n+m+1} \leq |M_0 \alpha_{11}^{-1} + M \sum_{j=2}^{m+1} \alpha_{j1}^{-1}|. \quad (1.136)$$

Пусть неравенство (1.136) не выполняется. Предположим, что

$$d_j + \lambda_j d_{n+m+1} > |M_0 \alpha_{11}^{-1} + M \sum_{j=2}^{m+1} \alpha_{j1}^{-1}|.$$

Тогда заменяем λ_j на $\lambda_j \pm 1$ в зависимости от знака d_{n+m+1} . При этом

$$a_j + (\lambda_j \pm 1) a_{n+m+1} \leq 0; \quad d_j + (\lambda_j \pm 1) d_{n+m+1} > 0.$$

В этом случае (1.135) имеет неограниченно большое значение функционала, что невозможно. Пусть $d_j + \lambda_j d_{n+m+1} < 0$.

В этом случае в оптимальном решении (1.135) переменная ω_j равна нулю.

Пусть для некоторого $j \in J_2$

$$d_j > |M_0 \alpha_{11}^{-1} + M \sum_{j=2}^{m+1} \alpha_{j1}^{-1}|.$$

В этом случае, проведя еще одно унимодулярное преобразование для $\lambda_j = \pm 1$ (в зависимости от знака d_{n+m+1}), получим задачу с неограниченно большим значением функционала, что невозможно. На основании утверждения 2 $d_j > 0, \forall j \in J_2$.

Таким образом, показано, что в (1.135), изоморфной задаче (1.107), коэффициенты функционала и ограничений удовлетворяют неравенствам:

$$\begin{aligned} 0 &\leq \max\{d_j, j \in J_2; d_j + \lambda_j d_{n+m+1}, j \in J_1\} \leq \\ &\leq |M_0 (-1)^n \prod_{i=1}^m p_i + M \sum_{j=2}^{m+1} (-1)^n f_{i-1} \prod_{j=1}^m p_j|; \end{aligned} \quad (1.137)$$

$$0 \leq \max\{a_j, j \in J_2; d_j + \lambda_j d_{n+m+1}, j \in J_1\} \leq \prod_{j=0}^m p_j. \quad (1.138)$$

Покажем, что область, которой принадлежат допустимые целочисленные решения задачи (1.135), соответствующие всем допустимым целочисленным решениям задач (1.93), (1.94), является ограниченной.

На основании утверждения 1 переменные $\omega_j, j = \overline{1, m+n+1}$ могут принимать допустимые целочисленные решения в областях

$$\sum_{j=1}^n \omega_j \leq k, \quad \omega_j \geq 0 \quad \text{либо} \quad 0 \leq \omega_j \leq k, \quad j = \overline{1, n}; \quad (1.139)$$

$$0 \leq \omega_{n+j} \leq b_j + \max_i |a_{ji}^{(-)}| k, \quad j = \overline{1, m}. \quad (1.140)$$

В соответствии с (1.134)

$$u_{n+m+1}^{\bar{J}_1} = \omega_{n+m+1} - \sum_{i=1}^{\bar{J}_1} \lambda_{ji} \omega_{ji}.$$

Переменная ω_{n+m+1} принимает целочисленные значения в области

$$0 \leq \omega_{n+m+1} \leq k.$$

Разобьем множество J_1 на два непересекающихся подмножества J_1^1 и J_1^2 ($J_1 = J_1^1 \cup J_1^2$), по следующему признаку:

$$\forall j \in J_1^1, j \leq n, \quad \forall j \in J_1^2, j > n.$$

Множества J_1^1, J_1^2 разобьем на подмножества так:

$$J_1^1 = J_1^{11} \cup J_1^{12}; \quad J_1^2 = J_1^{21} \cup J_1^{22};$$

$$\forall j \in J_1^{11} \lambda_j > 0; \quad \forall j \in J_1^{12} \lambda_j < 0; \quad \forall j \in J_1^{21} \lambda_j > 0;$$

$$\forall j \in J_1^{22} \lambda_j < 0.$$

Тогда, очевидно,

$$\begin{aligned} -k \max_{j \in J_1^{11}} \lambda_j - \sum_{j \in J_1^{21}} \lambda_j (b_{j-n} + \max |a_{j-n, k}^{(-)}| k) &\leq u_{n+m+1}^{\bar{J}_1} \leq \\ &\leq k + (1 + \max_{j \in J_1^{12}} |\lambda_j| + \sum_{j \in J_1^{22}} |\lambda_j|) (b_{j-n} + \max |a_{j-n, k}^{(-)}| k). \end{aligned} \quad (1.141)$$

Пусть для определенности $a_{n+m+1} < 0, a_{n+m+1} < 0$. Для случая $a_{n+m+1} > 0, a_{n+m+1} > 0$ выкладки проводятся аналогично.

Положим

$$u_{n+m+1}^{\bar{J}_1} = \beta_1 - \omega_{n+m+1}^1; \quad 0 \leq \omega_{n+m+1}^1 \leq \beta_1 + |\beta_2|.$$

Тогда задача (1.135) переписывается в виде

$$\max \left\{ \sum_{j \in J_2} d_j \omega_j + \sum_{j \in J_1} (d_j + \lambda_j a_{n+m+1}) \omega_j + |d_{n+m+1}| \omega_{n+m+1}^1 + a_{n+m+1} \beta_1 \right\} \quad (1.142)$$

$$\sum_{j \in J_2} a_j \omega_j + \sum_{j \in J_1} (a_j + \lambda_j a_{n+m+1}) \omega_j + |a_{n+m+1}| \omega_{n+m+1}^1 \leq b - a_{n+m+1} \beta_1; \quad (1.143)$$

$$0 \leq \omega_j \leq k, \quad j = \overline{1, n}; \quad 0 \leq \omega_{n+j} \leq b_j + \max_i |a_{ji}^{(-)}| k, \quad j = \overline{1, m},$$

$$0 \leq \omega_{n+m+1}^1 \leq \beta_1 + |\beta_2|.$$

Утверждение 6 доказано.

Утверждение 7.

Задача (1.93), (1.94) может быть сведена к задаче „о ранце” с булевыми переменными и положительными коэффициентами функционала и ограничения.

Доказательство. Задача (1.107), а также задача (1.142), (1.143) сводятся к задаче „о ранце” с булевыми переменными с положительными коэффициентами функционала и ограничений следующим образом. Поскольку каждая переменная, входящая в задачи (1.107) и (1.142), (1.143), является неотрицательной и ограниченной, она заменяется суммой булевых переменных. Например, если $0 \leq x \leq d$, то $x = \sum_{i=1}^d x_i, \forall x_i \in \{0, 1\}$. Если в функционал и в связное ограничение задач (1.107), (1.142), (1.143) подставить вместо переменных их булевы аналоги, то получим две задачи „о ранце” с булевыми переменными, положительными коэффициентами функционала и ограничений, каждая из которых изоморфна исходной задаче линейного целочисленного программирования.

При создании эффективных приближенных алгоритмов решения задачи „о ранце” может быть полезным представление ее в виде, полученном Г. Брэдли [19], предложившим алгоритм, который позволяет последовательно сворачивать ограничения задачи линейного целочисленного программирования, заданной в конечных областях пространства переменных. В результате общая задача линейного целочисленного программирования сводится к задаче „о ранце”. Приведем краткое изложение и анализ алгоритма Г. Брэдли.

Сначала рассмотрим два равенства:

$$f^T x = \delta; \quad g^T x = \gamma, \quad (1.144)$$

$x \in C, x$ – целые числа, δ, γ – целые, f, g – целые n -векторы, C – ограниченное множество.

Рассмотрим преобразованную систему ограничений вида

$$(\alpha f + \beta g)^T x = \alpha \delta + \beta \gamma, \quad x \in C, \quad (1.145)$$

x – целые.

Введем следующие константы:

$$sp1 \equiv \sup (g^T x - \gamma); \quad x \in C, \quad \text{sgn}(\beta) [f^T x - \delta] \leq -|\beta|, \quad (1.146)$$

x – целые;

$\text{sgn}(\beta)$ – знак β (если $\beta = 0$, то $\text{sgn}(\beta) = 0$);

$$sp2 \equiv \sup (g^T x - \gamma), \quad x \in C, \quad \text{sgn}(\beta) [f^T x - \delta] \geq |\beta|, \quad (1.147)$$

x – целые;

$$sp3 \equiv \sup (f^T x - \delta), \quad x \in C, \quad \text{sgn}(\alpha) [g^T x - \gamma] \geq |\alpha|, \quad (1.148)$$

x – целые;

$$sp4 \equiv \sup (f^T x - \delta), \quad \bar{x} \in C, \quad \text{sgn}(\alpha) [g^T x - \gamma] \leq -|\alpha|, \quad (1.149)$$

x – целые.

Если задачи (1.146)–(1.149) не имеют допустимого решения или оно меньше чем ноль, соответствующий sp_i равен нулю. Определим IF_1, IF_2, IF_3, IF_4 с помощью замещения sup задачи (1.146)–(1.149) на inf . Если ни одна из задач не имеет допустимого решения или inf больше чем ноль, то положим соответствующий IF_i равный нулю.

Теорема 1 [19]. Если α, β являются взаимно простыми целыми и, по крайней мере, одно из следующих условий выполняется

$$A1. \alpha > sp_1; A2. \alpha < IF_1; A3. \beta > sp_3; A4. -\beta < IF_3,$$

и, по крайней мере, одно из следующих условий удовлетворяется

$$B1. -\alpha > sp_2; B2. -\alpha < IF_2; B3. \beta > sp_4; B4. \beta < IF_4,$$

множество решений (1.144) идентично множеству решений (1.145).

Теорема 1 позволяет произвольную задачу линейного целочисленного программирования, заданную в конечных областях фазового пространства, свести к задаче „о ранце”.

Рассмотрим задачу линейного целочисленного программирования вида

$$\begin{aligned} \max C^T x; \\ Ax = b, \quad 0 \leq x \leq d. \end{aligned} \quad (1.150)$$

Считаем, что $\forall x_j \geq 0$, а так как область ограничена, то существует конечный вектор d . В [19] показано, что теорема 1 справедлива, если коэффициенты α и β выбирать по области $0 \leq x \leq d$ при условии, что область C ей принадлежит. Тогда, последовательно применяя теорему 1 для замены двух ограничений одним, задачу (1.150) сведем к задаче „о ранце” вида

$$\begin{aligned} \max C^T x; \\ f^T x = \delta, \quad 0 \leq x \leq d, \end{aligned} \quad (1.151)$$

где $f^T x = \delta$ – результирующая строка ограничений в результате последовательного применения теоремы 1.

Теорема 2 [19]. Задача линейного целочисленного программирования

$$\begin{aligned} \max C^T x; \\ Ax = b, \quad 0 \leq x \leq d, \end{aligned} \quad (1.152)$$

x – целые,
может быть сведена к задаче линейного целочисленного программирования

$$\max (h^T x + \gamma); \quad (1.153)$$

$$\mu_1 \leq h^T x \leq \mu_2, \quad 0 \leq x \leq d,$$

x – целые.

Задачи (1.152), (1.153) имеют идентичные множества оптимальных решений, кроме того, совпадают и множества их допустимых целочисленных решений. $C^T x = h^T x + \gamma$ для любого целочисленно-допустимого решения.

Доказательство теоремы 2 приведем полностью, так как оно потребуется в дальнейшем.

Как было показано выше, задача (1.152) сводится к задаче (1.151) с помощью теоремы 1. Поскольку допустимая область является ограниченной, то существуют такие константы θ_1 и θ_2 , что задача (1.151) эквивалентна следующей задаче:

$$\begin{aligned} & \max x_0; \\ & x_0 - C^T x = 0; \quad f^T x = \delta, \quad 0 \leq x \leq d; \quad \theta_1 \leq x_0 \leq \theta_2, \end{aligned} \quad (1.154)$$

x, x_0 – целые. Применяем теорему 1 к задаче (1.154).

Положим $\alpha = 1$ и выбираем β таким, чтобы удовлетворялись условия теоремы 1. Тогда получим такую эквивалентную задачу:

$$\begin{aligned} & \max x_0, \\ & x_0 - (C - \beta f)^T x = \beta \delta; \quad 0 \leq x \leq d; \quad \theta_1 \leq x_0 \leq \theta_2, \end{aligned} \quad (1.155)$$

x, x_0 – целые.

Положим

$$h = C - \beta f; \quad \gamma = \beta \delta; \quad \mu_1 = \theta_1 - \beta \delta; \quad \mu_2 = \theta_2 - \beta \delta. \quad (1.156)$$

Заменяя x_0 в функционале и ограничениях, получаем

$$\begin{aligned} & \max \{ (C - \beta f)^T x + \beta \delta \}, \\ & 0 \leq x \leq d; \quad \mu_1 = \theta_1 - \beta \delta \leq (C - \beta f)^T x \leq \theta_2 - \beta \delta = \mu_2. \end{aligned}$$

Теорема доказана.

Задача (1.153) может быть сведена к задаче булевого „ранца“.

Общеизвестным недостатком алгоритма Г. Брэдли является следующий: стремительный рост коэффициентов при увеличении числа связанных ограничений. Качественная характеристика роста величины коэффициентов вектора h оценивается по следующей цепочке задач оптимизации:

$$\begin{aligned} \max_{0 \leq x \leq 1} (10^2 - 10x) &= 10^2; & \max_{0 \leq x \leq 1} (10^4 - 10^3 x) &= 10^4; \\ \max_{0 \leq x \leq 1} (10^8 - 10^7 x) &= 10^8; & \max_{0 \leq x \leq 1} (10^{16} - 10^{15} x) &= 10^{16}; \\ \max_{0 \leq x \leq 1} (10^{32} - 10^{31} x) &= 10^{32}; & \max_{0 \leq x \leq 1} (10^{64} - 10^{63} x) &= 10^{64} \end{aligned}$$

И т. д.

Астрономический рост величины коэффициентов вектора с увеличением числа ограничений исходной задачи линейного целочисленного программирования и является основным препятствием при его использовании для практических целей. Как было показано выше, предложен способ сведения задачи линейного целочисленного программирования общего вида, определенный в конечной области пространства переменных, к задаче „о ранце” вида

$$\begin{aligned} & \max \hat{C}^T x ; \\ & \sum_{i=1}^{n+m+1} \hat{a}_i x_i \leq \hat{b}, \quad x_j \geq 0, \quad j = \overline{1, n+m} \quad \forall x_j \in \mathcal{D}_j, \end{aligned} \quad (1.157)$$

где n – число переменных исходной задачи линейного целочисленного программирования; m – число связанных ограничений исходной задачи, в которую входит ограничение вида $\sum_{i=1}^n x_i \leq k$ (предполагается, что $\forall x_i \geq 0$).

Показано, что все допустимые решения исходной задачи линейного целочисленного программирования соответствуют допустимым решениям задачи линейного целочисленного программирования вида

$$\begin{aligned} & \max C^T y ; \\ & \sum_{i=1}^{n+m+1} a_i y_i \leq b, \\ & 0 \leq y_i \leq d_i, \quad i = \overline{1, n+m+1} \quad \forall a_i > 0 \end{aligned} \quad (1.158)$$

при условии $C^T y \geq L_1$. Величины коэффициентов векторов C ($C^T = (C_1, \dots, C_{n+m+1})$), a ($a^T = (a_1, \dots, a_{n+m+1})$) коэффициентов b, L_1 определяются параметрами, вычисляемыми по исходной задаче линейного целочисленного программирования. Очевидно, скорость роста коэффициентов задачи (1.158) качественно существенно меньшая, чем у коэффициентов в задаче „о ранце”, построенной по алгоритму Брэдли. Тем не менее с точки зрения создания приближенных алгоритмов задача „о ранце” Брэдли по своей структуре более простая, чем задача (1.158). Естественным является предложение объединить оба подхода. Иными словами использовать идеи алгоритма Брэдли к задаче (1.158), применяя свертку ограничений только один раз, для получения более простых задач линейного целочисленного программирования.

1. Задачу (1.158) преобразуем к виду

$$\sum_{i=1}^{n+m+1} a_i y_i + y_{n+m+2} = b; \quad C^T x + y_{n+m+3} = L_2 + M_1; \quad (1.159)$$

$$L_2 \geq L_1, \quad 0 \leq y_i \leq d_i, \quad i = \overline{1, n+m+1}, \quad 0 \leq y_{n+m+2} \leq b,$$

$$0 \leq y_{n+m+3} \leq M_1,$$

$\forall y_i$ – целые.

Константа M_1 определяется из условия

$$0 \leq M_1 \leq y_{opt} - L_2,$$

y_{opt} — оптимальное решение задачи (1.152) как задачи линейного программирования. Более просто $y'_{opt} \geq y_{opt}$ можно получить по величине оптимального решения задачи (1.157) как задачи линейного программирования. При этом необходимо помнить, что функционалы задач (1.157) и (1.158) для взаимно однозначных допустимых векторов x и y отличаются на константу. Все допустимые целочисленные векторы задачи (1.159) соответствуют допустимым векторам задачи (1.158) с показателем качества, большим или равным заданной константы L_2 .

К двум равенствам задачи (1.159) применим теорему 1. В результате получим задачу

$$f^T y = \gamma;$$

$$y^T = (y_1, \dots, y_{n+m+1}, y_{n+m+2}, y_{n+m+3}), 0 \leq y_i \leq d_i, i = \overline{1, n+m+1}; \quad (1.160)$$

$$0 \leq y_{n+m+2} \leq b; \quad 0 \leq y_{n+m+3} \leq M_1. \quad (1.161)$$

Таким образом, поиск допустимого решения задачи линейного целочисленного программирования (1.158) со значением показателя качества, не меньше заданной величины L_2 , свелся к решению диофантова уравнения (1.160) при ограничениях (1.161).

2. Представим задачу (1.158) в следующей эквивалентной форме:

$$\max x_0; \\ x_0 - c^T y = 0; \quad \sum_{i=1}^{n+m+1} a_i y_i + y_{n+m+2} = b; \quad 0 \leq y_i \leq d_i; \quad i = \overline{1, n+m+1}; \quad (1.162)$$

$$0 \leq y_{n+m+1} \leq b; \quad L_1 \leq L_2 \leq x_0 \leq y'_{opt} \geq y_{opt}.$$

Применим теорему 1 к задаче (1.162). Положим $\alpha = 1$ и выбираем β таким, чтобы удовлетворялись условия теоремы 1. Тогда получим такую эквивалентную задачу:

$$\max x_0; \\ x_0 - (c^1 - \beta a^1)^T y^1 = \beta b, \quad 0 \leq y_i \leq d_i, \quad i = \overline{1, n+m+1}, \quad (1.163) \\ 0 \leq y_{n+m+2} \leq b; \quad L_2 \leq x_0 \leq y'_{opt},$$

где

$$c^{1T} = (c_1, \dots, c_{n+m+1}, 0); \quad a^{1T} = (a_1, \dots, a_{n+m+1}, 1);$$

$$y^{1T} = (y_1, \dots, y_{n+m+1}, y_{n+m+2}).$$

Положим

$$h = c^1 - \beta a^1; \quad \gamma = \beta b; \quad \mathcal{H}_1 = L_2 - \gamma, \quad \mathcal{H}_2 = y'_{opt} - \gamma. \quad (1.164)$$

Заменяя x_0 в функционале и ограничениях, получаем:

$$\max \{h^T y^1 + \gamma\};$$

$$\pi_1 \leq h^T y' \leq \pi_2, \quad 0 \leq y_i \leq d_i, \quad i = \overline{1, n+m+1}, \quad (1.165)$$

$$0 \leq y_{n+m+2} \leq b,$$

y' – целый.

Как было отмечено выше, задача (1.165) может быть эффективно решена алгоритмом, изложенным в [19].

Кроме того, для поиска допустимого целочисленного программирования с показателем качества, превышающим заданный уровень, по задаче (1.165) может быть сформулирована следующая задача линейного целочисленного программирования.

Найти целый вектор y' , удовлетворяющий ограничениям

$$\pi_1 \leq h^T y' \leq \pi_2, \quad 0 \leq y_i \leq d_i, \quad i = \overline{1, n+m+1}, \quad (1.166)$$

$$0 \leq y_{n+m+2} \leq b.$$

Задача (1.166) позволяет строить эффективные приближенные алгоритмы ее решения.

Примечание. Рассмотрим модель (1.27), (1.28), заданную в виде

$$\max \sum_{j=1}^n c_j x_j; \quad (1.167)$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = \overline{1, m}; \quad \sum_{i=1}^n x_i \leq k \quad \forall x_j \geq 0, x_j \equiv 0 \pmod{1}.$$

Помимо свертки Брэдли, рассмотренной выше, к задаче можно применить свертку, рассмотренную в [1]. В работе приведены эффективные алгоритмы свертки неотрицательных целочисленных равенств вида

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = \overline{1, m} \quad (1.168)$$

$\forall x_j \geq 0, x_j \equiv 0 \pmod{1}, a_{ij}, b_i$ – целые числа.

Ограничения (1.167) могут быть сведены к модели (1.168) следующим образом.

Пусть J – множество индексов, для каждого из которых удовлетворяется условие:

$\forall i \in J$, существует $a_{ji} < 0$.

Тогда путем введения дополнительных равенств вида

$$x_i + y_i = k, \quad y_i \geq 0 \quad \forall i \in J,$$

а также дополнительных неотрицательных переменных $Z_j, j = \overline{1, m}$ превращающих неравенства (1.167) в равенства, модель ограничений (1.167) сводится к модели (1.168). Для этого для всех $i \in J$ в ограничениях (1.167) при отрицательных коэффициентах a_{ij} вместо x_i подставим $k - y_i$. После этого к построенной модели можно применять свертку [1]. Однако полученный „ранец“ не обладает специфическими свойствами (см. п. 4, гл. 1).

4. АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ ЛИНЕЙНОГО ЦЕЛОЧИСЛЕННОГО ПРОГРАММИРОВАНИЯ, ИСПОЛЬЗУЮЩИЙ ИНФОРМАЦИЮ, СОДЕРЖАЩУЮСЯ В ЭКВИВАЛЕНТНОЙ ЗАДАЧЕ „О РАНЦЕ”

Во втором и третьем параграфах настоящей главы показано, что ЗЛЦП общего вида может быть сведена к задаче с одним связным ограничением с положительными коэффициентами и неотрицательными переменными. Эквивалентная задача „о ранце” обладает следующим свойством: если коэффициенты исходной матрицы ограничений, содержащей фиксированное число связных ограничений, а также число k , входящее в дополнительное ограничение $\sum_{i=1}^n x_i \leq k$ по модулю ограничены полиномом фиксированной степени от n – числа переменных задачи, то методом динамического программирования эта задача решается с числом элементарных операций, ограниченное полиномом фиксированной степени от n . Как показано в п. 1 гл. 3, это обозначает, что выделенный класс задач дискретной оптимизации принадлежит классу P . Поскольку получены явные аналитические выражения для коэффициентов результирующей задачи „о ранце”, то заранее можно определить, имеет ли практический смысл решать конкретную ЗЛЦП методом динамического программирования. Очевидно, в случае большого числа связных ограничений метод динамического программирования является практически неприемлемым в силу чрезвычайно больших по величине коэффициентов результирующей задачи „о ранце”. Именно по этой причине до настоящего времени не нашли широкого практического применения все имеющиеся результаты для ЗЛЦП общего вида, основанные на агрегации целочисленных ограничений.

В данном параграфе излагается точный метод решения ЗЛЦП общего вида, имеющий экспоненциальную оценку сложности вычислений, однако при этом обладающий рядом специфических особенностей, которые позволяют эффективно применять его для решения ЗЛЦП общего вида достаточно большой размерности, приемлемой для большинства практических задач.

В результате анализа эквивалентной задачи „о ранце” вводятся так называемые „обратные приоритеты” основных переменных, которые позволяют определить порядок и желаемую величину переменных в исходной ЗЛЦП.

„Обратные приоритеты” позволяют определить одно из наиболее эффективных направлений, относительно которого по матрице ограничений исходной ЗЛЦП организуется формально экспоненциальный перебор для поиска оптимального решения. Однако большое число проведенных экспериментальных исследований для ЗЛЦП с матрицами ограничений, организуемых случайным образом, показало, что в силу эффективности выбранного направления перебора оптимальные либо близкие к оптимальному значению допустимые целочисленные решения получаются

в большинстве случаев на начальных этапах перебора. Это связано с тем, что перебор начинается с компромиссного изменения малоприоритетных в соответствии с „обратными приоритетами” основных переменных задачи без изменения значений высокоприоритетных переменных для получения дискретного решения. Необходимо подчеркнуть еще одну особенность метода.

По эквивалентной задаче „о ранце” формируются только „обратные приоритеты”, сам же направленный перебор осуществляется по исходной матрице ограничений, т. е. задача решается без использования больших чисел.

По эквивалентной задаче „о ранце” приводятся правила отсечения для предложенной схемы перебора, которые невозможно было бы сформулировать только по исходной ЗЛЦП общего вида. Однако большинство из этих правил при практическом счете могут не использоваться, так как приемлемое допустимое решение в большинстве случаев приведенного вычислительного эксперимента получается практически сразу. Это замечание имеет особое практическое значение в тех случаях, когда ЛПР (лицо, принимающее решение) задаст границу значения показателя качества, превышение которой на допустимом целочисленном решении является удовлетворительным для ЛПР решением, т. е. ЗЛЦП формируется как задача распознавания.

В заключение отметим, что алгоритм излагается для ЗЛЦП вида (1.93), (1.94), на ЗЛЦП (1.93)–(1.95) он распространяется естественным образом.

Распишем кратко ход преобразования исходной ЗЛЦП. На первом этапе ЗЛЦП с прямоугольной матрицей ограничений неравенств вида

$$\max \sum_{j=1}^n C_j x_j ; \quad (1.169)$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = \overline{1, m}; \quad (1.170)$$

$$x_j \geq 0 \text{ — целые,} \quad (1.171)$$

$a_{ij}, b_i, C_j > 0$ — целые числа, которые дополняются вспомогательным ограничением

$$\sum_{j=1}^n x_j \leq k, \quad (1.172)$$

где k — некоторая константа, определяемая практическим содержанием задачи, преобразуется к ЗЛЦП с квадратной матрицей ограничений неравенств:

$$\max \sum_{j=1}^n C_j x_j + M_0 x_{n+1} + M \sum_{j=n+2}^{n+m+1} x_j, \quad (1.173)$$

$$x_j \geq 0, \quad j = \overline{1, n}; \quad (1.174)$$

$$\sum_{j=1}^n x_j + \rho_0 x_{n+1} \leq k ; \quad (1.175)$$

$$\sum_{j=1}^n a_{ij} x_j - f_i x_{n+1} + p_i x_{n+m+1} \leq b_i, \quad i = \overline{1, m}; \quad (1.176)$$

$$x_j \equiv 0 \pmod{1}, \quad j = \overline{1, n+m+1}, \quad (1.177)$$

где x_{n+i} , $i = \overline{1, m+1}$ – дополнительные переменные, коэффициенты при которых выбираются из условий:

$$p_0 > k, \quad p_i > b_i - s_i k, \quad f_i > b_i - s_i(k + p_0), \quad i = \overline{1, m};$$

$$M > (U - L)k, \quad M_0 > U p_0;$$

$$U = \max(\max_{1 \leq j \leq n} c_j, 0); \quad (1.178)$$

$$L = \min(\min_{1 \leq j \leq n} c_j, 0);$$

$$s_i = \min(\min_{1 \leq j \leq n} a_{ij}, 0), \quad i = \overline{1, m}.$$

При этом для всякого решения x^* , удовлетворяющего системе ограничений (1.174)–(1.177), при котором

$$\sum_{j=1}^n c_j x_j^* + M_0 x_{n+1}^* + M \sum_{i=1}^m x_{n+i}^* \geq Lk, \quad (1.179)$$

справедливы условия

$$x_j^* = 0, \quad j = \overline{n+1, n+m+1}.$$

Доказательство этого утверждения приведено в п. 1.

Примечание. Условие $\forall c_j \geq 0$ не снижает общности класса ЭЛЦП (1.169)–(1.171). Действительно, пусть J_1 – множество индексов, для которых $c_j < 0$. Тогда в силу ограничения $\sum_{i=1}^n x_i \leq k$ делаем замену переменных

$$x_j = k - y_j, \quad 0 \leq y_j \leq k, \quad \forall j \in J_1.$$

ЗЛЦП (1.169)–(1.172) заменяется эквивалентной:

$$\max \left\{ \sum_{j \in J_1} (-c_j y_j) + \sum_{j \in \bar{J}_1} c_j x_j \right\},$$

$$-\sum_{j \in J_1} a_{ij} y_j + \sum_{\substack{j=1 \\ j \in \bar{J}_1}}^n a_{ij} x_j \leq b_i - \bar{J}_1 k, \quad i = \overline{1, m};$$

$$\sum_{j \in J_1} y_j - \sum_{\substack{j=1 \\ j \in \bar{J}_1}}^n x_j \geq (\bar{J}_1 - 1)k;$$

$$\forall j \in J_1, \quad 0 \leq y_j \leq k, \quad \forall x_j \geq 0;$$

$$\sum_{j \in J_1} y_j + \sum_{\substack{j=1 \\ j \in \bar{J}_1}}^n x_j \leq (\bar{J}_1 + 1)k,$$

где \bar{J}_1 – количество индексов в множестве J_1 . Полученная ЗЛЦП принадлежит классу ЗЛЦП (1.93)–(1.95), на которую, как было указано выше, излагаемый метод естественным образом распространяется.

сводим к задаче „о ранце“

$$\begin{aligned} \max & \sum_{j=1}^{n+m+1} \hat{c}_j z_j, \\ z_j & \geq 0, \quad j = \overline{1, n}, \\ \sum_{j=1}^{n+m+1} t_j z_j & \leq k, \quad k = k - \sum_{i=1}^m \hat{t}_{n+i} b_i, \\ z_{n+2} & \geq 0, \\ & \vdots \\ z_{n+m+1} & \geq 0. \end{aligned} \quad (1.186)$$

Нижняя граница функционала ЗЛЦП (1.186), определяющая допустимое решение для ЗЛЦП (1.169)–(1.172) согласно (1.179) и проведенным заменам переменных (1.180) и (1.185)

$$L_1 = Lk - \sum_{j=1}^m \hat{c}_{n+1+j} b_j. \quad (1.187)$$

Затем ЗЛЦП (1.186) сводится к задаче „о ранце“ с положительными коэффициентами ограничения и функционала и неотрицательными ограниченными переменными.

Для удобства изложения произведем переобозначение переменных, переставляя незнакоопределенную переменную

$$z'_j = z_j, \quad j = \overline{1, n}, \quad z'_j = z_{j+1}, \quad j = \overline{n+1, n+m}, \quad z'_{n+m+1} = z_{n+1}, \quad (1.188)$$

и соответствующие перестановки среди коэффициентов ограничений и функционала.

Далее произведем замену переменных

$$\bar{z}' = P \bar{w}, \quad (1.189)$$

где P – унимодулярная целочисленная матрица вида

$$P = \begin{bmatrix} 1 & & & & & \\ 0 & 1 & & & & 0 \\ \vdots & & \ddots & & & \\ \lambda_1 & \lambda_2 & \dots & \lambda_{n+m-1} & \lambda_{n+m} & 1 \end{bmatrix} \quad (1.190)$$

где $\lambda_j, j = \overline{1, n+m}$ выбираются из условия

$$0 \leq t'_j + \lambda_j t'_{n+m+1} \leq |t'_{n+m+1}|. \quad (1.191)$$

В результате замены (1.189) получаем такую ЗЛЦП:

$$\begin{aligned} \max & \sum_{j=1}^{n+m+1} d_j w_j, \\ w_j & \geq 0, \quad j = \overline{1, n+m}; \\ \sum_{j=1}^{n+m+1} s_j w_j & \leq k_1, \end{aligned} \quad (1.192)$$

где

$$\begin{aligned} 0 \leq s_j & \leq |s_{n+m+1}| = |t'_{n+m+1}| = \prod_{i=0}^m p_i, \quad j = \overline{1, n+m}; \\ 0 \leq d_j & \leq |d_{n+m+1}| = |\hat{c}'_{n+m+1}|, \quad j = \overline{1, n+m}. \end{aligned}$$

Согласно (1.141) вычисляем границы на переменную w_{n+m+1}

$$\beta_1 \leq W_{n+m+1} \leq \beta_2 \quad (1.193)$$

и учетом знака S_{n+m+1} делаем следующую замену:

$$а) S_{n+m+1} > 0 \Rightarrow y_{n+m+1} = W_{n+m+1} - \beta_1, \quad (1.194а)$$

$$б) S_{n+m+1} < 0 \Rightarrow y_{n+m+1} = \beta_2 - W_{n+m+1}. \quad (1.194б)$$

Следовательно,

$$0 \leq y_{n+m+1} \leq \beta_2 - \beta_1. \quad (1.195)$$

Задача (1.193) тогда примет вид

$$\begin{aligned} \max \sum_{j=1}^{n+m} d_j w_j + |d_{n+m+1}| y_{n+m+1}; \\ \sum_{j=1}^{n+m} s_j w_j + |s_{n+m+1}| y_{n+m+1} \leq B, \\ w_j \geq 0, j = \overline{1, n+m}, y_{n+m+1} \geq 0, \end{aligned} \quad (1.196)$$

где, согласно (1.194а), (1.194б),

$$B = \begin{cases} k_1 - S_{n+m+1} \beta_1, \\ k_1 - S_{n+m+1} \beta_2 \end{cases} \quad (1.197)$$

или

$$\begin{aligned} \max \sum_{j=1}^{n+m+1} f_j y_j; \\ \sum_{j=1}^{n+m+1} A_j y_j \leq B, \\ y_j \geq 0, j = \overline{1, n+m+1}, \\ y_j \equiv 0 \pmod{1}, j = \overline{1, n+m+1}. \end{aligned} \quad (1.198)$$

Нижняя граница функционала ЗЛЦП (1.198), определяющая допустимое решение задачи (1.169)–(1.172) с учетом (1.187) и проведенных преобразований (1.190), а также в соответствии с (1.194а), (1.194б), будет иметь вид

$$L_k = \begin{cases} L_1 - d_{n+m+1} \beta_1, \\ L_1 - d_{n+m+1} \beta_2, \end{cases} \quad (1.199)$$

т. е. для всякого решения $y^* = (y_j^*)_1^{n+m+1}$, удовлетворяющего ограничениям задачи (1.198), при котором

$$\sum_{j=1}^{n+m+1} f_j y_j^* \geq L_k,$$

при возврате к переменным задачи (1.173)–(1.177) будут выполняться условия

$$x_j = 0, j = \overline{n+1, n+m+1},$$

а $x_j = y_j$ для $j = \overline{1, n}$.

Справедливо и обратное. Любому допустимому решению задачи (1.169)–(1.172) соответствует решение задачи (1.198) со значением показателя качества выше значения L_k .

В дальнейшем переменные с индексом $j = \overline{1, n}$ будем называть основными, а с индексами $j = \overline{n+1, n+m+1}$ — дополнительными переменными.

Отметим существование аналитических зависимостей между основными и дополнительными переменными в решениях ЗЛЦП (1.198) со значением показателя качества выше величины L_k :

$$y_{n+i} = b_i - \sum_{j=1}^n a_{ij} y_j, \quad i = \overline{1, m};$$

соответственно в случаях (1.194, а) и (1.194, б)

$$y_{n+m+1} = \begin{cases} \sum_{j=1}^n y_j - \sum_{j=1}^{n+m} \lambda_j y_j - \beta_1, \\ \beta_2 = \sum_{j=1}^n y_j + \sum_{j=1}^{n+m} \lambda_j y_j. \end{cases} \quad (1.200)$$

Для основных переменных задачи (1.198) выполняется также ограничение

$$\sum_{j=1}^n y_j \leq k.$$

Отметим, что на комбинациях из только дополнительных переменных нельзя получить допустимое решение задачи (1.198) с показателем качества выше величины L_k , так как в этом случае $x_j = 0, j = \overline{1, n}$.

Существенным недостатком задачи (1.198) является большая величина коэффициентов ограничения и функционала, что делает невозможным применение на практике для решения стандартных приемов и методов, в частности метода динамического программирования, который является наиболее эффективным для подобных задач.

Определение. Прямым приоритетом p_j переменной y_j в задаче (1.198) является отношение f_j к $A_j, j = \overline{1, n+m+1}$:

$$p_j = \frac{f_j}{A_j}, \quad j = \overline{1, n+m+1}.$$

Утверждение 1. Наибольший прямой приоритет в задаче (1.198) имеет переменная y_{n+m+1} .

Доказательство. Для доказательства рассмотрим задачу „о ранце” (1.192)

$$\max \sum_{j=1}^{n+m+1} d_j w_j$$

$$\begin{bmatrix} 1 & & & & & & & & & \\ & 1 & & & & & & & & 0 \\ & & \ddots & & & & & & & \\ & 0 & & \ddots & & & & & & 1 \\ S_1 & S_2 & \dots & S_{n+m} & S_{n+m+1} & & & & & \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{n+m+1} \end{bmatrix} \begin{cases} \geq 0, \\ \geq 0, \\ \vdots \\ \leq k_1. \end{cases}$$

Решим ее как задачу линейного программирования. Тогда все неравенства будут выполняться как строгие равенства, так как матрица квадратная и, следовательно, $w_j = 0, j = \overline{1, n+m}$, $w_{n+m+1} \neq 0$ значит, ей соответствует максимальный приоритет.

Поскольку преобразование (1.194) величины коэффициентов не изменяет, то максимальный приоритет в задаче (1.198) у переменной y_{n+m+1} .

Утверждение доказано.

Следствие. Наибольшее значение функционала в задаче (1.198) равно $p_{n+m+1} B$ на решении

$$y_{n+m+1} = \frac{B}{A_{n+m+1}}, \quad y_j = 0, \quad j = \overline{1, n}.$$

Обозначим $p_* = p_{n+m+1}$. Отметим, что решение $y_{n+m+1} = \left\lfloor \frac{B}{A_{n+m+1}} \right\rfloor$, $y_j = 0, j = \overline{1, n}$, будет недопустимым исходя из соотношений между задачами и переменными, и, следовательно, $f_{n+m+1} \left\lfloor \frac{B}{A_{n+m+1}} \right\rfloor < L_K$.

При поиске решений задачи „о ранце” получил распространенный прием упорядочения переменных по убыванию величин p_j . Однако он предполагает независимость переменных, и решение ищется на комбинациях наиболее приоритетных переменных. А так как в данном случае существуют аналитические зависимости между переменными и в решении должны присутствовать основные переменные, которые не являются наиболее приоритетными, то применение этого приема нецелесообразно. Кроме того, как показывают вычислительные эксперименты, дополнительные переменные оставляют основным переменным чрезвычайно малый резерв, что приводит к практически полному перебору возможных комбинаций.

Возникает задача определения приоритетов переменных, учитывающих специфику данной задачи.

В приведенных ниже утверждениях не используются аналитические зависимости между основными и дополнительными переменными, а только фиксируется факт, что в оптимальном решении задачи (1.198) обязательно присутствуют основные и дополнительные переменные. Не учитываются также ограничения на переменные. Основное назначение приведенных ниже утверждений – определение эффективного направления поиска оптимального решения.

Утверждение 2. Пусть Z^* значение функционала на оптимальном решении задачи (1.198), тогда верхняя граница максимального значения функционала исходной задачи ЛЦП (1.169)–(1.171) при данном Z^* может быть получена на одной основной и одной дополнительной переменной.

Доказательство. При анализе решений будем рассматривать поочередно все основные переменные и наиболее приоритетную дополнительную переменную y_{n+m+1} .

Выполним следующие аналитические выкладки для каждой основной переменной $j = \overline{1, n}$:

$$p_*(B - V_j) + p_j V_j = Z^* \quad (1.201)$$

$$d_j = \frac{V_j}{A_j} \quad (1.202)$$

$$S_j = d_j C_j \quad (1.203)$$

Найдем $S^* = S_j^* = \max_{1 \leq j \leq n} S_j$. (1.204)

Таким образом, S^* – наибольшее из S_j , с индексом j^* – это максимальное значение исходного функционала, если получать решение из одной основной и одной дополнительной переменной с приоритетом p_* .

Проанализируем, что будет со значением функционала, если взять не одну основную переменную, а комбинацию из основных переменных, например, переменных с индексами j^* и l .

Пусть количество переменной Y_{j^*} в „смеси“ $d_{j^*}^c < d_{j^*}$. Тогда из (1.202) следует, что $V_{j^*}^c < V_{j^*}$, а из (1.204) получаем

$$\begin{aligned} S^* = S_{j^*} > S_l &\Rightarrow S_{j^*} - S_l > 0 \Rightarrow \\ \Rightarrow \frac{C_{j^*}}{A_{j^*}} V_{j^*} - \frac{C_l}{A_l} V_l &= \frac{1}{A_{j^*} A_l} (C_{j^*} A_l V_{j^*} - C_l A_{j^*} V_l) = \\ = \frac{1}{A_{j^*} A_l} (C_{j^*} A_l \frac{p_* B - z^*}{p_* - p_{j^*}} - C_l A_{j^*} \frac{p_* B - z^*}{p_* - p_l}) &= \\ = \frac{p_* B - z^*}{A_{j^*} A_l} (\frac{C_{j^*} A_l}{p_* - p_{j^*}} - \frac{C_l A_{j^*}}{p_* - p_l}) > 0 \Rightarrow \\ \Rightarrow \frac{C_{j^*} A_l}{p_* - p_{j^*}} - \frac{C_l A_{j^*}}{p_* - p_l} > 0 \end{aligned} \quad (1.205)$$

Определим $V_l^c > 0$.

$$p_* (B - V_{j^*}^c - V_l^c) + p_{j^*} V_{j^*}^c + p_l V_l^c = z^*,$$

$$V_l^c = \frac{p_* B - (p_* - p_{j^*}) V_{j^*}^c - z^*}{p_* - p_l} > 0. \quad (1.206)$$

Пусть S_l – значение функционала исходной задачи ЛЦП на комбинации из переменных с индексами j и l . Тогда

$$S_l = d_{j^*}^c C_{j^*} + d_l^c C_l.$$

Найдем знак разности $S^* - S_l$:

$$\begin{aligned} S^* - S_l &= d_{j^*} C_{j^*} - d_{j^*}^c C_{j^*} - d_l^c C_l = \frac{C_{j^*}}{A_{j^*}} (V_{j^*} - V_{j^*}^c) - \frac{C_l}{A_l} V_l^c = \\ &= \frac{C_{j^*}}{A_{j^*}} \cdot \frac{p_* B - (p_* - p_{j^*}) V_{j^*}^c - z^*}{p_* - p_{j^*}} - \frac{C_l}{A_l} \cdot \frac{p_* B - (p_* - p_{j^*}) V_{j^*}^c - z^*}{p_* - p_l} = \\ &= \frac{1}{A_{j^*} A_l} (p_* B - (p_* - p_{j^*}) V_{j^*}^c - z^*) (\frac{C_{j^*} A_l}{p_* - p_{j^*}} - \frac{C_l A_{j^*}}{p_* - p_l}). \end{aligned} \quad (1.207)$$

Второй сомножитель выражения (1.207) больше нуля согласно (1.206), так как $p_* - p_l > 0$ из утверждения 1. Выражение в скобках больше нуля согласно (1.205). Таким образом, выражение (1.207) больше нуля, и, следовательно, значение функционала исходной ЗЛЦП, при сделанных допущениях, на комбинации из основных переменных меньше, чем на одной основной переменной. Наличие только одной высокоприоритетной переменной Y_{n+1} в „смеси“ доказывается аналогично.

Действительно, пусть в „смеси“ находится дополнительная переменная с приоритетом $p' < p_*$ и основная с приоритетом p_j и V_j – объем, удовлетворяющий равенству,

$$p'(B - V_j) + p_j V_j = Z^* .$$

Тогда, если объем $B - V_j$ будет занимать U_{n+m+1} , то

$$p_*(B - V_j) + p_j V_j > Z^*$$

и V_j можно увеличить на величину

$$\frac{(B - V_j)(p_* - p')}{p_* - p_j} ,$$

что приводит к увеличению показателя качества исходной ЗЛЦП при $C_j > 0$.

Приведенные выкладки очевидным образом позволяют утверждать, что если в „смеси” находятся как дополнительные переменные, так и основные, то, заменяя дополнительные переменные переменной U_{n+m+1} , можно лишь увеличить значение показателя качества исходной ЗЛЦП, а, как показано было выше, при одной дополнительной переменной U_{n+m+1} оптимальным является введение переменной U_{j^*} .

Утверждение доказано.

Пусть $S_{j_1} \geq S_{j_2} \geq \dots \geq S_{j_n}$ — последовательность, полученная по утверждению 2. Обозначим порядок $\{j_1, j_2, \dots, j_n\}$ через I_ϕ .

Обоснуем способ вычисления последовательности I_ϕ .

Утверждение 3. Последовательность I_ϕ может быть получена при любом $Z < p_* B$.

Доказательство. Проанализируем выражение для вычисления S_j :

$$S_j = \frac{C_j}{A_j(p_* - p_j)} (p_* B - Z^*) ,$$

так как $Z^* < p_* B$ согласно следствию утверждения 1, то $(p_* B - Z^*) > 0$. Однако $(p_* B - Z) > 0$ при любом $Z < p_* B$. А поскольку первый сомножитель от Z не зависит, то утверждение доказано.

С учетом ограничений исходной ЗЛЦП и ограничений на переменные нам не удастся загрузить переменную X_{j^*} в количестве d_{j^*} , но при помощи утверждения 2 можно обосновать порядок загрузки основных переменных.

Первой загружается переменная с индексом j_1 в максимально большом количестве с учетом ограничений ЗЛЦП (1.169)–(1.172), затем — переменная с индексом j_2 с учетом тех же ограничений и т. д. Описанная процедура заканчивается получением допустимого решения либо нарушением ограничений исходной ЗЛЦП. После этого проводится процедура дополнительного перебора с описанными ниже правилами отсечения для улучшения полученного допустимого решения либо получения допустимого решения путем компромиссного перебора на малоприоритетных переменных с последующим его улучшением.

Утверждение 4. Значение основной переменной в оптимальном решении задачи (1.198) не превышает следующей величины:

$$y_j \leq \frac{p_* B - L_k}{p_* A_j - f_j}, \quad j = \overline{1, n} \quad (1.208)$$

Доказательство. Приведем величины

$$\alpha_j = (p_* - p_j) A_j, \quad j = \overline{1, n},$$

которые можно рассматривать как ухудшение значения функционала качества за счет заполнения объема A_j переменной y_j , а не y_{n+m+1} .

Согласно примечаниям к задаче (1.198) значение функционала должно быть больше L_k , а максимально возможное значение функционала, согласно следствию утверждения 1, равно $p_* B$, тогда справедливо следующее выражение:

$$p_* B - \alpha_j y_j \geq L_k,$$

преобразование которого доказывает утверждение.

Суммируя результаты утверждений 2-4, определим величины

$$\phi_j = \frac{p_* B - L_k}{p_* A_j - f_j} c_j, \quad j = \overline{1, n} \quad (1.209)$$

Утверждение 5. Пусть имеется последовательность $I_\phi = \{j_1, j_2, \dots, j_n\}$ и соответствующее значение $\phi_{j_l}, l = \overline{1, n}$, тогда справедливы следующие высказывания:

- оптимальное значение функционала задачи (1.169)-(1.172) не превышает ϕ_{j_1} или $k C_{\max}$, где $C_{\max} = \max_{1 \leq j \leq n} C_j$.

- оптимальное значение функционала задачи (1.169)-(1.172) не превышает $\phi_{j_l}, l = \overline{2, n}$ или $k C_{\max}^l$, если в оптимальном решении переменные $x_{j_k} = 0, k = \overline{1, l-1}$, где $C_{\max}^l = \max_{1 \leq i \leq n} C_{j_i}$.

Доказательство. Оптимальное значение функционала задачи (1.169)-(1.172) не превышает $k C_{\max}$ или $k C_{\max}^l$ (если переменные $x_{j_k} = 0, k = \overline{1, l-1}$), что очевидно в силу ограничения (1.172). Напомним, что на допустимых решениях

$$x_{j_k} = y_{j_k}, \quad k = \overline{1, n}.$$

Докажем первое высказывание. Необходимо показать, что если ввести переменную y_{j_1} в количестве $\frac{p_* B - L_k}{p_* A_{j_1} - f_{j_1}}$ а из дополнительных переменных оставим самую высокоприоритетную, то больше никакую основную переменную нельзя будет ввести и, следовательно, по утверждениям 2 и 3 имеем верхнюю границу значения функционала задачи (1.169)-(1.172). Покажем это.

Пусть y_{j_l} - любая из основных переменных $l = \overline{2, n}$.

Тогда

$$\begin{aligned} A_{n+m+1}y_{n+m+1} + A_{j_l}y_{j_l} + A_{j_1} \frac{\rho_* B - L_k}{\rho_* A_{j_1} - f_{j_1}} &\leq B, \\ f_{n+m+1}y_{n+m+1} + f_{j_l}y_{j_l} + f_{j_1} \frac{\rho_* B - L_k}{\rho_* A_{j_1} - f_{j_1}} &\geq L_k \end{aligned} \quad (1.210)$$

Преобразовав (1.210), получим

$$\begin{aligned} A_{n+m+1}y_{n+m+1} + A_{j_l}y_{j_l} &\leq \frac{L_k - \rho_{j_1} B}{\rho_* - \rho_{j_1}}, \\ f_{n+m+1}y_{n+m+1} + f_{j_l}y_{j_l} &\geq \rho_* \frac{L_k - \rho_{j_1} B}{\rho_* - \rho_{j_1}}. \end{aligned} \quad (1.211)$$

Решив эту систему, получим

$$y_{j_l} = 0, \quad y_{n+m+1} = \frac{L_k - \rho_{j_1} B}{f_{n+m+1} - \rho_{j_1} A_{n+m+1}}. \quad (1.212)$$

Второе высказывание доказывается аналогично.

Утверждение доказано.

Из утверждения 5 вытекает следующее следствие: пусть получено допустимое решение задачи ЛЦП (1.169)–(1.172) со значением функционала Z , тогда невозможно получить допустимое решение с более высоким значением функционала только на комбинациях переменных, у которых максимальное ϕ_j меньше Z .

Приведем теперь алгоритм, который основан на доказанных выше утверждениях и предназначен для точного либо приближенного решения задачи (1.169)–(1.172).

Предполагается, что по задаче (1.169)–(1.172) получены эквивалентная задача „о ранце” вида (1.198), а также последовательность I_G и значения ϕ_{j_l} , $l = \overline{1, n}$.

Алгоритм представляет собой направленный перебор, организованный в соответствии с порядком, описанным после утверждения 3. Усечение области перебора производится после получения каждого допустимого решения в соответствии с утверждением 5.

Алгоритм предполагает участие ЛПР в решении задачи. Так, в п. 9 ЛПР оценивает качество получаемого решения и в зависимости от величины возможного улучшения показателя качества и трудоемкости этого улучшения управляет процессом счета.

Алгоритм. Начальные установки и обозначения:

а) $k^T = k$, $b_i^T = b_i$, $i = \overline{1, m}$, – текущие значения ограничений;

б) $j_t = j_1$ – индекс текущей анализируемой переменной;

в) $j_0 = j_1$ – индекс первой ненулевой переменной согласно I_G ;

г) $I_G = \{j_1, \dots, j_n\}$ – множество „свободных” переменных;

д) $j_k = n + 1$ – первый индекс множества „отсекаемых” переменных;

- е) $I_\varphi = \{0\}$ – множество зафиксированных переменных;
 ж) Z^* – значение функционала на лучшем допустимом решении.

1. Выбрать индекс переменной j_t из I_ϕ :

$$I_\varphi = I_\phi \cup j_t, \quad I_c = I_c \setminus j_t.$$

2. Определить максимальные допустимые значения $x_{j_t}^{(i)}$, исходя из следующих оценок.

Рассмотрим случаи:

1°. $a_{ij_t} > 0, b_i^T \geq 0$.

Найти

$$R = \min_{i \in I_c} (\min a_{it}, 0),$$

$$x_{j_t}^{(i)} = \left\lfloor \frac{b_i^T - Rk^T}{a_{ij_t} - R} \right\rfloor = \left\lfloor \frac{b_i^T + |R|k^T}{|R| + a_{ij_t}} \right\rfloor;$$

2°. $a_{ij_t} > 0, b_i^T < 0$.

Найти $R = \min_{i \in I_c} (\min a_{it}, 0)$

если $R = 0$, то перейти к п.6

если $R < 0$, то $x_{j_t}^{(i)} = \left\lfloor \frac{b_i^T - Rk^T}{a_{ij_t} - R} \right\rfloor = \left\lfloor \frac{|R|k^T - |b_i^T|}{|R| + a_{ij_t}} \right\rfloor,$

если $x_{j_t}^{(i)} < 0$ то перейти к п. 6;

3°. $a_{ij_t} < 0, b_i^T \geq 0 \Rightarrow x_{j_t}^{(i)} = k^T;$

4°. $a_{ij_t} < 0, b_i^T < 0,$

если $a_{ij_t}k^T \leq b_i^T$, то $x_{j_t}^{(i)} = k^T$, иначе найти $R = \{\min a_{it} / a_{it} < a_{ij_t},$

$l \in I_c\}$, если такой коэффициент найден, то $x_{j_t}^{(i)} = \left\lfloor \frac{b_i^T - Rk^T}{a_{ij_t} - R} \right\rfloor =$

$\left\lfloor \frac{|R|k^T - b_i^T}{|R| - a_{ij_t}} \right\rfloor$, иначе перейти к п. 6,

если $x_{j_t}^{(i)} \leq 0$, то перейти к п. 6;

5°. $a_{ij_t} = 0, x_{j_t}^{(i)} = k^T.$

3. $x_{j_t} = \min_{1 \leq i \leq m} x_{j_t}^{(i)};$

$$b_i^T = b_i^T - a_{ij_t} x_{j_t}, \quad i = \overline{1, m},$$

$$k^T = k^T - x_{j_t},$$

если $j_t = j_0$ & $x_{j_t} = 0$, то $j_0 = j_{t+1}$ и если $j_0 = j_k$ то решение не улучшаемо, перейти к п. 10.

4. Если $I_c \neq \{\emptyset\}$, то $j_t = j_{t+1}$, если $c_{j_t} \leq 0$, то когда $\exists b_i^T < 0, i = 1, m$, то перейти к 1, иначе $x_{j_t} = 0, l = \overline{t, n}$, и перейти к 7.

Если $I_c = \{0\}$, то, если $\exists b_i^T < 0$, $i = \overline{1, m}$, перейти к 5; если получено допустимое решение X , то перейти к 7.

$$5. b_i^T = b_i^T + x_{j_n} a_{ij_n}.$$

$$k^T = k^T + x_{j_n}.$$

6. Найти $j_r = \{\max_{j_t} |x_{j_t} \neq 0, j_t \in \{j_1, \dots, j_{t-1}\}\}$, если такого индекса нет, то больше допустимых решений нет и перейти к 10.

$$x_{j_r} = x_{j_r}^{-1},$$

$$b_i^T = b_i^T + a_{ij_r}, \quad i = \overline{1, m}.$$

$$k^T = k^T + 1;$$

если $j_r = j_0$ & $x_{j_r} = 0$, то $j_0 = j_{r+1}$ и если $j_0 = j_k$, то решение не улучшено и перейти к п. 10.

$$I_\varphi = I_\varphi / j_{r+2} \dots / j_t,$$

$$I_c = I_c \cup j_{r+2} \cup \dots \cup j_t,$$

$$j_t = j_{r+1}.$$

перейти к 2.

7. $Z = \sum_{i=1}^n C_j x_{j_t}$. Если $Z > Z^*$ то $x^* = x, Z^* = Z$, решение улучшено и перейти к 8, иначе перейти к 5.

8. Максимальное возможное улучшение функционала не превышает величины $\epsilon_{j_0} - Z^*$.

Найти $j_k = \{\min_{j_t} \epsilon_{j_t} < Z^*, j_t \in I_0\}$, т. е. необходимо „ветвить” еще $j_k - j_0$ переменных.

9. Если решение x^* удовлетворяет, то перейти к 10, если нет, - к 5.

10. Конец.

Покажем, что в результате направленного перебора, производимого в алгоритме, не исключается оптимальное решение. Для этого необходимо проверить справедливость производимых в п. 2 алгоритма оценок.

Во всех случаях решается задача с двумя ограничениями

$$\max x_{j_t}^{(i)}$$

$$\sum_{j \in I_c} x_j^{(i)} + x_{j_t}^{(i)} \leq k^T, \quad (1.213)$$

$$\sum_{j \in I_c} a_{ij} x_j^{(i)} + a_{ij_t} x_{j_t}^{(i)} \leq b_i^T,$$

$x_j \geq 0$ - целые, $j \in \{I_c \cup j_t\}$.

Если данная задача неразрешима, то необходимо изменять значение переменных из множества $\{I_\varphi \setminus j_t\}$, а именно по-

следнюю ненулевую зафиксированную переменную, что и делает-ся в 6 алгоритма.

Рассмотрим все случаи разрешимости и неразрешимости задачи (1.213).

1°. ($a_{ijt} > 0, b_j \geq 0$).

Очевидно, $x_{jt}^{(i)}$ примет максимальное значение в „смеси” с переменной, у которой самый большой по модулю отрицательный коэффициент.

Допустим такая переменная есть и ее индекс j_r .

Тогда необходимо решить следующую систему:

$$\begin{aligned} a_{ijt} x_{jt}^{(i)} + a_{ijr} x_{jr}^{(i)} &= b_i^T, \\ x_{jt}^{(i)} + x_{jr}^{(i)} &= k^T. \end{aligned}$$

Ограничения записаны как равенства, так как следует полностью использовать выделенный объем. Ее решение $x_{jt}^{(i)} = \frac{b_i^T - a_{ijr} k^T}{a_{ijt} - a_{ijr}}$.

Поскольку интерес представляет целочисленное решение, то

$$x_{jt}^{(i)} = \left\lfloor \frac{b_i^T - a_{ijr} k^T}{a_{ijt} - a_{ijr}} \right\rfloor.$$

Если в ограничении нет отрицательных коэффициентов, то

$$x_{jt}^{(i)} = \left\lfloor \frac{b_i^T}{a_{ijt}} \right\rfloor.$$

2°. Отметим, что задача (1.213) неразрешима, если среди коэффициентов $a_{ij}, j \in I_C$ нет отрицательных.

Предположим, что отрицательные коэффициенты есть, а наибольший по модулю отрицательный коэффициент $-a_{ijr}$. Очевидно, что наибольшее значение переменная $x_{jt}^{(i)}$ примет в комбинации с переменной $x_{jr}^{(i)}$.

Тогда решим систему

$$\begin{aligned} a_{ijr} x_{jr}^{(i)} + a_{ijt} x_{jt}^{(i)} &= b_i^T, \\ x_{jr}^{(i)} + x_{jt}^{(i)} &= k^T. \end{aligned}$$

Округление ее решения $x_{jt}^{(i)} = \left\lfloor \frac{b_i^T - a_{ijr} k^T}{a_{ijt} - a_{ijr}} \right\rfloor$ является решением

задачи (1.213), если $x_{jt}^{(i)} \geq 0$, а это возможно при $b_i^T \geq a_{ijr} k^T$.

3°. ($a_{ijt} < 0, b_i^T \geq 0$) и случай 4° ($a_{ijt} = 0$).

Решение задачи (1.213) очевидно: $x_{jt}^{(i)} = k^T$.

5°. ($a_{ijt} < 0, b_i^T < 0$).

Переменная $x_{jt}^{(i)}$ примет максимально возможное значение, равное k^T , при выполнении неравенства $a_{ijt} k^T \leq b_i^T$. Если послед-

нее неравенство не выполняется, то задача (1.213) разрешима, если существуют такие коэффициенты $a_{ij}, j \in I_C$, что $a_{ij}k^T \leq b_i^T$. Пусть такие коэффициенты есть. Отметим, что при этом $a_{ij} < a_{ijt}$. Тогда переменная $x_{jt}^{(i)}$ примет максимальное значение в „смеси“ с переменной, у которой наибольший по модулю отрицательный коэффициент. Пусть это коэффициент a_{ijr} , тогда

$$x_{jt}^{(i)} = \left\lfloor \frac{b_i^T - a_{ijr}k^T}{a_{ijr}} \right\rfloor$$

Опишем назначение остальных пунктов алгоритма, связанных с изменением значений переменных.

В 3 производится „фиксация“ значения переменной x_{jt} , исходя из оценок, полученных в 2, а также проверяется возможность проведения отсечения по утверждению 5.

В 4 производится проверка на получение допустимого решения. Если при данных значениях зафиксированных переменных не может быть получено допустимое решение, то в п. 6 алгоритма уменьшается на единицу ненулевая наименее приоритетная по последовательности I_C зафиксированная переменная и вновь производится попытка выполнить описанное после утверждения 3 правило загрузки основных переменных из множества I_C .

Оценки, производимые в п. 2 алгоритма, достаточно просты, однако они позволяют устранить появление некоторых недопустимых комбинаций при переборе, а также значительно уменьшают диапазон изменения каждой переменной.

При помощи этих оценок в ряде случаев возможно значительно уменьшить величины ϕ_{jt}^* , $l = \overline{1, n-1}$. Найдем, например, новое значение ϕ_{jr}^* . Пусть при $x_{jt} = 0, l = \overline{1, r-1}$ найдено значение x_{jr} по оценкам п. 2, тогда

$$\phi_{jr}^* = c_{jr}x_{jr} + c_{jr+1} \frac{p \cdot B - x_{jr}(p \cdot A_{jr} - f_{jr}) - L_k}{p \cdot A_{jr+1} - f_{jr+1}}. \quad (1.214)$$

Для найденных ϕ_{jt}^* , $l = \overline{1, n-1}$ полностью справедливо утверждение 5.

Благодаря направленному перебору на большом количестве примеров оптимальное решение либо решение, близкое к оптимальному по показателю качества, находилось одним из первых.

Поскольку эффективность усечения области перебора определяется величиной показателей качества у лучшего из найденных допустимых решений, то представляет интерес организация параллельного согласованного выполнения вышеизложенного алгоритма, т. е. одновременно выполняется ε ($\varepsilon \leq n$) алгоритмов со следующими начальными условиями. Для первого алгоритма начальные условия совпадают с начальными условиями, приведенными перед алгоритмом; для второго предполагается, что $x_{j1} = 0$; для l -го алгоритма $l = \overline{2, \varepsilon}$, $x_{ji} = 0, i = \overline{1, l-1}$.

Величина ε определяется возможностями системы по организации параллельного счета.

Согласование заключается в непересечении по анализируемым комбинациям переменных и использовании результатов соседних алгоритмов при проведении возможных отсечений, а также в остановке выполнения ряда алгоритмов. Несколько возможных отсечений будет приведено ниже.

Параллельный запуск эффективен также в том случае, когда требуется найти не оптимальное решение, а решение с достаточно высоким значением показателя качества, например при поиске решения со значением показателя качества выше заданного.

Теперь сформулируем ряд правил отсечения, которые позволяют сократить область перебора, а также могут быть использованы при организации процедур параллельного счета.

Определение. Назовем $I_{\phi} = \{j_1, j_2, \dots, j_n\}$ — последовательностью обратных приоритетов основных переменных, т. е. x_{j_1} — самая высокоприоритетная по обратным приоритетам переменная, x_{j_n} — самая низкоприоритетная.

Правило 1. Пусть I_{ϕ} — множество индексов зафиксированных переменных, C^* — наилучшее значение показателя качества задачи (1.169)–(1.172), полученное на допустимом решении, Z^* — соответствующее C^* значение показателя качества в задаче (1.198), x_{j_r} — самая высокоприоритетная по обратным приоритетам переменная из множества незафиксированных переменных, т. е. $j_r \in I_{\phi}'$ и $C_{j_r} > 0$. Тогда, если

$$\sum_{j \in I_{\phi}'} C_j x_j + C_{j_r} x_{j_r} < C^* + 1,$$

где

$$x_{j_r} = \frac{\rho_*(B - \sum_{j \in I_{\phi}'} A_j x_j) - (Z^* + 1 - \sum_{j \in I_{\phi}'} f_j x_j)}{\rho_* A_{j_r} - f_{j_r}}, \quad (1.216)$$

то при данных значениях $x_j, j \in I_{\phi}'$ невозможно получить значение функционала задачи (1.169)–(1.172) свыше C^* или для задачи (1.198) свыше Z^* .

Покажем справедливость вышеизложенного правила. По утверждению 2 максимальное приращение функционала задачи (1.169)–(1.172) при данных условиях можно получить на одной переменной x_{j_r} . Найдем такое максимальное возможное значение переменной x_{j_r} , которое можно ввести в задаче (1.198), чтобы получить значение функционала свыше Z^* , т. е. как минимум $Z^* + 1$. Для этого решим систему

$$f_{n+m+1} x_{n+m+1} + f_{j_r} x_{j_r} = Z^* + 1 - \sum_{j \in I_{\phi}'} f_j x_j,$$

$$A_{n+m+1} x_{n+m+1} + A_{j_r} x_{j_r} = B - \sum_{j \in I_{\phi}'} A_j x_j.$$

Запишем ее решение:

$$x_{j_r} = \frac{\rho_*(B - \sum_{j \in I'_\Phi} A_j x_j) - (Z^* + 1 - \sum_{j \in I'_\Phi} f_j x_j)}{\rho A_{j_r} - f_{j_r}}.$$

Если $C_{j_r} x_{j_r} < C^* - \sum_{j \in I'_\Phi} C_j x_j + 1$, то при данных значениях x_j , $j \in I'_\Phi$ невозможно получить значение функционала свыше C^* .

Правило 2. Пусть справедливы условия правила 1, тогда если

$$\forall x_{j_l} = \frac{\rho_*(B - \sum_{j \in I'_\Phi} A_j x_j) - (Z^* + 1 - \sum_{j \in I'_\Phi} f_j x_j)}{\rho_* A_{j_l} - f_{j_l}} < 1,$$

где $j_l \in \{I_G \setminus I'_\Phi\}$,

то при данных значениях x_j , $j \in I'_\Phi$ невозможно получить значение функционала задачи (1.198) свыше Z^* .

Правило 3. Пусть I'_Φ – множество индексов зафиксированных переменных, C^* – наилучшее значение показателя качества задачи (1.169)–(1.172), полученное на допустимом решении. Тогда, если

$$\sum_{j \in I'_\Phi} C_j x_j + k^T C_{\max} \leq C^*,$$

где $k^T = k - \sum_{j \in I'_\Phi} x_j$ и

$$C_{\max} = \max_{j \in \{I_G \setminus I'_\Phi\}} \max(C_j, 0),$$

то при заданных значениях x_j , $j \in I'_\Phi$ невозможно получить значение функционала задачи (1.169)–(1.172) свыше C^* .

Последние два правила отсекация очевидны.

Следует также отметить, что возможно было создать алгоритм перебора по задаче (1.198), однако выполнение перебора по задаче (1.169)–(1.172) легче, чем по задаче „о ранце” (1.198), из-за существенно меньшей величины коэффициентов.

Если к ЗЛЦП (1.169)–(1.172) добавить ограничения вида $0 \leq x_j \leq d_j$, то необходимо выполнить следующие изменения в вычислительном алгоритме.

При проведении оценок в 2 алгоритма необходимо решить задачу:

$$\begin{aligned} & \max x_{j_t}^{(i)}, \\ & \sum_{j \in I_C} a_{ij} x_j^{(i)} + a_{ij_t} x_{j_t}^{(i)} \leq b_i^T, \\ & \sum_{j \in I_C} x_j + x_{j_t} \leq k^T, \\ & 0 \leq x_j \leq d_j, \quad j \in \{I_C \cup j_t\}. \end{aligned}$$

Рассмотрим, например, алгоритм решения приведенной задачи для случая 1°: $a_{ij_t} > 0$ и $b_i^T \geq 0$.

1. Определить $I_1^{(i)} = \{j \mid a_{ij} < 0, j \in I_C\}$.

При этом в

$$I_1^{(i)} = \{j_1, j_2, \dots, j_r\} \quad a_{ij_1} \leq a_{ij_2} \leq \dots \leq a_{ij_r}.$$

2. Если $I_l = \{0\}$, то $x_{jt}^{(l)} = \lfloor \frac{b_l^T}{a_{ijl}} \rfloor$, и перейти на п. 7, иначе $l = 1$.

$$3. \quad x_{jt} = \left\lfloor \frac{b_l^T - \sum_{k=1}^{l-1} a_{ijk} d_{jk} - (k^T - \sum_{k=1}^{l-1} d_{jk}) a_{ijl}}{a_{ijl} - a_{ijl}} \right\rfloor,$$

$$x_{jt} = k^T - \sum_{k=1}^{l-1} d_{jk} - x_{jt}.$$

4. Если $x_{jt}^{(l)} > d_{jt}$, то $x_{jt} = d_{jt}$ и перейти на п. 7.

5. Если $x_{jt}^{(l)} > d_{jt}$, то $I_1^{(l)} = I_1^{(l)} \setminus j_l$ и перейти на п. 6, иначе перейти на п. 7.

6. Если $I_1^{(l)} \neq \{0\}$, то $l = l + 1$ и перейти на п. 3, иначе

$$7. \text{ Конец.} \quad x_{jt} = \left\lfloor \frac{b_l^T - \sum_{k=1}^{l-1} a_{ijk} d_{jk}}{a_{ijl}} \right\rfloor.$$

В частном случае, если выполняется неравенство

$$k^T \geq \sum_{j \in I_C} d_j + d_{jt},$$

то решение очевидно:

$$x_{jt}^{(i)} = \min \left\{ \left\lfloor \frac{b_l^T - \sum_{j \in I_C} a_{ij} d_j}{a_{ijl}} \right\rfloor, d_{jt} \right\}.$$

Алгоритм проведения отсечения по правилу 1 при добавлении ограничений на переменные, очевидно, претерпевает следующие изменения:

пусть I_Φ – множество индексов зафиксированных переменных; $I_C = \{j_1, j_2, \dots, j_r\}$ – множество индексов незафиксированных переменных, упорядоченных по обратным приоритетам; C^* – наилучшее значение показателя качества задачи (1.169)–(1.172), полученное на допустимом решении; Z^* – соответствующее C^* значение показателя качества в задаче (1.198).

$$1. \quad C_\Phi = \sum_{j \in I_\Phi} C_j x_j,$$

$$A_\Phi = \sum_{j \in I_\Phi} A_j x_j,$$

$$f_\Phi = \sum_{j \in I_\Phi} f_j x_j, \quad l = 1.$$

$$2. \quad x_{j_l} = \frac{p_*(B - A_\Phi) - (Z^* + 1 - f_\Phi)}{p_* A_{j_l} - f_{j_l}}.$$

3. Если $C_\Phi + C_{j_l} x_{j_l} < C^* + 1$, то перейти на 7.

4. Если $x_{j_l} \leq d_{j_l}$ и $C_\Phi + C_{j_l} d_{j_l} \geq C^* + 1$, то условия для проведения отсечения не выполняются, и перейти на 8.

$$5. \quad I_C' = I_C \setminus j_l,$$

если $I_C = \{0\}$, то перейти на 7.

$$6. \quad C_\Phi = C_{j_l} d_{j_l},$$

$$A_\Phi = A_\Phi + A_{j_l} d_{j_l},$$

$$f_\Phi = f_\Phi + f_{j_l} d_{j_l}, \quad l = l + 1,$$

перейти на 2.

7. При данных значениях $x_j, j \in I'_\Phi$ невозможно получить решение задачи (1.169)–(1.172) с ограничениями вида $0 \leq x_j \leq d_j$ и со значением функционала свыше C^* .

8. Конец.

Определим необходимый объем памяти для решения ЗЛЦП описанным выше алгоритмом.

Для решения ЗЛЦП необходимо хранить, кроме исходной задачи, следующие информационные векторы:

– x и x^* : размерностью n чисел (n – количество переменных исходной ЗЛЦП), определяющие текущее решение и наилучшее допустимое решение;

– b : размерностью n (n чисел, определяющих величины оценок для проведения отсечений по утверждению 5);

– I_b : размерностью n (последовательность индексов обратных приоритетов);

– b^T : размерностью m (m – число связанных ограничений исходной ЗЛЦП) – текущий вектор ресурса.

Для проведения отсечений по правилу 1 необходимо хранить также коэффициенты ограничения и функционала в эквивалентной задаче „о ранце” при основных переменных и наиболее приоритетной дополнительной переменной.

Из анализа алгоритма видно, что во время решения дополнительная память не требуется, что определяет вычислительную эффективность предложенного подхода по сравнению с классическими методами решения ЗЛЦП общего вида, а именно методом ветвей и границ и методом Гомори.

Данный алгоритм имеет экспоненциальную оценку сложности вычислений, как и методы ветвей и границ и Гомори, однако его отличительными особенностями являются эффективная направленность вычисления решений и простота проводимых отсечений.

Проиллюстрируем на примере этапы преобразования ЗЛЦП общего вида к задаче „о ранце”, изложенные в предыдущих параграфах, а также процесс решения ЗЛЦП общего вида, основанный на сведении ее к задаче „о ранце” и изложенный в п. 4.

Пример 1. Рассмотрим весь ход преобразования ЗЛЦП общего вида к задаче „о ранце”. Пусть

$$\max(3x_1 + 4x_2 + x_3 + 3x_4 + 2x_5 + x_6 + 5x_7 + 4x_8 - 2x_9 + 2x_{10})$$

при условии

$$3x_1 - x_2 + 2x_3 - 5x_5 + 2x_6 - 2x_7 + 2x_8 - x_9 \leq -4;$$

$$-2x_1 + x_3 + 4x_4 + 3x_5 - x_6 - 4x_8 + 2x_9 + 3x_{10} \leq 4;$$

$$2x_1 - 3x_2 + 3x_3 - 4x_4 - x_6 + 2x_7 - 3x_9 + 4x_{10} \leq -2;$$

$$\sum_{j=1}^{10} x_j \leq 8;$$

$$x_j \equiv 0 \pmod{1}, \quad j = 1, 2, \dots, 10.$$

Перейдем к ЗЛЦП с квадратной матрицей ограничений (ЗЛЦП (1.173)–(1.177)).

Вычислим коэффициенты при дополнительных переменных $x_{11}, x_{12}, x_{13}, x_{14}$ из оценок (1.178)

$$U = 5;$$

$$L = -2;$$

$$D_0 = 9 > 8;$$

$$M_0 = 46 > (5 \cdot 9 = 45), M = 57 > 56 = (5 - (-2)) \cdot 8.$$

Напомним, что p_i и f_i , $i = 1, 2, 3$ должны быть взаимно простыми (п. 2).

Тогда

$$p_1 = 41 > (-4 - (-5) \cdot 8) = 36;$$

$$p_2 = 37 > (-4 - (-4) \cdot 8) = 36;$$

$$p_3 = 31 > (-2 - (-4) \cdot 8) = 30;$$

$$f_1 = 83 > (-4 - (-5) \cdot (8 + 9)) = 81;$$

$$f_2 = 73 > (4 - (-4) \cdot (8 + 9)) = 72;$$

$$f_3 = 67 > (-2 - (-4) \cdot (8 + 9)) = 66.$$

и ЗЛЦП (1.173)–(1.177) примет вид

$$\max(3x_1 + 4x_2 + x_3 + 3x_4 + 2x_5 + x_6 + 5x_7 + 4x_8 - 2x_9 + 2x_{10} + 46x_{11} + 57(x_{12} + x_{13} + x_{14}))$$

при ограничениях

$$x_j \geq 0, \quad j = 1, 2, \dots, 10,$$

$$\sum_{j=1}^{10} x_j + 9x_{11} \leq 8,$$

$$3x_1 - x_2 + 2x_3 - 5x_5 + 2x_6 - 2x_7 + 2x_8 - x_9 - 83x_{11} + 41x_{12} \leq -4;$$

$$-2x_1 + x_3 + 4x_4 + 3x_5 - x_6 - 4x_8 + 2x_9 + 3x_{10} - 73x_{11} + 37x_{13} \leq 4;$$

$$2x_1 - 3x_2 + 3x_3 - 4x_4 - x_6 + 2x_7 - 3x_9 + 4x_{10} - 67x_{11} + 31x_{14} \leq -2;$$

$$x_j \equiv 0 \pmod{1}, \quad j = 1, 2, \dots, 14.$$

Для определения матрицы A_1 (1.181) необходимо решить диофантово уравнение (1.182), или уравнение

$$47027 \cdot \hat{a}_1 + 95201 \cdot \hat{a}_2 + 92783 \hat{a}_3 + 101639 \hat{a}_4 = 1,$$

коэффициенты в котором определены по (1.183).

Согласно утверждению 4 из п. 3 ищем такое решение диофантова уравнения, которое удовлетворяет условию

$$|\hat{a}_i| \leq p_{i-1} - 1, \quad i = 2, 3, 4.$$

По этому утверждению

$$\hat{a}_1 = x_1^{(4)} + t_3 \cdot 101639 x_1^{(3)} + t_2 \cdot 2993 x_1^{(2)} + t_1 \cdot 83;$$

$$\hat{a}_2 = x_2^{(4)} + t_3 \cdot 101639 x_2^{(3)} + t_2 \cdot 2993 x_2^{(2)} - t_1 \cdot 41;$$

$$\hat{a}_3 = x_3^{(4)} + t_3 \cdot 101639 x_3^{(3)} - t_2 \cdot 37;$$

$$\hat{a}_4 = x_4^{(4)} - t_3 \cdot 31;$$

$$1 = 47027 x_1^{(4)} + 95201 x_2^{(4)} + 92783 x_3^{(4)} + 101639 x_4^{(4)} ;$$

$$31 = 47027 x_1^{(3)} + 95201 x_2^{(3)} + 92783 x_3^{(3)} ;$$

$$1147 = 47027 x_1^{(2)} + 95201 x_2^{(2)} .$$

При помощи алгоритма, изложенного в п. 2, найдем частное решение диофантова уравнения

$$47027 x_1^{(4)} + 95201 x_2^{(4)} + 92783 x_3^{(4)} + 101639 x_4^{(4)} = 1 .$$

Прямая таблица

$x_1^{(4)}$	$x_2^{(4)}$	$x_3^{(4)}$	$x_4^{(4)}$						
47027	95201	92783	101639						
0	1147	45756	7585	47027					
0	0	1023	703	0	1147				
0	0	320	0	0	444	703			
0	0	0	0	0	124	63	320		
0	0	0	0	0	61	0	5	63	
0	0	0	0	0	1	0	0	3	5

Обратная таблица

$x_1^{(4)}$	$x_2^{(4)}$	$x_3^{(4)}$	$x_4^{(4)}$						
*	*	*	*	0	1	0	0	0	0
*	*	*	*	0	1	0	-12	0	0
*	*	*	*	0	1	59	-12	0	0
*	*	-131	*	0	1	59	-12	0	0
*	*	-131	189	0	1	59	-12	0	0
*	3976	-131	189	0	1	59	-12	0	0
-8199	3976	-131	189	0	1	59	-12	0	0

Получим решение

$$x_1^{(4)} = -8199, x_2^{(4)} = 3976, x_3^{(4)} = -131, x_4^{(4)} = 189.$$

Преобразуем уравнение

$$47027 x_1^{(3)} + 95201 x_2^{(3)} + 92783 x_3^{(3)} = 31$$

в уравнение

$$1517 x_1^{(3)} + 3071 x_2^{(3)} + 2993 x_3^{(3)} = 1$$

и решим его.

Прямая таблица

$x_1^{(3)}$	$x_2^{(3)}$	$x_3^{(3)}$				
1517	3071	2993				
0	37	1476	1517			
0	0	33	0	37		
0	0	0	0	4	33	
0	0	0	0	0	1	4

Обратная таблица

$x_1^{(3)}$	$x_2^{(3)}$	$x_3^{(3)}$				
*	*	*	0	0	1	0
*	*	*	0	-8	1	0
*	*	9	0	-8	1	0
*	-359	9	0	-8	1	0
709	-359	9	0	-8	1	0

Получаем решение $x_1^{(3)} = 709, x_2^{(3)} = -359, x_3^{(3)} = 9$.

Преобразуем уравнение

$$47027x_1^{(2)} + 95201x_2^{(2)} = 1147$$

в уравнение $41x_1^{(2)} + 83x_2^{(2)} = 1$, его решение

$$x_1^{(2)} = -2, x_2^{(2)} = 1.$$

Теперь решим систему

$$\hat{a}_4 = x_4^{(4)} - t_3 \cdot 31 = 189 - t_3 \cdot 31;$$

$$|\hat{a}_4| \leq 30;$$

$$\hat{a}_4 = 3, t_3 = 6.$$

Тогда

$$\hat{a}_3 = x_3^{(4)} + t_3 \cdot 101639x_3^{(3)} - t_2 \cdot 37 =$$

$$= -131 + 6 \cdot 101639 \cdot 9 - t_2 \cdot 37 =$$

$$= 5488375 - t_2 \cdot 37;$$

$$|\hat{a}_3| \leq 36,$$

откуда $\hat{a}_3 = 17, t_2 = 148334$.

Аналогично

$$\hat{a}_2 = x_2^{(4)} + t_3 \cdot 101639x_2^{(3)} + t_2 \cdot 2993x_2^{(2)} - t_1 \cdot 41 =$$

$$= 3976 + 6 \cdot 101639 \cdot (-359) + 148334 \cdot 2993 \cdot 1 - t_1 \cdot 41 =$$

$$= 225037232 - t_1 \cdot 41;$$

$$|\hat{a}_2| \leq 40;$$

$$|\hat{a}_2| = 40, t_1 = 5488712.$$

Тогда

$$\hat{a}_1 = x_1^{(4)} + t_3 \cdot 101639x_1^{(3)} + t_2 \cdot 2993x_1^{(2)} + t_1 \cdot 83 = -121.$$

Унимодулярная матрица (1.181) имеет вид

$$\begin{bmatrix} -1 & & & & & & & & & & & & & \\ & -1 & & & & & & & & & & & & \\ & & -1 & & & & & & & & & & & \\ & & & -1 & & & & & & & & & & \\ & & & & -1 & & & & & & & & & \\ & & & & & -1 & & & & & & & & \\ & & & & & & -1 & & & & & & & \\ & & & & & & & -1 & & & & & & \\ & & & & & & & & -1 & & & & & \\ & & & & & & & & & -1 & & & & \\ & & & & & & & & & & -1 & & & \\ & & & & & & & & & & & -1 & & \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -121 & 40 & 17 & 3 \\ 3 & -1 & 2 & 0 & -5 & 2 & -2 & 2 & -1 & 0 & -83 & 41 & 0 & 0 \\ -2 & 0 & 1 & 4 & 3 & -1 & 0 & -4 & 2 & 3 & -73 & 0 & 37 & 0 \\ 2 & 3 & 3 & -4 & 0 & -1 & 2 & 0 & -3 & 4 & -67 & 0 & 0 & 31 \end{bmatrix}$$

Для перехода к ЗЛЦП (1.184) необходима матрица A_1^{-1}
Процедура обращения описана в п. 2.

Так, если $A_1 = \begin{pmatrix} -E & 0 \\ \beta & \alpha \end{pmatrix}$, согласно (2.33);

то $A_1^{-1} = \begin{pmatrix} -E & 0 \\ \alpha^{-1}\beta & \alpha^{-1} \end{pmatrix} = \begin{pmatrix} -E & 0 \\ t & \alpha^{-1} \end{pmatrix}$, согласно (2.35).

Рассмотрим получение компонент A_1^{-1} :

$$\alpha^{-1} = (\alpha_{ij}^{-1}), \quad i = \overline{1, 4}, \quad j = \overline{1, 4}, \quad \alpha_{i1}^{-1}, \quad i = 1, 2, 3, 4 -$$

это коэффициенты диофантова уравнения (1.182), $\alpha_{ij}^{-1}, i = \overline{1, 4}, j = \overline{2, 4}$
можно получить по (1.66)-(1.68):

$$\alpha_{ij}^{-1} = (-1)^{11} \frac{\hat{\alpha}_j}{p_{j-1}} \prod_{\substack{k=1 \\ k \neq i}}^3 p_k, \quad j = 2, 3, 4;$$

$$\alpha_{ij}^{-1} = (-1)^{11} \hat{\alpha}_j f_{i-1} \frac{1}{p_{j-1}} \prod_{\substack{k=1 \\ k \neq i}}^3 p_k, \quad i = 2, 3, 4, \quad j = 2, 3, 4, \quad i \neq j;$$

$$\alpha_{ii}^{-1} = \frac{1}{p_{i-1}} (1 - (-1)^{10} \hat{\alpha}_i f_{i-1}) \prod_{\substack{k=1 \\ k \neq i}}^3 p_k, \quad i = 2, 3, 4$$

или проще по (1.64)-(1.65)

$$\alpha_{ij}^{-1} = -\frac{\hat{\alpha}_j}{p_{j-1}} \alpha_{i1}^{-1}, \quad i = 1, 2, 3, 4, \quad j = 2, 3, 4, \quad i \neq j;$$

$$\alpha_{ij}^{-1} = -\frac{1}{p_{j-1}} (1 - \alpha_{i1}^{-1} \hat{\alpha}_j), \quad i = 1, 2, 3, 4, \quad j = 2, 3, 4, \quad i = j.$$

Конечный вид матрицы α^{-1}

$$\begin{bmatrix} 47 & 027 & -45 & 880 & -21 & 607 & -4551 \\ 95 & 201 & -92 & 879 & -43 & 741 & -9213 \\ 92 & 783 & -90 & 520 & -42 & 630 & -8979 \\ 101 & 639 & -99 & 160 & -46 & 699 & -9836 \end{bmatrix}.$$

Значение $t = (t_{ij}), i = \overline{1, 4}, j = \overline{1, 10}$ можно получить по (1.70)-(1.71)
или по (1.69).

$$t_{ij} = \left(\sum_{k=2}^4 \alpha_{ik}^{-1} \alpha_{k-1,j} \right) + \alpha_{i1}^{-1}, \quad i = \overline{1, 4}, \quad j = \overline{1, 10}.$$

Поэлементный состав матрицы t :

$t_{1,1} = 56\ 501,$	$t_{1,2} = 106\ 560,$	$t_{1,3} = -79\ 993,$	$t_{1,4} = -21\ 197,$
$t_{1,5} = 211\ 606,$	$t_{1,6} = -18\ 575,$	$t_{1,7} = 129\ 685,$	$t_{1,8} = 41\ 695,$
$t_{1,9} = 63\ 346,$	$t_{1,10} = -35\ 998,$	$t_{2,1} = -114\ 380,$	$t_{2,2} = 215\ 719,$
$t_{2,3} = -161\ 937,$	$t_{2,4} = -42\ 911,$	$t_{2,5} = 428\ 373,$	$t_{2,6} = -37\ 603,$
$t_{2,7} = 262\ 533,$	$t_{2,8} = 84\ 407,$	$t_{2,9} = 128\ 237,$	$t_{2,10} = -72\ 874,$
$t_{3,1} = -111\ 475,$	$t_{3,2} = 210\ 240,$	$t_{3,3} = -157\ 824,$	$t_{3,4} = -41\ 821,$
$t_{3,5} = 417\ 493,$	$t_{3,6} = -36\ 648,$	$t_{3,7} = 255\ 865,$	$t_{3,8} = 82\ 263,$
$t_{3,9} = 124\ 980,$	$t_{3,10} = -71\ 023,$	$t_{4,1} = -122\ 115,$	$t_{4,2} = 230\ 307,$
$t_{4,3} = -172\ 888,$	$t_{4,4} = -45\ 813,$	$t_{4,5} = 457\ 342,$	$t_{4,6} = 40\ 146,$
$t_{4,7} = 280\ 287,$	$t_{4,8} = 90\ 115,$	$t_{4,9} = 136\ 909,$	$t_{4,10} = -77\ 802.$

Аналитические выражения коэффициентов задачи (1.184) через элементы A_j^i и коэффициенты ЗЛЦП (1.173)–(1.177) приведены в (1.74)–(1.77) и (1.79). Так,

$$t_j = p_0 t_{1,j}^{-1}, \quad j = 1, 2, \dots, 10;$$

$$\hat{t}_j = p_0 \alpha_{1,(j-n)}^{-1}, \quad j = 11, 12, 13, 14;$$

$$\hat{C}_i = -C_i + M_0 t_{1,i} + M \sum_{j=2}^4 t_{ji}, \quad i = 1, 2, \dots, 10;$$

$$\hat{C}_{10+i} = M_0 \alpha_{1,i}^{-1} + M \sum_{j=2}^4 \alpha_{ji}^{-1}, \quad i = 1, 2, 3, 4.$$

и

$\hat{t}_1 = -508\ 510,$	$\hat{t}_2 = 959\ 039,$	$\hat{t}_3 = -719\ 938,$
$\hat{t}_4 = -190\ 774,$	$\hat{t}_5 = 1\ 904\ 453,$	$\hat{t}_6 = -167\ 176,$
$\hat{t}_7 = 1\ 167\ 164,$	$\hat{t}_8 = 375\ 254,$	$\hat{t}_9 = 570\ 113,$
$\hat{t}_{10} = -323\ 983,$	$\hat{t}_{11} = -412\ 920,$	$\hat{t}_{12} = -194\ 463,$
$\hat{t}_{13} = -40\ 959,$	$\hat{t}_{14} = 423\ 243,$	
$\hat{C}_1 = -22\ 433\ 339,$	$\hat{C}_2 = 42\ 309\ 818,$	$\hat{C}_3 = -31\ 760\ 672,$
$\hat{C}_4 = -8\ 416\ 130,$	$\hat{C}_5 = 84\ 016\ 730,$	$\hat{C}_6 = -7\ 375\ 080,$
$\hat{C}_7 = 51\ 490\ 550,$	$\hat{C}_8 = 16\ 554\ 711,$	$\hat{C}_9 = 25\ 151\ 100,$
$\hat{C}_{10} = -14\ 292\ 753,$	$\hat{C}_{11} = 18\ 216\ 343,$	$\hat{C}_{12} = -8\ 578\ 912,$
$\hat{C}_{13} = -1\ 806\ 942,$	$\hat{C}_{14} = 18\ 671\ 753,$,

Далее проводим замену переменных (1.185)

$$Z_j = -\xi_j, \quad j = 1, 2, \dots, 10, \quad Z_{11} = \xi_{11};$$

$$Z_{12} = -4 - \xi_{12}, \quad Z_{13} = 4 - \xi_{13}, \quad Z_{14} = -2 - \xi_{14}.$$

После этой замены в задаче „о ранце“ (1.186):

$$t_j = -\hat{t}_j, \quad j = 1, 2, \dots, 10, \quad t_{11} = \hat{t}_{11}, \quad t_j = -\hat{t}_j, \quad j = 12, 13, 14;$$

$$\hat{C}_j = -\hat{C}_j, \quad j = 1, 2, \dots, 10, \quad \hat{C}_{11} = \hat{C}_{11}, \quad \hat{C}_j = -\hat{C}_j, \quad j = 12, 13, 14.$$

При этом

$$K_1 = 8 - (-412920) \cdot (-4) - (-194463) \cdot 4 - (-40959) \cdot (-2) = -955738;$$

$$L_1 = -2 \cdot 8 - (-18216343) \cdot (-4) - (-8578912) \cdot 4 - (-1806942) \cdot (-2) = -42163624.$$

Переобозначение переменных (1.188) величины и знака коэффициентов не изменяют, а меняют лишь их порядок.

Определим коэффициенты $\lambda_j, j = 1, 2, \dots, 13$ в матрице P (1.190) для проведения замены переменных $Z' = P W$ (1.189).

Так, из неравенства

$$\begin{aligned} 0 &\leq 508510 + \lambda_1 \cdot 423243 \leq 423243; \\ 0 &\leq -959039 + \lambda_2 \cdot 423243 \leq 423243; \\ 0 &\leq 719938 + \lambda_3 \cdot 423243 \leq 423243; \\ 0 &\leq 190774 + \lambda_4 \cdot 423243 \leq 423243 \end{aligned}$$

определим соответственно: $\lambda_1 = -1$; $\lambda_2 = 3$; $\lambda_3 = -1$; $\lambda_4 = 0$.

Аналогично определяются и остальные коэффициенты:

$$\lambda_5 = 5, \lambda_6 = 0, \lambda_7 = 3, \lambda_8 = 1, \lambda_9 = 2, \lambda_{10} = 0, \lambda_{11} = 0, \lambda_{12} = 0, \lambda_{13} = 0.$$

Теперь определим коэффициенты в ЗЛЦП (1.192):

$$\begin{aligned} S_1 &= 508510 + (-1) \cdot 423243 = 85267; \\ S_2 &= -959039 + 3 \cdot 423243 = 310690; \\ S_3 &= 719938 + (-1) \cdot 423243 = 296693. \end{aligned}$$

Далее аналогично:

$$\begin{aligned} S_4 &= 190774, & S_5 &= 211762, \\ S_6 &= 167176, & S_7 &= 102565, \\ S_8 &= 47989, & S_9 &= 276373, \\ S_{10} &= 323983, & S_{11} &= 412920, \\ S_{12} &= 194463, & S_{13} &= 40959, & S_{14} &= 423243. \end{aligned}$$

Отметим, что при замене (1.189) и выборе λ_j по (1.191), согласно утверждению 6 из п. 3, величины $d_j, j = 1, 2, \dots, 13$ в ЗЛЦП (1.192) не превышают

$$\begin{aligned} d_1 &= 22433339 + (-1) \cdot 18671753 = 3761586, \\ d_2 &= -42308918 + 3 \cdot 18671753 = 13706341, \\ d_3 &= 31760672 + (-1) \cdot 18671753 = 13088919, \\ d_4 &= 8416130 + 0 \cdot 18671753 = 8416130. \end{aligned}$$

Аналогично

$$\begin{aligned} d_5 &= 9342035, & d_6 &= 7375080, \\ d_7 &= 4524709, & d_8 &= 2117042, \\ d_9 &= 12192406, & d_{10} &= 14292753, \\ d_{11} &= 18216343, & d_{12} &= 8578912, \\ d_{13} &= 1806942, & d_{14} &= 18671753. \end{aligned}$$

Переменные $W_j, j = 1, 2, \dots, 13$ в задаче (1.192) неотрицательные. Определим границы изменения переменной W_{14} по (1.141):

$$\begin{aligned} B_2 &= -8 \cdot 5 + 0 = -40; \\ B_1 &= 8 \cdot (1 + 1) + 0 = 16, \end{aligned}$$

т. е.

$$-40 \leq W_{14} \leq 16.$$

Делаем замену переменных (1.194)

$$y_{14} = W_{14} - B_2, \quad 0 \leq y_{14} \leq 56$$

и переходим к ЗЛЦП (1.198), в которой

$$\begin{aligned} A_j &= S_j, \quad j = 1, 2, \dots, 14; \\ f_j &= d_j, \quad j = 1, 2, \dots, 14; \end{aligned}$$

$$\begin{aligned} B &= -955738 - 423243 \cdot (-40) = 15943982 \quad \text{по (1.197)}, \\ L_{\kappa} &= -42163624 - 18671753 \cdot (-40) = 704706496 \quad \text{по (1.199)}. \end{aligned}$$

Упорядочение переменных $y_j, j = \overline{1, 14}$ по прямым приоритетам

$$\{14, 11, 12, 13, 2, 9, 10, 3, 5, 4, 6, 7, 1, 8\}.$$

Упорядочение переменных $y_j, j = 1, 2, \dots, 10$ по обратным приоритетам

$$I_C = \{7, 2, 8, 1, 4, 5, 10, 3, 6, 9\}.$$

Решим задачу алгоритмом, предложенным в п. 4.

Начальные установки:

$$\begin{aligned} b_1^T &= -4, & b_2^T &= 4, & b_3^T &= -2, & k^T &= 8, \\ I_C &= \{7, 2, 8, 1, 4, 5, 10, 3, 6, 9\}, & I_\Phi &= \{0\}, \\ j_t &= 7. \end{aligned}$$

Тогда

$$1. I_\Phi = \{7\}, I_C = \{2, 8, 1, 4, 5, 10, 3, 6, 9\}.$$

2. Проведем оценки:

$$\begin{aligned} x_7^{(1)} &= k^T = 8, \text{ случай } 5^\circ, \\ x_7^{(2)} &= k^T = 8, \text{ случай } 4^\circ, \\ x_7^{(3)} &= \left\lfloor \frac{-2 - (-4) \cdot 8}{2 - (-4)} \right\rfloor = 5, \text{ случай } 2^\circ, R = a_{3,4} = -4. \end{aligned}$$

$$3. x_7 = \min\{x_7^{(1)}, x_7^{(2)}, x_7^{(3)}\} = x_7^{(3)} = 5;$$

$$b_1^T = b_1^T - a_{1,7} x_7 = 6;$$

$$b_2^T = b_2^T - a_{2,7} x_7 = 4;$$

$$b_3^T = b_3^T - a_{3,7} x_7 = -12;$$

$$k^T = k^T - x_7 = 3.$$

4. Поскольку $I_C \neq \{\emptyset\}$, то $j_t = 2$ и переходим к 1.

$$1. I_\Phi = \{7, 2\}, I_C = \{8, 1, 4, 5, 10, 3, 6, 9\}.$$

$$2. x_2^{(1)} = k^T = 3, \text{ случай } 3^\circ,$$

$$x_2^{(2)} = k^T = 3, \text{ случай } 5^\circ,$$

$$x_2^{(3)} = \left\lfloor \frac{-12 - (-4) \cdot 3}{-3 - (-4)} \right\rfloor = 0, \text{ случай } 4^\circ, R = a_{3,4} = -4.$$

3. $x_2 = 0$. Значения $b_i^T, i = 1, 2, 3$, и k^T не изменяются.

4. Поскольку $I_C \neq \{\emptyset\}$, то $j_t = 8$, и переходим к 1.

$$1. I_\Phi = \{7, 2, 8\}, I_C = \{1, 4, 5, 10, 3, 6, 9\}.$$

$$2. x_8^{(1)} = \left\lfloor \frac{6 - (-5) \cdot 3}{2 - (-5)} \right\rfloor = 3, \text{ случай } 1^\circ R = a_{1,5} = -5,$$

$$x_8^{(2)} = k^T = 3, \text{ случай } 3^\circ,$$

$$x_8^{(3)} = k^T = 3, \text{ случай } 5^\circ.$$

$$3. x_8 = 3,$$

$$b_1^T = b_1^T - a_{1,8} x_8 = 0, \quad b_2^T = 16, \quad b_3^T = -12, \quad k^T = 0.$$

4. Поскольку $I_C \neq \{\emptyset\}$, но $k^T = 0$ и $b_3^T = -12 < 0$, то получена недопустимая комбинация, и переходим к 5.

$$5. b_1^T = b_1^T + a_{1,8} x_8 = 6, b_2^T = 4, b_3^T = -12, k^T = 3.$$

$$6. j_r = 5.$$

$$x_5 = x_5 - 1 = 4,$$

$$b_1^T = b_1^T + a_{1,5} = 4, b_2^T = 4, b_3^T = -10, k^T = 4.$$

$$I_\Phi = \{7, 2\}, I_C = \{8, 1, 4, 5, 10, 3, 6, 9\}, j_t = 2$$

и переходим к 2.

$$2. x_2^{(1)} = k^T = 4, \text{ случай } 3^\circ,$$

$$x_2^{(2)} = k^T = 4, \text{ случай } 5^\circ,$$

$$x_2^{(3)} = k^T = 4, \text{ случай } 4^\circ, \text{ так как } -3 \cdot 4 \leq -10.$$

$$3. x_2 = 4, b_1^T = b_1^T - a_{1,2} x_2 = 8, b_2^T = 4, b_3^T = 2, k^T = 0.$$

4. Получено допустимое решение, так как $k^T = 0$

$$x_7 = 4, x_2 = 4, x_j = 0, j \in I_C,$$

$$C^* = C_7 x_7 + C_2 x_2 = 36.$$

Поскольку $C_2 k = 4 \cdot 8 = 32 < C^* = 36$, по утверждению 5 из п. 4, то при $x_7 = 0$ невозможно получить значение функционала свыше 32 по правилу отсечения 3 из п. 4. Следовательно, рассмотрим случай, когда $x_7 \neq 0$. Остальные комбинации $C x_7 \in \{1, 2, 3, 4\}$ отсекаются по правилу 3.

Например, при $x_7 = 4, x_2 = 3, k^T = 1, I_\Phi = \{7, 2\}$

$$\sum_{j \in I_\Phi} C_j x_j + C_8 k^T = 36 \leq 36.$$

Поэтому $x_7 = 4, x_2 = 4, x_j = 0, j = 1, 3, 4, 5, 6, 8, 9, 10$ – оптимальное решение.

Подробно процесс решения ЗЛЦП по алгоритму из п. 4 рассмотрен в примере 2.

Пример 2. Пусть необходимо решить ЗЛЦП вида

$$\max (x_1 + 2x_2 + 5x_3 + 2x_4 + 4x_5 + 3x_6);$$

$$-x_1 + 2x_2 + x_3 - 3x_4 - 4x_5 - 2x_6 \leq -2;$$

$$x_1 - 2x_2 - x_3 + 3x_4 + 4x_5 + 2x_6 \leq 2;$$

$$-2x_1 - x_2 + 3x_3 - 2x_4 + 2x_5 + 3x_6 \leq 7;$$

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 10;$$

$$x_j \geq 0, x_j \equiv 0 \pmod{1}, j = 1, 2, \dots, 6.$$

Заметим, что при помощи первых двух неравенств задано равенство.

Эквивалентная ЗЛЦП „о ранце” имеет следующие значения коэффициентов, см. (1.198):

$$A_1 = 8108,$$

$$A_2 = 4533,$$

$$A_3 = 168235,$$

$$A_4 = 128052,$$

$$A_5 = 291754,$$

$$A_6 = 47334,$$

$$A_7 = 110055,$$

$$A_8 = 170027,$$

$$A_9 = 176341,$$

$$A_{10} = 300817,$$

$$f_1 = 262868,$$

$$f_2 = 149953,$$

$$f_3 = 5454884,$$

$$f_4 = 4151977,$$

$$f_5 = 9459907,$$

$$f_6 = 1534739,$$

$$f_7 = 3668464,$$

$$f_8 = 5513018,$$

$$f_9 = 5717746,$$

$$f_{10} = 9753805.$$

и пики

$$B = 16395191,$$

$$L_1 = 531603580.$$

Построим последовательность обратных приоритетов.
Для этого определим значения $\phi_j, j = 1, 2, \dots, 6$ по (1.209):

$$\phi_1 = \frac{p_{10}B - L_k}{p_{10}A_1 - f_1} = \frac{f_{10}B - L_k A_{10}}{f_{10}A_1 - f_1 A_{10}} =$$

$$= \frac{9753805 \cdot 16395191 - 531603580 \cdot 300817}{9753805 \cdot 8108 - 262868 \cdot 300817} = 11,720;$$

$$\phi_2 = \frac{f_{10}B - L_k A_{10}}{f_{10}A_2 - f_2 A_{10}} = 25,338.$$

Аналогично $\phi_3 = 53,346$; $\phi_4 = 22,697$; $\phi_5 = 37,783$; $\phi_6 = 27,710$.

Отсортируем их по убыванию. Получим следующую последовательность индексов: {3, 5, 6, 2, 4, 1}.

Эта последовательность по утверждению 3 п. 4 является последовательностью обратных приоритетов, так как она вычислена при $Z = L_k < p_{10}B = 531603916, 994$.

Процесс решения задачи, согласно алгоритму из п. 4, представляет собой последовательность однотипных шагов. На каждом шаге выполняются следующие пункты алгоритма:

1. Выбор текущей переменной x_{j_t}
2. Определение оценок $x_{j_t}^{(i)}, i = \overline{1, m}$.
3. Фиксация значения переменной x_{j_t} и изменение значений $b_i^T, i = \overline{1, m}, k^T$.

4. Проверка возможности дальнейшего „ветвления” и проверка на получение допустимого решения.

Сокращение значения последней ненулевой зафиксированной переменной производится в п. 6 алгоритма. При переходе от 4 к 6 необходимо выполнить п. 5, если $x_{j_n} \neq 0$.

Сделаем начальные установки:

$$b_1^T = -2, b_2^T = 2, b_3^T = 7, k^T = 10,$$

$$I_C = \{3, 5, 6, 2, 4, 1\}, I_\Phi = \{0\},$$

$$j_t = 3.$$

шаг 1

1) $I_\Phi = \{3\}, I_C = \{5, 6, 2, 4, 1\}$;

2) $x_3^{(0)}$ (случай 2°), $R = a_{1,5} = -4$,

$$x_3^{(1)} = \left\lfloor \frac{b_1^T - R k^T}{a_{1,3} - R} \right\rfloor = \left\lfloor \frac{-2 - (-4) \cdot 10}{1 - (-4)} \right\rfloor = \left\lfloor \frac{38}{5} \right\rfloor = 7,$$

$$x_3^{(2)}: (3^\circ), x_3^{(2)} = k^T = 10,$$

$$x_3^{(3)}: (1^\circ), R = a_{3,4} = -2,$$

$$x_3^{(3)} = \left\lfloor \frac{b_3^T - R k^T}{a_{3,3} - R} \right\rfloor = \left\lfloor \frac{7 - (-2) \cdot 10}{3 - (-2)} \right\rfloor = \left\lfloor \frac{27}{5} \right\rfloor = 5;$$

3) $x_3 = \min\{x_3^{(1)}, x_3^{(2)}, x_3^{(3)}\} = x_3^{(3)} = 5$,

$$b_1^T = b_1^T - a_{1,3} x_3 = -2 - 1 \cdot 5 = -7,$$

$$b_2^T = b_2^T - a_{2,3} x_3 = 2 - (-1) \cdot 5 = 7,$$

$$b_3^T = b_3^T - a_{3,3} x_3 = 7 - 3 \cdot 5 = -8,$$

$$k^T = k^T - x_3 = 10 - 5 = 5.$$

4. Поскольку $I_C \neq \{0\}$, то $j_t = 5$, и переходим к следующему шагу.

Шаг 2

1) $I_\Phi = \{3, 5\}, I_C = \{6, 2, 4, 1\}$;

2) $x_5^{(1)}: (4^\circ)$, так как $a_{1,5}k^T = -4 \cdot 5 \leq -7$, то $x_5^{(1)} = -5$,

$$x_5^{(2)}: (1^\circ), R = a_{2,2} = -2, x_5^{(2)} = \left\lfloor \frac{7 - (-2) \cdot 5}{4 - (-2)} \right\rfloor = \left\lfloor \frac{17}{6} \right\rfloor = 2,$$

$$x_5^{(3)}: (2^\circ), R = a_{3,4} = -2, x_5^{(3)} = \left\lfloor \frac{-8 - (-2) \cdot 5}{2 - (-2)} \right\rfloor = \left\lfloor \frac{2}{4} \right\rfloor = 0;$$

3) $x_5 = x_5^{(3)} = 0$.

Значения $b_i^T, i = 1, 2, 3$ и k^T не изменяются.

4. Поскольку $I_C \neq \{0\}$, то $j_t = 6$, и переходим к следующему шагу.

Шаг 3

1) $I_\Phi = \{3, 5, 6\}, I_C = \{2, 4, 1\};$

2) $x_6^{(1)}: (4^\circ)$, так как $a_{1,6}k^T = -2 \cdot 5 \leq -7$, то $x_6^{(1)} = 5$.

$$x_6^{(2)}: (1^\circ), x_6^{(2)} = \left\lfloor \frac{7 - (-2) \cdot 5}{2 - (-2)} \right\rfloor = \left\lfloor \frac{17}{4} \right\rfloor = 4,$$

$$x_6^{(3)}: (2^\circ), x_6^{(3)} = \left\lfloor \frac{-8 - (-2) \cdot 5}{3 - (-2)} \right\rfloor = \left\lfloor \frac{2}{5} \right\rfloor = 0;$$

3) $x_6 = x_6^{(3)} = 0$.

Значения $b_i^T, i = 1, 2, 3$ и k^T не изменяются;

4. Поскольку $I_C \neq \{0\}$, то $j_t = 2$ и переходим к следующему шагу.

Шаг 4

1) $I_\Phi = \{3, 5, 6, 2\}, I_C = \{4, 1\};$

2) $x_2^{(1)}: (2^\circ), x_2^{(1)} = \left\lfloor \frac{-7 - (-3) \cdot 5}{2 - (-3)} \right\rfloor = \left\lfloor \frac{8}{5} \right\rfloor = 1,$

$$x_2^{(2)}: (3^\circ), x_2^{(2)} = k^T = 5,$$

$$x_2^{(3)}: (4^\circ), x_2^{(3)} = \left\lfloor \frac{-8 - (-2) \cdot 5}{-1 - (-2)} \right\rfloor = \left\lfloor \frac{2}{1} \right\rfloor = 2;$$

3) $x_2 = x_2^{(1)} = 1,$

$$b_1^T = -7 - 2 = -9,$$

$$b_2^T = 7 - (-2) = 9,$$

$$b_3^T = -8 - (-1) = -7,$$

$$k^T = 5 - 1 = 4.$$

4. Поскольку $I_C \neq \{0\}$, то $j_t = 4$, и переходим к шагу 5.

Шаг 5

1) $I_\Phi = \{3, 5, 6, 2, 4\}, I_C = \{1\};$

2) $x_4^{(1)}: (4^\circ)$, так как $-3 \cdot 4 \leq -9$, то $x_4^{(1)} = 4$,

$$x_4^{(2)}: (1^\circ), R = 0, x_4^{(2)} = \left\lfloor \frac{9}{3} \right\rfloor = 3,$$

$x_4^{(3)}: (4^\circ)$, так как $-2 \cdot 4 \leq -7$, то $x_4^{(3)} = 4$;

3) $x_4 = x_4^{(2)} = 3.$

$$b_1^T = 0, b_2^T = 0, b_3^T = -1,$$

$$k^T = 1;$$

4) Поскольку $I_C \neq \{0\}$, то $j_t = 1$, и переходим к следующему шагу.

Шаг 6

1) $I_\Phi = \{3, 5, 6, 2, 4, 1\}, I_C = \{0\};$

2) $x_1^{(1)}: (3^\circ), x_1^{(1)} = k^T = 1,$

$$x_1^{(2)}: (1^\circ), x_1^{(2)} = 0,$$

$$x_1^{(3)}: (4^\circ), x_1^{(3)} = k^T = 1;$$

3) $x_1 = x_1^{(2)} = 0;$

$$b_i, i = 1, 2, 3 \text{ и } k^T \text{ не изменяются;}$$

4. Поскольку $I_C = \{0\}$ и $b_3^T = -1 < 0$, то решение недопустимо и необходимо сократить значения переменной x_4 : $x_4 = x_4 - 1 = 2$.

При этом

$$\begin{aligned}b_1^T &= b_1^T + a_{1,4} = -3, \\b_2^T &= b_2^T + a_{2,4} = 3, \\b_3^T &= b_3^T + a_{3,4} = -1 - 2 = -3, \\k^T &= k^T + 1 = 2, \\j_t &= i.\end{aligned}$$

Эти операции проводятся в 6 алгоритма.

Шаг 7

$x_1^{(1)}: (4^\circ), x_4^{(1)}$ – недопустимо, так как $-1 \cdot 2 > -3$. Следовательно, переходим к 6 и уменьшаем $x_4: x_4 = x_4 - 1 = 1$. При этом $b_1^T = -6, b_2^T = 6, b_3^T = -5, k^T = 4, j_t = 1$.

Шаг 8

$x_1^{(1)}: (4^\circ), x_4^{(1)}$ – недопустимо, аналогично шагу 7, и, следовательно,

$$\begin{aligned}x_4 &= x_4 - 1 = 0, \\b_1^T &= -9, b_2^T = 9, b_3^T = -7, k^T = 4, j_t = 1.\end{aligned}$$

Шаг 9

$x_1^{(1)}$ – недопустимо аналогично шагу 7, и сокращаем (согласно п. 6 алгоритма) переменную $x_2: x_2 = x_2 - 1 = 0$:

$$\begin{aligned}b_1^T &= -7, b_2^T = 7, b_3^T = -8, k^T = 5, \\I_\Phi &= \{3, 5, 6, 2\}, I_C = \{4, i\}, \\j_t &= 4.\end{aligned}$$

Шаг 10

- 1) $I_\Phi = \{3, 5, 6, 2, 4\}, I_C = \{1\}$;
- 2) $x_1^{(1)}: (4^\circ), x_4^{(1)} = k^T = 5, x_2^{(2)}: (1^\circ), R_4 = 0, x_4^{(2)} = \lfloor \frac{7}{3} \rfloor = 2, x_3^{(2)}: (4^\circ), x_4^{(2)} = k^T = 5,$
- 3) $x_4 = x_4^{(2)} = 2, b_1^T = -1, b_2^T = 1, b_3^T = -4, k^T = 3.$

4. Поскольку $I_C \neq \{0\}$, то $j_t = 1$, и переходим к шагу 11.

Шаг 11

- 1) $I_\Phi = \{3, 5, 6, 2, 4, 1\}, I_C = \{0\}$,
- 2) $x_1^{(1)}: (4^\circ), x_4^{(1)} = 3,$

$$\begin{aligned}x_1^{(2)}: (1^\circ), x_4^{(2)} &= \lfloor \frac{1}{1} \rfloor = 1, \\x_3^{(2)}: (4^\circ), x_4^{(2)} &= 3;\end{aligned}$$

- 3) $x_4 = x_4^{(2)} = 1,$

$$b_1^T = 0, b_2^T = 0, b_3^T = -2, k^T = 2;$$

- 4) Поскольку $I_C = \{0\}$, но $b_3^T = -2 < 0$ и переходим к 5 алгоритма:

$$b_1^T = b_1^T + a_{1,1} = -1, b_2^T = 1, b_3^T = -4, k^T = 3,$$

а затем в 6 сокращаем величину x_4 :

$$x_4 = x_4 - 1 = 1,$$

$$b_1^T = b_1^T + a_{1,4} = -4, b_2^T = 4, b_3^T = -6, k^T = 4,$$

$$I_\Phi = \{3, 5, 6, 2, 4\}, I_C = \{1\}, j_t = 1.$$

Шаг 12

- 1) $I_{\Phi} = \{3, 5, 6, 2, 4, 1\}$, $I_C = \{0\}$;
- 2) $x_1^{(0)}: (4^0), x_1^{(0)} = k^T = 4$,
 $x_1^{(2)}: (1^0), x_1^{(2)} = \begin{bmatrix} 4 \\ 1 \end{bmatrix} = 4$,
- 3) $x_1^{(3)}: (4^0), x_1^{(3)} = k^T = 4$;
- 3) $x_1 = 4, b_1^T = b_1^T - a_{1,1}x_1 = 0, b_2^T = 0, b_3^T = 2, k^T = 0$;
- 4) Поскольку $I_C = \{0\}$ и все $b_i^T \geq 0, i = 1, 2, 3$, то получено допустимое решение:

$$x_3 = 5, x_5 = x_6 = x_2 = 0, x_4 = 1, x_1 = 4,$$

$$C^* = \sum_{j=1}^6 C_j x_j = 31.$$

Найдем также значение функционала в эквивалентной задаче „о ранце”. Для этого приведем определенные по (1.200) значения дополнительных переменных:

$$x_7 = 0, x_8 = 0, x_9 = 2, x_{10} = 50.$$

Тогда

$$Z^* = \sum_{j=1}^{10} f_j x_j = 531603611.$$

Проанализируем полученное решение с целью проведения отсечений по утверждению 5 из п. 4.

Определим уточненную оценку по (4.46) для G_5^H

$$1) \text{ если } x_3 = 0, \text{ то } x_5 = x_5^{(2)} = \left\lfloor \frac{2 - (-2) \cdot 10}{4 - (-2)} \right\rfloor = \left\lfloor \frac{22}{6} \right\rfloor = 3,$$

$$2) G_5^H = C_5 x_5 + C_6 \frac{\rho_7 B - x_5(\rho_7 A_5 - f_5) - L_k}{\rho_7 A_6 - f_6} = 4 \cdot 3 +$$

$$+ 3 \cdot 6 \cdot 287 = 30,86.$$

Таким образом, по утверждению 5 при $x_3 = 0$ не удастся получить значение функционала выше G_5^H . Следовательно, для определения оптимального решения исходной задачи необходимо проверить все комбинации переменных с переменной

$$x_3 \in \{1, 2, 3, 4, 5\}.$$

На данном этапе ЛПР может прекратить процесс счета, если удовлетворяет полученное решение, в противном случае переходит к последующему ветвлению, т. е. к 5 алгоритма:

$$b_1^T = b_1^T + a_{1,1} x_1 = 0 - 1 \cdot 4 = -4,$$

$$b_2^T = b_2^T + a_{2,1} x_1 = 0 + 1 \cdot 4 = 4,$$

$$b_3^T = b_3^T + a_{3,1} x_1 = 2 - 2 \cdot 4 = -6,$$

$$k^T = k^T + x_1 = 0 + 4 = 4.$$

Затем в 6 алгоритма сокращаем переменную x_4 :

$$x_4 = x_4 - 1 = 0,$$

$$b_1^T = -7, b_2^T = 7, b_3^T = -8, k^T = 5,$$

$$I_{\Phi} = \{3, 5, 6, 2, 4\}, I_C = \{1\}.$$

Заметим, что по правилу отсечения 3 из п. 4

$$\sum_{j \in I_{\Phi}} C_j x_j + k^T C_1 = 5 \cdot 5 + 5 \cdot 1 = 30 < 31,$$

следовательно, при данных значениях $x_j, j \in I_{\Phi}$ нельзя получить значение функционала исходной задачи свыше $C^* = 31$.

Переходим к 6. Сокращаем переменную x_3 :

$$x_3 = x_3 - 1 = 4,$$

$$b_1^T = b_1^T = a_{1,3} = -6, \quad b_2^T = 6, \quad b_3^T = -5, \quad k^T = 6;$$

$$I_{\Phi} = \{3\}, \quad I_C = \{5, 6, 2, 4, 1\}, \quad j_t = 5.$$

Через четыре шага алгоритма придем к комбинации:

$$x_3 = 4, \quad x_5 = 1, \quad x_6 = 0, \quad x_2 = 1, \quad x_4 = 0,$$

$$k^T = 4, \quad I_C = \{1\}, \quad I_{\Phi} = \{3, 5, 6, 2, 4\}.$$

По правилу отсечения 3:

$$\sum_{j \in I_{\Phi}} C_j x_j + k^T C_1 = 26 + 4 \cdot 1 = 30 < 31,$$

и, следовательно при текущих значениях $x_j, j \in I_{\Phi}$ невозможно получить значение функционала свыше 30.

Сокращаем значение переменной x_2 :

$$x_2 = x_2 - 1 = 0$$

и через шаг приходим к комбинации

$$x_3 = 4, \quad x_5 = 1, \quad x_6 = 0, \quad x_2 = 0, \quad x_4 = 0,$$

которую согласно правилу отсечения 3 также необходимо исключить.

Сокращаем значение переменной x_5 :

$$x_5 = x_5 - 1 = 0$$

и через два шага приходим к комбинации

$$x_3 = 4, \quad x_5 = 0, \quad x_6 = 1, \quad x_2 = 1.$$

Проведем отсечение по правилу 1 из п. 4 (см. 1.216).

$$x_4 = \frac{P_7(B - A_3 \cdot 4 - A_6 - A_2) - (Z^* + 1 - f_3 \cdot 4 - f_6 - f_2)}{P_7 A_4 - f_4}$$

(Z^* – определено после шага 12);

$$x_4 = \frac{32,42438 \cdot 15670384 - 508102384}{30,73248} = 3,3026.$$

Поскольку $\sum_{j \in I_{\Phi}} C_j x_j + C_4 x_4 = 25 + 6,605 = 31,605 < 32$, то сокращаем значение переменной x_2 :

$$x_2 = x_2 - 1 = 0.$$

Дальнейшее отсечение удобно представить в виде табл., в которой приведены соответственно: значения переменных в анализируемой комбинации, номер правила отсечения (1 или 3 согласно п. 4), значение функционала на зафиксированных переменных, максимальное приращение функционала и сумма последних двух значений.

Таблица 2

Значение переменных						Правила отсечения	$\sum_{j \in I_{\Phi}} C_j x_j$	$C_{j^*} x_{j^*}$	Σ
x_3	x_5	x_6	x_2	x_4	x_1				
4	0	1	0	-	-	1	23	8.34	31.34
4	0	0	2	-	-	1	24	7.24	31.24
4	0	0	1	-	-	1	22	8.98	30.98
4	0	0	0	-	-	1	20	10.72	30.72
3	1	1	-	-	-	1	22	9.28	31.28
3	1	0	-	-	-	3	19	12	31
3	0	-	-	-	-	1	15	15.95	30.95
2	2	-	-	-	-	1	18	12.74	30.74
2	1	-	-	-	-	1	14	15.63	29.63
2	0	-	-	-	-	1	10	18.52	28.52
1	3	-	-	-	-	1	17	12.41	29.41
1	2	-	-	-	-	1	13	15.30	28.30
1	1	-	-	-	-	1	9	18.20	27.20
1	0	-	-	-	-	1	5	21.08	26.08

Например, для полученной комбинации

$$x_3=4, x_5=0, x_6=1, x_2=0$$

проводится отсечение по правилу 1:

$$\sum_{j \in I_{\Phi}} C_j x_j = C_3 x_3 + C_5 x_5 + C_6 x_6 + C_2 x_2 = 23.$$

$$x_4 = \frac{p_7(B-4 \cdot A_3 - A_6) - (z^* + 1 - 4 \cdot f_3 - f_6)}{p_7 A_4 - f_4} =$$

$$= \frac{32,42438 \cdot 15674917 - 508249337}{32,42328 \cdot 128052 - 4151977} = 4,1714.$$

Приращение функционала равно

$$C_4 x_4 = 8,3429,$$

так как

$$\sum_{j \in I_{\Phi}} C_j x_j + C_4 x_4 = 31,34 < 32,$$

то сокращаем значение переменной x_6

$$x_6 = x_6 - 1 = 0$$

и через один шаг приходим к комбинации

$$x_3=4, x_5=0, x_6=0, x_2=2$$

и т. д.

После последнего проведенного отсечения необходимо сократить переменную x_3

$$x_3 = x_3 - 1 = 0.$$

Однако, как было показано после шага 12, при $x_3=0$ невозможно получить значение функционала свыше $6_5^H = 30.86$. Следовательно, полученное первое допустимое решение $x_3=5, x_5=x_6=x_2=0, x_4=1, x_1=4$ является оптимальным.

В результате проводимых отсечений удалось избежать получения 97 допустимых решений с ненулевым значением x_3 и значением функционала не превышающих $C^* = 31$.

Глава 2. СЕПАРАБЕЛЬНОЕ ЦЕЛОЧИСЛЕННОЕ ПРОГРАММИРОВАНИЕ

ОПТИМИЗАЦИЯ СЛОЖНЫХ СИСТЕМ

При формализации сложных процессов и явлений во многих прикладных областях возникают задачи, в которых число переменных и связей измеряется сотнями и даже многими тысячами, что практически затрудняет использование ЭВМ в задачах анализа и синтеза сложных систем.

Существует [7] три пути использования ЭВМ для такого рода задач.

1. Большая задача принадлежит к хорошо изученному классу, для которого разработаны и теоретически обоснованы численные методы, успешно зарекомендовавшие себя для небольших задач соответствующего типа. В виду значительной размерности задачи для вычислительного алгоритма может быть недостаточна оперативная память ЭВМ, а прямое использование внешних носителей информации связано с увеличенными затратами машинного времени.

Возникает необходимость каким-то образом разбить большую задачу на ряд задач меньших размерностей, но так, чтобы в конце концов получить решение исходной задачи.

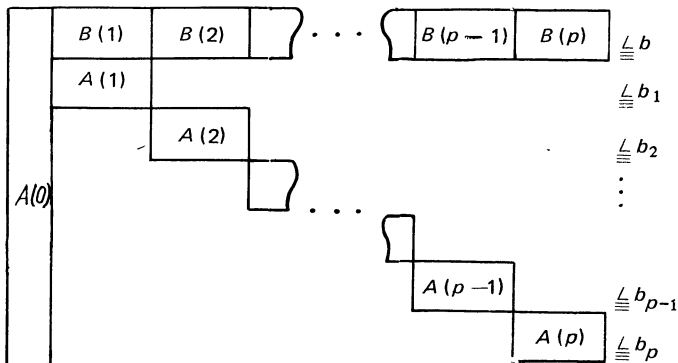
2. Прямое решение многомерной задачи невозможно (сопряжено с астрономическими затратами машинного времени). Эффективным является применение приемов искусственной декомпозиции. В этом случае искусственная схема разложения рассматривается как метод решения многомерной задачи.

3. Использование декомпозиции для большой задачи непосредственно диктуется ее спецификой. В качестве примера можно привести модели с явно выраженной блочной структурой связей. Каждая из таких моделей описывает функционирование системы, состоящей из некоторого числа подсистем, при этом выделяются локальные переменные и связи внутри подсистем, имеются общие условия для всей системы. Оптимизация таких двухуровневых систем послужила одним из источников построения вычислительных методов декомпозиции для соответствующих задач блочного программирования. Структура системы и множество взаимосвязей между подсистемами могут иметь более сложный характер, но для задач третьего типа общим является тот факт, что исходная матрица ограничений, формально описывающая модель системы, частично заполненная.

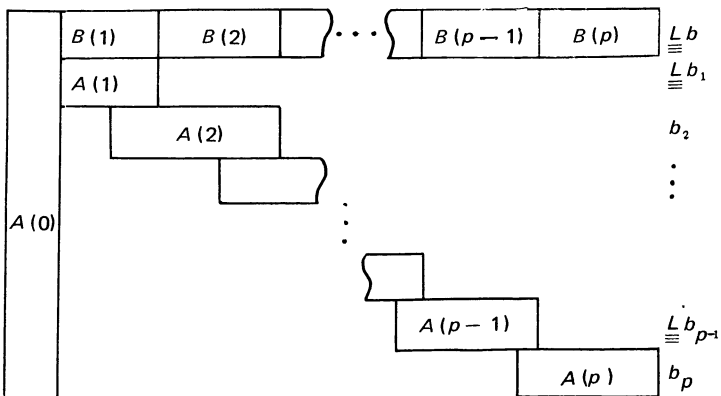
Во второй главе рассматриваются сложные системы третьего типа, оптимизация которых приводит к моделям целочисленного программирования (линейного, сепарабельного, нелинейного)

с частично заполненной матрицей ограничений, а также следующие классы задач.

Задачи сепарабельного целочисленного программирования с частично блочной структурой ограничений вида



Задачи сепарабельного целочисленного программирования с „лестничной” структурой ограничений вида



Блоки ограничений $A(i)$, $B(i)$ рассматриваются следующих видов:

$$A(i) = \| a_{ij}^i(x_{ji}) \|, \quad B(i) = \| b_{ip}^i(x_{pi}) \|, \quad (2.1)$$

$$A(i) = \| a_{ij}^i(x_{ji}) \|, \quad B(i) = \| b_{ip}^i(x_{pi}) \|, \quad (2.2)$$

$$\forall |a_{ij}^i(x_{ji})| \leq O(n^t), \quad |b_{ip}^i(x_{pi})| \leq O(n^t), \quad (2.3)$$

где $O(n^t)$ – полином фиксированной степени t от n , элементы матриц $A(i)$. $B(i)$ – целочисленные функции.

Задача нелинейного целочисленного программирования с частично блочной структурой ограничений вида

$$\max f(x_1, \dots, x_n) = \sum_{i=1}^p f_i(x_{\bar{n}_{i-1}+1}, \dots, x_{\bar{n}_{i-1}+n_i}) = \sum_{i=1}^p f_i(\bar{x}_{n_i})$$

$$\sum_{j=1}^p B_j(\bar{x}_{n_j}) \leq b, \quad A_i(x_{n_i}) \leq b_i, \quad i = \overline{1, p},$$

где $\bar{n}_i = \sum_{j=1}^i n_j$; $i = \overline{2, p}$; $\bar{n}_p = n$, $(x_{\bar{n}_{i-1}+1}, \dots, x_{\bar{n}_i})^T = \bar{x}_{n_i}$,
 $i = \overline{2, p}$, $\bar{x}_{n_1} = (x_1, \dots, x_{n_1})^T$,

$$B_j(\bar{x}_{n_j}) = \left\| \begin{array}{c} b_{j1}(\bar{x}_{n_j}) \\ \vdots \\ b_{jm}(\bar{x}_{n_j}) \end{array} \right\|, \quad A_j(\bar{x}_{n_j}) = \left\| \begin{array}{c} a_{j1}(\bar{x}_{n_j}) \\ \vdots \\ a_{jm_j}(\bar{x}_{n_j}) \end{array} \right\|$$

$b_{ij}(\bar{x}_{n_i})$, $a_{ij}(\bar{x}_{n_i})$ – нелинейные целочисленные функции.

Задача целочисленного сепарабельного программирования с частично заполненной матрицей ограничений вида

$$\begin{aligned} \max \sum_{i=1}^n f_i(x_i), \\ \sum_{j=1}^n f_{ij}(x_j) \leq b_i, \quad i = \overline{1, m} \\ \forall x_i \equiv 0 \pmod{1}, \quad \forall x_i \in \mathcal{D}_i \end{aligned}$$

Матрица $F = (f_{ij}(\bar{x}_j))$, $i = \overline{1, m}$, $j = \overline{1, n}$ является частично заполненной.

В данной главе исследуются приведенные четыре класса задач нелинейной целочисленной оптимизации, отражающие процесс оптимизации структуры и функционирования больших систем.

В каждом из приведенных классов выделяются достаточно широкие подклассы задач, для которых приводятся точные алгоритмы решения с количеством операций, ограниченным полиномом фиксированной степени от n - числа переменных задачи.

2. АЛГОРИТМ РЕШЕНИЯ ОДНОГО КЛАССА ЗАДАЧ СЕПАРАБЕЛЬНОГО ЦЕЛОЧИСЛЕННОГО ПРОГРАММИРОВАНИЯ С ЧАСТИЧНО БЛОЧНОЙ СТРУКТУРОЙ ОГРАНИЧЕНИЙ, ИМЕЮЩИЙ ПОЛИНОМИАЛЬНУЮ ОЦЕНКУ СЛОЖНОСТИ ОТНОСИТЕЛЬНО ЧИСЛА ПЕРЕМЕННЫХ ЗАДАЧИ

Алгоритм решения задачи сепарабельного целочисленного программирования для удобства изложения рассматривается последовательно для задач с одним ограничением, фиксированным числом ограничений, частично блочной структурой ограничений.

Рассмотрим следующую задачу:

$$\max \sum_{i=1}^n f_i(x_i); \quad \sum_{i=1}^n a_i x_i \leq b; \quad (2.4)$$

$$\forall x_i \in \mathcal{D}_i, \quad |x_i| \leq k, \quad k > 0, \quad \forall x_i \equiv 0 \pmod{1} \quad (2.5)$$

где $\mathcal{D}_i = \{x_i^j\}$, $j = \overline{1, N_i}$, a_i – целые числа.

Зададим функции $\Lambda_i(\varepsilon_i)$ и множества $\Lambda_i, i = \overline{1, n} : \Lambda_1(\varepsilon_1)$ принимает значение $\Lambda_1(\varepsilon_1^i), i = \overline{1, N_1}, \varepsilon_1^i = a_1 \cdot x_1^i, i = \overline{1, N_1}, \Lambda_1(\varepsilon_1^i) = f_1(x_1^i)$.

Каждому значению функции $\Lambda_1(\varepsilon_1^i)$ приписывается значение x_1^i переменной x_1 . Множество Λ_1 совпадает с множеством $\{f_1, x_1^i\}, i = \overline{1, N_1}$. Множество Λ_2 строится по функции $\Lambda_2(\varepsilon_2)$ следующим образом:

$$\Lambda_2 = \{\Lambda_2(\varepsilon_1^i + a_2 x_2^j) = \Lambda_2(\varepsilon_2^k) = f_1(x_1^i) + f_2(x_2^j); i = \overline{1, N_1}, j = \overline{1, N_2}, (x_1^i, x_2^j)\}$$

Если одному и тому же значению аргумента функции $\Lambda_2(\varepsilon_2)$ соответствуют различные цепочки значений аргументов x_1, x_2 , то оставляется та цепочка $x_1^{i_1}, x_2^{i_2}$, которой соответствует максимум функции $\Lambda_2(\varepsilon_2)$. В процессе построения множества Λ_2 формируется множество \bar{p}_2 , где \bar{p}_2 - множество различных значений аргумента ε_2 . Аргумент ε_2 принимает p_2 различных значений из множества $\bar{p}_2. \varepsilon_2 \in \{\varepsilon_2^i\}, i = \overline{1, p_2}$. Оценка чисел p_i будет дана ниже ($p_i = N_i$). Множество Λ_1 строится по функции $\Lambda_1(\varepsilon_1^i) (i \leq n)$ аналогично.

$$\Lambda_l = \{\Lambda_l(\varepsilon_{l-1}^i + a_l x_l^j) = \Lambda_l(\varepsilon_l^k) = \sum_{i=1}^{l-1} f_i(x_i^{j_i}) + f_l(x_l^j), \\ i = \overline{1, p_{l-1}}; j = \overline{1, N_l}; (x_1^{j_1}, \dots, x_{l-1}^{j_{l-1}}, x_l^j)\}.$$

В процессе построения множеств Λ_l формируются множества \bar{p}_l , которым принадлежат значения аргументов ε_l . Если одному и тому же значению ε_l функции $\Lambda_l(\varepsilon_l)$ соответствуют различные цепочки значений аргументов x_1, \dots, x_l то оставляется та цепочка $x_1^{i_1}, x_2^{i_2}, \dots, x_l^{i_l}$ которой соответствует максимум функционала $\sum_{i=1}^l f_i(x_i)$. Оптимальное решение задачи (2.4), (2.5) определяется по построенным значениям функции следующим образом.

Выделяется множество значений

$$\Omega = \{\Lambda_n(\varepsilon_n^j)\}, \varepsilon_n^j \in b.$$

Оптимальным решением является цепочка значений аргументов x_1, \dots, x_n приписанная максимальному значению функции $\Lambda_n(\varepsilon_n)$ из множества Ω . Оценим диапазон, которому принадлежат различные возможные значения ε_n . Очевидно,

$$k \sum_i a_i^- \leq \varepsilon_i \leq k \sum_i a_i^+, \quad i = \overline{1, n}$$

где a_i^- - отрицательные коэффициенты, a_i^+ - положительные коэффициенты ограничений (2.5). Таким образом, число различных значений ε_n имеет верхнюю оценку $\max_{i=\overline{1, n}} 2k |a_i| n$. Эта оценка является линейной относительно n с коэффициентом

$2k \max |a_i|$ т. е. является полиномиальной относительно n , если величины k , $\max_{i=1, \overline{n}} |a_i|$ ограничены полиномами фиксированной степени относительно n .

$$p_i \leq p_n \leq 2k \max_{i=1, \overline{n}} |a_i| n, \quad i = \overline{1, n-1}.$$

Следовательно, полиномиальной относительно n является и общая оценка числа операций изложенного алгоритма.

Рассмотрим задачу вида

$$\max \sum_{i=1}^n f_i(x_i); \quad (2.6)$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = \overline{1, m}, \quad \forall x_i \in \mathfrak{D}_i \quad (2.7)$$

$$\mathfrak{D}_i = \{x_i^j\}, \quad j = \overline{1, N_i}, \quad \forall x_i \in \mathfrak{D}_i |x_i| \leq k, \\ x_i \equiv 0 \pmod{1}, \quad i = \overline{1, n},$$

где a_{ij}, b_i — целочисленные коэффициенты, произвольных знаков.

Задаем функции $\Lambda_l(\varepsilon_1^l, \dots, \varepsilon_m^l)$, $l = \overline{1, n}$. Множество Λ_l значений функции $\Lambda_l(\varepsilon_1^l, \dots, \varepsilon_m^l)$, ($l \leq n$) строится по множеству значений аргументов функции $\Lambda_{l-1}(\varepsilon_1^{l-1}, \dots, \varepsilon_m^{l-1})$ следующим образом:

$$\Lambda_l = \{ \Lambda_l(\varepsilon_1^{l-1, i_1} + a_{1l} x_l^j, \varepsilon_2^{l-1, i_2} + a_{2l} x_l^j, \dots, \\ \dots, \varepsilon_m^{l-1, i_m} + a_{ml} x_l^j) = \Lambda_l(\varepsilon_1^{l, j_1}, \varepsilon_2^{l, j_2}, \dots, \varepsilon_m^{l, j_m}) = \\ = \sum_{i=1}^{l-1} f_i(x_i^{j_i}) + f_l(x_l^j); \quad i_1 = \overline{1, p_1^{l-1}}; \dots, i_m = \overline{1, p_m^{l-1}}, j = \overline{1, N_l}; \\ (x_1^{j_1}, x_2^{j_2}, \dots, x_{l-1}^{j_{l-1}}, x_l^j) \}.$$

В процессе построения множества Λ_l строятся множества p_j^l — множества различных значений аргументов ε_j^l (p_j^l — мощность множеств $\overline{p_j^l}$). Если одному и тому же набору значений аргументов $\varepsilon_1^l, \dots, \varepsilon_m^l$ функции $\Lambda_l(\varepsilon_1^l, \dots, \varepsilon_m^l)$ соответствуют различные цепочки значений аргументов x_1, \dots, x_l то оставляется та цепочка $x_1^{i_1}, x_2^{i_2}, \dots, x_l^{i_l}$ которой соответствует максимум функционала $\sum_{i=1}^l f_i(x_i)$. Оптимальное решение задачи (2.6)–(2.7) определяется по множеству Λ_n , т. е. по построенным значениям функции $\Lambda_l(\varepsilon_1^l, \dots, \varepsilon_m^l)$ так.

Определяется множество значений $\Omega = \{ \Lambda_n(\varepsilon_1^{j_1}, \varepsilon_2^{j_2}, \dots, \varepsilon_m^{j_m}) \}$ для которого

$$\varepsilon_1^{j_1} \leq b_1; \\ \varepsilon_2^{j_2} \leq b_2; \\ \vdots \\ \varepsilon_m^{j_m} \leq b_m.$$

Оптимальным решением задачи (2.6)–(2.7) является цепочка значений аргументов x_1, \dots, x_n приписанная максимальному значению функции $\Lambda_n(\varepsilon_1^n, \dots, \varepsilon_m^n)$ из множества \mathcal{Q} .

Оценим мощность множества Λ_n . Очевидно,

$$p_j^l \leq p_j^n, \quad l = \overline{1, n-1}, \quad j = \overline{1, m},$$

$$p_j^n \leq 2k \max_{i=\overline{1, n}} |a_{ij}| n,$$

откуда мощность множества Λ_n не превышает величин

$$\prod_{j=1}^m (2k \max_{i=\overline{1, n}} |a_{ij}| \cdot n)^m.$$

При фиксированном m мощность множества Λ_n является полиномом m степени от n с коэффициентами, определенными числами $k, \max_{i=\overline{1, n}} |a_{ij}|$, $j = \overline{1, m}$. Если значения коэффициентов a_{ij} , $i = \overline{1, n}, j = \overline{1, m}$, k ограничены полиномом фиксированной степени от n , то мощность множества Λ_n , а следовательно, и число операций предложенного алгоритма имеют полиномиальную оценку относительно числа переменных n .

Рассмотрим задачу

$$\max \sum_{i=1}^n f_i(x_i), \quad (2.8)$$

$$B(1)\bar{x}_1 + B(2)\bar{x}_2 + \dots + B(p)\bar{x}_p \leq b, \quad (2.9)$$

$$A(1)\bar{x}_1 \leq b_1, \quad (2.10)$$

$$A(2)\bar{x}_2 \leq b_2, \quad (2.11)$$

$$\dots \dots \dots A(p)\bar{x}_p \leq b_p, \quad (2.12)$$

$$\forall x_i \in \mathcal{D}_i; \quad \mathcal{D}_i = \{x_i^j\}, \quad j = \overline{1, N_i}, \quad \forall x_i \in \mathcal{D}_i, \quad |x_i| \leq k,$$

$$x_i \equiv 0 \pmod{1}, \quad \bar{x}_1 = (x_1, \dots, x_{n_1})^T, \quad \bar{x}_2 = (x_{n_1+1}, \dots, x_{n_1+n_2})^T, \dots$$

$$\dots, \quad \bar{x}_p = (x_{n_1+\dots+n_{p-1}+1}, \dots, x_n)^T.$$

Целочисленные матрицы коэффициентов $B(i), A(i)$ имеют размеры соответственно: $(m \times n_i), (m_i \times n_i)$, $i = \overline{1, p}$.

Схема вычислений, приведенная для задачи (2.6)–(2.7) применительно к задаче (2.8)–(2.12), приобретает следующий вид.

Строим функции

$$\Lambda_i(\varepsilon_1^i, \dots, \varepsilon_{m_i+m}^i), \quad i = \overline{1, p},$$

по ограничениям $\begin{pmatrix} B(i) \\ A(i) \end{pmatrix} \bar{x}_i, x_i \in \mathcal{D}_i, i = \overline{1, p}$, в полном соответствии с описанной выше процедурой построения функций $\Lambda_i(\varepsilon_1^i, \dots, \varepsilon_{m_i}^i)$ для задачи (2.6)–(2.7). Рассмотрим функцию $\Lambda_{n_1}(\varepsilon_1^{n_1}, \dots, \varepsilon_{m_1}^{n_1}, \varepsilon_{m_1+1}^{n_1}, \dots, \varepsilon_{m_1+m_1}^{n_1})$. Она принимает значения из множества Λ_{n_1} . Каждому значению функции

$$\Lambda_{n_1}(\varepsilon_1^{n_1 i_1}, \dots, \varepsilon_m^{n_1 i_m}, \varepsilon_{m+1}^{n_1 i_{m+1}}, \dots, \varepsilon_{m+m_1}^{n_1 i_{m+m_1}}),$$

где $\varepsilon_j^{n_1 i_j} \in \bar{p}_j^{n_1}$, $j = \overline{1, m+m_1}$ ($\bar{p}_j^{n_1}$ – множество различных значений аргумента $\varepsilon_j^{n_1}$), приписана оптимальная последовательность значений переменных x_1, x_2, \dots, x_{n_1} . Значение функции $\Lambda_{n_1}(\cdot)$ равно значению составляющей функционала $\sum_{i=1}^{n_1} f_i(x_i)$ на приписанной последовательности значений аргументов x_1, x_2, \dots, x_{n_1} .

Определяем множество Ω_1 – множество всех различных значений последовательности аргументов $\varepsilon_1^{n_1}, \dots, \varepsilon_m^{n_1}$, для каждого из которых существует, по крайней мере, один набор значений аргументов $\varepsilon_{m+1}^{n_1}, \dots, \varepsilon_{m+m_1}^{n_1}$, таких, что приписанная соответствующему значению $\Lambda_{n_1}(\cdot)$ последовательность значений аргументов x_1, \dots, x_{n_1} удовлетворяет ограничениям (2.10). Каждому элементу из множества Ω_1 припишем ту последовательность значений аргументов x_1, \dots, x_{n_1} , удовлетворяющую ограничениям (2.10), на которой достигается максимум выражения $\sum_{i=1}^{n_1} f_i(x_i)$

Строим функции

$$\Lambda_{n_1+i}(\varepsilon_1^{n_1+i}, \dots, \varepsilon_m^{n_1+i}, \varepsilon_{m+1}^i, \dots, \varepsilon_{m+m_2}^i), \quad i = \overline{1, n_2}$$

по ограничениям

$$\begin{pmatrix} B(2) \\ A(2) \end{pmatrix} \bar{x}_2$$

в полном соответствии с описанной выше процедурой построения функций $\Lambda_i(\varepsilon_1^i, \dots, \varepsilon_m^i)$.

Множество значений функции $\Lambda_{n_1+i}(\cdot)$ строится по начальному набору значений последовательности аргументов $\varepsilon_1^{n_1}, \dots, \varepsilon_m^{n_1}$ из множества Ω_1 .

Если обозначить через Λ_{n_1+i} множество значений функции $\Lambda_{n_1+i}(\cdot)$, то

$$\begin{aligned} \Lambda_{n_1+i} = & \{ \Lambda_{n_1+i}(\varepsilon_1^{n_1+i} + b_{11}(2) x_{n_1+i}^j, \dots, \varepsilon_m^{n_1+i} + b_{m_1}(2) x_{n_1+i}^j, \\ & a_{11}(2) x_{n_1+i}^j, \dots, a_{m_2}(2) x_{n_1+i}^j) = \Lambda_{n_1+i}(\varepsilon_1^{(n_1+i)j_1}, \dots, \\ & \dots, \varepsilon_m^{(n_1+i)j_m}, \varepsilon_{m+1}^{1jm+1}, \dots, \varepsilon_{m+m_2}^{1jm+m_2}) = \sum_{k=1}^{n_1} f_k(x_k^{jk} + f_{n_1+i}(x_{n_1+i}^j), \\ & (x_1^j, \dots, x_{n_1}^j, x_{n_1+i}^j), \forall (\varepsilon_1^{n_1 i_1}, \dots, \varepsilon_m^{n_1 i_m}) \in \Omega_1, j = \overline{1, N_{n_1+i}} \}, \end{aligned}$$

$b_{ij}(2), a_{ij}(2)$ – элементы матриц $B(2), A(2)$

Тогда

$$\Lambda_{n_1+i} = \{ \Lambda_{n_1+i}(\varepsilon_1^{n_1+i-1, i_1} + b_{11}(2) x_{n_1+i}^j, \dots$$

$$\begin{aligned} & \dots, \varepsilon_m^{n_1+i-1, i_m} + b_{m_i}(2) x_{n_1+i}^j, \varepsilon_{m+1}^{i-1, i_{m+1}} + a_{i_i}(2) x_{n_1+i}^j, \dots, \varepsilon_{m+m_2}^{i-1, i_{m+m_2}} \\ & + a_{m_2 i}(2) x_{n_1+i}^j = \Lambda_{n_1+i}(\varepsilon_1^{n_1+i, j_1}, \dots, \varepsilon_m^{n_1+i, j_m}, \varepsilon_{m+1}^{i, j_{m+1}}, \dots \\ & \dots, \varepsilon_{m+m_2}^{i, j_{m+m_2}}) = \sum_{k=1}^{n_1+i-1} f_k(x_k^{j_k}) + f_{n_1+i}(x_{n_1+i}^j), \\ & (x_1^j, \dots, x_{n_1+i-1}^j, x_{n_1+i}^j), \forall \varepsilon_1^{n_1+i-1, i_1} \in \bar{\rho}_1^{n_1+i-1}, \dots \\ & \dots, \varepsilon_m^{n_1+i-1, i_m} \in \bar{\rho}_m^{n_1+i-1}, \varepsilon_{m+1}^{i-1, i_{m+1}} \in \bar{\rho}_{m+1}^{i-1}, \dots, \varepsilon_{m+m_2}^{i-1, i_{m+m_2}} \in \\ & \in \bar{\rho}_{m+m_2}^{i-1}; j=1, N_{n_1+i}, i=2, n_2. \end{aligned}$$

По построению для каждого набора значений аргументов функции $\Lambda_{n_1+i}(\cdot)$ остается одна цепочка значений аргументов x_1, \dots, x_{n_1+i} ; $i=1, n_2$ которой соответствует максимальное значение $\Lambda_{n_1+i}(\cdot)$.

Рассмотрим функцию $\Lambda_{n_1+n_2}(\varepsilon_1^{n_1+n_2}, \dots, \varepsilon_m^{n_1+n_2}, \varepsilon_{m+1}^{n_2}, \dots, \varepsilon_{m+m_2}^{n_2})$.

Ее аргументы $\varepsilon_i^{n_1+n_2}, i=1, m$; $\varepsilon_{m+i}^{n_2}, i=1, m_2$ принимают значения из множеств $\bar{\rho}_i^{n_1+n_2}, i=1, m$; $\bar{\rho}_i^{n_2}, i=m+1, m+m_2$ полученные при поэтапном построении функции $\Lambda_{n_1+n_2}(\cdot)$.

Построим множество Ω_2 – множество всех различных значений последовательности аргументов $\varepsilon_i^{n_1+n_2}, i=1, m$, для каждого из которых существует, по крайней мере, один набор значений аргументов $\varepsilon_{m+i}^{n_2}, i=1, m_2$ таких, что приписанная соответствующему значению $\Lambda_{n_1+n_2}(\cdot)$ последовательность значений переменных $x_{n_1+1}, \dots, x_{n_1+n_2}$ удовлетворяет ограничениям (2.8). Каждому элементу из множества Ω_2 припишем ту последовательность значений аргументов $x_1, \dots, x_{n_1}, x_{n_1+1}, \dots, x_{n_1+n_2}$ удовлетворяющую ограничениям (2.11), на которой достигается максимум выражения $\sum_{i=1}^{n_1+n_2} f_i(x_i)$.

Аналогично строятся функции

$$\Lambda_{\sum_{i=1}^k n_{i+1}}^{i-1}(\varepsilon_1^{\bar{n}_{j-1}+i}, \dots, \varepsilon_m^{\bar{n}_{j-1}+i}, \varepsilon_{m+1}^i, \dots, \varepsilon_{m+m_j}^i), i=\bar{n}_j; \bar{n}_k = \sum_{i=1}^k n_i;$$

множества $\Lambda_{\sum_{i=1}^j n_{i+1}}^{j-1}$ по ограничениям $\left(\frac{B(j)}{A(j)}\right) \bar{x}_j, j=3, p$, а также соответствующие множества $\Omega_j, j=3, p-1$.

Оптимальное решение задачи (2.8)–(2.12) определяется последовательностью значений аргументов x_1, \dots, x_n приписанной одному из построенных значений функции $\Lambda_n(\cdot)$ для которого удовлетворяются ограничения (2.9)–(2.12) и которому соответствует наибольшее значение показателя качества (2.8).

Утверждение 1. Решение задачи (2.8)–(2.12), полученное описанным алгоритмом, является строго оптимальным решением этой задачи.

Доказательство. Справедливость утверждения 1 вытекает из следующего факта: если решить задачу (2.8)–(2.12) алгоритмом, приведенным для решения задачи (2.6)–(2.7), т. е. построить функции

$$\Lambda_i(\varepsilon_1^i, \dots, \varepsilon_m^i + \sum_{i=1}^p m_i), \quad i = \overline{1, n}$$

то значения показателя качества на оптимальных последовательностях, полученных обоими алгоритмами, совпадают.

Проиллюстрируем схему доказательства для задачи вида

$$\max \sum_{i=1}^n f_i(x_i); \quad (2.13)$$

$$B(1)\bar{x}_1 + B(2)\bar{x}_2 \leq b; \quad (2.14)$$

$$A(1)\bar{x}_1 \leq b_1; \quad (2.15)$$

$$A(2)\bar{x}_2 \leq b_2. \quad (2.16)$$

На общую модель (2.8)–(2.12) эта схема доказательства обобщается очевидным образом при помощи метода математической индукции. Строим функции $\Lambda_i(\varepsilon_1^i, \dots, \varepsilon_m^i, \varepsilon_{m+1}^i, \dots, \varepsilon_{m+m_1}^i, \varepsilon_{m+m_1+1}^i, \dots, \varepsilon_{m+m_1+m_2}^i)$, $i = \overline{1, n}$ в соответствии с алгоритмом, приведенным для задачи (2.6)–(2.7). Пусть $i = n_1$. Легко видеть, что значения аргументов $\varepsilon_{m+m_1+j}^i$, $i = \overline{1, n_1}$, $j = \overline{1, m_2}$ в силу ограничений задачи (2.13)–(2.16) тождественно равны нулю. Сравним функции $\Lambda_i(\cdot)$, $i = \overline{n_1+1, n}$ и $\Lambda_i^{(1)}(\cdot)$, $i = \overline{n_1+1, n}$. $\Lambda_i^{(1)}(\cdot)$ – функция, построенная с помощью алгоритма, приведенного для задачи (2.8)–(2.12). Согласно структуре ограничений (2.14)–(2.16) для $i = \overline{n_1+1, n}$ аргументы $\varepsilon_{m+1}^i, \dots, \varepsilon_{m+m_1}^i$ функции $\Lambda_i(\cdot)$ принимают одно и то же множество фиксированных значений, определенное на шаге с номером n_1 .

В соответствии с правилом построения функции $\Lambda_{n_1+1}^{(1)}(\cdot)$ по функции $\Lambda_{n_1}^{(1)}(\cdot)$ каждому элементу из множества Ω_1 приписана только одна последовательность значений аргументов x_1, \dots, x_{n_1} , на которой достигается максимум $\sum_{i=1}^{n_1} f_i(x_i)$. Функция $\Lambda_{n_1}(\cdot)$ для каждого элемента из Ω_1 содержит все последовательности значений переменных x_1, \dots, x_{n_1} удовлетворяющие ограничениям (2.15). Однако все различные последовательности значений аргументов функции $\Lambda_{n_1}(\cdot)$ при одном и том же наборе значений аргументов $\varepsilon_1^{n_1}, \dots, \varepsilon_m^{n_1}$ в силу структуры ограничений (2.15)–(2.16) порождают одни и те же последовательности значений аргументов $x_{n_1+1}, \dots, x_{n_1+n_2} = x_n$. Следовательно, значение показателя качества (2.13) на оптимальных последовательностях функций $\Lambda_n(\cdot)$ и $\Lambda_n^{(1)}(\cdot)$ совпадают.

Таким образом, приведенный алгоритм решения задачи (2.8)–(2.12) является точным. Найдем верхнюю оценку числа операций изложенного алгоритма.

Верхняя оценка числа операций, необходимых для построения множества различных последовательностей значений аргументов и самой функции $\Lambda_{np}(\cdot)$ равна

$$2kn(\max_{i,j,t} |b_{ij}(t)| / 2kn)^m [\max_{i,j,t} |a_{ij}(t)| \max_t n_t \cdot 2k]^{m \max_t m_t}, \quad (2.17)$$

где $b_{ij}(t)$ – элементы матрицы $B(t)$; $a_{ij}(t)$ – элементы матрицы $A(t)$. Под операцией понимается совокупность вычислений, необходимых для нахождения значения аргументов и самой величины функции $\Lambda_i(\cdot)$ по фиксированному значению аргументов функции $\Lambda_{i-1}(\cdot)$, $i = \overline{2, n}$, а также для построения последовательности значений переменных, приписанных найденному значению аргументов функции $\Lambda_i(\cdot)$ (Если этой последовательности значений аргументов функции $\Lambda_i(\cdot)$ была уже приписана последовательность значений переменных, то она либо сохраняется, либо заменяется новой, для которой значение $\Lambda_i(\cdot)$ имеет большую величину).

Оценка (2.17) является полиномиальной от n в том случае, когда полиномом от n являются параметры

$$k \leq O(n^{t_1}); \max_{i,j,t} |b_{ij}(t)| \leq O(n^{t_2}); \max_{i,j,t} |a_{ij}(t)| \leq O(n^{t_3}), \quad (2.18)$$

где $O(n^{t_i})$, $i = \overline{1, 3}$ – полином от n фиксированных степеней t_i , t_2, t_3 соответственно. $\forall m, m_i \leq T$, T не зависит от n . Тогда оценка (2.17) будет иметь вид

$$2O(n^{t_1})n [O(n^{t_2}) \cdot 2O(n^{t_1})]^m [O(n^{t_3}) \max_t n_t 2O(n^{t_1})]^{m \max_t m_t}. \quad (2.19)$$

Оценка (2.19) позволяет утверждать, что общее число элементарных операций приведенного алгоритма для решения задачи (2.8)–(2.12) также ограничено полиномом фиксированной степени от n .

Рассмотрим изменение структуры матрицы ограничений (2.9)–(2.12) при увеличении n -числа переменных задачи; в случае выполнения условия

$$m \leq T, \quad \forall m_i \leq T, \quad (T - \text{не зависит от } n).$$

В этом случае интерпретация модели может быть следующей. Модель (2.8)–(2.12) описывает сложную систему, состоящую из подсистем. Ограничения (2.5) – общесистемные связанные ограничения, ограничения (2.10)–(2.12) – ограничения каждой из подсистем. При увеличении n может изменяться либо размерность векторов n_i , $i = \overline{1, p}$ (это случай с фиксированным числом подсистем), либо n изменяется на величины n_i , $i = p, p+1, \dots$ что соответствует добавлению новых подсистем в рассматри-

ваемую сложную систему. Как показывает оценка (2.19), в обоих случаях оценка решения задачи (2.8)–(2.12) остается полиномиальной относительно n . Изменим ограничения, накладываемые на блоки $A(i)$, $i = \overline{1, p}$.

Пусть

$$\forall m_i \in O(n^{z_i}), \quad n_i \leq T, \quad (2.20)$$

т. е. число строк блоков ограничено полиномами фиксированной степени от n , а число столбцов каждого блока ограничено величиной T , от n не зависящей.

Внесем изменение в вышеизложенный алгоритм с целью получения и в этом случае полиномиальной относительно оценки решения задачи (2.8)–(2.12).

Формируем не более чем $(2k)^{n_i}$, $i = \overline{1, p}$ различных последовательностей значений компонент вектора переменных \bar{x}_i , $i = \overline{1, p}$ множество которых обозначим через Λ^i . Из этого множества последовательностей исключаются те, для которых i -й блок ограничений не удовлетворяется. По остальным последовательностям строится множество

$$\begin{aligned} \Lambda_{\sum_{j=1}^i n_j}^i &= \{ \Lambda_{\sum_{j=1}^i n_j}^i (\varepsilon_{j=1}^{\sum_{j=1}^i n_j, i_1}, \dots, \varepsilon_m^{\sum_{j=1}^i n_j, i_m} + \sum_{j=1}^{n_i} b_{m_j}(i) x_{ij}^{k_j}) = \\ &= \Lambda_{\sum_{j=1}^i n_j}^i (\varepsilon_{j=1}^{\sum_{j=1}^i n_j, j^1}, \dots, \varepsilon_m^{\sum_{j=1}^i n_j, j^m}) = \sum_{l=1}^{n_1 + \dots + n_{i-1}} f_l(x_l^{j^l}) + \sum_{j=1}^{n_i} f_{\sum_{j=1}^{i-1} n_j + j}(x_{ij}^{k_j}); \\ &\quad \forall (\varepsilon_{j=1}^{\sum_{j=1}^i n_j, i_1}, \dots, \varepsilon_m^{\sum_{j=1}^i n_j, i_m}) \in \bar{p}_{\sum_{j=1}^{i-1} n_j}, \end{aligned} \quad (2.21)$$

$$\forall (x_{i_1}^{k_1}, \dots, x_{i_{n_i}}^{k_{n_i}}) \in \Lambda^i (x_1^{j^1}, \dots, x_{\sum_{j=1}^{i-1} n_j}^{j^i}, x_{i_1}^{k_1}, \dots, x_{i_{n_i}}^{k_{n_i}}), \quad i = \overline{1, p},$$

где функция $\Lambda_{\sum_{j=1}^i n_j}^i(\cdot)$, $i = \overline{1, p}$ строится по первым m ограничениям (2.9), $\bar{p}_{\sum_{j=1}^{i-1} n_j}$ – множество последовательностей значений аргументов функции $\Lambda_{\sum_{j=1}^{i-1} n_j}^{i-1}(\cdot)$, $i = \overline{2, p}$.

Если одному и тому же набору значений аргументов $\varepsilon_{j=1}^{\sum_{j=1}^i n_j}, \dots, \varepsilon_m^{\sum_{j=1}^i n_j}$ соответствуют различные цепочки значений аргументов $x_1, \dots, x_{\sum_{j=1}^i n_j}$ то оставляет одна, доставляющая максимум соответствующей составляющей функционала (2.8). Оптимальное решение определяется из множества Λ_n по максимальному значению функции $\Lambda_n(\varepsilon_1^n, \dots, \varepsilon_m^n)$ при условии, что последовательность значений аргументов $\varepsilon_1^n, \dots, \varepsilon_m^n$ удовлетворяет ограничениям (2.9)–(2.12). Оптимальное решение – это приспаянная цепочка значений аргументов x_1, \dots, x_n найденному максимальному значению функции $\Lambda_n(\varepsilon_1^n, \dots, \varepsilon_m^n)$.

Найдем верхнюю оценку числа операций изложенного алгоритма. Для этого определим верхнюю оценку числа операций на i -м шаге $i = \overline{1, p}$. Число операций, необходимых для проверки удовлетворения последовательности значений компонент

вектора переменных \bar{x}_i из множества Λ^i i -му блоку ограничений:

$$O(n^{z_i})(2k)^{n_i}, \quad i = \overline{1, p}.$$

Под операцией понимается генерация последовательности, подстановка ее в каждое ограничение i -го блока, проверка выполнения соответствующего ограничения.

Величина $(\max_{i,j,t} |b_{ij}(t)| / 2kn)^m$ — это верхняя оценка числа различных последовательностей значений аргументов функции $\Lambda_i(\cdot)$, $i = \overline{1, p}$.

Следовательно, верхняя оценка числа операций построения множества $\Lambda_{\sum_{j=1}^i n_j}$ по множеству $\Lambda_{\sum_{j=1}^{i-1} n_j}$, $i = \overline{2, p}$.

$$(\max_{i,j,t} |b_{ij}(t)| / 2kn)^m (2k)^{n_i} + O(n^{z_i})(2k)^{n_i}. \quad (2.22)$$

Первое слагаемое формулы (2.22) — это верхняя оценка числа операций, каждая из которых реализует один шаг (2.21). Таким образом верхняя оценка числа операций для получения оптимального решения

$$\sum_{i=1}^p [(\max_{i,j,t} |b_{ij}(t)| / 2kn)^m (2k)^{n_i} + O(n^{z_i})(2k)^{n_i}] + (\max_{i,j} |b_{ij}(p)| / 2kn)^m \quad (2.23)$$

Второе слагаемое (2.23) — это количество числа операций, необходимое для построения оптимального решения по множеству Λ_p . При сделанных допущениях на размер матриц $A(i)$, $i = \overline{1, p}$

$$\forall m_i \leq O(n^{z_i}), \quad n_i \leq T.$$

Оценка (2.23) является полиномиальной относительно числа n -переменных задачи (2.8)–(2.12).

Далее рассмотрим случай, когда часть матриц $A(i)$ удовлетворяет условию (2.18), остальные — условиям (2.20). Алгоритм нахождения оптимального решения в этом случае является очевидным обобщением двух изложенных выше модификаций. Действительно, пусть J_1 и J_2 — множества индексов, для которых выполняются следующие условия:

$\forall i \in J_2$, матрицы $A(i)$ имеют размеры $m_i \times n_i$, $m_i \leq T$, $n_i \leq O(n^{z_i})$,

$\forall i \in J_1$, матрицы $A(i)$ имеют размеры $m_i \times n_i$, $m_i \leq O(n^{z_i})$; $n_i \leq T_1$.

$O(n^{z_i})$ полином от n степени Z_i , не зависящей от n ; T , T_1 — константы, от n не зависящие.

Пусть $j \in J_1$. Тогда для всех i , $i = \overline{1, n_j}$ строятся множества $\Lambda_{\sum_{l=1}^{i-1} n_l+i}$ по функциям $\Lambda_{\sum_{l=1}^{i-1} n_l+i}(\cdot)$ а также множество Ω_j — по первой модификации алгоритма. Если при этом $j-1 \in J_1$, то $\Lambda_{\sum_{l=1}^{j-1} n_l+1}$ строится по множеству Ω_{j-1} . Если $j-1 \in J_2$, то множество

$\Lambda_{\sum_{l=1}^{j-1} n_{l+1}}$ строится по множеству $\bar{p}_{\sum_{l=1}^{j-1} n_l}$. Пусть $j \in J_2$. Если $j - 1 \in J_2$ то строится множество $\Lambda_{\sum_{l=1}^j n_l}$ и $\bar{p}_{\sum_{l=1}^j n_l}$ по множеству $\bar{p}_{\sum_{l=1}^{j-1} n_l}$. Если $j - 1 \in J_1$, то множества $\Lambda_{\sum_{l=1}^j n_l}$ и $\bar{p}_{\sum_{l=1}^j n_l}$ строятся по множеству Ω_{j-1} .

Найдем верхнюю оценку числа операций для получения оптимального решения задачи (2.8)–(2.12).

Очевидно, для построения множества Λ_n верхней оценкой числа операций будет следующая:

$$\sum_{j \in J_1} n_j 2k (\max_{i,j,l} |b_{ij}(l)| 2kn)^m [\max_{l,j,t} |a_{ij}(l)| \max_t n_l 2k]^{m \sum_{l=1}^j n_l} + \\ + \sum_{j \in J_2} (\max_{i,j,l} |b_{ij}(l)| 2kn)^m (2k)^{n_j} + O(n^{z_j}) (2k)^{n_j} :$$

Для получения оптимального решения дополнительно необходимо

$$(\max_{i,j} |b_{ij}(\rho)| 2kn)^m$$

операций, если $\rho \in J_2$ и

$$(\max_{i,j} |b_{ij}(\rho)| 2kn)^m [\max_{l,j} |a_{ij}(\rho)| n_\rho 2k]^{m \rho},$$

если $\rho \in J_1$.

Отсюда следует, что полученная оценка числа вычислений оптимального решения задачи (2.8)–(2.12) является полиномиальной относительно n -числа переменных задачи.

Все приведенные алгоритмы требуют запоминания не более чем полинома фиксированной степени от n чисел, величина каждого из которых по модулю также ограничена полиномом фиксированной степени от n .

Пример

$$f = (x_1^2 + 3x_2 + 5x_3^2 + 9x_4 - x_5^3 + 7x_6 - x_7 + 10x_8 - 5x_9 + 4x_{10} + x_{11}^3 - \\ - x_{12} - x_{13}^2 + x_{14} + x_{15}^2) \rightarrow \max$$

$$1) x_1 - 2x_2 + x_3 - 3x_4 - 2x_5 + x_6 - x_7 + x_8 - x_9 + 3x_{10} - x_{11} - x_{12} + x_{13} - x_{14} + x_{15} \leq 4$$

$$2) 2x_1 + 3x_2 + 4x_3 + 7x_4 - 2x_5 - x_6 + 2x_7 - 2x_8 + x_9 + 2x_{10} - x_{11} + x_{12} - x_{13} + 2x_{14} - x_{15} \leq 10;$$

$$3) x_1 + 2x_2 \leq 5;$$

$$4) 2x_1 + 6x_2 \leq 1;$$

$$5) -2x_3 + x_4 + x_5 \leq 2;$$

$$6) x_3 + x_4 - 5x_5 \leq 2;$$

$$7) 6x_6 - x_7 - x_8 \leq -5;$$

$$8) 4x_6 - x_7 - 3x_8 \leq 1;$$

$$9) x_9 + 2x_{10} + x_{11} + x_{12} \leq 8;$$

$$10) x_9 + x_{10} - x_{11} - x_{12} \leq -5$$

$$11) x_{13} - 2x_{14} + x_{15} \leq 5$$

$$12) -x_{13} + x_{14} - x_{15} \leq -1$$

$$x_1 = \{0, 1, 2, 3, 4, 5\};$$

$$x_6 = \{0, 1, 2\};$$

$$x_{11} = \{0, 2\};$$

$$x_2 = \{0, 1, 2, 3\};$$

$$x_7 = \{0, 1, 2\};$$

$$x_{12} = \{0, 3\};$$

$$x_3 = \{0, 1, 2, 3\};$$

$$x_8 = \{0, 1, 2, 3\};$$

$$x_{13} = \{0, 1, 2\};$$

$$x_4 = \{0, 1, 2\};$$

$$x_9 = \{0, 2, 4\};$$

$$x_{14} = \{-1, 0, 1\};$$

$$x_5 = \{0, 1, 2, 3\};$$

$$x_{10} = \{0, 1, 3\};$$

$$x_{15} = \{1, 5, 7\};$$

$$\forall x_j \equiv 0 \pmod{1}$$

Для решения приведенного примера потребовалось построить 572 цепочки. Оптимальным решением является

$$\max f = 129.$$

$$x_1 = 0; x_2 = 0; x_3 = 3; x_4 = 1; x_5 = 2; x_6 = 0; x_7 = 2;$$

$$x_8 = 3; x_9 = 0; x_{10} = 0; x_{11} = 2; x_{12} = 3; x_{13} = 0;$$

$$x_{14} = 1; x_{15} = 7.$$

При решении примера сначала конструировались допустимые цепочки переменных по блокам неравенств 1 - (3), (4); 2 - (5), (6); 3 - (7), (8); 4 - (9), (10); 5 - (11), (12). После этого по неравенствам (1), (2) из построенных допустимых цепочек аргументов в соответствии с общей вычислительной процедурой находилось оптимальное решение.

3. ДЕКОМПОЗИЦИОННЫЕ АЛГОРИТМЫ ЗАДАЧИ БЛОЧНОГО ЛИНЕЙНОГО ЦЕЛОЧИСЛЕННОГО ПРОГРАММИРОВАНИЯ ОБЩЕГО ВИДА

Рассмотрим задачу линейного целочисленного программирования вида

$$\max \sum_{i=1}^n c_i x_i; \quad (2.24)$$

$$B(1)\bar{x}_1 + B(2)\bar{x}_2 + \dots + B(p)\bar{x}_p \leq b; \quad (2.25)$$

$$A(1)\bar{x}_1 \leq b_1; \quad (2.26)$$

$$A(2)\bar{x}_2 \leq b_2; \quad (2.27)$$

$$A(p)\bar{x}_p \cong b_p, \quad (2.28)$$

$$|x_i| \leq k, \quad x_i \equiv 0 \pmod{1}, \quad i = \overline{1, n}, \quad (2.29)$$

$$\begin{aligned} \bar{x}_1 &= (x_1, \dots, x_{n_1})^T, \quad \bar{x}_2 = (x_{n_1+1}, \dots, x_{n_1+n_2})^T, \dots \\ \dots, \bar{x}_p &= (x_{\sum_{j=1}^{p-1} n_j+1}, \dots, x_n)^T. \end{aligned}$$

Целочисленные матрицы коэффициентов $B(l)$, $A(l)$ имеют размеры соответственно: $(m \times n_l)$, $(m_l \times n_l)$, $l = \overline{1, p}$. Было показано, что задача ЛЦП (2.24)–(2.28) при ограничениях, накладываемых на числа m , m_l , n_l в п. 1 гл. 2, может быть решена алгоритмом с полиномиальной оценкой сложности числа операций относительно n - числа переменных задачи ЛЦП (2.24)–(2.28).

В этом параграфе опишем возможность декомпозиции задач блочного линейного целочисленного программирования на последовательность, ограниченную полиномом фиксированной степени от n задач линейного целочисленного программирования меньшей размерности, чем исходная.

Рассмотрим задачу (2.24)–(2.28). Ее алгоритм решения состоит из следующих этапов.

1. Строим функцию $\Lambda_l(\varepsilon_1^l, \dots, \varepsilon_m^l)$, $l = \overline{1, p}$. Множество Λ_l значений функции $\Lambda_l(\varepsilon_1^l, \dots, \varepsilon_m^l)$, $l \leq p$, по множеству Λ_{l-1} значений функции $\Lambda_{l-1}(\varepsilon_1^{l-1}, \dots, \varepsilon_m^{l-1})$ строится следующим образом (см. п. 1 гл. 2):

$$\begin{aligned} \Lambda_l &= \{ \Lambda_l(\varepsilon_1^{l-1, i_1} + b_{1l}(1)x_1^j, \varepsilon_2^{l-1, i_2} + b_{2l}(1)x_2^j, \dots, \varepsilon_m^{l-1, i_m} + \\ &+ b_{ml}(1)x_m^j) = \Lambda_l(\varepsilon_1^{l, j_1}, \varepsilon_2^{l, j_2}, \dots, \varepsilon_m^{l, j_m}) = \sum_{i=1}^{l-1} c_i x_i + c_l x_l^j, \quad (2.30) \\ \forall \varepsilon_1^{l-1, i_1} \in \bar{p}_1^{l-1}, \dots, \varepsilon_m^{l-1, i_m} \in \bar{p}_m^{l-1}, \quad l = \overline{1, p}; \\ \forall x_i^j \in |x_i^j| \leq k, \quad (x_1^j, x_2^j, \dots, x_{l-1}^j, x_l^j). \end{aligned}$$

При этом, если одному и тому же набору значений аргументов $\varepsilon_1^l, \dots, \varepsilon_m^l$ функции $\Lambda_l(\varepsilon_1^l, \dots, \varepsilon_m^l)$ соответствуют различные цепочки значений аргументов x_1, \dots, x_l то остается одна цепочка $x_1^{j_1}, x_2^{j_2}, \dots, x_l^{j_l}$, которой соответствует максимум функционала $\sum_{i=1}^l c_i x_i$.

Рассмотрим множество Λ_{n_1}

$$\begin{aligned} \Lambda_{n_1} &= \{ \Lambda_{n_1}(\varepsilon_1^{n_1, j_1}, \dots, \varepsilon_m^{n_1, j_m}) (x_1^{j_1}, \dots, x_{n_1}^{j_{n_1}}), \\ \forall (\varepsilon_1^{n_1, j_1}, \dots, \varepsilon_m^{n_1, j_m}) \in \bar{p}^{n_1}. \end{aligned}$$

Для каждого набора значений аргументов $\varepsilon_1^{n_1}, \dots, \varepsilon_m^{n_1}$ функции $\Lambda_{n_1}(\varepsilon_1^{n_1}, \dots, \varepsilon_m^{n_1})$ формируем следующие задачи линейного целочисленного программирования

$$\max \sum_{i=1}^{n_1} c_i x_i; \quad (2.31)$$

$$B(1) \bar{x}_1 = \bar{\varepsilon}_{n_1}^j; \quad (2.32)$$

$$A(2) \bar{x}_1 \leq b_1; \quad (2.33)$$

$$|x_j| \leq k, \quad j = \overline{1, n_1}, \quad (2.34)$$

для всех $\bar{\varepsilon}_{n_1}^j \in \bar{\rho}^{n_1}$.

В качестве допустимых последовательностей значений аргументов функции $\Lambda_{n_1}(\varepsilon_1^{n_1}, \dots, \varepsilon_m^{n_1})$ остаются только те, для которых задачи (2.31)–(2.34) разрешимы.

Таким образом, $\bar{\rho}^{n_1}$ – это множество последовательностей аргументов $\varepsilon_1, \dots, \varepsilon_m$ для которых задача (2.31)–(2.34) разрешима. Соответствующим последовательностям значений аргументов функции $\Lambda_{n_1}(\varepsilon_1^{n_1, i_1}, \dots, \varepsilon_m^{n_1, i_m})$ приписываем последовательность значений переменных x_1, \dots, x_{n_1} , которая является решением ЗЛЦП (2.31)–(2.32) при $\bar{\varepsilon}_{n_1}^j = (\varepsilon_1^{n_1, i_1}, \dots, \varepsilon_m^{n_1, i_m})^T$, а само значение функции становится равным значению показателя качества ЗЛЦП (2.31)–(2.32) на ее оптимальном решении.

2. Формируем множества

$$\begin{aligned} \Lambda_l = \{ & \Lambda_l(\varepsilon_1^{l-1, i_1} + b_{1(l-n_1)}(2)x_l^j, \varepsilon_2^{l-1, i_2} + b_{2(l-n_1)}(2)x_l^j, \dots \\ & \dots, \varepsilon_m^{l-1, i_m} + b_{m(l-n_1)}(2)x_l^j) = \Lambda_l(\varepsilon_1^{l, j_1}, \varepsilon_2^{l, j_2}, \dots, \varepsilon_m^{l, j_m}) = \\ & = \sum_{i=1}^{l-1} c_i x_i^{j_i} + c_l x_l^j; \quad \forall (\varepsilon_1^{l-1}, \dots, \varepsilon_m^{l-1}) \in \bar{\rho}^{l-1}; \quad \forall -k \leq x_l^j \leq k, \\ & (x_1^{j_1}, x_2^{j_2}, \dots, x_{l-1}^{j_{l-1}}, x_l^j) \}, \quad l = \overline{n_1+1, n_1+n_2}. \end{aligned}$$

Если одному и тому же набору значений аргументов $\varepsilon_1^l, \dots, \varepsilon_m^l$ функции $\Lambda_l(\varepsilon_1^l, \dots, \varepsilon_m^l)$ соответствуют различные цепочки значений аргументов x_1, \dots, x_l то оставляем одну, для которой значение $\sum_{i=1}^l c_i x_i$ является максимальным.

$\bar{\rho}^l, l = \overline{n_1+1, n_1+n_2}$ – это множество значений последовательностей аргументов $\varepsilon_1^l, \dots, \varepsilon_m^l$ полученное в результате последовательного построения множеств $\Lambda_l, l = \overline{n_1+1, n_1+n_2}$.

Рассмотрим множество $\Lambda_{n_1+n_2}$. Для каждого $\bar{\varepsilon}_{n_1+n_2}^{j_2} \in \bar{\rho}^{n_1+n_2}$ и всех $\bar{\varepsilon}_{n_1}^{j_1} \in \bar{\rho}^{n_1}$ формируем задачу ЛЦП вида

$$\max \sum_{i=n_1+1}^{n_1+n_2} c_i x_i = C_{\max}(\bar{\varepsilon}_{n_1}^{j_1}), \quad (2.35)$$

$$B(2) \bar{x}_2 = \bar{\varepsilon}_{n_1+n_2}^{j_2} - \bar{\varepsilon}_{n_1}^{j_1}; \quad (2.36)$$

$$A(2) \bar{x}_2 \leq b_2, \quad \forall \bar{\varepsilon}_{n_1}^{j_1} \in \bar{\rho}^{n_1}, \quad (2.37)$$

$$\forall |x_j| \leq k, \quad j = \overline{n_1+1, n_2}.$$

В качестве допустимых последовательностей аргументов функции $\Lambda_{n_1+n_2}(\varepsilon_1^{n_1+n_2}, \dots, \varepsilon_m^{n_1+n_2})$ остаются только те из множеств $\bar{\rho}^{n_1+n_2}$ для которых хотя бы одна задача (2.35)–(2.37) является разрешимой хотя бы для одного $\bar{\varepsilon}_i^{j_1} \in \bar{\rho}^{n_1}$. Последовательности $\bar{\varepsilon}_i^{j_1} \in \bar{\rho}^{n_1+n_2}$, для которых задачи (2.35)–(2.37) не разрешимы, из множества $\bar{\rho}^{n_1+n_2}$ исключаются. Каждому значению вновь конструируемой функции $\Lambda_{n_1+n_2}(\bar{\varepsilon}_i^{j_1})$ приписывается одна цепочка значений переменных $x_1, \dots, x_{n_1+n_2}$ вида $(x_1^j, \dots, x_{n_1}^{j_{n_1}}, x_{n_1+1}, \dots, x_{n_1+n_2})$ на которой достигается максимум

$$\max_{\bar{\varepsilon}_i^{j_1}} \{ \Lambda_{n_1}(\bar{\varepsilon}_i^{j_1}) + C_{\max}(\varepsilon_{n_1}^{j_1}) \} = \Lambda_{n_1+n_2}(\bar{\varepsilon}_i^{j_1}),$$

где $x_1^j, \dots, x_{n_1}^{j_{n_1}}$ – последовательность, приписанная $\Lambda_{n_1}(\bar{\varepsilon}_i^{j_1})$ и на $\bar{\varepsilon}_i^{j_1}$ достигается максимум предыдущего выражения, $x_{n_1+1}, \dots, x_{n_1+n_2}$ – решение задачи (3.12)–(3.14) для $\bar{\varepsilon}_i^{j_1}$ и $\bar{\varepsilon}_i^{j_1}$.

3. Рассмотрим этап с номером k , $k = 3, \dots, p$. Формируем множества

$$\begin{aligned} \Lambda_l &= \{ \Lambda_l(\varepsilon_1^{l-1, l_1} + b_1(l - \sum_{i=1}^{k-1} n_i)(k) x_i^j \varepsilon_2^{l-1, l_2} + b_2(l - \sum_{i=1}^{k-1} n_i)(k) x_i^j, \dots \\ &\dots, \varepsilon_m^{l-1, l_m} + b_m(l - \sum_{i=1}^{k-1} n_i)(k) x_i^j = \Lambda_l(\varepsilon_1^{l, j_1}, \dots, \varepsilon_m^{l, j_m}) = \\ &= \sum_{i=1}^{l-1} c_i x_i^{j_i} + c_l x_l^j; \quad \forall (\varepsilon_1^{l-1}, \dots, \varepsilon_m^{l-1}) \in \bar{\rho}^{l-1}, \forall x_i^j \in |x_i^j| \leq k, \\ & (x_1^j, x_2^j, \dots, x_{l-1}^{j_{l-1}}, x_l^j) \}, \quad l = \overline{n_1 + \dots + n_{k-1}, n_1 + \dots + n_k}. \end{aligned}$$

Если одному и тому же набору значений аргументов $\varepsilon_1^l, \dots, \varepsilon_m^l$ функции $\Lambda_l(\varepsilon_1^l, \dots, \varepsilon_m^l)$ соответствуют по построению различные цепочки значений аргументов x_1^j, \dots, x_l^j то оставляется одна, максимизирующая функционал $\sum_{i=1}^l c_i x_i^j$. Описанный алгоритм построения множеств Λ_l для $l \in \{ \sum_{i=1}^k n_i; k = \overline{3, p} \}$ дополняется следующими вычислительными процедурами.

Пусть $l = \sum_{i=1}^k n_i$. Рассмотрим множество Λ_l . Для каждого $\bar{\varepsilon}_i^{j_2} \in \bar{\rho}^l$ и всех $\bar{\varepsilon}_i^{j_1} \in \bar{\rho}^{l-n_k}$ формируем задачу ЛЦП вида

$$\max \sum_{i=n_1+\dots+n_{k-1}+1}^{n_1+\dots+n_k} c_i x_i = C_{\max}(\bar{\varepsilon}_i^{j_1}); \quad (2.38)$$

$$B(k) \bar{x}_k = \bar{\varepsilon}_i^{j_2} - \bar{\varepsilon}_i^{j_1}; \quad (2.39)$$

$$A(k) \bar{x}_k \leq b_k, \quad (2.40)$$

$$\forall \bar{\varepsilon}_i^{j_1} \in \bar{\rho}^{l-n_k}, \quad \forall |x_j| \leq k, \quad j = \overline{\sum_{i=1}^{k-1} n_i + 1, \sum_{i=1}^k n_i}.$$

В качестве допустимых последовательностей значений аргументов функции $\Lambda_l(\varepsilon_1^l, \dots, \varepsilon_m^l)$ в множестве $\bar{\rho}^l$ остаются только те, для которых задача (2.38)–(2.40) разрешима хотя бы для одного

$\bar{\varepsilon}_{l-n_k}^{j_1} \in \bar{\rho}^{l-n_k}$. Каждому значению вновь конструируемой функции $\Lambda_l(\bar{\varepsilon}_l^{j_1})$ приписывается одна цепочка значений переменных x_1, \dots, x_l вида $(x_1^{j_1}, \dots, x_{l-n_k}^{j_1-n_k}, \check{x}_{l-n_k+1}, \dots, \check{x}_l)$ на которой достигается максимум выражения

$$\max_{j_1} \{ \Lambda_{l-n_k}(\bar{\varepsilon}_{l-n_k}^{j_1}) + C_{\max}(\bar{\varepsilon}_{l-n_k}^{j_1}) \} = \Lambda_l(\bar{\varepsilon}_l^{j_1}),$$

где $x_1^{j_1}, \dots, x_{l-n_k}^{j_1-n_k}$ — последовательность, приписанная $\Lambda_{l-n_k}(\bar{\varepsilon}_{l-n_k}^{j_1})$ и на $\bar{\varepsilon}_{l-n_k}^{j_1}$ достигается максимум предыдущего выражения, $\check{x}_{l-n_k+1}, \dots, \check{x}_l$ решение задачи (2.38)–(2.40) для $\bar{\varepsilon}_{l-n_k}^{j_1}$.

4. По построенному множеству Λ_n определяется решение задачи ЛЦП (2.24)–(2.28). Этим решением является приписанная последовательность значений переменных x_1, \dots, x_n соответствующая максимальному значению функции $\Lambda_n(\cdot)$ на множестве значений ее аргументов $\bar{\rho}_n$ при выполнении дополнительного условия: вектор значений первых ее m аргументов $\varepsilon_1, \dots, \varepsilon_m$ удовлетворяет неравенству (2.25).

Утверждение. Изложенный декомпозиционный алгоритм решения ЗЛЦП (2.24)–(2.28) является точным.

Доказательство. Сравним декомпозиционный алгоритм с алгоритмом, приведенным в п. 2. Из построения множеств Ω_j , $j = \bar{1}, \bar{p}$ и $\bar{\rho}^l$, $l = \sum_{i=1}^k n_i$, $k = \bar{1}, \bar{p}$ вытекает, что $\Omega_j = \bar{\rho}^l$ ($k = j$), а значения функции $\Lambda_n(\cdot)$ в обоих алгоритмах на одинаковых последовательностях значений ее аргументов из множеств Ω_j и $\bar{\rho}^l$ совпадают.

Найдем оценку числа ЗЛЦП решаемых в декомпозиционном алгоритме. Верхняя оценка числа множества $\bar{\rho}^l$ равна

$$|\rho^l| \leq \{ 2k \max_{j,t} |b_{ij}(t)| \sum_{i=1}^l n_i \}^m, \quad l = \bar{1}, \bar{p}.$$

Следовательно, общее число задач ЛЦП имеют следующую верхнюю оценку.

$$\sum_{l=2}^p (2k)^{2m} (\max_{j,t} |b_{ij}(t)|)^m 2^m \left(\sum_{i=1}^l n_i \sum_{i=1}^{l-1} n_i \right)^m + (2k \max_{j,t} |b_{ij}(t)|)^m n_l^m \quad (2.41)$$

так как $n = \sum_{k=1}^{\bar{p}} n_i$, то оценка (2.41) является полиномиальной

относительно n — числа переменных задачи ЛЦП (2.24)–(2.28) при выполнении условий п. 2: $\forall m, m_i \leq T$, T не зависит от n , величины коэффициентов матриц $B(i)$, $A(i)$, $i = \bar{1}, \bar{p}$ по модулю ограничены полиномами фиксированной степени от n . Верхняя оценка размерности каждой из решаемых ЗЛЦП имеет вид: число строк ограничений $m + \max m_i$, число переменных $\max n_i$, а также n_i ограничений вида $|x_i| \leq k$. Из изложенного

выше следует, если удовлетворяются условия п. 2 гл. 2., оценка решения ЗЛЦП вида (2.31)–(2.34), (2.35)–(2.37), (2.38)–(2.40) является полиномиальной относительно n .

Оценка числа операций приведенного декомпозиционного алгоритма при выполнении условия п. 2 гл. 2 является также полиномиальной относительно n – числа переменных задачи ЛЦП (2.24)–(2.28). Для малых m ($1 \div 4$) предложенный декомпозиционный алгоритм может оказаться более эффективным, чем алгоритм п. 2 гл. 2. Очевидным образом декомпозиционный алгоритм распространяется на случай сепарабельного функционала

$$\max \sum_{i=1}^n f_i(x_i)$$

и сепарабельных ограничений (2.1)–(2.2) гл. 2. В этом случае решаем последовательность, имеющую полиномиальную относительно n оценку, задач сепарабельного целочисленного программирования, каждая из которых существенно меньшей размерности, чем исходная и полиномиальную относительно n оценку числа операций для ее решения. Объем необходимой памяти ограничен также полиномом относительно n .

Рассмотрим возможность использования метода сведения задач ЛЦП с фиксированным числом связанных ограничений к задаче ЛЦП с одним связным ограничением. Наиболее эффективным такое сведение является для задач блочного линейного целочисленного программирования (2.24)–(2.28) со следующими дополнительными условиями:

$$\forall x_i \geq 0, \quad i=1, n;$$

неравенства $|x_i| \leq k$ отсутствуют.

В состав матриц ограничений $A(i)$ входят неравенства

$$\sum_{i=1}^{n_i} x_{j_i-1} n_{j+1} \leq k_i > 0, \quad i=\overline{1, p}. \quad (2.42)$$

Тогда ЗЛЦП (2.38)–(2.40) имеет вид

$$\begin{aligned} \max C(k)^T \bar{x}^k \\ B(k) \bar{x}^k = \bar{E}_l^{j_2} - \bar{E}_{l-n_k}^{j_1}; \end{aligned} \quad (2.43)$$

$$\begin{aligned} A(l) \bar{x}_k \leq b_k; \\ \sum_{i=1}^{n_k} x_{k_i} \leq k_k, \quad \bar{x}_k \geq \bar{0}, \end{aligned}$$

где $C(k) = (C_{n_1+\dots+n_{k-1}+1}, \dots, C_{n_1+\dots+n_k})^T$, $l = \sum_{j=1}^k n_j$.

ЗЛЦП (2.42)–(2.43) сводим к ЗЛЦП вида

$$\max \sum_{i=1}^{n_k} C_i(k) x_{k_i}; \quad (2.44)$$

4. АЛГОРИТМЫ РЕШЕНИЯ ОДНОГО КЛАССА ЗАДАЧ ЦЕЛОЧИСЛЕННОГО СЕПАРАБЕЛЬНОГО ПРОГРАММИРОВАНИЯ С "ЛЕСТНИЧНОЙ" СТРУКТУРОЙ МАТРИЦЫ ОГРАНИЧЕНИЙ И ПОЛИНОМИАЛЬНОЙ ОТНОСИТЕЛЬНО ЧИСЛА ПЕРЕМЕННЫХ СЛОЖНОСТЬЮ

В данном параграфе алгоритм решения задачи целочисленного сепарабельного программирования, изложенный в п. 2, будет обобщен на случай матриц ограничений, не сводящихся к частично блочно-диагональной структуре.

Рассмотрим следующую задачу целочисленного сепарабельного программирования:

$$\sum_{i=1}^n f_i(x_i), \quad (2.46)$$

$$(B(1) B(2) \dots B(p)) x \leq b; \quad (2.47)$$

$$(A(1) 0 \dots 0) x \leq b_1; \quad (2.48)$$

$$(0 A(2) \dots 0) x \leq b_2; \quad (2.49)$$

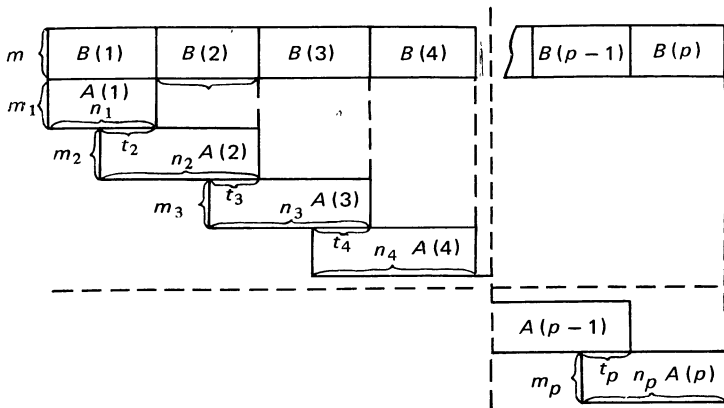
$$\dots$$

$$(0 0 \dots A(p)) x \leq b_p; \quad (2.50)$$

$$\forall x_i \in \mathfrak{D}_i; \mathfrak{D}_i = \{x_i^j\}, j = \overline{1, N_i}, |x_i^j| \leq k,$$

$$\forall x_i \equiv 0 \pmod{1}, i = \overline{1, n}.$$

Коэффициенты матрицы $B(i)$, $A(i)$ целочисленные. Структура ограничений (2.47)–(2.50) следующая:



Размеры матриц $B(i)$ соответственно

$$B(1) - (m \times n_1), B(i) - [m \times (n_i - t_i)], i = \overline{2, p}.$$

Числа $t_i, i = \overline{2, p}$ удовлетворяют условию

$$n_i - t_{i+1} > t_i, \quad i = \overline{2, p-1}.$$

Первоначально рассмотрим случай, когда $\forall m, m_i \leq T$, T не зависит от n , величина коэффициентов матриц $B(i)$, $A(i)$, $i = \overline{1, p}$ ограничена по модулю полиномами фиксированной степени от n . Легко видеть, что задача (2.46)–(2.50) не может быть сведена к задаче (2.4)–(2.8), т. е. к задаче с частично блочной диагональной структурой.

Изложим модификацию алгоритма, описанного в п. 2, приводящего к полиномиальным оценкам решения задачи (2.46)–(2.51).

Строим функции $\Lambda_i(\cdot)$, $i = \overline{1, p}$ по следующему правилу.

$$\Lambda_l(\varepsilon_1^l, \dots, \varepsilon_{m+m_l}^l), \quad l = \overline{1, n_1 - t_2} \text{ строится по ограничениям}$$

$$\begin{pmatrix} B(1) & \dots & B(p) \\ A(1) & 0 & \dots & 0 \end{pmatrix} \bar{x}, \quad x_i \in \mathcal{D}_i, \quad i = \overline{1, n_1 - t_2}$$

в соответствии с правилами построения функции $\Lambda_l(\cdot)$ изложенными в п. 2. Множество Λ_l значений функции $\Lambda_l(\varepsilon_1^l, \dots, \varepsilon_{m+m_l}^l)$, $l = \overline{1, n_1 - t_2}$ по множеству значений аргументов функции $\Lambda_{l-1}(\varepsilon_1^{l-1}, \dots, \varepsilon_{m+m_l}^{l-1})$ строится следующим образом:

$$\begin{aligned} \Lambda_l &= \{ \Lambda_l(\varepsilon_1^{l-1, i_1} + b_{1l}(1)x_1^j, \varepsilon_2^{l-1, i_2} + b_{2l}(1)x_2^j, \dots, \varepsilon_m^{l-1, i_m} + b_{ml}(1)x_m^j, \\ &\quad \varepsilon_{m+1}^{l-1, i_{m+1}} + a_{1l}(1)x_1^j, \dots, \varepsilon_{m+m_l}^{l-1, i_{m+m_l}} + a_{m_l l}(1)x_{m_l}^j) = \\ &= \Lambda_l(\varepsilon_1^{l, j_1}, \dots, \varepsilon_{m+m_l}^{l, j_{m+m_l}}) = \sum_{i=1}^{l-1} f_i(x_i^{j_i}) + f_l(x_l^j), \quad (2.51) \\ &\forall \varepsilon_1^{l-1, i_1} \in \bar{\rho}_1^{l-1}, \dots, \forall \varepsilon_{m+m_l}^{l-1, i_{m+m_l}} \in \bar{\rho}_{m+m_l}^{l-1}, \quad j = \overline{1, N_l}, \\ &\quad (x_1^j, x_2^j, \dots, x_{l-1}^j, x_l^j) \}. \end{aligned}$$

$\bar{\rho}_l^l$ — множество значений аргумента ε_i^l функции $\Lambda_l(\cdot)$. Если одному и тому же набору значений аргументов $\varepsilon_1^l, \dots, \varepsilon_{m+m_l}^l$ функции $\Lambda_l(\varepsilon_1^l, \dots, \varepsilon_{m+m_l}^l)$ соответствуют различные цепочки значений аргументов x_1, \dots, x_l , то оставляется та цепочка x_1^j, \dots, x_l^j , которой соответствует максимум функции $\Lambda_l(\cdot)$ на этом наборе значений ее аргументов.

Множество $\Lambda_{n_1 - t_2 + 1}$ строится таким образом:

$$\begin{aligned} \Lambda_{n_1 - t_2 + 1} &= \{ \Lambda_{n_1 - t_2 + 1}(\varepsilon_1^{n_1 - t_2, i_1} + b_{1(n_1 - t_2 + 1)}(1)x_{n_1 - t_2 + 1}^j, \dots, \\ &\quad \varepsilon_{m+m_1}^{n_1 - t_2, i_{m+m_1}} + a_{m_1(n_1 - t_2 + 1)}(1)x_{n_1 - t_2 + 1}^j, a_{11}(2)x_{n_1 - t_2 + 1}^j, \dots, \\ &\quad \dots, a_{m_2}(2)x_{n_1 - t_2 + 1}^j) = \Lambda_{n_1 - t_2 + 1}(\varepsilon_1^{n_1 - t_2 + 1, j_1}, \dots, \varepsilon_{m+m_1}^{n_1 - t_2 + 1, j_{m+m_1}}, \end{aligned}$$

$$\begin{aligned} \varepsilon_{m+m_1+1}^{1, j, m+m_1+1}, \dots, \varepsilon_{m+m_1+m_2}^{1, j, m+m_1+m_2} &= \sum_{i=1}^{n_1-t_2} f_i(x_i^{j_i}) + f_{n_1-t_2+1}(x_{n_1-t_2+1}^j), \\ \forall \varepsilon_1^{n_1-t_2, i_1} \in \bar{\rho}_1^{n_1-t_2}, \dots, \varepsilon_{m+m_1}^{n_1-t_2, i_{m+m_1}} \in \bar{\rho}_{m+m_1}^{n_1-t_2}; \\ j &= \overline{1, N_{n_1-t_2+1}}, (x_1^j, \dots, x_{n_1-t_2}^j, x_{n_1-t_2+1}^j) \}. \end{aligned} \quad (2.52)$$

Если одной и той же последовательности значений аргументов

$$\varepsilon_1^{n_1-t_2+1}, \dots, \varepsilon_{m+m_1}^{n_1-t_2+1}, \varepsilon_{m+m_1+1}^1, \dots, \varepsilon_{m+m_1+m_2}^1$$

функции $\Lambda_{n_1-t_2+1}(\cdot)$ соответствуют различные цепочки значений аргументов $x_1^{i_1}, \dots$

$\dots, x_{n_1-t_2+1}^{i_{n_1-t_2+1}}$, которой соответствует максимум функции $\Lambda_{n_1-t_2+1}(\cdot)$ на этом наборе значений ее аргументов.

Множества $\Lambda_{n_1-t_2+i}$, $i = \overline{1, t_2}$ строятся по множествам $\Lambda_{n_1-t_2+i-1}$ аналогичным образом

$$\begin{aligned} \Lambda_{n_1-t_2+i} &= \{ \Lambda_{n_1-t_2+i-1}(\varepsilon_1^{n_1-t_2+i-1, i_1} + b_{1(n_1-t_2+i)}(1) x_{n_1-t_2+i}^j, \dots \\ &\dots, \varepsilon_m^{n_1-t_2+i-1, i_m} + b_{m(n_1-t_2+i)}(1) x_{n_1-t_2+i}^j, \varepsilon_{m+1}^{n_1-t_2+i-1, i_{m+1}} + a_{1(n_1-t_2+i)}(1) x_{n_1-t_2+i}^j \\ &\times \varepsilon_{m+m_1}^{n_1-t_2+i-1, i_{m+m_1}} + a_{m_1(n_1-t_2+i)}(1) x_{n_1-t_2+i}^j, \varepsilon_{m+m_1+1}^{i-1, i_{m+m_1+1}} + \\ &+ a_{i_1}(2) x_{n_1-t_2+i}^j, \dots, \varepsilon_{m+m_1+m_2}^{i-1, i_{m+m_1+m_2}} + a_{m_2 i}(2) x_{n_1-t_2+i}^j) = \\ &= \Lambda_{n_1-t_2+i}(\varepsilon_1^{n_1-t_2+i, j_1}, \dots, \varepsilon_{m+m_1}^{n_1-t_2+i, j_{m+m_1}}, \varepsilon_{m+m_1+1}^{i, j_{m+m_1+1}}, \dots, \varepsilon_{m+m_1+m_2}^{i, j_{m+m_1+m_2}}) = \\ &= \sum_{i=1}^{n_1-t_2+i-1} f_i(x_i^{j_i}) + f_{n_1-t_2+i}(x_{n_1-t_2+i}^j); \forall \varepsilon_1^{n_1-t_2+i-1, i_1} \in \bar{\rho}_1^{n_1-t_2+i-1}, \dots \\ &\dots, \forall \varepsilon_{m+m_1}^{n_1-t_2+i-1, i_{m+m_1}} \in \bar{\rho}_{m+m_1}^{n_1-t_2+i-1}, \forall \varepsilon_{m+m_1+1}^{i-1, i_{m+m_1+1}} \in \bar{\rho}_{m+m_1+1}^{i-1}, \dots \\ &\dots, \forall \varepsilon_{m+m_1+m_2}^{i-1, i_{m+m_1+m_2}} \in \bar{\rho}_{m+m_1+m_2}^{i-1}, j = \overline{1, N_{n_1-t_2+i}}, \\ &(x_1^j, \dots, x_{n_1-t_2+i-1}^j, x_{n_1-t_2+i}^j) \}. \end{aligned} \quad (2.53)$$

Если одной и той же последовательности значений аргументов

$$\varepsilon_1^{n_1-t_2+i}, \dots, \varepsilon_{m+m_1}^{n_1-t_2+i}, \varepsilon_{m+m_1+1}^i, \dots, \varepsilon_{m+m_1+m_2}^i$$

функции $\Lambda_{m_1-t_2+i}(\cdot)$ соответствуют различные цепочки значений аргументов $x_1^{i_1}, \dots, x_{n_1-t_2+i}^{i_{n_1-t_2+i}}$, то оставляется та цепочка $x_1^{i_1}, \dots$

$\dots, x_{n_1-t_2+i}^{i_{n_1-t_2+i}}$, которой соответствует максимум функции $\Lambda_{m_1-t_2+i}(\cdot)$ на этом наборе значений ее аргументов. Определим множество ρ_i как декартово произведение множеств

$$\bar{\rho}_1^{n_1}, \dots, \bar{\rho}_{m+m_1}^{n_1}, \bar{\rho}_{m+m_1+1}^{t_2}, \dots, \bar{\rho}_{m+m_1+m_2}^{t_2}.$$

Следовательно, множество ρ_1 – это множество всех различных последовательностей значений аргументов $\varepsilon_1^{n_1}, \dots, \varepsilon_{m+m_1}^{n_1}, \varepsilon_{m+m_1+1}^{t_2}, \dots, \varepsilon_{m+m_1+m_2}^{t_2}$ функции $\Lambda_{n_1}(\cdot)$ полученных в результате построения множества Λ_{n_1} . По построению множества Λ_{n_1} каждому элементу из множества ρ_1 приписана одна последовательность значений переменных x_1, \dots, x_{n_1} . Зададим множество Ω_1 всех значений последовательностей аргументов $\varepsilon_1^{n_1}, \dots, \varepsilon_m^{n_1}, \varepsilon_{m+m_1+1}^{t_2}, \dots, \varepsilon_{m+m_1+m_2}^{t_2}$ удовлетворяющих следующему условию.

Для каждого элемента из Ω_1 существует, по крайней мере, один элемент из ρ_1 включающий в себя этот элемент из Ω_1 , которому приписана последовательность значений аргументов x_1, \dots, x_{n_1} удовлетворяющая ограничениям (2.48). Каждому элементу из Ω_1 припишем только одну из возможных последовательностей значений аргументов x_1, \dots, x_{n_1} на которой достигается максимум функционала $\sum_{i=1}^{n_1} f_i(x_i)$ (возможных последовательностей значений аргументов x_1, \dots, x_{n_1} столько, сколько различных элементов из ρ_1 включает в себя данный элемент из Ω_1).

Построим множество Λ_{n_1+1} .

$$\begin{aligned} \Lambda_{n_1+1} &= \{ \Lambda_{n_1+1}(\varepsilon_1^{n_1+1}, i_1, b_{1,1}(2) x_{n_1+1}^j, \dots, \varepsilon_m^{n_1+1}, i_m, b_{m,1}(2) x_{n_1+1}^j, \\ &\varepsilon_{m+m_1+1}^{t_2+1}, i_{m+m_1+1}, a_{1(t_2+1)}(2) x_{n_1+1}^j, \dots, \varepsilon_{m+m_1+m_2}^{t_2+1}, i_{m+m_1+m_2}, a_{m_2(t_2+1)}(2) x_{n_1+1}^j) = \\ &= \Lambda_{n_1+1}(\varepsilon_1^{n_1+1, j_1}, \dots, \varepsilon_m^{n_1+1, j_m}, \varepsilon_{m+m_1+1}^{t_2+1, j_{m+m_1+1}}, \dots, \varepsilon_{m+m_1+m_2}^{t_2+1, j_{m+m_1+m_2}}) = \\ &= \sum_{i=1}^{n_1+1} f_i(x_i^{j_i}) + f_{n_1+1}(x_{n_1+1}^j), \forall \varepsilon_1^{n_1+1}, \dots, \varepsilon_m^{n_1+1}, \varepsilon_{m+m_1+1}^{t_2+1}, \dots, \\ &\dots, \varepsilon_{m+m_1+m_2}^{t_2+1} \in \Omega_1; \quad j = \overline{1, N_{n_1+1}}, \\ &(x_1^{j_1}, \dots, x_{n_1+1}^j, x_{n_1+1}^j) \}. \end{aligned} \quad (2.54)$$

Если одной и той же последовательности значений аргументов $\varepsilon_1^{n_1+1}, \dots, \varepsilon_m^{n_1+1}, \varepsilon_{m+m_1+1}^{t_2+1}, \dots, \varepsilon_{m+m_1+m_2}^{t_2+1}$ функции $\Lambda_{n_1+1}(\cdot)$ соответствуют различные цепочки значений аргументов x_1, \dots, x_{n_1+1} , то оставляется та цепочка, на которой функция $\Lambda_{n_1+1}(\cdot)$ достигает максимума. При построении множества Λ_{n_1+1} строятся множества

$$\bar{\rho}_1^{n_1+1}, \dots, \bar{\rho}_m^{n_1+1}, \bar{\rho}_{m+m_1+1}^{t_2+1}, \dots, \bar{\rho}_{m+m_1+m_2}^{t_2+1},$$

которым принадлежат значения аргументов $\varepsilon_1^{n_1+1}, \dots, \varepsilon_m^{n_1+1}, \varepsilon_{m+m_1+1}^{t_2+1}, \dots, \varepsilon_{m+m_1+m_2}^{t_2+1}$.

Приведенные формулы (2.51)–(2.54) для построения множеств

$$\Lambda_l, l = \overline{1, n_1 - t_2}; \Lambda_{n_1 - t_2 + 1}, \Lambda_{n_1 - t_2 + i}, i = \overline{2, t_2}; \Lambda_{n_1 + 1},$$

правило построения множества Ω_i по множеству ρ_i полностью описывают алгоритм построения остальных множеств Λ_i ,

$$\rho_i, i = \overline{2, p-1}, \Omega_i, i = \overline{2, p-2},$$

множеств $\bar{\rho}_1^{\sum_{i=1}^k n_i + j - \sum_{i=2}^k t_i}, \dots, \bar{\rho}_m^{\sum_{i=1}^k n_i + j - \sum_{i=2}^k t_i}, \bar{\rho}_{m+\sum_{i=1}^k m_i + 1}, \dots,$

$$\rho_{m+\sum_{i=1}^k m_i}^{t_{k+1}+j}, j = \overline{1, n_{k+1} - t_{k+1} - t_{k+2}},$$

$$\bar{\rho}_1^{\sum_{i=1}^k n_i + j - \sum_{i=2}^k t_i}, \dots, \bar{\rho}_m^{\sum_{i=1}^k n_i + j - \sum_{i=2}^k t_i}, \bar{\rho}_{m+\sum_{i=1}^k m_i + 1}^{t_{k+1}+j}, \dots, \bar{\rho}_{m+\sum_{i=1}^k m_i}^{t_{k+1}+j},$$

$\bar{\rho}_{m+\sum_{i=1}^k m_i + 1}^j, \dots, \bar{\rho}_{m+\sum_{i=1}^k m_i}^j; j = \overline{n_{k+1} - t_{k+1} - t_{k+2} + 1, n_{k+1} - t_{k+1}},$
 которым принадлежат значения аргументов

$$\varepsilon_1^{\sum_{i=1}^k n_i - \sum_{i=2}^k t_i + j}, \dots, \varepsilon_m^{\sum_{i=1}^k n_i + j - \sum_{i=2}^k t_i}, \varepsilon_{m+\sum_{i=1}^k m_i + 1}^{t_{k+1}+j}, \dots, \varepsilon_{m+\sum_{i=1}^k m_i}^{t_{k+1}+j}$$

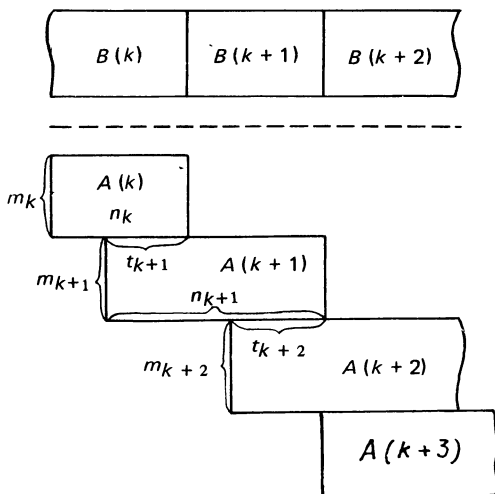
функции $\Delta_{\sum_{i=1}^k n_i - \sum_{i=2}^k t_i + j}(\cdot); j = \overline{1, n_{k+1} - t_{k+1} - t_{k+2}};$

а также аргументов $\varepsilon_1^{\sum_{i=1}^k n_i - \sum_{i=2}^k t_i + j}, \dots, \varepsilon_m^{\sum_{i=1}^k n_i - \sum_{i=2}^k t_i + j}, \varepsilon_{m+\sum_{i=1}^k m_i + 1}^{t_{k+1}+j}, \dots,$

$\dots, \varepsilon_{m+\sum_{i=1}^k m_i}^{t_{k+1}+j}, \varepsilon_{m+\sum_{i=1}^k m_i + 1}^j, \dots, \varepsilon_{m+\sum_{i=1}^k m_i}^j$
 функции

$$\Delta_{\sum_{i=1}^k n_i - \sum_{i=2}^k t_i + j}(\cdot); j = \overline{n_{k+1} - t_{k+1} - t_{k+2} + 1, n_{k+1} - t_{k+1}}; k = \overline{1, p-1}. \quad (2.55)$$

Построение множеств и функций, указанных в (2.55), относится при фиксированном k к следующему фрагменту матрицы ограничений:



Формула (2.54) для определения множества $\Lambda_{\sum_{i=1}^k n_i - \sum_{i=2}^k t_{i+1}}^k$, формула (2.51) для определения $\Lambda_{\sum_{i=1}^k n_i - \sum_{i=2}^k t_{i+1}}$, $i = 2, n_{k+1} - t_{k+1} - t_{k+2}$, формула (2.52) для построения множества $\Lambda_{\sum_{i=1}^k n_i - \sum_{i=2}^k t_{i+1} + n_{k+1} - t_{k+1} - t_{k+2} + 1}$, формула (2.53) для построения множества $\Lambda_{\sum_{i=1}^k n_i - \sum_{i=2}^k t_{i+1} + n_{k+1} - t_{k+1} - t_{k+2} + i}$, $i = 2, t_{k+2}$ преобразуются следующим образом.

$$\begin{aligned} & \text{Положим } \sum_{i=1}^k n_i - \sum_{i=2}^k t_i = \bar{n}_k, \quad m + \sum_{i=1}^k m_i = \bar{m}_k. \\ & \Lambda_{\bar{n}_k+1} = \{ \Lambda_{\bar{n}_k+1}(\varepsilon_1^{\bar{n}_k, i_1} + b_{11}(k+1) x_{\bar{n}_k+1}^j, \dots, \varepsilon_m^{\bar{n}_k, i_m} + \\ & + b_{m1}(k+1) x_{\bar{n}_k}^j, \varepsilon_{\bar{m}_k+1}^{t_{k+1}, i_{\bar{m}_k+1}} + a_{1(t_{k+1}+1)}(k+1) x_{\bar{n}_k+1}^j, \dots \\ & \dots \varepsilon_{\bar{m}_k+1}^{t_{k+1}, i_{\bar{m}_k+1}} + a_{m_{k+1}(t_{k+1}+1)}(k+1) x_{\bar{n}_k+1}^j) = \Lambda_{\bar{n}_k+1}(\varepsilon_1^{\bar{n}_k+1, j_1}, \dots \\ & \dots, \varepsilon_m^{\bar{n}_k+1, j_m}, \varepsilon_{\bar{m}_k+1}^{t_{k+1}, j_{\bar{m}_k+1}}, \dots, \varepsilon_{\bar{m}_k+1}^{t_{k+1}+1, j_{\bar{m}_k+1}}) = \sum_{i=1}^{\bar{n}_k} f_i(x_i^{j_i}) + \\ & + f_{\bar{n}_k+1}(x_{\bar{n}_k+1}^j); \forall \varepsilon_1^{\bar{n}_k, i_1}, \dots, \varepsilon_m^{\bar{n}_k, i_m}, \varepsilon_{\bar{m}_k+1}^{t_{k+1}, i_{\bar{m}_k+1}}, \dots, \\ & \varepsilon_{\bar{m}_k+1}^{\bar{n}_k, i_{\bar{m}_k+1}} \in \Omega_k; j = \overline{1, N_{\bar{n}_k+1}}; (x_1^{j_1}, \dots, x_{\bar{n}_k}^{j_{\bar{n}_k}}, x_{\bar{n}_k+1}^j) \}. \quad (2.56) \end{aligned}$$

Если одной и той же последовательности значений аргументов $\varepsilon_1^{\bar{n}_k+1}, \dots, \varepsilon_m^{\bar{n}_k+1}, \varepsilon_{\bar{m}_k+1}^{t_{k+1}+1}, \dots, \varepsilon_{\bar{m}_k+1}^{t_{k+1}+1}$ функции $\Lambda_{\bar{n}_k+1}(\cdot)$ соответствуют различные цепочки значений аргументов $\bar{x}_1, \dots, x_{\bar{n}_k+1}$, то оставляется та цепочка, на которой функция $\Lambda_{\bar{n}_k+1}(\cdot)$ достигает максимума.

Множество Ω_k строится по множеству ρ_k , определенному как декартово произведение множеств $\bar{\rho}_1^{\bar{n}_k}, \dots, \bar{\rho}_m^{\bar{n}_k}, \bar{\rho}_{\bar{m}_k+1}^{\bar{n}_k}, \dots, \bar{\rho}_{\bar{m}_k+1}^{t_{k+1}}$, которым принадлежат соответственно значения аргументов $\varepsilon_1^{\bar{n}_k}, \dots, \varepsilon_m^{\bar{n}_k}, \varepsilon_{\bar{m}_k+1}^{\bar{n}_k}, \dots, \varepsilon_{\bar{m}_k+1}^{t_{k+1}}$ функции $\Lambda_{\bar{n}_k}(\cdot)$.

Таким образом, ρ_k — это множество всех различных последовательностей значений аргументов

$$\varepsilon_1^{\bar{n}_k}, \dots, \varepsilon_m^{\bar{n}_k}, \varepsilon_{\bar{m}_k+1}^{\bar{n}_k}, \dots, \varepsilon_{\bar{m}_k+1}^{t_{k+1}}, \dots, \varepsilon_{\bar{m}_k+1}^{t_{k+1}+1}$$

полученных в результате построения множества $\Lambda_{\bar{n}_k}$. По построению множества $\Lambda_{\bar{n}_k}$ каждому элементу из множества ρ_k приписана одна последовательность значений переменных $x_1, \dots, x_{\bar{n}_k}$.

Тогда зададим множество Ω_k как множество всех значений последовательностей аргументов $\varepsilon_1^{\bar{n}_k}, \dots, \varepsilon_m^{\bar{n}_k}, \varepsilon_{\bar{m}_{k+1}}^{t_{k+1}}, \dots, \varepsilon_{\bar{m}_{k+1}}^{t_{k+1}}$ удовлетворяющих следующему условию: для каждого элемента из Ω_k существует, по крайней мере, один элемент из ρ_k , включающий в себя тот элемент из Ω_k , которому приписана последовательность значений аргументов $x_1, \dots, x_{\bar{n}_k}$, удовлетворяющая ограничения

$$(0, \dots, 0, A(k), 0, \dots, 0) x \leq b_k,$$

где $x^T = (x_1, \dots, x_{\bar{n}_k}, 0, \dots, 0)$.

Каждому элементу из Ω_k припишем только одну из возможных последовательностей значений аргументов $x_1, \dots, x_{\bar{n}_k}$, на которой достигается максимум функционала $\sum_{i=1}^{\bar{n}_k} f_i(x_i)$.

$$\begin{aligned} \Lambda_{\bar{n}_k+i} = & \{ \Lambda_{\bar{n}_k+i}(\varepsilon_1^{\bar{n}_k+i-1, i_1} + b_{i_1}(k+1) x_{\bar{n}_k+i}^j, \dots, \varepsilon_m^{\bar{n}_k+i-1, i_m} + \\ & + b_{m_i}(k+1) x_{\bar{m}_{k+1}}^j, \varepsilon_{\bar{m}_{k+1}}^{t_{k+1}+i-1, i\bar{m}_{k+1}} + a_{1(t_{k+1}+i)}(k+1) x_{\bar{n}_k+i}^j, \dots, \\ & \varepsilon_{\bar{m}_{k+1}}^{t_{k+1}+i-1, i\bar{m}_{k+1}} + a_{1(t_{k+1}+i)}(k+1) x_{\bar{n}_k+i}^j) = \Lambda_{\bar{n}_k+i}(\varepsilon_1^{\bar{n}_k+i, j_1}, \dots, \\ & \varepsilon_m^{\bar{n}_k+i, j_m}, \varepsilon_{\bar{m}_{k+1}}^{t_{k+1}+i, j\bar{m}_{k+1}}, \dots, \varepsilon_{\bar{m}_{k+1}}^{t_{k+1}+i, j\bar{m}_{k+1}}) = \sum_{l=1}^{\bar{n}_k+i-1} f_l(x_l^j) + \\ & + f_{\bar{n}_k+i}(x_{\bar{n}_k+i}^j); \forall \varepsilon_1^{\bar{n}_k+i-1} \in \bar{\rho}_1^{\bar{n}_k+i-1}, \dots, \varepsilon_m^{\bar{n}_k+i-1} \in \bar{\rho}_m^{\bar{n}_k+i-1}, \\ & \varepsilon_{\bar{m}_{k+1}}^{t_{k+1}+i-1} \in \bar{\rho}_{\bar{m}_{k+1}}^{t_{k+1}+i-1}, \dots, \varepsilon_{\bar{m}_{k+1}}^{t_{k+1}+i-1} \in \bar{\rho}_{\bar{m}_{k+1}}^{t_{k+1}+i-1}; j=1, \overline{N_{\bar{n}_k+i}}, \\ & (x_1^j, \dots, x_{\bar{n}_k+i-1}^j, x_{\bar{n}_k+i}^j) \}, \quad i=2, \overline{n_{k+1}-t_{k+1}-t_{k+2}}. \quad (2.57) \end{aligned}$$

Если одной и той же последовательности значений аргументов $\varepsilon_1^{\bar{n}_k+i}, \dots, \varepsilon_m^{\bar{n}_k+i}, \varepsilon_{\bar{m}_{k+1}}^{t_{k+1}+i}, \dots, \varepsilon_{\bar{m}_{k+1}}^{t_{k+1}+i}$ функции $\Lambda_{\bar{n}_k+i}(\cdot)$ соответствуют различные цепочки значений аргументов $x_1, \dots, x_{\bar{n}_k+i}$, то оставляется та цепочка на которой функция $\Lambda_{\bar{n}_k+i}(\cdot)$ достигает максимума.

$$\begin{aligned} \Lambda_{\bar{n}_{k+1}-t_{k+2}+1} = & \{ \Lambda_{\bar{n}_{k+1}-t_{k+2}+1}(\varepsilon_1^{\bar{n}_{k+1}-t_{k+2}+1, i_1} + b_{1(n_{k+1}-t_{k+2}-t_{k+1}+1)}(k+1) x \\ & \times x_{\bar{n}_{k+1}-t_{k+2}+1}^j, \dots, \varepsilon_m^{\bar{n}_{k+1}-t_{k+2}+1, i_m} + b_{m(n_{k+1}-t_{k+1}-t_{k+2}+1)}(k+1) x \\ & \times x_{\bar{n}_{k+1}-t_{k+2}+1}^j, \varepsilon_{\bar{m}_{k+1}}^{n_{k+1}-t_{k+2}+1, i\bar{m}_{k+1}} + a_{1(n_{k+1}-t_{k+2}+1)}(k+1) x_{\bar{n}_{k+1}-t_{k+2}+1}^j, \dots, \\ & \varepsilon_{\bar{m}_{k+1}}^{n_{k+1}-t_{k+2}+1, i\bar{m}_{k+1}} + a_{m_{k+1}(n_{k+1}-t_{k+2}+1)}(k+1) x_{\bar{n}_{k+1}-t_{k+2}+1}^j, a_{11}(k+2) x \\ & \times x_{\bar{n}_{k+1}-t_{k+2}+1}^j, \dots, a_{m_{k+2}1}(k+2) x_{\bar{n}_{k+1}-t_{k+2}+1}^j) = \\ & = \Lambda_{\bar{n}_{k+1}-t_{k+2}+1}(\varepsilon_1^{\bar{n}_{k+1}-t_{k+2}+1, j_1}, \dots, \varepsilon_m^{\bar{n}_{k+1}-t_{k+2}+1, j_m} \varepsilon_{\bar{m}_{k+1}}^{n_{k+1}-t_{k+2}+1, j\bar{m}_{k+1}}, \dots, \\ & \varepsilon_{\bar{m}_{k+1}}^{n_{k+1}-t_{k+2}+1, j\bar{m}_{k+1}}, \varepsilon_{\bar{m}_{k+1}}^{1, j\bar{m}_{k+1}+1}, \dots, \varepsilon_{\bar{m}_{k+2}}^{n_{k+1}-t_{k+2}+1, j\bar{m}_{k+2}}) = \sum_{i=1}^{\bar{n}_{k+1}-t_{k+2}+1} f_i(x_i^j) + \end{aligned}$$

$$\begin{aligned}
& + f_{\bar{n}_{k+1}-t_{k+2}+1}(x_{\bar{n}_{k+1}-t_{k+2}+1}^j); \forall \varepsilon_1^{\bar{n}_{k+1}-t_{k+2}}, i_1 \in \rho_1^{\bar{n}_{k+1}-t_{k+2}}, \dots, \\
& \varepsilon_m^{\bar{n}_{k+1}-t_{k+2}}, i_m \in \bar{\rho}_m^{\bar{n}_{k+1}-t_{k+2}}, \varepsilon_{\bar{m}_{k+1}}^{\bar{n}_{k+1}-t_{k+2}}, i_{\bar{m}_{k+1}} \in \bar{\rho}^{\bar{n}_{k+1}-t_{k+2}}, \dots, \varepsilon_{\bar{m}_{k+1}}^{\bar{n}_{k+1}-t_{k+2}}, i_{\bar{m}_{k+1}} \in \\
& \in \bar{\rho}_{\bar{m}_{k+1}}^{\bar{n}_{k+1}-t_{k+2}}, j=1, N_{\bar{n}_{k+1}-t_{k+2}+1}; (x_1^j, \dots, x_{\bar{n}_{k+1}-t_{k+2}+1}^j, x_{\bar{n}_{k+1}-t_{k+2}+1}^j) \}. \quad (2.58)
\end{aligned}$$

Если одной и той же последовательности значений аргументов

$$\varepsilon_1^{\bar{n}_{k+1}-t_{k+2}+1}, \dots, \varepsilon_m^{\bar{n}_{k+1}-t_{k+2}+1}, \varepsilon_{\bar{m}_{k+1}}^{\bar{n}_{k+1}-t_{k+2}+1}, \dots, \varepsilon_{\bar{m}_{k+1}}^{\bar{n}_{k+1}-t_{k+2}+1}, \varepsilon_{\bar{m}_{k+1}+1}, \dots, \varepsilon_{\bar{m}_{k+2}}^1$$

функции $\Lambda_{\bar{n}_{k+1}-t_{k+2}+1}(\cdot)$ соответствуют различные цепочки значений аргументов $x_1, \dots, x_{\bar{n}_{k+1}-t_{k+2}+1}$, то оставляется цепочка, которой соответствует максимум функции $\Lambda_{\bar{n}_{k+1}-t_{k+2}+1}(\cdot)$.

$$\begin{aligned}
\Lambda_{\bar{n}_{k+1}-t_{k+2}+1} &= \{ \Lambda_{\bar{n}_{k+1}-t_{k+2}+1}(\varepsilon_1^{\bar{n}_{k+1}-t_{k+2}+i-1}, i_1, b_{1/(n_{k+1}-t_{k+2}-t_{k+1}+i(k+1))} \times \\
& \times x_{\bar{n}_{k+1}-t_{k+2}+i}^j, \dots, \varepsilon_m^{\bar{n}_{k+1}-t_{k+2}+i-1}, i_m, b_m(n_{k+1}-t_{k+1}-t_{k+2}+i)(k+1)^x \\
& \times x_{\bar{n}_{k+1}-t_{k+2}+i}^j, \varepsilon_{\bar{m}_{k+1}}^{\bar{n}_{k+1}-t_{k+2}+i-1}, i_{\bar{m}_{k+1}}, a_{1/(n_{k+1}-t_{k+2}+i)}(k+1)^x \\
& \times x_{\bar{n}_{k+1}-t_{k+2}+i}^j, \dots, \varepsilon_{\bar{m}_{k+1}}^{\bar{n}_{k+1}-t_{k+2}+i-1}, i_{\bar{m}_{k+1}}, a_{\bar{m}_{k+1}}(n_{k+1}-t_{k+2}+i)(k+1)^x \\
& \times x_{\bar{n}_{k+1}-t_{k+2}+i}^j, \varepsilon_{\bar{m}_{k+1}+1}^{i-1, i_{\bar{m}_{k+1}+1}} + a_{i_1}(k+2) x_{\bar{n}_{k+1}-t_{k+2}+i}^j, \dots, \\
& \varepsilon_{\bar{m}_{k+2}}^{i-1, i_{\bar{m}_{k+2}}} + a_{m_{k+2}i}(k+2) x_{\bar{n}_{k+1}-t_{k+2}+i}^j = \Lambda_{\bar{n}_{k+1}-t_{k+2}+i}(\varepsilon_1^{\bar{n}_{k+1}-t_{k+2}+i}, j^i, \\
& \dots, \varepsilon_m^{\bar{n}_{k+1}-t_{k+2}+i}, j^m, \varepsilon_{\bar{m}_{k+1}}^{\bar{n}_{k+1}-t_{k+2}+i}, j^{\bar{m}_{k+1}}, \dots, \varepsilon_{\bar{m}_{k+1}}^{\bar{n}_{k+1}-t_{k+2}+i}, j^{\bar{m}_{k+1}}, \\
& \varepsilon_{\bar{m}_{k+1}+1}^{i, j^{\bar{m}_{k+1}+1}}, \dots, \varepsilon_{\bar{m}_{k+2}}^{i, j^{\bar{m}_{k+2}}} = \sum_{l=1}^{\bar{n}_{k+1}-t_{k+2}+i-1} f_l(x_l^j) + f_{\bar{n}_{k+1}-t_{k+2}+i}^x \\
& \times (x_{\bar{n}_{k+1}-t_{k+2}+i}^j); \forall \varepsilon_1^{\bar{n}_{k+1}-t_{k+2}+i-1} \in \bar{\rho}_1^{\bar{n}_{k+1}-t_{k+2}+i-1}, \dots, \\
& \varepsilon_m^{\bar{n}_{k+1}-t_{k+2}+i-1}, i_m \in \bar{\rho}_m^{\bar{n}_{k+1}-t_{k+2}+i-1}, \varepsilon_{\bar{m}_{k+1}}^{\bar{n}_{k+1}-t_{k+2}+i-1}, i_{\bar{m}_{k+1}} \in \\
& \in \bar{\rho}_{\bar{m}_{k+1}}^{\bar{n}_{k+1}-t_{k+2}+i-1}, \dots, \varepsilon_{\bar{m}_{k+1}}^{\bar{n}_{k+1}-t_{k+2}+i-1}, i_{\bar{m}_{k+1}} \in \bar{\rho}_{\bar{m}_{k+1}}^{\bar{n}_{k+1}-t_{k+2}+i-1}, \\
& \varepsilon_{\bar{m}_{k+1}+1}^{i-1, j^{\bar{m}_{k+1}+1}} \in \bar{\rho}_{\bar{m}_{k+1}+1}^{i-1}, \dots, \varepsilon_{\bar{m}_{k+2}}^{i-1, j^{\bar{m}_{k+2}}} \in \bar{\rho}_{\bar{m}_{k+2}}^{i-1}, j=1, N_{\bar{n}_{k+1}-t_{k+2}+i}, \\
& (x_1^j, \dots, x_{\bar{n}_{k+1}-t_{k+2}+i-1}^j, x_{\bar{n}_{k+1}-t_{k+2}+i}^j) \}, i=2, t_{k+2}. \quad (2.59)
\end{aligned}$$

Если одной и той же последовательности значений аргументов $\varepsilon_1^{\bar{n}_{k+1}-t_{k+2}+i}, \dots, \varepsilon_m^{\bar{n}_{k+1}-t_{k+2}+i}, \varepsilon_{\bar{m}_{k+1}}^{\bar{n}_{k+1}-t_{k+2}+i}, \dots, \varepsilon_{\bar{m}_{k+1}}^{\bar{n}_{k+1}-t_{k+2}+i}, \varepsilon_{\bar{m}_{k+1}+1}, \dots, \dots, \varepsilon_{\bar{m}_{k+2}}^1$, функции $\Lambda_{\bar{n}_{k+1}-t_{k+2}+i}(\cdot)$ соответствуют различные цепочки значений аргументов $x_1, \dots, x_{\bar{n}_{k+1}-t_{k+2}+i}$, то оставляется цепочка, которой соответствует максимум функции $\Lambda_{\bar{n}_{k+1}-t_{k+2}+i}(\cdot)$. Оптимальное решение задачи (2.46)–(2.50) определяется после-

довательностью значений аргументов x_1, \dots, x_n , приписанной одному из построенных значений функции $\Lambda_n(\varepsilon_1^n, \dots, \varepsilon_m^n, \varepsilon_{m-p-1}^{np}, \dots, \varepsilon_{m-p}^{np})$, для которого удовлетворяются ограничения (2.47)–(2.50) и соответствует наибольшее значение показателя качества (2.46).

Утверждение. Решение задачи (2.46)–(2.50), полученное описанным алгоритмом, является строго оптимальным решением этой задачи.

Доказательство. Для доказательства сформулированного утверждения достаточно построить функцию $\Lambda_n(\varepsilon_1^n, \dots, \varepsilon_{m+\sum_{i=1}^p m_i})$ по алгоритму, приведенному для решения задачи (2.6)–(2.7). Очевидно, значения функционала (2.46) на оптимальных решениях задач (2.46)–(2.50), построенных по функциям $\Lambda_n(\varepsilon_1^n, \dots, \varepsilon_m^n, \varepsilon_{m-p-1}^{np}, \dots, \varepsilon_{m-p}^{np})$ и $\Lambda_n(\varepsilon_1^n, \dots, \varepsilon_{m+\sum_{i=1}^p m_i})$, совпадают.

Найдем верхнюю оценку числа операций изложенного алгоритма. Для этого определим верхнюю оценку числа операций, необходимых для построения множества различных последовательностей значений аргументов и самой величины функции $\Lambda_n(\varepsilon_1^n, \dots, \varepsilon_m^n, \varepsilon_{m-p-1}^{np}, \dots, \varepsilon_{m-p}^{np})$. Эта оценка равна

$$2kn(\max_{i,j,l} |b_{ij}(l)| 2kn)^m [\max_{i,j,l} |a_{ij}(l)| \max_l n_l 2k]^{2 \max m_i}, \quad (2.60)$$

где $b_{ij}(l)$, $a_{ij}(l)$ – элементы матриц $B(l)$, $A(l)$. Под операцией понимается совокупность вычислений, необходимых для нахождения значения аргументов и самой величины функции $\Lambda_i(\cdot)$ по фиксированному значению аргументов функции $\Lambda_{i-1}(\cdot)$ для одного значения переменной $x_i \in \mathcal{D}$, а также для построения последовательности значений аргументов x_1, \dots, x_i , приписанных значению функции $\Lambda_i(\cdot)$ (если этой последовательности значений аргументов $\Lambda_i(\cdot)$ была уже приписана последовательность значений x_1, \dots, x_i , то она сохраняется либо заменяется новой, для которой значение функции $\Lambda_i(\cdot)$ имеет большую величину). Оценка (2.60) является полиномиальной от n , когда полиномами от n являются следующие параметры:

$$k \leq O(n^{t_1}), \max_{i,j,l} |b_{ij}(l)| \leq O(n^{t_2}), \\ \max_{i,j,l} |a_{ij}(l)| \leq O(n^{t_3}), \quad (2.61)$$

где $O(n^{t_i})$, $i = \overline{1, 3}$ полиномы от n степеней t_1, t_2, t_3 соответственно, $\forall m, m_i \leq T$, T от n не зависит. Тогда оценка (2.60) будет иметь вид

$$2[O(n^{t_1})]n \{ [O(n^{t_2})] 2[O(n^{t_1})]n \}^m \times \\ \times \{ [O(n^{t_3})] \max_l n_l 2[O(n^{t_1})] \}^{2 \max m_i} \quad (2.62)$$

и явится полиномом фиксированной степени от n -числа переменных задачи. Отсюда можно утверждать, что оценка элементарных операций, необходимых для построения оптимального решения задачи (2.46)–(2.50), окажется также полиномиальной относительно n .

Модель (2.46)–(2.50) в общем случае не позволяет построить алгоритм с полиномиальными оценками сложности для случая

$$\forall m_i \leq O(n^{2i}), \quad n_i \leq T,$$

т. е. когда число строк матриц $A(i)$, $i = \overline{1, p}$ ограничено полиномом фиксированной степени от n , а число столбцов ограничено величиной, от n не зависящей. В этом отличие модели (2.46)–(2.50) п. 2 от (2.24)–(2.28).

Однако если матрицы $A(i)$, $i = \overline{1, p}$ удовлетворяют условиям чередования в случае, когда

$$m_i \leq O(n^{2i}), \quad n_i \leq T, \quad m_{(i \pm 1)} \leq T, \quad n_{(i \pm 1)} \leq O(n^{2i \pm 1}),$$

то полиномиальный относительно n алгоритм может быть построен очевидным образом. Рассмотрим его на следующем фрагменте:

$$m \left\{ \begin{array}{|c|c|c|} \hline B(1) & B(2) & B(3) \\ \hline \end{array} \right\} \stackrel{\leq}{=} b \quad (2.63)$$

$$m_1 \left\{ \begin{array}{|c|} \hline n_1 A(1) \\ \hline \end{array} \right\} \stackrel{\leq}{=} b_1 \quad (2.64)$$

$$m_2 \left\{ \begin{array}{|c|c|} \hline t_2 \quad n_2 A(2) \\ \hline \end{array} \right\} \stackrel{\leq}{=} b_2 \quad (2.65)$$

$$m_3 \left\{ \begin{array}{|c|} \hline t_3 A(3) \\ \hline \end{array} \right\} \stackrel{\leq}{=} b_3$$

Пусть $m_1 \leq T_1$, $m_2 \leq O(n^{2^2})$, $m_3 \leq T_1$, $n_2 \leq T_2$. Строим функцию $\Lambda_{n_1-t_2}(\varepsilon_1^{n_1-t_2}, \dots, \varepsilon_{m+m_1}^{n_1-t_2})$ и множество $\bar{p}^{n_1-t_2}$, элементами

которого являются все последовательности значений аргументов $\varepsilon_1^{n_1-t_2}, \dots, \varepsilon_{m+m_1}^{n_1-t_2}$. Формируем не более чем $(2k)^{n_2}$ различных последовательностей значений переменных $x_{n_1-t_2+1}, \dots, x_{n_1+n_2-t_2}$. Множество этих последовательностей обозначим символом Λ^1 .

Строим множество

$$\begin{aligned} \Lambda_{n_1+n_2-t_2} &= \{ \Lambda_{n_1+n_2-t_2}(\varepsilon_1^{n_1-t_2}, i_1 + \sum_{j=n_1-t_2+1}^{n_1} b_{1j}(1) x_j^{k_j} + \\ &+ \sum_{j=1}^{n_2-t_2} b_{1j}(2) x_{n_1+j}^{k_{n_1+j}}, \dots, \varepsilon_m^{n_1-t_2}, i_m + \sum_{j=n_1-t_2+1}^{n_1} x_j^{k_j} b_{mj}(1) + \sum_{j=1}^{n_2-t_2} b_{mj}(2) x \\ &\times x_{n_1+j}^{k_{n_1+j}}, \sum_{j=1}^{t_3} a_{1j}(3) x_{n_1+n_2-t_2-t_3+j}^{k_{n_1+n_2-t_2-t_3+j}}, \dots, \sum_{j=1}^{t_3} a_{m3j}(3) x_{n_1+n_2-t_2-t_3+j}^{k_{n_1+n_2-t_2-t_3+j}} \} = \\ &= \Lambda_{n_1+n_2-t_2}(\varepsilon_1^{n_1+n_2-t_2, j_1}, \dots, \varepsilon_m^{n_1+n_2-t_2, j_m}, \varepsilon_{m+m_1+m_2+1}^{n_1+n_2-t_2, j_{m+m_1+m_2+1}}, \dots, \end{aligned}$$

$$\begin{aligned} \varepsilon_{m+m_1+m_2+m_3}^{n_1+n_2-t_2, j, m+m_1+m_2+m_3} &= \sum_{j=1}^{n_1-t_2} f_j(x_j^{ij}) + \sum_{j=n_1-t_2+1}^{n_1+n_2-t_2} f_j(x_j^{kj}), \\ \forall (\varepsilon_1^{n_1-t_2, i_1}, \dots, \varepsilon_m^{n_1-t_2, i_m}) \in \rho_m^{n_1-t_2}, \forall (x_{n_1-t_2+1}^{k_{n_1-t_2+1}}, \dots, x_{n_1+n_2-t_2}^{k_{n_1+n_2-t_2}}) \in \Lambda_1, \\ (x_1^{i_1}, \dots, x_{n_1-t_2}^{i_{n_1-t_2}}, x_{n_1-t_2+1}^{k_{n_1-t_2+1}}, \dots, x_{n_1+n_2-t_2}^{k_{n_1+n_2-t_2}}) &\} \end{aligned} \quad (2.66)$$

где $\rho_m^{n_1-t_2}$ – множество всех различных последовательностей значений аргументов $\varepsilon_1^{n_1-t_2}, \dots, \varepsilon_m^{n_1-t_2}$, входящих в множество $\bar{\rho}^{n_1-t_2}$.

Опишем некоторые особенности построения множества $\Lambda_{n_1+n_2-t_2}$ и функции $\Lambda_{n_1+n_2-t_2}(\cdot)$. Каждому элементу из множества $\rho_m^{n_1-t_2}$ может соответствовать более чем одна цепочка значений переменных $x_1, \dots, x_{n_1-t_2}$. В выражении (2.66) цепочки значений переменных $x_1, \dots, x_{n_1-t_2}$ обозначены $x_1^{i_1}, \dots, x_{n_1-t_2}^{i_{n_1-t_2}}$. Цепочки значений переменных $x_{n_1-t_2+1}, \dots, x_{n_1+n_2-t_2}$ обозначены $x_{n_1-t_2+1}^{k_{n_1-t_2+1}}, \dots, x_{n_1+n_2-t_2}^{k_{n_1+n_2-t_2}}$.

Таким образом, фиксированной последовательности значений аргументов

$$\varepsilon_1^{n_1-t_2}, \dots, \varepsilon_m^{n_1-t_2} \rightarrow \varepsilon_1^{n_1-t_2, i_1}, \dots, \varepsilon_m^{n_1-t_2, i_m},$$

а также фиксированной последовательности значений аргументов

$$x_{n_1-t_2+1}, \dots, x_{n_1+n_2-t_2} \rightarrow x_{n_1-t_2+1}^{k_{n_1-t_2+1}}, \dots, x_{n_1+n_2-t_2}^{k_{n_1+n_2-t_2}},$$

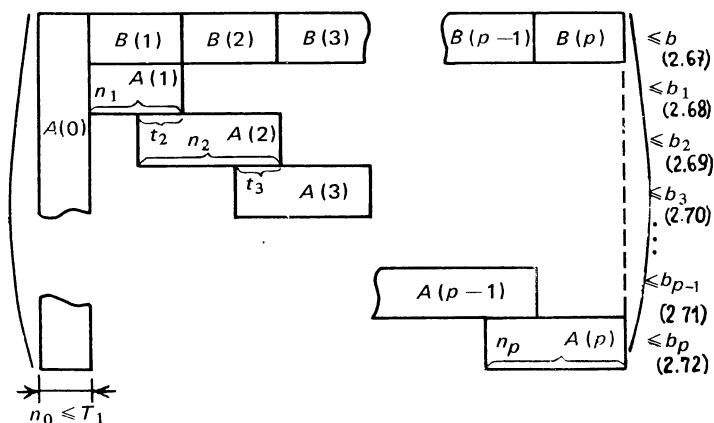
определяющих последовательность значений аргументов функции $\Lambda_{n_1+n_2-t_2}(\cdot)$, может соответствовать множество различных последовательностей значений переменных $x_1, \dots, x_{n_1+n_2-t_2}$.

Обозначим через $\rho^{n_1+n_2-t_2}$ множество полученных в результате построения различных значений аргументов функции $\Lambda_{n_1+n_2-t_2}(\cdot)$. Из множества $\rho^{n_1+n_2-t_2}$ исключим те элементы, у которых все приписанные цепочки значений переменных $x_1, \dots, x_{n_1+n_2-t_2}$ не удовлетворяют ограничениям (2.64)–(2.65). У оставшихся элементов множества $\rho^{n_1+n_2-t_2}$ исключим все цепочки значений переменных $x_1, \dots, x_{n_1+n_2-t_2}$ не удовлетворяющих ограничениям (2.64)–(2.65). Если после описанной процедуры элементам из множества $\rho^{n_1+n_2-t_2}$ остаются приписанными более чем одна цепочка значений переменных $x_1, \dots, x_{n_1+n_2-t_2}$ то остается одна цепочка, на которой функция $\Lambda_{n_1+n_2-t_2}(\cdot)$ принимает максимальное значение. Эта цепочка и фигурирует в выражении (2.66). Таким образом сконструированное множество обозначим по-прежнему (по аналогии с предыдущими обозначениями) через $\bar{\rho}^{n_1+n_2-t_2}$.

Для $i = n_1 + n_2 - t_2 + 1, \dots, n_1 + n_2 + n_3 - t_2 - t_3 - t_4$ $\Lambda_i(\cdot)$ строятся по обычной схеме, после чего описанная процедура повторяется т. д. По построенной функции $\Lambda_n(\cdot)$

определяется оптимальное решение задачи. Легко видеть, что предложенный алгоритм имеет полиномиальную оценку сложности относительно η -числа переменных задачи. „Лестничную” структуру ограничений (2.46)–(2.50) можно усложнить.

Пусть матрица имеет вид



Коэффициенты матриц $A(0)$, $A(i)$, $B(i)$, $i = \overline{1, p}$ — целые числа. Матрица ограничений (2.67)–(2.72) отличается от матрицы ограничений (2.47)–(2.50) лишь дополнительной матрицей $A(0)$, число столбцов которой не превышает величины T_1 , T_1 не зависит от η -числа переменных задачи. Задача (2.46), (2.67)–(2.72) может быть решена алгоритмом, полиномиальным относительно η . Действительно, формируем не более чем $(2k)^{n_0}$ различных последовательностей значений переменных x_1, \dots, x_{n_0} . Множество этих последовательностей обозначим символом Λ^0 . Строим множество Λ_{r_0} и функцию $\Lambda_{n_0}(\cdot)$ следующим образом:

$$\begin{aligned} \Lambda_{n_0} &= \left\{ \Lambda_{n_0} \left(\sum_{j=1}^{n_0} a_{1j} x_j^{k_j}, \dots, \sum_{j=1}^{n_0} a_{\overline{m}pj} x_j^{k_j} \right) = \Lambda_{n_0} (\varepsilon_1^{n_0, i_1}, \dots, \varepsilon_{\overline{m}p}^{n_0, j_1 m_p}) = \right. \\ &= \left. \sum_{j=1}^{n_0} f_j(x_j^{k_j}); \forall (x_1^{k_1}, \dots, x_{n_0}^{k_{n_0}}) \in \Lambda^0; (x_1^{k_1}, \dots, x_{n_0}^{k_{n_0}}) \right\}. \end{aligned}$$

Если одной и той же последовательности значений аргументов $\varepsilon_1^{n_0, i_1}, \dots, \varepsilon_{\overline{m}p}^{n_0, j_1 m_p}$ приписаны различные цепочки значений переменных x_1, \dots, x_{n_0} то остается одна, на которой функция $\Lambda_{n_0}(\cdot)$ достигает максимума.

Организуем множество p^{n_0} — множество всех различных последовательностей значений аргументов функции $\Lambda_{n_0}(\cdot)$. Λ_{n_0+i} , $\Lambda_{n_0+i}(\cdot)$, $i = 1, n_1 - t_2$ строятся по следующему алгоритму:

$$\begin{aligned} \Lambda_{n_0+i} &= \{ \Lambda_{n_0+i}(\varepsilon_1^{n_0+i-1, i_1} + b_{i_1}(1) x_{n_0+i}^j, \dots, \varepsilon_m^{n_0+i-1, i_m} + b_{m_i}(1) x_{n_0+i}^j, \\ &\varepsilon_{m+1}^{n_0+i-1, i_{m+1}} + a_{i_1}(1) x_{n_0+i}^j, \dots, \varepsilon_{\bar{m}_1}^{n_0+i-1, i_{\bar{m}_1}} + a_{m_i}(1) x_{n_0+i}^j, \varepsilon_{\bar{m}_1+1}^{n_0, i_{\bar{m}_1+1}}, \dots, \\ &\dots, \varepsilon_{\bar{m}_p}^{n_0, i_{\bar{m}_p}}) = \Lambda_{n_0+i}(\varepsilon_1^{n_0+i, j_1}, \dots, \varepsilon_{\bar{m}_1}^{n_0+i, j_{\bar{m}_1}}, \dots, \varepsilon_{\bar{m}_1+1}^{n_0, i_{\bar{m}_1+1}}, \dots, \varepsilon_{\bar{m}_p}^{n_0, i_{\bar{m}_p}}) = \\ &= \sum_{k=1}^{n_0+i-1} f_k(x_k^{j_k}) + f_{n_0+i}(x_{n_0+i}^j); \forall (\varepsilon_1^{n_0+i-1, i_1}, \dots, \varepsilon_{\bar{m}_1}^{n_0+i-1, i_{\bar{m}_1}}, \varepsilon_{\bar{m}_1+1}^{n_0, i_{\bar{m}_1+1}}, \\ &\dots, \varepsilon_{\bar{m}_p}^{n_0, i_{\bar{m}_p}}) \in p^{n_0+i-1}, j = \overline{1, N_{n_0+i}}, (x_1^j, \dots, x_{n_0+i-1}^{j_{n_0+i-1}}, x_{n_0+i}^j) \}. \end{aligned}$$

Если одной и той же последовательности значений аргументов $\varepsilon_1^{n_0+i}, \dots, \varepsilon_{\bar{m}_1}^{n_0+i}, \varepsilon_{\bar{m}_1+1}^{n_0}, \dots, \varepsilon_{\bar{m}_p}^{n_0}$ приписаны различные цепочки значений переменных x_1, \dots, x_{n_0+i} , то остается одна, на которой функция $\Lambda_{n_0+i}(\cdot)$ достигает максимума.

$\Lambda_{n_0+i}, \Lambda_{n_0+i}(\cdot), i = \overline{1, t_2+1, n_1}$ строятся следующим образом:

$$\begin{aligned} \Lambda_{n_0+i} &= \{ \Lambda_{n_0+i}(\varepsilon_1^{n_0+i-1, i_1} + b_{i_1}(1) x_{n_0+i}^j, \dots, \varepsilon_m^{n_0+i-1, i_m} + b_{m_i}(1) x_{n_0+i}^j, \\ &\varepsilon_{m+1}^{n_0+i-1} + a_{i_1}(1) x_{n_0+i}^j, \dots, \varepsilon_{m+m_1}^{n_0+i-1, i_{m+m_1}} + a_{m_i}(1) x_{n_0+i}^j, \\ &\varepsilon_{\bar{m}_1+1}^{n_0+i-1-n_1+t_2, i_{\bar{m}_1+1}} + a_{i_1(i-n_1+t_2)}(2) x_{n_0+i}^j, \dots, \varepsilon_{\bar{m}_2}^{n_0+i-1-n_1+t_2, i_{\bar{m}_2}} + \\ &+ a_{m_2(i-n_1+t_2)}(2) x_{n_0+i}^j, \varepsilon_{\bar{m}_2+1}^{n_0, i_{\bar{m}_2+1}}, \dots, \varepsilon_{\bar{m}_p}^{n_0, i_{\bar{m}_p}}) = \\ &= \Lambda_{n_0+i}(\varepsilon_1^{n_0+i, j_1}, \dots, \varepsilon_{\bar{m}_1}^{n_0+i, j_{\bar{m}_1}}, \varepsilon_{\bar{m}_1+1}^{n_0+i-n_1+t_2, j_{\bar{m}_1+1}}, \dots, \varepsilon_{\bar{m}_2}^{n_0+i-n_1+t_2, i_{\bar{m}_2}}, \\ &\varepsilon_{\bar{m}_2+1}^{n_0, i_{\bar{m}_2+1}}, \dots, \varepsilon_{\bar{m}_p}^{n_0, i_{\bar{m}_p}}) = \sum_{k=1}^{n_0+i-1} f_k(x_k^{j_k}) + f_{n_0+i}(x_{n_0+i}^j), \\ &\forall (\varepsilon_1^{n_0+i-1, i_1}, \dots, \varepsilon_{\bar{m}_1}^{n_0+i-1, i_{\bar{m}_1}}, \varepsilon_{\bar{m}_1+1}^{n_0+i-1-n_1+t_2, i_{\bar{m}_1+1}}, \dots, \varepsilon_{\bar{m}_2}^{n_0+i-1-n_1+t_2, i_{\bar{m}_2}}, \\ &\varepsilon_{\bar{m}_2+1}^{n_0, i_{\bar{m}_2+1}}, \dots, \varepsilon_{\bar{m}_p}^{n_0, i_{\bar{m}_p}}) \in p^{n_0+i-1}; j = \overline{1, N_{n_0+i}}; \\ &(x_1^j, \dots, x_{n_0+i-1}^{j_{n_0+i-1}}, x_{n_0+i}^j) \}. \end{aligned}$$

p^{n_0+i} – множество различных последовательностей значений аргументов функции $\Lambda_{n_0+i}(\cdot)$. Если для одной и той же последовательности значений аргументов

$$\varepsilon_1^{n_0+i}, \dots, \varepsilon_{\bar{m}_1}^{n_0+i}, \varepsilon_{\bar{m}_1+1}^{n_0+i-n_1+t_2}, \dots, \varepsilon_{\bar{m}_2}^{n_0+i-n_1+t_2}, \varepsilon_{\bar{m}_2+1}^{n_0}, \dots, \varepsilon_{\bar{m}_p}^{n_0}$$

приписаны различные цепочки значений переменных x_1, \dots, x_{n_0+i} , то остается одна, на которой функция $\Lambda_{n_0+i}(\cdot)$ достигает максимума.

Рассмотрим множество $p^{n_0+n_1}$. Исключим из него все элементы, которым приписаны цепочки, не удовлетворяющие ограничениям (2.68). Из множества $p^{n_0+n_1}$ сформируем множество

(для удобства обозначим его также $p^{n_0+n_1}$), состоящее из различных последовательностей значений аргументов $\varepsilon_1^{n_0+n_1}, \dots, \dots, \varepsilon_m^{n_0+n_1}, \varepsilon_{\bar{m}_1+1}^{n_0+t_2}, \dots, \varepsilon_{\bar{m}_2}^{n_0+t_2}, \varepsilon_{\bar{m}_2+1}^{n_0}, \dots, \varepsilon_{\bar{m}_p}^{n_0}$. Если при этом одной и той же последовательности переменных $\varepsilon_1^{n_0+n_1}, \dots,$

$\dots, \varepsilon_m^{n_0+n_1}, \varepsilon_{\bar{m}_1+1}^{n_0+t_2}, \dots, \varepsilon_{\bar{m}_2}^{n_0+t_2}, \varepsilon_{\bar{m}_2+1}^{n_0}, \dots, \varepsilon_{\bar{m}_p}^{n_0}$ приспаны различные цепочки значений переменных $x_1, \dots, x_{n_0+n_1}$, то оставляем одну, на которой функция $\Lambda_{n_0+n_1}(\cdot)$ достигает максимума.

Для $i = 1, n_2 - t_2 - t_3$:

$$\begin{aligned} \Lambda_{n_0+n_1+i} &= \{ \Lambda_{n_0+n_1+i}(\varepsilon_1^{n_0+n_1+i-1, i_1}, b_{1i}(2) x_{n_0+n_1+i}^j, \dots, \varepsilon_m^{n_0+n_1+i-1, i_m} \\ &+ b_{mi}(2) x_{n_0+n_1+i}^j, \varepsilon_{\bar{m}_1+1}^{n_0+t_2+i-1, i_{\bar{m}_1+1}}, a_{1(t_2+l)}(2) x_{n_0+n_1+i}^j, \dots, \\ &\varepsilon_{\bar{m}_2}^{n_0+t_2+i-1, i_{\bar{m}_2}}, a_{m_2(t_2+i)}(2) x_{n_0+n_1+i}^j, \varepsilon_{\bar{m}_2+1}^{n_0, i_{\bar{m}_2+1}}, \dots, \varepsilon_{\bar{m}_p}^{n_0, i_{\bar{m}_p}}) = \\ &= \Lambda_{n_0+n_1+i}(\varepsilon_1^{n_0+n_1+i, j_1}, \dots, \varepsilon_m^{n_0+n_1+i, j_m}, \varepsilon_{\bar{m}_1+1}^{n_0+t_2+i, j_{\bar{m}_1+1}}, \dots, \varepsilon_{\bar{m}_2}^{n_0+t_2+i, j_{\bar{m}_2}}, \\ &\varepsilon_{\bar{m}_2+1}^{n_0, i_{\bar{m}_2+1}}, \dots, \varepsilon_{\bar{m}_p}^{n_0, i_{\bar{m}_p}}) = \sum_{k=1}^{n_0+n_1+i-1} f_k(x_k^{j_k}) + f_{n_0+n_1+i}(x_{n_0+n_1+i}^j); \forall \varepsilon_1^{n_0+n_1+i, i_1}, \dots, \\ &\dots, \varepsilon_m^{n_0+n_1+i-1, i_m}, \varepsilon_{\bar{m}_1+1}^{n_0+t_2+i-1, i_{\bar{m}_1+1}}, \dots, \varepsilon_{\bar{m}_2}^{n_0+t_2+i-1, i_{\bar{m}_2}}, \varepsilon_{\bar{m}_2+1}^{n_0, i_{\bar{m}_2+1}}, \dots, \varepsilon_{\bar{m}_p}^{n_0, i_{\bar{m}_p}}) \in \\ &\in p^{n_0+n_1+i-1}; j = \overline{1, N_{n_0+n_1+i}}; (x_1^{j_1}, \dots, x_{n_0+n_1+i-1}^{j_{n_0+n_1+i-1}}, x_{n_0+n_1+i}^j) \}. \end{aligned}$$

Для всех $i \geq n_0 + n_1 + n_2 - t_2 - t_3$ $\Lambda_i, \Lambda_i(\cdot)$ строится очевидным образом. В частности, функция $\Lambda_n(\cdot)$ является функцией следующих аргументов:

$$\varepsilon_1^n, \dots, \varepsilon_m^n, \varepsilon_{\bar{m}_p-1+1}^{n_p}, \dots, \varepsilon_{\bar{m}_p}^{n_p}.$$

Оптимальным решением задачи (2.46), (2.67)–(2.72) является та приспанный одной из последовательности значений аргументов функции $\Lambda_n(\cdot)$ цепочка значений переменных x_1, \dots, x_n , которая удовлетворяет ограничениям (2.67)–(2.72) и на которой функция $\Lambda_n(\cdot)$ достигает максимума.

Для доказательства полиномиальной сложности изложенного алгоритма необходимо показать, что количество элементов множеств p^i ограничено полиномом фиксированной степени от n . Множества

$$p^{\bar{n}_k}, k = \overline{2, p-1}, \bar{n}_k = n_0 + \sum_{i=1}^k n_i - \sum_{i=2}^k t_i$$

необходимо рассматривать до преобразования.

Количество элементов в множестве p^{n_0} не превышает величины $(2k)^{n_0}$. Количество элементов в множествах p^{n_0+i} , $i = \overline{1, n_1}$ удовлетворяет условию $|p^{n_0+i}| \leq |p^{n_0+n_1}|$. Количество элементов множества $p^{n_0+n_1}$ до преобразования не превышает величины

$$(2k)^{n_0}(2kan_1)^{m+m_1}(2kat_2)^{m_2}, \quad (2.73)$$

где $\alpha = \max_{i,j,t} \{|b_{ij}(t)|, |a_{ij}(t)|\}$.

После преобразования количество элементов множества $p^{n_0+n_1-t_2}$ не превышает величины

$$(2k)^{n_0}(2kan_1)^m(2kat_2)^{m_2}. \quad (2.74)$$

Аналогично до преобразования множеств $p^{\bar{n}_k}$, $k = \overline{2, p-1}$,

$$p^{\bar{n}_k-j} \leq |p^{\bar{n}_k}|, \quad j = \overline{1, n_k - t_k}, \quad k = \overline{2, p-1}.$$

Количество элементов множества $\bar{p}^{\bar{n}_k}$ до преобразования не превышает величины:

$$(2k)^{n_0}(2k\alpha(\bar{n}_k - n_0))^m(2k\alpha \cdot n_k)^{m_k}(2kat_{k+1})^{m_{k+1}} \quad (2.75)$$

После преобразования количество элементов множеств $p^{\bar{n}_k}$ не превышает величины

$$(2k)^{n_0}[2k\alpha(\bar{n}_k - n_0)]^m(2kat_{k+1})^{m_{k+1}}, \quad k = \overline{2, p-1}. \quad (2.76)$$

Количество элементов множеств p^n не превышает величины

$$(2k)^{n_0}[2k\alpha(n - n_0)]^m(2kan_p)^{m_p}. \quad (2.77)$$

Формулы (2.76), (2.77) являются полиномами фиксированной степени от n , они остаются полиномиальными, если величины k и α ограничены полиномами фиксированной степени от n .

Очевидно, что все приведенные в п. 4 алгоритмы требуют объема памяти, не превышающей по величине полинома фиксированной степени от n бит.

Пример

$$\max \left\{ f = \frac{1}{2}x_1^2 + 2x_2 + 3x_3 + \frac{1}{2}x_4^2 + x_5 - x_6 - \frac{1}{4}x_7^3 + x_8 - 6x_9^3 + \right. \\ \left. + 3x_{10}^3 + 2x_{11}^2 + x_{12} - x_{13} - x_{14} + x_{15}^3 \right\}.$$

$$x_1 + x_2 - 2x_3 + x_4 - x_5 + x_6 + x_7 - x_8 - x_9 - x_{10} - x_{11} + x_{12} - 2x_{13} - 2x_{14} + 3x_{15} \leq 6, \quad (1)$$

$$x_1 - x_2 + 3x_3 - 5x_4 + 2x_5 - x_6 - x_7 + x_8 + x_9 - 2x_{10} + 3x_{11} - x_{12} + x_{13} - x_{14} - x_{15} \leq 3, \quad (2)$$

$$2x_1 + 3x_2 - x_3 \leq 9, \quad (3)$$

$$x_1 - x_2 + x_3 \leq 1, \quad (4)$$

$$2x_3 + 3x_4 - x_5 \leq 5, \quad (5)$$

$$x_3 - x_4 + 2x_5 \leq 5, \quad (6)$$

$$\begin{array}{rcl}
x_4 - x_5 + x_6 & \leq & 4, (7) \\
-x_4 + x_5 - x_6 & \leq & 3, (8) \\
x_7 + x_8 + x_9 & \leq & 3, (9) \\
2x_7 + x_8 + x_9 & \leq & 4, (10) \\
-5x_9 + x_{10} - x_{11} & \leq & 2, (11) \\
6x_9 + x_{10} + x_{11} & \leq & -4, (12) \\
x_{11} - x_{12} - x_{13} & \leq & 4, (13) \\
x_{11} + x_{12} + x_{13} & \leq & 3, (14) \\
4x_{13} + x_{14} - 2x_{15} & \leq & 2, (15) \\
5x_{13} + x_{14} + 2x_{15} & \leq & -6, (16)
\end{array}$$

$$\begin{array}{l}
x_1 = 0, 2; \quad x_2 = 0, 1, 2; \quad x_3 = 0, 1, 2; \quad x_4 = 0, 2; \quad x_5 = 0, 2, 4, 6; \\
x_6 = 0, 1, 2; \quad x_7 = -2, -1, 0, 1, 2; \quad x_8 = 0, 1, 3, 6; \quad x_9 = -6, -1, 0, 1, 6; \\
x_{10} = -1, 0, 1, 2; \quad x_{11} = -1, 0, 1, 2; \quad x_{12} = -1, 0, 1, 2; \quad x_{13} = -2, 0, 2; \\
x_{14} = 1, 2, 3; \quad x_{15} = 1, 2, 3.
\end{array}$$

Решение приведенного примера потребовало построения 916 целочек. Оптимальное решение следующее: $\max f = 15$

$$\begin{array}{l}
x_1 = 2; \quad x_2 = 2; \quad x_3 = 1; \quad x_4 = 2; \quad x_5 = 2; \quad x_6 = 0; \quad x_7 = -2; \\
x_8 = 1; \quad x_9 = -1; \quad x_{10} = -1; \quad x_{11} = 2; \quad x_{12} = 2; \quad x_{13} = -2; \quad x_{14} = 1; \quad x_{15} = 1.
\end{array}$$

При решении примера сначала конструировались допустимые цепочки переменных по блокам неравенств: (3)–(4); (5), (6); (7), (8); (9), (10); (11), (12); (13), (14); (15), (16). Затем решение находилось по неравенствам (1), (2) в соответствии с описанной вычислительной процедурой по построенным на первом этапе допустимым цепочкам.

5. АЛГОРИТМЫ РЕШЕНИЯ ОДНОГО КЛАССА ЗАДАЧ ЦЕЛОЧИСЛЕННОГО СЕПАРАБЕЛЬНОГО ПРОГРАММИРОВАНИЯ С ЧАСТИЧНО ЗАПОЛНЕННОЙ МАТРИЦЕЙ ОГРАНИЧЕНИЙ, ИМЕЮЩИХ ПОЛИНОМИАЛЬНУЮ ОЦЕНКУ СЛОЖНОСТИ ОТНОСИТЕЛЬНО ЧИСЛА ПЕРЕМЕННЫХ

Рассмотрим задачу сепарабельного программирования вида

$$\max \sum_{i=1}^n f_i(x_i), \quad (2.78)$$

$$Ax \leq b, \quad \forall x_j \equiv 0 \pmod{1}, \quad (2.79)$$

$$\forall x_i \in \mathcal{D}_i, \quad \mathcal{D}_i = \{x_i^j\}, \quad j = \overline{1, N_i}, \quad |x_i| \leq k.$$

Матрица A удовлетворяет следующим условиям:

а) последние $n_i \leq T_1$ столбцов ее могут содержать все ненулевые элементы; T_1 – постоянная величина, не зависящая от n ;

б) коэффициенты матрицы A – целые числа;

в) $A^{(i)}$ – i -е столбцы матрицы A содержат не более $m_i \leq T_2$ ненулевых элементов, T_2 не зависит от n ;

г) m – число строк матрицы A ограничено полиномом фиксированной степени от n .

Покажем, что задача (2.78), (2.79) может быть решена алгоритмом с псевдополиномиальной оценкой сложности относительно n -числа переменных задачи (2.78), (2.79). Строим множество

$$\Lambda_j = \{ \Lambda_j(0, \dots, 0, a_{l_1} x_1^j, 0, \dots, 0, a_{l_m} x_m^j, 0, \dots) = \\ = \Lambda_j(0, \dots, \varepsilon_{l_1}^{1, l_1}, 0, \dots, 0, \varepsilon_{l_m}^{1, l_m}, 0, \dots, 0) = f_j(x_1^j); j = \overline{1, N_j}, (x_1^j) \}, \quad (2.80)$$

где a_{l_1}, \dots, a_{l_m} – ненулевые элементы столбца $A^{(1)}$.

Конструируем ρ^1 – множество различных последовательностей значений аргументов $\varepsilon_{l_1}^1, \dots, \varepsilon_{l_m}^1$ полученных в процессе построения множества Λ_j .

Рассмотрим алгоритм построения

$$\Lambda_i = \{ \Lambda_i(\varepsilon_1^{i-1, l_1}, \dots, \varepsilon_{p_1}^{i-1, l_{p_1}}, \varepsilon_{p_1}^{i-1, l_{p_1}+1}, \dots, \varepsilon_{p_1+1}^{i-1, l_{p_1}+1}, \dots, \\ \varepsilon_{p_2}^{i-1, l_{p_2}}, \varepsilon_{p_2}^{i-1, l_{p_2}+1}, \dots, \varepsilon_{p_{m_i}}^{i-1, l_{p_{m_i}}}, \varepsilon_{p_{m_i}}^{i-1, l_{p_{m_i}}+1}, \dots, \varepsilon_{p_{m_i}+1}^{i-1, l_{p_{m_i}}+1}, \dots, \\ \varepsilon_{p_{m_i}}^{i-1, l_{p_{m_i}+1}}, \dots, \varepsilon_m^{i-1, l_m}) = \Lambda_i(\varepsilon_1^{i, l_1} = \varepsilon_1^{i-1, l_1}, \dots, \varepsilon_{p_1}^{i, l_{p_1}} = \varepsilon_{p_1}^{i-1, l_{p_1}}, \\ \varepsilon_{p_1}^{i, l_{p_1}+1} = \varepsilon_{p_1}^{i-1, l_{p_1}+1}, \dots, \varepsilon_{p_2}^{i, l_{p_2}} = \varepsilon_{p_2}^{i-1, l_{p_2}}, \varepsilon_{p_2}^{i, l_{p_2}+1} = \varepsilon_{p_2}^{i-1, l_{p_2}+1}, \dots, \\ \dots, \varepsilon_{p_{m_i}}^{i, l_{p_{m_i}}} = \varepsilon_{p_{m_i}}^{i-1, l_{p_{m_i}}}, \varepsilon_{p_{m_i}}^{i, l_{p_{m_i}+1}} = \varepsilon_{p_{m_i}}^{i-1, l_{p_{m_i}+1}}, \\ \dots, \varepsilon_{p_{m_i}+1}^{i, l_{p_{m_i}+1}} = \varepsilon_{p_{m_i}+1}^{i-1, l_{p_{m_i}+1}}, \dots, \varepsilon_m^{i, l_m} = \varepsilon_m^{i-1, l_m}) = \sum_{j=1}^{i-1} f_j(x_1^j) + f_i(x_i^i), \\ \forall (\varepsilon_1^{i-1, l_1}, \dots, \varepsilon_m^{i-1, l_m}) \in \rho^{i-1}, j = \overline{1, N_i}; \\ (x_1^1, \dots, x_{i-1}^{i-1}, x_i^i) \}, \quad i = \overline{1, n-n_1}, \quad (2.81)$$

где $a_{p_i}, a_{p_{m_i}}$ – ненулевые элементы столбца $A^{(i)}$ матрицы A .

Если одной и той же последовательности значений аргументов $\varepsilon_1^i, \dots, \varepsilon_m^i$ приписаны различные цепочки значений переменных x_1, \dots, x_i то остается одна, на которой функция $\Lambda_i(\cdot)$ достигает максимума; $\rho^i, i = 2, n - n_1$ – множество различных последовательностей значений аргументов $\varepsilon_1^i, \dots, \varepsilon_m^i$, полученное в результате построения функции $\Lambda_i(\cdot)$ в соответствии с (2.81).

Рассмотрим функцию $\Lambda_{n-n_1}(\cdot)$ и множество различных последовательностей значений ее аргументов $\varepsilon_1^{n-n_1}, \dots, \varepsilon_m^{n-n_1}$, обозначенное ρ^{n-n_1} . Формируем не более $(2k)^{n_1}$ различных последовательностей значений переменных x_{n-n_1+1}, \dots, x_n , множество которых обозначим символом Λ^1 .

Строим множество Λ_n по следующему правилу:

$$\Lambda_n = \{ \Lambda_n(\varepsilon_1^{n-n_1, l_1}, \dots, \varepsilon_{n_1}^{n-n_1, l_{n_1}} x_{n-n_1+j}^{k_j}, \dots, \varepsilon_m^{n-n_1, l_m}) \}$$

$$\begin{aligned}
& + \sum_{j=1}^{n_1} a_{m(n-n_1+j)} x_{n-n_1+j}^{k_j} = \Lambda_n(\varepsilon_1^{n_1, j_1}, \dots, \varepsilon_m^{n_1, j_m}) = \\
& = \sum_{i=1}^{n-n_1} f_i(x_i^{j_i}) + \sum_{j=n-n_1+1}^n f_j(x_j^{k_j}); \quad \forall (\varepsilon_1^{n-n_1, i_1}, \dots, \varepsilon_m^{n-n_1, i_m}) \in \rho^{n-n_1}, \\
& \forall (x_{n-n_1+1}^{k_1}, \dots, x_n^{k_n}) \in \Lambda^1, \\
& (x_1^{j_1}, \dots, x_{n-n_1}^{j_{n-n_1}}, x_{n-n_1+1}^{k_1}, \dots, x_n^{k_n}) \}. \quad (2.82)
\end{aligned}$$

Если одной и той же последовательности значений аргументов $\varepsilon_1^n, \dots, \varepsilon_m^n$ приписаны различные цепочки значений переменных x_1, \dots, x_n , то остается одна, на которой функция $\Lambda_n(\cdot)$ достигает максимума.

Оптимальным решением задачи (2.78), (2.79) является та цепочка значений аргументов x_1, \dots, x_n приписанная элементу из ρ^n , удовлетворяющему ограничению (2.79), на которой функция $\Lambda_n(\cdot)$ достигает максимального значения.

Покажем, что предложенный алгоритм решения задачи (2.78), (2.79) является полиномиальным относительно n -числа переменных задачи. Для этого необходимо показать, что количество элементов в множествах $\rho_i, i = \overline{1, n-n_1}$ ограничено полиномом фиксированной степени n .

Очевидно, что

$$|p^{i-1}| \leq |p^i|, \quad i = \overline{2, n-n_1},$$

где $|p^i|$ – мощность множества p^i .

Найдем верхнюю оценку мощности множества p^i . Обозначим:

$$\begin{aligned}
a &= \max a_{ij} \\
1 &\leq i \leq m \\
1 &\leq j \leq n-n_1.
\end{aligned}$$

Тогда мощность множества $|p^{n-n_1}|$ ограничена величиной

$$|p^{n-n_1}| \leq [2ka(n-n_1)]^{T_2} \quad (2.83)$$

Оценка (2.83) получена для крайнего случая, когда все ненулевые элементы первых $n-n_1$ столбцов матрицы A находятся в одних и тех же T_2 строках этой матрицы. Тогда мощность множества

$$|p^n| \leq (2k)^{T_1} [2ka(n-n_1)]^{T_2}. \quad (2.84)$$

Оценка (2.84) является полиномом фиксированной степени n . Она остается полиномом фиксированной степени n и в случае, когда величины k, a ограничены полиномами фиксированной степени n :

$$k \leq O(n^{t_1}), \quad a \leq O(n^{t_2}), \quad t_1, t_2 \text{ от } n \text{ не зависят.}$$

Рассмотрим задачу сепарабельного программирования вида

$$\max \sum_{i=1}^n f_i(x_i); \quad (2.85)$$

$$\sum_{j=1}^n f_{ij}(x_j) \leq b_i, \quad i = \overline{1, m},$$

$$\forall x_i \equiv 0 \pmod{1}, \quad \forall x_i \in D_i, \quad D_i = (x_i^j), \quad j = \overline{1, N_i}, \quad \forall x_i \in D_i | x_i \leq k, \quad (2.86)$$

$$\text{где } \forall |f_{ij}(x_j)| \leq O(n^t), \quad x_j \in D_j,$$

где $O(n^t)$ – полином от n степени t , не зависящего от n ; $\forall f_{ij}(x_j)$ на целых значениях аргументов x_j принимают целочисленные значения. Пусть

$$F = (f_{ij}(x_j)), \quad i = \overline{1, m}, \quad j = \overline{1, n} -$$

матрица, удовлетворяющая условиям, аналогичным условиям, наложенным на матрицу A : последние $n_1 \leq T_1$ столбцов ее могут содержать все ненулевые элементы; $F^{(i)}$ – столбец матрицы F содержит не более $m_i \leq T_2$ ненулевых элементов. Числа T_1, T_2 фиксированы и не зависят от $n, i = \overline{1, n} - n_1$.

Тогда задача (2.85), (2.86) может быть решена полиномиальным относительно n алгоритмом, совпадающим с вышеизложенным, и отличающимся лишь очевидным образом измененными формулами построения множеств Λ_i и функций $\Lambda_i(\cdot)$.

Так, аналог (2.80):

$$\begin{aligned} \Lambda_i &= \{ \Lambda_i(0, \dots, 0 f_{i,1}(x_1^j), 0, \dots, 0 f_{i,m_1}(x_1^j), 0, \dots) = \\ &= \Lambda_i(0, \dots, 0, \varepsilon_{l_1}^{i, l_1}, 0, \dots, \varepsilon_{l_m}^{i, l_m}, 0, \dots, 0) = \\ &= f_i(x_1^j), \quad j = \overline{1, N_1}, \quad (x_1^j) \}, \end{aligned}$$

где $f_{l_1}, \dots, f_{l_{m_1}}$ – ненулевые элементы $F^{(i)}$.

В этом случае в отличие от (2.80) необходимо добавить, что если одной и той же последовательности значений аргументов $\varepsilon_{l_1}^1, \dots, \varepsilon_{l_m}^1$ приписаны различные значения переменной x_1 , то оставляем то значение, на котором $\Lambda_i(\cdot)$ достигает максимума.

Аналог (2.81):

$$\begin{aligned} \Lambda_i &= \{ \Lambda_i(\varepsilon_1^{i-1, l_1}, \dots, \varepsilon_{p_1-1}^{i-1, l_{p_1}}, \varepsilon_{p_1}^{i-1, l_{p_1}} + f_{p_1, i}(x_i^j), \varepsilon_{p_1+1}^{i-1, l_{p_1+1}}, \dots, \\ &\varepsilon_{p_2-1}^{i-1, l_{p_2-1}}, \varepsilon_{p_2}^{i-1, l_{p_2}} + f_{p_2, i}(x_i^j), \dots, \varepsilon_{p_{m_i}-1}^{i-1, l_{p_{m_i}-1}}, \varepsilon_{p_{m_i}}^{i-1, l_{p_{m_i}}} + f_{p_{m_i}, i}(x_i^j), \\ &\varepsilon_{p_{m_i}+1}^{i-1, l_{p_{m_i}+1}}, \dots, \varepsilon_m^{i-1, l_m}) = \Lambda_i(\varepsilon_1^{i, l_1} = \varepsilon_1^{i-1, l_1}, \dots, \varepsilon_{p_1-1}^{i, l_{p_1-1}} = \\ &= \varepsilon_{p_1-1}^{i-1, l_{p_1-1}}, \varepsilon_{p_1}^{i, l_{p_1}} = \varepsilon_{p_1}^{i-1, l_{p_1}}, \varepsilon_{p_1+1}^{i, l_{p_1+1}} = \varepsilon_{p_1+1}^{i-1, l_{p_1+1}}, \dots, \varepsilon_{p_2-1}^{i, l_{p_2-1}} = \varepsilon_{p_2-1}^{i-1, l_{p_2-1}}, \varepsilon_{p_2}^{i, l_{p_2}}, \dots \} \end{aligned}$$

$$\begin{aligned} \varepsilon_{\rho_{m_i-1}}^{i, i, \rho_{m_i-1}} &= \varepsilon_{\rho_{m_i-1}}^{i-1, i, \rho_{m_i-1}}, \varepsilon_{\rho_{m_i}}^{i, j, \rho_{m_i}}, \varepsilon_{\rho_{m_i+1}}^{i, i, \rho_{m_i+1}} = \varepsilon_{\rho_{m_i}}^{i-1, i, \rho_{m_i+1}}, \dots, \varepsilon_m^{i, i, m} = \\ &= \varepsilon_m^{i-1, i, m} = \sum_{j=1}^{i-1} f_j(x_j^{i,j}) + f_i(x_i^j), \quad \forall (\varepsilon_1^{i-1, i_1}, \dots, \varepsilon_m^{i-1, i_m}) \in \rho^{i-1}, \\ & \quad j = \overline{1, N_i}; (x_1^i, \dots, x_{i-1}^{i-1}, x_i^j) \}. \end{aligned}$$

Аналог (2.82):

$$\begin{aligned} \Lambda_n &= \{ \Lambda_n(\varepsilon_1^{n-n_1, i_1} + \sum_{j=1}^{n_1} f_{1(n-n_1+j)} x_{n-n_1+j}^{k_j}), \dots, \varepsilon_m^{n-n_1, i_m} + \\ &+ \sum_{j=1}^{n_1} f_{m(n-n_1+j)} x_{n-n_1+j}^{k_j}) = \Lambda_n(\varepsilon_1^{n, j_1}, \dots, \varepsilon_m^{n, j_m}) = \sum_{i=1}^{n-n_1} f_i(x_i^{j_i}) + \\ &+ \sum_{j=n-n_1+1}^n f_j(x_j^{k_j}), \quad \forall (\varepsilon_1^{n-n_1, i_1}, \dots, \varepsilon_m^{n-n_1, i_m}) \in \rho^{n-n_1}; \\ & \quad \forall (x_{n-n_1+1}^{k_1}, \dots, x_n^{k_n}) \in \Lambda^1; (x_1^j, \dots, x_{n-n_1}^{j_{n-n_1}}, x_{n-n_1+1}^{k_1}, \dots, x_n^{k_n}) \}. \end{aligned}$$

Аналог (2.83):

$$|\rho^{n-n_1}| \leq [2(n-n_1) O(n^t)]^{T_2} \quad (2.87)$$

Аналог (2.84):

$$|\rho^n| \leq (2k)^{T_1} [2(n-n_1) O(n^t)]^{T_2} \quad (2.88)$$

Выражения (2.87), (2.88) являются полиномами фиксированной степени относительно n -числа переменных задачи.

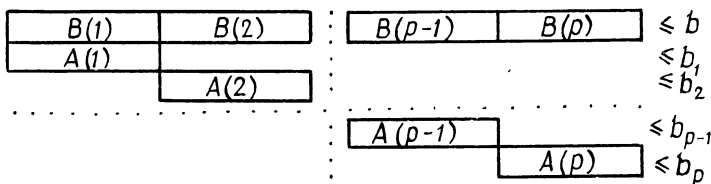
Рассмотренные в пп. 2, 4 задачи сепарабельного целочисленного программирования с фиксированным числом линейных ограничений, частично-блочной структурой, „лестничной“ структурой линейных ограничений могут быть решены алгоритмами с полиномиальной оценкой сложности относительно n – числа переменных задачи для случая сепарабельных ограничений.

Структура с фиксированным числом ограничений

$$\begin{aligned} \max \sum_{i=1}^n f_i(x_i), \\ A \leq B \end{aligned}$$

A – матрица размером $m \times n$ элементами которой являются целочисленные функции $a_{ij}(x_j)$, $i = \overline{1, m}$, $j = \overline{1, n}$, m не зависит от n , $\forall |a_{ij}(x_j)| \leq O(n^t)$, $x_j \in D_j$, t не зависит от n .

Частично-блочная структура:



$$\max \sum_{i=1}^n f_i(x_i);$$

$B(i)$ - матрицы размером $m \times n_i$ элементами которой являются целочисленные функции $B_{lj}^{(i)}(x_{\sum_{k=1}^{i-1} n_{k+j}})$, $l = \overline{1, m_i}$, $j = \overline{1, n_i}$; $A(i)$ - матрица размером $m_i \times n_i$ элементами которой являются:

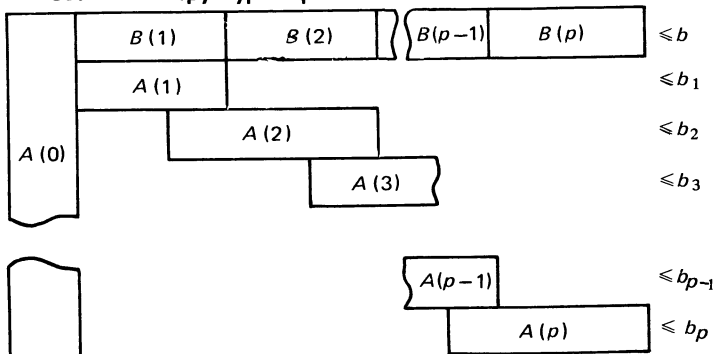
$$a_{lj}^{(i)}(x_{\sum_{k=1}^{i-1} n_{k+j}}), \quad l = \overline{1, m_i}, \quad j = \overline{1, n_i};$$

$$\forall |a_{lj}^{(i)}(x_{\sum_{k=1}^{i-1} n_{k+j}})| \leq O(n^t); \quad x_{\sum_{k=1}^{i-1} n_{k+j}} \in D_{\sum_{k=1}^{i-1} n_{k+j}};$$

$$\forall |b_{lj}^{(i)}(x_{\sum_{k=1}^{i-1} n_{k+j}})| \leq O(n^t); \quad x_{\sum_{k=1}^{i-1} n_{k+j}} \in D_{\sum_{k=1}^{i-1} n_{k+j}},$$

t не зависит от n .

Лестничная структура ограничений



Размеры матриц $A(0)$, $B(i)$, $A(i)$, $i = \overline{1, p}$ совпадают с размерами аналогичных матриц, приведенных в п. 4.

Для сокращения записи элементы матриц $A(0)$, $B(i)$, $A(i)$, $i = \overline{1, p}$ определим следующим образом. Если элементы $a_{ij}(k)$, $k = 0, 1, \dots, p$ либо $b_{ij}(k)$, $k = \overline{1, p}$ матриц $A(k)$, $B(k)$ в ограничениях задачи сепарабельного программирования (п. 4) умножаются на переменную x_l , то

$$a_{ij}(k)x_l \rightarrow a_{ij}^{(k)}(x_l), \quad k = \overline{0, p}, \quad (2.89)$$

$$b_{ij}(k)x_l \rightarrow b_{ij}^{(k)}(x_l), \quad k = \overline{1, p},$$

где целочисленные функции $a_{ij}^{(k)}(x_l)$, $b_{ij}^{(k)}(x_l)$ – элементы матриц $A(k)$, $B(k)$. При этом

$$\begin{aligned} \forall |a_{ij}^{(k)}(x_l)| \leq O(n^t), \quad x_l \in D_l; \\ \forall |b_{ij}^{(k)}(x_l)| \leq O(n^t), \quad x_l \in D_l, \end{aligned} \quad (2.90)$$

t не зависит от n .

Алгоритмы решения сформулированных задач полностью совпадают с алгоритмами решения соответствующих задач, изложенными в пп. 2, 4 с единственным изменением: в формулы для построения множеств Λ_i , функций $\Lambda_i(i)$, $i = \overline{1, n}$ вместо выражений $a_{ij}(k)x_l$, $b_{ij}(k)x_l$ подставляются соответственно выражения $a_{ij}^{(k)}(x_l)$, $b_{ij}^{(k)}(x_l)$.

В силу выполнения условия (2.90), для всех случаев, рассмотренных в пп. 2, 4, соответствующие алгоритмы имеют полиномиальную оценку сложности решения относительно n – числа переменных задачи.

При построении функций $\Lambda_i(\varepsilon_j^i, \dots, \varepsilon_m^i)$ во всех рассмотренных ранее задачах величина диапазона возможных значений аргументов ε_j^i , $j = \overline{1, m}$, $i = \overline{1, n}$ для линейных ограничений не превышает

$$2ka_ji, [-ka_ji, ka_ji] \quad (2.91)$$

Здесь $a_j = \max_{1 \leq l \leq i} |a_{jl}|$, где a_{jl} – элементы матрицы A ограничений задачи (будем считать, что все ранее рассмотренные случаи ограничений – частично-блочные, „лестничные”, с частично заполненной матрицей, являются частными случаями матрицы A).

При достаточно больших n диапазон длины $2ka_jn$ может быть большим, что резко увеличит временные характеристики используемого алгоритма оптимизации. Для сокращения объема вычислений можно воспользоваться следующим приемом.

Пусть j -е ограничение матрицы A

$$\sum_{l=1}^n a_{jl} x_l \leq b_j.$$

Тогда верхнюю границу диапазона возможных значений аргумента ε_j^i можно представить в следующем виде:

$$\varepsilon_j^i \leq b_j + k\bar{a}_j(n-i-1), \quad (2.92)$$

где $\bar{a}_j = \max_{i+1 \leq l \leq n} |a_{jl}|$, совмещение границ диапазона (2.91), (2.92) уменьшит процедуру перебора допустимых решений.

Пусть $\sum_{l=1}^n a_{jl} x_l = b_j$.

В этом случае аналог (2.92):

$$b_j - k\bar{a}_j(n-i) \leq \varepsilon_j^i \leq b_j + k\bar{a}_j(n-i). \quad (2.93)$$

Если элементами матрицы A являются $a_{ij}(x_j)$,

$$|a_{jl}(x_l)| \leq k_{jl}, \quad x_l \in D_l,$$

то аналог формулы (2.91):

$$\left[-\sum_{l=1}^i k_{jl}, \sum_{l=1}^i k_{jl} \right];$$

аналог (2.92):

$$\varepsilon_j^i \leq b_j + \sum_{l=i+1}^n k_{jl};$$

аналог (2.93):

$$b_j - \sum_{l=i+1}^n k_{jl} \leq \varepsilon_j^i \leq b_j + \sum_{l=i+1}^n k_{jl}.$$

Очевидно, что все приведенные в этом параграфе алгоритмы требуют память, величина которой ограничена полиномом фиксированной степени от числа переменных.

Пример

Решение контрольного примера:

$$\max f = \frac{1}{5}x_1^2 - x_2 + x_3 + \frac{1}{2}x_4^2 - 2x_5^3 + 5x_6 - 3x_7^3 + 4x_8 - x_9 + 2x_{10}^2 - \\ - 2x_{11} - x_{12} + \frac{1}{4}x_{13}^2 - \frac{1}{2}x_{14}^2,$$

$$\begin{aligned} -x_3 + x_4 + x_5 + x_7 + x_9 + x_{13} &\leq 6, \\ 2x_1 + 4x_2 + 6x_3 + 6x_4 - 2x_5 + 3x_7 - 2x_{11} + x_{14} &\leq 7, \\ 3x_1 + x_2 - x_6 + 2x_8 + 3x_{14} &\leq 5, \\ 2x_2 + 5x_4 + 2x_6 + 3x_{11} + 5x_{12} + 4x_{13} &\leq 3, \\ 2x_1 + 2x_3 + 2x_4 - 4x_8 - 2x_{10} &\leq -5, \\ 3x_3 + 6x_5 + 2x_{10} - 4x_{12} &\leq 6. \end{aligned}$$

Значение переменных:

$$\begin{aligned} x_1 &= 0; 5, & x_8 &= 0; 1; 2, \\ x_2 &= 0; 2; 4, & x_9 &= 0; 2; 3, \\ x_3 &= 0; 1; 2, & x_{10} &= 0; 5, \\ x_4 &= 0; 2, & x_{11} &= 0; 7, \\ x_5 &= 0; 1, & x_{12} &= 0; 3, \\ x_6 &= -1; 0; 1; 2, & x_{13} &= 0; 4, \\ x_7 &= -1; 0, & x_{14} &= 0; 2. \end{aligned}$$

Матрица ограничений:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
ε_1	0	0	-1	1	1	0	1	0	1	0	0	0	3	0
ε_2	2	4	6	6	-2	0	3	0	0	0	-2	0	0	1
ε_3	3	1	0	0	0	-1	0	2	0	0	0	0	0	3
ε_4	0	2	0	5	0	2	0	0	0	0	3	5	4	0
ε_5	2	0	2	2	0	0	0	-4	0	2	0	0	0	0
ε_6	0	0	3	0	6	0	0	0	0	2	0	-4	0	0

Решение приведенного примера потребовало построения 271 цепочки. Оптимальное решение:

$$x_1=0; x_2=0; x_3=1; x_4=0; x_5=1; x_6=1; x_7=1; x_8=2; x_9=0; \\ x_{10}=0; x_{11}=0; x_{12}=0; x_{13}=0; x_{14}=0.$$

6. АЛГОРИТМ РЕШЕНИЯ ОДНОГО КЛАССА ЗАДАЧ НЕЛИНЕЙНОГО ЦЕЛОЧИСЛЕННОГО ПРОГРАММИРОВАНИЯ С ЧАСТИЧНО-БЛОЧНОЙ СТРУКТУРОЙ ОГРАНИЧЕНИЙ, ИМЕЮЩЕЙ ПОЛИНОМИАЛЬНУЮ ОЦЕНКУ СЛОЖНОСТИ ОТНОСИТЕЛЬНО ЧИСЛА ПЕРЕМЕННЫХ

Рассмотрим следующую задачу нелинейного целочисленного программирования:

$$\max f(x_1, \dots, x_n); \quad (2.94)$$

$$F(x_1, \dots, x_n) \leq b,$$

$$\forall x_i \in D_i; |x_i| \leq k, k \geq 0, \forall x_i \equiv 0 \pmod{1}, \quad (2.95)$$

где

$$D_i = \{x_i^j\}, j = \overline{1, N_i}$$

$$F(x_1, \dots, x_n) = \begin{pmatrix} F_1(x_1, \dots, x_n) \\ \vdots \\ F_{\bar{m}}(x_1, \dots, x_n) \end{pmatrix}, b = \begin{pmatrix} b_1 \\ \vdots \\ b_{\bar{m}_p} \end{pmatrix}$$

$$\bar{m}_p = m + \sum_{i=1}^p m_i; b^T = (\bar{b}_m, \bar{b}_{m_1}, \dots, b_{m_p}),$$

$$\text{где } \bar{b}^T = (b_1, \dots, b_m), \bar{b}_{m_i}^T = (b_{\bar{m}_{i-1}+1}, \dots, b_{m_i}).$$

Функции $F_i(x_1, \dots, x_n)$, $i = \overline{1, m}$ целочисленные, т. е. для целочисленных значений своих аргументов принимают целочисленные значения.

Функция $f(x_1, \dots, x_n)$ задана в виде

$$f(x_1, \dots, x_n) = \sum_{i=1}^p f_i(x_{\bar{n}_{i-1}+1}, \dots, x_{\bar{n}_i}) = \sum_{i=1}^p f_i(\bar{x}_{n_i}), \quad (2.96)$$

$$\text{где } \bar{n}_i = \sum_{j=1}^i n_j, i = \overline{2, p}; \bar{n}_p = n,$$

$$(x_{\bar{n}_{i-1}+1}, \dots, x_{\bar{n}_i})^T = x_{n_i}^T, i = \overline{2, p}, \bar{x}_{n_i} = (x_1, \dots, x_{n_i}).$$

Матрица ограничений (2.95) имеет структуру:

$$\left(\begin{array}{c} m_1 \{ \begin{array}{|c|c|} \hline B_1(x_{n_1}) & B_2(x_{n_2}) \\ \hline \end{array} \} \quad \left. \begin{array}{l} \\ \\ \\ \end{array} \right\} \begin{array}{l} B_p(x_{n_p}) \\ \\ \\ \end{array} \right) \begin{array}{l} \underline{\underline{b}} \\ \underline{\underline{b_{m_1}}} \\ \underline{\underline{b_{m_2}}} \\ \underline{\underline{b_{m_p}}} \end{array} \quad (2.97)$$

$m \leq T, \forall n_i \leq T$, T не зависит от n , $m_i, i = \overline{1, p}$ в общем случае могут быть ограничены полиномами фиксированной степени n ($m_i \leq O(n^{t_i}); O(n^{t_i})$ – полиномы от n фиксированной степени t_i ;

$$B_i(\bar{x}_{n_i}) = \{b_{ij}(\bar{x}_{n_i})\}, \quad j = \overline{1, m}; \quad i = \overline{1, p};$$

$$A_i(\bar{x}_{n_i}) = \{a_{ij}(\bar{x}_{n_i})\}, \quad j = \overline{1, m_i}, \quad i = \overline{1, p}.$$

Целочисленные функции $b_{ij}(\bar{x}_{n_i}), i = \overline{1, p}, j = \overline{1, m}$ для всех $x_i \in D_i, i = \overline{1, n}$ удовлетворяют ограничениям:

$$|b_{ij}(\bar{x}_{n_i})| \leq O(n_i^{t_i}) \quad (2.98)$$

либо

$$|b_{ij}(\bar{x}_{n_i})| \leq O(n^{t_i}), \quad i = \overline{1, p}, \quad j = \overline{1, m}; \quad (2.99)$$

$O(n^{t_i})$ полином от n фиксированной степени t_i .

Покажем, что для таким образом сформулированной задачи нелинейного целочисленного программирования может быть предложен алгоритм с полиномиальной оценкой сложности относительно n – числа переменных задачи.

Образует множество Ω_i^1 всех различных последовательностей значений переменных вектора $\bar{x}_{n_i}, i = \overline{1, p}$. При этом $|\Omega_i^1| \leq (2k)^{n_i}$. Формируем множество

$$\Lambda_{n_i} = \{ \Lambda_{n_i}(\varepsilon_1^{n_i, i_1} = b_{i_1}(\bar{x}_{n_i}^k), \dots, \varepsilon_m^{n_i, i_m} = b_{i_m}(\bar{x}_{n_i}^k) = f(\bar{x}_{n_i}^k), \forall \bar{x}_{n_i}^k \in \Omega_i^1; (\bar{x}_{n_i}^k) \} \quad (2.100)$$

Множество Ω_i строится по множеству Ω_i^1 путем исключения из него всех элементов, не удовлетворяющих ограничениям

$$A_i(\bar{x}_{n_i}^k) \leq \bar{b}_{m_i}.$$

Образует ρ^1 – множество различных последовательностей значений аргументов $\varepsilon_1^{n_i}, \dots, \varepsilon_m^{n_i}$. Если одному и тому же элементу из множества ρ^1 приписано более чем одно значение вектора \bar{x}_{n_i} , то оставляется одно, на котором функция $\Lambda_{n_i}(\cdot)$ достигает максимума. Множества $\Lambda_{\bar{n}_i}, i = \overline{2, p}$ строятся аналогичным образом:

$$\begin{aligned} \Lambda_{\bar{n}_i} &= \{ \Lambda_{\bar{n}_i}(\varepsilon_1^{\bar{n}_i-1, i_1} + b_{i_1}(\bar{x}_{n_i}^k), \dots, \varepsilon_m^{\bar{n}_i-1, i_m} + b_{i_m}(\bar{x}_{n_i}^k) = \\ &= \Lambda_{\bar{n}_i}(\varepsilon_1^{\bar{n}_i, j_1}, \dots, \varepsilon_m^{\bar{n}_i, j_m}) = \sum_{l=1}^{j_1} f_l(\bar{x}_{n_i}^{j_l}) + f_i(\bar{x}_{n_i}^k), \\ &\forall \varepsilon_1^{\bar{n}_i-1}, \dots, \varepsilon_m^{\bar{n}_i-1} \in \rho^{i-1}, \\ &\forall \bar{x}_{n_i}^k \in \Omega_i, (\bar{x}_{n_i}^{j_1}, \dots, \bar{x}_{n_i}^{j_{i-1}}, \bar{x}_{n_i}^k) \}. \end{aligned} \quad (2.101)$$

Множество Ω_i строится по множеству Ω_i^1 путем исключения из него всех элементов, не удовлетворяющих ограничениям

$$A_i(\bar{x}_{n_i}^k) \leq \bar{b}_{m_i}.$$

Образует p^i – множество различных последовательностей значений аргументов $\varepsilon_1^{n_i}, \dots, \varepsilon_m^{n_i}$. Если одному и тому же элементу из множества p^i приписано более чем одно значение вектора (x_1, \dots, x_{n_i}) то остается одно, на котором функция $\Lambda_{\bar{n}_i}(\cdot)$ достигает максимума.

Оптимальное решение задачи (2.94), (2.95) определяется по множеству $\Lambda_{\bar{n}_p}$; \bar{x}_{opt} элемент p^p , удовлетворяющий ограничениям

$$\sum_{j=1}^p B_j(\bar{x}_{n_j}) \leq \bar{b}_m$$

на котором функция $\Lambda_{\bar{n}_p}(\cdot)$ достигает максимума. Доказательство строгой оптимальности изложенного алгоритма является очевидным. Оценим сложность предложенного алгоритма.

Сложность алгоритма построения множеств \mathcal{Q}_i по множествам \mathcal{Q}_i^k для k , не зависящего от n , ограничено полиномом степени t_i от n . Пусть реализуется оценка (2.98). Тогда верхняя оценка числа элементов множества p^i , $i = \overline{1, p}$

$$|p^i| \leq [2 \sum_{j=1}^i O(n_j^{t_j})]^m, \quad i = \overline{1, p}. \quad (2.102)$$

Если справедлива оценка (2.99), то для всех $i = \overline{1, p}$

$$|p^i| \leq [2p \cdot O(n^{t_i})]^m \quad (2.103)$$

так как величина p ограничена полиномом первой степени от n , то оценка (2.102), (2.103) – это полиномы фиксированной степени от n и, следовательно, предложенный алгоритм решения задачи (2.94), (2.95) имеет полиномиальную оценку сложности относительно числа переменных задачи.

Необходимо отметить, что оценка (2.99) предусматривает возможность ограничения величины k полиномом фиксированной степени от n . Однако и в этом случае сложность алгоритма останется полиномиальной, так как количество элементов \mathcal{Q}_i^k , $i = \overline{1, p}$ будет ограничено полиномом фиксированной степени от n .

Уменьшить реальное число вычислений можно путем следующей модификации алгоритма. При построении функций $\Lambda_{\bar{n}_i}(\varepsilon_1^{n_i}, \dots, \varepsilon_m^{n_i})$ величина диапазона возможных значений аргументов $\varepsilon_j^{n_i}$, $j = \overline{1, m}$, $i = \overline{1, p}$ не превышает

$$[-\sum_{t=1}^i O(n_t^{t_i}); \sum_{t=1}^i O(n_t^{t_i})] \quad (2.104)$$

в случае оценки (2.90) и

$$[-i O(n^{t_i}), i O(n^{t_i})] \quad (2.105)$$

в случае оценки (2.99).

Пусть j - е ограничение задачи (2.94), (2.95) $j = \overline{1, m}$ имеет вид

$$\sum_{i=1}^p b_{ji}(\bar{x}_{n_i}) \leq b_j.$$

Тогда верхняя граница диапазона возможных значений аргумента

$$\varepsilon_j^{\bar{n}_i} \leq b_j + \sum_{t=0}^{p-i-1} O(n_{p-t}^t), \quad i = \overline{1, p}, \quad (2.106)$$

если реализуется оценка (2.98), и

$$\varepsilon_j^{\bar{n}_i} \leq b_j + (p-i-1) O(n^t), \quad i = \overline{1, p}, \quad (2.107)$$

если реализуется оценка (2.99).

Пусть j - е ограничение задачи (2.94), (2.95) $j = \overline{1, m}$ имеет вид

$$\sum_{i=1}^p b_{ji}(\bar{x}_{n_i}) = b_j.$$

Тогда для оценки (2.98)

$$b_j - \sum_{t=0}^{p-i-1} O(n_{p-t}^t) \leq \varepsilon_j^{\bar{n}_i} \leq b_j + \sum_{t=0}^{p-i-1} O(n_{p-t}^t), \quad i = \overline{1, p} \quad (2.108)$$

для оценки (2.99)

$$b_j - (p-i-1) O(n^t) \leq \varepsilon_j^{\bar{n}_i} \leq b_j + (p-i-1) O(n^t) \quad (2.109)$$

Совместное использование оценок (2.104)–(2.109) при построении множеств $\Lambda_{\bar{n}_i}$ существенно уменьшает необходимое число вычислений.

7. ИСПОЛЬЗОВАНИЕ АЛГОРИТМА ПОСЛЕДОВАТЕЛЬНОГО АНАЛИЗА И ОТСЕВА ВАРИАНТОВ БЕЗ ПОШАГОВОГО КОНСТРУИРОВАНИЯ РЕШЕНИЯ

Рассмотренная в пп. 2–5 схема решения различного класса задач целочисленного программирования приводит к полиномиальным относительно числа переменных задачи алгоритмам. Однако эта схема может привести к необходимости выполнения вычислений, с трудом реализуемых на современных ЭВМ.

Существенное изменение реального объема вычислений (без изменения верхней теоретической их оценки) может быть достигнута за счет использования на начальном этапе решения задачи алгоритмической схемы метода анализа и отсева вариантов без пошагового конструирования решения [7].

Суть вычислительной схемы заключается в следующем [7]:

пусть задана задача дискретной оптимизации

$$\bar{v} = \arg \min f(v), \quad v \in D(v) \quad (2.110)$$

где $D(\vartheta)$ задается так:

$$g_p(\vartheta) \leq g_p^*, \quad p=1, Q, \quad \vartheta \in V = \prod_{j=1}^n U_j,$$

$$U_j = \{U_{j(t_1)}, \dots, U_{j(t_j)}, \dots, U_{j(\bar{z}_j)}\}, \quad j=1, \bar{n}, \quad |U_j| = \bar{z}_j < \infty \quad (2.111)$$

$|V| = N = \prod_{i=1}^n \bar{z}_i$, f, g_p - произвольная функция дискретного аргумента.

Процедура W последовательного отсева значений переменных $U_j(t_j) \in U_j$ ($j = \bar{1}, \bar{n}$) состоит из „элементарных” процедур W^p , $p = \bar{1}, \bar{Q}$ каждая из которых заключается в отсеве по p -у ограничению элементов $U_j = U_j(t_j)$, $t_j \in J_j = \{1, \bar{z}_j\}$, $j = \bar{1}, \bar{n}$. Удовлетворяющих неравенству

$$\vartheta^{p(t_j)} = (U_1^{p(t_1)}, \dots, U_{j-1}^{p(t_{j-1})}, U_j(t_j), U_{j+1}^{p(t_{j+1})}, \dots), \\ U_n^{p(t_n)} \equiv \vartheta^{p(t_j)} \setminus V_j, \quad U_j(t_j) = \arg \min_{V \in (V \setminus V_j) \cup \{U_j(t_j)\}} g_p(\vartheta) > g_p^* \quad (2.112)$$

Вариант $\vartheta^{p(t_j)}$ обеспечивает оптимум функции $g_p(\vartheta)$ при фиксированном элементе $U_j(t_j)$.

Обозначим множество элементов V_j , оставшихся после применения процедуры W^p , через U_j^1 . Тогда после применения процедуры W получим „усеченное” множество возможных вариантов

$$V^{(1)} = \prod_{p=1}^Q \prod_{j=1}^n U_j^p = \prod_{j=1}^n U_j^{(1)},$$

где $U_j^{(1)} = \prod_{p=1}^Q U_j^p$ - множество элементов из V_j , оставшихся после применения процедуры W . Таким образом, $V \supseteq V^{(1)}$. Далее к множеству $V^{(1)}$ снова применим процедуру W , получаем $V^{(2)}$ и т. д.

При последовательном применении к множеству V процедуры W получим цепочку вложенных множеств

$$V = V^0 \supseteq V^{(1)} \supseteq V^{(2)} \supseteq \dots \supseteq V^{(r)} \supseteq V^{(r+1)}.$$

Обоснование неоднократного применения процедуры W заключается в том, что неотсеивающийся на некоторой итерации применения процедуры W элемент $U_j(t_j)$ может отсеяться на последующих итерациях, поскольку на данной итерации могут отсеяться элементы, входящие в вектор $(\vartheta^{p(t_j)} | V_j)$ и на следующей итерации условия (2.112) выполняются.

В [7] приведены следующие утверждения:

условие $V^{(r)} = V^{(r+1)}$ является условием окончания применения процедуры W ;

число применений процедуры W к задаче (2.110) конечно и ограничено величиной

$$\sum_{j=1}^n \bar{z}_j.$$

Для задач линейного целочисленного программирования, а также сепарабельного целочисленного программирования всех классов, приведенных в пп. 1–5, каждая процедура W является полиномиальным алгоритмом относительно n – количества переменных задачи. Следовательно, применение процедуры W к задачам целочисленного линейного либо сепарабельного программирования рассмотренных классов, а затем решение этих задач алгоритмами, приведенных в пп. 2–5, требует выполнения общего количества операций, ограниченного полиномом фиксированной степени от n – числа переменных задачи. Практическая целесообразность применения процедуры W очевидна.

Глава 3. ЦЕЛОЧИСЛЕННЫЕ МОДЕЛИ В СЛОЖНЫХ СИСТЕМАХ УПРАВЛЕНИЯ

1. АНАЛИЗ ЭФФЕКТИВНОСТИ РЕШЕНИЯ NP -ПОЛНЫХ ЗАДАЧ

Большая часть дискретных математических моделей, построенных для анализа, синтеза и функционирования сложных систем, являются либо NP -полными, либо не менее простыми, чем класс NP -полных задач. В частности, рассмотренные задачи целочисленного линейного программирования и целочисленного сепарабельного программирования принадлежат этому классу.

В данном параграфе на основании краткого обзора проблем, связанных с решением NP -полных задач, сделана попытка определить возможные направления исследований по повышению эффективности их решения и с этих позиций проанализировать результаты, приведенные в гл. 1 и 2.

Анализ трудностей, возникающих при вычислениях, на пути создания эффективных методов решения дискретных задач, привел к постановке следующей проблемы: можно ли исключить перебор при решении дискретных задач, или, иными словами, существует ли принципиальная возможность найти оптимальное решение, не перебирая всех или почти всех вариантов в задаче? Эта проблема исследуется в теории NP -полных задач, сформировавшейся на основе работ С. Куна, Р. Карпа, Л. Левина и др. Главными объектами теории являются класс NP всех переборных задач и класс P переборных задач, разрешаемых за полиномиальное время на машине Тьюринга.

Краткий обзор теории NP -полноты приводится в соответствии с [5] для задач распознавания свойств и обобщается затем на оптимизационные задачи. Ключевыми моментами в теории NP -полноты являются:

1. *Задача распознавания Π* . Она состоит из двух множеств: множества D_{Π} всех возможных индивидуальных задач и множества Y_{Π} , $Y_{\Pi} \in D_{\Pi}$ индивидуальных задач с ответом „Да“.

2. *Функция Length: $D \rightarrow R^+$* . Эта функция связана с каждой задачей распознавания, она не зависит от конкретной схемы кодирования и полиномиально эквивалентна длине кода индивидуальной задачи, получаемой при любой разумной схеме кодирования.

Например, в задаче распознавания

$$A_x = b, \quad x \geq C,$$

где нужно определить, существует ли X , для которого $cX > a$, функция Length может иметь вид

$$L = m \cdot n + \lceil \log |p| \rceil$$

Здесь m и n – размеры матрицы A , p – произведение всех ненулевых коэффициентов, определяющих ограничения задачи распознавания.

3. Детерминированные машины Тьюринга и класс P. Детерминированная одноленточная машина Тьюринга состоит из управляющего устройства с конечным числом состояний, читающей (пишущей) головки, которая может считывать и записывать символы на неограниченной в обе стороны ленте, разделенной на бесконечное число одинаковых ячеек, занумерованных целыми числами $\dots, -2, -1, 0, 1, 2, \dots$. Задача распознавания переводится в детерминированную программу для машины Тьюринга, которая определяется следующими компонентами: конечным множеством Γ символов, записываемых на ленте; подмножеством $\Sigma \subset \Gamma$ входных символов и выделенным пустым символом $b \in \Gamma / \Sigma$; конечным множеством состояний Q , в котором выделено начальное состояние q_0 и два заключительных состояния q_Y – „Да”, q_N – „Нет”; функцией перехода δ :

$$(Q \setminus \{q_Y, q_N\}) \times \Gamma \rightarrow Q \times \Gamma \times \{-1, 1\}.$$

Входом для детерминированной программы является слово $X \in \Sigma^*$ где Σ^* – множество всех слов, каждое из которых соответствует индивидуальной задаче распознавания. Слово записывается на ленте в ячейках с номерами $1, 2, \dots, |X|$ по одному символу в ячейке. Все другие ячейки в начальный момент времени содержат пустой символ. Программа начинает работу, находясь в состоянии q_0 , при этом головка находится над ячейкой с номером 0.

Процесс вычислений осуществляется последовательно шаг за шагом. Если текущее состояние q есть q_Y или q_N , то процесс вычислений заканчивается, при этом результат вычислений „Да”, если $q = q_Y$ и „Нет”, если $q = q_N$. В противном случае текущее состояние принадлежит множеству $Q \setminus \{q_Y, q_N\}$. При этом головка читает на ленте некоторый символ $S \in \Gamma$ и определено значение $\delta(q, S)$. Пусть $\delta(q, S) = (q', S', \Delta)$. Тогда головка стирает S , пишет на этом месте S' и сдвигается на одну ячейку влево при $\Delta = -1$ или вправо при $\Delta = +1$. Одновременно управляющее устройство переходит из состояния q в q' .

Детерминированная программа M называется полиномиальной, если существует такой полином P , что для всех n , $n = 1, 2, \dots, T_M(n) \leq P(n)$, где $T_M(n) = \max \{m : \text{существует такое слово } X \in \Sigma^*, |X| = n, \text{ что вычисление по программе } M \text{ на входе } X \text{ требует времени } m\}$.

Задача распознавания Π принадлежит классу P , если существует полиномиальная программа, которая решает эту задачу.

4. Недетерминированные машины Тьюринга и класс NP . Введем понятие недетерминированного алгоритма. Такой алгоритм состоит из двух различных стадий – угадывания и проверки. По заданной индивидуальной задаче распознавания J на первой стадии происходит „угадывание” некоторой структуры S . Затем J и S вместе подаются в качестве входа на стадию проверки, которая выполняется обычным детерминированным образом и заканчивается либо ответом „Да”, либо ответом „Нет”. Недетерминированный алгоритм реализуется недетерминированной одноленточной машиной Тьюринга, представляющей собой модификацию одноленточной детерминированной машины Тьюринга путем включения в ее состав угадывающего модуля.

Считают, что недетерминированный алгоритм, решающий задачу распознавания Π , работает в течение „полиномиального времени”, если найдется такой полином P , что для любой индивидуальной задачи распознавания (имеющий ответ „Да”) найдется некоторая догадка S , приводящая на стадии детерминированной проверки на входе (J, S) к ответу „Да” за время $p(\text{Length}[J])$. Отсюда следует, что „размер” угадываемой структуры S обязательно ограничен полиномом от $\text{Length}[J]$.

Класс NP – это класс всех задач распознавания Π , которые при разумном кодировании могут быть решены недетерминированными алгоритмами за полиномиальное время.

5. Зависимость между классами P и NP . Вопрос о зависимости классов P и NP имеет фундаментальное значение для теории NP -полных задач. Очевидно, $P \subseteq NP$. В настоящее время существует гипотеза о том, что это включение является строгим, т. е. $P \neq NP$. Строго эта гипотеза не доказана. Не доказано также и обратное. Не известны общие методы превращения недетерминированных полиномиальных алгоритмов в детерминированные. Поэтому целью теории NP -полных задач является получение результатов вида: „если $P \neq NP$, то ...”.

Один из наиболее мощных результатов, связывающий детерминированный и недетерминированные алгоритмы, заключается в следующем [5]: если $P \subseteq NP$ то существует такой полином P , что Π может быть решена детерминированным алгоритмом с временной сложностью $O(2^{P(n)})$.

6. Полиномиальная сводимость и NP -полные задачи. Введем понятие языка L_M детерминированной программы M следующим образом: $L_M = \{X \in \Sigma^* : M \text{ останавливается в состоянии } q_Y\}$.

Говорят, что имеет место полиномиальная сводимость языка $L_1 \subseteq \Sigma_1^*$ к языку $L_2 \subseteq \Sigma_2^*$, если существует функция $f : \Sigma_1^* \rightarrow \Sigma_2^*$, удовлетворяющая двум условиям:

1) существует детерминированная программа для одноленточной машины Тьюринга, вычисляющая f с временной сложностью, ограниченной полиномом;

2) для любого $X \in \Sigma_1^*$, $X \in L_1$, тогда и только тогда, когда $f(X) \in L_2$.

Полиномиальная сводимость языка L_1 и L_2 обозначается символом $L_1 \approx L_2$.

Очевидно, если $L_1 \approx L_2$, то из $L_2 \in P$ следует, что $L_1 \in P$. (Эквивалентное утверждение: из $L_1 \in P$ следует, что $L_2 \in P$). Язык L называется NP -полным, если $L \in NP$ и любой другой язык $L' \in NP$ сводится к L . Иными словами, задача распознавания P называется NP -полной, если любая другая задача распознавания $P' \in NP$ сводится к P .

Таким образом, с NP -полными задачами отождествляются самые трудные задачи из NP . Отсюда, в частности, следует вывод: если хотя бы одна NP -полная задача может быть решена за полиномиальное время, то и все задачи из NP также могут быть решены за полиномиальное время. Интересен также вывод, сделанный из предположения, что $P \neq NP$ [5]: существуют задачи из NP , не разрешимые за полиномиальное время и не являющиеся NP -полными. Справедливо следующее утверждение: если L_1 и L_2 принадлежат классу NP , а L_1 — NP -полный язык (соответствующая этому языку задача распознавания NP -полная) и $L_1 \approx L_2$, то L_2 также NP -полный язык. С помощью этого утверждения легко доказать NP -полноту новой задачи P . Если известна хотя бы одна NP -полная задача, то для доказательства NP -полноты новой задачи достаточно показать, что $P \in NP$ и какая-то одна известная NP -полная задача P' сводится к P . Первой NP -полной задачей стала задача распознавания из булевой логики, которую обычно называют „Выполнимость“. С. Кун показал, что эта задача принадлежит NP . Получив общее представление недетерминированной программы машины Тьюринга, конкретизация которого позволяет найти произвольную задачу, принадлежащую NP , он показал, что эта задача полиномиально сводится к задаче „Выполнимость“. В настоящее время существует очень много задач, для которых доказана NP -полнота.

7. Применение теории NP -полноты для анализа задач. Предположим, что установлена NP -полнота некоторой задачи P . Тогда необходимо определить, существуют ли подзадачи $P' \in P$, разрешаемые за полиномиальное время, что позволит сузить класс задач, точное решение которых требует экспоненциального времени.

Введем функцию $\max [J]$, $J \in P$, которая ставит в соответствие любой задаче J целое число $\max [J]$, соответствующее величине максимального по модулю числа в J . Алгоритм ре-

шения задачи распознавания Π называется псевдополиномиальным по времени алгоритмом, или просто псевдополиномиальным алгоритмом, если его временная функция ограничена сверху полиномом от двух аргументов: $Length [J]$ и $max [J]$, $\forall J \in \Pi$.

Назовем задачу Π задачей с числовыми параметрами, если не существует такого полинома P , что $max[J] \leq p(Length[J])$, $\forall J \in \Pi$.

Пусть задача Π разрешима псевдополиномиальным алгоритмом. Выделим класс задач $\Pi' \subset \Pi$ для которого выполняется условие:

$$\forall J \in \Pi', \max [J] \leq p(Length [J]).$$

Тогда любая задача $J \in \Pi'$ решается полиномиальным алгоритмом. Таким образом, хотя задача Π является NP -полной, но трудности, возникающие при ее решении, связаны с тем, что в формулировку задачи могут входить довольно большие числа. Ограничение по модулю величин этих чисел приводит к полиномиальным алгоритмам их решения. Например, задача „о целочисленном ранце“ методом динамического программирования решается с псевдополиномиальной временной сложностью. Однако ограничение по модулю коэффициентов, входящих в функционал либо в единственное ограничение, превращает метод динамического программирования в полиномиальный алгоритм. С этим, в частности, связана высокая практическая эффективность решения задач „о булевом ранце“ с десятками тысяч переменных.

Для произвольной задачи распознавания Π и полинома p (с целыми коэффициентами) обозначим через Π_p подзадачу, получаемую из Π рассмотрением только тех индивидуальных задач J , для которых выполнено соотношение

$$\max [J] \leq p(Length [J]).$$

Задача Π называется задачей NP -полной в сильном смысле, если Π принадлежит NP и существует такой полином p с целыми коэффициентами, что задача Π_p является NP -полной. В частности, если задача $\Pi - NP$ -полная и не является задачей с числовыми параметрами, то задача Π автоматически NP -полная в сильном смысле.

Отсюда вывод: если задача $\Pi - NP$ -полная в сильном смысле, то она не может быть решена псевдополиномиальным алгоритмом при $P \neq NP$.

Действительно, в этом случае задача Π_p не решается полиномиальным алгоритмом, а это возможно только тогда, когда для задачи Π не существует псевдополиномиального алгоритма. В [5] показано, что имеются задачи с числовыми параметрами NP -полные в сильном смысле.

8. Подходы к решению NP -полных задач. Комбинаторные оптимизационные задачи. Дадим формальное определение понятия „оптимизационная задача“ [5]. Комбинаторная оптимизационная задача Π есть либо задача минимизации, либо задача максимизации и состоит из следующих трех частей: множества D_{Π} индивидуальных задач; конечного множества $S_{\Pi}(J)$ допустимых решений индивидуальной задачи J для каждой $J \in D_{\Pi}$; функции m_{Π} , сопоставляющей каждой индивидуальной задаче $J \in D_{\Pi}$ и каждому допустимому решению $\phi \in S_{\Pi}(J)$ некоторое целое число $m_{\Pi}(J, \phi)$, называемое величиной решения ϕ . Если Π – задача минимизации (соответственно максимизации), то оптимальным решением индивидуальной задачи $J \in D_{\Pi}$ является такое допустимое решение $\phi^* \in S_{\Pi}(J)$, что для всех $\phi \in S_{\Pi}(J)$ выполнено неравенство $m_{\Pi}(J, \phi^*) \leq m_{\Pi}(J, \phi)$ (соответственно $m_{\Pi}(J, \phi^*) \geq m_{\Pi}(J, \phi)$). Для обозначения оптимального решения задачи J используют символ $OPT_{\Pi}(J)$ или $OPT(J)$. Алгоритм A называется приближенным алгоритмом решения задачи Π , если для любой индивидуальной задачи $J \in D_{\Pi}$ с помощью алгоритма A находят некоторое допустимое решение $\phi \in S_{\Pi}(J)$. Через $A(J)$ обозначим величину $m_{\Pi}(J, \phi)$ того возможного решения ϕ , которое алгоритм A строит по J . Если $A(J) = OPT(J), \forall J \in D_{\Pi}$, то A называется точным алгоритмом решения задачи Π .

Для комбинаторных задач, в частности оптимизационных комбинаторных задач, вводится понятие NP -трудных задач, неформальное определение которых является следующим: оптимизационная комбинаторная задача называется NP -трудной, если соответствующая ей задача распознавания является NP -полной. Это означает, что решение оптимизационной комбинаторной задачи не проще, чем решение соответствующей ей задачи распознавания. Формально определение NP -трудной задачи в [5] вводится с помощью понятий словарного отношения, полиномиальной сводимости по Тьюрингу, и оракульной машины Тьюринга. Утверждение о том, что решение оптимизационной комбинаторной задачи не проще, чем решение соответствующей ей задачи распознавания, основывается на очевидной возможности решения задачи распознавания с помощью соответствующей ей оптимизационной комбинаторной задачи. Если задача распознавания принадлежит классу P , то отсюда вытекает полиномиальная разрешимость соответствующей оптимизационной комбинаторной задачи. Так, в [14] показано, что из полиномиальной разрешимости задачи распознавания линейного программирования вытекает полиномиальная разрешимость оптимизационной задачи линейного программирования.

Пусть Π задача минимизации (соответственно максимизации), а J – произвольная индивидуальная задача из D_{Π} . Определим отношение $R_A(J)$ следующим образом:

$$R_A(J) = \frac{A(J)}{OPT(J)} \quad (\text{соответственно } R_A(J) = \frac{OPT(J)}{A(J)}).$$

Погрешностью приближенного алгоритма A решения задачи Π называют величину

$$R_A = \inf \{ r \geq 1 : R_A(J) \leq r \}$$

для всех индивидуальных задач $J \in D_\Pi$.

Очевидно, что при построении приближенных алгоритмов необходимо стремиться к тому, чтобы R_A было как можно ближе к единице.

Назовем аппроксимационной схемой для оптимизационной задачи Π алгоритм A , который, начиная работать на входе, состоящем из двух объектов – индивидуальной задачи $J \in D_\Pi$ и „желаемой точности“ $\varepsilon > 0$, – выдает такое допустимое решение $\hat{b} \in S_\Pi(J)$, что

$$R_{A\varepsilon}(J) \leq 1 + \varepsilon.$$

Приближенная схема A называется приближенной схемой с полиномиальным временем работы или просто приближенной полиномиальной схемой, если для каждого $\varepsilon > 0$ соответствующий алгоритм A_ε имеет полиномиальную временную сложность. Приближенная схема A называется вполне полиномиальной приближенной схемой, если ее временная сложность ограничена полиномом от $Length [J]$ и $\frac{1}{\varepsilon}$. В [5] приведена вполне полиномиальная приближенная схема решения оптимизационной задачи „о булевом рюкзаке (ранце)“.

Оказывается, что при $P \neq NP$ не существует полиномиального приближенного алгоритма A для решения задачи „о рюкзаке“ с оценкой $|A(J) - OPT(J)| \leq k$, где k – некоторая фиксированная константа [5].

Следовательно, невозможно построить полиномиальный приближенный алгоритм, решающий с произвольной наперед заданной точностью, в смысле выше приведенного неравенства, задачу линейного целочисленного программирования, сепарабельного целочисленного программирования, а также произвольную нелинейную задачу целочисленного программирования.

В заключение приведем ряд общих теорем, связывающих NP -трудные задачи и приближенные схемы их решений [5].

Теорема 1. Если существует такой полином q , зависящий от двух переменных, что для любой индивидуальной задачи J выполнено соотношение

$$OPT(J) < q(Length [J], max [J]),$$

то из существования вполне полиномиальной приближенной схемы для решения задачи Π следует существование псевдополиномиального оптимизационного алгоритма решения задачи Π .

Следствие. Пусть Π – оптимизационная задача с целочисленными величинами решений, удовлетворяющая приведенной выше теореме. Если Π является NP -трудной в сильном смысле задачей (не существует псевдополиномиального оптимизационного алгоритма), то при $P \neq NP$ не может быть решена вполне полиномиальной приближенной схемой.

Теорема 2. Пусть Π – задача минимизации, величины всех решений которой есть неотрицательные целые числа. Предположим, что для некоторого фиксированного $k \in R^+$ ей соответствует следующая задача распознавания: „Задана индивидуальная задача $J \in D_{\Pi}$. Является ли $OPT(J) \leq k$ NP -трудной?“ Тогда при $P \neq NP$ не существует полиномиального приближенного алгоритма A решения задачи Π с погрешностью $R_A < 1 + 1/k$ и Π не может быть решена полиномиальной приближенной схемой.

Приведенный краткий обзор теории NP -полноты завершим анализом возможных направлений построения эффективных алгоритмов решения NP -трудных задач комбинаторной оптимизации. В предположении истинности гипотезы $P \neq NP$ NP -трудная комбинаторная задача оптимизации, не может быть точно решена полиномиальной вычислительной схемой; для задач целочисленной оптимизации не существует полиномиальной приближенной схемы решения с произвольной точностью, так как такой схемы не существует для задачи „о рюкзаке“. Следовательно, построение и теоретические исследования точных и на их основе приближенных вычислительных схем могут вестись в следующих направлениях.

Пусть Π есть NP -трудная задача комбинаторной оптимизации.

1. Из класса Π выделяется подкласс $\Pi' \subset \Pi$, для которого при неэкспоненциально больших числовых параметрах каждой индивидуальной задачи $J \in \Pi'$ строится точная полиномиальная вычислительная схема.

2. Конструируется экспоненциальная вычислительная схема точного решения оптимизационной комбинаторной задачи Π и определяются условия, выполнение (невыполнение) которых в процессе реализации алгоритма применительно к произвольной задаче $J \in \Pi$ приводит к полиномиальной оценке сложности вычислений точного алгоритма.

При практическом применении алгоритма к конкретному классу задач дискретной оптимизации возникает проблема статистического исследования частот появления индивидуальных задач J , требующих экспоненциального времени вычислений, построения приближенных полиномиальных схем их решения, а также

получения статистических оценок отклонения от оптимального решения. Пример такого построения алгоритма для решения NP -трудной задачи приведен в [13].

3. На основании изучения структуры NP -трудной задачи и возможных ее эквивалентных преобразований определяется эффективное направление вычислительной схемы последовательного конструирования оптимального решения, обладающей экспоненциальной временной сложностью и полиномиальной памятью. Вводятся правила отсева конкурирующих вариантов, признаки оптимальности допустимого решения и оценка отклонения от оптимального решения, позволяющая на ранних стадиях алгоритма останавливать вычисление на оптимальном или достаточно „хорошем” допустимом решении. Вычислительная схема должна позволять организацию параллельного вычисления с системным эффектом, т. е. промежуточные результаты отдельных ветвей вычислений должны влиять на эффективность вычислений каждой вычислительной ветви. Практическая эффективность вычислительной схемы должна быть подтверждена статистическими исследованиями.

Учитывая сказанное выше, оценим результаты, приведенные в первой и второй главах. Как было показано, задачи линейного целочисленного программирования, сепарабельного целочисленного программирования, нелинейного целочисленного программирования являются NP -трудными.

В первой главе показано, что если коэффициенты матрицы ограничений по модулю ограничены полиномом фиксированной степени L , где L – число переменных задачи с неотрицательными коэффициентами, то:

1) задача групповой минимизации применительно к ЗЛЦП с неотрицательными переменными и фиксированным числом ограничений может быть решена алгоритмом X_U с числом операций, ограниченным полиномом с неотрицательными коэффициентами фиксированной степени L ;

2) ЗЛЦП с неотрицательными переменными и фиксированным числом связанных ограничений может быть сведена к задаче о целочисленном „ранце”, которая методом динамического программирования может быть решена с вычислительной сложностью, ограниченной полиномом с положительными коэффициентами и фиксированной степени относительно L ;

3) задачи сепарабельного целочисленного программирования блочной структуры, „лестничной” структуры ограничений, с частично заполненной матрицей ограничений специального вида, а также один класс задач нелинейного целочисленного программирования с помощью специально построенных схем последовательного конструирования вариантов могут быть решены с вычислительной сложностью, ограниченной полиномом с положи-

тельными коэффициентами и фиксированной степени относительно n .

Справедливо следующее утверждение.

Утверждение 1. Классы 1)–3) задач комбинаторной оптимизации принадлежат классу P .

Доказательство. Для любой индивидуальной задачи J из указанных классов построим функцию $Length(J)$. Тогда

$$n < Length[J].$$

Любая индивидуальная задача J из указанных классов решается алгоритмами, количество операций и необходимая память которых ограничена полиномом с положительными коэффициентами фиксированной степени относительно n . Следовательно, вычислительная сложность этих алгоритмов ограничена полиномом фиксированной степени относительно функции $Length(J)$. Утверждение доказано.

Примечание. При доказательстве предполагалось, что время выполнения всех элементарных операций (умножения, сложения, вычитания двух чисел), величины которых по модулю ограничены полиномом фиксированной степени относительно n , а также вычисление значений нелинейных функций при допустимом значении их аргументов ограничено полиномом фиксированной степени относительно n .

Очевидно, что согласно требованиям п. 1) и 2) задач комбинаторной оптимизации являются частным случаем класса 3). Однако отсюда не следует теоретическая и практическая нецелесообразность результатов, приведенных в первой главе. Действительно, по исходной ЗЛЦП и построенной ей соответствующей задаче „о целочисленном ранце” удалось найти эффективную схему решения, удовлетворяющую большинству требований, изложенных в п. 3.

Сам факт принадлежности задачи классу P не означает автоматическую возможность ее эффективного решения. Действительно, приведенные во второй главе алгоритмы решения различных классов задач сепарабельного целочисленного программирования, хотя и позволили доказать их принадлежность классу P , однако часто могут оказаться слабо эффективными в вычислительном плане – соответствующая степень полинома от n может быть чрезмерно высокой.

Тем не менее, получение полиномиальных оценок дает возможность заранее выделить такие индивидуальные задачи достаточно большой размерности, которые с полной гарантией могут быть точно решены за приемлемое время.

2. ОБЛАСТИ ПРИМЕНЕНИЯ ДИСКРЕТНЫХ МОДЕЛЕЙ В СЛОЖНЫХ СИСТЕМАХ УПРАВЛЕНИЯ

Дискретные модели оптимизации являются одним из наиболее распространенных классов математических моделей, с помощью которых формализуются задачи анализа и синтеза самых разно-

образных по своему назначению сложных систем планирования и управления. Описание различных классов моделей и конкретных областей их применения является самостоятельной темой исследования. В этом параграфе приводится обзор опубликованных ранее результатов автора по построению дискретных моделей оптимизации в сложных системах управления.

В [8] приведены три группы неравенств со следующими свойствами:

$$1) \quad a - b - N_z \leq 0, \quad -a + b - N_z \leq 0, \quad z \in \{0, 1\}. \quad (3.1)$$

Если допустимые значения a и b ограничены по модулю, то выбор соответствующего положительного числа N гарантирует при $z = 0$ совпадение значений a и b , при $z = 1$ – возможность того, что переменные a и b могут принимать произвольные допустимые решения;

$$2) \quad a - N_z \leq 0, \quad a + N \geq N_z, \quad z \in \{0, 1\}. \quad (3.2)$$

Если допустимые значения a ограничены по модулю, то выбор соответствующего положительного числа N гарантирует выполнение следующих условий: если $a < 0$, то $z = 0$ если $a > 0$ то $z = 1$;

$$3) \quad n\psi \leq \sum_{i=1}^n x_i, \quad \psi \geq \sum_{i=1}^n x_i - n + 1, \quad \forall x_i \in \{0, 1\}, \quad (3.3)$$

$$\psi \in \{0, 1\};$$

$$\sum_{i=1}^n x_i \leq n\psi \leq n \sum_{i=1}^n x_i, \quad \forall x_i \in \{0, 1\}, \quad \psi \in \{0, 1\}. \quad (3.4)$$

Неравенства (3.3) задают конъюнкцию $x_1 \wedge \dots \wedge x_n = \psi$. Неравенства (3.4) задают дизъюнкцию $x_1 \vee \dots \vee x_n = \psi$. Приведенные три группы неравенств позволяют строить линейные дискретные аналоги разнообразных в общем случае нелинейных задач планирования и управления.

1. Сведение полиномиальной задачи нелинейного программирования к задаче смешанного целочисленного программирования

В [8] рассмотрена следующая задача нелинейного программирования:

$$\max f(x), \quad X(x) \leq b, \quad x = (x_1, \dots, x_n)^T.$$

В состав показателя качества и ограничений входят следующие функции:

$$\sum_{i=1}^{k_l} l_i^l \prod_{j=1}^{n_{ij}} x_{ij}^{a_{ij}}, \quad l \in L, \quad a_{ij} > 0, \quad \alpha_j \leq x_j \leq \beta_j.$$

Остальные составляющие показателя качества и ограничений являются линейными функциями от x .

Сформулированная задача нелинейного программирования очевидным образом сводится к задаче смешанного линейного целочисленного программирования. Метод сведения, изложенный в [8], основан на возможности аппроксимации функции $\exp(x)$ в области $a \leq x \leq b$ ($a > 0$) с любой наперед заданной точностью следующей линейной дискретной моделью:

$$\exp(x) \approx \sum_{j=1}^k (b_j y_j + d_j \lambda_j), \quad \lambda_j a_j \leq y_j \leq \lambda_j a_{j+1}, \\ a = \overline{1, k}; \quad a_1 = a, \quad a_{k+1} = b, \quad \forall \lambda_j \in \{0, 1\}, \quad \sum_{j=1}^k \lambda_j = 1, \quad \sum_{j=1}^k y_j = x.$$

Коэффициенты $b_j, d_j, j = \overline{1, k}$ определяются из условий:

$$b_j a_j + d_j = \exp(a_j), \quad \exp(a_1) = \exp(a), \\ b_j a_{j+1} + d_j = \exp(a_{j+1}), \quad \exp(a_{k+1}) = \exp(b).$$

2. Задача математического программирования с невыпуклыми ограничениями

$$\max C^T x, \quad (3.5)$$

$$|x| \leq N_1 E, \quad \cup \{A_j x \leq b_j\}, \quad j = \overline{1, m}, \quad \forall x_j \equiv 0 \pmod{1} \quad (3.6)$$

E – единичная матрица, $A_j, j = \overline{1, m}$ в общем случае прямоугольные матрицы.

Задача (3.5)–(3.6) может быть сведена к модели линейного целочисленного программирования следующим образом:

$$\max C^T x,$$

$$A_j x - b_j - N \alpha E_j \leq 0, \quad j = \overline{1, m},$$

$$\sum_{j=1}^m \alpha^j \leq m-1, \quad |x| \leq N_1 E, \quad \forall \alpha^j \in \{0, 1\}, \quad \forall x_j \equiv 0 \pmod{1},$$

где $N > 0$ – заданное число такое, что при любых допустимых значениях компонент вектора x система неравенств $A_j x - b_j - N E_j \leq 0, j = \overline{1, m}$ всегда выполняется. Если x_j – непрерывные переменные, то полученная модель является моделью смешанного линейного целочисленного программирования.

3. Задача математического программирования с логическими условиями

В [11] приведена следующая формулировка задачи математического программирования с логическими условиями:

$$\min \hat{f}(x), \quad (3.7)$$

$$\exists \bar{g}_k^{y^t}(x) \in \bigcup_{i=1}^{j^k} \{Q_{ki}^{y^t} x \leq b_{ki}^{y^t}\}, \quad t = \overline{1, m_1^k} \quad (3.8)$$

$${}^n g_{y^k}^{k_j}(x) = 0, \quad j = \overline{1, m_{2y}^k}, \quad k = \overline{1, k_1}, \quad (3.9)$$

$$\text{то } \exists \bar{l}_k^{yt}(x) \in \bigcup_{i=1}^{J_k^{yt}} \{L_{ki}^{yt} x \leq \hat{b}_{ki}^{yt}\}, t=1, \overline{n_1^k} \quad (3.10)$$

$$\text{и } \rho_k^{yt}(x) = 0, t=1, \overline{n_2^k}, k=1, \overline{k_2}, \nu=1, \overline{\nu}. \quad (3.11)$$

Если условия (3.8)–(3.11) не выполняются, то

$$x \in \bigcup_{i=1}^J \{Q^i x \leq b_i\}, f^i(x) \leq 0, i=1, \overline{n_1}.$$

Функционал (3.7) задается следующим образом:

$$y = \bar{\varphi}(x). \quad (3.12)$$

$$\text{Если } \exists \bar{m}_k^{yt}(y) \bigcup_{i=1}^{J_k^{yt}} \{\Omega_{ki}^{yt} y \leq \bar{\omega}_{ki}^{yt}\}, t=1, \overline{T_k^y} \quad (3.13)$$

$$\text{и } \eta_k^{yt}(y) = 0, t=1, \overline{T_{1k}^y}, k=1, \overline{k_3} \quad (3.14)$$

$$\text{то } \hat{f}(x) = f_\nu(x), \nu=1, \overline{\nu_1} \quad (3.15)$$

При невыполнении условия (3.13)–(3.14)

$$\hat{f}(x) = f(x),$$

где $x = (x_1, \dots, x_n)^T$ вектор искомых переменных, $q_{\nu}^{kj}(x)$, $p_k^{yt}(x)$, $f^i(x)$, $f(x)$, $y = (y_1, \dots, y_m)^T$, $f_\nu(x)$, $n_k^{yt}(y)$ – линейные скалярные функции своих аргументов.

$$\bar{g}_k^y(x) = (g_{1k}^y(x), \dots, g_{nk}^y(x))^T, \bar{l}_k^{yt}(x) = (l_{1k}^{yt}(x), \dots, l_{nk}^{yt}(x))^T,$$

$$m_k^{yt}(y) = (m_{1k}^{yt}(y), \dots, m_{mk}^{yt}(y))^T; \bar{\varphi}(x) = (\varphi_1(x), \dots, \varphi_m(x))^T$$

векторные линейные функции.

$$Q_{ki}^{yt}, L_{ki}^{yt}, \Omega_{ki}^{yt}, Q_i \text{ – матрицы.}$$

В [11] показано, что задача (3.7)–(3.15) на основании приведенных выше трех групп неравенств сводится к задаче смешанного линейного целочисленного программирования. Если величины коэффициентов, входящие в задачу (3.7)–(3.15), ограничены по модулю постоянной величиной, то с заданной погрешностью задача (3.7)–(3.15) сводится к задаче линейного целочисленного программирования с частично заполненной матрицей ограничений и может быть точно решена алгоритмом с числом операций, ограниченным полиномом от n – числа переменных задачи. При этом предполагается, что все параметры задачи (3.7)–(3.15), кроме n , не изменяются.

4. Программное управление динамическими системами с постоянной и переменной структурой объекта управления

В [8] рассмотрена линейная нестационарная динамическая система вида

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \quad (3.16)$$

которую необходимо оптимально по быстродействию перевести из точки $x(t_0)$ в область, определяемую системой ограничений $Cx \leq b$ при условиях

$$\bigcup_j \{C_{1j}(t)x(t) \leq b_{1j}(t)\}, \bigcup_j \{B_{1j}(t)u(t) \leq b_{2j}(t)\}, \quad (3.17)$$

где $x(t)$ – вектор фазовых координат, $u(t)$ – вектор управления, $A(t)$, $B(t)$, C , $C_{1j}(t)$, $B_{1j}(t)$ – матрицы, b , $b_{1j}(t)$, $b_{2j}(t)$ – векторы.

В [8] задача (3.16)–(3.17) обобщается на следующий случай: рассматривается задача управления нестационарной динамической системой с переменной структурой объекта управления и ограничений.

Пусть заданы интервал управления $[t_0, T]$ и допустимые области начальных условий

$$C^0 x(t_0) \leq b^0, \quad C^n x(t) \leq b^{n*} \quad (3.18)$$

областей $\bigcup_j \{P_{ji}(t)x \leq b_{ji}(t)\} = P_j, j = \overline{1, r}$.

Области $P_j (j = \overline{1, r})$ удовлетворяют условию

$$\bar{0} \in P_j, j = \overline{1, r}.$$

Необходимо, чтобы фазовая траектория системы в процессе управления прошла первоначально через область P_1 , затем последовательно через области P_2, \dots, P_r . Между моментами времени первого попадания фазовой траектории в области $P_{j-1}, P_j (j = 2, r)$ динамическую систему управления задают моделью

$$\dot{x}(t) = A_{1\dots j-1}(t)x(t) + B_{1\dots j-1}(t)u(t),$$

$$\bigcup_i \{C'_{i1\dots j-1}(t)x(t) \leq b'_{i1\dots j-1}(t)\}, \bigcup_i \{B'_{i1\dots j-1}(t)u(t) \leq b'_{i1\dots j-1}(t)\}. \quad (3.19)$$

В момент первого попадания фазовой траектории в область P_j происходит мгновенное изменение структуры объекта управления и ограничений:

$$\dot{x}(t) = A_{1\dots j}(t)x(t) + B_{1\dots j}(t)u(t)$$

$$\bigcup_i \{C'_{i1\dots j}(t)x(t) \leq b'_{i1\dots j}(t)\}, \bigcup_i \{B'_{i1\dots j}(t)u(t) \leq b'_{i1\dots j}(t)\}. \quad (3.20)$$

До первого момента времени попадания фазовой траектории в область P_1 систему управления описывают динамической моделью

$$x(t) = A(t)x(t) + B(t)u(t) \quad (3.21)$$

$$U \{ C_i'(t)x(t) \leq b_i'(t) \}, U \{ B_i'(t)u(t) \leq b_i^2(t) \}.$$

В [8] на основании приведенных выше трех групп неравенств дискретный аналог задачи (3.16)–(3.17) и (3.18)–(3.20) сводится к моделям смешанного линейного целочисленного программирования, которые стандартным приемом могут с заданной погрешностью быть сведены к моделям линейного целочисленного программирования.

Используя свойство марковости обыкновенных систем дифференциальных уравнений, можно сконструировать целочисленные модели, удовлетворяющие условиям гл. 2, и, следовательно, получить оптимальное решение за число операций, ограниченное полиномом от n^* – количества точек дискретизации интервала времени управления $[t_0, T]$. Этот прием описан в п. 4 настоящей главы при рассмотрении немарковских дискретных динамических процессов управления.

5. Анализ и синтез нелинейных динамических систем

Рассмотрим устойчивость нелинейной системы дифференциальных уравнений

$$\dot{x}_j(t) = f_j(t, x_1, \dots, x_n), \quad j = \overline{1, n} \quad (3.22)$$

В [8] приведено следующее утверждение 1.

Пусть в некоторой области Ω фазового пространства E_n существует функция $V(t, x)$, для которой

$$k_1 \sum_{i=1}^n x_i^2 \leq V(t, x) \leq k_2 \sum_{i=1}^n x_i^2, \\ \frac{dV(t, x(t))}{dt} \leq -\alpha(t, x(t)) \quad (3.23)$$

$$t \in [t_0, T], \quad x(t) \in \Omega, \quad k_1, k_2 > 0, \quad \alpha(t, x(t)) > 0,$$

где неравенство (3.23) определено с учетом (3.22). Тогда если вектор-функция $f(t, x)$ удовлетворяет всем условиям существования, непрерывности и единственности решения системы дифференциальных уравнений (3.22) в области Ω , то в этой области оценить решение системы (3.22) можно следующим образом:

$$\sum_{i=1}^n x_i^2(t) \leq \frac{k_2}{k_1} \sum_{i=1}^n x_i^2(t_0) \exp[-k_3(t-t_0)], \quad (3.24)$$

где $k_2 \geq 0, \alpha(t, x) \geq k_3 V(t, x), t \in [t_0, T], x(t) \in \Omega$. Приведенное утверждение имеет ряд следствий, позволяющих оценить устойчивость системы (3.22) в конечных областях фазового пространства, не включающую начало координат.

Следствие 1. Если исследуемая система имеет вид

$$\dot{x}(t) = f(x(t)) \quad (3.25)$$

и существует функция $V(x)$, удовлетворяющая в области Ω выше приведенным условиям, то для всех решений системы (3.22), принадлежащих Ω , справедлива оценка (3.24) при любом $t \geq t_0$.

Следствие 2. Если область $R \in \Omega$ содержит все векторы x , принадлежащие множеству $r_1 \leq \sum_{i=1}^n x_i^2 \leq r_2$, то для всех начальных условий системы (3.22), удовлетворяющих соотношению при $\sum_{i=1}^n x_{i0}^2 \leq r_1$, $\frac{k_2}{k_1} r_1 \leq r_2$ или при $\sum_{i=1}^n x_{i0}^2 \leq \frac{k_2}{k_1} \sum_{i=1}^n x_{i0}^2 \leq r_2$, можно указать такое значение $\delta > 0$, для которого при $t \in [t_0, T]$

$$\sum_{i=1}^n x_i^2(t) \leq \delta. \quad (3.26)$$

Для системы (3.25) оценка (3.26) справедлива при $t \in [t_0, \infty]$. Верхняя граница δ

$$\frac{k_2}{k_1} \sum_{i=1}^n x_{i0}^2, \quad \text{если } \sum_{i=1}^n x_{i0}^2 > r_1 \text{ и } \frac{k_2}{k_1} r_1, \quad \text{если } \sum_{i=1}^n x_{i0}^2 \leq r_1.$$

Следствие 3. При доказательстве утверждения не использовался факт, что $x(t) \equiv 0$ является решением системы (3.22). Следовательно, оценка (3.24) и следствия 1, 2 справедливы для случая, когда система (3.22) записана не в отклонениях от некоторого частного решения исходной системы дифференциальных уравнений, а относительно произвольной непрерывной траектории. Таким образом, чем меньше отклонение $\frac{k_2}{k_1}$ в области Ω , тем быстрее затухает переходный процесс, тем при больших отклонениях по евклидовой норме $x(t_0)$ от начала координат система остается устойчивой.

В [8] задача построения наилучшей (в смысле минимума отношения $\frac{k_2}{k_1}$) из класса $\{\sum_{i=1}^n a_i \varphi_i(x)\}$ функции $V(\dot{x})$ сводится к задаче линейного программирования.

Здесь a_i – неизвестные коэффициенты, $\varphi_i(x)$ – произвольные известные базовые функции.

Принцип сведения заключается в том, что по области Ω строится сеть Ω^* и для каждой точки $x^* \in \Omega^*$ для искомой $V_{\text{опт}}(x)$ накладываются ограничения

$$k_1 \sum_{i=1}^n (x_i^*)^2 \leq V_{\text{опт}}(x^*) \leq k_2 \sum_{i=1}^n (x_i^*)^2, \quad (3.27)$$

$$\dot{V}(x^*) \leq -\alpha(x^*), \quad \forall x^* \in \Omega^*, \quad \alpha(x^*) > 0, \quad (3.28)$$

для стационарного случая и для каждой пары (x^*, t^*) , $\forall x^* \in \Omega^*$, $\forall t^* \in T^*$, где $T^* = \{t_0, t_1, \dots, t_k = T\}$ накладываются ограничения

$$k_1 \sum_{i=1}^n (x_i^*)^2 \leq V_{\text{опт}}(t^*, x^*) \leq k_2 \sum_{i=1}^n (x_i^*)^2, \quad (3.29)$$

$$\dot{V}(t^*, x^*) \leq -\alpha(x^*), \quad (3.30)$$

для нестационарного случая.

Неравенства (3.27)–(3.28) и (3.29), (3.30) входят в состав соответствующих задач гиперболического программирования, переменными которой являются $k_1, k_2, a_i, i = \overline{1, L}$. Очевидным образом эти задачи преобразуются в задачи линейного программирования. Выделены оценки для шага дискретизации Δ и Δt , на основании которых строятся сеть Ω^* и множество T^* , гарантирующее построение оптимальной функции с заданной погрешностью относительно функционала $\min \frac{k_2}{k_1}$, предложен метод последовательного конструирования $V_{\text{опт}}$.

Изложенный подход распространен на стохастические системы дифференциальных уравнений вида

$$dx(t) = b(t, x) dt + \sum_{r=1}^k \sigma_r(t, x) dz_r(t),$$

$$x(t) = (x_1(t), \dots, x_n(t))^T,$$

где $z_r(t)$ – винеровские независимые процессы.

Сформулировано утверждение 2, которое является аналогом утверждения 1 для стохастических дифференциальных уравнений, на основании которого строятся соответствующие задачи линейного программирования.

В заключение отметим, что на основании предложенного в п. 5 подхода к исследованию устойчивости нелинейных дифференциальных уравнений, а также результатов, приведенных в п. 1 настоящего параграфа, в [8] задача структурного синтеза технически устойчивой нелинейной динамической системы была сведена к задаче смешанного линейного целочисленного программирования.

3. РАСПРЕДЕЛЕНИЕ РЕСУРСОВ В ЗАДАЧЕ ПЛАНИРОВАНИЯ ВЫПОЛНЕНИЯ ЧАСТИЧНО-УПОРЯДОЧЕННОГО МНОЖЕСТВА РАБОТ

Задано множество работ $D = \{d_i\}, i = \overline{1, n}$. На множестве работ D определено отношение частичного порядка в виде ориентированного ациклического графа, удовлетворяющего следующим условиям:

а) количество вершин графа (i -й вершине графа соответствует i -я работа), из которых выходят $k_i n$ ориентированных дуг ($k_i > 0$), ограничено числом, не зависящим от n ;

б) из остальных вершин графа количество выходящих ориентированных дуг ограничено числом, не зависящим от n .

Каждая работа выполняется отдельным подразделением, которому необходимо выделить ресурсы для ее выполнения. Длительность выполнения t_i i -й работы связана с ресурсами следующим образом:

$$t_i = f_i(x^i) \quad (3.40)$$

где x^i – целочисленное количество обобщенного ресурса, необходимое для выполнения i -й работы, $f_i(\cdot)$ – целочисленная функция. x^i определяется следующим образом:

$$x^i = \left[\sum_{j=1}^m b_{ij} x_j^i \right],$$

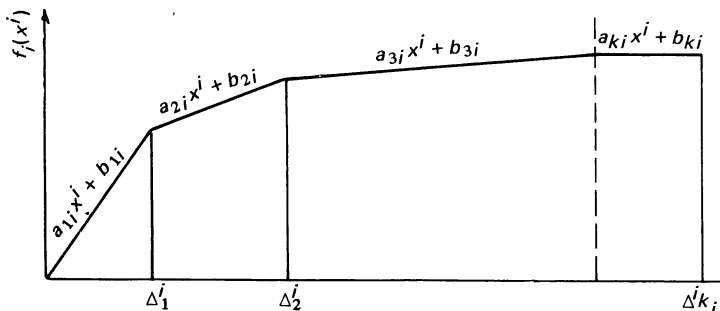
где $[\cdot]$ – ближайшее целое число в масштабе измерений i -го обобщенного ресурса, b_{ij} – заданные целочисленные коэффициенты, x_j^i – целочисленное количество j ресурса, выделенного для i -й работы.

Для каждого i , $i = \overline{1, n}$, существует множество индексов g_i таких, что для $\forall j \in g_i$

$$x_j^i \leq \sum_{l=1}^{p_j} a_{jl} x_{jl}^i, \quad (3.41)$$

где a_{jl} – целочисленные коэффициенты.

Функция $f_i(x^i)$ может быть целочисленным полиномом и ломаной линией, изображенной на рисунке



Здесь $\forall a_{ij}, b_{ij}$ – целые числа. Целочисленность коэффициентов a_{ij}, b_{ij} можно достичь соответствующим выбором единицы измерения времени одинакового для всех i . $\forall k_i \leq k$, k не зависит от n .

Тогда функция $f_i(x^i)$ может быть представлена в виде

$$f_i(x^i) = \sum_{j=1}^{k_i} (a_{ij} x^{ij} + b_{ij} \lambda_j^i)$$

при ограничениях

$$x^i = \sum_{j=1}^{k_i} x^{ij}; \quad x^i \leq \Delta_{k_i}^i, \quad 0 \leq x^{il} \leq \Delta_l^i \lambda_l^i,$$

$$\lambda_1^i \Delta_1^i \leq x^{i2} \leq \Delta_2^i \lambda_i^2, \dots, \lambda_i^{k_i-1} \Delta_{k_i-1}^i \leq x^{i k_i} \leq \Delta_{k_i}^i \lambda_i^{k_i}, \quad (3.41)$$

$$\lambda_i^j \in \{0, 1\}, \quad j = \overline{1, k_i}, \quad \sum_{j=1}^{k_i} \lambda_i^j = 1.$$

На m ресурсах наложены следующие ограничения:

$$\sum_{i=1}^n x_j^i \leq M_j, \quad j = \overline{1, m}.$$

Необходимо найти длительности и времена выполнения работ, удовлетворяющих вышеприведенным ограничениям по критериям:

минимизация взвешенной суммы расходов ресурсов при ограничении на общее время выполнения всех работ;

минимизация общего времени выполнения всех работ.

Математическая модель описанной задачи имеет вид

$$\min \sum_{j=1}^m c_j \sum_{i=1}^n x_i^j; \quad (3.42)$$

$$t_i \geq t_{i_0} + \sum_{j=1}^{k_{i_0}} (a_{i_0 j} x^{i_0 j} + b_{i_0 j} \lambda_{i_0}^j); \quad (3.43)$$

$$l = \overline{1, J_i}, \quad i = \overline{1, n};$$

$$0 \leq x^{i1} \leq \Delta_1^i \lambda_i^1, \lambda_i^1 \Delta_1^i \leq x^{i2} \leq \Delta_2^i \lambda_i^2, \dots, \lambda_i^{k_i-1} \Delta_{k_i-1}^i \leq x^{i k_i} \leq \Delta_{k_i}^i \lambda_i^{k_i}, \quad (3.44)$$

$$\lambda_i^j \in \{0, 1\}, \quad j = \overline{1, k_i}, \quad \sum_{j=1}^{k_i} \lambda_i^j = 1; \quad (3.45)$$

$$\sum_{j=1}^{k_i} x^{ij} = \left[\sum_{j=1}^m b_{ij} x_j^i \right]; \quad (3.46)$$

$$x_j^i \leq \sum_{l=1}^{p_j^i} a_{jl} x_{jl}^i, \quad \forall j \in g_i, \quad i = \overline{1, n}; \quad (3.47)$$

$$\sum_{j=1}^n x_j^i \leq M_i; \quad (3.48)$$

$$t_{i_m} + \sum_{j=1}^{k_{i_m}} a_{i_m j} x^{i_m j} + b_{i_m j} \lambda_{i_m}^j \leq T_{i_m}, \quad m = \overline{1, M}. \quad (3.49)$$

J_i – множество работ непосредственно предшествующих работе i ; d_{i_m} ($m = \overline{1, M}$) – множество работ, не имеющих непосредственных приемников; t_i – целочисленные моменты времени начала выполнения i -й работы.

Функционал имеет вид

$$\min t \quad (3.50)$$

Ограничениями задачи являются ограничения (3.43)–(3.48), а также дополнительно:

$$t \geq t_{i_m} + \sum_{j=1}^{k_{i_m}} (a_{i_m j} x^{i_m j} + b_{i_m j} \lambda_{i_m}^j), \quad m = \overline{1, M}. \quad (3.51)$$

Если величины всех коэффициентов, входящих в неравенства (3.43)–(3.49), ограничены полиномами фиксированной степени от

A^t, B_i^t, C_1^t, C_2^t – матрицы, C^t, b_1^t, b_2^t – векторы. Необходимо систему (3.58)–(3.60) оптимально по быстродействию привести в область

$$C y \leq b \quad (3.61)$$

где C – матрица, b – вектор. Иными словами, необходимо найти последовательность управлений, которой бы соответствовало минимальное число t_{min} , при котором

$$C y_{t_{min}} \leq b.$$

Такого рода задача и ее модификации для динамических систем рассматривались в гл. 2, 3 [8].

Покажем, что при выполнении определенных условий все задачи, приведенные в гл. 2, 3 [8], сформулированные для дискретных процессов при условии, что S не зависит от T , могут быть решены алгоритмами с полиномиальной оценкой числа операций относительно T – количества моментов времени, в которые задан дискретный управляемый процесс.

Рассмотрим следующую модель дискретной оптимизации:

$$\min \sum_{t=1}^T \omega_t \quad (2.62)$$

$$C_1^t (A^t y_{t-1} + \sum_{i=0}^S B_i^t x_{t-i} + c^t) \leq b_1^t + N \sum_{l=1}^t (1-z_l) E_1, \quad t = \overline{1, T}; \quad (3.63)$$

$$C_2^t x_t \leq b_2^t + N \sum_{l=1}^t (1-z_l) E_2, \quad t = \overline{1, T}; \quad (3.64)$$

$$C (A^t y_{t-1} - \sum_{i=0}^S B_i^t x_{t-i} + c^t) - b - N z_t E_3 \leq 0; \quad (3.65)$$

$$\forall z_t \in \{1, 0\}, \quad t = \overline{1, T};$$

$$\omega_0 = 1; \quad 2\omega_t \leq \omega_{t-1} + z_t; \quad \omega_t \geq \omega_{t-1} + z_t - 1; \quad t = \overline{1, T}; \quad (3.66)$$

$$\forall \omega_t \in \{0, 1\}.$$

E_1, E_2, E_3 – единичные матрицы, N – константа, определяемая условиями задачи, не зависящая от t . Ограничения (3.63), (3.64) реализуют условие (3.60).

В соответствии со структурой ограничений (3.63)–(3.66) минимуму функционала (3.62) соответствует оптимальное решение задачи управления (3.58)–(3.61).

Рассмотрим, например, решение задачи (3.62)–(3.66) – $y_t^{opt}, x_t^{opt}, z_t^{opt}, \omega_t^{opt}, t = \overline{1, T}$. Пусть t_{min} – момент попадания вектора $y_{t_{min}}$ в область $C y_{t_{min}} \leq b$. В силу структуры ограничений (3.63)–(3.65) для всех $t < t_{min}$, $z_t = 1; z_{t_{min}} = 0$. Поскольку

$\sum_{l=t_{min}}^t (1 - z_l) \geq 1$ для всех $t > t_{min}$, то ограничения на векторы y_t, x_t резко расширяются, N выбирается таким, чтобы

для любых $y_{t_{min}}, x_{t_{min}-1}, l = \overline{0, S}$ существовало допустимое решение задачи (3.62)–(3.66). Ограничения (3.66) формируют булевы переменные ω_t , удовлетворяющие следующим свойствам: для всех $t < t_{min}$, $\omega_t = 1$, для всех $t \geq t_{min}$, $\omega_t = 0$. Коэффициенты матриц $A^t, B_i^t, C_1^t, C_2^t, C$, векторов c^t, b_1^t, b_2^t являются рациональными числами. Компоненты векторов y_t, x_t являются непрерывными переменными. Если величина коэффициентов этих матриц и векторов определять с точностью до фиксированного числа a_1 знаков после запятой, а переменные y_{ti}, x_{ti} представить в виде

$$\forall y_{ti} = 10^{-a_1} \hat{y}_{ti}, \quad x_{ti} = 10^{-a_2} \hat{x}_{ti},$$

где a_1, a_2 – натуральные числа, не зависящие от T , $\forall \hat{y}_{ti}, \hat{x}_{ti}$ – целочисленные переменные, то задача (3.62)–(3.66) превращается в задачу линейного целочисленного программирования. Если при этом значения коэффициентов матриц $A^t, B_i^t, C_1^t, C_2^t, C$ векторов $c^t, b_1^t, b_2^t, = \overline{1, T}$ по модулю ограничены фиксированными числами, не зависящими от T , то полученная задача линейного целочисленного программирования удовлетворяет условиям п. 5, гл. 2, сформулированным для частично заполненных матриц ограничений и, следовательно, может быть решена точно алгоритмом, имеющим полиномиальную относительно T оценку качества числа операций.

С помощью результатов, приведенных в п. 2 гл. 3, задача (3.58)–(3.61), оставаясь полиномиально разрешимой, может быть обобщена на случай, когда ограничения (3.60), (3.61) задаются в виде

$$\bigcup_{j=1}^{L_1} \{c_{1j}^t y_t \leq b_{1j}^t\}; \quad \bigcup_{j=1}^{L_2} \{c_{2j}^t x_t \leq b_{2j}^t\}; \quad \bigcup_{j=1}^{L_3} \{c_j y \leq b_j\},$$

где L_1, L_2, L_3 не зависят от T .

n , то задачи линейного целочисленного программирования (3.42)–(3.49); (3.50), (3.43)–(3.49), (3.51) могут быть решены алгоритмом, имеющим полиномиальную оценку сложности относительно n -числа работ.

Действительно, матрица ограничений задачи распределения ресурсов является частично заполненной и удовлетворяет условиям, сформулированным в п. 5 гл. 2. Специфика группы ограничений (3.46) естественным образом может быть учтена при использовании алгоритма, изложенного в п. 5 гл. 2.

4. УПРАВЛЕНИЕ ДИСКРЕТНЫМИ ДИНАМИЧЕСКИМИ СИСТЕМАМИ

Предположим, что в некотором дискретном процессе с этапами $0, 1, 2, \dots, T$ преобразование состояния y_{t-1} этапа $t-1$ в состояние y_t этапа t осуществляется с помощью управлений $x_{t-s}, x_{t-s+1}, \dots, x_t$ и уравнений состояния:

$$y_t = p_t(y_{t-1}, x_{t-s}, \dots, x_t), \quad t=1, 2, \dots, T,$$

$$y_t = (y_t, \dots, y_{tn})^T, \quad x_t = (x_t, \dots, x_{tm})^T.$$

Эти уравнения охватывают возможности задержек во времени в процессе и влияние принятых ранее решений на формирование состояния в текущий момент. Предположим, что состояния y_t выбирают, исходя из множеств состояний Y_t , областей управления X_t и множеств Z_t . Для того чтобы процесс мог начаться, необходимо знать начальные условия, задающие состояние процесса y_0 в момент $t=0$ и выбранные в моменты $t=t-s, t=2-s, \dots, 0$ управления x_t^0 . Предположим, что развитие процесса при каждом значении t количественно оценивается вещественным сепарабельным функционалом.

$$\min \sum_{i=1}^{m_1} f'_{ti}(y_{t-1,i}) + \sum_{l=0}^s \sum_{j=1}^{m_2} f''_{t-l,j}(x_{t-l}).$$

Необходимо найти минимум суммарной по t оценки. Такая постановка задачи приводит к следующей немарковской динамической задаче дискретной оптимизации [15].

$$\min \sum_{t=1}^T \sum_{i=1}^{m_1} f'_{ti}(y_{t,i}) + \sum_{t=1}^T \sum_{j=1}^{m_2} \varphi_{tj}(x_t) + \sum_{i=1}^{m_1} f_0(y_{Ti}); \quad (3.52)$$

$$y_t = p_t(y_{t-1}, x_{t-s}, \dots, x_t), \quad x_t \in X_t, \quad y_t \in Y_t \quad (3.53)$$

$$(y_{t-1}, x_{t-s}, \dots, x_t) \in Z_t, \quad t=1, T, \quad (3.54)$$

$$x_t = x_t^0, \quad t=1-s, \dots, 0^*, \quad y_0 = y^0.$$

При $s=0$ динамическая дискретная система превращается в марковскую.

Рассмотрим случай, когда модель (3.52)–(3.54) приводит к алгоритмам оптимизации, имеющим полиномиальную сложность относительно T – количества моментов дискретизации.

Сепарабельная целочисленная модель.

Ограничения (3.53)–(3.54) зададим в виде

$$a_i^t y_{ti} = \sum_{j=1}^{m_1} a_{ij}^t (y_{(t-1)j}) + \sum_{l=0}^s \sum_{j=1}^{m_2} b_{ijl}^t (x_{t-l,j}), \quad i = \overline{1, m_1}; \quad (3.55)$$

$$a_i^t \leq x_{ti} \leq a_{2i}^t; \quad a_{3j}^t \leq y_{tj} \leq a_{4j}^t, \quad i = \overline{1, m_2}, \quad j = \overline{1, m_1} \quad (3.56)$$

$$\sum_{j=1}^{m_1} c_{ij}^t (y_{(t-1)j}) + \sum_{l=0}^s \sum_{j=1}^{m_2} d_{ijl}^t (x_{t-l,j}) \leq b_i^t, \quad i = \overline{1, J}, \quad |J| \leq M \quad (3.57)$$

M не зависит от t ; $\forall a_i^t, a_{1i}^t, a_{2i}^t, a_{3i}^t, a_{4i}^t$ – целые числа; $\forall a_{ij}^t(y_{(t-1)j}), b_{ijl}^t(x_{t-l,j}), c_{ij}^t(y_{(t-1)j}), d_{ijl}^t(x_{t-l,j})$ – целочисленные полиномы от $y_{(t-1)j}, x_{t-l,j}$ соответственно. Степень полиномов ограничена фиксированным числом, не зависящим от t . Значения коэффициентов полиномов по модулю ограничены фиксированным числом, не зависящим от T .

Переменными задачи (3.52)–(3.57) являются

$$y_{ti}, \quad t = \overline{1, T}, \quad i = \overline{1, m_1}, \quad x_{ti}, \quad t = \overline{0, T}, \quad i = \overline{1, m_2}.$$

В общем случае эти переменные являются непрерывными, однако если оптимальное решение ищется с точностью до фиксированного числа знаков после запятой, задача (3.52), (3.55)–(3.57) сводится к целочисленной задаче дискретной оптимизации относительно переменных $\hat{y}_{ti}, \hat{x}_{ti}$ путем замены $y_{ti} = 10^{-a} \hat{y}_{ti}$, $x_{ti} = 10^{-a} \hat{x}_{ti}$, a – фиксированное количество знаков после запятой. $\forall \hat{y}_{ti}, \hat{x}_{ti}$ – целочисленные переменные. Легко видеть что структура матрицы ограничений таким образом сформулированной задачи сепарабельного целочисленного программирования удовлетворяет условиям, сформулированным в п. 5 гл. 2 для частично заполненных матриц ограничений и, следовательно, строго может быть решена алгоритмом (п. 5, гл. 2) с полиномиальной оценкой числа операций относительно T – количества моментов времени, на котором задан дискретный процесс.

Дискретная модель, оптимальная по быстродействию.

Рассмотрим дискретную модель вида

$$y_t = A^t y_{t-1} + \sum_{i=0}^s B_i^t x_{t-i} + C^t, \quad t = \overline{1, T}; \quad (3.58)$$

$$x_t = x_t^0, \quad t = 1-s, \dots, 0; \quad y_0 = y^0, \quad (3.59)$$

при ограничениях

$$C_1^t y_t \leq b_1^t, \quad C_2^t x_t \leq b_2^t, \quad t = \overline{1, T}. \quad (3.60)$$

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. **Веселов С. И., Шевченко В. Н.** Об экспоненциальном росте коэффициентов агрегирующего уравнения: Тез. докл. 4 Феодосийской конф. по пробл. теорет. кибернетики. – Новосибирск: Ин-т математики СО АН СССР, 1977. – 53 с.
2. **Волкович В. Л., Волошин А. Ф.** Об одном алгоритме решения задачи дискретного сепарабельного программирования//Исслед. операций в АСУ. – 1977. – 9. – С. 50–52.
3. **Волкович В. Л., Волошин А. Ф.** Об одной схеме последовательного анализа и отсеивания вариантов//Кибернетика. – 1978. – № 4. – С. 46–52.
4. **Галочкин И., Нестеренко Ю. В., Шидковский А. Б.** Введение в теорию чисел. – М.: Изд-во Моск. ун-та, 1984. – 280 с.
5. **Гэри М., Джонсон Д.** Вычислительные машины и труднорешаемые задачи. – М.: Мир, 1982. – 416 с.
6. **Зайченко Ю. П.** Исследование операций. – К.: Вища шк., Головное изд-во, 1979. – 320 с.
7. **Михалевич В. С., Волкович В. Л.** Вычислительные методы исследования и проектирования сложных систем. – М.: Наука, 1983. – 285 с.
8. **Павлов А. А.** Линейные модели в нелинейных системах управления. – К.: Техніка, 1982. – 166 с.
9. **Павлов А. А.** Анализ оценок сложности вычислений точных алгоритмов решения задачи линейного целочисленного программирования общего вида//Автоматика. – 1985. – № 5. – С. 42–48.
10. **Павлов А. А.** Оптимизация больших систем//Вторая науч.-техн. конф. советских и польских мол. ученых – выпускников учеб. заведений СССР. Киев 14–17 окт. 1986 г. – К., 1986. – С. 101–104.
11. **Павлов А. А.** Сведение одного класса задач линейного программирования с логическими условиями к задаче смешанного линейного целочисленного программирования//Электрон. моделирование. – 1985. – № 1. – С. 46–48.
12. **Павлов А. А., Гершгорин А. Е.** Об одном методе сведения задачи линейного целочисленного программирования общего вида к задаче „о ранце“//Автоматика. – 1985. – № 1. – С. 52–56.
13. **Павлов А. А., Мисюра Е. Б.** Построение расписания обслуживания частично упорядоченного множества заданий по критерию минимизации суммарного штрафа/Киев. политехн. ин-т. К., 1986. – 58 с. – Деп. в УкрНИИТИ 17.07.86, № 1759–Ук86.
14. **Паладимитриу Х., Стайглиц К.** Комбинаторная оптимизация. Алгоритмы и сложность. – М.: Мир, 1985. – 510 с.
15. **К. Рихтер.** Динамические задачи дискретной оптимизации. – М.: Радио и связь, 1985. – 136 с.
16. **Т. Саати.** Целочисленные методы оптимизации и связанные с ними экстремальные проблемы. – М.: Мир, 1973. – 302 с.
17. **Автоматизированные системы управления гибкими технологиями** /В. И. Скурихин, А. А. Павлов, Э. П. Путилов, С. Н. Гриша. – К.: Техніка, 1987. – 184 с.
18. **Ху Т.** Целочисленное программирование и потоки в сетях. – М.: Мир, 1974. – 519 с.
19. **Bradley G. H.** Transformation of Integer Programs to Knapsack Problems //Discrete Mathematics. – 1971. – 1. – P. 29–45.

Монография

Александр Анатольевич Павлов

**АЛГОРИТМИЧЕСКОЕ
ОБЕСПЕЧЕНИЕ
СЛОЖНЫХ
СИСТЕМ УПРАВЛЕНИЯ**

Обложка художника

С. И. Райхлина

Художественный редактор

С. П. Духленко

Технический редактор

Е. С. Неведрова

Корректор

Е. М. Алексеева

Оператор

Н. П. Довлетукаева

ИБ № 11900

Подписано в печать 17.05.89. БФ 00051. Формат 84X108^{1/32}.
Бумага офсетная № 1. Гарнитура Цюрих. Печать офсетная. Усл. печ. л. 8,82. Усл. кр.-отт. 9,07. Уч.-изд. л. 8,16.
Тираж 1700 экз. Изд. № 8121. Зак. № 9-394 Цена 1 р. 60 к.

Головное издательство издательского объединения „Выща школа”, 252054, Киев-54, ул. Гоголевская, 7

Напечатано с оригинала-макета, подготовленного в Головном издательстве издательского объединения „Выща школа”, в Киевской книжно-журнальной типографии научной книги, 252004, Киев-4, ул. Репина, 4

1 р. 60 к.

