

# РЕШЕНИЕ ЗАДАЧИ РАСКРОЯ РУЛОННОГО ПОЛУФАБРИКАТА ДЛЯ НЕСКОЛЬКИХ АГРЕГАТОВ ПРОДОЛЬНОЙ РЕЗКИ С УЧЕТОМ ПОТРЕБЛЕНИЯ НА СЛЕДУЮЩЕМ ЭТАПЕ ПРОИЗВОДСТВА ДЛЯ ЦЕХА ГНУТЫХ ПРОФИЛЕЙ

**SOLUTION OF THE PROBLEM  
OF CUTTING A ROLL  
SEMI-FINISHED PRODUCTS FOR SEVERAL  
SLITTING UNITS, ACCORDING TO  
THE CONSUMPTION AT THE NEXT  
STAGE OF PRODUCTION FOR THE SHOP  
OF BENT PROFILES**

**O. Kuznetcova  
K. Harahnin**

*Summary.* This article is devoted to solving the problem of finding the optimal set of coil cuts for several cutting units at a metallurgical enterprise based on a mathematical model; a random search algorithm using the simplex method is used to solve the problem. The complexity of the problem increases with the number of input elements and belongs to the class of NP-hard problems. Criteria for evaluating the effectiveness of planning solutions are given, restrictions on the search problem are formulated, and a search algorithm is developed.

*Keywords:* search algorithm, cutting unit, metallurgical enterprise, coil cutting.

**Кузнецова Оксана Юрьевна**

Соискатель, Череповецкий государственный университет  
oukuznetcova@severstal.com

**Харахнин Константин Аркадьевич**

К.т.н., доцент, Череповецкий государственный университет  
harahninka@chsu.ru

*Аннотация.* Данная статья посвящена решению задачи поиска оптимального набора раскроев рулонов для нескольких агрегатов резки на металлургическом предприятии на основании математической модели, для решения задачи применяется алгоритм случайного поиска с применением симплекс-метода. Сложность задачи возрастает с увеличением количества входных элементов и относится к классу NP-сложных задач. Приводятся критерии оценки эффективности решения планирования, сформулированы ограничения задачи поиска, а также разработан алгоритм поиска.

*Ключевые слова:* алгоритм поиска, агрегат резки, металлургическое предприятие, раскрой рулонов.

## Введение

**Н**а сегодняшний день металлургические предприятия работают в условиях жесткой конкуренции, что в свою очередь приводит к высоким требованиям не только к качеству выпускаемой продукции, но и снижению внутренних издержек при строгом соблюдении сроков поставок по всем заказам предприятия. Эти требования обуславливают постоянное совершенствование систем планирования предприятия на всех уровнях, в том числе и системы формирования детальных расписаний, в которых формируются графики, позволяющие выполнить обязательства предприятия перед потребителями с учетом внутренних издержек. Перед планировщиком цеха гнутых профилей стоит задача своевременно обеспечить поступление входной заготовки — штрипса — для производства гнутых профилей в условиях малого наличия на складах

перед агрегатами резки и при этом минимизировать обрезь при порезке рулона в штрипс.

Управление ресурсами предприятия осуществляется системами различных уровней, за производственное планирование отвечают ERP (Enterprise Resource Planning) и APS (Advanced Planning and Scheduling) системы (рисунок 1). При этом ERP системы работают на более высоком уровне, формируя из всего списка заказов (портфеля заказов) объемный график производства на заданный период — блочный план. Блочный план определяем рекомендуемые диапазоны дат для производства определенного вида трубной продукции. Помимо блочного плана для построения точной очередности заданий на порезку и прокатку нужно понимать обеспечен ли каждый блок производства входной заготовкой, если заготовки недостаточно, то блочный план не будет выполнен. Для этого пла-

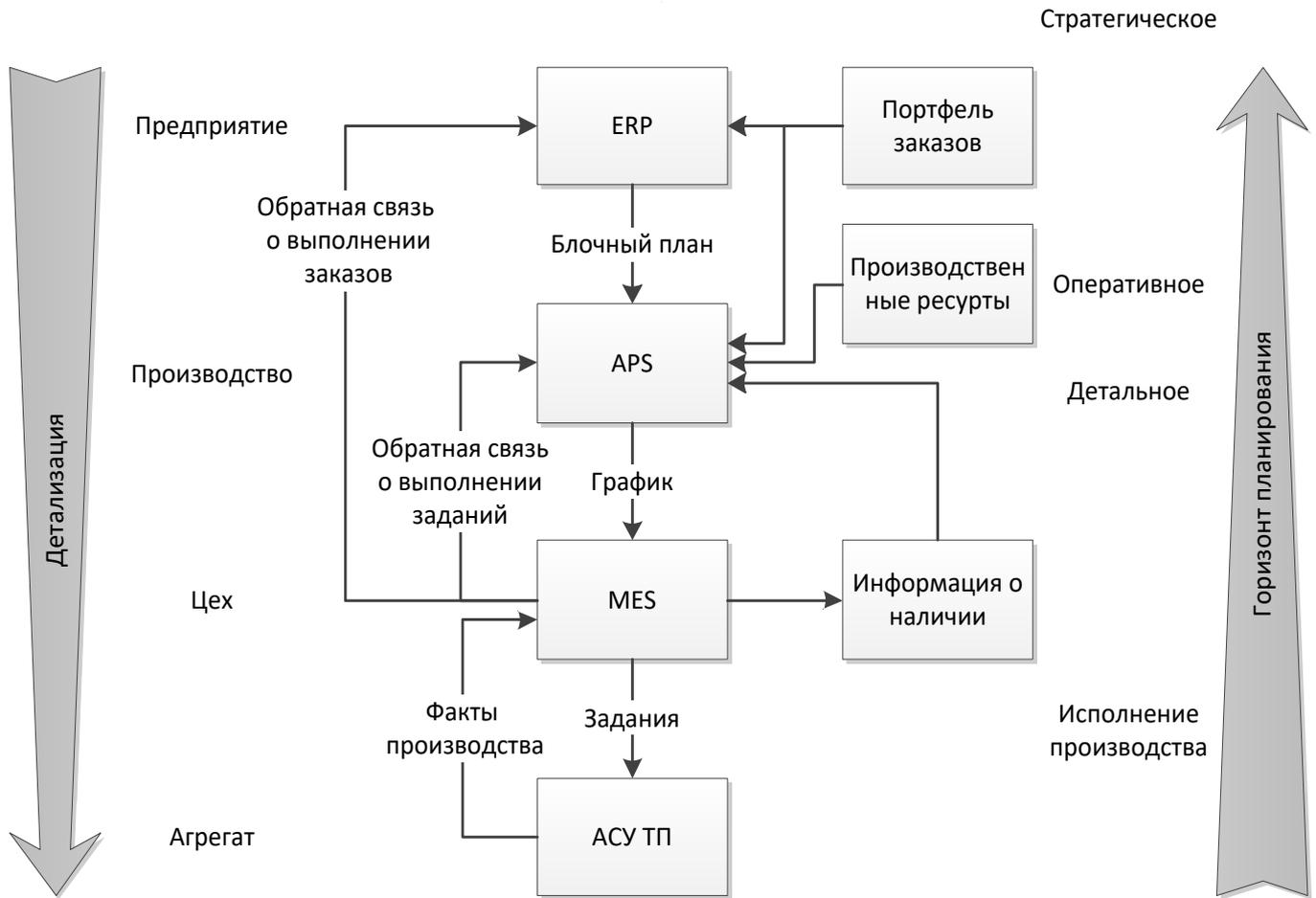


Рис. 1. Схема взаимодействия уровней планирования

норовщику нужно анализировать текущее наличием рулонов и заготовок — входных элементов для производства трубной продукции. Данные о наличии рулонов и заготовки на складах поступают планировщику из MES (Manufacturing execution system) системы в APS. На основании этих данных детальный планировщик анализирует фактическое наличие рулонов и заготовок на складе, а также текущий список заданий на каждом переделе, которым управляет. Для каждого задания в система рассчитан объем заготовки, которого не хватает для выполнения этого задания. Этот объем планировщик должен получить в результате порезки рулонов на штрипсы с параметрами, подходящими под это задание. Таким образом, известно какие рулоны есть на каждом участке, и какой объем штрипсовой заготовки требуется для обеспечения выполнения графика производства трубы. На основании этих данных планировщик формирует набор раскроев, обеспечивающих получение штрипсовой заготовки из рулонов на складе с учетом всех технологических ограничений. Последовательность раскроев определяет график порезки, который передается в производство в MES и АСУ ТП.

Задача детального планировщика состоит не только в том, чтобы сформировать исполнимое расписание с учетом всех ограничений, но и минимизировать потери производства на этапе выполнения заказов. Таким образом, при формировании расписания порезки планировщик не только решает задачу поиска оптимального раскроя, но и стремится сформировать расписание с учетом сроков поставки по выполняемым заказам.

#### Актуальность проблемы

Задаче поиска оптимального раскроя посвящено большое количество исследований различных ученых: Кантарович Л.В., Залгаллер В.А.[1], Гомори Р.[2,3], Гонгорра Р. В этих работах задача предложены методы решения задач раскроя с различными постановками и спецификой. Часть методов относится к методам линейного программирования. Но на практике не все задачи поиска раскроя являются линейными. Описанная производственная задача осложняется наличием нелинейных целей, что не позволяет решать ее напрямую методами линейного программирования. Описанная производ-

ственная задача относится к классу многомашинных задач с двухстадийным производством, т.к. сначала выполняется порезка рулона в штрипс, а затем из штрипса производится труба. Порезка может выполняться на различных агрегатах. Кроме того, если рассматривать график порезки с точки зрения обеспечения полуфабрикатом последующих операций производства трубного сортамента, то возникают дополнительные цели по обеспечению выполнения графика прокатки. Оптимальный план, с точки зрения раскроя, не всегда соответствует потребности в заготовке, полученной при порезке, на следующем переделе. Таким образом, требуется синхронизация графика потребления заготовки и раскроя с учетом этого потребления. С учетом задачи по минимизации наличия на складах и ограниченности ресурсов планировщик должен иметь возможность формировать график из текущего наличия рулонной заготовки на складах и потребности в штрипсе в первую очередь для ближайших заданий прокатного отделения. При наличии нескольких параллельно работающих агрегатов резки, обеспечивающих несколько параллельных агрегатов по производству труб, находящихся на разных участках, задача усложняется формированием загрузки с обеспечением эффективной логистики перемещений между агрегатами внутри разных участков. В совокупности задача является трудоемкой для ручного решения и требует соответствующей системы поддержки, позволяющей осуществлять поиск решения с учетом всех ограничений и поставленных целей.

### Постановка задачи

Задача составления расписания порезки является многопараметрической задачей оптимизации, учитывающей размер кампании порезки, обрезь при порезке, перемещения между агрегатами, а также запаздывание поступления полуфабриката после порезки для следующего передела. При формировании расписания необходимо учитывать технические параметры агрегатов: максимальный вес входного рулона, максимальная ширина входного рулона, количество полос в схеме раскроя максимальное и допустимое, а также ограничения на выходе: максимальный вес штрипса на выходе, максимальная ширина штрипса для агрегата.

Разработаем математическую модель задачи поиска оптимального набора раскроев. Пусть на складах имеются рулоны с различными характеристиками: марка, толщина, ширина, вес рулона. Рулоны с одинаковыми характеристиками объединим в партии P. При порезке рулонов и дальнейшем производстве трубы последовательность обработки рулонов одной партии не важна, т.к. рулоны одной партии обладают одинаковыми характеристиками, то нет разницы какой рулон

из партии будет порезан первым, а какой последним. Это позволит снизить вариативность задачи. Имеется множество заказов на трубу Z, каждый заказ потребляет заготовку с определенными характеристиками. На каждом заказе определена дата операции производства трубы — потребления штрипсов заготовки. Планировщику необходимо сформировать на основании заказов Z и наличия рулонов P такой набор раскроев  $X^*$ , при котором достигается максимум целевой функции F при выполнении всех ограничений задачи.

Введем следующие обозначения:

$z_i$  —  $i$ -тый заказ.

$z_i.Width$  — ширина заготовки для  $i$ -того заказа.

$z_i.Quantity$  — объем для  $i$ -того заказа.

$z_i.KDatePlanned$  — дата производства для  $i$ -того заказа.

$p_j$  —  $j$ -тая партия заготовки.

$p_j.Width$  — ширина рулонов  $j$ -той партии рулонов.

$p_j.Quantity$  — объем  $j$ -той партии.

$p_j.Nr$  — количество рулонов  $j$ -той партии.

$x_k$  —  $k$ -тый раскрой.

$x_k.Width$  — ширина  $k$ -того раскроя.

$x_k.Nr$  — количество рулонов  $k$ -того раскроя.

$MinQuantity$  — минимальный объем раскроя.

$l$  — количество раскроев.

$c_t$  — вес критерия оценки.

$F_{iter}$  — значение целевой функции на итерации  $iter$

$$f_1 = \sum_{k=1, j=1}^{l, m} p_{jk}.Width - x_k.Width \rightarrow \min \quad (1.1)$$

$$f_2 = \sum_{k=1, i=1}^{l, n} z_i.Quantity - x_{ik}.Quantity \rightarrow \min \quad (1.2)$$

$$f_3 = \sum_{k=1}^l x_k.Quantity / l \rightarrow \max \quad (1.3)$$

$$f_4 = \frac{\sum_{k=1, i=1}^{l, n} (z_i.Quantity - x_{ik}.Quantity) * z_i.KDatePlanned}{z_i.Quantity} \rightarrow \min \quad (1.4)$$

Формулы (1.1–1.4) являются ключевыми показателями эффективности (КПЭ) итогового решения  $f_1, f_2, f_3, f_4$ .  $t$  — количество ключевых показателей.

$$F = \sum_{t=1}^4 c_t * f_t \rightarrow \max \quad (2)$$

Целевая функция (2) — оценка итогового плана с учетом весов КПЭ, систему ограничений описывают формулы (3).

$$p_j.Width - x_{jk}.Width \geq 0 \quad (3.1)$$

$$z_i.Quantity - x_{ik}.Quantity \geq 0 \quad (3.2)$$

$$p_j.Nr - x_{jk}.Nr \geq 0 \quad (3.3)$$

$$x_k.Quantity \geq MinQuantity \quad (3.4)$$

## Теоретическая часть

Целевая функция  $F$  не является полностью линейной, при этом сформулированная задача может быть сведена к задаче линейного программирования. Для этого вместо  $l$  — количества раскроев, параметра зависящего от решения введем константу  $l_{const}$  которая не будет зависеть от решения задачи. В качестве значения  $l_{const}$  используем среднее количество формируемых раскроев на группе оптимизации. На каждом шаге будем корректировать это значение будем брать значение количества раскроев текущее  $l_{current}$  и увеличивать его на  $l_{const}$  таким образом  $l = l_{current} + l_{const}$ .

В общем виде задачи линейного программирования имеет детерминированный алгоритм решения, линейное программирование позволяет снизить размерность задачи. Т.е. алгоритмы поиска являются достаточно быстрыми.

Анализ входных данных показывает, что в данном случае количество вариантов не позволяет применять детерминированные методы на всем наборе входных данных. Кроме того, решение задачи стандартным методом при наличии нелинейной цели может привести к тому, что найденное решение не будет соответствовать поставленной задаче. Эти особенности требуют дополнительно проверки решения на соответствие поставленной задаче.

В последнее время особой популярностью пользуются генетически алгоритмы поиска оптимальных решений. Особенностью генетических алгоритмов является использование вероятностных правил выбора элементов, поиск ведется из различных начальных точек. Применим эти подходы для решения поставленной задачи

Для решения задачи будем использовать комбинированный метод случайного поиска и симплекс метода, как наиболее распространенного метода решения задачи линейного программирования. Будем выбирать набор партий и заказов с учетом вероятности и на выбранном наборе будет выполняться поиск с помощью симплекс-метода. После чего будет производиться оценка решения на всем наборе данных. Если она лучше предыдущей оценки, то решение будет применено. При этом в любом случае каждая партия из набора шага получит оценку эффективности и увеличит счетчик использования партии при поиске. Это позволит чаще выбирать те партии, которые ранее не использовались, а также учитывать их эффективность при следующем выборе набора для поиска решения. Не следует забывать, что выбор только часто выбираемых в решение элементов, может привести к ограниченности реше-

ния, поэтому, как и в генетических алгоритмах, будем добавлять случайные элементы.

Поиск будет завершён при выборе всех партий на одном шаге либо при достижении условия выхода из алгоритма.

## Алгоритм решения

Для решения задачи необходимо разбить партии  $P$  и заказы  $Z$  на непересекающиеся подмножества  $G$ , это позволит разбить задачу на группы оптимизации и искать решение на каждом подмножестве — группе оптимизации. Группы не могут пересекаться друг с другом в части элементов комбинирования — партий или заказов. Т.е. заказ  $Z$  из подгруппы  $G$  не может быть скомбинирован с заказами другой группы и на рулонах другой группы. Возьмем разбиение по марке стали. Если марка стали партии не соответствует параметрам заказа, то они не могут быть скомбинированы в одном раскрое.

Т.к. даже одна группа оптимизации может иметь большое количество элементов, то будем искать решение не на всей группе, а только на ее части — подгруппе оптимизации. На каждой подгруппе будем находить набор раскроев — решение задачи комбинирования заказов на рулоне с учетом всех ограничений. Подгруппы оптимизации пересекаются между собой с целью обеспечения наиболее равномерного использования входных элементов за счет вероятностного подхода в выборе элементов подзадачи.

Таким образом, входными данными для подзадачи будут являться часть рулонов и заказов группы оптимизации, обрабатываемой на текущей итерации по группам.

Алгоритм решения подзадачи приведен на рисунке 3.

Группы последовательно перебираются до достижения остановки алгоритма. Внутри группы оптимизации  $g$  выбирается опорный элемент — якорь. Опорный элемент будем выбирать с учетом вероятности из двух возможных вариантов. Сначала выбирается цель текущей итерации: улучшить решение и уменьшить штраф или поиск на случайном наборе. В зависимости от этого будем выбирать опорный элемент. В первом случае якорем будет элемент, для которого возможен максимальный прирост итогового КПЭ. Таким образом, в первом случае для улучшения решения будем выбирать заказ, дающий максимальное улучшение, а для уменьшения штрафа в качестве якоря выберем схему раскроя с максимальным суммарным штрафом за недостающий объем кампании. Для второго варианта якорем

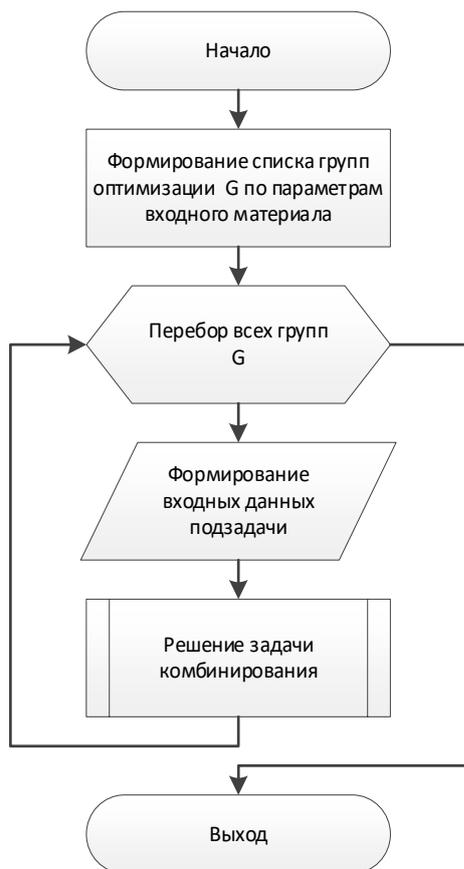


Рис. 2. Алгоритм обработки подзадач

является случайный элемент. Для опорной точки выберем окружение — текущее решение и добавим дополнительные элементы с учетом типа текущей итерации и вероятностных оценок элементов.

На сформированном наборе элементов выполним поиск с помощью симплекс метода и проверим решение на всем наборе. Если решение на всем наборе показывает улучшение целевой функции по сравнению со значением целевой функции после предыдущей итерации  $F_{iter} > F_{iter-1}$ , то решение будет применено. Счетчики элементов  $p_j.Used$  и  $p_j.Selected$ , выбранных для поиска, будут увеличены на 1 для каждой партии выбранного набора элементов. Если на полном наборе решение не улучшилось, т.е.  $F_{iter} \leq F_{iter-1}$ , то пометим партии, выбранные в решение увеличив счетчик  $p_j.Used$  на 1 для каждой партии. В следующий раз элементы с большим  $p_j.Selected$  будут выбраны с большей вероятностью. Нужно отметить, что увеличение вероятности не означает выбор именно этих партий. Для реализации выбора с учетом вероятности будем использовать отбор, в основе которого лежит колесо рулетки [4]. Этот метод отбора имеет случайную составляющую, что позволит не скатиться в локальный оптимум.

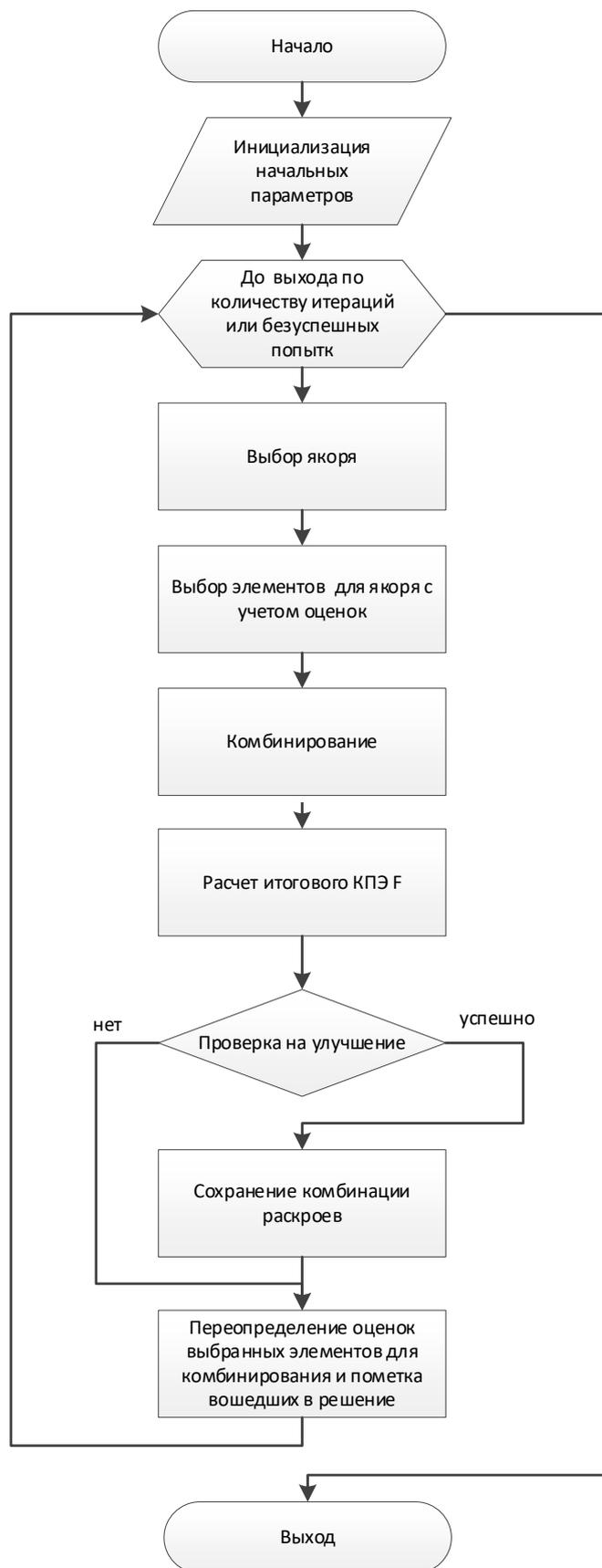


Рис. 3. Алгоритм решения подзадачи

В случае с итерацией со случайными элементами будут выбраны элементы, которые использовались реже всего, т.е.  $p_j$ . Used. Это позволит последовательно использовать различные элементы одной подгруппы.

После перебора всех групп оптимизации получаем набор схем порезки, связанных с заказами и рулонами. Схемы порезки  $X$  сформированы с учетом минимизации ожидания. Из полученных раскроев формируется последовательность порезки, последовательно выбирая раскрои с ближайшими к выполнению заказами.

## Выводы

Перед планировщиком стоит сложная многокритериальная задача оптимизации, которую приходится решать ежедневно в условиях быстроменяющейся ситуации с наличием входных рулонов и заказов, а также изменения доступности оборудования. Большая размерность задачи приводит к ошибкам и снижению показателей эффективности. Для решения поставленной задачи предложен алгоритм, который анализирует весь набор входных данных, и с учетом критериев оптимальности и ограничений строит исполнимый набор схем порезки. Предложенный комбинированный ме-

тод поиска оптимального решения позволяет находить решение на большом наборе данных за счет итераций на небольших подзадачах с возвратом к предыдущему решению при отсутствии улучшения по итоговому КПЭ. Разница в формулах расчета, вызванная необходимостью линеаризации, между КПЭ основной модели данных и оптимизатора может оказывать влияние на решение, но это влияние снижается также за счет проверки решения в основной модели на большом количестве итераций. Каждая итерация выполняется быстро за счет применения детерминированного алгоритма. Чтобы не попасть в локальный оптимум используется случайный подход при выборе элементов для комбинирования. Использование результатов предыдущих итераций позволяет сделать поиск более быстрым. Проверка корректности решения проводилась на тестовых наборах данных — бенчмарках. Бенчмарки созданы планировщиками и содержат в себе максимально эффективное решение, которое смогли найти планировщики вручную для проверки корректности работы оптимизатора. Решение с учетом весов заложенных для каждого КПЭ, полученное с применением комбинированного алгоритма, во всех случаях не хуже решения пользователя, а в части бенчмарков лучше решения пользователя. Все наложенные ограничения также соблюдаются.

## ЛИТЕРАТУРА

1. Канторович Л.В., Залгаллер В.А. Рациональный раскрой промышленных материалов. — Новосибирск: Наука, 1971.
2. P.C. Gilmore, R.E. Gomory. A linear programming approach to the cutting-stock problem // Operations Research. — 1961. — № 9. — С. 849–859.
3. P.C. Gilmore, R.E. Gomory. A linear programming approach to the cutting-stock problem — Part II // Operations Research. — 1963. — № 11. — С. 863–888.
4. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. — М.: Горячая линия Телеком, 2006. 452 с.

© Кузнецова Оксана Юрьевна (oukuznetcova@severstal.com), Харахнин Константин Аркадьевич (harahninka@chsu.ru).

Журнал «Современная наука: актуальные проблемы теории и практики»