

УДК 519.717

ВЫРОЖДЕННОСТЬ В ЗАДАЧЕ ФОРМАТНОГО РАСКРОЯ

Д.Ю. Саковнич

В рамках задачи форматного раскроя рассмотрена задача о минимальном количестве перенастроек ПРС. Предложен метод получения наиболее вырожденного решения в задаче форматного раскроя.

1. Введение

Во многих областях промышленности мы сталкиваемся с задачей оптимального использования материалов. На начальной стадии производства – это задача оптимального использования природных ресурсов, на конечной – оптимальное разбиение уже готовой продукции на части, требуемые заказчику. Примером такой задачи может являться следующая: имеются заготовки одномерного сырья некоторого фиксированного размера, а также размеры и объем продукции, которую необходимо произвести. Необходимо минимизировать отход сырья при условии выработки всего требуемого объема продукции. В частности, такая задача актуальна в целлюлозно-бумажной промышленности.

Бумагоделательная машина (БДМ) производит бумагу в виде тамбуров¹ некоторой достаточно большой длины, которые в дальнейшем разрезаются на продольно-резательном станке (ПРС) на рулоны определенного диаметра и формата² требуемые заказчику. Одному клиенту, как правило, требуются рулоны одного формата. Из-за особенностей технологии производства заказы на один вид бумаги объединяются в

¹Большой широкий вал, на который наматывается бумага вышедшая с БДМ. Ширина вала обычно от 2,5 до 9 метров

²Геометрическая характеристика рулона, такая как ширина, диаметр и т.д. Здесь под форматом рулона понимается только его ширина

группы (портфели) заказов, для которых и решается задача оптимизации. Так как минимизация отхода при раскрое бумажного полотна не является тривиальной задачей, то непосредственно перед производством или даже заранее составляется заявка-раскладка (план раскроя). Заявка-раскладка состоит из раскладок (способов раскроя), в которых указывается, по сути, расположение ножей ПРС по ширине тамбура в соответствии с нарезаемыми форматами по данной раскладке. Однократное нарезание рулонов по раскладке будем называть съемом. Для каждой раскладки указывается количество съемов (интенсивность или кратность использования способа раскроя) необходимое для нарезания определенной части заявки. Суммарное количество рулонов каждого формата, нарезанное по всем раскладкам с учетом их кратностей, должно быть равно требуемому количеству рулонов (то есть удовлетворять вектору требований). При этом необходимо стремиться к тому, чтобы суммарный попутный отход по всем раскладкам был минимальным.

Нетрудно заметить, что эта задача является частным случаем целочисленной задачи линейного раскроя (ЦЗЛР).

2. Задача целочисленного линейного раскроя

1. Постановка задачи Пусть имеется одномерное сырье (тамбур) длины L (ширина тамбура), из которого необходимо вырезать заготовки (рулоны) различных типов из множества M ($|M| = m$ – число элементов в множестве M , в нашей задаче m – число различных форматов в заявке). Длины заготовок задаются вектором $l[M]$ (форматы рулонов в заявке). Требуемое количество заготовок каждого типа обозначим через $b[M]$ (вектор требований). Требуется, удовлетворив вектор требований (ВТ) на заготовки, израсходовать минимальное количество штук сырья, что эквивалентно минимизации отхода (так как полезная площадь фиксирована вектором требований).

Пусть теперь N – множество номеров всех возможных способов раскроя тамбура (множество номеров всех возможных раскладок, включая и нетехнологичные, либо по количеству вырезаемых заготовок, либо с большим отходом). $|N| = n$ – число всех возможных способов раскроя. Из всех способов раскроя составим целочисленную неотрицательную матрицу $A[M, N]$, где $A[M, j]$ – способ раскроя с номером j . $A[i, j]$ – количество заготовок i -го типа, вырезаемых j -м способом раскроя (число рулонов i -го формата в j -ой раскладке). Очевидно, что способ раскроя $A[M, j]$ должен удовлетворять ограничениям

$$l[M] \cdot A[M, j] \leq L, A[M, j] \geq 0[M] \quad (1)$$

так как количество вырезаемых заготовок не может быть отрицательным, и суммарная длина вырезаемых в способе заготовок не должна превышать длины сырья.

Введем целочисленный неотрицательный вектор неизвестных $X[N]$ -интенсивности использования способов раскроя. $X[j]$ -интенсивность j -го способа раскроя.

Ясно, что однократное использование способа раскроя приводит к израсходованию одной штуки сырья длиной L . Тогда функция $f[X] = 1[N] \cdot X[N]$ выражает суммарное количество израсходованных штук сырья на вырезание всех требуемых заготовок. У данной функции мы должны найти минимум при условии вырезания всех заготовок

$$A[M, N] \cdot X[N] = b[M]$$

Таким образом, приходим к следующей задаче

$$f[X] = 1[N] \cdot X[N] \longrightarrow \min \quad (2)$$

$$A[M, N] \cdot X[N] = b[M] \quad (3)$$

$$X[N] \geq 0[N] \quad (4)$$

$$X[N]\text{-целочисленный} \quad (5)$$

в которой столбцы матрицы $A[M, N]$ удовлетворяют ограничению (1).

Если к ограничениям этой задачи добавить еще два (соответствующих максимуму конечного отхода и максимальному количеству вырезаемых в способе раскроя заготовок)

$$l[M] \cdot A[M, j] \geq L - p, j \in N, p - \text{максимальный конечный отход} \quad (6)$$

$$A[M, j] \cdot \mathbf{1}[M] \leq k - 1, k - \text{максимальное число ножей на ПРС} \quad (7)$$

то получим задачу целочисленного программирования, реализующую нахождение оптимальных раскладок и их интенсивностей.

2. Метод решения задачи Для этой задачи можно применить комбинированный алгоритм решения целочисленной задачи линейного раскроя [1, 2].

Суть его заключается в следующем. Изначально решается задача линейного раскроя без условия целочисленности посредством модифицированного симплекс-метода (МСМ) с тем изменением, что матрица раскроев $A[M, N]$ не хранится, а необходимый для ввода в базис столбец (наилучший раскрой $r[M] = A[M, j_0]$) генерируется из условия

$r[M] \cdot v[M] \rightarrow \max$, где $v[M]$ – вектор двойственных переменных (r -задача) [3]. При этом, очевидно, если величина $\varepsilon[j_0] = r[M] \cdot v[M] - 1 \leq 0$, то текущее решение оптимально, так как оценки остальных столбцов будут еще меньше. В противном случае найденный столбец $A[M, j_0]$ вводится в базис.

После того, как решение линейной задачи раскроя $X^*[N]$ найдено, его целевую функцию (суммарное число съемов) округляют вверх до ближайшего целого

$$K = [f^*] + 1 (\mathbf{1}[N] \cdot X^*[N] = f^*),$$

а само решение разделяют на целую и дробную части.

$$X^*[N] = [X^*[N]] + \{X^*[N]\} \triangleq \bar{X}[N] + \alpha[N]$$

Затем выделяют остаток вектора требований $\bar{b}[M] = A[M, N] \cdot \alpha[N]$ – целочисленный и разбивают его на k раскроев, используя какой-либо комбинаторный метод, где $k = \mathbf{1}[N] \cdot \alpha[N] + (1 - \{f^*\})$.

Можно показать, что решение такой целочисленной задачи существует [2]. Так как кратности элементов остатка вектора требований $\bar{b}[M]$ – небольшие и мы знаем число съемов в оптимальном решении, то комбинаторный метод не займет много времени.

3. Характер результата Как уже упоминалось, для получения решения на первом шаге применяется модифицированный симплекс метод, решением которого являются коэффициенты нецелочисленной линейной комбинации, состоящей из m линейно независимых способов раскроя и представляющей собой разбиение вектора требований по способам раскроя. Как правило, в этой линейной комбинации векторов все коэффициенты отличны от нуля и большинство из них нецелочисленные, то есть в решении, полученном на втором шаге, будет использовано большое количество съемов (максимальное количество съемов на втором шаге $m - 1$). Как показало тестирование комбинированного метода, все способы раскроя, полученные на втором шаге, различны и не совпадают со способами раскроя полученными в МСМ. Это приводит к тому, что в конечном оптимальном целочисленном решении используется большое число различных способов раскроя.

На практике, каждый новый способ раскроя означает предварительную настройку ПРС, что требует дополнительного времени и усилий персонала. Поэтому число различных раскроев участвующих в решении, по возможности нужно свести к минимуму.

Как правило, решение исходной задачи не единственно, и среди решений присутствуют такие, в которых используется различных раскроев существенно меньше исходного.

Поэтому возникает следующая задача: среди всех оптимальных решений исходной задачи выбрать то, которое содержит наименьшее число различных способов раскроя, то есть минимизировать число перенастроек ПРС.

3. Задача о минимальном количестве перенастроек ПРС

1. Постановка задачи Пусть f^* оптимальное значение целевой функции. Рассмотрим набор векторов такой, что

$$f(X[N]) = f^*$$

$$A[M, N] \cdot X[N] = b[M]$$

$$X[N] \geq \mathbf{0}[N]$$

$$X[N] \text{—целочисленный}$$

Каждая положительная компонента вектора $X[N]$ означает использование в решении одного дополнительного способа раскроя. Следовательно, задача минимизации числа различных способов раскроя участвующих в решении равносильна минимизации положительных компонент вектора $X[N]$, то есть нам необходимо найти самое вырожденное решение исходной задачи. Обычно складывается кардинально противоположная ситуация – вырожденных решений стараются избегать.

Как говорилось выше, решение, получаемое на втором шаге, состоит из различных способов раскроя, поэтому оптимизацию по числу перенастроек ПРС имеет смысл вести только на первом шаге решения, который представляет собой МСМ.

Различным решениям МСМ соответствуют различные базисные подматрицы, в которых все столбцы имеют нулевую оценку. Если мы переберем все возможные такие подматрицы, то, очевидно, мы переберем все решения и выберем наилучшее.

Прежде чем осуществлять перебор решений симплекс-метода, заметим, что оптимальное значение целевой функции в МСМ не является целым числом. Это означает, что среди решений задачи линейного раскроя без условия целочисленности отсутствуют полностью целочисленные решения ЦЗЛР, даже если существует решение, состоящее из m и менее способов раскроя. С другой стороны, в оптимальном решении

поставленной задачи могут не участвовать способы раскроя, используемые в оптимальном решении нецелочисленной задачи.

Для устранения этих недостатков, к уже имеющимся ограничениям добавим еще одно

$$\mathbf{1}[N] \cdot X[N] \geq K$$

которое можно записать в виде равенства

$$\mathbf{1}[N] \cdot X[N] - X[n+1] = K$$

Это условие того, что значение целевой функции не опустится ниже оптимального целочисленного значения. Так как целевая функция непрерывна и ее минимальное значение без дополнительного условия меньше либо равно K , то оптимальным значением целевой функции в новой задаче будет K , то есть обязательно целое значение. Следовательно, среди решений этой задачи могут быть и полностью целочисленные, и, если они существуют, то при полном переборе решений мы обязательно их найдем и выберем наилучшее. Базисность целочисленного решения рассматривается в [4].

Добавление одного условия, увеличивает размерность решаемой задачи на единицу. То есть размерность задачи уже не m , как было раньше, а $m+1$. Также, при решении этой задачи с помощью МСМ, добавится еще одна переменная x_{n+1} с нулевым коэффициентом в целевой функции которой соответствует способ раскроя $r_{n+1} = (0, 0, \dots, 0, -1)$.

Таким образом, приходим к следующей расширенной задаче линейного программирования

$$f(X[N]) = \mathbf{1}[N] \cdot X[N] \rightarrow \min$$

$$\begin{pmatrix} A[M, N] & 0 \\ \vdots & \vdots \\ 1 & \dots & 1 & -1 \end{pmatrix} \cdot X[N'] = \begin{pmatrix} B[M] \\ K \end{pmatrix}, N' = N + \{n+1\}$$

$$X[N] \geq \mathbf{0}[N]$$

Среди решений этой задачи, необходимо найти то, которое дает самое вырожденное решение исходной целочисленной задачи.

2. Методы решения Для решения поставленной задачи введем новую целевую функцию.

Допустим, что все раскладки, полученные перебором на втором шаге комбинированного метода, различны и не совпадают ни с одной из раскладок полученных в СМ. Как говорилось ранее, это вполне обосновано, так как на практике такие совпадения встречаются крайне редко.

Рассмотрим некоторое решение, полученное с помощью СМ, и оценим число различных раскроев, которое будет участвовать в окончательном целочисленном решении.

Пусть X некоторое оптимальное решение. Каждая положительная компонента вектора X добавляет один, отличный от остальных, способ раскроя. Дробные части компонент вектора X формируют остаток вектора требований и раскладываются на втором шаге комбинированного метода, причем количество съёмов полученных на этом шаге равно сумме дробных частей компонент вектора X .

Таким образом, каждое решение с точки зрения вырожденности характеризуется двумя параметрами:

$fr(X[N]) = \{X[N]\} \cdot 1[N]$ – сумма дробных частей решения X ;

$p(X[N])$ – количество положительных компонент вектора X .

Тогда целевую функцию можно определить следующим образом:

$$g(X[N]) = fr(X[N]) + p(X[N]) \rightarrow \min,$$

то есть как оценочное значение числа различных раскроев участвующих в решении.

Нетрудно заметить, что, найдя минимум этой функции, мы решим поставленную перед нами задачу по минимизации числа перенастроек ПРС.

Различным решениям МСМ соответствуют различные базисные подматрицы, составленные из столбцов с нулевой оценкой ("кандидатов в базис"). Поэтому первым действием при оптимизации перенастроек выберем все столбцы с нулевой оценкой. Это можно осуществить с помощью алгоритма решения r -задачи с тем изменением, что он не останавливался бы на первом столбце с нулевой оценкой (которая является максимальной, так как решение оптимально), а продолжил бы свою работу, генерируя тем самым все возможные столбцы с нулевой оценкой [1].

Далее, если количество столбцов с нулевой оценкой мало, то для нахождения решения можно воспользоваться полным перебором всех базисных подматриц составленных из этих столбцов. Это можно сделать, например, при помощи кодов Грея[5], которые позволяют перебирать

все подмножества в такой последовательности, что каждые два соседних подмножества отличаются лишь одним элементом, то есть переход от одной базисной матрицы к другой можно осуществлять с помощью итерации симплекс метода. При этом "по пути" нам могут встретиться линейно зависимые наборы векторов. Такие случаи нужно обрабатывать отдельно. Общее количество матриц, которое надо будет перебрать – $C_{\text{кол-во столбцов}}^m$.

Пример:

$$L = 11; l[M] = (7, 5, 4); b[M] = (5, 7, 12)$$

Оптимальное значение целевой функции расширенной задачи $f(X) = 12$.

Выпишем все столбцы с нулевой оценкой:

	1	2	3	4	5
7	1	0	0	0	0
5	0	2	0	0	1
4	1	0	2	0	1
	1	1	1	-1	1

Первые четыре столбца составляют базисную подматрицу полученную МСМ.

С помощью кодов Грея установим последовательность рассмотренных матриц, а также вычислим соответствующие этим матрицам решения и число участвующих в них способов раскрытия (в строке переходов в скобках указаны номера столбцов, из которых составлена матрица, а над стрелками – столбцы, вводящиеся в базис и выводящиеся из него).

переход матриц	(1, 2, 3, 4)	$\xrightarrow{+5-3}$	(1, 2, 5, 4)	$\xrightarrow{+3-1}$	(3, 2, 5, 4)	$\xrightarrow{+1-2}$	(3, 1, 5, 4)	$\xrightarrow{+2-4}$	(3, 1, 5, 2)
X	(5, 3.5, 3.5, 0)		(5, 0, 7, 0)		(вырожд.)		(0, 5, 7, 0)		(вырожд.)
g(X)	4		2		–		2		–

Как видно, вторая и четвертая матрица дают одинаковую степень вырожденности. Поэтому любую из них можно взять в роли окончательного решения.

На практике, столбцов с нулевой оценкой довольно много. Даже в задачах небольшой размерности ($m \approx 10$) их количество достигает 1000 ($C_{1000}^{10} = 263409560461970212832400 \sim 10^{25}$). Поэтому осуществить полный перебор базисов не представляется возможным.

Допустим теперь, что в базис можно вводить столбцы, постоянно уменьшая значение целевой функции. Тогда можно предложить следующий алгоритм решения.

Сначала, как и ранее, мы выделяем все столбцы с нулевой оценкой. Далее, на каждой итерации просматриваем все выбранные столбцы, с помощью итерации СМ вводим их в базис и вычисляем возможное значение целевой функции $g(X)$. Среди всех "кандидатов" в базис выбираем того, на котором сильнее всего уменьшилась целевая функция $g(X)$. Если столбец, уменьшающий целевую функцию, не найден, то алгоритм прекращает свою работу. После чего выделяется целая и дробная часть найденного решения и осуществляется перебор остатка вектора требований (второй шаг КМ).

3. Свойства целевой функции и алгоритм оптимизации Рассмотрим введенную целевую функцию.

Так как значение исходной целевой функции $f(X)$ (расширенной задачи) есть целое число, то $fr(X)$ тоже целое. Тогда мы имеем: $p(X)$ – целая дискретная функция; $fr(X)$ – целая дискретная функция. Следовательно, $g(X)$ – тоже целая дискретная функция как сумма целых функций. Из этого следует, что функция может убывать только дискретно. Но как быть в том случае, когда для того чтобы уменьшилась целевая функция необходимо сменить несколько базисных векторов?

Для этого можно воспользоваться, например, несовершенством машинной работы с числами с плавающей точкой.

При реализации этого подхода производятся операции над действительными числами с некоторой точностью (например, тип `double` – 18 знаков), то есть получаем не целую целевую функцию, каковой она является в теории, а некоторую кусочно-непрерывную возле целых значений функцию g' . g' получает дробное значение близкое к целому. Изменение g' на незначительную дробную величину не вносит никакой вклад в целевую функцию g , так как по сути, является целым числом, но позволят сравнить два решения, дающие одинаковое значение g , что позволяет алгоритму продолжить свою работу. Алгоритм продолжает свою работу до тех пор, пока изменение $g' > \Delta$, где Δ – эмпирически выведенная небольшая константа.

4. Пример Рассмотрим действие описанного алгоритма на небольшом примере.

Зададим следующие условия для нашей задачи:

$$L = 11(\text{длина тамбура});$$

$$l[M] = (2, 3, 5, 7) \text{ (длины заготовок);}$$

$$b[M] = (12, 7, 4, 7) \text{ (вектор требований).}$$

Начальная базисная подматрица и решение имеют вид:

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, X = (12, 7, 4, 7), f(X) = 30.$$

После применения симплекс-метода получаем следующее решение линейной задачи

$$A = \begin{pmatrix} 0 & 0 & 0 & 2 \\ 1 & 0 & 2 & 0 \\ 0 & 2 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}, X = (1, 0.5, 3, 6), f(X) = 10.5$$

Округляем найденное решение вниз до ближайшего целого

$$\bar{X} = (1, 0, 3, 6).$$

Вычисляем остаток вектора требований

$$\begin{aligned} b = b - A \cdot \bar{X} &= (12, 7, 4, 7) - \begin{pmatrix} 0 & 0 & 0 & 2 \\ 1 & 0 & 2 & 0 \\ 0 & 2 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 3 \\ 6 \end{pmatrix} = \\ &= (12, 7, 4, 7) - (12, 7, 3, 7) = (0, 0, 1, 0) \end{aligned}$$

Далее разбиваем остаток вектора требований с помощью комбинаторного алгоритма. Заметим, что весь остаток вектора требований укладывается на один тамбур, то есть на втором шаге комбинированного метода решения мы получим один способ раскроя с интенсивностью равной 1. Конечным же решением этой задачи без проведения оптимизации по перенастройкам ПРС будет

$$\begin{array}{cccc} 1 & 3 & 6 & 1 \\ \hline 0 & 0 & 2 & 0 \\ 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{array}, f(X) = 11$$

(столбцы матрицы представляют собой способы раскроя, элементы в первой строке – их интенсивности). В полученном оптимальном целочисленном решении используется четыре различных способа раскроя.

Это значение не является оптимальным для задачи минимизации перенастроек ПРС, поэтому применим описанный ранее алгоритм оптимизации.

В роли начальной базисной подматрицы расширенной задачи можно взять матрицу

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & -1 \end{pmatrix}.$$

Последний столбец матрицы A не содержит в себе заготовок ("пустой" способ раскроя), поэтому если его интенсивность будет отлична от нуля, то это будет означать, что весь тамбур будет отправлен в отход, что недопустимо. Но также, в нашей задаче, это будет означать, что значение целевой функции не оптимально (так как больше уже известного минимума), следовательно, в оптимальном решении, интенсивность этого способа раскроя будет равна нулю.

Вектор требований в расширенной задаче будет иметь вид $b = (12, 7, 4, 7, 11)$ (добавилась одна компонента равная значению целевой функции). Начальное решение $X = (12, 7, 4, 7, 19)$.

После применения симплекс-метода к этой задаче получим следующее решение

$$A = \begin{pmatrix} 0 & 0 & 1 & 2 & 0 \\ 0 & 0 & 3 & 0 & 1 \\ 0 & 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}, X = (1, 2, 2, 5, 1).$$

Несмотря на то, что это решение целочисленно, оно не является оптимальным с точки зрения количества перенастроек ПРС, так как содержит пять различных способов раскроя, даже в первом случае мы получили четыре.

Далее находим все столбцы с нулевой оценкой (в данном примере их 28) и вычисляем значение новой целевой функции:

4. Особенности расширенной задачи

Рассмотрим матрицу раскроев расширенной задачи линейного раскроя

$$A'[M+1, N+1] = \begin{pmatrix} & & & 0 \\ & A & & \vdots \\ & & & 0 \\ 1 & \dots & 1 & -1 \end{pmatrix}$$

Вычислим значение вектора двойственных переменных $v[M+1]$ расширенной задачи

$$v[M+1] = C[N'] \cdot A'^{-1}[N', M+1], \text{ где } N' - \text{ базис}$$

Выделим два случая: когда базисная подматрица A' содержит пустой раскрой и когда базисная подматрица A' в своем составе этого раскроя не содержит, и вычислим вектор двойственных переменных в обоих случаях.

1) A' содержит пустой раскрой. Тогда базисная подматрица имеет вид:

$$A'[M+1, N'+1] = \begin{pmatrix} & & & 0 \\ & A[M, N'] & & \vdots \\ & & & 0 \\ 1 & \dots & 1 & -1 \end{pmatrix} \Rightarrow$$

$$\Rightarrow A'^{-1}[N'+1, M+1] = \begin{pmatrix} & & & 0 \\ & A^{-1}[N', M] & & \vdots \\ & & & 0 \\ \sum_i A^{-1}[1, i] & \sum_i A^{-1}[2, i] & \dots & \sum_i A^{-1}[m, i] & -1 \end{pmatrix}$$

Вычислим $v[M+1]$.

$$v[M+1] = C[N'] \cdot A'^{-1}[N', M+1] = (1, 1, \dots, 1, 0) \cdot A'^{-1}[N', M+1] =$$

$$= \left(\sum_i A^{-1}[1, i] \sum_i A^{-1}[2, i] \dots \sum_i A^{-1}[m, i], 0 \right)$$

2) A' не содержит пустой раскрой.

$$A'[M+1, N'] = \begin{pmatrix} & & & A[M, N'] \\ & & & \vdots \\ & & & 0 \\ 1 & \dots & 1 & \end{pmatrix}$$

В этом случае вектор двойственных переменных вычисляется по формуле

$$v[M + 1] = C[N'] \cdot A'^{-1}[N', M + 1] = (1, 1, \dots, 1) \cdot A'^{-1}[N', M + 1]$$

Заметим, что в этом случае последняя строка матрицы $A'[M + 1, N']$ совпадает с вектором $C[N']$.

Тогда, так как $A' \cdot A'^{-1} = E$, то

$$\begin{aligned} v[M + 1] &= C[N'] \cdot A'^{-1}[N', M + 1] = A'[m + 1, N'] \cdot A'^{-1}[N', M + 1] = \\ &= E[m + 1, M + 1] = (0, \dots, 0, 1) \end{aligned}$$

Таким образом, мы получили, что если в базисной подматрице отсутствует пустой раскрой, то вектор двойственных переменных является постоянным. Вычислим по этому вектору оценку для какого-либо столбца $r[M + 1]$

$$\begin{aligned} \varepsilon[j] &= v[M + 1] \cdot r[M + 1] - C[j] = \\ &= (0, \dots, 0, 1) \cdot (r_1, \dots, r_m, 1) - 1 = 1 - 1 = 0 \end{aligned}$$

Следовательно, в этом случае все вектора имеют нулевую оценку, то есть все они являются кандидатами в базис. Этот факт, несколько осложняет применение описанного выше метода решения. Ведь прежде чем проводить оптимизацию, мы получаем все вектора с нулевой оценкой и сохраняем их в памяти. То есть в этом случае нам потребуется получить и загрузить в память все способы раскроя, количество которых очень велико. Поэтому в таких случаях количество загружаемых столбцов ограничивается некоторой константой, и алгоритм применяется для этого ограниченного числа столбцов, что, несомненно, сказывается на качестве решения.

5. Результаты тестирования

Результаты, полученные с помощью описанного алгоритма решения задачи о минимизации перенастроек ПРС, сравнивались с аналогичным продуктом фирмы АВВ внедряемым на ОАО "МБП СЛПК". Результаты сравнения решений двух алгоритмов приведены в таблице 1.

В первых столбцах таблицы содержится информация по заявке раскладке, такая как общее количество нарезаемых рулонов и количество среди них различных форматов. Далее в таблице представлена информация по вырожденности решения:

Без оптимизации – количество различных раскроев (перенастроек ПРС) в решении до оптимизации;

Табл. 1: Сравнение перенастроек ПРС

№	Всего рулонов	Различных форматов	Без оптимизации	С оптимизацией	АВВ
1	427	7	8	6	7
2	213	6	5	4	4
3	211	6	5	4	4
4	519	26	30	24	24
5	523	26	27	24	22
6	940	7	8	6	6

С оптимизацией – количество различных раскроев (перенастроек ПРС) в решении после применения описанного алгоритма оптимизации;

АВВ – количество различных раскроев в решении выдаваемом продуктом фирмы АВВ.

Из сравнительной таблицы видно, что, несмотря на многие предположения сделанные перед применением алгоритма, этот метод нахождения вырожденности дает результат сравнимый с результатом, выдаваемым одним из ведущих производителей программного обеспечения в этой области, а иногда, и превосходящий его.

Литература

1. **Никитенков В.Л., Саковнич Д.Ю.** О Реализации комбинированного алгоритма решения целочисленной задачи линейного раскроя // *Вестник Сыктывкарского университета. Сер.1: Мат. Мех. Инф.* 2006. Вып.6. С. 199–208.
2. **Никитенков В.Л.** О целочисленном решении задачи линейного раскроя // *Вестник Сыктывкарского университета. Сер.1: Мат. Мех. Инф.* 2006. Вып.6. С. 165–178.
3. **Романовский И.В.** Алгоритмы решения экстремальных задач. М.: Наука, 1977. 352 с.
4. **Никитенков В.Л.** Базисность оптимального решения целочисленной задачи линейного раскроя // *Вестник Сыктывкарского университета. Сер.1: Мат. Мех. Инф.* 2008. Вып.8. С. 69 – 74.
5. **Рейнгольд Э., Нивергельт Ю., Део Н.** Комбинаторные алгоритмы. Теория и практика. М.: Мир, 1980. 476 с.

Summary**Sakovnich D.Y.** Singularity in the format cutting problem

In scope of format cutting problem the problem of minimization winder adjustments is considered. It was offered the method for getting the most singular solution if the problem.

*Сыктывкарский университет**Поступила 24.02.2008*