

С.А. ЭКСУЗЯН

ОДНОМЕРНАЯ ЗАДАЧА РАСКРОЯ ЛЕНТ РАЗНЫХ ТИПОРАЗМЕРОВ И АЛГОРИТМ ПРИБЛИЖЕННОГО РЕШЕНИЯ

Разработаны приближенный алгоритм РАС и соответствующая про-грамма для решения одномерной задачи раскроя из лент, имеющих разные типоразмеры. Приведен анализ результатов работы программы алгоритма РАС на 1000 случайных примеров.

Ключевые слова: одномерная задача, NP-полная задача, приближенный алгоритм, раскрой, упаковка, сложность алгоритма.

Введение. Одномерная задача раскроя лент традиционно рассматривается для лент, имеющих одинаковую ширину (типоразмеры). Задача заключается в следующем: даны ширина и количество по ширинам заказов, которые необходимо раскраивать из заготовок, имеющих одинаковую ширину. Эта задача эквивалентна известной задаче об "упаковке в контейнерах", которая является NP-трудной в сильном смысле [1]. Поэтому алгоритмы определения близких к оптимальным вариантам раскроев лент имеют очевидное практическое применение. Для задачи об "упаковке в контейнерах" разработаны приближенные алгоритмы FF, BF, FFD, BFD [1,3], которые можно применять для решения одномерной задачи раскроя из лент, имеющих одинаковую ширину.

В [7] рассмотрена диалоговая версия задачи упаковки контейнеров с открытым концом, в которой контейнер может быть заполнен к уровню, превышающему его вместимость последним предметом. Эта задача является NP-трудной.

В [8] представлены приближенные алгоритмы FFF и FFH решения диалоговой версии задачи об "упаковке в контейнерах" при типах контейнеров, имеющих разные размеры. Однако в случае, когда все контейнеры имеют размер 1 (одинаковые размеры), эти алгоритмы совпадают с алгоритмом FF.

В диалоговых версиях задач предметы прибывают один за другим и должны быть назначены на контейнеры по мере их прибывания. Как только предмет упакован, его снова перемещать непозволительно.

В настоящей работе поставлена более общая задача одномерного раскроя из лент, имеющих разные типоразмеры (ширину). Эта задача

представляет определенный интерес как с теоретической, так и с практической точек зрения.

Для поставленной общей задачи в настоящей работе разработаны приближенный алгоритм РАС и соответствующая программа. Приведен анализ результатов работы программы алгоритма РАС на 1000 случайных примеров.

1. Основные определения и обозначения. Систему непустых подмножеств $C_1, C_2, \dots, C_p$ множества S назовем разбиением множества S , если выполнены $S = C_1 \cup C_2 \cup \dots \cup C_p$ и $C_i \cap C_j = \emptyset$ при $C_i \neq C_j$.

Задача об “упаковке в контейнерах” формулируется следующим образом: заданы конечное множество $U = \{u_1, u_2, \dots, u_n\}$ “предметов” и “размеры” $S(u) \in [0, 1]$ каждого $u \in U$ (размер “предмета” представлен рациональным числом); требуется найти такое разбиение множества U на непересекающиеся подмножества $U_1, U_2, \dots, U_k$, чтобы сумма размеров “предметов” в каждом подмножестве U_i не превосходила 1 и чтобы k было наименьшим возможным. Можно считать, что “предметы”, принадлежащие каждому множеству U_i , упаковываются в один контейнер размера 1, а наша цель – упаковать “предметы” множества U в как можно меньшее число контейнеров.

Эта задача NP-трудна в сильном смысле, так что мало надежды на отыскание эффективного алгоритма ее решения. Поэтому приближенные алгоритмы для решения этой задачи имеют очевидное применение. Разработаны приближенные алгоритмы FF, BF, FFD, BFD [1,3].

В алгоритме FF очередной предмет u_k помещается в контейнер, имеющий минимальный индекс из числа тех, которые подходят для размещения u_k .

В алгоритме BF очередной предмет u_k помещается в такой контейнер, который после помещения в него u_k будет иметь по размеру наименьшее неиспользованное место.

В алгоритме FFD предметы упорядочивают по невозрастанию размеров, и применяется алгоритм FF.

В алгоритме BFD предметы упорядочивают по невозрастанию размеров, и применяется алгоритм BF.

Обозначим через x целую часть числа x .

Количество контейнеров, требуемых в любом S алгоритме упаковки в контейнеры, обозначим через $C(L)$ при работе со списком предметов L , а оптимальное решение – через L^* .

Обозначим $R_C(\alpha) = \overline{\lim}_{L \rightarrow +\infty} C(L)/L^*$, где L пробегает все списки, в которых все элементы a_i удовлетворяют условию $a_i \in (0, \alpha]$.

В [3] доказано, что для всех списков L справедливы оценки $FF(L) \leq 17L^*/10+2$ и $BF(L) \leq 17L^*/10+2$.

В [2] доказана следующая теорема.

Теорема. Если K - положительное целое число, то существует такое множество предметов L , размеры которых - конечные дроби, $K=L^*$ и $FF(L)=BF(L)=[1.7L^*]$.

В [5] и [6] доказана следующая теорема.

Теорема. Для всех списков $L=[a_1, a_2, \dots, a_n]$ имеют место соотношения:

- 1) $BFD(L) \leq FFD(L)$, если $a_i \in [1/6, 1]$ и $i=1, 2, \dots, n$;
- 2) $BFD(L)=FFD(L)$, если $a_i \in [1/5, 1]$ и $i=1, 2, \dots, n$.

В [4] и [5] доказан следующий результат:

$$R_{FFD}(\alpha) = \begin{cases} 11/9 & \text{при } \alpha \in (1/2, 1] \\ 71/60 & \text{при } \alpha \in (8/29, 1/2] \\ 7/6 & \text{при } \alpha \in (1/4, 8/29] \\ 23/20 & \text{при } \alpha \in (1/5, 1/4]. \end{cases}$$

2. Постановка задачи. Задача об "упаковке в контейнерах" эквивалентна следующей одномерной задаче раскроя из лент, имеющих одинаковую ширину (размеры). Задано конечное множество $U=\{u_1, u_2, \dots, u_n\}$ заказов с размерами $S(u_i) \in (0, 1]$; $i=1, 2, \dots, n$ (размер заказа представлен рациональным числом), которыми необходимо раскроить ленты. Требуется найти такие $U_1, U_2, \dots, U_k$ подмножества для множества U , для которых выполнялось бы $U = \bigcup_{i=1}^k U_i$ и

$U_i \cap U_j = \emptyset$ ($i \neq j$ ($i=1, 2, \dots, k$; $j=1, 2, \dots, k$), чтобы сумма размеров заказов в каждом подмножестве не превосходила бы 1 и k было бы наименьшим возможным (все ленты имеют одинаковые размеры).

Можно считать, что заказы, принадлежащие каждому множеству U_i , раскраиваются из одной ленты, а наша цель - раскроить заказы множества U из как можно меньшего числа лент. Очевидно, что при этом сумма размеров неиспользованных мест использованных лент будет наименьшей возможной.

Рассмотрим следующую задачу раскроя лент при наличии типов лент разных размеров. Задача заключается в следующем: заданы конечное множество $U=\{u_1, u_2, \dots, u_n\}$ заказов с размерами $S(u_i) \in (0, 1]$; $i=1, 2, \dots, n$ (заказы из U , имеющие одинаковый размер, будем считать разными) и множество конечных типов лент $K=\{K_1, K_2, \dots, K_m\}$, имеющих разные размеры $S(K_j) \in (0, 1]$ ($j=1, 2, \dots, m$), такие, что $\max_{1 \leq j \leq m} \{S(K_j)\} \geq \max_{1 \leq i \leq n} \{S(u_i)\}$. Требуется найти такие $U_1^{T_1},$

$U_2^{T_2}, \dots, U_p^{T_p}$ подмножества для множества U , где $T_1, T_2, \dots, T_p \in K$, для которых выполнялось бы: $U = \bigcup_{i=1}^p U_i^{T_i}$; $U_i^{T_i} \cap U_j^{T_j} = \emptyset$ при $i \neq j$ ($i = 1, 2, \dots, p$;

$j=1, 2, \dots, p$); $\sum_{u \in U_i^{T_i}} S(u) \leq S(T_i)$ ($i=1, 2, \dots, p$) и $\sum_{i=1}^p (S(T_i) - \sum_{u \in U_i^{T_i}} S(u))$ была бы наименьшей возможной (сумма размеров неиспользованных мест исполь-

зованных лент была наименьшей возможной). Все заказы из множества $U_i^{T_i}$ раскраиваются на одной ленте типа T_i .

3. Приближенные алгоритмы решения задачи. Пусть имеются заказы $u_1, u_2, \dots, u_e$. Тип ленты K_j назовем заполняемым при заказах $u_1, u_2, \dots, u_e$, если существует натуральное число $t(j)$ ($t(j) < e$), при котором выполняются следующие два условия:

$$1) \sum_{i=1}^{t(j)} S(u_i') \leq S(K_j); \quad 2) \sum_{i=1}^{t(j)+1} S(u_i') > S(K_j).$$

При этом разность $OCT(K_j) = S(K_j) - \sum_{i=1}^{t(j)} S(u_i')$ назовем остатком при заполнении.

Пусть при заказах $u_1, u_2, \dots, u_e$ все типы лент K_j ($j=1, 2, \dots, m$) являются заполняемыми. Скажем, что тип ленты K_h ($1 \leq h \leq m$) удовлетворяет 1-му условию предпочтительности, если для него выполняется условие: $OCT(K_h) = \min \{ OCT(K_j) \}$.

$$1 \leq j \leq m$$

Ленту назовем использованной, если на ней раскроен хотя бы один заказ.

Использованную ленту L назовем подходящей для заказа u с количеством v , если $1 \leq [OCT(L)/S(u)] \leq v$.

Скажем, что для нераскроенных заказов u с количеством v подходящая лента L_e удовлетворяет 2-му условию предпочтительности, если для него выполняется условие: $OCT(L_e) - S(u) [OCT(L_e)/S(u)] = \min \{ OCT(L) - S(u) [OCT(L)/S(u)] \}$, где минимум берется по всем подходящим лентам L .

Пусть имеются нераскроенные заказы u с количеством v . Скажем, что тип ленты K_e ($1 \leq e \leq m$) удовлетворяет 3-му условию предпочтительности для нераскроенных заказов u с количеством v , если выполняется условие: $[v / (S(K_e) - S(u) [S(K_e)/S(u)])] \leq \min \{ [v / (S(K_j) - S(u) [S(K_j)/S(u)])] \}$, где минимум берется по всем типам лент K_j ($j=1, 2, \dots, m$), для которых $[S(K_j)/S(u)] \leq v$.

Алгоритм РАС. (Раскрой лент при наличии типов лент разных размеров).

Вход: Массив K для размеров разных типов лент (размерность m), массив U для разных размеров заказов (размерность n_1), массив V для количества заказов по разным размерам (размерность n_1).

Выход: Схема раскроя на типах лент, имеющих разные типоразмеры.

РАС1. Обнулить очередь ОЧЕР (размерность –

$$2 \max_{1 \leq j \leq m} \{S(K_j)\} / \min_{1 \leq i \leq n_1} \{S(u_i)\}.$$

РАС2. Упорядочить массивы K и U по невозрастанию размеров (при обмене элементов массива U местами обменять местами и их соответствующие элементы в массиве B). Установить $i \leftarrow 1$.

РАС3. Установить $v \leftarrow B(i)$.

РАС4. Если $v=0$, то перейти к шагу РАС12.

РАС5. Если нет подходящих лент для заказа u_i с количеством v , то перейти к шагу РАС7.

РАС6. Найти подходящую ленту L , удовлетворяющую 2-му условию предпочтительности. Раскраивать на ней заказы u_i по количеству $(OCT(L)/S(u_i))$. Установить $v \leftarrow v - [OCT(L)/S(u_i)]$. Перейти к шагу РАС4.

РАС7. Если $v < [S(K_1)/S(u_i)]$, то перейти к шагу РАС9.

РАС8. Выбрать тип лент K_e по 3-му условию предпочтительности для заказов u_i с количеством v . На $(v/(S(K_e)/S(u_i)))$ неиспользованных лентах типа K_e раскраивать заказы u_i с количеством $(S(K_e)/S(u_i))$ на каждой. Установить $v \leftarrow v - [v /$

$[S(K_e)/S(u_i)]] [S(K_e)/S(u_i)]$. При $v=0$ перейти к шагу РАС12.

РАС9. Вставить заказы u_i с количеством v в очередь ОЧЕР.

РАС10. Если не все типы лент заполняемы при всех заказах из очереди ОЧЕР, то перейти к шагу РАС12.

РАС11. Заказы из очереди ОЧЕР раскраивать на типе ленты (неиспользованной), выбранной по 1-му условию предпочтительности. Эти заказы снять из очереди ОЧЕР. Перейти к шагу РАС10.

РАС12. Установить $i \leftarrow i+1$. Если $i \leq n1$, то перейти к шагу РАС3.

РАС13. Если в очереди ОЧЕР есть заказы, то их раскраивать на таком типе ленты (неиспользованной), при котором все заказы из очереди ОЧЕР раскраиваются вместе и остается наименьшее неиспользованное место.

РАС14. Работа алгоритма РАС завершена.

Так как количество заказов - конечное число, а при каждом шаге раскроя по алгоритму РАС из множества заказов удаляются заказы, то работа алгоритма РАС завершается после выполнения шагов конечного числа.

$$\text{Пусть } n = \sum_{i=1}^{n1} B(i)$$

Оценим сложность алгоритма РАС.

Шаг РАС1 занимает время не более, чем $O(n)$. Шаг РАС2 занимает время $O(m^2)+O(n^2)$. Шаг РАС3 занимает время $O(n)$. Шаг РАС4 занимает время $O(n)$. Шаг РАС5 занимает время $O(n^2)$. Шаг РАС6 занимает время $O(n^2)$. Шаг РАС7 занимает время $O(n)$. Шаг РАС8 занимает время $O(mn)$. Шаг РАС9 занимает время $O(n)$. Шаг РАС10 занимает время $O(mn^2)$. Шаг РАС11 занимает время $O(mn^2)$. Шаг РАС12 занимает время $O(n)$. Шаг РАС13 занимает время $O(mn)$.

Следовательно, алгоритм РАС имеет временную сложность порядка $O(mn^2) + O(m^2)$.

Если в множестве заказов имеются заказы с одинаковыми размерами, то эти заказы будем считать разными.

Каждому решению одномерной задачи раскроя заказов из типов лент, имеющих разные типоразмеры, соответствует такое разбиение множества заказов, каждому подмножеству которого соответствует один типоразмер лент так, что сумма размеров заказов каждого подмножества не превосходит типоразмер, присвоенный этому подмножеству (все заказы, раскраиваемые из одной ленты, образуют одно подмножество, а заказы, раскраиваемые из разных лент, принадлежат разным подмножествам).

Справедливо и обратное утверждение. Любому разбиению множества заказов, каждому подмножеству которого соответствует один типоразмер лент и сумма размеров всех заказов каждого подмножества которого не превосходит соответствующий типоразмер, соответствует одно решение одномерной задачи раскроя заказов из типов лент, имеющих разные типоразмеры (предполагается, что все заказы любого подмножества раскраиваются из одной ленты, имеющей типоразмер, соответствующий данному подмножеству).

Докажем, что с помощью алгоритма РАС получается решение одномерной задачи раскроя заказов из типов лент, имеющих разные типоразмеры (необязательно оптимальное). Для этого докажем, что схеме раскроя, получаемой алгоритмом РАС, соответствует такое разбиение множества заказов, каждому подмножеству которого соответствует один типоразмер лент так, что сумма размеров всех заказов каждого подмножества не превосходит типоразмер, присвоенный этому подмножеству.

По алгоритму РАС раскрой лент реализуется только в шагах РАС6, РАС8, РАС11 и РАС13, в которых реализуется раскрой из неиспользованных и использованных лент.

В начале работы алгоритма РАС в качестве совокупности подмножеств выберем пустое множество.

В шаге РАС8 тип (типоразмер) ленты однозначно определяется по 3-му условию предпочтительности. Из $[v/[S(K_e)/S(u_i)]]$ неиспользованных лент этого типоразмера раскраиваются $[S(K_e)/S(u_i)]$ заказы размера $S(u_i)$ с каждого. В смысле подмножеств это означает: добавить к совокупности подмножеств $[v/[S(K_e)/S(u_i)]]$ подмножества, каждое из которых содержит заказ u_i с количеством $(S(K_e)/S(u_i))$ и типоразмер, выбранный по 3-му условию предпочтительности, поставить в соответствие каждому добавленному подмножеству. Все заказы, раскроенные в этом шаге, удаляются из множества заказов.

В шаге РАС11 тип (типоразмер) ленты и заказы из очереди однозначно определяются по 1-му условию предпочтительности. Из одной неиспользованной ленты определяемого типоразмера раскраиваются все определяемые заказы. В смысле подмножеств это означает: добавить к совокупности подмножеств одно подмножество, которое содержит только все

заказы, выбранные из очереди по 1-му условию предпочтительности, и типоразмер лент, выбранный по этому же условию, поставить в соответствие этому подмножеству. Все заказы, раскроенные на этом шаге, удаляются из очереди, следовательно, и из множества заказов.

В шаге PAC13 тип (типоразмер) ленты определяется одно-значно, и все заказы из очереди раскраиваются из одной неиспользованной ленты такого типоразмера. В смысле подмножеств это означает: добавить к совокупности подмножеств одно подмножество, которое содержит только все заказы из очереди, и определенный типоразмер поставить в соответствие этому подмножеству. После выполнения этого шага совокупность подмножеств уже организована (работа алгоритма завершена).

В шаге PAC6 заказы раскраиваются из использованных лент. Использованная лента однозначно выбирается по 2-му условию предпочтительности. В смысле подмножеств это означает: подмножеству, однозначно выбранному из совокупности подмножеств, добавить все заказы, раскроенные на этом шаге. Типоразмер, соответствующий этому подмножеству, не меняется. Все заказы, раскроенные на этом шаге, удаляются из множества заказов.

Полученная совокупность подмножеств не содержит пустого подмножества, так как при выполнении любого из шагов PAC6, PAC8, PAC11 и PAC13 к совокупности подмножеств добавляется новое непустое подмножество, или добавляются заказы к имеющимся подмножествам. Сечение любых двух подмножеств из совокупности подмножеств есть пустое множество, так как любой раскроенный за-каз удаляется из множества заказов. Следовательно, полученная совокупность подмножеств является разбиением для множества заказов, каждому подмножеству которого соответствует один типоразмер ленты.

Так как по алгоритму PAC сумма размеров всех заказов, раскроенных из любой ленты, не превосходит типоразмер этой ленты, а этот типоразмер и типоразмер соответствующего подмножества одинаковы, то сумма размеров всех заказов любого подмножества не превосходит типоразмер, соответствующий этому подмножеству.

Это означает, что алгоритм PAC решает задачу одномерного раскроя заказов из типов лент разных типоразмеров.

Рассмотрим алгоритм PAD, имеющий шаги PAD1, PAD2, ..., PAD14, который отличается от алгоритма PAC только тем, что на шаге PAD8 тип ленты выбирается не по 3-му условию предпочтительности, а по 1-му условию (остальное все в обоих алгоритмах выполняется одинаково).

Докажем, что совокупность всех заказов, раскроенных по схеме алгоритма PAD из одной любой ленты, невозможно раскраивать из одной ленты, имеющей типоразмер меньше, чем типоразмер этой ленты. Предположим обратное. Пусть в схеме раскроя, полученной алгоритмом PAD, существует тип ленты K_j , из которой раскраиваются заказы $u_{i1}, u_{i2}, \dots, u_{it}$, и эти заказы можно раскроить из типа ленты K_p , для которого выполняется условие $S(K_p) < S(K_j)$.

По алгоритму PAD эта лента типа K_j , первый раз использовалась на одном из шагов PAD8, PAD11, PAD13. Предположим, что при первом использовании этой ленты из него раскраивались заказы $u_{i1}, u_{i2}, \dots, u_{ig}$. Очевидно, что $g \leq t$.

Если эта лента типа K_j первый раз использовалась на шаге PAD13, то $g=t$, т.е. возникает противоречие, заключающееся в том, что на этом шаге K_j есть тип лент, имеющий наименьший размер, с которого можно было раскраивать $u_{i1}, u_{i2}, \dots, u_{ig}$.

Пусть эта лента типа K_j первый раз использовалась на шаге PAD8 (PAD11). Так как $S(u_{i1}) + S(u_{i2}) + \dots + S(u_{it}) \leq S(K_j)$ и по предположению имеем $S(u_{i1}) + S(u_{i2}) + \dots + S(u_{it}) \leq S(K_p)$, то $S(u_{i1}) + S(u_{i2}) + \dots + S(u_{ig}) \leq S(K_p)$. Это противоречит выбору типа ленты по условию предпочтительности 1 на шаге PAD8 (PAD11). Полученные противоречия доказывают истинность утверждения.

Анализ результатов работы алгоритма PAC на 1000 случайных примеров показал, что отходы, получаемые алгоритмом PAC для решения общей задачи, который разработан в настоящей работе, составляют в среднем 4,8%, а отходы, получаемые тем же алгоритмом из лент, имеющих одинаковые размеры, составляют в среднем 17,71%, где в качестве отхода взята разность суммы размеров всех лент, использованных алгоритмом PAC, и суммы размеров всех заказов.

Разработанный алгоритм имеет временную сложность в порядке $O(m^2) + O(mn^2)$, где m - количество по ширине типов лент, n - количество заказов. В результате работы разработанной программы получается схема раскроя заказов из имеющихся типов лент.

Разработанный алгоритм можно применять в машиностроении, металлургии, приборостроении, легкой промышленности и других отраслях.

СПИСОК ЛИТЕРАТУРЫ

1. **Гэри М., Джонсон Д.** Вычислительные машины и труднорешаемые задачи. - М.: Мир, 1982. - 416 с.
2. **Ивани А.М.** Оценка эффективности алгоритмов упаковки в контейнеры // Сб. статей "Проблемы кибернетики". - 1984. - Вып. 41. - С. 253-256.
3. Теория расписаний и вычислительные машины / Под ред. **Э.Г. Коффмана**. - М.: Наука, 1984. - 334 с.
4. **Johnson D.S.** Near-Optimal Bin-Packing Algorithms // Ph. D. thesis. - Electrical Engineering Department, Massachusetts Institute of Technology, MA, 1974.
5. **Johnson D. S., Demers A., Ullman J. D., Garrey M. R., Graham R. L.** Worst - Case Performance Bounds for Simple One - Dimensional Packing Algorithms // SIAM Journal on Computing. - 1974. - №3. - P. 299-326.
6. **Garey M.R., Graham R. L., Ullman J. D.** Worst - Case Analysis of Memory Allocation Algorithms. Proceedings // 4th Annual ACM Symposium on the Theory of Computing. - 1972. - P. 143.
7. **Zhang G.** Parameterized On-Line Open-End Bin Packing // Computing 60, 267-273(1998).

8. **Zhang G.** Worst-Case Analysis of the FFH Algorithm For-Online Variable-Sized Bin Packing // Computing 56, 165-172(1996).

Ванадзорский педаг. ин-т им. О. Туманяна. Материал поступил в редакцию 8.01.2002.

Ս.Հ. ԷԶՍՈՒԶՅԱՆ
ՏԱՐԲԵՐ ՏԻՊԱԶԱՓԵՐԻ ԺԱՊԱՎԵՆՆԵՐԻՑ ՄԻԱԶԱՓ ՁԵՎՄԱՆ
ԽՆԴԻՐԸ ԵՎ ՄՈՏԱՎՈՐ ԼՈՒԾՄԱՆ ԱԼԳՈՐԻԹՄԸ

Տարբեր տիպաչափերի ժապավեններից միաչափ ձևման խնդրի լուծման համար մշակված է PAC մոտավոր ալգորիթմ եվ համապատասխան ծրագիր: 1000 պատահական օրինակների համար բերված է PAC ալգորիթմի ծրագրի աշխատանքի արդյունքների վերլուծությունը:

S. H. EKSUZYAN
PROBLEM OF ONE-DIMENSIONAL BAND CUTTING OUT FOR
DIFFERENT DIMENSION-TYPES AND AN APPROXIMATE ALGORITHM SOLUTION

An approximate PAC algorithm and a corresponding program are worked out for solving the problem of one-dimensional cutting out of bands having different dimension-types. An analysis of the results of PAC algorithm's program work based on 1000 random examples has been made.