ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Май—июнь 2009. Том 16, № 3. С. 47–62

УДК 004

МЕТОД ГРУППИРОВКИ ДЛЯ РЕШЕНИЯ НЕПРЕРЫВНОЙ ЗАДАЧИ ЛИНЕЙНОГО РАСКРОЯ $^{*)}$

В. М. Картак

Аннотация. Рассматривается задача линейного раскроя большой размерности. Данную задачу можно интерпретировать как задачу линейного целочисленного программирования. С помощью предложенного метода группировки удаётся получить близкое к оптимальному начальное решение для соответствующей задачи непрерывной релаксации, что часто позволяет снизить время нахождения оптимального решения.

Ключевые слова: линейная релаксация, симплекс-метод, задача раскроя-упаковки.

Введение

Классическая задача линейного раскроя *One-Dimensional Cutting Stock Problem* (1dCSP) состоит в следующем: одномерный материал в виде прутков длины L необходимо разделить на заготовки меньших длин $\mathbf{l} = (l_1, l_2, \ldots, l_m)$ в требуемых количествах $\mathbf{b} = (b_1, b_2, \ldots, b_m)$ соответственно, m – число типов заготовок. Целью является минимизация количества использованного материала. Для краткости задачу 1dCSP со входом $E = (L, m, \mathbf{l}, \mathbf{b})$ обозначим через E. Без потери общности будем считать, что $l_1 \geqslant l_2 \geqslant \ldots \geqslant l_m$.

Задача 1dCSP в классификации Дикхоффа (Dyckhoff) [8] обозначается как 1/V/I/R. Известно, что она относится к классу NP-трудных задач (см. [1]) и имеет широкое практическое применение.

В [2, 10] показано, что проблема 1dCSP сводится к задаче линейного целочисленного программирования.

Обозначим способ раскроя прутка как m-мерный вектор с целочисленными неотрицательными компонентами $\mathbf{a}_j = (a_{1j}, \ldots, a_{mj})^T$,

$^{*)}$Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований (проект 08–01–00495), Программой сотрудничества «Михаил Ломоносов 2008» между Минобрнауки РФ и Немецкой службой академических обменов.

$j = 1, \ldots, N$, для которого выполнено ограничение $\sum\limits_{i=1}^{m} l_i \cdot a_{ij} \leqslant L$. Данный вектор $\mathbf{a}_j$, $j = 1, \ldots, N$, часто называется *вектором раскроя*, a_{ij} — количество i-х заготовок, входящих в j-й вектор раскроя, N — число всевозможных векторов раскроя.

Тогда задача 1dCSP может быть смоделирована [2, 10] как следующая задача линейного целочисленного программирования:

$$z = \sum_{j=1}^{N} x_j \longrightarrow \min \quad \text{s.t.} \quad A_E \mathbf{x} = \mathbf{b}, \quad \mathbf{x} \in \mathbb{Z}_+^N. \tag{1}$$

Матрица A_E размера $m \times N$ содержит все возможные векторы раскроя $\mathbf{a}_j$ как столбцы, целое x_j — число прутков, которые должны быть разделены в соответствии с вектором раскроя $\mathbf{a}_j$. Обозначим оптимальное значение задачи (1) как $Z^*(E)$.

На практике часто рассматривается линейная релаксация задачи (1), для которой отсутствует условие целочисленности вектора $\mathbf{x}$. В этом случае задача сформулируется следующим образом:

$$z = \sum_{j=1}^{N} x_j \longrightarrow \min \quad \text{s.t.} \quad A_E \mathbf{x} = \mathbf{b}, \quad \mathbf{x} \in \mathbb{R}_+^N. \tag{2}$$

Пусть $Z_s(E)$ — оптимальное значение целевой функции задачи (2).

Округлим найденное оптимальное значение задачи (2) сверху $\lceil Z_s(E) \rceil$. Полученное число можно использовать в качестве нижней границы для целочисленной задачи $Z^*(E)$.

Многочисленные вычислительные эксперименты показали, что у большинства задач 1dCSP эта нижняя граница достигается, т. е. $Z^*(E) = \lceil Z_s(E) \rceil$ (подробнее см. [3, 11]). Более того, на данный момент не известно ни одного примера, для которого $Z^*(E) > \lceil Z_s(E) \rceil + 1$. Эта нижняя граница является лучшей из известных.

В настоящее время существует множество алгоритмов решения задачи 1dCSP [6, 7, 12], которые на первом шаге получают решение линейной релаксации (2) и затем, отталкиваясь от него, находят оптимум задачи (1).

Однако для задач большой размерности ($m \gg 1$) время решения (2) существенно возрастает. В работе [5] высказана гипотеза, что число итераций растёт как $O(m^4)$.

Для повышения эффективности работы симплекс-алгоритма применяются различные подходы, в частности, предложен субградиентный метод [13], а в [15] исследованы методы генерации столбцов.

Известно, что время работы симплекс-алгоритма сильно зависит от выбора стартового решения. Данная статья посвящена построению «хорошего» стартового решения, позволяющего для некоторых классов задач 1dCSP существенно сократить время нахождения оптимума задачи (2).

1. Задача непрерывной релаксации

Для решения задачи непрерывной релаксации (2) используют симплекс-алгоритм с неявно заданной матрицей ограничений, так как число различных карт раскроя N растёт экспоненциально в зависимости от m и для реальных задач может быть достаточно велико. Данный подход предложен В. А. Залгаллером [2] и независимо Гилмором и Гомори [10]. Суть метода заключается в генерации только необходимых карт раскроя (столбцов матрицы A_E). Остановимся на нём подробнее.

Пусть найдено оптимальное решение задачи (2) на некотором подмножестве карт раскроя $P \subset A_E$ и y_i $(1 \leqslant i \leqslant m)$ — соответствующие ему двойственные оценки. Текущее решение может быть улучшено, если существует карта раскроя $\mathbf{a}'$ такая, что $\sum\limits_{i=1}^{m} a'_i \cdot y_i > 1$. В этом случае она должна быть добавлена в подмножество $P := P \cup \{\mathbf{a}'\}$. Повторяем процесс нахождения оптимума на новом подмножестве P. Если на некотором шаге не находится карта раскроя $\mathbf{a}'$, удовлетворяющая данному неравенству, то это значит, что оптимальное решение исходной здачи найдено. Поиск соответствующей карты раскроя $\mathbf{a}'$ осуществляется с помощью неограниченной задаче о рюкзаке [10].

Так как задача (2) решается с помощью прямого симплекс-метода, то на каждом его шаге генерируется допустимое базисное множество размера m и строится соответствующее ему базисное решение $\mathbf{x}$. Таким образом, любому допустимому базисному решению $\mathbf{x}$ соответствует набор, состоящий из m карт раскроя $A = (\mathbf{a}_j)$, $j = \overline{1, m}$, $\mathbf{a}_j \in A_E$. Матрица A невырожденна и связана с вектором $\mathbf{x}$ соотношением $A\mathbf{x} = \mathbf{b}$.

В качестве стартового множества P можно использовать любое допустимое базисное решение.

В [4] предлагается взять набор, состоящий из m элементарных карт раскроя $A = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m)$:

$$\mathbf{a}_1 = \left(\left[\frac{L}{l_1}\right], 0, \ldots, 0\right)^T, \quad \mathbf{a}_2 = \left(0, \left[\frac{L}{l_2}\right], 0, \ldots, 0\right)^T, \ldots,$$
$$\mathbf{a}_m = \left(0, \ldots, \left[\frac{L}{l_m}\right]\right)^T.$$

Видно, что матрица A диагональная и невырожденная. Соответствующее стартовое решение имеет вид

$$\mathbf{x} = \left(\frac{b_1}{a_{11}},\ \frac{b_2}{a_{22}},\ \ldots,\ \frac{b_m}{a_{mm}},\ 0,\ \ldots,\ 0\right). \tag{3}$$

В большинстве работ (например, [9, 13]) для построения начального решения предлагается использовать «метод первый подходящий» (FFD).

Мы предлагаем в качестве стартового решения использовать решение некоторой вспомогательной задачи 1dCSP меньшей размерности. Способом построения такой задачи выберем метод, известный в литературе как «метод группировки» [9].

2. Метод группировки

Общая идея метода группировки заключается в том, что из исходной задачи $E = (L, m, \mathbf{l}, \mathbf{b})$ генерируется группировочная задача $E^g = (L, m^g, \mathbf{l}^g, \mathbf{b}^g)$ по правилу объединения нескольких заготовок в один тип, тем самым понижая её размерность.

Пусть $n = \sum\limits_{i=1}^{m} b_i$ – общее число заготовок. В [9] для построения асимптотически приближённого решения задачи 1dCSP предлагается разбить общее множество заготовок

$$l^* = (\underbrace{l_1, \ldots, l_1}_{b_1}, \underbrace{l_2, \ldots, l_2}_{b_2}, \ldots, \underbrace{l_m, \ldots, l_m}_{b_m}) = (l_1^*, l_2^*, \ldots, l_n^*)$$

на группы G_k, $k = 1, \ldots, K$, содержащие одинаковое число заготовок, за исключением, быть может, последней группы. Число K выбирается равным $\lceil n/c \rceil$, где константа c — число заготовок в каждой группе — задаётся некоторым подходящим образом, $c \ll n$.

В [5] данный способ группировки используется для оценки значения $Z(E)$ снизу. Группировочная задача имеет вид $E^g = (L, m^g, l^g, b^g)$, где $m^g = K - 1$, $l_i^g = l_{ci+1}^*$, $b_i^g = c$, $1 \leqslant i \leqslant m^g$. В этом случае решение исходной задачи можно оценить как $Z(E^g) \leqslant Z(E) \leqslant Z(E^g) + c + \lceil n/c \rceil$.

Однако описанный способ группировки имеет следующие недостатки.

1. При разбиении на группы не учитывается геометрия заготовок (соотношение между их длинами), и в одну группу могут попасть заготовки с существенной разницей длин, что может привести к большой разнице между решениями группировочной и исходной задач.

2. Способ не применим, если $m < \lceil n/c \rceil$, так как после группировки число типов заготовок возрастает по сравнению с исходной задачей.

Далее мы предлагаем другие способы группировки.

2.1. Новые способы группировки. Целью группировки в нашем случае является получение новой задачи с меньшим числом типов заготовок, причём её решение должно быть близко к решению исходной задачи. Для достижения этой цели мы предлагаем строить группировочную задачу путём объединения наиболее близких по длине заготовок в одну группу. При этом поскольку в задаче E все заготовки одинаковой длины объединены в типы, то можно группировать только типы заготовок, а не каждую заготовку по отдельности. Тогда способ построения группировочной задачи будет выглядеть следующим образом.

Разобьём список всех типов заготовок на $m^g < m$ непересекающихся групп, содержащих k_1, k_2, …, k_{m^g} элементов соответственно:

$$\mathbf{l} = ((l_1, \ldots, l_{k_1}),\ (l_{k_1+1}, \ldots, l_{k_1+k_2}), \ldots,\ (l_{k_1+\cdots+k_{m^g-1}+1}, \ldots, l_m)).$$

Обозначим $p_1 = 1$, $p_2 = k_1 + 1$, $p_3 = k_1 + k_2 + 1$, …, $p_{m^g} = k_1 + \ldots + k_{m^g-1} + 1$. Тогда $l = (G_1, G_2, \ldots, G_{m^g})$, где $G_i = (l_{p_i}, \ldots, l_{p_{i+1}-1})$.

Группировочная задача имеет вид $E^g = (L,\ m^g,\ \mathbf{l}^g,\ \mathbf{b}^g)$, где

$$l_i^g = \max_{l_j \in G_i} l_j = l_{p_i}, \quad 1 \leqslant i \leqslant m^g, \qquad b_i^g = \sum_{j=p_i}^{p_{i+1}-1} b_j.$$

Предлагаем следующие способы разбивки l на группы.

По близости длин заготовок. В каждую группу попадают заготовки с близкими значениями длин. Для оценки близости используется некоторая константа $0 \leqslant \Delta \leqslant L$. При этом

$$\begin{aligned}
&G_1 = (l_1, \ldots, l_{p_2-1}), \quad l_1 - l_i \leqslant \Delta,\ 1 \leqslant i \leqslant p_2 - 1, \quad l_1 - l_{p_2} > \Delta,\\
&G_2 = (l_{p_2}, \ldots, l_{p_3-1}), \quad l_{p_2} - l_i \leqslant \Delta,\ p_2 \leqslant i \leqslant p_3 - 1, \quad l_{p_2} - l_{p_3} > \Delta,\\
&\qquad\qquad\qquad\qquad\qquad\qquad \ldots\\
&G_{m^g} = (l_{p_{m^g}}, \ldots, l_m).
\end{aligned}$$

Пример 1. $E = (L = 100,\ \ m = 4,\ \ \mathbf{l} = (99, 93, 83, 80),\ \ \mathbf{b} = (5, 2, 15, 10))$, $\Delta = 8$, $E^g = (L = 100,\ \ m^g = 2,\ \ \mathbf{l}^g = (99, 83),\ \ \mathbf{b}^g = (7, 25))$.

Проведём оценку разности решений $Z(E^g) - Z(E)$ в худшем случае. При $\Delta < \min(l_i - l_{i+1})$ группировочная задача E^g эквивалентна исходной задаче E, поэтому $Z(E) = Z(E^g)$.

Покажем, что $Z(E^g) - Z(E) \leqslant n - 1$. При $\Delta \geqslant l_1 - l_m$ имеем предельный (худший) случай группировочной задачи E^g. Она содержит

только один тип заготовок длины l_1 в количестве $n = \sum\limits_{i}^{m} b_i$. Обозначим $r = \lfloor L/l_1 \rfloor$. Тогда в этом случае $Z(E^g) = 1/r$.

Очевидно, что $Z(E) \geqslant 1/r$, тогда $Z(E^g) - Z(E) \leqslant (n-1)/r \leqslant n-1$. Достижимость этой оценки покажем на примере.

Пусть $E = (L,\ \ m = 2,\ \ \mathbf{l} = (l_1, l_2 = (L - l_1)/n),\ \ \mathbf{b} = (1, n-1))$, где $L \geqslant l_1 > L/2$. При этом $Z(E) = 1$. Для $\Delta \geqslant (nl_1 - L)/(n-1)$ группировочная задача имеет вид $E^g = (L,\ \ m^g = 1,\ \mathbf{l}^g = (l_1),\ \mathbf{b}^g = (n))$. Так как $l_1 > L/2$, то $Z(E^g) = n$. Следовательно, $Z(E^g) - Z(E) = n - 1$.

Как следует из приведённых оценок, разность $Z(E^g) - Z(E)$ зависит от выбора Δ. В качестве Δ мы использовали среднее значение разности длин заготовок, которое равно $\Delta = (l_1 - l_m)/(m-1)$.

По частоте. В одну группу объединяются заготовки, которые одинаковое число раз помещаются в прутке. При этом

$$
\begin{aligned}
&G_1 = (l_1, \ldots, l_{p_2 - 1}), \ \left[\frac{L}{l_1}\right] = \left[\frac{L}{l_i}\right], \quad 1 \leqslant i \leqslant p_2 - 1, \ \left[\frac{L}{l_1}\right] \neq \left[\frac{L}{l_{p_2}}\right], \\
&G_2 = (l_{p_2}, \ldots, l_{p_3 - 1}), \ \left[\frac{L}{l_{p_2}}\right] = \left[\frac{L}{l_i}\right], \quad p_2 \leqslant i \leqslant p_3 - 1, \ \left[\frac{L}{l_{p_2}}\right] \neq \left[\frac{L}{l_{p_3}}\right], \\
&\qquad\qquad\qquad\qquad \ldots \\
&G_{m^g} = (l_{p_{m^g}}, \ldots, l_m), \ \left[\frac{L}{l_{p_{m^g}}}\right] = \left[\frac{L}{l_i}\right], \quad p_{m^g} \leqslant i \leqslant m^g.
\end{aligned}
$$

Пример 2. $E = (L = 100,\ \ m = 5,\ \ \mathbf{l} = (29, 28, 26, 24, 22),\ \ \mathbf{b} = (10, 10,\ 10, 10, 10))$, $E^g = (L = 100,\ \ m^g = 2,\ \ \mathbf{l}^g = (29, 24),\ \ \mathbf{b}^g = (30, 20))$.

Оценим значение $Z(E^g) - Z(E)$ для данного способа группировки в худшем случае. Обозначим $q_i = [L/l_i^g]$. Тогда из способа построения стартового решения (3) для непрерывной релаксации задачи E^g следует, что

$$Z(E^g) \leqslant \sum_{i=1}^{m^g} \frac{b_i^g}{q_i}. \tag{4}$$

Значение $\sum\limits_{i=1}^{m} b_i l_i / L$ является «материальной» нижней границей для задачи E: $Z(E) \geqslant \sum\limits_{i=1}^{m} b_i l_i / L$.

Из неравенства $1/q_i \geqslant l_i^g/L > 1/(q_i + 1)$ и способа группировки получается

$$Z(E) \geqslant \sum_{i=1}^{m} \frac{b_i l_i}{L} \geqslant \sum_{i=1}^{m^g} \frac{b_i^g}{q_i + 1}. \tag{5}$$

Тогда из (4) и (5) следует

$$Z(E^g) - Z(E) \leqslant \sum_{i=1}^{m^g} \frac{b_i^g}{q_i} - \sum_{i=1}^{m^g} \frac{b_i^g}{q_i + 1} = \sum_{i=1}^{m^g} \frac{b_i^g}{q_i(q_i + 1)} \leqslant \frac{n}{2}.$$

Смешанный способ. Используются совместно оба способа группировки. При этом оценка в худшем случае будет точно такой, как и при втором способе группировки.

Докажем, что любое допустимое базисное решение задачи непрерывной релаксации (2) при исходных данных E^g можно преобразовать в допустимое базисное решение задачи непрерывной релаксации для исходных данных E. Эти действия назовём *процедурой разгруппировки.*

2.2. Процедура разгруппировки. Задача этой процедуры заключается в преобразовании решения задачи E^g в решение для E. Это всегда возможно сделать, так как каждой заготовке из E можно сопоставить заготовку не меньшей длины из E^g.

Без ограничения общности предположим, что задача E^g получена из исходной задачи E путём объединения первого и второго типов заготовок, т. е. $m^g = m - 1$, $l_1^g = l_1$, $b_1^g = b_1 + b_2$, $l_k^g = l_{k+1}$, $b_k^g = b_{k+1}$, $k = \overline{2, m^g}$. Это допущение можно принять, так как процедуру получения произвольной группировочной задачи E^g из E можно представить в виде цепочки

$$E = E_0^g \to E_1^g \to E_2^g \to \cdots \to E_{K_g-1}^g \to E_{K_g}^g = E^g, \tag{6}$$

где E_i^g отличается от E_{i+1}^g из (6) только объединением двух типов заготовок.

Пусть для E^g даны непрерывное базисное решение $\mathbf{x}^g = (x_1^g, x_2^g, ...x_{m^g}^g)$ (учитываются только ненулевые компоненты) и соответствующий ему набор базисных карт раскроя $A^g = (\mathbf{a}_1^g, \mathbf{a}_2^g, \ldots, \mathbf{a}_{m^g}^g)$.

Для разгруппировки необходимо преобразовать карты раскроя A^g в карты раскроя для E так, чтобы они образовали допустимое базисное решение. Для этого в A^g все заготовки из E^g, кроме первой, замещаются соответствующими заготовками из E. Это возможно сделать, так как они имеют одинаковые длины и комплектности. Карты раскроя, содержащие первую заготовку, разбиваются на два подмножества так, чтобы суммарное число первых заготовок в первом подмножестве было равно b_1, а во втором подмножестве — b_2, что достигается добавлением дополнительной карты. После этого в первом множестве первая заготовка

из E^g замещается первой заготовкой из E, а во втором множестве — второй заготовкой из E. Опишем эту процедуру подробнее.

Без ограничения общности предположим, что в A^g первыми стоят карты раскроя, не содержащие первой заготовки. Пусть число таких карт равно t, т. е. $a^g_{1j} = 0$, $1 \leqslant j \leqslant t$, и $a^g_{1j} > 0$, $t < j \leqslant m^g$.

Так как вектор $\mathbf{x}^g$ является решением (2), то

$$\sum_{j=1}^{m^g} a^g_{1j} \cdot x^g_j = \sum_{j=t+1}^{m^g} a^g_{1j} \cdot x^g_j = b^g_1 = b_1 + b_2.$$

Следовательно, найдётся номер $k > t$ такой, что $\sum\limits_{j=t+1}^{k} a^g_{j1} \cdot x^g_{j_t} \geqslant b_1$ и $\sum\limits_{t}^{k-1} a^g_{j_t 1} \cdot x^g_{j_t} < b_1$, где $t + 1 \leqslant k \leqslant m^g$.

Тогда процедура разгруппировки имеет следующий вид:

$$\begin{aligned}
&\mathbf{a}_j = \left(0, 0, a^g_{2j}, a^g_{3j}, \ldots, a^g_{m^g j}\right)^T, \quad x_j = x^g_j, \quad 1 \leqslant j \leqslant t,\\
&\mathbf{a}_j = \left(a^g_{1j}, 0, a^g_{2j}, a^g_{3j}, \ldots, a^g_{m^g j}\right)^T, \quad x_j = x^g_j, \quad t < j < k,\\
&\mathbf{a}_k = \left(a^g_{1k}, 0, a^g_{2k}, a^g_{3k}, \ldots, a^g_{m^g k}\right)^T, \quad x_k = \frac{b_1 - \sum\limits_{i=1}^{k-1} x^g_i}{a^g_{k1}},\\
&\mathbf{a}_j = \left(0, a^g_{1j}, a^g_{2j}, a^g_{3j}, \ldots, a^g_{m^g j}\right)^T, \quad x_j = x^g_j, \quad k < j \leqslant m^g,\\
&\mathbf{a}_{m^g+1} = \left(0, a^g_{1k}, a^g_{2k}, a^g_{3k}, \ldots, a^g_{m^g k}\right)^T, \quad x_{m^g+1} = x^g_k - x_k.
\end{aligned} \tag{7}$$

В результате действия процедуры получаем $m = m^g + 1$ допустимых карт раскроя для задачи E.

Описанные выше пребразования наглядно представляются в матричном виде:

$$A^g = \begin{pmatrix}
0 & \ldots & 0 & a^g_{1(t+1)} & \ldots & a^g_{1k} & \ldots & a^g_{1m^g}\\
a^g_{21} & \ldots & a^g_{2t} & a^g_{2(t+1)} & \ldots & a^g_{2k} & \ldots & a^g_{2m^g}\\
. & . & . & . & . & . & . & .\\
a^g_{m^g 1} & \ldots & a^g_{m^g t} & a^g_{m^g t+1} & \ldots & a^g_{m^g k} & \ldots & a^g_{m^g m^g}
\end{pmatrix},$$

$$A = \left(\begin{array}{cccccccccc|c}
0 & \ldots & 0 & a^g_{1(t+1)} & \ldots & a^g_{1k} & 0 & \ldots & 0 & & 0\\
\hline
0 & \ldots & 0 & 0 & \ldots & 0 & a^g_{1(k+1)} & \ldots & a^g_{1m^g} & & a^g_{1k}\\
\hline
a^g_{21} & \ldots & a^g_{2t} & a^g_{2(t+1)} & \ldots & a^g_{2k} & a^g_{2(k+1)} & \ldots & a^g_{2m^g} & & a^g_{2k}\\
. & . & . & . & . & . & . & . & . & & .\\
a^g_{m^g 1} & \ldots & a^g_{m^g t} & a^g_{m^g (t+1)} & \ldots & a^g_{m^g k} & a_{m^g (k+1)} & \ldots & a^g_{m^g m^g} & & a^g_{m^g k}
\end{array}\right).$$

Теорема 1. *Матрица A, построенная из невырожденной матрицы A^g по правилу (7), является невырожденной.*

Доказательство. Рассмотрим произвольную невырожденную матрицу C порядка m:

$$C = \begin{pmatrix} c_{11} & c_{12} & \dots & c_{1k} & \dots & c_{1m} \\ c_{21} & c_{22} & \dots & c_{2k} & \dots & c_{2m} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ c_{m1} & c_{m2} & \dots & c_{mk} & \dots & c_{mm} \end{pmatrix}.$$

Проведём с ней ряд преобразований. Сначала построим матрицу размера $(m+1)\times(m+1)$ по следующему правилу. Пусть $c_{1k} \neq 0$. Копируем k-й столбец матрицы C как $(m+1)$-й столбец матрицы C', начиная со второго элемента: $c'_{i(m+1)} = c_{ik}$, $i = 2, \dots, m$. Определим $c'_{1(m+1)} = 0$, $c'_{(m+1)(m+1)} = c_{1k}$. Все элементы $(m+1)$-й строки новой матрицы за исключением последнего элемента равны 0 ($c'_{(m+1)p} = 0$, $p = 1, \dots, m$):

$$C' = \left(\begin{array}{ccccccc|c} c_{11} & c_{12} & \dots & c_{1k} & c_{1(k+1)} & \dots & c_{1m} & 0 \\ c_{21} & c_{22} & \dots & c_{2k} & c_{2(k+1)} & \dots & c_{2m} & c_{2k} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ c_{m1} & c_{m2} & \dots & c_{mk} & c_{m(k+1)} & \dots & c_{mm} & c_{mk} \\ \hline 0 & 0 & \dots & 0 & 0 & \dots & 0 & c_{1k} \end{array}\right).$$

В матрице C' поменяем местами элементы 1-й и $(m+1)$-й строк с $(k+1)$-й до m-й позиции, получаем новую матрицу C'':

$$C'' = \left(\begin{array}{ccccccc|c} c_{11} & c_{12} & \dots & c_{1k} & 0 & \dots & 0 & 0 \\ c_{21} & c_{22} & \dots & c_{2k} & c_{2(k+1)} & \dots & c_{2m} & c_{2k} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ c_{m1} & c_{m2} & \dots & c_{mk} & c_{m(k+1)} & \dots & c_{mm} & c_{mk} \\ \hline 0 & 0 & \dots & 0 & c_{1(k+1)} & \dots & c_{1m} & c_{1k} \end{array}\right).$$

Найдём определитель матрицы C'', для этого к 1-й строке прибавим $(m+1)$-ю строку, а затем вычтем из $(m+1)$-го столбца k-й столбец. Мы получили определитель блочного вида, который равен

$$\det C'' = c_{1k} \cdot \det C \neq 0.$$

Видим, что с точностью до перестановки строк матрица C'' совпадает с матрицей A, если положить $C = A^g$, что доказывает теорему 1.

Следствие 1. *Вектор* $\mathbf{x}$ *и карты раскроя* $A = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m)$, *полученные по правилу* (7), *являются допустимым базисным решением для задачи* (2) *с исходными данными* E.

Пример 3. Пусть $E = (180, 4, \mathbf{l} = (99, 93, 83, 78), \mathbf{b} = (5, 2, 15, 10))$. Сгруппируем эту задачу первым способом ($\Delta = 7$): $(99, 5), (93, 2) \to (99, 7)$, $(83, 15), (78, 10) \to (83, 25)$.

Тогда $E^g = (180, 2, (99, 83), (7, 25))$. Решение непрерывной задачи для E^g имеет следующий вид:

$$A^g = \begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix}, \qquad x^g = (7, 9).$$

Разгруппировываем первый тип заготовок $(99, 7)$, а затем второй $(83, 25)$:

$$A_1^g = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \quad x_1^g = (5, 9, 2),$$
$$A_2^g = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 2 \end{pmatrix}, \quad x_2^g = (5, 5, 2, 4).$$

Получившиеся на последнем шаге значения $\left(A_2^g,\ x_2^g\right)$ являются допустимым базисным решением для исходной задачи E.

Исходя из вышесказанного предлагаем следующую схему использования метода группировки для повышения эффективности решения задачи (2).

Пусть дана задача $E = (L, m, \mathbf{l}, \mathbf{b})$.

Шаг 1. Строится группировочная задача $E^g = (L, m^g, \mathbf{l}^g, \mathbf{b}^g)$.

Шаг 2. Находится оптимальное LP решение задачи E^g в виде базисной матрицы A^g и вектора $\mathbf{x}^g$.

Шаг 3. Из A^g и $\mathbf{x}^g$ методом разгруппировки строятся A и $\mathbf{x}$.

Шаг 4. A и $\mathbf{x}$ используются в качестве начального решения для нахождения непрерывного решения задачи E.

Пример 4. $E = (L = 1000,\ m = 27,\ (399, 2),\ (396, 2),\ (393, 2),\ (387, 1),$
$(385, 2),\ (381, 1),\ (377, 1),\ (375, 2),\ (362, 1),\ (358, 3),\ (351, 1),\ (345, 1),$
$(342, 1),\ (338, 2),\ (331, 3),\ (325, 1),\ (322, 1),\ (316, 1),\ (303, 4),\ (298, 1),$
$(294, 2),\ (290, 2),\ (285, 1),\ (278, 3),\ (269, 2),\ (263, 5),\ (252, 2))$.

Группировочная задача, полученная с помощью первого способа группировки ($\Delta = 5{,}6$): $E^g = (L = 1000,\ m = 12,\ (399, 4),\ (393, 5),\ (381, 4),\ (362, 5), (345, 2), (338, 5), (325, 2), (316, 1), (303, 7), (290, 3), (278, 5), (263, 7))$.

Симплекс-алгоритм решает задачу E^g за 16 итераций $Z(E^g) = 16{,}67$. После разгруппировки A^g и $\mathbf{x}^g$ получившиеся A и $\mathbf{x}$ используются в качестве начального решения для E. Для получения оптимума необходимо сделать одну итерацию симплекс-алгоритма.

Для сравнения, задача E решается симплекс-алгоритмом без применения метода группировки за 45 итераций.

3. Численный эксперимент

Для численного эксперимента использовались случайно сгенерированные примеры, которые были разбиты на классы, предложенные в [9]. Класс задаётся набором $P = \{h, j, k, n, m\}$. Задача $E = (L, m, l, b)$ генерируется следующим образом: $L = k,\ h \leqslant l_i \leqslant j,\ \sum\limits_{i=1}^{m} b_i = n$. Для теста из [9] были взяты три основных класса $P1 = \{1, L/2, L, n, m\}$, $P2 = \{L/6, L/2, L, n, m\}$, $P3 = \{L/4, L/2, L, n, m\}$ и добавлен четвёртый $P4 = \{L/3, 2/3L, L, n, m\}$.

Для каждого класса были сгенерированны задачи с $m = (400, 1600, 3200)$ и $n = (m, 10000, 1000000)$ ($n = m$ означает, что в этих задачах вектор $\mathbf{b}$ единичный).

В качестве базового симплекс-метода был использован свободно распространяемый пакет COIN-LP (www.coin-or.org). Для сравнения каждый пример был решён тремя различными способами группировки, представленными в разд. 2.1, и без неё. Все примеры решались до получения оптимального решения с точностью до 10^{-6}. Начальные решения для симплекс-алгоритма строились с помощью метода «первый подходящий (FFD)». Результаты эксперимента приведены в табл. 1 и 2. Первая колонка содержит число различных типов заготовок, вторая (ТГ) — используемый тип группировки (0 — без группировки, 1,2,3 — соответствующие способы группировки), m^g — среднее число типов заготовок в группировочной задаче E^g, Ит.Г. — среднее число итераций, за которое была решена E^g, Рз. — средняя разница в процентах между оптимальным решением группировочной задачи и оптимумом самой задачи (в случае, когда задача решалась без группировки, приведена разница между начальным решением, построенным с помощью FFD и полученным оптимумом), Ит. — среднее число итераций, за которое была решена основная задача, Вр. — общее время решения задачи.

Эксперимент проводился на AMD 64 (Dual 2.21 GHz).

Основным параметром для оценки скорости получения решения предлагается использовать число итераций симплекс-алгоритма и затраченное время.

Как видно из табл. 1 и 2, второй способ группировки всегда проигрывает первому и третьему, и, как правило, его использование приводит только к увеличению числа итераций, поэтому он не может быть рекомендован к использованию.

Результаты, полученные для первого и третьего способов группировки несильно отличаются друг от друга. Это можно объяснить тем, что для случайно сгенерированных задач получаемое число типов заготовок m^g группировочной задачи слабо зависит от выбора способа группировки (первого или третьего).

Для классов $P1$, $P2$, $P4$ применение группировки приводит к уменьшению времени решения, при этом наиболее ощутимый результат получается для задач с большим значением $m = 3200$ (до 3,5 раз в классе $P2$).

Для класса $P3$ использование группировки приводило только к увеличению времени. Это объясняется тем, что данный класс относится к FFD — лёгким классам [14] и получаемое начальное решение несильно отличается от оптимального. В этом случае использовать алгоритм группировки не имеет смысла.

Также стоит заметить, что результат решения группировочной задачи (полученной первым или третьим способом) отличается от оптимума на сотые доли процента, поэтому данный способ можно использовать как приближённое решение для непрерывной релаксации (2).

4. Заключение

Мы видим, что представленный в работе метод группировки позволяет для ряда классов задач существенно сокращать время получения непрерывного решения для задачи линейного раскроя. Основным достоинством метода является его независимость от способа реализации прямого симплекс-метода, решающего непрерывную задачу линейного раскроя 1dCSP.

Автор благодарит сотрудников Института вычислительной математики Дрезденского технического университета (Германия) и лично Г. Белова и Г. Шайтхауэра за плодотворное сотрудничество, помощь в проведении численного эксперимента.

Т а б л и ц а 1

Результат численного эксперимента для классов $P1$ и $P4$

		$P1 = (5001, 9999, 20000, m, n)$					$P4 = (3333, 13333, 20000, m, n)$				
m	ТГ	m^g	Ит.Г.	Рз., %	Ит.	Вр.	m^g	Ит.Г.	Рз., %	Ит.	Вр.
$n = m$											
400	0	-	-	7	747	23,5	-	-	1,9	474	5,9
	1	202	366	0,06	563	23,5	204	185	0,08	243	4
	2	2	1	14	815	25,6	5	1	19,5	785	9,7
	3	203	369	0,06	562	23,5	206	186	0,08	240	4
1600	0	-	-	7	2098	319	-	-	1,7	2034	172
	1	754	1164	0,01	1208	267	784	1114	0,01	770	130
	2	2	1	14	2340	355	5	1	20	3422	289
	3	754	1171	0,01	1197	265	789	1137	0,01	755	132
3200	0	-	-	7	3642	1441	-	-	1,62	4450	1529
	1	1600	2075	0,005	1736	1097	1502	1391	0,02	2327	1007
	2	2	1	14	4138	1637	5	1	20	7470	2566
	3	1600	2079	0,005	1738	1098	1525	1410	0,02	2009	903
$n = 10000$											
400	0	-	-	7	753	33,1	0	0	2	684	14,8
	1	199	352	0,06	525	30,7	200	297	0,1	555	16,7
	2	2	1	14	805	35,3	5	1	20	1120	24,2
	3	200	348	0,06	525	30,7	201	310	0,1	550	17,1
1600	0	-	-	6	2169	337	-	-	1,8	2834	512
	1	751	1200	0,01	1129	262	784	1560	0,02	2013	638
	2	2	1	7	2418	375	5	1	20	4951	807
	3	751	1195	0,01	1128	262	795	1593	0,02	1954	639
3200	0	-	-	6	3822	1506	-	-	1,7	6427	4146
	1	1611	2144	0,005	1716	1101	1520	1659	0,021	4870	3696
	2	2	1	6	4344	1711	5	1	19	10497	6771
	3	1611	2148	0,005	1717	1102	1593	1724	0,02	4801	3696
$n = 1000000$											
400	0	-	-	7	757	23,3	-	-	1,9	684	13,1
	1	201	361	0,06	529	21,8	204	302	0,1	566	12,3
	2	2	1	13	799	24,5	5	1	21	1139	21,8
	3	204	366	0,05	519	21,7	206	307	0,1	551	12,1
1600	0	-	-	7	2113	322	-	-	1,7	2962	615
	1	747	1167	0,015	1097	250	781	1542	0,027	2223	537
	2	2	1	14	2330	355	5	1	20	5237	1087
	3	752	1180	0,014	1085	249	790	1577	0,026	2189	534
3200	0	-	-	7	3678	1439	-	-	1,6	5019	3308
	1	1606	2100	0,005	1518	1006	1506	2789	0,014	3601	2780
	2	2	1	14	4106	1606	5	1	20	9893	6520
	3	1606	2099	0,005	1510	1002	1535	2859	0,013	3538	2765

Т а б л и ц а 2

Результат численного эксперимента для классов $P2$ и $P3$

		$P2=(1667,4999,10000,m,n)$					$P3=(1,4999,10000,m,n)$				
m	ТГ	m^g	Ит.Г.	Рз., %	Ит.	Вр.	m^g	Ит.Г.	Рз., %	Ит.	Вр.
					$n=m$						
400	0	-	-	4,7	332	3,0	-	-	0,012	52	6,7
	1	200	170	0,06	38	1,11	200	19	0,164	76	9,5
	2	4	1	16	481	4,3	48	45	15	514	66
	3	205	217	0,039	14	1,13	231	19	0,12	68	9
1600	0	-	-	5	1583	103	-	-	0,002	181	115
	1	732	752	0,0002	1	22,4	751	61	0,047	302	200
	2	4	1	16	2167	171	93	173	15	2414	1534
	3	766	792	0,0002	1	24,7	838	139	0,036	208	156
3200	0	-	-	5	3643	1309	-	-	0,0006	176	184
	1	1270	1624	0,01	451	1607	1607	199	0,015	167	227
	2	4	1	17	4848	1741	129	440	15	5055	4702
	3	1353	1685	0,01	452	418	1661	200	0,014	159	222
					$n=10000$						
400	0	-	-	5,27	733	18,1	-	-	0,12	462	63,35
	1	197	404	0,07	435	15,6	197	290	0,12	481	75,6
	2	4	1	16,9	882	21,7	46	182	15	928	127,5
	3	202	467	0,06	409	15,9	228	410	0,09	430	77,2
1600	0	-	-	5	1924	205	-	-	0,02	967	640
	1	721	865	0,02	914	138	749	519	0,04	1666	1177
	2	4	1	16	2573	274	91	481	15	3287	2176
	3	767	909	0,02	896	141	828	420	0,04	1574	1116
3200	0	-	-	5	3597	1174	-	-	0,006	1300	1756
	1	1269	1534	0,005	487	357	1597	440	0,017	1637	2359
	2	4	1	16	4939	1612	127	486	15	6250	8443
	3	1336	1613	0,006	442	364	1649	426	0,017	1631	2355
					$n=1000000$						
400	0	-	-	4,9	943	34,5	-	-	0,13	818	137
	1	199	579	0,06	629	33,5	201	476	0,12	819	157
	2	4	1	17,4	1104	40,3	47	198	16	1367	229
	3	205	586	0,06	592	32,6	232	610	0,09	772	163
1600	0	-	-	5,1	2874	671	-	-	0,03	2489	2009
	1	722	1517	0,019	1493	508	748	1369	0,03	2555	2413
	2	4	1	17	3381	789	90	466	15	4725	2393
	3	750	1593	0,017	1455	514	828	1712	0,02	2420	2631
3200	0	-	-	5	5075	3766	-	-	0,014	4229	6952
	1	1268	2398	0,01	2667	2684	1619	2455	0,012	4201	7939
	2	4	1	17	4555	8046	126	677	15	8861	14568
	3	1370	2594	0,01	2665	4113	1662	2630	0,012	4194	8060

ЛИТЕРАТУРА

1. **Гери М. П., Джонсон Д. С.** Вычислительные машины и трудноразрешимые задачи. — М.: Мир, 1982. — 419 с.
2. **Канторович Л. В., Заллгаллер В. А.** Расчет рационального раскроя материалов.— Л.: Лениздат. 1951. — 199 с.
3. **Картак В. М.** Достаточные условия невыполнения свойства целочисленного округления для задачи линейного раскроя // Автоматика и телемеханика. — 2004. — № 4. — С. 55–62.
4. **Мухачева Э. А., Рубинштейн Г. Ш.** Математическое программирование. — Новосибирск: Наука, 1987. — 272 с.
5. **Applegate D. L., Buriol L. S., Dillard B. L., Johnson D. S., Shor P. W.** The cutting-stock approach to bin packing: theory and experiments // Proc. the fifth workshop on algorithm engineering and experimentation. — SIAM, 2003. — P. 1–15.
6. **Belov G., Scheithauer G.** A branch-and-cut-and-price algorithm for one-dimensional stock cutting and two-dimensional two-stage cutting // Europ. J. Oper. Research. — 2006. — Vol. 171, N 1. — P. 85–106.
7. **Claudio A., Valerio de Carvalho J. M.** A stabilized branch-and-price-and-cut algorithm for the multiple length cutting stock problem // Comput. Oper. Research. — 2008. — Vol. 35, N 4. — P. 1315—1328.
8. **Dyckhoff H.** A typology of cutting and packing problems // Europ. J. Oper. Research. — 1990. — Vol. 44, N 2. — P. 145–159.
9. **Fernandez de la Vega W., Lueker G. S.** Bin packing can be solved within $1+e$ in linear time // Combinatorica. — 1981. — N 1. — P. 349–355.
10. **Gilmore P., Gomory R.** A linear programming approach to the cutting-stock problem // Oper. Research. — 1961. — N 9. — P. 849–859.
11. **Marcotte O.** An instance of the cutting stock problem for which the rounding property does not hold // Oper. Research Letters. — 1986. — Vol. 4, N 5. — P. 239–243.
12. **Mukhacheva E., Belov G., Kartak V., Mukhacheva A.** Linear one-dimensional cutting-packing problems: numerical experiments with the sequential value correction method (SVC) and a modified branch-and-bound method (MBB) // Pesquisa Operacional. — 2000. — Vol. 20, N 2. — P. 153–168.
13. **Peeters M., Degraeve Z.** Optimal integer solutions to industrial cutting-stock problems: part 2. Benchmark Results // Inform. J. Computing. — 2003. — Vol. 15, N 1. — P. 58–81.
14. **Schwerin P., Wascher G.** The bin-packing problem: A problem generator and some numerical experiments with FFD packing and MTP // Intern. Transactions Oper. Research. — 1997. — Vol. 4, N 5/6. — P. 377–389.

15. **Vanderbeck F.** Computational study of a column generation algorithm for bin packing and cutting stock problems // Math. Programming. — 1999. — Vol. 86, N 3. — P. 565—594.

Картак Вадим Михайлович,
e-mail: kvmail@mail.ru

Статья поступила
19 июня 2008 г.
Переработанный вариант —
29 апреля 2009 г.