

Эволюционный алгоритм решения задачи рационального раскроя рулонного материала

В. Н. Балабанов*, Ю. А. Скобцов

Донецкий национальный технический университет

Препринт: 3 июня 2013 г.

Аннотация

Рассматривается задача рационального раскроя рулонного материала в многокритериальной постановке. Уточнения, внесенные в формальную модель, отражают необходимость учитывать технологический аспект при составлении планов продольного раскроя рулонов. Предложенный метод решения задачи рационального раскроя основан на оптимизационном аппарате эволюционных вычислений. При выборе способа представления решений в эволюционном алгоритме учтена комбинаторная природа исходной задачи. Комбинаторный подход при разработке эволюционных операторов усилен применением адаптивной генерации раскройных карт. Подготовлены тестовые задачи и поставлен ряд вычислительных экспериментов. Полученные результаты подтверждают состоятельность избранного подхода к построению эволюционного алгоритма, который может быть адаптирован для решения других задач рационального раскроя в схожей постановке.

Ключевые слова: рациональный раскрой, рулонные материалы, многокритериальная модель, комбинаторная оптимизация, эволюционный алгоритм.

1 Введение

Задача о рациональном раскрое впервые была сформулирована выдающимся советским математиком и экономистом Л. В. Канторовичем [1] и состоит в нахождении такого плана раскроя, который позволяет получить заготовки в нужном ассортименте и при этом обеспечивает минимальный расход материала. В общем случае задача о рациональном раскрое может быть классифицирована как NP-сложная задача комбинаторной оптимизации. Логическая структура раскройных задач во многом определяется формой и размерами применяемых заготовок, а также ассортиментом исходного материала [2]. В частности, различают одно- и двухмерный, единичный и комбинированный, прямоугольные произвольный и гильотинный, а также фигурный раскрой.

План, являющийся решением задачи рационального раскроя, представляет собой перечень раскройных карт, которые определяют способы раскроя отдельных частей исходного материала (например, рулонов или листов) на наборы заготовок. При построении формальных моделей задач рационального раскроя наибольшее распространение получил подход, основанный на сведениях исходной комбинаторной задачи к

*E-mail: akavrt@gmail.com

задаче целочисленного линейного программирования общего вида [3]. Критерий оптимальности формулируется таким образом, чтобы при поиске решений минимизировать расход, стоимость или потери исходного материала, использованного в раскрое. Актуальным направлением является построение уточненных многокритериальных моделей, в которых более полно учитываются ограничения, налагаемые конструкцией раскройного оборудования, либо раскрой рассматривается как одна из стадий технологического процесса производства некоторой продукции. Наряду с расходом материала наиболее часто требуется минимизировать общее количество настроечных операций, производимых с оборудованием в ходе выполнения плана раскроя [4, 5]. Многокритериальные модели, в которых совместно рассматривается более двух различных критериев, встречаются значительно реже [6].

В решении задач рационального раскроя широко используются как точные, так и приближенные методы [7, 8]. Применимость точных методов до последнего времени была ограничена задачами в классической постановке, предполагающей рассмотрение только одного оптимизационного критерия — минимизации расхода исходного материала. Для решения практических задач рационального раскроя был предложен ряд эвристических методов, которые позволяют находить решения, удовлетворяющие требованиям реального производства [4, 9–13].

Рулонные материалы широко применяются в металлургической, целлюлозно-бумажной, текстильной, химической и целом ряде других отраслей промышленности. В данной статье рассматривается задача рационального раскроя рулонного материала, выполняемого с помощью специализированного оборудования — линий продольной резки рулонов. При планировании раскроев такого типа требуется учитывать дополнительные критерии, применяемые для оценки качества получаемых решений, и ограничения, связанные с особенностями технологического процесса продольного раскроя рулонов [4, 11]. Изложение материала статьи организовано следующим образом: вначале дается уточненная формальная модель задачи рационального раскроя рулонов на линиях продольной резки, далее подробно рассматриваются различные аспекты построения эволюционного алгоритма, обсуждаются результаты вычислительных экспериментов и делаются соответствующие выводы.

2 Постановка задачи

При раскрое рулонов на линиях продольной резки материал раскраивается на полосы заданной ширины, все продольные резы выполняются от края до края, параллельно боковой кромке исходной полосы. Следовательно, продольный раскрой рулонного материала можно классифицировать как одномерный. В тоже время, в отличие от классической постановки одномерной задачи рационального раскроя, основанной на поштучном способе учета комплектности раскроя (когда требуемое количество заготовок каждого типа задается целым числом), при продольном раскрое рулонов может указываться только требуемая общая длина или масса полос заданной ширины, но не их точное количество. В некоторых источниках такой вид раскроя получил название полуторамерного [11].

Введем ряд обозначений. Положим, что ассортимент заготовок, которые требуется получить в результате раскроя рулонов, ограничен m наименованиями. Размеры заготовок и комплектность раскроя задаются следующим образом: для получения заготовки длиной l_i допускается использование произвольного количества отдельных полос шириной w_i , $i = 1, \dots, m$. Пусть исходный материал представлен перечнем из n рулонов, с каждым из которых сопоставлен порядковый номер $j = 1, \dots, n$. При-

годные для использования длину и ширину рулона с номером j обозначим через L_j и W_j , соответственно.

Множество раскройных карт опишем с помощью целых чисел a_{ik} , которые определяют количество полос шириной w_i , полученное при раскрое рулона по способу с номером k . Для того чтобы установить связь между раскройными картами и рулонами, введем булевы переменные T_{jk} :

$$T_{jk} = \begin{cases} 1, & \text{рулон } j \text{ раскраивается по способу } k, \\ 0, & \text{в противном случае.} \end{cases}$$

Если предположить, что все допустимые раскройные карты перечислены по $k = 1, \dots, K$, то задача нахождения рационального плана, который позволит раскроить рулоны на наборы полос заданного размера с минимальными потерями исходного материала, может быть сформулирована следующим образом:

$$z_1 = \min \sum_{k=1}^K \sum_{j=1}^n T_{jk} L_j \left(W_j - \sum_{i=1}^m a_{ik} w_i \right) \quad (1)$$

$$\sum_{k=1}^K \sum_{j=1}^n T_{jk} a_{ik} L_j \geq l_i \quad \forall i \quad (2)$$

$$T_{jk} \sum_{i=1}^m a_{ik} w_i \leq W_j \quad \forall j, k \quad (3)$$

$$\sum_{k=1}^K T_{jk} \leq 1 \quad \forall j \quad (4)$$

$$i \in \{1, \dots, m\}, j \in \{1, \dots, n\}, k \in \{1, \dots, K\}.$$

Известно, что операции настройки отдельных агрегатов линии продольной резки рулонов, выполняемые при смене раскройной карты, характеризуются значительной трудоемкостью [4, 11]. Повышение качества составляемых планов раскроя может быть достигнуто за счет повторного использования раскройных карт: при раскрое двух и более рулонов на одинаковые наборы полос количество необходимых настроечных операций сокращается, поскольку смена раскройной карты в таком случае не требуется. Для того чтобы формализовать данный критерий, введем вспомогательную функцию (5).

$$\delta(k) = \begin{cases} 1, & \text{если } \sum_{j=1}^n T_{jk} > 0, \\ 0, & \text{в противном случае.} \end{cases} \quad (5)$$

Критерий оптимальности для задачи минимизации количества различных раскройных карт, используемых в плане раскроя, может быть записан следующим образом:

$$z_2 = \min \sum_{k=1}^K \delta(k). \quad (6)$$

Следовательно, задача рационального планирования продольного раскроя рулонного материала может рассматриваться как многокритериальная задача комбинаторной оптимизации с векторным критерием (7) и системой линейных ограничений (2)–(4).

$$Z = (z_1, z_2) \quad (7)$$

Зачастую при рассмотрении задач рационального раскроя в подобной либо схожей постановке вместо многокритериальной решается задача скалярной оптимизации: вводятся коэффициенты, характеризующие относительную важность частных критериев, а векторный критерий заменяется соответствующим ему обобщенным критерием [4, 11, 14]. Эволюционные метаэвристические методы успешно применяются для решения задач комбинаторной оптимизации, в том числе и задач рационального раскроя [12]. С учетом комбинаторной сложности рассматриваемой задачи выбор эволюционного алгоритма в качестве общей схемы для построения приближенного метода решения задачи рационального раскроя рулонов представляется логичным.

3 Эволюционный алгоритм

Метаэвристические методы, на каждой итерации работающие с некоторым множеством текущих решений заданной мощности, условно выделяют в самостоятельный класс популяционных метаэвристик. Наибольшее распространение из числа методов, относящихся к данному классу, получили эволюционные алгоритмы, которые для эффективной организации поиска решений оптимизационных задач заимствуют ряд базовых принципов из дарвиновской теории эволюции [15]. В зависимости от особенностей решаемой задачи может использоваться дополнительное кодирование решений, когда с помощью некоторого правила решения преобразуются в «особи» — структуры данных, специфичные для той или иной разновидности эволюционного алгоритма. На этапе инициализации алгоритма формируется начальная популяция заданного размера, составленная из особей, сгенерированных случайным образом или же полученных с помощью вспомогательной процедуры. Каждая итерация эволюционного алгоритма состоит из последовательного выполнения эволюционных операторов, изменяющих структуру особей в текущей популяции, и формирования новой популяции при помощи механизма, имитирующего естественный отбор: в новую популяцию переносятся лишь наиболее приспособленные особи из текущей популяции. Степень приспособленности определяется значением целевой функции, рассчитанным для решения, соответствующего оцениваемой особи. Например, при решении задач минимизации вероятность попадания в следующую популяцию выше у особей с меньшими значениями целевой функции. Таким образом, для построения эволюционного алгоритма решения оптимизационной задачи требуется выбрать подходящий способ представления решений в виде особей, сформулировать целевую функцию, разработать эволюционные операторы и выполнить настройку параметров алгоритма.

При поиске решения рассматриваемой задачи требуется из множества исходных рулонов S выделить подмножество S_{used} , содержащее только те рулоны, которые будут раскроены, а также определить способы раскроя этих рулонов. В обозначениях формулировки (1)–(7):

$$S = \{1, \dots, n\}, S_{\text{used}} = \left\{ j \in S \mid \sum_{k=1}^K T_{jk} = 1 \right\}.$$

Следует отметить, что множество допустимых раскройных карт в формулировке (1)–(7) задано неявно, поскольку на практике задача генерации и перебора всех его элементов является трудноосуществимой вследствие значительной мощности множества. Исходя из этого, предлагается применить адаптивную генерацию раскройных карт, выполняемую по мере необходимости в процессе поиска решений с помощью специальной процедуры. На эволюционный алгоритм при этом возлагаются функ-

ции по подбору рулонов, вызову процедуры генерации раскройных карт и внесению соответствующих изменений в структуру плана раскроя.

3.1 Способ представления решений

Некоторые разновидности эволюционных алгоритмов, например, генетические алгоритмы и эволюционные стратегии, изначально разрабатывались как методы решения задач численной оптимизации — способы представления решений таких задач, предложенные в ранних работах, со временем стали считаться стандартными. Совершенно иначе обстоит дело при решении задач комбинаторной оптимизации, когда выбор правила, в соответствие с которым решения задачи преобразуются в особи, определяется логической структурой решаемой задачи.

Как уже было отмечено, при составлении плана продольного раскроя рулонов требуется учитывать не только потери материала, но также контролировать количество введенных в план различных способов раскроя рулонов. Удобным для реализации эффективных эволюционных операторов представляется такой способ представления решений, при котором отдельные рулоны, включенные в план, будут объединяться в группы G_q , каждой группе рулонов будет соответствовать собственный, отличный от других групп способ раскроя материала. Предположим, что номера раскройных карт, используемых в плане, известны и равняются P_q , $q = 1, \dots, Q$, тогда задача разбиения множества рулонов S_{used} на группы G_q может быть сформулирована следующим образом:

$$G_q = \{j \in S_{\text{used}} \mid T_{jk} = 1, k = P_q\}, P_q \in \{1, \dots, K\}, \sum_{k=1}^K \delta\left(k, \sum_{j=1}^n T_{jk}\right) = Q,$$

$$\bigcup_{q=1}^Q G_q = S_{\text{used}}, G_q \cap G_r = \emptyset \text{ при } q \neq r \text{ и } q, r \in \{1, \dots, Q\}.$$

Если изменения, вносимые в особь эволюционным оператором, изолируются в пределах некоторой группы $G_q \rightarrow G_q^*$ и $|G_q^*| \geq |G_q|$, то значение Q после выполнения такого оператора остается неизменным. Применение в эволюционном алгоритме операторов, которые соответствуют сформулированному выше требованию, упростит поиск решений с нужными характеристиками за счет более эффективного контроля значений критерия (6).

3.2 Целевая функция

Решение задачи рационального раскроя в многокритериальной постановке (1)–(7) предполагает использование математического аппарата Парето-оптимальности. Однако, при решении большинства практических задач рационального раскроя имеется возможность сопоставить стоимость единицы площади рулонного материала со средней трудоемкостью выполнения операций по настройке агрегатов линии продольной резки рулонов на заданную раскройную карту. Следовательно, система предпочтений лица, принимающего решение, в данном случае — составителя планов раскроя, может быть описана с помощью коэффициентов относительной важности C_1 и C_2 частных критериев (1) и (6), соответственно. Заменим векторный критерий (7) обобщенным критерием (8), который будем использовать в качестве целевой функции в эволюционном алгоритме.

$$Z^* = \min (C_1 z_1^* + C_2 z_2^*), \quad (8)$$

$$C_1 \geq 0, C_2 \geq 0, C_1 + C_2 = 1.$$

Одним из требований, предъявляемых к допустимым планам раскроя является их комплектность, т.е. возможность получить в результате выполнения плана заготовку каждого типа требуемой длины. В предложенной формулировке (1)–(7) данному требованию соответствует ограничение (2). Для удобства обозначим длину заготовки шириной w_i , полученную в результате выполнения плана раскроя через l_i^{prod} :

$$l_i^{\text{prod}} = \sum_{k=1}^K \sum_{j=1}^n T_{jk} a_{ik} L_j.$$

На практике для оценки качества допустимых планов раскроя также может использоваться критерий (9), который в дополнение к комплектности позволяет контролировать избыточную длину получаемых заготовок.

$$z_3 = \min \sum_{i=1}^m \left(l_i^{\text{prod}} - l_i \right) \quad (9)$$

Для того чтобы не вводить в целевую функцию третий частный критерий и коэффициент C_3 , избыточную длину полученных заготовок будем рассматривать как потери материала. Если предположить, что значение целевой функции рассчитывается только для тех особей, которым соответствуют комплектные планы раскроя, то критерий z_1^* может быть сформулирован следующим образом:

$$z_1^* = 1 - \frac{\sum_{i=1}^m l_i w_i}{\sum_{k=1}^K \sum_{j=1}^n T_{jk} L_j W_j}. \quad (10)$$

Значение критерия (10), рассчитанное для заданного комплектного плана раскроя, показывает, какая часть раскроенного материала пойдет в отход, если такой план будет реализован.

Нормализованный критерий (6), пригодный для подстановки в целевую функцию (8), имеет следующий вид:

$$z_2^* = \frac{\sum_{k=1}^K \delta(k) - 1}{\sum_{k=1}^K \sum_{j=1}^n T_{jk} - 1}. \quad (11)$$

Существуют различные подходы к разработке эволюционных алгоритмов для решения оптимизационных задач с ограничениями. В предлагаемом эволюционном алгоритме допускается наличие в популяции особей, которым соответствуют как комплектные, так и некомплектные планы раскроя. Нарушение остальных ограничений упреждается соответствующей реализацией эволюционных операторов. В процессе работы эволюционного алгоритма может потребоваться оценка приспособленности особи, которой соответствует некомплектный план раскроя. Для решения данной проблемы был использован прием, который основывается на попарном сравнении особей и предполагает введение вспомогательной целевой функции [16].

В дальнейшем допустимой будем называть особь, которой соответствует допустимое решение, недопустимой — ту, которой соответствует недопустимое решение. Вспомогательная целевая функция используется в случае, если требуется сравнить между собой приспособленность двух недопустимых особей. Рассматривается три возможных ситуации:

1. Сравняются две допустимых особи, для оценки приспособленности обеих используется базовая целевая функция, предпочтение отдается более приспособленной особи.

2. Сравниваются допустимая и недопустимая особи, предпочтение всегда отдается допустимой особи.
3. Сравниваются две недопустимых особи, для оценки приспособленности обеих используется вспомогательная целевая функция, предпочтение отдается более приспособленной особи.

Для использования в качестве вспомогательной целевой функции дополнительно был сформулирован критерий (12), который может трактоваться как мера недоукомплектованности плана раскроя.

$$Z_{\text{infeasible}}^* = \frac{1}{m} \sum_{i=1}^m \max \left(0, \frac{l_i - l_i^{\text{prod}}}{l_i} \right) \quad (12)$$

Примененный подход к построению целевой функции упростил реализацию эволюционных операторов и позволил повысить эффективность эволюционного алгоритма в целом.

3.3 Эволюционные операторы

В соответствие с теорией Дарвина одним из факторов эволюции является наследственная изменчивость. В результате случайных мутационных изменений либо рекомбинации генов, происходящей при скрещивании, особи приобретают новые признаки и свойства, которые, в свою очередь, служат материалом эволюционного процесса. В эволюционных алгоритмах наследственная изменчивость обеспечивается с помощью специальных операторов, применяемых к особям популяции в процессе поиска решений. Перечень таких операторов варьируется в зависимости от разновидности эволюционного алгоритма, например, в эволюционных стратегиях используется только оператор мутации, тогда как в генетических алгоритмах базовым является оператор рекомбинации, мутациям отводится лишь второстепенная роль.

Предлагаемый метаэвристический метод решения задачи рационального раскроя представляет собой эволюционную стратегию, в которой для получения новых особей используется несколько различных операторов мутации. Исходя из выбранного способа представления решений в виде перечня групп G_q , сложность реализованных операторов мутации условно можно оценить по количеству групп, претерпевающих изменения в результате выполнения каждого из них. Был разработан ряд относительно простых операторов мутации, вносящих изменения только в одну из групп особи, а также один более сложный оператор мутации, предназначенный для слияния нескольких групп особи в одну.

Вначале рассмотрим операторы мутации, при выполнении которых изменяются свойства только одной из групп G_q . Предположим, что в исходной особи случайным образом выбрана некоторая группа G_r , $1 \leq r \leq Q$. Для того чтобы из исходной особи получить новую, группа G_r может быть подвергнута следующим модификациям:

1. Изменен перечень входящих в группу рулонов, $G_r \rightarrow G_r^*$;
2. Изменен способ раскроя рулонов группы, $P_r \rightarrow P_r^*$;
3. Совместно выполнены первые два преобразования: $G_r, P_r \rightarrow G_r^*, P_r^*$.

Используя каждый из трех вышеперечисленных способов получения новой особи, всего было разработано и реализовано более десяти различных операторов мутации.

Предварительное тестирование алгоритма выявило, что операторы, выполняющие преобразование $G_r \rightarrow G_r^*$, по эффективности значительно уступают операторам других двух типов, если при изменении способа раскроя рулонов применяется адаптивная генерация раскройных карт. Для дальнейшего использования в эволюционном алгоритме было отобрано четыре наиболее эффективных оператора мутации, реализация которых основывается на алгоритме 1.

Алгоритм 1 Мутация особи, изменения локализованы в пределах группы G_r .

Шаг 1. Выбрать r : $1 \leq r \leq Q$.

Шаг 2. Изменить группу рулонов $G_r \rightarrow G_r^*$.

Шаг 3. С помощью специальной процедуры для измененной группы рулонов сгенерировать новую раскройную карту с номером P_r^* .

Следует отметить, что первый и последний шаги алгоритма 1 всеми операторами мутации выполняются одинаково, отличия проявляются в характере изменений, вносимых в группу рулонов G_r на втором шаге. Способы модификации содержимого групп рулонов, использованные при реализации операторов, приведены в табл. 1, запись G_r^M обозначает результат выполнения второго шага оператора мутации с порядковым номером M .

Таблица 1

Способы изменения группы рулонов при выполнении операторов мутации

Преобразование	Описание
$G_r \rightarrow G_r^1$	Группа G_r сохраняется в исходном виде, $G_r^1 = G_r$.
$G_r \rightarrow G_r^2$	В группу G_r добавляется новый, ранее не использованный в раскрое, рулон шириной W_{new} .
$G_r \rightarrow G_r^3$	Из группы G_r удаляется случайным образом выбранный рулон с номером $J \in G_r$.
$G_r \rightarrow G_r^4$	В группе G_r случайным образом выбранный рулон с номером $J \in G_r$ заменяется другим, ранее не использованным в раскрое, рулоном шириной W_{new} .

Очевидно, что если для раскроя нескольких рулонов применяется один и тот же способ, то общая ширина соответствующей раскройной карты не должна превышать ширину наиболее узкого из раскраиваемых рулонов. Для того чтобы лучше контролировать потери рулонного материала, выбор нового рулона шириной W_{new} , вводимого в группу при выполнении второго и четвертого операторов мутации, осуществляется в соответствие со следующим правилом:

$$W(G_r) = \min_{j \in G_r} W_j,$$

$$W_{\text{new}} = \min \{W_j \mid j \in S \setminus S_{\text{used}}, W_j \geq W(G_r)\}.$$

Если найти подходящий рулон не удастся, то группа рулонов G_r сохраняется в исходном виде, $G_r^* = G_r$.

В результате выполнения второго шага алгоритма 1 мы получаем новую особь, которая состоит из групп G_q , $q \in \{1, \dots, Q\}$, $q \neq r$, перенесенных в исходном виде, и одной измененной группы G_r^* . Если $|G_r^3| = 0$, то такая группа удаляется из особи, в остальных случаях раскройная карта с номером P_r , определявшая способ раскроя

рулонов из группы G_r в исходном решении, заменяется новой. Задача, решаемая на третьем шаге алгоритма 1, заключается в поиске для измененной группы рулонов G_r^* такой раскройной карты с номером P_r^* , введение которой в план раскроя даст наиболее приспособленную новую особь. В этом, собственно, и заключается адаптивность работы операторов мутации.

Рассмотрим, каким образом задача составления раскройной карты может быть сведена к задаче о ранце [17]. Частичным решением будем называть план раскроя, состоящий из групп рулонов G_q , $q \in \{1, \dots, Q\}$, $q \neq r$. Представим, что такой план был реализован. Длина полученной заготовки шириной w_i составит

$$l_i^{\text{prod}}(G_r^*) = \sum_{k=1}^K \sum_{j \in S_{\text{used}} \setminus G_r^*} T_{jk} a_{ik} L_j.$$

Раскройная карта для оставшейся группы рулонов G_r^* должна состояться таким образом, чтобы итоговый план раскроя был комплектным, поскольку от этого зависит приспособленность новой особи. Недостающая длина заготовки шириной w_i , которую требуется получить при раскрое группы рулонов G_r^* , равняется

$$l_i^{\text{resid}}(G_r^*) = \begin{cases} l_i - l_i^{\text{prod}}(G_r^*), & l_i > l_i^{\text{prod}}(G_r^*), \\ 0, & l_i \leq l_i^{\text{prod}}(G_r^*). \end{cases}$$

С помощью записи $\lceil a \rceil$ обозначим округление числа к ближайшему целому в большую сторону. Минимальное число полос шириной w_i , которое необходимо отрезать от каждого из рулонов, входящих в группу G_r^* , для получения заготовки соответствующего наименования общей длиной не менее $l_i^{\text{resid}}(G_r^*)$ рассчитывается по формуле (13).

$$d_i(G_r^*) = \left\lceil \frac{l_i^{\text{resid}}(G_r^*)}{\sum_{j \in G_r^*} L_j} \right\rceil \quad (13)$$

Задача составления такой допустимой раскройной карты, которая обеспечит минимальные потери материала при раскрое группы рулонов G_r^* , заключается в нахождении неизвестных $x_i \in \mathbb{Z}_+$, $i \in \{1, \dots, m\}$, удовлетворяющих требованиям (14)–(16).

$$\max \sum_{i=1}^m x_i w_i, \quad (14)$$

$$\sum_{i=1}^m x_i w_i \leq W(G_r^*), \quad (15)$$

$$x_i \leq d_i(G_r^*). \quad (16)$$

Решение задачи (14)–(16) определяет способ раскроя группы рулонов G_r^* , измененной на втором шаге алгоритма 1: $a_{ik} = x_i$, $i \in \{1, \dots, m\}$, $k = P_r^*$.

Поскольку каждая итерация эволюционной стратегии предполагает получение заданного количества новых особей с помощью применения оператора мутации к особям текущей популяции, общее количество мутаций, выполняемых за один запуск эволюционного алгоритма, является значительным. Предварительное тестирование базовой конфигурации эволюционного алгоритма выявило, что реализация процедуры генерации раскройных карт на основе точного метода решения задачи о ранце приводит к существенному снижению вычислительной эффективности предлагаемого метода. Исходя из этого, вместо существующих точных методов для решения

Алгоритм 2 Генерация раскройной карты.

Шаг 1. $W := W(G_r^*)$, $b_i := d_i(G_r^*)$, $x_i := 0$, $i \in \{1, \dots, m\}$.

Шаг 2. $I = \{i \in \mathbb{Z}_+ \mid i \leq m, b_i > 0, w_i \leq W\}$,
если $I = \emptyset$, то перейти на **шаг 5**.

Шаг 3. Случайным образом выбрать $t \in I$,
 $x_i := x_i + 1$, $b_t := b_t - 1$, $W := W - w_t$.

Шаг 4. Перейти на **шаг 2**.

Шаг 5. Выдать x_i .

задачи (14)–(16) был применен рандомизированный алгоритм 2, предложенный в работе [10].

Процедура, применяемая для генерации раскройной карты, выполняет серию повторных запусков алгоритма 2, каждый из запусков осуществляется с одним и тем же набором входных данных w_i , $W(G_r^*)$, $d_i(G_r^*)$, $i \in \{1, \dots, m\}$. Количество таких повторных запусков является параметром процедуры генерации раскройной карты, в дальнейшем обозначаемым как *pgp_trials*. Лучшее из решений задачи (14)–(16), найденных с помощью алгоритма 2, используется в качестве раскройной карты с номером P_r^* , определяющей способ раскроя группы рулонов G_r^* .

Количество содержащихся в особи групп рулонов Q при выполнении одного из четырех рассмотренных операторов мутации может измениться только в двух случаях:

1. $G_r \rightarrow G_r^*$: $|G_r| = 1$, $|G_r^3| = 0$ — удаление рулона на втором шаге алгоритма 1 приводит к исчезновению группы, состоящей из одного рулона;
2. $P_r \rightarrow P_r^*$: $\exists q \in \{1, \dots, Q\}$, $P_q = P_r^* \cup q \neq r$ — сгенерированная на третьем шаге алгоритма 1 раскройная карта совпадает с одной из раскройных карт, содержащихся в частичном решении.

Следует учитывать, что отличительной особенностью большинства практических задач рационального раскроя является многообразие допустимых раскройных карт. Исходя из этого, можно предположить, что слияние двух групп при выполнении шага $P_r \rightarrow P_r^*$ будет происходить достаточно редко. Рассмотренные ранее операторы мутации реализованы таким образом, чтобы при поиске решений контролировать потери материала, однако приспособленность особей также зависит и от значения критерия (11). Для того чтобы эволюционный алгоритм был эффективным при любых значениях коэффициентов C_1 и C_2 , введем в его состав еще один оператор мутации, выполняющий слияние нескольких групп рулонов в одну. Подобное изменение в структуре особи соответствует уменьшению количества различных раскройных карт, использованных в соответствующем ей плане раскроя. Для реализации нового оператора мутации, выполняющего слияние двух групп, используем алгоритм 3.

Первая из двух объединяемых групп, G_r , выбирается случайным образом. Для выбора второй группы, G_t , используется следующее правило:

$$W(G_t) = \min (W(G_q) - W(G_r))^2,$$

$$q \in \{1, \dots, Q\}, q \neq r.$$

Таким образом, на втором шаге алгоритма 3 выбирается две близкие по ширине группы G_r и G_t (под шириной группы подразумевается ширина наиболее узкого из рулонов, входящих в ее состав). Рулоны из группы G_t переносятся в группу G_r ,

Алгоритм 3 Мутация особи, слияние двух групп.

Шаг 1. Если $Q = 1$, прекратить вычисления.

Шаг 2. Выбрать r и t : $1 \leq r \leq Q$, $1 \leq t \leq Q$, $r \neq t$.

Шаг 3. Объединить две группы рулонов $G_r^* := G_r \cup G_t$.

Шаг 4. С помощью специальной процедуры для объединенной группы рулонов сгенерировать новую раскройную карту с номером P_r^* .

Шаг 5. $Q := Q - 1$, пересчитать значения индексов групп q .

затем при помощи рассмотренной ранее процедуры осуществляется генерация новой раскройной карты для измененной группы G_r^* .

Выбор эволюционным алгоритмом одного из пяти реализованных операторов при выполнении очередной мутации осуществляется случайным образом, операторы используются с одинаковой частотой. Следует отметить, что в качестве исходной особи при выполнении операторов мутации всегда используется не оригинал, а копия особи из текущей популяции. Это позволяет при необходимости включать в следующую популяцию как исходную особь, так и ее мутировавшую копию.

3.4 Модифицированная эволюционная стратегия

В эволюционных стратегиях (ЭС) каждая итерация базового алгоритма состоит из двух шагов: вначале из μ родительских особей текущей популяции с помощью оператора мутации получают λ новых особей-потомков, затем формируется новая популяция и осуществляется переход на следующую итерацию. μ и λ являются параметрами метода. В зависимости того, участвуют ли родительские особи в формировании новой популяции или нет, различают $(\mu + \lambda)$ -ЭС и (μ, λ) -ЭС: в первом случае в отборе μ наиболее приспособленных особей, которые будут включены в новую популяцию, участвуют как родительские особи, так и их потомки, во втором — только потомки [18].

Рассматриваемый эволюционный алгоритм представляет собой модификацию $(\mu + \lambda)$ -ЭС (МЭС), в которой для формирования новой популяции вместо стандартной двухшаговой схемы «мутация-отбор» применяется более совершенный механизм, предложенный в работе [19, с. 62]. Итерация МЭС описывается алгоритмом 4.

Алгоритм 4 Генерация новой популяции в модифицированной $(\mu + \lambda)$ -ЭС.

Шаг 1. Используя попарное сравнение и заданную целевую функцию, отсортировать содержимое текущей популяции в порядке убывания приспособленности особей.

Шаг 2. Выбрать первые $(\mu - \lambda)$ особей текущей популяции, перенести их *копии* в новую популяцию без изменений.

Шаг 3. Выбрать первые λ особей текущей популяции для дальнейшего использования их в качестве родительских особей.

Шаг 4. Получить λ новых особей, однократно применив оператор мутации к каждой из λ родительских особей, выбранных на **шаге 3**.

Шаг 5. Вставить λ новых особей, полученных на **шаге 4**, в новую популяцию.

В МЭС вводится дополнительное ограничение: количество потомков не должно превышать размер популяции, $\lambda \leq \mu$. В сравнении с классической версией $(\mu + \lambda)$ -

ЭС внесенные изменения позволяют поддерживать разнообразие особей, с которыми работает эволюционный алгоритм, на более высоком уровне. При этом сохраняется возможность предупредить разрушение мутацией наиболее приспособленных особей за счет переноса их копий в новую популяцию на втором шаге алгоритма 4 при $\lambda < \mu$.

Входными данными для эволюционного алгоритма являются условие решаемой задачи ($w_i, l_i, i = 1, \dots, m; W_j, L_j, j = 1, \dots, n$), а также заданные параметры эволюционной стратегии (μ, λ), целевой функции (C_1, C_2) и процедуры генерации раскройных карт (pgp_trials). Перед выполнением первой итерации необходимо инициализировать эволюционный алгоритм, поместив в начальную популяцию μ особей, сгенерированных случайным образом либо полученных с помощью некоторого вспомогательного метода. В предлагаемом эволюционном алгоритме реализован следующий подход: начальная популяция заполняется особями, которым соответствуют приближенные решения, найденные с помощью упрощенной версии последовательной эвристической процедуры [4]. Модификация оригинального метода [4] заключается в том, что вместо задачи (1)–(7) решается упрощенная задача (1)–(4): при построении планов раскроя критерий (6) не рассматривается, для большинства найденных таким образом решений выполняется $z_2^* \approx 1$.

В качестве критерия останова в эволюционном алгоритме используется следующее простое правило: вычисления прекращаются после выполнения заданного числа полных итераций МЭС.

4 Вычислительные эксперименты

Для проведения вычислительных экспериментов рассмотренный эволюционный алгоритм был реализован на языке программирования Java. Исходный код разработанной программной библиотеки находится в открытом доступе [20].

Существует достаточно обширный перечень работ, в которых обсуждается поиск решений задачи рационального раскроя с учетом критерия (6). В тоже время значительная часть таких работ посвящена рассмотрению классических одномерных задач, в которых применяется поштучный способ учета комплектности раскроя. Исходя из этого, опубликованные тестовые данные в большинстве своем не пригодны для использования в вычислительных экспериментах.

Для исследования работы эволюционного алгоритма был составлен набор из 64 тестовых задач различной сложности (см. табл. 2) и разработан формат их представления, основанный на XML. Тестовые задачи и документация с описанием формата хранятся в том же репозитории, что и исходный код разработанной библиотеки [20].

На этапе предварительного тестирования был определен такой набор параметров, при котором эволюционный алгоритм работает стабильно, т.е. обеспечивается воспроизводимость результатов, получаемых при решении тестовых задач. Значения параметров, составляющие данный набор, следующие: $\mu = 50, \lambda = 45, pgp_trials = 5$.

Для оценки эффективности предлагаемого метода была выполнена серия вычислительных экспериментов, в которых тестовые задачи решались с помощью эволюционного алгоритма и модификации последовательной эвристической процедуры (МП-ЭП), предложенной в работе [10]. Выбор второго метода обусловлен его пригодностью для решения задачи рационального раскроя в постановке (1)–(7). Вычислительные эксперименты осуществлялись в соответствии со следующей методикой: для каждой из 64 тестовых задач выполнялось по 10 последовательных запусков тестируемого метода с одним и тем же набором параметров, в каждом из запусков фиксировались его продолжительность и лучшее из найденных решений. В случае с эволюционным алгоритмом использовался приведенный выше базовый набор параметров, продол-

жительность запуска ограничивалась временем, необходимым для выполнения 2000 итераций МЭС.

Таблица 2

Классы тестовых задач

Класс задач	Количество	m	n	Описание
optimal	10	[4 .. 12]	[10 .. 300]	Задачи с известными оптимальными решениями
production	18	[4 .. 12]	[37 .. 143]	Задачи, основанные на производственных данных
random	36	[4 .. 20]	[13 .. 106]	Задачи, составленные с использованием случайной генерации данных

При решении тестовых задач было сделано допущение, что критерии (10) и (11) имеют одинаковую важность, т.е. $C_1 = C_2 = 0.5$. Соответствующим образом были настроены и параметры метода МПЭП. Результаты вычислительных экспериментов, подвергнутые статистической обработке, представлены в табл. 3.

Таблица 3

Характеристика планов раскроя, найденных с помощью МЭС и МПЭП

Класс задач	МЭС			МПЭП		
	z_1 , %	z_1^* , %	z_2	z_1 , %	z_1^* , %	z_2
optimal	1.1	2.6	2.6	12.3	15.4	4.6
production	3.4	13.3	1.7	15.4	18.9	5.5
random	3.2	15.7	2.6	10.4	19.5	6.4

Во всех вычислительных экспериментах использовалась рабочая станция с процессором Intel Core i5-3427U под управлением операционной системы OS X 10.8. В зависимости от класса решаемых задач средняя продолжительность одного запуска эволюционного алгоритма колебалась в пределах от 1 до 4 секунд. Аналогичный показатель для последовательной эвристической процедуры составил менее одной секунды.

Сравнив характеристики полученных решений, можно прийти к выводу, что эволюционный алгоритм позволяет находить более технологичные планы раскроя (критерий z_2) с меньшим уровнем совокупных потерь материала (критерий z_1^*). Также следует отметить эффективность избранного подхода к построению целевой функции — все планы раскроя, найденные с помощью эволюционного алгоритма, являются допустимыми решениями задачи (1)–(7).

5 Заключение

Эволюционные алгоритмы широко применяются для решения задач рационального раскроя как в классической, так и в уточненной многокритериальной постановке. Характерным отличием предложенного эволюционного метода решения задачи рационального планирования продольных раскроев рулонного материала является

использование составного оператора мутации, позволяющего в процессе поиска изменять решения несколькими различными способами. Реализованный подобным образом эволюционный алгоритм обладает схожими чертами с другим метаэвристическим методом — поиском с чередующимися окрестностями [21]: фактически, каждый из пяти операторов мутации неявным образом задает собственное правило построения окрестности текущего решения, а мутация особи представляет собой один шаг локального поиска в соответствующей окрестности. Эффективность эволюционного алгоритма, использующего составной оператор мутации, подтверждается результатами вычислительных экспериментов.

Дальнейшие исследования могут быть направлены на разработку бинарных эволюционных операторов, а также реализацию механизма автоматической подстройки параметров эволюционного алгоритма. Кроме того, предложенный метод может быть адаптирован для решения многокритериальных одномерных задач рационального раскроя с поштучным способом учета комплектности раскроя.

Список литературы

1. Канторович Л. В. Математические методы в организации и планировании производства \ Л. В. Канторович. — Л. : Изд-во ЛГУ, 1939. — 68 с.
2. Wascher G. An improved typology of cutting and packing problems \ G. Wascher, H. Haussner, H. Schumann \ \ European Journal of Operational Research. — 2007. — Vol. 183, N 3. — P. 1109–1130.
3. Valerio de Carvalho J. M. LP models for bin packing and cutting stock problems \ J. M. Valerio de Carvalho \ \ European Journal of Operational Research. — 2002. — Vol. 141, N 2. — P. 253–273.
4. Haessler R. W. Selection and design of heuristic procedures for solving roll trim problems \ R. W. Haessler \ \ Management Science. — 1988. — Vol. 34, N 12. — P. 1460–1471.
5. Foerster H. Pattern reduction in one-dimensional cutting stock problems \ H. Foerster, G. Wascher \ \ International Journal of Production Research. — 2000. — Vol. 38, N 7. — P. 1657–1676.
6. Belov G. Setup and open-stacks minimization in one-dimensional stock cutting \ G. Belov, G. Scheithauer \ \ INFORMS Journal on Computing. — 2007. — Vol. 19, N 1. — P. 27–35.
7. Haessler R. W. Cutting stock problems and solution procedures \ R. W. Haessler, P. E. Sweeney \ \ European Journal of Operational Research. — 1991. — Vol. 54, N 2. — P. 141–150.
8. Головистиков А. В. Задачи двумерной прямоугольной упаковки и раскроя : Обзор \ А. В. Головистиков \ \ Информатика. — 2008. — № 4. — С. 18–33.
9. Goulimis C. Optimal solutions for the cutting stock problem \ C. Goulimis \ \ European Journal of Operational Research. — 1990. — Vol. 44, N 2. — P. 197–208.
10. Vahrenkamp R. Random search in the one-dimensional cutting stock problem \ R. Vahrenkamp \ \ European Journal of Operational Research. — 1996. — Vol. 95, N 1. — P. 191–200.

11. An iterative sequential heuristic procedure to a real-life 1.5-dimensional cutting stock problem \ X. Song, C. B. Chu, Y. Y. Nie, J. A. Bennell \ \ European Journal of Operational Research. — 2006. — Vol. 175, N 3. — P. 1870–1889.
12. Golfeto R. R. A genetic symbiotic algorithm applied to the cutting stock problem with multiple objectives \ R. R. Golfeto, A. C. Moretti \ \ Advanced Modeling and Optimization. — 2009. — Vol. 11, N 4. — P. 473–501.
13. Research on improvement strategies and parameter analysis of ant colony algorithm for one-dimensional cutting stock problem \ L. Huang, H. Chen, B. Yang, C.-G. Zhou \ \ Advances in Computational Intelligence \ ed. by W. Yu, E. N. Sanchez. — Berlin [etc.] : Springer-Verlag, Inc., 2009. — P. 319–329.
14. Schilling G. An algorithm for the determination of optimal cutting patterns \ G. Schilling, M. C. Georgiadis \ \ Computers & Operations Research. — 2002. — Vol. 29, N 8. — P. 1041–1058.
15. Скобцов Ю. А. Основы эволюционных вычислений : Учебное пособие \ Ю. А. Скобцов. — Донецк : ДонНТУ, 2008. — 326 с.
16. Deb K. An efficient constraint handling method for genetic algorithms \ K. Deb \ \ Computer Methods in Applied Mechanics and Engineering. — 2000. — Vol. 186, N 2–4. — P. 311–338.
17. Martello S. Knapsack problems : algorithms and computer implementations \ S. Martello, P. Toth. — Chichester [etc.] : John Wiley & Sons Ltd., 1990. — 318 p.
18. Schwefel H.-P. Evolution and Optimum Seeking \ H.-P. Schwefel. — New York : Wiley-Interscience, 1995. — 456 p.
19. Michalewicz Z. Genetic algorithms + data structures = evolution programs \ Z. Michalewicz. — 3rd ed. — Berlin [etc.] : Springer-Verlag, 1998. — 387 p.
20. GitHub : EA-based solver for 1.5D MSSCSP [Electronic resource]. — 2013. — [Cited 2013, 1 June]. — Available from : <https://github.com/akavrt/csp>.
21. Mladenovic N. Variable neighborhood search \ N. Mladenovic, P. Hansen \ \ Computers & Operations Research. — 1997. — Vol. 24, N 11. — P. 1097–1100.