1.2 Метод ветвей и границ

1.2.1. Общая схема метода

Этот метод применим как к полностью целочисленным задачам, так и частично целочисленным задачам. Рассмотрим задачу

Рассмотрим задачу

$$c'x \to \max, \tag{17}$$

$$Ax \leq b, \tag{18}$$

$$d_{*j} \leq x_j \leq d_j^*, \; j \in J, \tag{19}$$

$$x_j \text{ -- целое, } j \in I, \tag{20}$$
$$x = (x_j, j \in J), \; J = \{1, 2, ..., n\}, \; I \subset J.$$

Заметим, что без ограничения общности для $j \in I$ числа $d_{*j}, \; d_j^*$ можно считать целыми.

Идея метода ветвей и границ основана на следующем элементарном факторе. Рассмотрим любую переменную $x_j, \; j \in I$.

Пусть l -- некоторое целое число, такое что

$$d_{*j} \leq l \leq d_j^* - 1.$$

Тогда оптимальное значение x_j^0 будет удовлетворять либо неравенству

$$l + 1 \leq x_j \leq d_j^*,$$

либо неравенству

$$d_{*j} \leq x_j \leq l.$$

Для иллюстрации возможности использования этого факта предположим, что в задаче (17) -- (19) условие целочисленности не учитывается. Пусть в результате решения непрерывной задачи мы получим оптимальный план, у которого $x_1 = 1\frac{1}{3}$.

Далее рассмотрим две задачи:
задачу

$$c'x \to \max, \tag{21}$$

$$Ax \le b, d_{*j} \le x_j \le d_j^*,\ j \in J \backslash 1, \tag{22}$$

$$d_{*1} \le x_1 \le 1, \tag{23}$$

и задачу (21), (22) с условием

$$2 \le x_1 \le d_1^*. \tag{24}$$

Предположим теперь, что каждая из полученных задач имеет оптимальные целочисленные решения соответственно x^{01} и x^{02} . Тогда очевидно, что решением исходной задачи (17) -- (20) является тот из векторов x^{01} или x^{02}, на котором значение $c'x$ больше.

1.2.2 Алгоритм

Рассмотрим общую итерацию метода. Пусть мы осуществляем *t*-ю итерацию.

В начале t -й итерации имеем:

1) число r_0^t , которое является оценкой снизу значения целевой функции исходной задачи на оптимальном плане;

2) список задач ЛП, которые подлежат решению. Эти задачи отличаются от (17)-- (19) и друг от друга только условиями (19);

\noindent 3) *n*-вектор μ , число μ_0 .

Замечание. На первой итерации, т.е. при $t = 1$, список задач ЛП состоит только из одной задачи (17) -- (19). Для определения $r_0^1,\ \mu,\ \mu_0$ можно поступить следующим образом.

Если известен какой-либо *целочисленный* план $\bar{x}$ задачи (17) -- (20), то полагаем $r_0^1 = c'\bar{x}$, $\mu = \overline{x},\ \mu_0 = 1$. Если такого плана нет, то полагаем $\mu_0 = 0$, в качестве μ берем любой *n*-вектор, а для построения оценки r_0^1 , удовлетворяющей неравенству $r_0^1 \le c'x^0$, где x^0 -- решение задачи (17) -- (20), можно привлечь любую дополнительную информацию. Например, если $c \ge 0$ и $d_* \ge 0$, то очевидно, что c'x^{0} \ge 0,\] следовательно, можно положить $r_0^1 = 0$. В самом худшем случае, когда нет никакой дополнительной информации, мы полагаем $r_0^1 = -\infty$.

На произвольной итерации *t* выполняем следующие шаги.

Шаг 1. Если основной список пуст, идем на шаг 5. В противном случае выбираем любую задачу ЛП из списка и идем на шаг 2.

Шаг 2. Решаем выбранную задачу ЛП. Если эта задача не имеет решения либо она имеет решение x^* , для которого

$$c'x^* \le r_0^t,$$

то полагаем $r_0^{t+1} = r_0^t$, вычеркиваем данную задачу ЛП из списка и возвращаемся к началу новой *(t+1)* -й итерации (идем на шаг 1, заменив *t* на *{t} + 1*).

Если выбранная задача ЛП имеет решение x^* и

$$c'x^* > r_0^t,$$

то идем на шаг 3.

Шаг 3. Если на решении x^* задачи ЛП выполняется условие целочисленности (20), то фиксируем это решение, т.е. полагаем $\mu = x^*, \ \mu_0 = 1$. Изменяем оценку

$$r_0^{t+1} = c'x^*,$$

вычеркиваем рассмотренную задачу ЛП из списка и переходим к новой *(t + 1)*-й итерации (идем на шаг 1, заменив *t* на *t+1*).

Если на решении x^* задачи ЛП условие целочисленности (20) не выполняется, то идем на шаг 4.

Шаг 4. Выберем любую переменную $x^*_{j_0}, \ j_0 \in I$, которая не удовлетворяет условию целочисленности. Пусть $x^*_{j_0} = l_{j_0}$, l_{j_0} -- нецелое число. Обозначим через $[l_{j_0}]$ целую часть числа l_{j_0} , т.е. $[l_{j_0}]$ -- это наибольшее целое, удовлетворяющее неравенству

$$[l_{j_0}] \leq l_{j_0}.$$

Например, [3] = 3, [3, 5] = 3, [--3, 5] = --4.

Удалим старую рассматриваемую задачу ЛП из списка, а вместо нее добавим две новые задачи ЛП. Эти задачи отличаются от задачи ЛП, выбранной на шаге 1, и друг от друга только прямыми ограничениями на переменную x_{j_0} .

В первой новой задаче ЛП эти ограничения имеют вид

$$\bar{d}_{*j_0} \leq x_{j_0} \leq [l_{j_0}],$$

во второй задаче эти ограничения имеют вид

$$[l_{j_0}] + 1 \leq x_{j_0} \leq \bar{d}^*_{j_0}.$$

Здесь $\bar{d}_{*j_0}, \bar{d}^*_{j_0}$ -- нижняя и верхняя границы на переменную x_{j_0} , которые были в задаче ЛП, выбранной на шаге 1.

Полагаем $r_0^{t+1} = r_0^t$, переходим к новой *(t+1)*-й итерации (идем на шаг 1, заменив *t* на *t+1*)).

Шаг 5. Останавливаем алгоритм. При $\mu_0 = 1$ вектор μ принимаем за решение задачи (17) -- (20). При $\mu_0 = 0$ в задаче (17) -- (20) нет допустимых планов.

Пример. Рассмотрим следующую задачу ЦЛП

$$3x_1 + 3x_2 + 13x_3 \to \max, \tag{25}$$

$$- - 3x_1 + 6x_2 + 7x_3 \leq 8, \tag{26}$$

$$6x_1 - -3x_2 + 7x_3 \leq 8, \tag{27}$$

$$0 \leq x_j \leq 5, \ j = \overline{1,3}, \tag{28}$$

$$x_j \text{ -- целое}, \ j = \overline{1,3}\,. \tag{29}$$

Перед первой итерацией список задач состоит из одной задачи № 1, которая совпадает с задачей (25) - (28).

Так как $x_j \geq 0,\ j=\overline{1,3}$, то очевидно, что

$$3x_1 + 3x_2 + 13x_3 \geq 0$$

при любом допустимом плане. Значит, в качестве оценки снизу целевой функции исходной задачи можем взять число $r_0^1 = 0$.

Опишем алгоритм решения задачи (25)- (28) по итерациям.

Итерация 1. Список состоит из одной задачи, у которой оценка снизу равна $r_0^1 = 0$.

Решим задачу № 1 из списка задач. Эта задача имеет решение:

$$x_1^* = x_2^* = 2\frac{2}{3},\ x_3^* = 0,\ c'x^* = 16. \text{ (задача № 1).}$$

Решение задачи № 1 нецелочисленное, $c'x^* = 16 > 0 = r_0^1$. Переходим к шагу 4.

Выберем переменную x_1 для ветвления. Задачу № 1 вычеркнем из списка, вместо неё включаем две новые задачи № 2, № 3.

Задача № 2: условия (25) -- (27) + прямые ограничения

$$3 \leq x_1 \leq 5, 0 \leq x_2 \leq 5, 0 \leq x_3 \leq 5.$$

Задача № 3: условия (25) -- (27) + прямые ограничения

$$0 \leq x_1 \leq 2, 0 \leq x_2 \leq 5, 0 \leq x_3 \leq 5.$$

Полагаем $r_0^2 = r_0^1 = 0$ и переходим к шагу 1, начиная новую итерацию.

Итерация 2. Список состоит из задач № 2, 3.

Из списка задач рассмотрим задачу № 2. Легко увидеть, что ограничения этой задачи несовместны. Значит, мы вычёркиваем эту задачу из списка, полагаем $r_0^3 = r_0^2 = 0$ и переходим к новой итерации.

Итерация 3. Теперь список состоит только из задачи № 3. Рассмотрим эту задачу. Она имеет решение

$$x_1^* = x_2^* = 2,\ x_3^* = \frac{2}{7},\ c'x^* = 15\frac{5}{7}. \text{ (задача № 3).}$$

Переменная x_3^* -- нецелая. Выбираем её для ветвления и идём на шаг 4. Удаляем задачу № 3 из списка, вместо неё включаем новые две задачи №~4, № 5, порождённые задачей № 3.

Задача № 4: условия (25) -- (27) + прямые ограничения

$$0 \leq x_1 \leq 2, 0 \leq x_2 \leq 5, 1 \leq x_3 \leq 5.$$

Задача № 5: условия (25) -- (27) + прямые ограничения

$$0 \leq x_1 \leq 2, 0 \leq x_2 \leq 5, x_3 = 0.$$

Полагаем $r_0^4 = r_0^3 = 0$ и переходим к следующей итерации.

Итерация 4. Список состоит из задач № 4, № 5. Рассмотрим задачу № 4. Она имеет решение

$$x_1^* = x_2^* = \frac{1}{3},\ x_3^* = 1,\quad c'x^* = 15. \text{ (задача № 4).}$$

Переходим к шагу 4, выбрав переменную x_2 для ветвления. Задачу № 4 вычёркиваем из списка, вместо неё включаем в список две новые задачи № 6, № 7, порождённые задачей № 4.

Задача № 6: условия (25) -- (27) + прямые ограничения

$$0 \leq x_1 \leq 2, 1 \leq x_2 \leq 5, 1 \leq x_3 \leq 5.$$

Задача № 7: условия (25) -- (27) + прямые ограничения

$$0 \leq x_1 \leq 2, x_2 = 0, 1 \leq x_3 \leq 5.$$

Полагаем $r_0^5 = r_0^4 = 0$ и переходим к новой итерации.

Итерация 5. На данной итерации из списка задач № 5, № 6, № 7 выберем задачу № 6. Анализируя условия задачи № 6, легко заметить, что она не имеет допустимых планов, поэтому вычёркиваем её из списка. Полагаем $r_0^6 = r_0^5 = 0$ и переходим к новой итерации, возвращаясь к шагу 1.

Итерация 6. Теперь список состоит из задач № 5, № 7. Выберем задачу № 7. Она имеет решение

$$x_1^* = x_2^* = 0, \ x_3^* = 1\frac{1}{7}, \ c'x^* = 14\frac{6}{7}. \text{ (задача № 7).}$$

Переходим к шагу 4, используя x_3 для ветвления. Задачу № 7 исключаем из списка, вместо неё включаем две новые задачи № 8, № 9, порождённые задачей № 7.

Задача № 8: условия (25) -- (27) + прямые ограничения

$$0 \leq x_1 \leq 2, x_2 = 0, 2 \leq x_3 \leq 5.$$

Задача № 9: условия (25) -- (27) + прямые ограничения

$$0 \leq x_1 \leq 2, x_2 = 0, x_3 = 1.$$

Полагаем $r_0^7 = r_0^6 = 0$ и переходим к следующей итерации.

Итерация 7. Список состоит из задач № 5, 8, 9. Выберем задачу № 8. Она не имеет решения, так как её ограничения несовместны. Вычёркиваем её из списка. Полагаем $r_0^8 = r_0^7 = 0$ и переходим к новой итерации.

Итерация 8. Список состоит из задач № 5, 9. Выберем задачу № 9. Эта задача имеет решение

$$x_1^* = x_2^* = 0, \ x_3^* = 1, \ \ c'x^* = 13. \text{ (задача № 9).}$$

Решение задачи № 9 -- целое, поэтому полагаем $\mu_0 = 1, \ \mu = x^*$. Задачу № 9 вычёркиваем из списка, полагаем $r_0^9 = r_0^8 = 13$ и переходим к новой итерации.

Итерация 9. Список состоит только из одной задачи № 5. Задача № 5 имеет решение

$$x_1^* = 2, \ x_2^* = 2^{\frac{1}{3}}, \ x_3^* = 0, \ c'x^* = 13. \text{ (задача № 5).}$$

Поскольку $c'x^* = 13 \leq r_0^9 = 13$, то мы не «дробим» задачу № 5, а просто вычёркиваем её из списка. Оценку r_0^9 не меняем: $r_0^{10} = r_0^9 = 13$.

Переходим к шагу 1 новой итерации.

Итерация 10. На новой итерации список пуст. Алгоритм заканчивает работу. Так как у нас $\mu_0 = 1$, то вектор μ принимаем за решение задачи:

$$\mu = x^0 = (0, 0, 1).$$

Задача решена.

Ход решение задачи схематично можно представить в виде дерева (см. рис. 1.2).

Замечания:

1. В описаном выше алгоритме возможность произвольного выбора возникает в двух местах:

    - при выборе задачи из списка;
    - при выборе нецелой переменной, по которой производится ветвление.

    Число итераций метода зависит от того, какой именно выбор мы осуществим в этих пунктах.

    К настоящему времени разработаны специальные процедуры, помогающие оценить "качество" выбора. Это повышает эффективность алгоритма.

2. На любой итерации (кроме первой) мы имеем список задач, которые надо решить, и которые отличаются от породившей их задачи (решение которой уже найдено) **только одним** прямым ограничением. Для эффективного решения задач из списка можно использовать двойственный симплекс-метод [2, 9], выбирая в качестве начального двойственного базисного плана оптимальный двойственный план порождающей задачи.

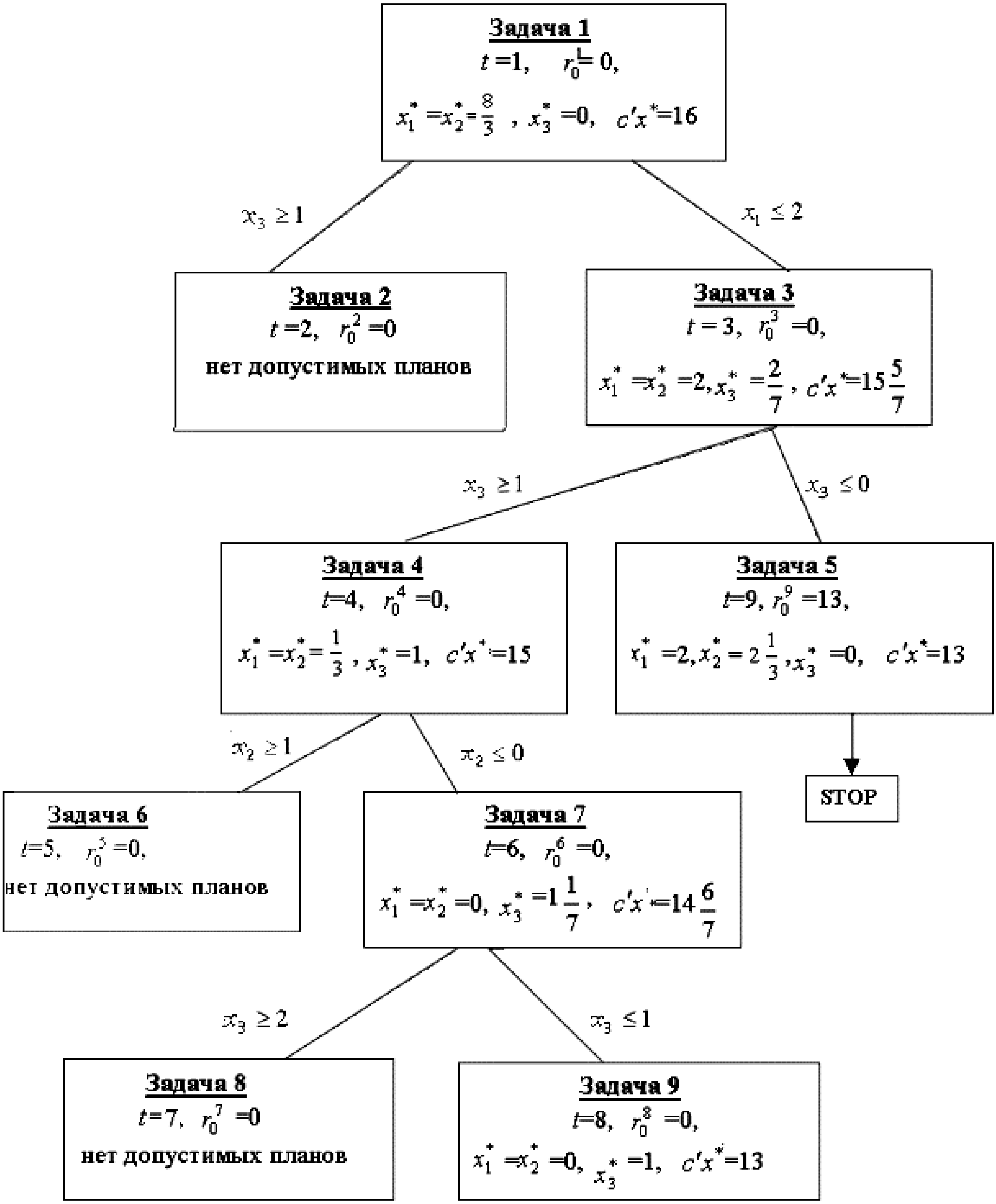

Рис. 1.2

1.2.4 Двойственный симплекс-метод для задачи линейного программирования с двусторонними ограничениями

Рассмотрим задачу линейного программирования следующего вида

$$\begin{gathered} c'x \to \max \\ Ax = b, \\ d_* \leq x \leq d^* \end{gathered} \tag{30}$$

где $A \in R^{m\times n}$ $c,\ d_*,\ d^* \in R^n;\ b \in R^m;\ \ c = (c_j,\ j = \overline{1,\ n}),\ d_* = (d_{*j},\ j = \overline{1,n}),\ \ d^* = (d^*_j, j = \overline{1,n}),$ $A = (A_j, j = \overline{1,n}),\ \ A_j \in R^m$ -- заданные параметры задачи. Без ограничения общности будем считать, что $rank\ A = m.$

Подмножество индексов $J_B = \{j_1, j_2, ..., j_m\} \subset J = \{1,\ 2,\ ...,\ n\}$ назовем базисом задачи (30), если $\det\ A_B \neq 0$, где $A_B = (A_j,\ j \in J_B)$ -- базисная матрица.

Решение задачи (30) двойственным методом начинается с задания некоторого базиса $J_B = \{j_1, j_2, ..., j_m\}$ и соответствующей ему базисной матрицы $A_B = (A_j, j \in J_B)$. Матрицу, обратную к базисной, обозначим через B : $B = A_B^{-1}$.

Итерация метода состоит из следующих шагов.

1. Найдем *m*-вектор

$$y' := c'_B B$$

и оценки

$$\Delta_j := y' A_j - c_j, j \in J;$$

сформируем множества

$$J_H = J \backslash J_B;\ \ J_H^+ = \{j \in J_H:\ \ \Delta_j \geq 0\};\ J_H = J_H \backslash J_H^+.$$

2. Построим вектор $\aleph = (\aleph_j, j \in J)$ по следующему правилу:

$$\aleph_j = d_{*j}, j \in J_H^+;\ \ \aleph_j = d_j^*, j \in J_H^-;$$

$$\aleph = (\aleph_j,\ j \in J) = B(b - \sum_{j \in J_H^+ \cup J_H^-} A_j \aleph_j).$$

3. Проверим критерий оптимальности: если выполняются соотношения

$$d_{*j} \leq \aleph_j \leq d_j^*,\ j \in J, \tag{31}$$

то вектор $x^0 := \aleph = (\aleph_j, j \in J)$ -- оптимальный план задачи (30). Решение задачи (30) прекращается. В противном случае (т.е. если соотношения (31) не выполняются) идем на шаг 4.

4. Найдем такой индекс $j_k \in J$, что $\aleph_{j_k} \notin [d_{*j_k}, d^*_{j_k}]$.

5. Положим

$$\mu_{j_k} = 1\text{, если } \aleph_{j_k} < d_{*j_k},$$

$$\mu_{j_k} = -1\text{, если } \aleph_{j_k} > d^*_{j_k}.$$

Подсчитаем *m*-вектор

$$\Delta y' = \mu_{j_k} e'_k B$$

и числа

$$\mu_j = \Delta y' A_j,\ j \in J_H^+ \cup J_H^-.$$

6. Найдем шаги $\sigma_j,\ j \in J_H = J_H^+ \cup J_H^-$ по правилу:

\sigma _{j} =\left\{\begin{array}{l} -\Delta _{j} /\mu _{j} ,\, \, \,\mbox{ if } j\in J_{H}^{+} \, \mbox{ and } \mu _{j} < 0\, \, \mbox{ or } \, \, j\in J_{H}^{-} \, \, \mbox{ and } \mu _{j} > 0\, \\ \\ \infty \, \, \mbox{ otherwise} \end{array}\right.

Положим $\sigma_0 = \min\limits_{j \in J_H} \sigma_j = \sigma_{j_*}$, здесь $j_* \in J_H$ -- индекс, на котором достигается минимум в последнем выражении. Если минимум достигается на нескольких индексах, то в качестве индекса j_* можно взять любой из них.

7. Если $\sigma_0 = \infty$, то прекращаем решение задачи (30), так как она не имеет допустимых планов. Если $\sigma_0 < \infty$, идем на шаг 8.

8. Построим новый коплан $\bar{\Delta} = (\bar{\Delta}_j, j \in J)$ по правилу:

$$\bar{\Delta}_j = \Delta_j + \sigma_0 \mu_j,\ j \in J_H \cup j_k;$$

$$\bar{\Delta}_j = 0, \quad j \in J_B \backslash j_k.$$

9. Построим новый базис $\bar{J}_B = (J_{\backslash} j_k) \cup j_*$, соответствующую ему базисную матрицу $\bar{A}_B = (A_j, j \in \bar{J}_B)$ и обратную матрицу $\bar{B} = \bar{A}_B^{-1}$ по правилу $\bar{B} = MB$, где $m \times m-$ матрица M получается из единичной $m \times m-$ матрицы заменой *k*-го столбца e_k на столбец d :

$$e_k = \begin{pmatrix} 0 \\ \dots \\ 0 \\ 1 \\ 0 \\ \dots \\ 0 \end{pmatrix}, \qquad d = -\begin{pmatrix} z_1 \\ \dots \\ z_{k-1} \\ -1 \\ z_{k+1} \\ \dots \\ z_m \end{pmatrix} \cdot \frac{1}{z_k}, z = \begin{pmatrix} z_1 \\ \dots \\ z_{k-1} \\ z_k \\ z_{k+1} \\ \dots \\ z_m \end{pmatrix} = BA_{j_*}.$$

10. Построим новые множества $\bar{J}_H,\ \bar{J}_H^-$ и $\bar{J}_H^+$:

$$\bar{J}_H = J \backslash \bar{J};$$

$\bar{J}_H^+ = (J_H^+ \backslash j_*) \cup j_k$, если $\mu_{j_k} = 1,\ j_* \in J_H^+$;
$\bar{J}_H^+ = (J_H^+ \backslash j_*)$, если $\mu_{j_k} = -1,\ j_* \in J_H^+$;
$\bar{J}_H^+ = (J_H^+ \cup j_k)$, если $\mu_{j_k} = 1,\ j_* \notin J_H^+$;
$\bar{J}_H^+ = J_H^+$, если $\mu_{j_k} = -1,\ j_* \notin J_H^+$;

$$\bar{J}_H^- = \bar{J}_H \backslash \bar{J}_H^+.$$

Идем на шаг 2, используя новые базис $\bar{J}$, коплан $\bar{\Delta}$, базисную матрицу $\bar{A}$ и обратную к ней матрицу $\bar{B}$.

Совокупность шагов 2 -- 10 назовем итерацией $J_B \to \bar{J}_B$. Данная итерация называется *невырожденной*, если $\sigma_0 > 0$.

Можно показать, что описанный алгоритм решает задачу за конечное число итераций, если в процессе его реализации встречается конечное число вырожденных итераций.