Э. А. Мухачева, А. С. Мухачева, Задача прямоугольной упаковки: методы локального поиска оптимума на базе блочных структур, *Автомат. и телемех.*, 2004, выпуск 2, 101–112

Автоматика и телемеханика, № 2, 2004

© 2004 г. **Э. А. МУХАЧЕВА, д-р. техн. наук,**
А. С. МУХАЧЕВА, канд. физ.-мат. наук
(Уфимский государственный авиационный технический университет)

ЗАДАЧА ПРЯМОУГОЛЬНОЙ УПАКОВКИ: МЕТОДЫ ЛОКАЛЬНОГО ПОИСКА ОПТИМУМА НА БАЗЕ БЛОЧНЫХ СТРУКТУР[1]

Рассматривается задача ортогональной упаковки прямоугольников в полубесконечную полосу и ее представление блок-структурами, позволяющими сводить проблему к решению задач линейного раскроя специального вида. На этой базе предложены схемы конструирования методов локального поиска оптимума и разработаны детерминированный и вероятностные алгоритмы. Приведены результаты численного эксперимента, подтверждающие эффективность новых методов.

1. Введение

Под задачами раскроя-упаковки понимается широкий класс проблем, допускающих различное толкование. Общим является наличие двух групп объектов. Между элементами этих групп устанавливается и оценивается соответствие. Качественную типологию в области раскроя-упаковки провел в 1991 г. H. Dyckhoff [1]. Среди различных моделей важное место занимают задачи ортогональной упаковки прямоугольных объектов в заданных областях. Эти задачи принято именовать 1.5D Bin Packing Problem (**1.5DBPP**) в случае упаковки в полубесконечную полосу и **2DBPP** – в листы прямоугольной формы [2]. Предметом изучения в настоящей статье является **1.5DBPP**. Вместе с тем предлагаемые здесь алгоритмы размещения в полубесконечной полосе можно легко модифицировать на случай упаковки предметов в листы. Кроме того, роль вспомогательной задачи выполняет задача линейного раскроя 1D Cutting Stock Problem, (**1DCSP**). Этой задаче также уделено внимание в статье.

Задача **1.5DBPP** состоит в следующем. Имеется прямоугольная полоса фиксированной ширины и полубесконечной длины. Требуется разместить прямоугольные предметы в полосе так, чтобы стороны прямоугольников были параллельны сторонам полосы; прямоугольники не пересекались между собой и со сторонами полосы; длина занятой части полосы достигала минимума. Эта задача встречается при решении многих прикладных проблем экономики и производства. К ней сводятся задачи распределения двумерного ресурса; раскрой рулонного и листового материала; упаковка контейнеров в транспортные средства; размещение оборудования и другие. Каждая из перечисленных проблем может входить в оптимизационное ядро соответствующей автоматизированной системы. Например, раскрой рулонного или листового материала представляет подсистему заготовительного производства АСУ ТП.

[1] Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проект № 01-01-00510) и фонда президента Росийской Федерации (проект № МК 145.2003.01).

На заре появления проблемы раскроя Л.В. Канторовичем и В.А. Залгаллером было предложено использовать для решения задач раскроя линейное программирование [3], вернее – непрерывную релаксацию для решения целочисленных задач раскроя. Это позволило разработать эффективные методы расчета линейного и гильотинного раскроя в условиях массового производства. Аналогичные методы появились за рубежом [4, 5]. Для расчета раскроев на каждом шаге линейного программирования решается задача о загрузке рюкзака. Для ее решения был разработан метод склейки [6]. На базе линейного программирования были разработаны алгоритмы условной оптимизации [7]. Эти и другие работы по существу закрыли проблему массового раскроя. Задачи раскроя-упаковки являясь NP-трудными проблемами дискретного программирования, потребовали развития других методов для их решения. Для получения точного решения используются методы отсечения [8], где разработаны гибридные алгоритмы для решения задач линейного и гильотинного раскроя с использованием простых эвристик после построения каждого очередного отсечения Гомори. Прием позволяет в подавляющем большинстве случаев быстро получать оптимальное решение. Среди комбинаторных методов наибольшее применение получил метод "ветвей и границ" [9, 10]. Асимптотически точный подход разработан для некоторого класса задач линейной и прямоугольной упаковки [11]. Ввиду NP-трудности задач, вызывают интерес приближенные и эвристические методы локального поиска оптимума. В [12] проведен подробный обзор однопроходных эвристик для решения задачи упаковки контейнеров, описан принцип самого худшего случая и проведен анализ среднего случая, исследовано поведение офф-лайн и он-лайн вариантов эвристик. Среди эвристик более высокого уровня выделяются *жадные* алгоритмы [13]. К сложным уровневым алгоритмам относится способ *последовательно–одиночного* размещения [14]. Внесение в эвристику элементов случайности значительно повышает ее эффективность. Краткий обзор вероятностных методов локального поиска проведен [15]. С использованием элементов случайности разрабатываются бурно развивающиеся метаэвристики, характеристики и обоснования которых можно найти в [16]. Часто применяются для решения задач раскроя-упаковки *генетические* алгоритмы, например для **1DBPP** [17], для **1.5DBPP** [18] и другие. Оригинальный способ кодирования генов применен в [19]. Этот подход оказался весьма результативным. С разработкой вероятностных алгоритмов появилась потребность в исследовании их поведения в зависимости от случайных параметров [20]. Здесь предлагается новый подход для решения **1.5DBPP**, который базируется на методологии блочных структур упаковок. На этой базе конструируются различные группы алгоритмов. Они позволяют быстро получать близкое к оптимуму решение для различных классов исходной информации.

2. Математическая модель задачи и способы кодирования упаковки

Задача 1.5DBPP. Имеются прямоугольная полоса заданной ширины W и неограниченной длины и набор из m прямоугольных предметов заданных размеров $(w_i; l_i)$, $i = \overline{1, m}$, где w_i – ширина; l_i – длина стороны, параллельной неограниченной грани полосы. Введем прямоугольную систему координат: оси Ox и Oy совпадают соответственно с нижней неограниченной и боковой сторонами полосы. Положение каждого прямоугольника P_i зададим координатами $(x_i; y_i)$ его левого нижнего угла.

Набор векторов $(x_i; y_i)$, $i = \overline{1, m}$ называется *прямоугольной упаковкой* (Rectangular Packing, **RP**), если для $i \neq j$; $i, j = \overline{1, m}$,

(1) $$x_i \geqslant (x_j + l_j) \vee x_j \geqslant (x_i + l_i),$$

или

(2) $$y_i \geqslant (y_j + w_j) \vee y_j \geqslant (y_i + w_i);$$

для $i = \overline{1, m}$

(3) $\qquad x_i \geqslant 0 \wedge y_i \geqslant 0 \wedge y_i + w_i \leqslant W.$

Условие (1) означает *раздвинутость* прямоугольников по оси Ox, условие (2) – *раздвинутость* по оси Oy. Раздвинутость по оси Ox или по оси Oy означает непересечение прямоугольников между собой. Выполнение условий ((1) или (2)) и (3) означает допустимость упаковки. Если длина занятой части допустимой упаковки полосы достигает минимума, то **RP** называется *оптимальной* упаковкой и является решением **1.5DBPP**. Исходную информацию для **1.5DBPP** принято задавать вектором (W, m, w, l), $w = (w_1, w_2, \ldots, w_m)$, $l = (l_1, l_2, \ldots, l_m)$. Решение задачи определяет упаковка **RP** с координатами $(x_i; y_i)$, $i = \overline{1, m}$, прямоугольников.

Приведем два способа кодирования **RP**. Первый применяется многими авторами [18]. На втором способе базируются предлагаемые здесь алгоритмы.

Приоритетные списки. Предположим, что известна допустимая упаковка **RP**. В качестве шифра **RP** принято использовать список $\pi = \{1(\pi), 2(\pi), \ldots, i(\pi), \ldots \ldots, m(\pi)\}$, в котором $i(\pi)$ – номер прямоугольника, занимающего в π позицию i. Для фиксированного списка π с помощью того или иного алгоритма размещения (*декодера*) вычисляют координаты $(x_i; y_i)$ прямоугольника P_i и строят эскиз упаковки. Ее длина зависит от перестановки элементов в π и от используемого декодера [21].

Блок-структуры. Пусть имеется прямоугольная упаковка **RP**. Проведем через правые стороны прямоугольников вертикальные резы, они разбивают **RP** на прямоугольные *вертикальные блоки* одной и той же ширины W и различной длины. Пусть длина **RP** равна L. Проведем через верхние стороны прямоугольников горизонтальные линии. Тогда **RP** разобьется на *горизонтальные блоки* одной и той же длины L и различной ширины. Таким образом, мы получаем две блок-структуры для **RP**, вертикальную и горизонтальную. Каждому блоку j сопоставим *кортеж* (запись номеров прямоугольников, пересекающих блок) и длину χ_j вертикального, η_j горизонтального блоков. В качестве шифров блок-структур используем списки:

(4) $$S = \{1(j), 2(j), \ldots, i(j), \ldots\}\, \chi_j, \quad j = \overline{1, r};$$
$$\widetilde{S} = \{1(j), 2(j), \ldots, i(j), \ldots\}\, \eta_j, \quad j = \overline{1, q},$$

где $i(j)$ – номер прямоугольника в позиции i, пересекающего блок j, r – количество вертикальных, q – горизонтальных блоков. На рис. 1,*а* и 1,*б* изображены вертикальная и горизонтальная блок-структуры упаковки с $m = 6$. Разбиения на блоки

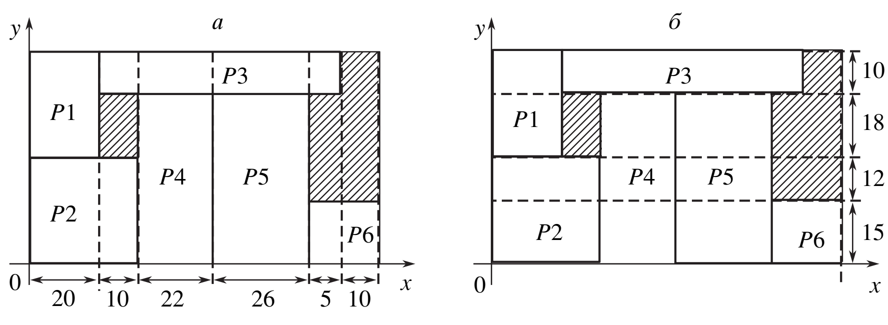

Рис. 1. Блок-структуры упаковок.

указаны штриховыми линиями. Там же указаны длины (ширины) блоков. Легко проверить, что блок-структурам отвечают следующие списки:

$$S = \{(2,1)20;\ (2,3)10;\ (4,3)22;\ (5,3)26;\ (6,3)5;\ (6)10\};$$
$$\widetilde{S} = \{(2,4,5,6)15;\ (2,4,5)12;\ (1,4,5)18;\ (1,3)10\}.$$

Длина $L(\mathbf{RP}) = 20+10+22+26+5+10 = 93$ и ширина $\widetilde{N}(\mathbf{RP}) = 15+12+18+10 = 55$.

Таким образом, имея упаковку, легко получить пару списков S и $\widetilde{S}$, отвечающих блок-структурам. Нас интересует обратная задача: по исходной информации $(W;m;w;l)$ найти списки $S(\widetilde{S})$, соответствующие упаковке. Обозначим: $I_j(\widetilde{I}_j)$ – множество прямоугольников, пересекающих j-й вертикальный (горизонтальный) блок; $I_j^+(\widetilde{I}_j^+)$ – множество прямоугольников $i \in I_j (i \in \widetilde{I}_j)$, заканчивающихся в j-м блоке; $I_j^-(\widetilde{I}_j^-)$ – множество прямоугольников $i \in I_j (i \in \widetilde{I}_j)$, начинающихся в j-м блоке. Определяющими для блок-структур являются следующие необходимые свойства соответствия:

Л е м м а 1. *Если список $S(\widetilde{S})$ соответствует допустимой упаковке* **RP** *задачи* **1.5DBPP**, *то он удовлетворяет следующим свойствам.*

1°. ***Разнородность*** *прямоугольников. Элементы $i(j)$ каждого кортежа j различные (один и тот же прямоугольник не может повторяться в одном кортеже);*

2°. ***Продолженность*** *прямоугольников. Если некоторый элемент $i(j)\overline{\in} I_j^+(\widetilde{I}_j^+)$, то $i(j) \in I_{j+1}(\widetilde{I}_{j+1})$ (если прямоугольник не заканчивается в кортеже j, то он продолжен в следующем кортеже).*

3. Задачи прямоугольно-ориентированного линейного раскроя

Базовой здесь является следующая хорошо известная проблема [22].

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

Задача **1DCSP** задается информационным вектором $(Z;m;\lambda;b)$; $\lambda = (\lambda_1, \dots \lambda_m)$; $b = (b_1, \dots b_m)$. Вектор $\alpha^j = (a_{1j}, a_{2j}, \dots, a_{mj})^T \in Z_+^m$ описывает j-й шаблон раскроя; компоненты a_{ij} указывают количество получаемых заготовок типа i. Матрица $A = (a_{ij})$, $i = \overline{1,m}$; $j = \overline{1,n}$, называется раскройной матрицей. Обозначим через χ_j, $j = \overline{1,n}$ количество стержней, раскраиваемых по шаблону j. Тогда проблема планирования оптимального раскроя материала сводится к решению следующей задачи:

$$\min N = \left\{ \sum_{j=1}^{n} \chi_j \;\middle|\; \chi = (\chi_1, \chi_2, \dots, \chi_n) \in Z_+^n; \sum_{j=1}^{n} \chi_j \alpha^j = b \right\}. \tag{5}$$

Если векторы α^j и количество n различных шаблонов известны, то (5) является задачей линейного целочисленного программирования. Однако в реальных задачах векторы α^j не заданы в явном виде и число n экспоненциально зависит от размерности m исходной задачи. Тогда (5) представляет задачу с неявно заданной матрицей ограничений, а неизвестными, кроме χ_j, являются векторы α^j (шаблоны) и число n. Известны методы решения этой задачи. Например, метод отсекающих плоскостей, приведенный в [8] и комбинаторные алгоритмы [22]. Здесь для решения (5) с дополнительными ограничениями применяются схемы локального поиска.

Заметим, что шаблон раскроя j может быть задан кортежем $(1(j), 2(j), \ldots$ $\ldots, i(j), \ldots)$, в котором перечислены номера $i(j)$ заготовок, получаемых по шаблону j. Тогда решение задачи (5) представляет совокупность из n различных кортежей с указанием количества χ_j стержней, раскраиваемых по j-му шаблону. Его можно записать как

$$S = (\{1(j), 2(j), \ldots, i(j), \ldots\})\,\chi_j, \quad j = \overline{1,n}; \quad N = \sum_{i=1}^{n} \chi_j, \tag{6}$$

где S – список кортежей, N – расход материала, вычисляется через S и используется в качестве рекорда.

Если положить $Z = W$; $\lambda_i = w_i$; $b_i = l_i$; $i = \overline{1,m}$, то **1.5DBPP** трансформируется в **1DCSP** с $(Z = W; m; \lambda = w; b = l)$. Пара $(S; N)$ определяет ее допустимое решение.

$$S = (1(j), 2(j), \ldots, i(j), \ldots)\,\chi_j, \quad j = \overline{1,r}, \quad N = \sum_{j=1}^{r} \chi_j, \tag{7}$$

где χ_j – количество стержней, раскраиваемых по кортежу (шаблону) j.

Если положить $Z = N$; $\lambda_i = l_i$; $b_i = w_i$; $i = \overline{1,m}$, то **1.5DBPP** трансформируется в задачу линейного раскроя **1DCSP*** с $(Z = N; m; \lambda = l; b = w)$. Пара $(\widetilde{S}; \widetilde{N})$ определяет ее допустимое решение

$$\widetilde{S} = (1(j), 2(j), \ldots, i(j), \ldots)\,\eta_j, \quad j = \overline{1,q}, \quad \widetilde{N} = \sum_{j=1}^{q} \eta_j, \tag{8}$$

где η_j – количество стержней, раскраиваемых по кортежу (шаблону) j.

Определим прямоугольно–ориентированные задачи линейного раскроя (Rectangular Oriented **CSP**, **RCSP**).

Задача RCSP. При исходных данных **1.5DBPP** найти допустимое решение $(S; N)$ задачи **1DCSP**, удовлетворяющее свойствам 1°, 2°.

Задача RCSP*. При исходных данных **1.5DBPP** и $Z = N$ найти допустимое решение $(\widetilde{S}; \widetilde{N})$ задачи **1DCSP***, удовлетворяющее свойствам 1°, 2° и условию

$$\widetilde{N} \leqslant W. \tag{9}$$

Допустимое решение задачи **RCSP(RCSP*)** называется *прямоугольно–ориентированным линейным раскроем* (Rectangular Oriented Linear Cutting, **ROLC (ROLC*)**), ему отвечает вертикальная (горизонтальная) блок-структура, заданная парой $(S; N)$ $((\widetilde{S}; \widetilde{N}))$.

Блок-структуры **ROLC (ROLC*)** можно найти, решая последовательно задачи **RCSP** и **RCSP***. Однако, только они не обеспечивают построения **RP**.

4. Алгоритмы на базе перестановок элементов в блоках

Здесь мы будем опираться только на одну, вертикальную блок-структуру.

Заметим, что решение **ROLC** линейного раскроя не зависит от порядка следования элементов в кортежах. Различные **ROLC**, различающиеся только перестановкой

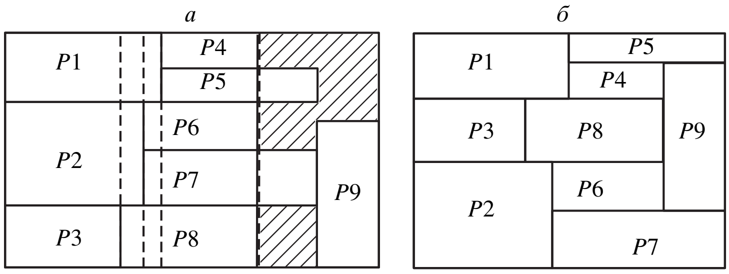

Рис. 2. Перестройка.

элементов в блоках, назовем *эквивалентными.* Прямоугольно-ориентированный линейный раскрой назовем *нормальным* (normal, **n.ROLC**), если существует ему эквивалентный, являющийся прямоугольной упаковкой. Очевидно утверждение:

Лемма 2. *Если пара $(S;N)$ является* **n.ROLC**, *то блок-структура прямоугольной упаковки длины $L = N$ принадлежит множеству эквивалентных* **ROLC**.

Поиск прямоугольной упаковки длины $L = N$ сводится к перестановке элементов внутри блоков. Детерминированный алгоритм направленной перестановки, метод *перестройки*, описан в [23]. Здесь мы приводим его краткую характеристику.

Метод перестройки (Reconstruction, **REC**). Предположим, что имеется план линейного раскроя, удовлетворяющий условиям 1° и 2°, т.е. прямоугольно-ориентированный линейный раскрой, **ROLC**. Среди эквивалентных **ROLC** требуется найти, если это возможно, отвечающий прямоугольной упаковке **RP**. Очевидным является следующее утверждение:

Лемма 3. **ROLC**, *отвечающий исходным данным* **1.5DBPP** *и заданный парой $(S;N)$ удовлетворяет условиям* (1) *или* (2) *непересечения прямоугольников в том, и только в том случае, если для каждой пары $(j, j+1)$ кортежей из S имеет место следующее свойство:*

3°. ***Размещаемость.*** *Если некоторые элементы $i \in I_j^+, j \in S$, то вместо них в освободившихся областях должны разместиться все новые элементы $(j+1)$-го кортежа.*

На этом свойстве базируется алгоритм *перестройки.*

Пусть имеется допустимое решение (S, N) задачи **RCSP**. Каждому кортежу списка (7) отвечают блоки, удовлетворяющие свойствам 1° и 2°. Предположим, что при этом для некоторой пары $(j, j+1)$ соседних кортежей нарушено свойство 3°. В этом случае будем говорить, что возникла ситуация *перестройки*, а соответствующие прямоугольники (элементы) назовем *критическими.* Если для кортежа j возникает ситуация перестройки, то осуществляется перестановка критических и других элементов в предыдущих кортежах, обеспечивающая, если возможно, нужное расположение прямоугольников в кортеже j. На рис. 2,*а* и 2,*б* приведены упаковки различной длины до и после перестройки. В [23] приведены быстрые переборные алгоритмы сложности (m^2). Алгоритм *перестройки* является детерминированным. Его удобно применять в рамках вероятностных алгоримов в качестве декодера. Однако не всякая ситуация перестройки преодолима и тогда $L > N$.

Генетический блочный алгоритм (Genetic Block Algorithm, **GBA**). Пусть задана исходная информация (W, m, w, l) для решения **1.5DBPP**. Затем формируется исходная информация (Z, m, λ, b) для задачи **1DCSP**. Решая эту задачу с учетом свойств 1° и 2°, получаем **ROLC**, заданный парой $(S;N)$. Полученная блок-структура, вообще говоря, не является упаковкой. Ее длина $\Lambda = N$ может использоваться в качестве квазиграницы **RP** в рассматриваемой окрестности.

Генетический алгоритм решения задачи **1.5DBPP** интерпретируется как эволюционный процесс, связанный с перестановкой элементов в кортежах. Каждой допустимой упаковке **RP** отвечает ее блочное представление (4), кортежи в нем расположены в установленном порядке и связаны друг с другом (свойство 2°). Это позволяет интерпретировать их как *гены*, а блок-структуру, соответствующую **ROLC**, можно интерпретировать *хромосомой*, содержащей сцепленные между собой *гены*. Местоположение гена в хромосоме является *локусом*, а альтернативные формы одного и того же гена, расположенные в одинаковых локусах хромосомы, интерпретируются *аллелями*. Хромосома, содержащая в своих локусах конкретные значения аллелей, представляет *генотип*. Конечное множество всех допустимых генотипов образует *генофонд*. Первое допустимое решение определяется путем использования алгоритма **SVC** с ограничениями 1° и 2° для **RCSP**. Так определяют начальное **ROLC**, заданное парой $(S; N)$. Далее, с помощью того или иного декодера находят прямоугольную упаковку длины L. Путем перестановки элементов в кортежах **ROLC** определяют эквивалентные **ROLC**$'$ и отвечающие им *соседние* решения длины L'. Таким образом фиксированный **ROLC** со значением N функции цели определяет окрестность, в которой реализуется локальный поиск оптимума. *Степенью* $\mu(rp)$ *приспособленности* особи rp является значение L длины занятой части полосы прямоугольной упаковки **RP**. *Оценочной функцией* в окрестности с фиксированным значением $\Lambda = N$, является величина $\Delta = (L - \Lambda)/L$. Множество эквивалентных **ROLC** образует *ареал*, заданный окрестностью N, а совокупность особей (допустимых упаковок), принадлежащих ареалу, образует *популяцию* P. Численность генофонда популяции в множестве эквивалентных **ROLC**, определяется параметром $k = \sum\limits_{\nu=1}^{n} r_\nu!$, где r_ν – количество начатых в блоке ν элементов, n – количество блоков. В общем случае экстремальной задачи **1.5DBPP** популяция соответствует совокупности допустимых решений. С помощью основных генетических процедур *кроссовер* и *селекция* может быть найдена особь с показателем $L = \Lambda$ в случае, когда исходный **ROLC** является нормальным. На этом заканчивается работа алгоритма. Иначе, выполняя заданное количество генетических итераций, находят **RP** с $L < \Lambda$ и вычисляют значения оценочной функции. Далее ареал расширяется за счет применения *мутации*, перехода к новой окрестности. Тогда численность генофонда $k = n! \sum\limits_{\nu=1}^{n} r_\nu!$.

Перечислим основные процедуры **GBA**.

Хромосома – вычисление **ROLC** с помощью модификаций алгоритма **SVC** [22].

Декодер – построение допустимой прямоугольной упаковки **RP**, пользуясь алгоритмом *перестройки* в сочетании с блочным декодером [21].

Генофонд – построение аллелей путем перестановок элементов в кортежах.

Кроссовер – выбор случайной хромосомы (родителя), выбор гена (блока) хромосомы, перестановка двух случайных элементов в блоках.

Мутация – построение новой начальной хромосомы.

Алгоритм **GBA** состоит из выполнения следующих шагов:

G1. Построение блок-структуры начальной хромосомы.

G2. Построение начальной популяции. Выполняются процедуры *генофонд* и *декодер*. Повторяется до получения заданного количества особей в популяции.

G3. *Кроссовер* и занесение в популяцию наиболее приспособленных особей.

G4. *Мутация* и переход на G1.

Выполняется алгоритм до тех пор, пока не достигнута *квазиграница*, полученная **SVC** или не выполнено заданное количество шагов.

5. Метод парных списков локального поиска оптимальной упаковки

Метод парных списков основан на утверждениях, связанных с допустимостью **RP**.

Л е м м а 4. *Блок-структуры* **ROLC** *и* **ROLC***, *отвечающие исходным данным* **1.5DBPP** *и заданные парами* $(S;N)((\widetilde{S};\widetilde{N}))$, *удовлетворяют условиям* (1) *или* (2) *непересечения прямоугольников в том и только в том случае, если для них справедливо следующее свойство:*

4°. ***Непересечение.*** *Для любого вертикального и любого горизонтального блока выполняется одно из следующих условий:*

(a) *для любой пары* $(i_1,i_2)\in I_k$, $k\in S$ *и любого кортежа* $j\in\widetilde{S}$: *если* $i_1\in\widetilde{I}_j$, *то* $i_2\notin\widetilde{I}_j$ *или если* $i_2\in\widetilde{I}_j$, *то* $i_1\notin\widetilde{I}_j$;

(b) *для любой пары* $(i_1,i_2)\in\widetilde{I}_j$, $j\in\widetilde{S}$ *и любого кортежа* $k\in S$: *если* $i_1\in I_k$, *то* $i_2\notin I_k$ *или если* $i_2\in I_k$, *то* $i_1\notin I_k$.

Заметим, что выполнение 4° не является достаточным условием эквивалентности **ROLC** и **RP**. На рис. 3 изображена блок-структура **ROLC**, удовлетворяющая 4°, которая не является **RP**. Простым следствием из леммы 4 является следующий достаточный признак эквивалентности **ROLC** и **RP**.

Т е о р е м а 1. *Блок-структуры* **ROLC(ROLC*)** *с исходными данными* **1.5DBPP**, *заданные парами* $(S;N)((\widetilde{S};\widetilde{N}))$ *и удовлетворяющие свойству* 4°, *отвечают упаковке* **RP**, *если координаты* (x_i,y_i) *всех прямоугольников вычислены по формулам:*

$$x_{i(k)}=\sum_{j=1}^{k-1}\chi_j,\quad i(k)\in I_k^-;\quad y_{i(k)}=\sum_{j=1}^{\widetilde{k}-1}\eta_j,\quad i(\widetilde{k})\in\widetilde{I}_{\widetilde{k}}^-, \tag{10}$$

где I_k^- $(\widetilde{I}_{\widetilde{k}}^-)$ *– множество прямоугольников с началом в блоке* $k(\widetilde{k})$.

Рандомизированный поиск парных списков (Random Doublicity List Search, **RDLS**). По общему методу локального спуска алгоритм состоит из выполнения процедур:

RL1. *Инициализация.* Выбрать начальное допустимое решение и вычислить для него значение критерия оптимальности (верхняя граница).

RL2. *Поиск соседнего решения.* Выбрать допустимое соседнее решение из окрестности начального и вычислить для него значение оценочной функции.

RL3. *Анализ перехода.* Проверить, следует ли совершить переход к новому решению: если да, то принять новое решение в качестве текущего. Иначе текущим оставить предыдущее решение и вернуться на RL2.

RL4. *Конец.* Завершить работу алгоритма и вывести решение.

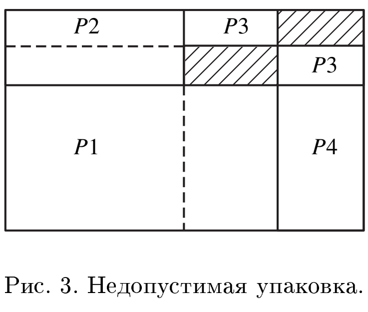

Рис. 3. Недопустимая упаковка.

Процедуры RL1 и RL2 в схеме парных списков представляют последовательное решение пары задач **RCSP** и **RCSP***. Для поиска соответствующих им списков S и $S*$ на каждой итерации применяется алгоритм **FF** с учетом 1°, 2°, 4°.

Рандомизированный поиск парных списков генерирует новый список π путем перестановки всех элементов. Используя полученный список π решаем **RCSP** (алгоритм **FF** с учетом 1° и 2°) и получаем пару $(S;N)$. **RCSP*** решается с помощью **FF** с учетом 1°, 2° и 4°. После получения первого варианта решения $(\widetilde{S};\widetilde{N})$ возможны следующие случаи.

1. Выполнено условие $\widetilde{N} \leqslant W$, переходим к RL3.
2. Оказалось, что $\widetilde{N} > W$, переходим к RL2, отбраковывая **ROLC**.

Процедура *анализ перехода* оставляет лучшую из найденных упаковок и оценивает в процентах отклонение ее длины от лучшего решения **RCSP**.

6. Численный эксперимент

Целью численного эксперимента являлось сравнение эффективности *генетического блочного* алгоритма (**GBA**) и *рандомизированного поиска парных списков* (**RDLS**) между собой, а также сравнение с известными методами: *классическим генетическим* алгоритмом (CGA) [18] и мультиметодным генетическим алгоритмом (**MMA**) [19]. Здесь приведен небольшой срез эксперимента.

В качестве исходных использовались данные, сгенерированные случайным образом. При этом были заданы параметры: $W = 1000$ – ширина полосы; $m = 20,\ 40,\ 60,\ 80,\ 100$ – количество прямоугольников; (w_i, l_i) – размеры i-го прямоугольника, $i = \overline{1,m}$.

Расчеты проводились для трех наборов исходных данных с параметрами: ν_1 – нижнее ограничение ширины прямоугольника по ширине, т.е. $w_i \geqslant \nu_1 W$; ν_2 – верхнее ограничение ширины, т.е. $w_i \leqslant \nu_2 W$; ω_1 – нижнее ограничение длины прямоугольника по ширине, т.е. $l_i \geqslant \omega_1 W$; ω_2 – верхнее ограничение длины, т.е. $l_i \leqslant \omega_2\, W$.

Расчеты проводились на ПЭВМ Pentium III-933 для сгенерированных наборов данных.

Набор № 1: $\nu_1 = 0{,}10$; $\nu_2 = 0{,}50$; $\omega_1 = 0{,}15$; $\omega_2 = 0{,}50$.

Набор № 2: $\nu_1 = 0{,}25$; $\nu_2 = 0{,}40$; $\omega_1 = 0{,}35$; $\omega_2 = 0{,}60$.

Набор № 3: $\nu_1 = 0{,}10$; $\nu_2 = 0{,}15$; $\omega_1 = 0{,}15$; $\omega_2 = 0{,}20$.

В каждом классе задач и для каждого набора было просчитано по 50 примеров и средние результаты решения в виде коэффициентов раскроя (Cutting Coefficient, **CC**) записаны в ячейки таблице. Коэффициент $\mathbf{CC} = \sum\limits_{i=1}^{m} w_i l_i \Big/ (L \times W)$.

Жирным шрифтом в таблице выделены лучшие значения коэффициентов **CC**.

Результаты численного эксперимента

	Набор № 1				Набор № 2				Набор № 3			
m	GBA	RDLS	GCA	MMA	GBA	RDLS	GCA	MMA	GBA	RDLS	GCA	MMA
20	94,10	**94,65**	89,76	94,27	**94,29**	92,86	90,67	85,22	87,07	**91,90**	85,06	90,12
40	94,29	94,03	89,65	**94,70**	**94,50**	94,19	92,10	93,22	93,85	**94,04**	92,19	92,83
60	94,18	94,10	88,85	**95,62**	**95,63**	94,65	91,51	93,61	**93,92**	93,56	92,16	93,42
80	94,11	93,87	88,34	**95,57**	**94,95**	94,22	91,27	94,29	**94,57**	94,32	92,00	93,31
100	93,91	93,91	87,83	**95,80**	**95,96**	94,37	90,62	95,21	**95,08**	94,88	91,72	93,84

Анализируя результаты, можно сделать следующие выводы:

– эффективность блочных алгоритмов мало зависит от выделенных наборов данных и от количества m прямоугольников;

– лучшие решения получены с помощью блочных алгоритмов для прямоугольников средних или мелких габаритов (наборы № 2 и № 3);

– как правило, **GBA** дает лучший коэффициент раскроя по сравнению с **RDLS**; это можно объяснить тем, что в двойственной схеме применялся примитивный рандомизированный вариант простой эвристики **FF**;

– обе блочные схемы значительно эффективнее генетического алгоритма **GCA**;

– мультиметодный алгоритм И.П. Норенкова по своей эффективности для $m \leqslant 100$ мало отличается от схемы поиска парных списков и блочного генетического алгоритма. Он показал лучшие решения для набора № 1, смеси прямоугольников различных размеров.

Заметим, что временные затраты были отведены примерно равные для всех алгоритмов и увеличивались по мере роста m. Однако, при $m > 100$ эффективность **MMA** значительно выше. Для $m \geqslant 500$ удалось получать решения лишь с помощью **MMA**. Это можно объяснить тем, что алгоритм **MMA** работает фактически без декодера, изменяя случайно алгоритмы упаковки.

7. Заключение

Статья посвящена блочной методологии конструирования алгоритмов прямоугольной упаковки: введены понятия блок-структур и блочные способы кодирования упаковок; сформулированы и обоснованы основные свойства блок-структур, в том числе необходимые и достаточные условия эквивалентности блок-структур и прямоугольной упаковки. На этой основе разработаны методы прямоугольной упаковки на базе линейного раскроя специального вида и предложены две группы алгоритмов. Первая использует вертикальную блок-структуру, вторая – вертикальную и горизонтальную структуры. Разработаны конкретные реализации общих схем: детерминированный алгоритм перебора элементов в блоках (пререстройка); генетический блочный алгоритм и рандомизированный поиск парных списков. Приведены результаты среза численного сравнения эффективности алгоритмов с известными генетическими алгоритмами (классическим и мультиметодным алгоритмом И.П. Норенкова). Блочные алгоритмы не уступают им. Основным достоинством является открытая методология разработки блочных алгоритмов. На этой базе могут создаваться более эффективные эвристики, приближенные и точные алгоритмы. Более того, методология может применяться и для разработки методов расчета параллелепипедной упаковки.

ПРИЛОЖЕНИЕ

Доказательство леммы 1. Нетрудно видеть, что 1° и 2° являются необходимыми, но не достаточными условиями соответствия $S(\widetilde{S})$ и **RP**. Необходимость очевидна. Отрицание достаточности следует из контрпримера: при изменении порядка (P_2, P_1) на (P_1, P_2) следования прямоугольников в блоке № 1, см. рис. 1(a), прямоугольники P_2 и P_3 в блоке № 2 пересекутся.

Доказательство леммы 3. Необходимость очевидна. Пусть теперь имеет место свойство 3°. Тогда прямоугольники раздвинуты по оси Oy, т.е. условие (2) выполнено. Это означает по определению непересечение прямоугольников между собой.

Доказательство леммы 4. Необходимость очевидна. Пусть теперь имеет место условие (a) свойства 4°. Тогда если два прямоугольника i_1 и i_2 пересекают один и тот же вертикальный блок, то они не могут пересечь один и тот же горизонтальный блок. А это означает раздвинутость прямоугольников по оси Oy, т.е. (2). Аналогично, выполнение условия (b) свойства 4° означает раздвинутость прямоугольников i_1 и i_2 по оси Ox, т.е. (1). Выполнение условия (a) или (b) свойства 4° означает непересечение прямоугольников P_{i_1} и P_{i_2} по определению.

Доказательство теоремы 1. Пусть имеются блок-структуры **ROLC (ROLC*)**, удовлетворяющие свойству 4°, и координаты (x_i, y_i) прямоугольников вычислены через блок-структуры по формулам (10). Совокупность прямоугольников с координатами (x_i, y_i) является упаковкой. На основании леммы 4 прямоугольники не пересекаются между собой, т.е. выполнены условия (1) или (2). Кроме того, по определению **ROLC*** справедливо (9). А это, в свою очередь, с учетом нулевых координат прямоугольников, касающихся осей Ox и Oy, означает выполнение условий (3). Таким образом, мы имеем допустимую упаковку **RP** с блок-структурами **ROLC(ROLC*)**.

СПИСОК ЛИТЕРАТУРЫ

1. *Dyckhoff H.* A typology of cutting and packing problems // F.R. Germany, 1991.
2. *Hinxman A.* The Trim-Loss and Assortment Problems: A Survey // Europ. J. Oper. Res. 1980. V. 11. P. 863–888.
3. *Канторович Л.В., Залгаллер В.А.* Расчет рационального раскроя материалов. Л.: Лениздат, 1951.
4. *Gilmory P., Gomory R.* Multistage cutting stock problem of two and more dimensions // Oper. Res. 1965. V. 13. № 1. P. 94–120.
5. *Terno J., Lindeman R., Scheithauer G.* Zuschnitprobleme und ihre praktische Losung. Leiprig, 1987.
6. *Романовский И.В.* Алгоритмы решения экстремальных задач. М.: Наука, 1977.
7. *Мухачева Э.А.* Рациональный раскрой промышленных материалов: Применение АСУ. М.: Машиностроение, 1984.
8. *Belov G., Scheithauer G.* A Cutting Plane Algorithm for the One Dimensional Cutting Stock Problem with Multiple Stock Lengths // Europ. J. Oper. Res. 2002. 141(2). P. 274–294.
9. *Martello S., Toth P.* Knapsack problems: Algorithms and Computer Implementations. Chichester. John Wiley&Sons. 1990.
10. *Кацев С.В.* Об одном классе дискретных минимаксных задач // Кибернетика. 1979. № 5. С. 139–141.
11. *Гимади Э.Х., Залюбовский В.В.* Задача упаковки в контейнеры: асимптотически точный подход // Изв. вузов. Математика. 1997. № 12. С. 25–33.
12. *Coffman E., Garey M., Jchonson D.* Approximation algorithms for bin-packing. An updated survey // Algorithm Design for Computer System Design (Ausiello G., Lucertini M., Serafini P. eds) Berlin etal. 1984. P. 49–106.
13. *Мухачева Э.А. Валеева А.Ф.* Метод динамического перебора в задаче двумерной упаковки // Информ. технологии. 2000. № 5. С. 30–37.
14. *Стоян Ю.Г., Гиль Н.И.* Методы и алгоритмы размещения плоских геометрических объектов. Киев: Наук. думка, 1976.
15. *Кочетов Ю.А.* Вероятностные методы локального поиска для задач дискретной оптимизации / Дискретная математики и ее приложения. Сб. лекций молодежных и научных школ. М.: МГУ, 2001. С. 87–117.
16. *Aarts E., Lenstra J.* Local Search in Combinatorial Optimization. John Wiley&Sons, 1996.
17. *Falkenauer E.* A hybrid Grouping Genetic Algorithm for Bin Packing // J. Heuristics. 1998. V. 2. № 1. P. 5–30.

18. *Liu D., Teng H.* An improved BL-algorithm for genetic algorithm of the orthogonal packing of rectangles // Europ. J. Oper. Res. 1999. 112. P. 413–420.

19. *Норенков И.П.* Эвристики и их комбинации в генетических методах дискретной оптимизации // Информ. технологии. 1999. № 1. С. 2–7.

20. *Гончаров Е.Н., Кочетов Ю.А.* Поведение вероятностных жадных алгоритмов для многостадийной задачи размещения // Дискретный анализ и исследование операций. 1999. Сер. 2. 6. № 1. С. 12–32.

21. *Мухачева А.С., Чиглинцев А.В., Смагин М.А., Мухачева Э.А.* Задачи двумерной упаковки: развитие генетических алгоритмов на базе смешанных процедур локального поиска оптимального решения // Информ. технологии. 2001. № 9. Приложение. 25с.

22. *Mukhacheva E.A., Belov G.N., Kartak V.M., Mukhacheva A.S.* Linear one-dimensional cutting-packing problems: numerical experiments with sequential value correction method (SVC) and a modified branch-and-bound method (MBB) // Pesquisa Operacional. V. 20. № 2. P. 153–168.

23. *Мухачева Э.А. Мухачева А.С.* Метод перестройки для решения задачи прямоугольной упаковки // Информ. технологии. 2000. № 4. С. 30–36.

Статья представлена к публикации членом редколлегии А.И. Кибзуном.

Поступила в редакцию 27.06.2003