Уфа : УГАТУ, 2008 Вестник УГАТУ • Управление, ВТиИ Т. 10, № 2 (27). С. 154–158

УДК 004.421

В. М. КАРТАК

ОБНОВЛЕННАЯ НИЖНЯЯ ГРАНИЦА ДЛЯ ЗАДАЧИ УПАКОВКИ ПРЯМОУГОЛЬНИКОВ В ПОЛУБЕСКОНЕЧНУЮ ПОЛОСУ

Рассматривается задача упаковки прямоугольников в полубесконечную полосу (two-dimensional strip packing problem, 2DSPP), которая имеет широкое применение в промышленности. Она является NP-трудной. Рассматриваются алгоритмы расчета нижней границы, базирующиеся на ее матричном представлении и линейном программировании. *Метод ветвей и границ ; раскрой-упаковка ; нижняя граница*

Задача упаковки прямоугольников в полубесконечную полосу (или двумерная задача раскроя-упаковки, two-dimensional strip packing problem, 2DSPP), состоит в следующем: требуется разместить заданные прямоугольники в полубесконечной полосе так, чтобы длина занятой части полосы достигла минимума. Согласно Dyckhoff's типологии раскроя и упаковки [1,], двумерная задача раскроя-упаковки имеет тип 2/N/O.

Эта задача является NP-трудной, поэтому алгоритмы, нацеленные на получение оптимального решения, представляют собой различные модификации метода ветвей и границ [2].

Нижней границей решения данной задачи называется число, величина которого не превосходит длины полосы оптимальной упаковки [3]. Наличие хороших нижних границ очень важно для сокращения перебора в точных методах, а также для оценки приближенных решений.

Самой простой нижней границей мы можем считать отношение суммарной площади всех прямоугольников к ширине полосы.

Другие способы подсчета нижней границы представлены в работе Martello и Toth [3]. Они показали, что в случаях, когда число малых элементов мало, имеет смысл сконцентрировать внимание на больших элементах.

S. P. Feket и *J. Schepers* [4] предложили использовать двойственно выполнимые функции для вычисления нижней границы. Идея этого подхода заключается в следующем: пусть $R = \{R_i, i = \overline{1,m}\}$ — набор прямоугольников некоторой задачи 2DSPP, а F — двойственно выполнимая функция, тогда любая нижняя граница для преобразованной задачи 2DSPP с размерами элементов $F(R) = \{F(R_i), i = \overline{1,m}\}$ также является нижней границей для исходной задачи.

Представленные нижние границы относятся к классу элементарных нижних границ и вычисляются за время $O(m\log(m))$, но при этом разность между полученным значением и оптимальным может быть существенная.

A. Bortfeld предложил использовать линейное программирование для получения нижней границы [5]. Для этого задача 2DSPP сопоставляется с задачей линейного раскроя с дополнительными ограничениями.

В данной статье предложен улучшенный подход к определению нижней границы, базирующийся на ее матричном представлении и дальнейшем использовании методов линейного программирования для оценки оптимального решения.

1. ПОСТАНОВКА ЗАДАЧИ

Имеется полубесконечная полоса S фиксированной ширины W и множество прямоугольников $R = \{R_i, i = \overline{1,m}\}$, имеющих размеры (l_i, w_i), где l_i — длина, w_i — ширина i-го прямоугольника, m — количество прямоугольников. Обозначим исходные данные как (W, m, R). Требуется разместить прямоугольники в полосе так, чтобы длина занятой части полосы достигла минимума.

Допустимой упаковкой прямоугольников R в полубесконечную полосу S называется размещение прямоугольников внутри полосы S, удовлетворяющее следующим условиям:

- стороны прямоугольников из R параллельны сторонам полосы S;

• прямоугольники из R между собой не перекрываются (не имеют общих внутренних точек).

Введем прямоугольную систему координат, в которой оси O_x, O_y совпадают с нижней и левой сторонами полосы, и обозначим допустимую упаковку через $P = \{P_1, P_2, ...P_m\}$, где $P_i = (x_i, y_i)$ — координаты левого нижнего угла i-го прямоугольника. Допустимая упаковка P_o, для которой длина занятой части полосы достигает минимума, называется *оптимальной*.

2. МАТРИЧНОЕ ПРЕДСТАВЛЕНИЕ УПАКОВКИ

Пусть для некоторой задачи (W, m, R) известна упаковка P. Выполним мысленно вертикальные «резы», проходящие через правые стороны прямоугольников R [6]. Таким образом, вся упаковка делится на вертикальные полосы. Обозначим k_x — число вертикальных полос. Аналогично выполним мысленно горизонтальные резы, проходящие через верхние стороны прямоугольников, k_y — число горизонтальных полос.

Вертикальному разбиению упаковки на полосы сопоставим следующую пару (A^x, z^x), где: A^x —матрица размерности $m \times k_x$, элементы которой вычисляются по правилу:

$$a^x_{ij} = \begin{cases} 1 - \text{если} \quad i\text{-й прямоугольник} \\ \text{пересекается } j\text{-й вертикальной полосой,} \\ 0 - \text{в противном случае;} \end{cases}$$

$z^x = (z^x_1, z^x_2, \ldots, z^x_{k_x})$ — вектор, элементами которого являются значения длин вертикальных полос.

Очевидно, что «единицы» в строках матрицы A^x идут непрерывно. Это свойство еще называется «продолженностью единиц» [5].

Аналогично горизонтальному разбиению сопоставим матрицу A^y размерности $m \times k_y$ и вектор $z^y = (z^y_1, z^y_2, \ldots, z^y_{k_y})$.

Полученные указанным способом матрицы A^x и A^y названы *матрицами упаковки*, а векторы z^x и z^y — *векторами упаковки*.

Связь между упаковкой P и матрицами упаковок определяют следующие утверждения.

Утверждение 1. (*Критерий существования допустимой упаковки*) *Для того чтобы существовала допустимая прямоугольная упаковка, длина которой не превышала бы L, необходимо и достаточно, чтобы выполнялись следующие условия:*

• *существуют допустимые* (A^x, z^x) и (A^y, z^y),где $\sum_{i=1}^{k_x} z^x_i \le L$ и $\sum_{i=1}^{k_y} z^y_i \le W$;

• *если в некотором столбце k_0 матрицы A^x имеются строки i_1 и i_2 такие, что $a^x_{i_1 k_0} = 1$* и $a^x_{i_2 k_0} = 1$, *то для любого столбца $1 \le k \le k_y$ матрицы* A^y верно $a^y_{i_q k} \cdot a^y_{i_2 k} = 0$ (т. е. $a^y_{i_1 k}$ и $a^y_{i_2 k}$ *одновременно не могут быть равны* 1).

Утверждение 2. *Для нахождения оптимальной упаковки достаточно найти допустимые пары* (A^x, z^x) и (A^y, z^y) *такие, что* $\sum_{i=1}^{k_x} z^x_i \to \min$.

3. НИЖНЯЯ ГРАНИЦА НА БАЗЕ ЛИНЕЙНОГО РАСКРОЯ

Последнее утверждение можно интерпретировать следующим образом: в задаче 2DSPP в качестве нижней границы можно использовать оценку величины $\min \sum_{i=1}^{k_x} z^x_i$, на значение которой влияет только пара (A^x, z^x). (Пара (A^y, z^y) необходима для проверки допустимости построенной упаковки). Подсчитаем нижнюю границу, используя аналогию между задачей построения пары (A^x, z^x) и задачей линейного раскроя.

Задача линейного раскроя (Cutting Stock Problem, 1DCSP). Имеется материал, поступающий в виде стержней длины H. Путем его раскроя требуется получить набор из m различных предметов заданных длин λ_i, $i = \overline{1,m}$ и в необходимом количестве b_i каждого вида $i = \overline{1,m}$. Требуется раскроить материал на линейные предметы (заготовки) с минимальными затратами материала [8].

Добавим в задачу 1DCSP дополнительное условие: потребуем, чтобы раскрой стержня содержал только одну заготовку одного типа. Обозначим новую задачу (Binary Cutting Stock Problem, 1DBCS) как $E = (H, m, \lambda, \mathbf{b})$; $\lambda = (\lambda_1, \ldots, \lambda_m)$; $b = (b_1, \ldots, b_m)$.

Пусть вектор $\alpha^j = (a_{1j}, a_{21}, \ldots, a_{mj})^T \in Z + m$, где $a_{ij} \in \{0, 1\}$ описывает j-ю карту раскроя (способ раскроя стержня): его компонента $a_{ij} = 1$, если заготовка типа i получается из данного стержня, и $a_{ij} = 0$ в противном случае. Матрица $A = (a_{ij})$, $i = \overline{1,m}$; $j = \overline{1,N}$ называется *раскройной матрицей*, здесь N — число всевозможных карт раскроя. Обозначим через x_j, $j = \overline{1,N}$ *интенсивность* применения карты раскроя j, т. е. число прутков, которые должны быть разделены в соответствии с картой раскроя α^j. Тогда задача

1DBCSP сводится к решению следующей задачи линейного целочисленного программирования:

$$\begin{aligned} Z^(E) &= \sum_{j=1}^{N} x_j \to \min, \\ Ax &= b, \quad x \in Z + N. \end{aligned} \tag{1}$$

Для получения нижней границы оптимального значения задачи (1) рассмотрим аналогичную задачу линейного программирования, в которой отсутствует условие целочисленности компонент вектора x:

$$\begin{aligned} Z(E) &= \sum_{j=1}^{N} x_j \to \min, \\ Ax &= b, \quad x \in R + N. \end{aligned} \tag{2}$$

Поскольку число N для реальных задач достаточно велико, то задача (2) решается симплекс-методом с неявно заданной матрицей ограничений. При этом максимальное число ненулевых элементов x_j не превосходит заданного числа m (размера базисного множества).

Вернемся к задаче 2DSPP. Пусть заданы исходные данные (W, m, R) для прямоугольной упаковки. Если положить $H = W$; $\lambda_i = w_i$; $b_i = l_i$; $i = \overline{1, m}$, то задаче построения пары (A^x, z^x) можно сопоставить 1DBCSP с параметрами E^v=($H = W$; m; $\lambda = w$; $b = l$). Назовем эту задачу, *вертикальной задачей линейного раскроя* (Vertical Cutting Stock Problem, VCSP). Соответствующая ей задача линейного программирования имеет вид:

$$\begin{aligned} Z(E^v) &= \sum_{j=1}^{N} x_j \to \min, \\ A_v x &= l, \quad x \in R + N. \end{aligned} \tag{3}$$

Очевидно, что любое допустимое заполнение матриц (A^x, z^x) даст нам допустимое решение соответствующей задачи VCS. Обратное не верно, так как решение VCSP не удовлетворяет свойству 2 («продолженности единиц»). Тем не менее, любая нижняя граница для задачи VCSP будет также являться нижней границей для $\min \sum\limits_{i=1}^{k_x} z_i^x$, а следовательно, и для задачи 2DSPP. Таким образом, в качестве нижней границы можно использовать решение (3).

4. УЛУЧШЕННАЯ НИЖНЯЯ ГРАНИЦА

Представленную в предыдущем разделе нижнюю границу можно улучшить. Для этого определим *горизонтальную задачу линейного раскроя* (Horizontal Cutting Stock Problem, HCSP) для горизонтального разбиения упаковки как соотвествующую задачу 1DBCS с параметрами $E^h = (H = L;\ m;\ \lambda = l;\ \mathbf{b}{=}w)$. Соответствующая задача линейного программирования имеет вид:

$$\begin{aligned} Z(E^h) &= \sum_{j=1}^{M} y_j \to \min, \\ A_h y &= w, \quad y \in R + M, \end{aligned} \tag{4}$$

где L — длина занятой полосы, A_h — матрица, состоящая из всех возможных карт раскроя, соответствующих горизонтальному разбиению, M — число таких векторов.

Обозначим через L_d текущую нижнюю границу для RP. Очевидно, что если для E^h будет верно $Z(E^h) > W$, это будет означать, что исходный набор прямоугольников не может быть упакован в полосу длины L_d и ширины не более W. В этом случае первое условие утверждения 1 не выполняется, L_d нужно увеличить.

Так как размеры прямоугольников целочисленные, то и длина занятой части полосы может быть только целым значением, поэтому увеличивать L_d можно на единицу. Базируясь на этом факте можно предложить следующий алгоритм расчета начальной нижней границы.

Алгоритм 1.
Расчет начальной нижней границы

Шаг 0. Дано W, $w = (w_1, ..., w_m)$, $l = (l_1, \ldots, l_m)$ (*входная информация для 2DSPP*).

Шаг 1. $L_d \leftarrow \lceil Z(E^v) \rceil$ (*начальное значение для нижней границы*).

Шаг 2. Если $Z(E^h) > W$, то $L_d = L_d + 1$ переход к **Шагу 1**. (*увеличение нижней границы*).

Шаг 3. Стоп, L_d — текущая нижняя граница.

Алгоритм 1 использует только первое условие из утверждения 1. Для проверки второго условия теоремы введем дополнительное ограничение в задачу (4), которое запрещает использовать в решении карты раскроя, содержащие фиксированную пару элементов

(p, q) (см. также [9]).

$$
\begin{aligned}
&Z_{(p,q)}(E^v) = \sum_{j=1}^{N} x_j \to \min, \\
&A_v x = l, \quad x \in R + N, \quad x_i = 0, \\
&i \in \left\{ j \,|\, (a^j_{vp} = 1)\,\&\,(a^j_{vq} = 1) \right\}.
\end{aligned} \tag{5}
$$

Данная задача линейного программирования будет рассматривать только карты раскроя a^j, не содержащие элементы с номерами p и q. Обозначим за $Z_{(p,q)}(E^v)$ оптимальное решение задачи (5).

Аналогично определим $Z_{(p,q)}(E^h)$ для следующей задачи:

$$
\begin{aligned}
&Z_{(p,q)}(E^h) = \sum_{j=1}^{M} y_j^{\to} \min, \\
&A_h y = w, \quad y \in R + M, \quad y_i = 0, \\
&i \in \left\{ j \,|\, (a^j_{hp} = 1)\,\&\,(a^j_{hq} = 1) \right\}.
\end{aligned} \tag{6}
$$

Утверждение 3. *Пусть для некоторого значения L существует пара (p, q), такая, что $Z_{(p,q)}(E^v) > L$ и $Z_{(p,q)}(E^h) > W$. Тогда не существует упаковки P, длина занятой части полосы которой не превосходит L.*

Следовательно, если для некоторого значения нижней границы L_d выполняется условие утверждения 3, то не существует упаковки, размещающейся в полосе длины не более L_d, а следовательно, нижнюю границу L_d нужно увеличивать.

Обозначим за $Q = \{(p_k, q_k)\}$ множество пар элементов с номерами p_k и q_k Анологично (5) и (6) определим :

$$
\begin{aligned}
&Z_Q(E^v) = \sum_{j=1}^{N} x_j^{\to} \min,, \\
&A_v x = l, \quad x \in R + N \quad x_i = 0, \\
&i \in \left\{ j \,|\, (a^j_{vp_k} = 1)\,\&\,(a^j_{vq_k} = 1) \right\}
\end{aligned} \tag{7}
$$

$$
\begin{aligned}
&Z_Q(E^h) = \sum_{j=1}^{M} y_j^{\to} \min, \\
&A_h y = w, \quad y \in R + M \quad y_i = 0, \\
&i \in \left\{ j \,|\, (a^j_{hp_k} = 1)\,\&\,(a_h {q_k}^j = 1) \right\}.
\end{aligned} \tag{8}
$$

Утверждение 4. *Пусть для некоторого значения L выполняется одно из двух условий:*

$Z_{Q^v}(E^h) > W$,

где Q^v={$(p_k, q_k) | Z_{(p,q)}(E^v) > L$},

$Z_{Q^h}(E^v) > L$,

где $Q^h = \{(p_k, q_k) | Z_{(p,q)}(E^h) > W\}$.

Тогда не существует упаковки RP, длина занятой части полосы которой не превосходит L.

Базируясь на утверждении 4, можно предложить следующий алгоритм расчета нижней границы.

Алгоритм 2.
Расчет улучшенной нижней границы

Шаг 0. Дано W, $w = (w_1, \ldots, w_m)$, $l = (l_1, \ldots, l_m)$ — *входная информация для* 2DSPP. $Q^v = 0$.

Шаг 1. $L_d \leftarrow$ Алгоритм 1 — *начальное значение для нижней границы*.

Шаг 2. Для каждой пары элементов (p_k, q_k), входящих в решение VCSP, выполняем: если $Z_{(p,q)}(E^v) > L$, *то* $Q^v = Q^v + (p_k, q_k)$.

Шаг 3. Если $Z_{Q^v}(E^h) > W$, то $L_d = L_d + 1$, $Q^v = 0$, переход к **Шагу 2**.

Шаг 4. Для каждой пары элементов (p_k, q_k), входящих в решение HCSP выполняем: если $Z_{(p,q)}(E^h) > L$, то $Q^h = Q^h + (p_k, q_k)$.

Шаг 5. Если $Z_{Q^h}(E^v) > W$, то $L_d = L_d + 1$, $Q^h = 0$, переход к **Шагу 4**.

Шаг 6. Стоп, L_d — текущая нижняя граница.

Замечания:

1) пара (p_k, q_k) входит в решение **VCSP (HCSP)**, если в оптимальном решении задачи линейного программирования (3) существует карта раскроя j , такая, что $(a^j_{hp_k} = 1)\,\&\,(a^j_{hq_k} = 1)$.

2) для расчета величины $Z_{(p,q)}(E^v)$ лучше воспользоваться двойственным симплекс-методом, так как решение задачи $Z_(E^v)$ без добавочного ограничения уже известно.

3) так как число шагов представленного алгоритма зависит от числа возможных пар (p, q) (их число растет как m^2), то вычислительная сложность будет $O(m^2) \cdot P(m)$, где $P(m)$ — трудоемкость выполнения шага двойственного симплекс-метода, которая для данной задачи псевдополиномиальна.

5. ЧИСЛЕННЫЙ ЭКСПЕРИМЕНТ

Численный эксперимент был проведен следующим образом: точный алгоритм решения задачи 2DSPP [7]. В качестве ниж-