

УДК 519.717

**ТЕТРИС-ЭЛИМИНАТОР И АЛГОРИТМ  
ПОСЛЕДОВАТЕЛЬНОГО УТОЧНЕНИЯ ЛЮФТА В  
ВЕКТОРЕ ТРЕБОВАНИЙ**

*В.Л.Никитенков, А.В.Тюфяков*

Для решения целочисленной задачи линейного раскроя предложены алгоритмы, позволяющие получить оптимальное решение либо для исходного вектора требований, либо при наличии минимального люфта. Обсуждаются вопросы вырожденности полученных решений и результаты численных экспериментов.

## **1. Введение**

Целочисленная задача линейного раскроя<sup>[1]</sup> возникает, например, в несерийном производстве, когда количество изделий, на которые идут заготовки задано<sup>[6]</sup>. Существуют эффективные алгоритмы решения этой задачи и процедуры их реализации, основанные на методе дихотомии<sup>[3-5]</sup>. При решении задач линейного раскроя (обычных и целочисленных) было замечено, что оптимальные значения целевых функций в линейной и целочисленной задачах отличаются менее чем на единицу. Если это верно, то целочисленная задача оказывается "почти" линейной, и к ней можно применить комбинацию модифицированного симплекс-метода<sup>[2,6]</sup> (на первом этапе) и процедуры<sup>[4]</sup> (для решения очень небольшой по объему задачи). В связи с решением задачи форматного раскроя возникла задача минимизации числа перенастроек продольно-резательных станков (ПРС), которая сводится к поиску наиболее вырожденного решения ЦЗЛР. Для последней задачи был предложен алгоритм, основанный на переборе векторов-"кандидатов" в оптимальный базис<sup>[8]</sup>. Ниже предлагается алгоритм последовательного изменения вектора требований на заготовки с целью его минимальной корректировки для получения оптимального решения ЦЗЛР, являющегося, как правило, вырожденным.

## 2. Целочисленная задача линейного раскроя

Рассмотрим задачу раскроя одномерного сырья (прутьев, бумажных или картонных тамбуров, профильного материала, характеризующегося своей длиной) длины на заготовки (рулоны заданного формата, отрезки профиля заданной длины), длины которых задаются вектором

$$l[M] = (l_1, \dots, l_m); |M| = m$$

Вектор требований на заготовки имеет вид

$$b[M] = (b_1, \dots, b_m)$$

Требуется минимизировать число использованных штук сырья при условии выполнения требований на заготовки каждого типа. Будем называть способом раскроя целочисленный вектор. Требуется минимизировать число использованных штук сырья при условии выполнения требований на заготовки каждого типа. Будем называть способом раскроя целочисленный вектор  $R[M]$ , т. что

$$l[M] \cdot R[M] \leq L \quad (2.1)$$

Из всех возможных способов раскроя составим неотрицательную целочисленную матрицу  $A[M, N]$  ( $|N| = n$  - число всех возможных способов раскроя) в которой каждый столбец удовлетворяет условию (2.1). Пусть теперь  $x[N]$ - вектор интенсивностей использования способов раскроя  $A[M, j]$  (способ раскроя используется  $x_j$  раз, или по указанному способу раскраивается штук сырья). Тогда приходим к следующей задаче ЛП:

$$\begin{cases} f(x) = 1[N] \cdot x[N] \rightarrow \min \\ A[M, N] \cdot x[N] = b[M] \\ x[N] \geq 0[N] \end{cases} \quad (2.2)$$

в которой столбцы матрицы системы ограничений удовлетворяют условию (2.1). Эта задача ЛП называется задачей линейного раскроя. Задача (2.2) с дополнительным условием:

$$x[N]\text{-целочисленный вектор}$$

носит название целочисленной задачи **линейного раскроя (ЦЗЛР)**.

Задача линейного раскроя решается модифицированным алгоритмом симплекс-метода, причем матрица  $A[M, N]$  не хранится (с ростом числа типов заготовок  $m$  и кратностей - числа заготовок в способах раскроя, число её столбцов растёт весьма быстро (см. рис. 1)), а необходимый для ввода в базис способ раскроя генерируется путем решения задачи о наилучшем раскрое<sup>[2]</sup>, которая представляет собой целочисленную задачу о ранце с двойственными переменными в качестве коэффициентов целевой функции и ограничениями (2.1):

$$\begin{cases} y[M] \cdot R[M] \rightarrow \max \\ l[M] \cdot R[M] \leq L \end{cases} \quad (2.3)$$

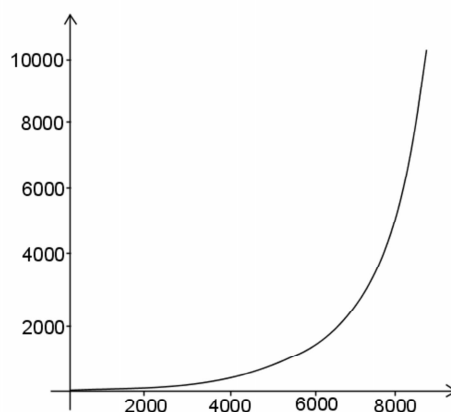


Рис. 1

### 3. Комбинированный алгоритм решения ЦЗЛР

Пусть теперь  $\tilde{x}[N]$  - оптимальное решение линейной задачи раскроя, т.е.

$$\begin{aligned} f(\tilde{x}) &= 1[N] \cdot \tilde{x}[N] = K \\ A[M, N] \cdot \tilde{x}[N] &= b[M] \\ \tilde{x}[N] &\geq 0[N] \end{aligned} \quad (3.4)$$

(если  $K$  - нецелое, можно сделать одно отсечение по целевой функции)

Выделим целую  $[\tilde{x}[N]]$  и дробную  $\{\tilde{x}[N]\}$  части вектора  $\tilde{x}[N]$  и, подставляя в (3.4), имеем

$$\begin{aligned} \mathbf{1}[N] \cdot [\tilde{x}[N]] + \mathbf{1}[N] \cdot \{\tilde{x}[N]\} &= K \\ A[M, N] \cdot [\tilde{x}[N]] + A[M, N] \cdot \{\tilde{x}[N]\} &= b[M] \\ [\tilde{x}[N]] &\geq \mathbf{0}[N] \text{- целочисленный} \\ \mathbf{1}[N] &> \{\tilde{x}[N]\} \geq \mathbf{0}[N] \end{aligned} \quad (3.5)$$

Или отдельно для дробной части решения

$$\begin{aligned} \mathbf{1}[N] \cdot \{\tilde{x}[N]\} &= K - \mathbf{1}[N] \cdot [\tilde{x}[N]] = k \\ A[M, N] \cdot \{\tilde{x}[N]\} &= b[M] - A[M, N] \cdot [\tilde{x}[N]] = \tilde{b}[M] \\ \mathbf{1}[N] &> \{\tilde{x}[N]\} \geq \mathbf{0}[N] \\ k &\text{ - целое} \end{aligned} \quad (3.6)$$

Целочисленный вектор  $\tilde{b}[M]$  будем называть остатком вектора требований. Формулы (3.6)<sub>2,3</sub> означают, что данный остаток является конической комбинацией способов раскрыя. С учетом (3.6)<sub>1</sub>, умножая (3.6)<sub>2</sub> на  $l[M]$ , получим

$$l[M] \cdot \tilde{b}[M] \leq kL \quad (3.7)$$

Разбивая (одним из комбинаторных алгоритмов, например<sup>[4]</sup>) полученный остаток вектора требований  $\tilde{b}[M]$  на наименьшее число раскрыев, получим

$$\begin{aligned} \sum_{k=1}^s A[M, J_k] \cdot \bar{x}[J_k] &= \tilde{b}[M], \quad J_k \in \bar{N} \subseteq N \\ \sum_{k=1}^s \bar{x}[J_k] &= k \text{ - целое} \end{aligned}$$

Тогда целочисленный вектор  $x^*[N] = [\tilde{x}[N]] + (\bar{x}[\bar{N}], 0[N \setminus \bar{N}])$  будет оптимальным решением ЦЗЛР.

#### 4. Проблема вырожденности оптимального решения ЦЗЛР

При форматном раскрые бумажного полотна на рулоны имеет место проблема уменьшения числа перенастроек продольно резательных станков (ПРС). Это приводит к требованию поиска такого оптимального решения ЦЗЛР, которое содержало бы в оптимальном базисе как

можно меньше способов раскрыя с положительными целочисленными интенсивностями, т.е. было бы максимально вырожденным. Идея поиска такого решения среди базисных решений ЦЗЛР с дополнительным ограничением на значение целевой функции привела к созданию алгоритма замены "кандидатов" в оптимальный базис - векторов, имеющих в оптимальном решении нулевую оценку<sup>[8]</sup>. Однако, при достаточно большом числе форматов, перебор всех возможных "кандидатов" для ввода в базис становится затруднительным, и приходится ограничиваться некоторым приближением (в плане вырожденности) к оптимальному решению. Попытки использовать для этих целей отклонения в векторе требований (люфт) приводят опять же к переборным алгоритмам.

## 5. Алгоритм последовательного изменения люфта

Пусть  $x_0[N]$  - оптимальное решение линейной задачи раскрыя

$$A \cdot x_0 = b_0, \quad f(x_0) = 1[N] \cdot x_0[N] = f_{min}$$

$b_0$  - первоначальный вектор требований,

$A$  - оптимальная базисная матрица.

Найдем решение  $x^*$ , близкое к  $x_0$ , которое дает целочисленное решение с минимально возможным люфтом в исходном векторе требований, следуя алгоритму:

1.  $k = 0, b_k = b_0$
2.  $Ax_k = b_k$
3.  $\alpha_k = b_0 - A \cdot [x_k]$  (\*)
4.  $b_{k+1} = b_k + \alpha_k$  (\*\*)
5.  $k = k + 1, goto 1$

Попробуем упростить приведенную схему. Заметим, что из (\*) следует

$$\alpha_k = b_0 - (b_k + A \cdot \{x_k\})$$

Далее, используя (\*\*), рекуррентно, получим

$$b_{k+1} = b_0 + \alpha_0 + \alpha_1 + \dots + \alpha_k$$

Очевидно, что

$$\begin{aligned}
 \alpha_0 &= A \cdot \{x_0\} \\
 \alpha_1 &= b_0 - (b_1 - A \cdot \{x_1\}) = \\
 &= b_0 - (b_0 + \alpha_0 - A \cdot \{x_1\}) = A \cdot \{x_1\} - A \cdot \{x_0\} \\
 \alpha_2 &= b_1 - (b_2 - A \cdot \{x_2\}) = \\
 &= b_0 - (b_0 + \alpha_0 + \alpha_1 - A \cdot \{x_2\}) = A \cdot \{x_2\} - A \cdot \{x_1\} \\
 &\dots \\
 \alpha_k &= A \cdot \{x_k\} - A \cdot \{x_{k-1}\}
 \end{aligned}$$

и для суммы находим

$$\alpha_0 + \alpha_1 + \dots + \alpha_k = A \cdot \{x_{k-1}\}$$

Тогда алгоритм может быть записан в виде

$$A \cdot x_k = b_0 + A \cdot \{x_k\}, \quad k = 0, 1, 2, \dots \quad (5.8)$$

Покажем теперь, что

$$\{x_k\} = \{(k+1)x_0\} \quad (5.9)$$

Для  $k = 1$  имеем (умножая (5.8) на обратную базисную матрицу  $B$ )

$$\begin{aligned}
 x_1 &= x_0 + \{x_0\} \\
 \{x_1\} &= \{[x_0] + \{x_0\} + \{x_0\}\}
 \end{aligned}$$

Предположим, что формула (5.9) верна для произвольного фиксированного  $k = n$ . Докажем её справедливость для  $k = n + 1$ .

$$\begin{aligned}
 x_{n+1} &= x_0 + \{x_n\} = x_0 + \{(n+1)\{x_0\}\} \\
 \{x_{n+1}\} &= \{[x_0] + \{x_0\} + (n+1)\{x_0\} - [(n+1)\{x_0\}]\} = \\
 &= \{\{x_0\} + (n+1)\{x_0\}\} = \\
 &= \{(n+2)\{x_0\}\}
 \end{aligned}$$

Утверждение доказано. Далее, подставляя (5.9) в (5.8) и умножая последнее равенство на  $B$ , запишем алгоритм в окончательном виде

$$\begin{cases} A \cdot x_0 = b_0 \\ x_n = \{n\{x_0\}\}, \quad n = 1, 2, \dots \end{cases} \quad (5.10)$$

Графическая иллюстрация работы алгоритма показана на Рис. 2.

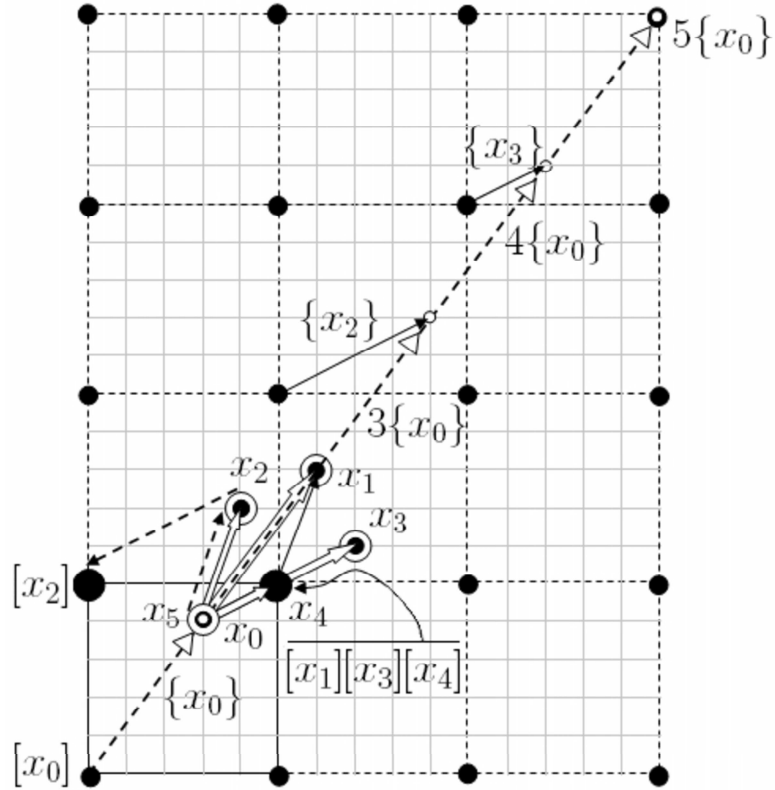


Рис. 2

Непосредственно из (5.10) следует:

1. Последовательность решений зацикливается через  $N$  шагов, где  $N$  - наименьшее общее кратное знаменателей дробной части оптимального нецелочисленного решения  $x_0$ ;
2. В этой последовательности есть решения как с суммарным люфтом вверх, так и вниз;
3. Среди них можно выбрать решение с наименьшим люфтом:

$$A \cdot [x_p] = b_0 + (A \cdot \{x_p\} - A \cdot \{x_{p-1}\})$$

Имеет место также утверждение о том, что оптимальная базисная матрица не изменяется по ходу алгоритма. Последняя не зависит от вектора требований и будет давать оптимальное решение при условии его допустимости. В силу неотрицательности дробной части, непосредственно из (5.8) и (5.10) следует, что

$$x_n \geq x_0 \geq 0 \text{ и } A \cdot x_n = b_0 + A \cdot \{x_{n-1}\}$$

т.е. решение  $x_n$  допустимо для вектора требований  $b_0 + A \cdot \{x_{n-1}\}$

Продемонстрируем работу алгоритма на примере, соответствующем Рис. 2.:

$$\begin{aligned} L &= 40, l = (4, 2), b = (36, 96) \\ x_0 &= (3.6, 4.8), \{x_0\} = (0.6, 0.8), N = 5 \\ \text{Оптимальная базисная матрица } A &= \text{diag}(10, 20) \end{aligned}$$

(Остаток вектора требований  $A \cdot \{x_0\} = (6, 16)$  делится на два раскроя: (2, 16)- безотходный и (4, 0) - отход 24. Суммарный отход - 24).

Далее последовательно получаем:

$$\begin{aligned} x_1 &= x_0 + \{x_0\} = (3.6, 4.8) + (0.6, 0.8) = (4.2, 5.6) \\ x_2 &= x_0 + \{2x_0\} = (3.6, 4.8) + (0.2, 0.6) = (3.8, 5.4) \\ x_3 &= x_0 + \{3x_0\} = (3.6, 4.8) + (0.8, 0.4) = (4.4, 5.2) \\ x_4 &= x_0 + \{4x_0\} = (3.6, 4.8) + (0.4, 0.2) = (4, 5) \\ x_5 &= x_0 + \{5x_0\} = (3.6, 4.8) = x_0 - \text{зацикливание} \end{aligned}$$

Суммарный отход во всех решениях равен нулю. Однако, тот факт, что используемая в алгоритме исходная оптимальная базисная матрица не меняется (при наличие нескольких оптимальных базисных матриц) может приводить к ситуации, когда оптимальное целочисленное решение не будет найдено. Это иллюстрируется следующим примером:

$$\begin{aligned} L &= 12, l = (6, 4, 3, 2), b = (10, 40, 60, 40) \\ x_0 &= (5, \frac{40}{3}, 15, \frac{20}{3}), \{x_0\} = (0, \frac{1}{3}, 0, \frac{2}{3}), N = 3, f_{min} = 40 \end{aligned}$$

$$\text{Оптимальная базисная матрица } A = \text{diag}(2, 3, 4, 6)$$

Применяя алгоритм, находим цепочку из трех решени ( $N = 3$ )

$$\begin{aligned} x_0 &= (5, \frac{40}{3}, 15, \frac{20}{3}), [x_0] = (5, 13, 15, 6), \text{люфт } (0, -1, 0, -4), f_{min} = 39 \\ x_1 &= (5, \frac{41}{3}, 15, \frac{22}{3}), [x_0] = (5, 13, 15, 7), \text{люфт } (0, -1, 0, 2), f_{min} = 40 \\ x_2 &= (5, 14, 15, 8), [x_0] = (5, 14, 15, 8), \text{люфт } (0, 2, 0, 2), f_{min} = 41 \end{aligned}$$



Однако, при  $f_{min} = 40$  имеется оптимальное целочисленное вырожденное решение без люфта

$$10 \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} + 30 \begin{bmatrix} 0 \\ 1 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 10 \\ 40 \\ 60 \\ 40 \end{bmatrix}$$

Таким образом, заключаем, что эффективная работа алгоритма существенно зависит от вида исходного оптимального решения линейной задачи. Идея использования элиминации вектора требований для поиска наиболее вырожденного оптимального решения (или хорошего приближения к нему) может оказаться здесь плодотворной.

### 6. "Жадный" алгоритм решения ЦЗЛР (тетрис-элиминатор)

Представим себе некий аналог известной компьютерной игры "тетрис" с несколько видоизмененными правилами. Пусть коробка заполнена столбцами различной высоты, как показано на рис.3.

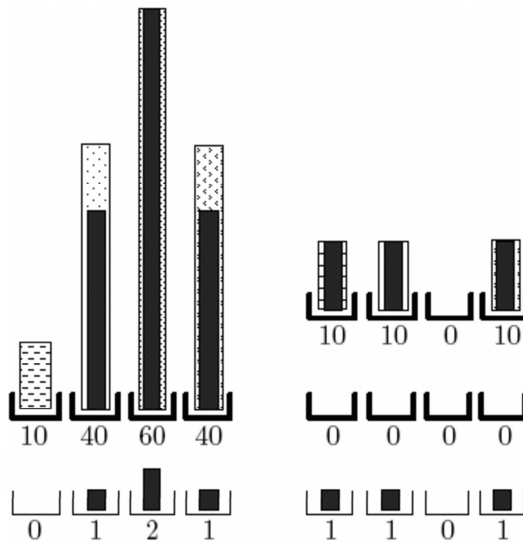


Рис. 3

Имеется набор комбинаций столбцов меньшей высоты, образующих фигуры, две из которых приведены там же на рис. 3. ниже.

Требуется с помощью указанных комбинаций, каждая из которых используется с некоторой целочисленной кратностью (см. рис. 3), исчерпать коробку, используя минимальное количество комбинаций, и с наименьшей суммарной кратностью.

Столбец в коробке можно рассматривать как компоненту вектора требований на форматы. Столбец в комбинации - это количество рулонов данного формата в раскладке. Таким образом, если мы используем комбинации с наименьшей суммарной кратностью, то получим минимальное число съёмов. А если используем минимальное число комбинаций, то минимизируем число различных используемых раскладок.

Алгоритм решения:

1. Исходная базисная матрица  $A[M, N']$ :

$$A[M, j] = e_j \cdot [L/l[j]], \quad j \in 1 : m$$

2. Обратная базисная матрица  $B[N', M]$ :

$$B[N', j] = e_j \cdot [L/l[j]]^{-1}, \quad j \in 1 : m$$

3. Вектор двойственных переменных  $v[M]$ :

$$v[i] = [L/l[j]]^{-1}, \quad j \in 1 : m$$

4.  $r$  - задача

$$\left. \begin{array}{l} v[M] \cdot r[M] \rightarrow \max \\ r[M] \cdot l[M] \leq L \end{array} \right\} \Rightarrow r[M] = A[M, j_0]$$

$A[M, j_0]$  исключаем из дальнейшего рассмотрения

5. Текущее базисное решение  $x[N'] = B[N', M] \cdot b[M]$ :

$$x[i] = [L/l[i]]^{-1} \cdot b[i], \quad i \in 1 : m$$

6. Текущий вектор направления улучшения решения  $z[N']$ :

$$z[N'] = B[N', M] \cdot r[M]$$

$$z[i] = [L/l[i]]^{-1} \cdot r[i] \geq 0, \quad i \in 1 : m$$

7. Значение новой базисной переменной

$$x[j_0] = [\Theta] : \quad \Theta = \min \left\{ \frac{x[i]}{z[i]} = \frac{b[i]}{r[i]} : r[i] > 0 \right\} = \frac{b[i_0]}{r[i_0]}$$

(В определении  $\Theta$  участвует и отношение  $b[m+1]/r[m+1]$ ,  
 $b[m+1]$  - суммарный отход,  $r[m+1]$  - отход раскроя)

Если  $x[j_0] = 0$ , то формат  $i_0$  исключаем из рассмотрения и переходим к шагу 4.

8.  $b_1[M] = b[M] - x[j_0] \cdot A[M, j_0]$ ,  $b[M] := b_1[M]$   
 if  $b_1[M] \cdot l[M] \leq L$  then *END* else *GOTO 4*

■ На шаге 4 можно находить всех кандидатов для ввода в базис. Если, при этом, в векторе двойственных переменных убрать целую часть, то кандидатами для ввода в базис будут наименее отходные способы раскроя ■

Рассмотрим пример

$$L = 13, l = (7, 5, 3), b = (29, 31, 11)$$

$$b[M] \cdot l[M] = 30 \cdot L + 1, (\text{min возможный отход} - 12)$$

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 2 & 2 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 1 & 0 & 2 & 1 & 0 & 4 & 3 & 2 & 1 \\ 1 & 0 & 3 & 6 & 0 & 3 & 1 & 5 & 8 & 1 & 4 & 7 & 10 \end{bmatrix}$$

Задача имеет оптимальное базисное целочисленное решение

$$25 \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}_{(1)} + 4 \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix}_{(0)} + 3 \begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix}_{(0)} = \begin{bmatrix} 29 \\ 31 \\ 11 \end{bmatrix}$$

32 раскладки, отход - 25.

Возможные значения отхода:

1,  $L - 1 = 12$ ,  $2L - 1 = 25$ ,  $3L - 1 = 38$  и т. д.

$v = (1/1, 1/2, 1/4)$

Кандидаты для ввода в базис:

*отх.*  
 $(1, 1, 0) | 1 - 1, 5$   
 $(1, 0, 2) | 0 - 1, 5$   
 $(0, 2, 1) | 0 - 1, 25$

I. Начнем с отхода,  $b[m+1] = 1$

Верём  $(1, 1, 0|1)$ ,  $\Theta = \min \left\{ \frac{29}{1}, \frac{1}{1} \right\} = 1$   
ОТХ.

$$1 \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, b_1 = \begin{bmatrix} 29 \\ 31 \\ 11 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 28 \\ 30 \\ 11 \end{bmatrix}$$

Остальные раскрои должны быть безотходными.

$$5 \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 5 \\ 0 \\ 10 \end{bmatrix}, b_1 = \begin{bmatrix} 28 \\ 30 \\ 11 \end{bmatrix} - \begin{bmatrix} 5 \\ 0 \\ 10 \end{bmatrix} = \begin{bmatrix} 23 \\ 30 \\ 1 \end{bmatrix}$$

$$1 \begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix}, b_1 = \begin{bmatrix} 23 \\ 30 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 23 \\ 28 \\ 0 \end{bmatrix}$$

Возможно использование только отходных раскroев  
 Решения нет.

II.  $b[m+1] = L - 1 = 12$

$$12 \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 12 \\ 12 \\ 0 \end{bmatrix}, b_1 = \begin{bmatrix} 29 \\ 31 \\ 11 \end{bmatrix} - \begin{bmatrix} 12 \\ 12 \\ 0 \end{bmatrix} = \begin{bmatrix} 17 \\ 19 \\ 11 \end{bmatrix}$$

Остальные раскрои должны быть безотходными

$$5 \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 5 \\ 0 \\ 10 \end{bmatrix}, b_1 = \begin{bmatrix} 17 \\ 19 \\ 11 \end{bmatrix} - \begin{bmatrix} 5 \\ 0 \\ 10 \end{bmatrix} = \begin{bmatrix} 12 \\ 19 \\ 1 \end{bmatrix}$$

$$1 \begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix}, b_1 = \begin{bmatrix} 12 \\ 19 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 12 \\ 17 \\ 0 \end{bmatrix}$$

Решения нет.

III.  $b[m+1] = 25$

$$25 \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 25 \\ 25 \\ 0 \end{bmatrix}, b_1 = \begin{bmatrix} 29 \\ 31 \\ 11 \end{bmatrix} - \begin{bmatrix} 25 \\ 25 \\ 0 \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \\ 11 \end{bmatrix}$$

$$4 \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 4 \\ 0 \\ 8 \end{bmatrix}, b_1 = \begin{bmatrix} 4 \\ 6 \\ 11 \end{bmatrix} - \begin{bmatrix} 4 \\ 0 \\ 8 \end{bmatrix} = \begin{bmatrix} 0 \\ 6 \\ 3 \end{bmatrix} = 3 \begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix}$$

Найдено оптимальное целочисленное решение

Следует заметить, что тетрис-элиминатор не всегда дает самое вы-  
рожденное решение, что подтверждается следующим примером:

$$L = 35, l = (9, 5, 3), b = (101, 101, 102)$$

Матрица  $A$  будет иметь следующий вид:

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ 3 & 3 & 2 & 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 3 & 2 & 1 & 0 & 5 & 4 & 3 & 2 & 1 & 0 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ 1 & 2 & 0 & 2 & 4 & 5 & 0 & 2 & 3 & 5 & 7 & 8 & 0 & 1 & 3 & 5 & 6 & 8 & 10 & 11 \\ 0 & 2 & 2 & 1 & 0 & 2 & 1 & 0 & 2 & 1 & 0 & 2 & 0 & 2 & 1 & 0 & 2 & 1 & 0 & 2 \end{bmatrix}$$

$$b[M] \cdot l[M] = 49 \cdot L + 5$$

минимально возможный отход — 30.

Задача имеет оптимальное целочисленное решение

$$20 \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix}_{(0)} + 17 \begin{bmatrix} 1 \\ 4 \\ 2 \end{bmatrix}_{(0)} + 12 \begin{bmatrix} 2 \\ 1 \\ 4 \end{bmatrix}_{(0)} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}_{(30)} = \begin{bmatrix} 101 \\ 101 \\ 102 \end{bmatrix}$$

50 съёмов, 4 раскладки, отход — 30.

Кандидаты в оптимальный базис:  $v = (1/3, 1/7, 1/11)$

$$(3, 1, 1) - 285/231$$

$$(2, 1, 4) - 271/231$$

$$(1, 1, 7) - 257/231$$

$$(1, 4, 2) - 251/231$$

$$(0, 1, 10) - 243/231$$

$$(0, 4, 5) - 235/231$$

$$1) 33 \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 99 \\ 33 \\ 33 \end{bmatrix}, b_1 = \begin{bmatrix} 101 \\ 101 \\ 102 \end{bmatrix} - \begin{bmatrix} 99 \\ 33 \\ 33 \end{bmatrix} = \begin{bmatrix} 2 \\ 68 \\ 69 \end{bmatrix}$$

$$2) 1 \begin{bmatrix} 2 \\ 1 \\ 4 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 4 \end{bmatrix}, b_1 = \begin{bmatrix} 2 \\ 68 \\ 69 \end{bmatrix} - \begin{bmatrix} 2 \\ 1 \\ 4 \end{bmatrix} = \begin{bmatrix} 0 \\ 67 \\ 65 \end{bmatrix}$$

$$3) 13 \begin{bmatrix} 0 \\ 4 \\ 5 \end{bmatrix} = \begin{bmatrix} 0 \\ 52 \\ 65 \end{bmatrix}, b_1 = \begin{bmatrix} 0 \\ 67 \\ 65 \end{bmatrix} - \begin{bmatrix} 0 \\ 52 \\ 65 \end{bmatrix} = \begin{bmatrix} 0 \\ 15 \\ 0 \end{bmatrix}$$

$$4) 2 \begin{bmatrix} 0 \\ 7 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 14 \\ 0 \end{bmatrix}, b_1 = \begin{bmatrix} 0 \\ 15 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 14 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$\text{Решение: } 33 \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix} + 1 \begin{bmatrix} 2 \\ 1 \\ 4 \end{bmatrix} + 13 \begin{bmatrix} 0 \\ 4 \\ 5 \end{bmatrix} + 2 \begin{bmatrix} 0 \\ 7 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 101 \\ 101 \\ 102 \end{bmatrix}$$

50 съёмов, 5 раскладок, отход — 30

Можно предложить модификацию тетрис-элиминатора, которая в некоторых случаях даёт более вырожденное оптимальное решение. Отличие от вышеизложенного алгоритма заключается в выборе очередного раскроя для элиминации вектора требований.

а). Пусть  $S$  таково, что

$$b[M] \cdot l[M] = S \cdot L$$

б). Построим "образ" требуемого раскроя по правилу:

$$\tilde{r}[i] = [b[i]/S + 0.5]$$

(округление до ближайшего целого)

в). Уточнение "образа"

Если  $\tilde{r}[M] \cdot l[M] = L - \tilde{L} < L$ , то дополняем  $\tilde{r}[M]$  до наиболее безотходного раскроя, "забивая"  $\tilde{L}$ , в первую очередь, заготовками из числа ненулевых компонент  $\tilde{r}$  в порядке убывания величины отношения  $b[i]/\tilde{r}[i]$  (например, в рамках  $r$ -задачи с длиной сырья  $\tilde{L}$  и вектором дохода  $v[M] = l[M]$ )

Если  $\tilde{r}[M] \cdot l[M] = L + \tilde{L} > L$ , то уменьшаем его до раскроя, отбрасывая заготовки из числа ненулевых компонент  $\tilde{r}$  в порядке возрастания  $b[i]/\tilde{r}[i]$ .

В итоге получаем раскрой  $r[M]$  для элиминации текущего вектора требований. Далее выполняются шаги 7,8 и алгоритм либо завершается, либо переходит к выбору очередного "образа".

Пример (см. рис. 2):  $L = 12$ ,  $l = (6, 4, 3, 2)$ ,  $b = (10, 40, 60, 40)$

$$A = \begin{matrix} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 1 & & & & & & & & & & & \\ 2 & & & & & & & & & & & \\ 3 & & & & & & & & & & & \\ 4 & & & & & & & & & & & \\ 5 & & & & & & & & & & & \\ 6 & & & & & & & & & & & \\ 7 & & & & & & & & & & & \\ 8 & & & & & & & & & & & \\ 9 & & & & & & & & & & & \\ 10 & & & & & & & & & & & \\ 11 & & & & & & & & & & & \end{matrix}$$

(приведены только безотходные раскрои).

1. а).  $b[M] \cdot l[M] = 480 = 40 \cdot L$   $S = 40$

б). "Образ"  $\tilde{r}[M] =$   
 $= ([10/40 + 0.5], [40/40 + 0.5], [60/40 + 0.5], [40/40 + 0.5]) =$   
 $= (0, 1, 2, 1) = r_1$  - безотходный раскрой № 7

$\boxed{7}$  Интенсивность использования  $x_7 = 30$

$\boxed{8}$   $b_1[M] = (10, 40, 60, 40) - (0, 30, 60, 30) = (10, 10, 0, 10)$

2. а).  $b_1[M] \cdot l[M] = 120 = 10 \cdot L$   $S = 10$

б). "Образ"  $\tilde{r}[M] = (1, 1, 0, 1) = r_2$  - безотходный раскрой № 2

$\boxed{7}$  Интенсивность использования  $x_2 = 10$

$\boxed{8}$   $b_2[M] = \mathbf{0}[M]$  (конец).

Получено оптимальное решение:

$$10 \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} + 30 \begin{bmatrix} 0 \\ 1 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 10 \\ 40 \\ 60 \\ 40 \end{bmatrix}$$

имеющее степень вырожденности 2.

Элиминатор без модификации здесь получает оптимальное решение

$$5 \begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \end{bmatrix} + 13 \begin{bmatrix} 0 \\ 3 \\ 0 \\ 0 \end{bmatrix} + 15 \begin{bmatrix} 0 \\ 0 \\ 4 \\ 0 \end{bmatrix} + 6 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 6 \end{bmatrix} + 1 \begin{bmatrix} 0 \\ 1 \\ 0 \\ 4 \end{bmatrix} = \begin{bmatrix} 10 \\ 40 \\ 60 \\ 40 \end{bmatrix}$$

которое не только не вырождено, но и не является базисным.

В то же время изложенная модификация для первого примера приводит к неоптимальному решению

$$25 \underbrace{\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}}_{(1)} + 2 \underbrace{\begin{bmatrix} 0 \\ 0 \\ 4 \end{bmatrix}}_{(1)} + 1 \underbrace{\begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix}}_{(0)} + 1 \underbrace{\begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}}_{(7)} = \begin{bmatrix} 29 \\ 31 \\ 11 \end{bmatrix}, \text{ (33 раскладки с от-}$$

ходом 38).

Однако, выбрав в качестве базисных три первых (наиболее безотходных) раскроя и введя в базис безотходный раскрой  $(1, 0, 2)$  с оценкой  $\varepsilon = 1/8$ , находим оптимальное решение полученное выше элиминатором без модификации

$$25 \underbrace{\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}}_{(1)} + 4 \underbrace{\begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix}}_{(0)} + 3 \underbrace{\begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix}}_{(0)} = \begin{bmatrix} 29 \\ 31 \\ 11 \end{bmatrix}, \text{ 32 раскладки, отход} - 25$$

Для поиска целочисленного решения задачи раскроя представляется интересным использование комбинации тетрис-элиминатора с алгоритмом последовательного изменения люфта по следующей схеме:

◆ Начальное приближение находим тетрис-элиминатором (оно может оказаться и оптимальным)

◆ На его основе формируем начальную базисную матрицу и применяем МСМ

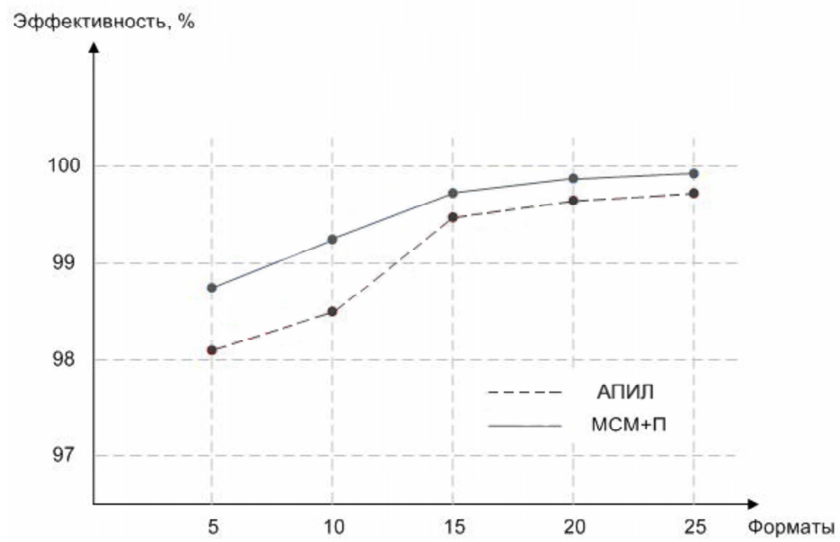
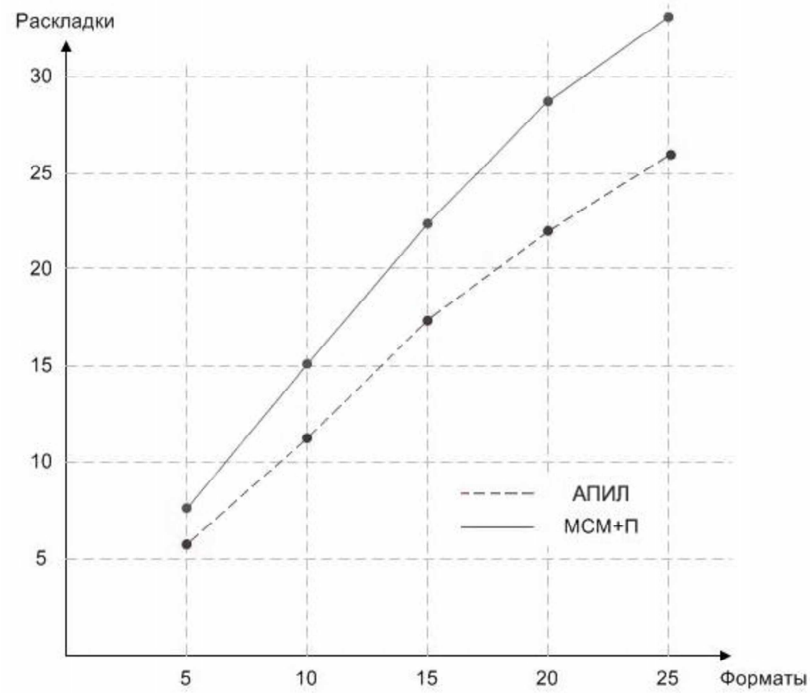
◆ Если решение нецелочисленное используем алгоритм последовательного изменения люфта.

## 7. Выводы и результаты численных экспериментов

В соответствии с изложенными алгоритмами был проведен ряд численных экспериментов: Алгоритм последовательного изменения люфта (АПИЛ) сравнивался с Модифицированным симплекс-методом с перебором (МСМ+П) на предмет вырожденности и эффективности (отношение суммарной длины всех заготовок требуемых форматов к суммарной длине использованных тамбуров) получаемого решения.

Ниже на рисунках представлены результаты численных экспериментов





## Литература

1. **Канторович Л.В., Залгаллер В.А.** Рациональный раскрой промышленных материалов. М.:Наука, 1971. // М.:Наука, 1971.
2. **Грибов А.Б., Романовский И.В.** Программирование симплекс-метода и его вариантов на АЛГОЛЕ // *Оптимальное планирование, 1963. вып. 12. С. 5-27*
3. **Романовский И.В.** Наилучшее разбиение множества на заданное число подмножеств // *АЛГОЛ-процедуры. Изд. ЛГУ, 1970. вып. 6.*
4. **Кацев С.Б., Романовский И.В.** Две задачи целочисленного программирования // *АЛГОЛ-процедуры. Изд. ЛГУ, 1975. вып. 13.*
5. **Кацев С.Б.** Решение одной обобщенной задачи о разбиении множеств. // *Кибернетика, 1977. № 5. С.115-120.*
6. **Сакович Д.Ю.** Вырожденность в задаче форматного раскроя. // *Вестник Сыктывкарского университета. Сер.1: Мат. Мех. Инф. 2008. Вып. 8. С. 75-90.*
7. **Никитенков В.Л.** О целочисленном решении задачи линейного раскроя // *Вестник Сыктывкарского университета. Сер.1: Мат. Мех. Инф. 2006. Вып. 6. С. 165-178.*
8. **Романовский И.В.** Алгоритмы решения экстремальных задач. // М.:Наука, 1977. 352 с.

### Summary

**Nikitenkov V.L., Tyufyakov A.V.** Tetris-eliminator and the algorithm of successive refinement of backlash in the vector requirements

To solve the integer problem of linear cutting proposed algorithms to obtain optimal solution either for the initial vector requirements, or if there is tighter tolerances. Discussed in the Ask the degeneracy of the obtained solutions and numerical results.