Уфа: УГАТУ Вестник УГАТУ 2001. № 1 (3). С. 111–119

НАУЧНЫЕ СТАТЬИ И ДОКЛАДЫ • ИНФОРМАТИКА И СИСТЕМОТЕХНИКА

УДК 519.8

# НЕСОБСТВЕННЫЕ ЗАДАЧИ ОПТИМАЛЬНОГО РАСПРЕДЕЛЕНИЯ ДВУХМЕРНОГО РЕСУРСА: ГЕОМЕТРИЧЕСКИЕ ОБРАЗЫ И ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ НА БАЗЕ ИХ БЛОЧНОЙ СТРУКТУРЫ

Э. А. МУХАЧЕВА

Факультет информатики и робототехники УГАТУ
Тел: (3472) 23 79 67 E-mail: Elita@vmk.ugatu.ac.ru

Классическая проблема оптимального распределения ресурса (Resources Distribute, RD) сводится к решению задачи линейного программирования (Linear Programming, LP). Однако в реальных условиях появляется ряд дополнительных ограничений, которые выводят ее из класса собственных задач LP. Вместе с тем геометрическая интерпретация задач RD позволяет применять к ее решению комбинаторные методы. Среди таковых в последнее время широкое распространение получили метаэвристики, в том числе — генетические алгоритмы. Оригинальность использования последних состоит в способах структурирования задач. В статье предлагается два таких способа: блочный и гильотинный. Приведены срезы вычислительных экспериментов

*Оптимальное распределение ресурса; прямоугольная упаковка; раскрой; линейное программирование; генетические алгоритмы*

## ВВЕДЕНИЕ

Под задачами распределения ресурса понимается широкий класс моделей, объединенных однообразной логической структурой и допускающих различное содержание. Логической основой для отнесения какой-либо проблемы к классу задач распределения ресурса является наличие двух групп объектов. К первой группе относятся крупные объекты, ко второй группе — более мелкие элементы. Для каждого объекта и элемента известны объемы запаса и потребления ресурса. Требуется установить соответствие и порядок назначений между элементами и объектами. В отечественной и зарубежной литературе эти задачи встречаются под различными названиями. Остановимся на некоторых из них.

**Задача загрузки рюкзака.** Объектом является рюкзак, элементами - загружаемые в него предметы. Известны предельная грузоподъемность рюкзака, вес, количество и стоимость загружаемых предметов. Необходимо рассчитать загрузку рюкзака с максимальной суммарной стоимостью. Для решения этой задачи применяются методы динамического программирования. Известны более сложные модели задачи о рюкзаке. Например, необходимо распределить все предметы по рюкзакам различной вместимости. Другим ограничением может являться учет центра тяжести предмета. К такой же модели сводится задача распределения одномерного ресурса. Ресурс поступает в виде определенных порций, его необходимо распределить между потребителями. Это может быть банковская задача выдачи кредитов различным потребителям из различных отделений банка с учетом выбранного критерия оптимальности. Другим примером является задача раскроя запаса материала, поступающего в виде стержней на одномерные заготовки. Критерием оптимальности здесь служит общий расход стержней при выполнении плана по заготовкам. Эта классическая модель задачи распределения ресурса широко используется авторами при разработке и исследовании алгоритмов. Следует заметить, что она является NP-трудной и точных методов полиномиальной сложности для ее решения не существует.

Работа поддержана РФФИ, проект 99-01-00937

**Задача расписания обслуживания клиентов.** Например, рассматривается ситуация послепродажного обслуживания автомобилей: постановка на очередь и выполнение заказов. Предположим, что имеется некоторый объем ресурсов (запасных частей, узлов, красителей), при которых суммарный ресурс по одновременно выполненным работам не должен превышать допустимого значения. Известна также продолжительность выполнения каждой работы. Требуется определить последовательность выполнения заказов, минимизирующую общую продолжительность работ. Это задача распределения двухмерного ресурса, который имеет материальную и временную составляющие. Материальная составляющая ограничена, а временная стремится к минимуму. Удобна геометрическая интерпретация этой задачи. Пространство выполнения работ представляется в виде полубесконечной полосы: на оси ординат отражены запасы материального ресурса, а на оси абсцисс — временного. Каждый заказ можно изобразить в виде прямоугольника: ширина указывает необходимый материальный ресурс на выполнение работы, а длина — время (срок выполнения). Тогда эта задача интерпретируется как плотная упаковка прямоугольников в полубесконечной полосе с минимально возможной занятой частью полосы. Для ее решения применяются комбинаторные методы поиска локального или глобального оптимума. Такого вида задачи имеют самостоятельный интерес.

**Задача плотного размещения геометрических объектов в заданной области.** Известны различные постановки такого рода задач. Объектами размещения могут служить прямоугольники, цилиндры, параллелепипеды (контейнеры, ящики), отсеки транспортных средств произвольной конфигурации. Элементами являются, как правило, грузы в виде коробок и ящиков, мебель, станки, детали и узлы. При рассмотрении производственных ситуаций возникает ряд дополнительных, технологических и организационных ограничений. Для задач небольшого объема (количество прямоугольников $<20$) могут применяться точные методы. Широкое распространение получают метаэвристики.

**Задача раскроя запаса материала.** Эта задача является обобщением задачи о рюкзаке при наличии многих объектов. В массовом производстве она часто допускает применение линейного программирования. Более сложный случай — целочисленный. Тогда разрабатываются специальные приемы для ее решения, точные методы типа ветвей и границ, эвристические и метаэвристики. Среди задач раскроя запаса различаются также одно-, двух- и трехмерные задачи. В качестве функции цели могут выступать различные критерии, в том числе и многопараметрические.

Перечень различных задач и интерпретаций для ситуаций распределения ресурса можно продолжить. В настоящей статье в качестве основной рассматривается двухмерная задача, допускающая геометрическую интерпретацию в виде прямоугольной упаковки.

## 1. ОСНОВНЫЕ МОДЕЛИ РАСПРЕДЕЛЕНИЯ РЕСУРСА

В качестве проблемы оптимального распределения ресурса принято рассматривать следующую задачу математического программирования.

**Задача RD.** Имеется $n$ технологических способов распределения (переработки) ресурса (сырья) и $m$ видов заказов (выпускаемой продукции). При этом объем заказов (выход конечной продукции) из единицы ресурса при различных технологических способах характеризуется $m$-мерными векторами

$$\alpha^j = (a_{1j}, a_{2j}, ..., a_{ij}, ..., a_{mj}), \quad j = \overline{1,n}.$$

Компоненты $a_{ij}$ означают объем $i$-го заказа при $j$-м способе распределения ресурса. Кроме того, заданы положительные числа $b_i$, $i = \overline{1,m}$, указывающие пропорцию, в которой требуются различные виды заказов. Задача сводится к определению интенсивностей $x_j$ использования имеющихся технологических способов для получения одного комплекта заказов с минимальными затратами ресурса. Это означает, что ищется $n$-мерный вектор

$$x = (x_1, x_2, ..., x_j, ..., x_n), x_j \geq 0, \quad j = \overline{1,n}, \tag{1}$$

удовлетворяющий системе линейных неравенств

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = \overline{1,m}, \tag{2}$$

и доставляющий минимум линейной функции

$$\mu(x) = \sum_{j=1}^{n} x_j. \tag{3}$$

Векторы интенсивностей (1) называются *допустимыми*, если они удовлетворяют условиям (2). Допустимый вектор, доставляющий минимум функции (3), называется *оптимальным*. Задача (1)–(3) об оптимальном распределении ресурса является задачей линейного программирования. Из общей теории LP вытекает следующий признак оптимальности.

**Утверждение 1.** *Допустимый вектор (1) интенсивностей применения технологических способов распределения ресурса в том и только в том случае является оптимальным, когда заказам можно сопоставить оценки*

$$y_1, y_2, \ldots, y_m, \qquad i = \overline{1, m}, \qquad (4)$$

*удовлетворяющие следующим условиям:*

*а) при этих оценках ни один из способов распределения ресурсов не является сверхрентабельным*

$$\sum_{i=1}^{m} a_{ij} y_i \leq 1, \qquad j = \overline{1, n};$$

*б) в рассматриваемом плане распределения ресурса фактически используются только рентабельные способы, т. е.*

$$\sum_{i=1}^{m} a_{ij} y_i = 1, \quad \text{если} \quad x_j > 0.$$

Величины (4), удовлетворяющие условиям (а) и (б), были названы Л. В. Канторовичем объективно-обусловленными оценками (о. о. оценками) [1]. Использование аппарата о. о. оценок представляет собой мощный *метод последовательного улучшения допустимого вектора* решения задач RD, допускающих толкование в виде LR.

В приведенной модели RD сделаны два важных допущения: векторы $\alpha^j$ наперед заданы и интенсивности $x_j$ их применения являются непрерывными величинами. В реальных задачах, когда ресурс распределяется между большим количеством заказов, каждый способ представляет собой решение самостоятельной технологической задачи. Возникает проблема вычисления и хранения информации о способах распределения ресурса. Если же известен закон формирования способа распределения ресурса (использования сырья), то необходимость в их перечислении отпадает. Задача LP решается на множестве неявно заданных способов с генерацией вводимого в базис вектора $\alpha^j$ на каждом шаге процесса LP [2]. Труднее с требованием непрерывности переменных $x_j$. Смысловое содержание интенсивности применения способа означает целочисленность его применения. Обойти целочисленность удается лишь в условиях массового (крупносерийного) производства, когда расчет делается на крупную партию выпускаемых изделий. Кроме того, встречаются постановки задачи, когда ресурса заведомо не хватает для выполнения заказов целиком. Тогда речь идет о лучшем использовании ресурса исходя из дополнительных соображений. Иногда рассматривают не один, а сразу несколько ресурсов, учет которых может производиться в одинаковых единицах измерения. Это приводит к задачам распределения двух-, трехмерного ресурса. Чаще встречаются одномерные и двухмерные задачи распределения ресурса. Задачи распределения ресурса в приведенных условиях выводят проблему из области собственно LP, и по аналогии с другими прикладными проблемами их можно назвать *несобственными задачами оптимального распределения ресурса*. Логической основой для отнесения какой-либо проблемы к классу задач RD является наличие двух групп объектов. К первой группе относятся, как правило, крупные объекты (носители ресурса), ко второй группе — малые (заказы). Требуется установить соответствие и порядок назначений между заказами и крупными объектами, носителями ресурса. Таким образом, под задачами RD понимается широкий класс моделей, объединенных единой логической основой и допускающих различное толкование. В отечественной и зарубежной литературе они встречаются также под следующими названиями: задача раскроя запаса материала; задача плотного размещения геометрических объектов в заданных областях; задача упаковки контейнеров; задача планирования помещений; задача распределения памяти вычислительной машины; задача составления расписания многопроцессорных систем и другие. Эти задачи допускают изящную геометрическую интерпретацию, что закрепило за классом проблем распределения ресурса название задач раскроя-упаковки (Cutting-Packing, C-P).

Далее в качестве основной рассматривается двухмерная задача RD. Причем одна из компонент ресурса ограничена, вторая — нет. Эта ситуация возникает, когда ресурс имеется в избытке или, наоборот, его заведомо не хватает. Последнее типично для несобственных задач (нет допустимого решения). Ищется план распределения ресурса с минимальным расходом неограниченной компоненты.

Действия по выполнению плана (сокращение запасов или приобретение недостающего ресурса) составляют задачи второй очереди и нами не рассматриваются. Двухмерный ресурс интерпретируется в виде полубесконечной полосы, ширина $W$ которой отвечает ограниченной компоненте ресурса, а длина — второй, неограниченной компоненте. Заказам соответствуют прямоугольники с размерами $w_i; l_i, i = \overline{1,m}$, где размеры прямоугольников — компоненты ресурса для $i$-го заказа. Распределение ресурса означает размещение (упаковку) прямоугольников в полосе параллельно ее граням без пересечений между собой и с гранями полосы. Таким образом, рассматриваемый вопрос об оптимальном распределении двухмерного ресурса сводится к следующей задаче прямоугольной упаковки (2 Dimensional Bin Packing Problem, 2DBPP).

**Задача 2DBPP.** Имеется прямоугольная полоса заданной ширины $W$ и неограниченной длины и набор прямоугольных элементов с размерами $(w_i;\ l_i,\ i = \overline{1,m})$. Введем прямоугольную систему координат: оси $O_x$ и $O_y$ совпадают со сторонами полосы. Горизонтальное положение каждого прямоугольника $i$ зададим вектором $(x_i; y_i)$ с минимальными координатами. Набор векторов $(x_i;\ y_i)$ называется *прямоугольной упаковкой* (Rectangular Packing, RP), если

для $i, j = \overline{1,m};\ \ i \neq j$:

$$\begin{array}{lcl} x_i \geq x_j + l_j & \text{или} & x_j \geq x_i + l_i; \\ y_i \geq y_j + w_j & \text{или} & y_j \geq y_i + w_i; \end{array}$$

для $i = \overline{1,m}$ :

$$x_i \geq 0; \quad y_i \geq 0; \quad y_i + w_i \leq W.$$

Если, кроме того, длина $l(RP)$ занятой части полосы достигает минимума, то RP называется *оптимальной упаковкой* и является решением задачи 2DBPP. Задача 2DBPP, а следовательно, и задача 2DRD являются NP-трудной и переборных алгоритмов полиномиальной сложности для ее решения не существует.

Для данной постановки 2DBPP известны эвристические полиномиальные алгоритмы, порождающие упаковку полосы. Простейший из алгоритмов — «нижний левый» (Bottom Left, BL). Он состоит в поочередном размещении прямоугольников согласно приоритетному списку (Priority List, PL). Аналог этого алгоритма для решения задачи одномерной упаковки, 1DBPP, первый подходящий (First Fit, FFD), часто дает оптимальный результат [3]. Однако в случае 2DBPP оптимум достигается редко. По схеме BL с процедурами приоритета и повторения реализуется метод последовательного уточнения оценок (Sequential Value Correction, SVC). Упорядочивание элементов основано на экономическом смысле объективно-обусловленных оценок Л. В. Канторовича в линейном программировании [4]. В этой статье приведено изложение вероятностных алгоритмов, основанных на линейной аппроксимации прямоугольной упаковки. Это генетический блочный алгоритм (Genetic Block Algorithm, GBA) [5] и генетический гильотинный алгоритм (Genetic Guillotine Algorithm, GGA). Сначала описаны основы детерминированного переборного алгоритма на базе аппроксимации 2DBPP планом линейного раскроя (Reconstruction, REC).

Алгоритмы REC и GBA используют одну и ту же идею: перестановку элементов в линейных раскроях, аппроксимирующих фрагменты (блоки) RP. Перестановка элементов в отдельных раскроях, процедура «*перестройки*» приводят к построению множества различных упаковок специального класса, отвечающих одному и тому же плану линейного раскроя. Однако искомая упаковка может оказаться вне этого класса. Для доказательства этого факта потребуется полный перебор всех перестановок (алгоритм полного перебора). В основе GGA находится методология И. П. Норенкова: использование последовательности эвристик в качестве аллелей генов.

## 2. АППРОКСИМАЦИЯ ДВУХМЕРНОЙ УПАКОВКИ ЛИНЕЙНЫМ РАСКРОЕМ

**Постановка задачи одномерного раскроя (Cutting Stock Problem, CSP).** Заданы длина $L$ раскраиваемого материала (стержней) и длины $l_i$ получаемых из него элементов $m$ наименований, а также необходимое количество $b_i$ каждого элемента вида $i = \overline{1,m}$. Требуется рассчитать оптимальный план раскроя, обеспечивающий минимальный расход стержней.

Задача CSP совпадает с целочисленной задачей RD и описывается моделью линейного целочисленного программирования:

$$\min N = \left\{ \sum_{j=1}^{n} x_j \;\middle|\; \sum_{j=1}^{n} x_j \alpha^j = \beta = \right.$$
$$\left. = (b_1, \ldots, b_m),\ x_j \in Z_+,\ j = \overline{1,n} \right\}, \quad (5)$$

где $\alpha^j = (a_{1j}, a_{2j}, \ldots, a_{mj})$ — векторы, характеризующие способы раскроя; компоненты $a_{ij}$ указывают количество элементов вида $i$, полученных по карте $j$; целая переменная $x_j$ — интенсивность применения раскроя $j$; $n$ — количество различных способов раскроя; $N$ — количество затрачиваемых стержней. Частным случаем при $b_i$=1, $i = \overline{1,m}$, является задача 1DBPP. Обе задачи NP-трудные, для их решения наряду с трудоемкими точными методами применяются различные эвристические алгоритмы [6, 7], в том числе и метаэвристики. Подробно линейная аппроксимация задачи 2DBPP приведена в [5]. Для полноты и ясности дальнейшего изложения повторим здесь основные понятия и утверждения.

Рассмотрим некоторую прямоугольную упаковку. Разобьем ее на прямоугольные блоки одной и той же ширины $W$ и различной длины $L_j$, $j = \overline{1,k}$. При этом начало первого блока совпадает с началом полосы, а конец — с концом самого короткого прямоугольника, входящего в блок. Каждый последующий блок $j$ =2,... начинается, как только заканчивается предыдущий. Каждый блок мысленно разрезаем на одинаковые полосы длиной 1 мм и шириной $W$. Тогда блоку $j = \overline{1,k}$ можно сопоставить линейный раскрой $j$ полос длины $W$ с интенсивностью применения $x_j$, равной длине блока. Прямоугольная упаковка может быть представлена в виде совокупности кортежей

$$(1(j), 2(j), \ldots)x(j), \quad j = 1, 2, \ldots \tag{6}$$

Этот список представляет собой решение CSP с исходной информацией вида: $(L; m; \lambda_i; b_i)$, где $L = W$ — длина стержня; $m$ — количество различных элементов; $\lambda_i = w_i$ — длина элемента $i$; $b_i = l_i$ — требуемое количество $i$-х элементов.

Обозначим через $\pi$ план линейного раскроя (допустимое решение задачи (4)), через $\lambda(\pi)$ — соответствующее значение функции цели.

План $\pi$ линейного раскроя, отвечающий RP, назовем *прямоугольно-ориентированным линейным раскроем* (Rectangular Oriented Linear Cutting, ROLC).

Очевидно следующее

**Утверждение 2.** *Для любой допустимой упаковки RP длина $l(RP)$ занятой части полосы совпадает со значением линейной функции $\lambda(\pi)$ в соответствующей задаче линейного раскроя.*

Обозначим: $I_j$ — список номеров прямоугольников, принадлежащих $j$-му кортежу; $I_j^+ = \{i \in I_j / i \notin I_{j+1}\}$ — список номеров прямоугольников, заканчивающихся в $j$-м кортеже; $I_j^- = \{i \in I_j / i \notin I_{j-1}\}$ — список номеров прямоугольников, начинающихся в $j$-м кортеже.

**Утверждение 3.** *Для того чтобы план $\pi$ линейного раскроя, представленный списком кортежей вида (6), являлся ROLC, необходимо и достаточно выполнения следующих условий:*

$1^0$. *Элементы $i(j)$ любого кортежа $j$ различные* (разнородность).

$2^0$. *Если $i \in I_\mu$ и $i \notin I_\mu^0$, то $i \in I_{\mu+1}$, т.е. элемент $i$ является обязательным в кортеже $(j+1)$* (продолженность).

$3^0$. *Место элемента постоянно во всех кортежах, в которых он содержится* (постоянство места).

$4^0$. *Если для некоторых $i$ и $j$ $i \in I_{j+k}$, $k = \overline{0,p}$; $i \in I_j^-$; $i \in I_{j+p}^+$, то $l_i = \sum_{k=0}^{p} x_{j+k}$* (соответствие интенсивностей длинам прямоугольников);

$5^0$. *Если элементы $i \in I_j^+$ размещены в нескольких непрерывных областях блока (кортежа) $j$, то в продолжениях этих областей на $(j+1)$-й блок размещаются элементы $i \in I_{j+1}^-$* (размещаемость).

Линейный раскрой, отвечающий исходным данным в задаче 2DBPP и удовлетворяющий условиям $1^0$ и $2^0$, назовем *ослабленным прямоугольно ориентированным линейным раскроем* (Delicate ROLC, d.ROLC).

Различные d.ROLC, отличающиеся только перестановкой элементов в блоках, являются *эквивалентными* d.ROLC, они представляют идентичные планы $\pi$ линейного раскроя.

**Утверждение 4.** *Для того чтобы d.ROLC или эквивалентный ему (d.ROLC)′ с исходными данными задачи 2DBPP был ROLC и отвечал прямоугольной упаковке длины $l = \lambda$, необходимо и достаточно выполнения условия $5^0$ утверждения 3.*

Назовем d.ROLC *нормальным* (*Normal*, n.d.ROLC), если существует ему эквивалентный (d.ROLC)′, являющийся ROLC для RP с $l = \lambda$.

**Утверждение 5.** *Если d.ROLC является n.d.ROLC, то искомый ROLC для RP с $l = \lambda$ можно найти на множестве эквивалентных (d.ROLC)′.*

Предположим, что задача d.ROLC решена и известно оптимальное решение длины $\lambda$. Это решение представлено списком кортежей вида (6). Каждому кортежу отвечает блок, удовлетворяющий условиям $1^0$ и $2^0$. Однако совокупность таких блоков не обязательно представляет прямоугольную упаковку (Утверждение 4). Определяющим для RP с $l = \lambda$ является выполнение условия $5^0$ Утверждения 2. Предположим, что для некоторой пары соседних кортежей $(j, j+1)$ нарушено условие $5^0$. В этом случае будем говорить, что при переходе с кортежа $j$ к кортежу $(j+1)$ возникла ситуация *перестройки*. Процедуру, реализующую нужную перестановку прямоугольников в блоках (элементов в кортежах), назовем *перестройкой* (Reconstruction, REC). Быстрые переборные алгоритмы REC описаны в [8]. Там же приведены результаты вычислительного эксперимента.

## 3. ГЕНЕТИЧЕСКИЙ БЛОЧНЫЙ АЛГОРИТМ (GBA)

**Характеристика алгоритма.** Алгоритм решения задачи 2DBPP интерпретируется как эволюционный процесс, связанный с перестановкой элементов в кортежах и соответственно в PL с целью отыскания глобального минимума для $l$.

Каждой допустимой упаковке RP отвечает ее блочное представление (6). Кортежи в (6) расположены в установленном $v = 1, 2, \ldots, n$ порядке и связаны друг с другом (условие 2°). Это позволяет интерпретировать их как *гены*. Тогда блочную структуру, соответствующую d.ROLC, можно интерпретировать *хромосомой*, содержащей $k$ сцепленных между собой *генов* (кортежей), которые расположены «слева–направо». Согласно хромосомной теории наследственности передача качественных принципов, закодированных в генах, будет осуществляться через хромосомы от «родителей» к «потомкам».

Местоположение определенного гена в хромосоме является *локусом*, а альтернативные формы одного и того же гена, расположенные в одинаковых локусах хромосомы, называются *аллелями*. В случае RP аллели представляют различные перестановки элементов в генах одного и того же локуса $(v)$.

Хромосома, содержащая в своих локусах конкретные значения аллелей, является *генотипом*. Конечное множество всех допустимых генотипов образует *генофонд*.

При взаимодействии особи $rp$ с внешней средой ее генотип характеризуется *степенью приспособленности* $\mu(rp)$ особи $rp$ к внешней среде. В качестве внешней среды принимается критерий оптимальности. В рассматриваемом случае степенью приспособленности $\mu(rp)$ особи $rp$ является значение $l(rp)$ длины занятой части полосы прямоугольной упаковки RP. В данном случае чем меньше $\mu(rp) = l$, тем лучше особь $rp$ приспособлена к внешней среде.

В качестве *ареала* — области, в пределах которой только и могут встречаться особи, участвующие в эволюционном процессе, — будет рассматриваться область поиска RP. При этом множество эквивалентных d.ROLC образует *усеченный ареал*, а совокупность особей $(rp_1, rp_2, \ldots, rp_\rho)$, принадлежащих ареалу, образует *популяцию* P. Число $\rho$, характеризующее количество особей в популяции, называют *численностью* популяции. Численность генофонда популяции в множестве эквивалентных d.ROLC, $\rho = \sum_{v=1}^{n} r_v!$, где $r_v$ — количество начатых в блоке $v$ элементов. В общем случае экстремальной задачи 2DBPP популяция соответствует совокупности всех допустимых решений. С помощью основных генетических процедур «*скрещивание*» и «*отбор*» может быть найдена особь с показателем $l = \lambda$ в случае, когда начальная хромосома (d.ROLC) является нормальной (Утверждение 5). Иначе *ареал* расширяется за счет мутации, в нашем случае введения сильно ослабленных (удовлетворяющих только условию 1°) ROLC (e.d.ROLC). Численность генофонда при этом $\rho = n! \sum_{v=1}^{n} r_v!$

**Численный эксперимент.** Для проведения эксперимента использовался генератор линейных задач BPPGEN, преобразованный для генерации задач прямоугольной упаковки [3]. Каждая шестерка $(W; m; v_1; v_2; \omega_1; \omega_2)$ входных данных генератора описывает два специальных класса задач BPP или, что то же самое, один класс задач 2DBPP. Параметрами задачи RP являются: $W$ — ширина полосы; $m$ — количество прямоугольников (размерности задачи); $v_1$ — нижняя граница ширины предметов по отношению к ширине $W$ полосы, т. е. $w_i \geq v_i \times W$, $i = \overline{1, m}$; $v_2$ — верхняя граница ширины предметов по отношению к ширине $W$ полосы, т. е. $w_i \leq v_i \times W$, $i = \overline{1, m}$; $w_I$ — нижняя граница длины предметов по отношению к ширине $W$ полосы, т. е. $l_i \geq w_i \times W$, $i = \overline{1, m}$; $w_2$ — верхняя граница длины предметов по отношению к ширине $W$ полосы, т. е. $l_i \geq w_2 \times W$, $i = \overline{1, m}$.

Во всех экспериментах $W = 225$; $m \leq 80$; $v_1 = 0{,}15$; $v_2 = 0{,}2, 0{,}4, 0{,}6$; $w_1 = 0{,}15$; $w_2 =$

$= 0{,}5$. Каждый эксперимент проводился в одном классе задач. В каждом классе решалось по 50 задач.

Таблица 1

**Результаты численного эксперимента с GBA**

$v_1 = 0{,}15;\ w_1 = 0{,}15;\ w_2 = 0{,}5$

| $v_2$ | m=20 | | m=40 | | m=60 | | m=80 | |
|---|---|---|---|---|---|---|---|---|
| | $\Delta_{rec}$ | $\Delta_{gba}$ | $\Delta_{rec}$ | $\Delta_{gba}$ | $\Delta_{rec}$ | $\Delta_{gba}$ | $\Delta_{rec}$ | $\Delta_{gba}$ |
| 0.2 | 6.8 | 0.38 | 6.7 | 1.3 | 4.2 | 0.56 | 3.85 | 0.91 |
| 0.4 | 6.6 | 0.73 | 7.3 | 1.73 | 7.1 | 2.62 | 7.04 | 2.67 |
| 0.6 | 2.2 | 0.15 | 4.42 | 0.78 | 4.6 | 0.75 | 2.2 | 1.18 |

Для этого эксперимента были использованы методы REC и GBA. В табл. 1 указана эффективность алгоритмов, оцененная по отклонению $\Delta_{alg}$ в % полученных решений от эталона $\lambda$ нижней границы занятой части полосы. Подчеркнуты в таблице лучшие решения. Эксперимент проведен на компьютере Pentium II 400 Mhz.

### 4. ГЕНЕТИЧЕСКИЙ ГИЛЬОТИННЫЙ АЛГОРИТМ

**Характеристика алгоритма.** При исходных данных задачи 2DBPP требуется найти гильотинный раскрой рулона на заданные элементы, который обеспечит минимальную длину $l^*(PR)$ использованной части полосы. Под гильотинным понимается раскрой, реализуемый последовательностью сквозных резов, параллельных кромкам материала.

Эта задача подробно рассматривалась различными авторами [9]. В частности, в работе [10] приведены послойные алгоритмы для задачи гильотинного раскроя в условиях единичного производства. Решение с помощью детерминированного послойного алгоритма с $l \geq l^*$ находится за один проход. При этом возникает эффект «хвоста», когда в конце полосы остаются «плохо» заполненные области. Этот недостаток частично сглаживается за счет применения оценок прямоугольников, аналогично интерпретации двойственных переменных (оценок) Л.В. Канторовича, *метода последовательного уточнения оценок* [11]. При этом исправление раскроя практически происходит только на нескольких первых итерациях, далее процесс оказывается вялотекущим и возможно зацикливание. Для решения задачи 2DBPP в гильотинной постановке предлагается генетический алгоритм, которому в меньшей степени присущи указанные недостатки. Ген означает способ упаковки очередного элемента, аллели — последовательности алгоритмов для его упаковки. Принципиальным здесь оказывается представление структуры гильотинного раскроя как последовательности вложенных друг в друга прямоугольных блоков, именуемых далее *корзинами*. Последовательность аллелей представляет собой различные алгоритмы типа *первый подходящий* и *жадный* алгоритм. В случае случайного выбора алгоритма *первый подходящий* укладывается один или несколько одинаковых элементов и формируется новая корзина, в случае жадного алгоритма заполняется наилучшим образом боковая грань очередной корзины.

Сформулируем эвристики, с помощью которых решаются подзадачи. Они основаны на очень простых стратегиях и часто приводят к хорошим результатам. Среди них выделяются методы типа *первый подходящий* и *жадные* алгоритмы. Эвристики для решения задач линейного раскроя приведены в работе [7] и адаптированы нами для двухмерного случая. Нами использовались следующие алгоритмы.

- *First-fit* (FF) — *первый подходящий*. На каждом шаге текущий элемент помещается в частично заполненную корзину с наименьшим порядковым номером, имеющую подходящую остаточную емкость. Если ни одна корзина не удовлетворяет этим условиям, то элемент упаковывается в новую корзину.
- *Best-fit* (BF) — *лучший подходящий*. На каждом шаге текущий элемент сопоставляется той корзине, которая имеет наименьшую допустимую остаточную емкость. Предпочтение отдается корзине с наименьшим порядковым номером. В случае отсутствия таковой берется новая.
- *Worst-fit* (WF) — *худший подходящий*. В противоположность «лучшему подходящему» выбирается корзина с наибольшей подходящей остаточной емкостью. Если не существует ни одной, берется новая корзина.

Причем, если предметы могут быть ориентированы двумя способами, количество эвристик удваивается.

- *Gridi* (Gr) — *жадный алгоритм*. Сводится к решению очередной задачи 0–1 рюкзак [12]. При этом оптимизация осуществляется по ширине корзины, а длина $l$ заполненной части корзины определяется длиной самой длинной вложенной в неё заготовкой. Остатки по длине определяют новые корзины.

Структура хромосомы в генетическом алгоритме выглядит следующим образом: $[(h_1, q_1), (h_2, q_2), ..., (h_P, q_P)]$, где $h_j$ — номер эвристики; $q_j$ — количество предметов, размещаемых с помощью указанной эвристики.

Генетические операции *кроссовер* и *мутация* определяются в классическом варианте генетического алгоритма [13]. Однозначное восстановление раскроя из такой хромосомы подразумевает наличие приоритетного списка определенной каким-либо образом последовательности номеров типов элементов (например, по убыванию площади предметов).

Функцией приспособленности для особей будет длина занятой части полосы $l^*$, полученная после восстановления раскроя. Чем меньше $l^*$, тем более приспособленной является особь.

Оценить качество полученного раскроя можно с помощью коэффициента раскроя (КРА), равного отношению полезной площади ко всей израсходованной.

Структура упаковки представляется в виде линейного списка, состоящего из звеньев следующего вида: $(N, x, y, w_b, l_b, n_{\text{пр}}, k)$, где $N$ — порядковый номер корзины; $x, y$ — координаты левого нижнего угла корзины; $w_b, l_b$ — ширина и длина корзины; $n_{\text{пр}}$ — тип предмета (номер в приоритетном списке); $k$ — количество размещенных предметов.

Весь список будет иметь вид:

*Начало*
$\langle (N_1, x_1, y_1, w_{b1}, l_{b1}, n_{\text{пр}1}, k_1)?$
$(N_2, x_2, y_2, w_{b2}, l_{b2}, n_{\text{пр}2}, k_2)?$
$\ldots ?$
$(N_s, x_s, y_s, w_{bs}, l_{bs}, n_{\text{пр}s}, k_s) \rangle$
*Конец.*

В этом списке каждой корзине соответствует одно звено.

**Численный эксперимент.** Параметры исследуемых задач совпадают с описанными в разд. 2. В дополнение вводится параметр $M$ — общее количество элементов.

Таблица 2

**Результаты численного эксперимента с GGA**

$m = 20;\ v_1 = 0{,}15;\ w_1 = 0{,}15;\ w_2 = 0{,}5$

| $v_2$ | $m = 20$ | | $m = 40$ | | $m = 60$ | | $m = 80$ | |
|---|---|---|---|---|---|---|---|---|
| | $K$ (fbg) | $K$ (fbwg) | $K$ (fbg) | $K$ (fbwg) | $K$ (fbg) | $K$ (fbwg) | $K$ (fbg) | $K$ (fbwg) |
| 0.2 | 91.39 | 91.26 | 93.83 | 93.70 | 93.82 | 94.00 | 93.83 | 94.45 |
| 0.4 | 90.09 | 90.78 | 93.60 | 92.82 | 93.66 | 93.21 | 93.93 | 93.67 |
| 0.6 | 91.04 | 91.79 | 93.24 | 93.64 | 93.70 | 94.10 | 94.75 | 94.36 |

В табл. 2 приведен срез эксперимента для случая, когда $m = 20$, а общее количество элементов $M = 20, 40, 60, 80$; $v_1 = 0{,}15$; $v_2 = 0{,}2; 0{,}4; 0{,}6$; $w_1 = 0{,}15$; $w_2 = 0{,}5$. Эксперимент проведен в следующих вариантах:

Вариант 1. Совокупность алгоритмов FF, BF и Gr (fbg).

Вариант 2. Совокупность алгоритмов FF, BF, WF и Gr (fbwg).

Эффективность алгоритмов оценивается показателем

$$K = \frac{1}{Wl} \sum_{i=1}^{m} w_i l_i,$$

где $l$ — достигнутая длина полосы.

## ЗАКЛЮЧЕНИЕ

Рассмотрена задача распределения ресурса с рядом дополнительных ограничений, выводящих ее из области собственно линейного программирования. Задача RD сведена к решению двухмерной проблемы прямоугольной упаковки, 2DBPP. Для решения последней, которая является NP-трудной, предложены два генетических алгоритма. Первый, блочный алгоритм, основан на аппроксимации 2DBPP задачей линейного раскроя. Второй, гильотинный алгоритм, базируется на гильотинной структуре упаковки. Алгоритмы обладают определенными достоинствами и недостатками. К достоинствам можно отнести их высокую эффективность по сравнению с детерминированными эвристиками, быстроту получения решения, гибкость и вариативность. Недостатком блочного алгоритма является то, что он не учитывает возможности поворота элементов на 90°. С помощью гильотинного алгоритма можно получать решения задачи упаковки, 2DBPP, однако эффективность его применения для этого случая весьма сомнительна. Следует продолжать работу над совершенствованием алгоритмов в следующих направлениях:

- гибридизация с детерминированными процедурами перестройки;
- обеспечение поворота элементов в GBA;
- расширение перечня элементарных алгоритмов и вариантов их применения в GGA;
- сравнение эффективности GBA и GGA для $M = m$;
- расширение областей эксперимента с использованием новых данных Internet-sait;
- сравнение эффективности с другими метаэвристиками, например, с алгоритмом *поиска с запретами*.

## СПИСОК ЛИТЕРАТУРЫ


1. **Канторович Л. В., Залгаллер В. А.** Рациональный раскрой промышленных материалов. Новосибирск: Наука, Сиб. отд., 1971. 320 с.
2. **Мухачева Э. А., Рубинштейн Г. Ш.** Математическое программирование // Новосибирск: Наука, Сиб. отд., 1987, 272 с.
3. **Schwerin P., Wascher G.** The bin-packing problem: a problem generator and some numerical experiments with FFD packing and MTP // Int. Trans. in Operational Research. 1997. Vol. 4, No 5/6. P. 337–384.
4. **Mukhacheva E. A., Zalgaller V. A.** Linear programming cutting problems // Int. J. of Software Eng. and Knowledge Eng. 1993. V. 3. No 4. P. 463–476.
5. **Мухачева Э. А., Мухачева А. С., Чиглинцев А. В.** Генетический алгоритм блочной структуры в задачах двухмерной упаковки // Информационные технологии. М.: Машиностроение, 1999. № 11. С. 13–18.
6. **Мухачева Э. А., Мухачева А. С., Белов Г. Н.** Метод последовательного уточнения оценок: алгоритм и численный эксперимент для задачи одномерного раскроя // Информационные технологии. М.: Машиностроение, 2000. № 2. С. 11–17.
7. **Scholl A., Klein R., Juergens G.** BISON: A fast hybrid procedure for exactly solving the on-dimensional Bin-Packing Problem // Computers and Operational Research. 1997. V. 24, No 7. P. 627–645.
8. **Мухачева Э. А., Мухачева А. С.** Метод перестройки для решения задачи прямоугольной упаковки // Информационные технологии. М.: Машиностроение, 2000. № 4. С. 30–36.
9. **Мухачёва Э. А.** Рациональный раскрой промышленных материалов. Применение в АСУ // М.: Машиностроение, 1984. 176 с.
10. **Adamowich M., Allano A.** A solution of the rectangular cutting-stock problem // Trans. on System, Man. 1976. V. SMC-6, №4. P. 302–310.
11. **Мухачёва Э. А., Верхотуров М. А., Мартынов В. В.** Модели и методы расчета раскроя-упаковки геометрических объектов // Уфа: УГАТУ, 1998. 216 с.
12. **Мухачёва Э. А., Валеева А. Ф.** Методы динамического перебора в задачах двухмерной упаковки // Информационные технологии. М.: Машиностроение, 2000. № 5. С. 30–36.
13. **Батищев Д. И.** Генетические алгоритмы решения экстремальных задач / Под ред. акад. АЕН Я. Е. Львовича: Учеб. пособие. Воронежск. гос. техн. ун-т; Нижегородск. гос. ун-т. Воронеж, 1995. 69 с.


## ОБ АВТОРЕ

**Мухачева Элита Александровна,** проф. каф. вычисл. математики и кибернетики УГАТУ. Дипл. преподаватель математики (БГПИ 1952), д-р техн. наук по системам автоматизации проектирования (защ. в Мосстанкине, 1984). Труды по исследованию операций, математическому моделированию, эффективности алгоритмов.