Записки научных
семинаров ПОМИ
Том 312, 2004 г.

Э. А. Мухачева, А. С. Мухачева

Л. В. КАНТОРОВИЧ И ЗАДАЧИ РАСКРОЯ–УПАКОВКИ: НОВЫЕ ПОДХОДЫ ДЛЯ РЕШЕНИЯ КОМБИНАТОРНЫХ ЗАДАЧ ЛИНЕЙНОГО РАСКРОЯ И ПРЯМОУГОЛЬНОЙ УПАКОВКИ

В течение последних пятидесяти лет проблема *раскроя-упаковки* (Cutting&Packing, **C&P**) привлекает внимание ученых и производственников. Причина интереса к задачам **C&P** состоит в их принадлежности к NP-трудным проблемам, в широкой применимости к их решению математического программирования и комбинаторных методов, в сложности многих задач реального мира, которые можно реализовать только как NP-трудные.

Под задачами раскроя-упаковки понимается широкий класс проблем, допускающих различное прикладное толкование. Общим фактором этого класса задач является наличие двух групп объектов. Между элементами групп устанавливается и оценивается соответствие. Впервые классификация задач **C&P** была проведена Л. В. Канторовичем и В. А. Залгаллером в 1971 г. [2]. Проведенный ими обзор касался в основном литературы отечественных авторов. Качественная типология в области раскроя-упаковки проведена в 1991 г. германским ученым H. Dykhoff [15]. Она принята в мировой практике и используется при изучении моделей и методов решения задач **C&P**. Обзор задач **C&P** и методов их решения расширен H. Dykhoff, J. Terno и G. Scheithauer [14]. Аналитический обзор в области раскроя и упаковки выполнен по заданию РФФИ Э. А. Мухачевой, А. С. Мухачевой, А. Ф. Валеевой и В. М. Картаком [9].

Систематическое изучение проблемы раскроя-упаковки заложено в книге Л. В. Канторовича и В. А. Залгаллера [3, 2]. Впервые задачи линейного и гильотинного раскроя описывают-

Работа поддержана Фондом Президента Российской Федерации, проект № МК 145.2003.01.

ся моделью линейного программирования с неявно заданной информацией о столбцах матрицы (раскроях). Для их генерации В. А. Залгаллером был предложен прием, предвосхитивший появившееся позднее динамическое программирование. Прорывом в прикладной области явились объективно-обусловленные оценки (о.о.оценки) заготовок. Последние представляют экономический смысл двойственных переменных в оптимальном плане раскроя. Таким образом, о.о.оценки оказались мощным инструментом анализа плана раскроя в процессе заказа материала. Изменения оценок в процессе решения задачи были умело использованы Леонидом Витальевичем и Виктором Абрамовичем для получения рациональных раскроев. Уже тогда были предприняты попытки внедрения экономных планов для ряда производств в Ленинграде (Кировский завод, вагоностроительный завод им. Егорова). Аналогичные методы появились за рубежом только через 10 лет в работах P. Gilmore&R. Gomory [16] и позднее в книге J. Terno, R. Lindeman&G. Scheithauer [22]. Для решения задачи генерирования линейного и гильотинного прямоугольного раскроя были разработаны метод склейки И. В. Романовского [12] и сеточный метод Э. А. Мухачевой [6]. При этом интерес Л. В. Канторовича к задачам **C&P** не ослаб. Книга Э. А. Мухачевой [6] была написана и опубликована под личным патронажем Леонида Витальевича. Основная цель Л. В. Канторовича – иллюстрация и применение линейного программирования в промышленности – была достигнута в условиях массового и серийного производства. Аналогичные методы были применены и для задач гильотинного прямоугольного раскроя, когда возможными являются только *сквозные* резы, параллельные кромкам раскраиваемого материала. Однако задачи **C&P** по своей сути являются проблемами дискретной оптимизации, и их решение с помощью линейного программирования – не более чем *непрерывная релаксация*. Далее задачи **C&P** рассматриваются как прикладные проблемы исследования операций. Н. Гэри и Д. Джонсон показали, что простейшая *задача одномерной упаковки* (1 Dimensional Bin Packing Problem, **1DBPP)** принадлежит к классу NP-трудных задач [1]. Этот факт становится классикой в комбинаторной оптимизации.

1. Целочисленная задача линейного раскроя: применение метода отсекающих плоскостей

1.1. Двухэтапные подходы на базе линейного и линейного целочисленного программирования. Материал для раскроя на заготовки заданных размеров поступает в виде одинаковых стержней (полос). Требуется раскроить стержни на заготовки с наименьшими затратами материала. Задача линейного раскроя **(1DCSP)** определяется следующим информационным вектором:

$$(L; m; \lambda = (\lambda_1, \ldots, \lambda_i, \ldots, \lambda_m);\ \beta = (b_1, \ldots, b_i, \ldots, b_m)), \qquad (1)$$

где L – длина раскраиваемого материала; m – количество различных заготовок; λ_i, $i = \overline{1,m}$, – длины (веса) заготовок; b_i, $i = \overline{1,m}$, – требуемое количество заготовок. Вектор $(a_{1j}, a_{2j}, \ldots, a_{mj})^T \in Z_+^m$ описывает шаблон раскроя j с компонентами a_{ij}, указывающими в нем количество заготовок типа $i = \overline{1,m}$. Матрица $A = (a_{ij})$, $i = \overline{1,m}$, $j = \overline{1,n}$, где n – количество шаблонов, называется раскройной матрицей; $\alpha^j \in A$ означает допустимый раскрой, если

$$\sum_{i=1}^{m} a_{ij} l_i \leqslant L, \quad j = \overline{1,n}. \qquad (2)$$

Обозначим через x_j, $j = \overline{1,n}$, – интенсивность применения раскроя j. Проблема планирования оптимального раскроя материала сводится к следующей задаче.

Задача 1DCSP. Требуется минимизировать функцию

$$\mu(x) = \left\{ \sum_{j=1}^{n} x_j \middle| x = (x_1, x_2, \ldots, x_n) \in Z_+^n; \sum_{j=1}^{n} x_j\, \alpha^j = \beta \right\}. \qquad (3)$$

Если считать, что все векторы α^j, $j = \overline{1,n}$, заданы, то **1DCSP** является задачей линейного целочисленного программирования. Однако количество n различных шаблонов раскроя очень велико, поэтому столбцы матрицы A задаются неявно через условия (2). Они генерируются на каждой итерации линейного программирования путем решения вспомогательной задачи динамического программирования. Это позволило решать проблемы раскроя с использованием линейного программирования и находить непрерывные значения компонент вектора x и функции $\mu(x)$. С помощью этого подхода Л. В. Канторович и В. А. Залгаллер и предложили решать задачи линейного и гильотинного раскроя в условиях массового или крупносерийного производства.

В 1985 г. O. Marcotte сформулировал важные свойства целочисленного решения для задачи линейного раскроя [19]: *целочисленные свойства* (Integer Property, **IP**), если целочисленный оптимум μ^* совпадает с оптимальным решением μ задачи непрерывной релаксации; *свойство округления до целого вверх* (Integer Round-up Property, **IRUP**), если $\mu^* = \lceil \mu \rceil$; *модифицированное свойство округления до целого вверх*, **MIRUP**, если $\mu^* \leqslant \lceil \mu \rceil + 1$. Относительно этих свойств высказано следующее предположение.

Предположение. Стандартная одномерная задача раскроя обладает свойством **MIRUP**.

Это предположение до сих пор не опровергнуто, но и не доказано.

На этой базе для случаев единичного или мелкосерийного производства был предложен двухэтапный процесс расчета целочисленного раскроя. На первом этапе с помощью линейного программирования решается задача непрерывной релаксации. Найденное оптимальное значение $\mu(x)$ округляется сверху до ближайшего целого $\mu_c(x) = \lceil \mu(x) \rceil$, а полученные значения x_j округляются до целого снизу. При этом $\mu_c(x)$ служит нижней границей оптимума для исходной задачи и формируется остаточная задача (Surplus Problem, **SP**) из недополученных заготовок. Для решения **SP** на втором этапе применяется эвристика. Развитие двухэтапного подхода привело к использованию отсекающих плоскостей Гомори. При добавлении сечений непрерывное решение становится все ближе к целочисленному оптимуму. Для генерации новых сечений в случае, когда имеется только базисная матрица, используется *процедура округления Хватала–Гомори* (Chvatal-Gomory Rounding Procedure, **C-GRP**) [13]. На втором этапе используется эвристика – *метод последовательного уточнения оценок*, **SVC**.

1.2. Характеристика алгоритма SVC.

Метод **SVC** представляет собой многопроходную эвристику, предназначенную для решения задач одно-, двух- и трехмерного раскроя. Идея метода, основанная на использовании экономического смысла двойственных переменных, была предложена Э. А. Мухачевой и В. А. Залгаллером [21].

Метод последовательного уточнения оценок (**SVC**) реализуется по схеме **FFD** с процедурами приоритета и повторения. Например, применяется простейшая схема **FFD**, она имитирует действия мастера в раскройном цехе. Пусть известен *приоритетный список* (Priority List, **PL**), в котором каким-то образом упорядочены заготовки. На j-м такте от стержня отрезается первая из **PL** *возможная* заготовка с номером i, длина l_i которой не превосходит длины L стержня. Потребность b_i в этой заготовке корректируют, полагая $b_i = b_i - 1$. Если новое значение $b_i = 0$, то элемент i удаляется из **PL**. От полученного остатка длины $L := L - l_i$ вновь отрезается первая из **PL** *возможная* заготовка. Если таковой нет, то такт закончен и полученный раскрой j характеризуется вектором $\alpha^j = (a_{1j}, a_{2j}, \ldots, a_{ij}, \ldots, a_{mj})$, компоненты a_{ij} которого указывают количество i-х заготовок в j-м раскрое. Следующий $(j+1)$-й такт начинается с раскроя целого стержня. Если **PL**$= \varnothing$, т.е. все требуемые заготовки получены, то конец. В методе **SVC** упорядочивание элементов основано на экономическом смысле объективно-обусловленных оценок (о.о.оценок) Л. В. Канторовича в линейном программировании [2]. А именно, они имеют смысл подетальных норм расхода материала в оптимальном плане. В допустимом плане линейного непрерывного раскроя в качестве оценок принимают двойственные переменные. С помощью последних проверяется оптимальность плана и в противном случае намечается путь его улучшения. В целочисленном случае имеется разрыв двойственности и непосредственное применение линейного программирования и о.о.оценок становится невозможным. Мы используем только идею пересчета оценок с целью улучшения плана раскроя.

Рассмотрим задачу (3) раскроя стержня (**1DCSP**). План целочисленного раскроя удобно представить в виде совокупности N раскроев с единичной применяемостью каждого из них. Тогда искомый план описывается матрицей $A = (a_{ij})$, $i = \overline{1,m}$; $j = \overline{1,N}$, в которой неизвестными являются элементы a_{ij} и количество столбцов N. Первый план (допустимое решение) получают с помощью метода **FFD**. Соответствующее значение целевой функции N_f принимается в качестве верхней границы. Естественной нижней границей является $N_0 = \lceil \sum_{i=1}^{m} l_i b_i / L \rceil$, или полагают $N_0 := N_\mu = \lceil \mu(x) \rceil$, где $\mu(x)$ – оптимальное решение непрерывного аналога задачи (3), полученное с помощью линейного программирования. В качестве *"псевдо-оценок"* y_i принимаются подетальные нормы расхода материала, связанные с конкретным планом A_f, полученным **FFD**-алгоритмом. Далее строится **PL** по убыванию удельных оценок $y_{i0} = y_i^0/l_i$. На следующих этапах с помощью **PL** организуется поиск очередной карты раскроя методом ветвей и границ с максимальной суммой *"псевдо-оценок"* заготовок. При этом оценки и **PL** исправляются после построения каждого очередного вектора α^j. Процесс заканчивается на последнем шаге некоторого этапа s в случае, если значение целевой функции $N_s = N_0$, или по числу итераций.

Пусть на первом этапе после выполнения **FFD** получены раскрои α^j и **PL**, в котором заготовки упорядочены по убыванию их удельных псевдо-оценок y_{i0}. В раскрое j длина i-х заготовок равна $a_{ij}l_i$. Учитывая остаток h_j, в качестве нормы длины стержня в j-м раскрое примем величину $\left(a_{ij}l_i\frac{L}{L-h_j}\right)$. Тогда для всех раскроев $j = \overline{1,N}$ с учетом требуемого количества b_i заготовок i-го вида получаем подетальную норму длины i-й заготовки

$$y_i^0 = \frac{l_i L}{b_i}\sum_{j=1}^{N}\frac{a_{ij}}{L-h_j}, \quad i = \overline{1,m}. \tag{4}$$

Эта величина и принимается в качестве начального значения псевдо-оценки i-й заготовки.

В последующих итерациях способы раскроя строятся по модифицированной схеме **FFD**. На очередном j-м такте находят раскрой стержня с максимальной суммарной псевдо-оценкой заготовок. Для этого используется динамический метод решения задачи рюкзака на базе текущего **PL**. С помощью последнего значительно сокращается перебор заготовок на предмет включения их в раскрой. После построения j-го раскроя перевычисляются псевдо-оценки заготовок i с $a_{ij} > 0$ по формулам

$$y_i^j = \frac{1}{b_i}(a_{i,j-1}l_i\frac{L}{L-h_{j-1}} + y_i^{j-1}(b_i - a_{i,j-1})), \quad \forall i : a_{i,j-1} > 0. \tag{5}$$

Новая оценка является взвешенной средней величиной старой и новой норм использования материала. Далее **PL** корректируется в соответствии с удельными оценками.

Вычисление "псевдо-оценок" (4) и (5) на базе модифицированной схемы **FFD** с повторением представляет *детерминированный* *вариант* **SVC**. Однако на некоторых классах **FFD**-*трудных* задач алгоритм **SVC** плохо выводит их из этих областей. Предлагается модификация, *метод случайного уточнения оценок* (Sequential Value Correction Stochastic, **SVCS**), позволяющая варьировать весами старых и новых значений оценок при их корректировке. Тогда текущие оценки подсчитываются по следующим формулам, найденным эмпирическим путем:

$$y_i^j = \frac{1}{\Omega \times (b_i + b_i^j) + a_{i,j-1}} \left(a_{i,j-1} l_i \frac{L}{L - h_{j-1}} + y_i^{j-1} \times \Omega \times (b_i + b_i^j) \right), \quad (6)$$

где b_i^j – требуемое количество i-х заготовок после получения j-го раскроя, Ω – некоторый параметр, который следует менять в пределах от 1 до нескольких единиц. Это вносит элемент случайности в процесс поиска решений и позволяет предотвращать зацикливание. Кроме того веса старых псевдо-оценок уменьшаются с уменьшением величины запроса b_i^j заготовок. Это одна из многих возможных формул, показавших свою эффективность.

Для каждого примера существует несколько значений Ω, которые приносят очень хорошее решение. Так как не существует метода для их определения, предлагается следующее: использовать Ω, изменяющееся линейно в интервале $[1, 3]$ с периодом 30 генераций раскроев со случайным искажением. Алгоритм прекращает действовать при достижении нижней границы или через данное число итераций. У этого метода нет теоретического обоснования, что и служит прекрасным стимулом продолжения работы для авторов и их последователей. Следует отметить особенность метода: за первые несколько итераций, которые проходят за доли секунды, происходит основное улучшение решения. А улучшение значения целевой функции на последнюю единицу может потребовать несколько десятков итераций. Результаты эксперимента приведены в работе [20].

2. Блочная технология конструирования алгоритмов для решения задач прямоугольной упаковки полубесконечной полосы

При решении задач прямоугольной и параллелепипедной упаковки рассматриваются две задачи: планирование упаковок (раскроя) и задач генерирования шаблона одного раскроя. Теоретически они сходны с соответствующими проблемами линейного раскроя (или упаковки). Однако вторая задача оказывается много сложнее. А именно, она является NP-трудной, и точные методы приемлемы только для решения задач небольшой размерности. Здесь предложена блочная технология, позволяющая сводить решение двумерной задачи к одной или двум задачам линейного раскроя с дополнительными ограничениями.

2.1. Постановка задачи 1.5DBP.

Представленная здесь технология предназначена для решения задач упаковки в полубесконечную полосу **(1.5DBP)**. Она легко модифицируется на задачи упаковки двух- и трехмерных контейнеров.

Остановимся подробнее на постановке этой задачи. Пусть имеются прямоугольная полоса заданной ширины W и неограниченной длины и набор из m прямоугольных предметов заданных размеров (w_i, l_i), $i = 1, \ldots, m$, 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.

Задача 1.5DBPP. Набор векторов (x_i, y_i), $i = 1, \ldots, m$, называется прямоугольной упаковкой (Rectangular Packing, **RP**), если для $i \neq j$; $i, j = 1, \ldots, m$

$$x_i \geqslant (x_j + l_j) \vee x_j \geqslant (x_i + l_i); \tag{7}$$

или

$$y_i \geqslant (y_j + w_j) \vee y_i \geqslant (y_i + w_i); \tag{8}$$

для $i = 1, \ldots, m$

$$x_i \geqslant 0 \wedge y_i \geqslant 0 \wedge y_i + w_i \leqslant W \tag{9}$$

Условия (7) или (8) означают непересечение прямоугольников между собой, а (9) – с гранями полосы. На рис. 1 изображены допустимая и недопустимая упаковки. Если длина L занятой части полосы достигает минимума, то **RP** – оптимальная упаковка. Исходную информацию для **1.5DBPP** принято задавать вектором $(W; m; w; l)$; $w = (w_1, \ldots, w_m)$; $l = (l_1, \ldots, l_m)$.

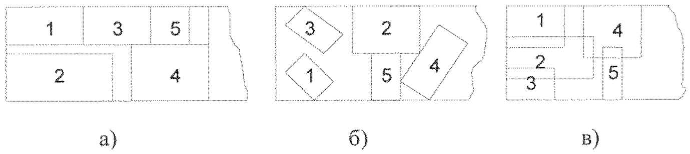

Рис. 1. Упаковки прямоугольников: а – допустимая ортогональная упаковка; б – неортогональная упаковка; в – недопустимая ортогональная упаковка.

Различаются задачи прямоугольной упаковки и гильотинного раскроя. В том и другом случаях требуется разделить большой прямоугольник на малые так, чтобы стороны прямоугольников были параллельны сторонам большого прямоугольника и выбранная функция цели достигала минимума. Задачи упаковки и гильотинного раскроя различаются технологией разделения. Гильотинность предполагает возможность только сквозных резов, параллельных кромкам раскраиваемого материала. Задачи гильотинного раскроя берут свое начало от одномерного случая и рассматривались как обобщение последнего в первых работах Л. В. Канторовича, В. А. Залгаллера и P. Gilmore & R. Gomory. Подробно эта задача в условиях массового и серийного производства описана в работах Э. А. Мухачевой [6] в России и J. Terno, R. Lindeman & G. Scheithauer в Германии [22]. Метод склейки И. В. Романовского, разработанный вначале для задач линейного раскроя, также был обобщен на случай гильотинного раскроя [12]. В современных работах авторы часто исходят из тщательной проработки **1DCSP**. Например, метод отсекающих плоскостей, подробно разработанный и исследованный Г. Беловым и G. Scheithauer, обобщается на случай гильотинного раскроя [13]. Что касается задач негильотинной упаковки, то далее рассматривается их сведение к решению **1DCSP**.

2.2. Блок-структуры упаковки и их свойства.

Задача линейного раскроя с дополнительными ограничениями рассматривается на отдельных этапах решения задачи негильотинной упаковки. Это стало возможным при использовании блочных моделей упаковки. Представление упаковки в виде блок -структур и развитие на этой базе методов расчета начато в середине 90-х годов после содержательной беседы с В. А. Залгаллером.

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 χ_j. Пусть длина **RP** равна N. Проведем через верхние стороны прямоугольников горизонтальные линии. Тогда **RP** разобьется на *горизонтальные блоки* одной и той же длины N и различной ширины η_j, причем $\sum_j \eta_j \leqslant W$. Таким образом, мы получаем две блок-структуры для **RP**, вертикальную и горизонтальную. Каждому блоку j сопоставим *кортеж* (запись номеров прямоугольников, пересекающих блок), длину χ_j вертикального блока и ширину η_j горизонтального блока. В качестве шифров блок-структур используем пары списков $(S,N)((\tilde{S},\tilde{N}))$, где

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

$$\tilde{S}=\{1(j,2(j),\dots,i(j),\dots\}\eta_j,\quad j=\overline{1,q},\quad \tilde{N}=\sum_{j=1}^{q}\eta_j,\qquad(10)$$

где $i(j)$ – номер прямоугольника в позиции i, пересекающего блок j; r – количество вертикальных блоков, q – количество горизонтальных блоков.

Пример 1. На рис. 2(а) и 2(б) изображены вертикальная и горизонтальная блок-структуры упаковки с $m=6$. Разбиения на блоки указаны штриховыми линиями. Там же указаны длины (ширины) блоков. Легко проверить, что этим блок-структурам отвечают следующие списки:

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

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

Таким образом, имея упаковку, легко получить пару списков S и $\tilde{S}$, отвечающих блок-структурам. Нас интересует другая задача: по исходной информации $(W;m;w;l)$ найти списки $S(\tilde{S})$, со-

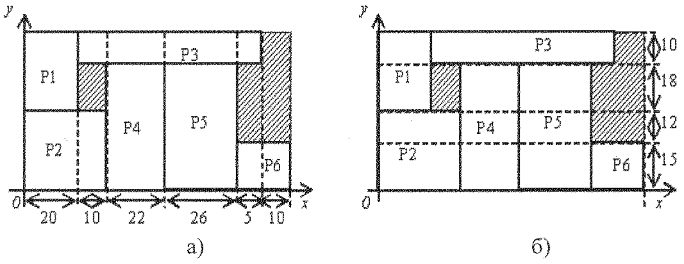

Рис. 2. Блок-структуры упаковки: а – вертикальная блок-структура; б – горизонтальная блок-структура.

ответствующие упаковке. Обозначим: $I_j(\tilde{I}_j)$ – множество прямоугольников, пересекающих j-й вертикальный (горизонтальный) блок; $I_j^+(\tilde{I}_j^+)$ – множество прямоугольников $i \in J_j$ $(i \in \tilde{I}_j)$, заканчивающихся в j-м блоке; $I_j^-(\tilde{I}_j^-)$ – множество прямоугольников $i \in I_j$ $(i \in \tilde{I}_j)$, начинающихся в j-м блоке. Важными являются следующие необходимые свойства соответствия:

Лемма 1. *Пусть список $S(\tilde{S})$ построен на основании исходной информации $(W; m; w; l)$ задачи* **1.5DBP**. *Если этот список соответствует допустимой упаковке* **RP**, *то он удовлетворяет следующим свойствам:*

1°. Разнородность прямоугольников. *Элементы $i(j)$ каждого кортежа $I_j(\tilde{I}_j)$ различные.*

2°. Продолженность прямоугольников. *Если некоторый элемент $i(j) \notin I_j^+(\tilde{I}_j^+)$, то $i(j) \in I_{j+1}(\tilde{I}_{j+1})$.*

3°. 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 $j \in \tilde{S}$: если $i_1 \in \tilde{I}_j$, то $i_2 \notin \tilde{I}_j$ или если $i_2 \in \tilde{I}_j$, то $i_1 \notin \tilde{I}_j$;*

(b) *для любой пары $(i_1, i_2) \in \tilde{I}_j$, $j \in \tilde{S}$, 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 $i_1 \notin I_k$.*

4°. Непересечение с границами полосы: $\sum_{j=1}^{r} \chi_j = N$; $\sum_{j=1}^{q} \eta_j = \tilde{N} \leqslant W$.

5°. Координаты $(x_i; y_i)$, $i = \overline{1, m}$, *вычисляются по формулам*

$$x_{i(k)} = \sum_{j=1}^{k-1} \chi_j, \quad i(k) \in I_k^-; \quad y_{i(k)} = \sum_{j=1}^{\tilde{k}-1} \eta_j, \quad i(\tilde{k}) \in \tilde{I}_{\tilde{k}}^-. \tag{11}$$

Лемма 2. *Свойство 3° является достаточным условием непересечения прямоугольников в упаковке.*

Доказательство. Пусть имеет место условие (a) свойства 3°. Тогда, если прямоугольники P_{i_1} и P_{i_2} пересекают один и тот же вертикальный блок, то они не могут пересечь один и тот же горизонтальный блок, т.е. они содержатся в различных горизонтальных блоках. А это означает раздвинутость прямоугольников по оси Oy, т.е. (8). Аналогично, выполнение условия (b) означает раздвинутость прямоугольников по оси Ox. Выполнение (a) или (b) означает непересечение прямоугольников по определению.

Утверждение 1. (Достаточные условия соответствия). Блок-структуры (S, N) и $(\tilde{S}, \tilde{N})$, удовлетворяющие свойствам 1°, 2°, 5° и свойствам 3°, соответствуют допустимой упаковке **RP** с координатами $(x_i; y_i)$, $i = \overline{1, m}$, вычисленными по формулам (11).

Доказательство. Пусть блок-структуры (S, N) и $(\tilde{S}, \tilde{N})$ удовлетворяют свойствам 1°, 2°, 5° и условию 3°. Условие 1° обеспечивает неразрывность прямоугольников по вертикали, условие 2° – продолженность (неразрывность) прямоугольников по горизонтали. Таким образом, 1°, 2° гарантируют целостность прямоугольников в блок-структурах. Выполнение 3° означает непересечение прямоугольников между собой ((7) или (8)), а выполнение 5° – непересечение с гранями полосы (9). Тогда блок-структурам соответствует допустимая упаковка по определению. Очевидно, что координаты $(x_i; y_i)$ прямоугольников P_i, $i = \overline{1, m}$, удовлетворяют условиям (11).

2.3. Технология блочных структур: вероятностные блок-алгоритмы локального поиска.

На основе блок-структур разработаны методы парных списков: детерминированные алгоритмы используются в качестве алгоритмов размещения прямоугольников при заданном или текущем приоритетном списке их номеров. В рамках вероятностного метода локального поиска оптимума простые алгоритмы применяются в качестве декодеров для построения упаковки по заданным спискам ее деталей [10].

Назовем задачу **1DCSP**, удовлетворяющую условиям 1° и 2°, прямоугольно ориентированной задачей линейного раскроя и обозначим ее **RCSP**.

Двойственный метод локального поиска оптимальной упаковки. Двойственная схема генерирует новый список π с помощью некоторого алгоритма Π.

1. *Решение специальных задач линейного раскроя.* Используя полученный список π, решаем задачу прямоугольно ориентированного линейного раскроя (**RCSP**) простым однопроходным алгоритмом **D** с учетом 1° и 2° и получаем пару (S, N). **RCSP*** также решается с помощью **D** и с учетом 1°, 2° и 3°.

2. *Анализ.* После получения каждого варианта решения $(\tilde{S}; \tilde{N})$ возможны следующие случаи:

2.1. Выполнено условие $\tilde{N} \leqslant W$. Запоминаем рекорд N и пару списков S, $\tilde{S}$. При достижении нижней границы или выполнении заданного числа итераций, переходим к процедуре 4. Иначе переходим к выполнению процедуры 3.

2.2. Оказалось, что $\tilde{N} > W$. Если выполнено заданное число итераций, то для рекорда N и соответствующих списков переходим к 4. Иначе переходим к выполнению процедуры 3, отбраковывая ранее полученные решения (S, N) и $(\tilde{S}; \tilde{N})$.

3. *Изменение списка* π. Генерируем новый список π_{new}, согласно выбранного алгоритма Π, и переходим к выполнению процедуры 1.

4. *Координаты упаковки.* Процедура *анализ перехода* оставляет лучшую из найденных упаковок. Для списков S и $\tilde{S}$ по формулам (11) подсчитывают координаты прямоугольников и изображают эскиз упаковки.

При выборе индивидуальных алгоритмов **D** и Π реализуются различные схемы локального поиска оптимальной упаковки, от простейшего наивного до оптимизационного алгоритма.

При реализации наивной схемы в качестве алгоритма **D** выбран “*первый подходящий*” (First Fit, **FF**); после получения пары $(S; \tilde{S})$ процедура 2 отбрасывает отбракованные решения и выбирается рекорд; алгоритм Π на процедуре 3 осуществляет случайную перестановку элементов списка π [4].

Другим примером является метод “*замещения*” (Substitution, **Sub**), модификации которого применяются в качестве алгоритма типа **D**. При использовании **Sub** всегда возможна ситуация, когда $\tilde{N} \leqslant W$, т.е. отбраковка решений **RCSP*** не происходит. Модификация **Sub** с помощью жадных стратегий приводит к значительному повышению его эффективности [10].

На сегодня лучшие результаты получены при использовании в качестве головного метода П *генетического блочного алгоритма* (Genetic Block Algorithm, **GBA**), а в качестве метода **D** – блочного декодера (Block Decoder, **BD**) [5]. Лучшие результаты обеспечиваются также применением метода **SVC** для построения начального решения и его блочной структуры. Таким образом, задача решается в окрестности **SVC**, изменение которой происходит при использовании процедуры *мутации*.

2.4. Численный эксперимент.

В цитируемых работах [4, 8] и [10] приведены результаты численных экспериментов, подтверждающих эффективность технологии блочных структур. Здесь мы ограничимся расчетами на безотходных примерах Евы Хоппер [17]. В таблице 1 приведены результаты, полученные с помощью следующих алгоритмов:

– **E.Hopper**, гибридный генетический [17];
– **GBA**, генетический блочный алгоритм [8];
– **GSubRec**, генетический классический алгоритм с декодером замещения с перестройкой [10];
– **GMM**, мультиметодный генетический алгоритм [11];
– **SP**, алгоритм парных последовательностей [18];
– **TS**, метаэвристики “поиск с запретами” [7];
– **DSP**, динамический перебор [9].

Как видно из таблицы 1, лучшие результаты показал блочный генетический алгоритм, построенный на базе блочных технологий. Худшие результаты показал алгоритм E. Hopper. Генетический алгоритм с декодером **Sup**, основанным на блочной технологии, показывает средние результаты, вполне сопоставимые с мультиметодным генетическим алгоритмом И. П. Норенкова. Алгоритм **SeqPair**, разработанный в Японии, отстает от наших отечественных алгоритмов. Заметим, что все сказанное верно лишь на классе задач E. Hopper. Окончательные выводы можно сделать после проведения обширного эксперимента.

Декодеры блочной структуры **BD** и **Sub** восстанавливают все упаковки Е. Хоппер при условии сохранения заданного порядка π прямоугольников. Это свойство названо нами *реставрацией*.

В заключение авторы выражают благодарность В. А. Залгаллеру и И. В. Романовскому за внимание к нашей работе, терпение и полезные советы при обсуждении результатов. Приоритет Л. В. Канторовича во всех наших разработках в области раскроя

Таблица 1. Результаты на безотходных примерах Евы Хоппер (отклонение от оптимального решения).

Класс задач	$C1$	$C2$	$C3$	$C4$	$C5$	$C6$	$C7$	Ср. знач.	Кол. лучш. реш.
Кол. предметов	16-17	25	28-29	49	72-73	97	196-197		
E.Hopper	4.0	6.0	4.0	3.0	3.0	3.0	4.0	3.9	0
GBA	0.0	0	3.3	3.3	2.2	2.8	2.1	2.0	2
GSubRec	1.5	2.2	3.3	2.8	2.2	2.5	3.2	2.5	0
GMM	0	2.2	3.3	3.8	2.6	2.8	2.2	2.4	1
SP	0	6.6	3.3	2.2	1.9	2.7	2.9	2.8	2
TS	0	6.6	3.3	2.7	1.9	1.3	1.1	2.4	3
DSP	0	0	2.2	3.3	1.8	4.2	4.7	2.3	4

и упаковки несомненен, а его полезные советы и подсказки, внимание к производственным вопросам раскроя дают о себе знать и спустя много лет. Остается современным его высказывание о практическом использовании методов рационального раскроя:

"В большинстве производств сохраняется ощутимый резерв экономии материала на стадии раскроя. Систематически этот резерв мало изучается". Л. В. Канторович. Решение расширенного заседания бюро Научного совета ГКНТ по проблеме использования оптимизационных задач по вопросу "Оптимизация раскроя заготовок", 15–17 апреля 1983 г., г. Звенигород.

Литература

1. М. Гэри, Д. Джонсон, *Вычислительные машины и труднорешаемые задачи*. Мир, М. (1982).
2. Л. В. Канторович, В. А. Залгаллер, *Рациональный раскрой промышленных материалов*. Наука СО, Новосибирск (1971).

3. Л. В. Канторович, В. А. Заллгаллер, *Расчет рационального раскроя материалов.* Лениздат, Л. (1951).
4. А. С. Мухачева, С. Х. Куреленков, М. А. Смагин, Р. Р. Ширгазин, *Методы локального поиска оптимума прямоугольной упаковки с использованием двойственной схемы.* — Информационные технологии, № 10 (2002), 26–31.
5. А. С. Мухачева, А. В. Чиглинцев, М. А. Смагин, Э. А.Мухачева, *Задачи двумерной упаковки: развитие генетических алгоритмов на базе смешанных процедур локального поиска оптимального решения.* — Информационные технологии, № 9 (2001), приложение.
6. Э. А. Мухачева, *Рациональный раскрой промышленных материалов. Применение в АСУ.* Машиностроение, М. (1984).
7. Э. А. Мухачева, А. И. Ермаченко, Т. М. Сиразетдинов, А. Р. Усманова, *Метод поиска минимума с запретами в задачах двумерного гильотинного раскроя.* — Информационные технологии, № 6 (2001), 25–31.
8. Э. А. Мухачева, А. С. Мухачева, А. В. Чиглинцев, *Генетический алгоритм блочной структуры в задачах двумерной упаковки.* — Информационные технологии, № 11 (1999), 13–18.
9. Э. А. Мухачева, А. С. Мухачева, А. Ф. Валеева, В. М. Картак, *Методы локального поиска оптимума в задачах ортогонального раскроя и упаковки: аналитический обзор и перспективы развития.* — Информационные технологии, № 5 (2004), приложение, 2–17.
10. А. С. Мухачева, *Технология блочных структур локального поиска оптимума в задачах прямоугольной упаковки.* — Информационные технологии, № 5 (2004), приложение, 18–31.
11. И. П. Норенков, *Эвристики и их комбинации в генетических методах дискретной оптимизации.* — Информационные технологии, № 1 (1999), 2–7.
12. И. В. Романовский, *Алгоритмы решения экстремальных задач.* Наука, М. (1977).
13. G. Belov, G. Scheithauer, *A cutting plane algorithm for the one-dimensional cutting stock problem with multiple stock lengths.* — European Journal of Operational Research **141** (2002), 274–294.
14. H. Dyckhoff, G. Scheithauer, J. Terno, *Cutting and packing.* — In: Annotated Bibliographies in Combinatorial Optimization, M. Dell'Amico, F. Maffioli, S. Martello (eds.), John Wiley&Sons (1997), pp. 393–412.
15. H. Dykhoff, *A typology of cutting and packing problems.* — European Journal of Operational Research **44** (1990), 145–159.
16. P. C. Gilmore, R. E. Gomory, *A linear programming approach to the cutting-stock problem.* — Operations Research **9** (1961), 849–859.
17. E. Hopper, B. Turton, *An empirical investigation of meta-heuristic and heuristic algorithms for a 2D packing problem.* — EJOR **128** (2001), 34–57.
18. S. Imahori, M. Yaguira, T. Ibaraki, *Local search heuristics for the rectangle packing problem with general spatial costs.* — In: MIC'2001 – 4th Metaheuristics International Conference (2001), pp. 471–476.
19. O. Marcotte, *The cutting stock problem and integer rouding.* — Math. Program. **33**, No. 1 (1985), 82–92.

20. E. A. Mukhacheva, G. N. Belov, V. M. Kartak, A. S. Mukhacheva, *Linear one-dimensional cutting-packing problems: numerical experiments with sequential value correction method (SVC) and a modified branch-and-bound method (MBB).* — Pesquisa Operacional **20**, No. 2 (2000), 153–168.
21. E. A. Mukhacheva, V. A. Zalgaller, *Linear programming cutting problems.* — International Journal of Software Engineering and Knowledge Engineering **3**, No. 4 (1993), 463–476.
22. J. Terno, R. Lindeman, G. Scheithauer, *Zuschnitprobleme und Ihre Praktische Losung.* Leipzig (1987).

Mukhacheva E. A., Mukhacheva A. S. L. V. Kantorovich and cutting-packing problems: new approaches to combinatorial problems of linear cutting and rectangular packing.

In the middle of the XXth century L. V. Kantorovich and V. A. Zalgaller suggested to solve the problems of thrifty material using while cutting it with the help of linear programming. This resulted in permanent relaxation of planning the rational cutting problem and, as a matter of fact, closed up the problem in mass production. The paper briefly describes the ways of realization of the method for one-dimensional cutting. The problem is extended to integer cases typical for any cutting problem. The block structure technology has been worked out for two-dimensional cutting-packing problems. This technology reduces to solution of some special planning problem of one-dimensional cutting that can be solved by linear programming with the help of simple heuristics. Some calculating schemes and results of numerical experiments with wasteless packing are also shown in the paper. The comparison with other algorithms proves the efficiency of the block method.

Уфимский государственный
авиационный технический
университет
E-mail: elita@vmk.ugatu.ac.ru

Поступило 15 апреля 2004 г.