

УДК 004:666

А. Е. ТАРАСОВ

РЕШЕНИЕ ЗАДАЧИ ОДНОМЕРНОГО РАСКРОЯ МАТЕРИАЛА РАЗЛИЧНЫХ ДЛИН НА БАЗЕ ГИБРИДИЗАЦИИ ЭВОЛЮЦИОННЫХ АЛГОРИТМОВ

Рассматривается задача одномерного раскроя, часто возникающая в промышленности, в том числе в строительной индустрии при производстве окон. Для ее решения предложена гибридизация двух эволюционных алгоритмов: метода последовательного уточнения оценок (SVC) и одноточечного эволюционного алгоритма (1+1) — EA. Первый конструирует раскрой для стационарного списка подачи материала различной длины; второй находит последовательность раскроя материала при фиксированном порядке подачи деталей. *Строительная индустрия; одномерный раскрой материала; эволюционный алгоритм; барьерная политика*

ВВЕДЕНИЕ

Экономное использование материальных ресурсов является одним из приоритетных направлений современных исследований. Анализ технологических процессов в различных отраслях промышленности показывает, что основные этапы заготовительного производства связаны с раскроем материала. Моделирование размещения деталей на картах раскроя является важным этапом технологического процесса для экономии ресурсов. Расчет рационального размещения деталей с последующим раскроем материала позволяет существенно сократить его расход. Однако существующая практика раскроя даже с применением систем оптимизации плохо справляется с проблемой ресурсосбережения. Возникающие отходы до 20 % покрываются у потребителя за счет завышенной цены. Создание технологий, основанных на оптимизационных методах, является одной из важных задач ресурсосбережения. Объединить обе проблемы — оптимизации и промышленной реализации — одна из важных и трудных задач науки и промышленности.

Задача одномерного раскроя одного типа материала хорошо изучена. Она описывается моделью линейного целочисленного программирования [1]. Для ее решения в условиях массового производства применяется релаксация (упрощение) линейным программированием. При этом используется один из методов линейного программирования с неявно заданной информацией о столбцах матрицы ограничений (карт раскроя). В такой постанов-

ке задача линейного раскроя рассматривалась Л. В. Канторовичем, В. А. Залгаллером в работах [2], [3] в 1951 и 1971 гг. и Р. Gilmore, R. Gomory [4] в 1961 г. Эффективный алгоритм для раскроя материала в серийном производстве был предложен Г. Беловым для рассматриваемого случая раскроя материала смешанных длин [5]. Однако он в некоторых случаях оказывается слишком трудоемким и его практическое использование затруднено. Параллельно развиваются эвристические методы. Наиболее продвинутыми можно считать генетические методы для решения задач раскроя материала одинаковой длины, разработанные E. Falkenauer [6].

Созданию эффективных, проблемно-ориентированных методов раскроя материала смешанных длин остается открытой проблемой, решению которой и посвящена статья.

1. ЗАДАЧА ОПТИМИЗАЦИИ ЕДИНИЧНОГО РАСКРОЯ МАТЕРИАЛА РАЗЛИЧНОЙ ДЛИНЫ

Постановка задачи. Рассматривается производственная ситуация, возникающая в условиях единичного (мелкосерийного) производства. Материал для раскроя на детали заданных размеров поступает в виде хлыстов (стержней) различных типоразмеров в заданном количестве. Требуется составить план раскроя, минимизирующий расход материала, состоящий из различных карт раскроя с указанием интенсивности их применения.

Задача может быть задана следующим информационным вектором:

$$\begin{aligned} (m, M, \lambda = (l_1, l_2, \dots, l_m + M), \\ \beta = (b_1, b_2, \dots, b_{m+M})), \end{aligned}$$

где m — число различных деталей; M — число типов материала; l_1, \dots, l_m — длины деталей, l_{m+1}, \dots, l_{m+M} — длины прутков каждого типа; b_1, \dots, b_m — количество деталей каждого типа; b_{m+1}, \dots, b_{m+M} — количество имеющихся в наличии прутков каждого типа. При этом все величины l_i и b_i — целые.

Обозначим $n := m + M$ — размерность задачи, тогда $P = \sum_{i=m+1}^n l_i$ — суммарная длина (вес) одного комплекта материала.

Вектор $\alpha^j = (a_{1j}, \dots, a_{nj})^T \in Z_+^n$ описывает карту раскроя j (Cutting Pattern, CP) с компонентами a_{ij} , указывающими число деталей типа i для $1 \leq i \leq m$ и тип раскраиваемого материала для $m+1 \leq i \leq n$; $a_{ij} = 1$, если раскраивается материал типа i , в противном случае $a_{ij} = 0$.

Карта раскроя α^j называется допустимой, если

$$\sum_{i=1}^m a_{ij} l_i \leq \sum_{i=m+1}^n a_{ij} l_i, \quad \sum_{i=m+1}^n a_{ij} = 1 \quad (1)$$

Наряду с векторами $\alpha = (a_1, a_2, \dots, a_n)$ и $\beta = (b_1, b_2, \dots, b_n)$ введем m -мерные подвекторы $a = (a_1, \dots, a_m)^T$ и $b = (b_1, \dots, b_m)^T$, характеризующие составляющую деталей для векторов α и β . Введем понятие собственно допустимого раскроя, заданного вектором α , удовлетворяющим (1) и условию

$$a \leq b. \quad (2)$$

В задаче раскроя с прутками различных длин (Multi Cutting Stock Problem, MCSP) искомыми является набор из η собственных карт раскроя, соответствующий им вектор $x = (x_1, x_2, \dots, x_\eta)$ с компонентами x_j — интенсивностями применения раскроя α^j . При этом должны быть получены все детали полностью с минимальными затратами материала. Обозначим l_j — длина прутка, раскраиваемого по карте α^j .

Задача MCSP. При заданных исходных данных требуется минимизировать функцию

$$\mu(x) = \sum_{j=1}^{\eta} l_j x_j \quad (3)$$

на множестве векторов

$$x = (x_1, x_2, \dots, x_\eta), \quad (4)$$

удовлетворяющих условиям:

$$x \in Z_+^\eta; \quad (5)$$

$$\sum_{j=1}^{\eta} a_{ij} x_j \geq b_i, \quad i = \overline{1, m}; \quad (6)$$

$$\sum_{j=1}^{\eta} a_{ij} x_j \leq b_i, \quad i = \overline{m+1, n}. \quad (7)$$

Формально задача может быть решена одним из методов целочисленного программирования с применением релаксации линейным программированием. Но такой подход, как уже указывалось, является трудоемким.

2. ГИБРИДНЫЙ МЕТОД ЭВОЛЮЦИОННОЙ МЕТАЭВРИСТИКИ ДЛЯ РЕШЕНИЯ ЗАДАЧИ MCS

Эволюционный алгоритм (Evolution Algorithm, EA) моделирует развитие популяций особей, представляющих решения задачи оптимизации. Показателем оценки особи i служит пригодность g_i , т. е. целесообразность ее участия в развитии популяции. Ключевыми моментами EA являются механизмы отбора особей с хорошей пригодностью. Рассмотрим общую схему EA. Популяция размерности s определяется как вектор $\Pi = (p_1, p_2, \dots, p_s)$, компоненты p_i , $i = \overline{1, s}$ которого являются особями. Предположим, что имеется некоторый алгоритм Init для построения начальной популяции $\Pi^{(0)}$. К ее преобразованию на каждой итерации применяют операторы T_c — селекции, T_p — репродукции и T_b — выживания. Реализуются операторы в зависимости от прикладного содержания эволюционного процесса.

Общая схема EA.

0. Применить Init для построения $\Pi^{(0)}$, положить: итерация $t := 0$.

1. Пока не выполнено условие останова:

1.1. $\Pi' := T_c(\Pi^{(t)})$;

1.2. $\Pi'' := T_p(\Pi')$;

1.3. $\Pi' := T_b(\Pi^t, \Pi'')$.

2. Положить $t := t + 1$.

3. Конец: лучшее из найденных решений.

Простейшим из эволюционных алгоритмов является $(1+1)$ -EA. Суть его в том, что популяция состоит из одной особи, которая производит одного потомка. Поиск лучшего

решения осуществляется на всем множестве допустимых решений.

Алгоритм (1+1)-EA.

0. Применить Init для построения начального решения: π^0 (начальный список подачи деталей на раскрой), LC^0 (линейный раскрой), рекорд $R = L^0$;

1. Для $t := 0$ до t_{\max} выполнить:

1.1. $LC^{(t)} = M(LC^{(t-1)})$; $\pi^t = M(\pi^{t-1})$ (мутация M состоит в применении декодера к списку π^t);

1.2. Рекорд R^t ;

2. Конец: лучшее из найденных решений.

Представляет интерес гибридизация алгоритмов, реализуемая по схеме вложения. При этом в качестве основной (внешней схемы) примем (1+1) – EA, а внутренней – метод последовательного уточнения оценок (Sequential Value Correction, SVC) [7]. Алгоритмы работают с разными приоритетными списками π : (1+1) – EA с π^l , в котором упорядочены прутки разной длины для подачи их на раскрой; Sub работает с π^d , в нем упорядочены детали для их подачи на раскрой. Во внутренней схеме работа алгоритма SVC начинается с инициализированных списков π^d и π^l . При этом список π^l – фиксированный, а π^d подвергается эволюции с помощью алгоритма SVC. В результате его работы отыскивается список π^d , такой, что раскрой по установленному в нем порядку оказался лучшим. Затем переходят к внешней схеме, фиксируя π^d , с помощью (1+1) – EA находят π^l , для которого раскрой оказался лучшим при фиксированном π^d . В этом состоит одна итерация гибридного эволюционного процесса.

Опишем алгоритм SVC.

В SVC оптимизация осуществляется как при генерировании карты раскроя, так и при решении задачи в целом. С этой целью предлагается использовать аппарат «псевдо-оценок», сходный с применяемыми в линейном программировании объективно-обусловленными оценками Л. В. Канторовича [1]. Последние совпадают с двойственными переменными, отвечающими оптимальному решению прямой и двойственной задач линейного программирования. Известно, что объективно-обусловленные оценки имеют смысл подетальных норм расхода материала, отвечающие оптимальному плану раскроя. В целочисленном случае в качестве псевдо-оценок принимаются подетальные нормы расхода материала, рассчитанные на основе некоторого раскройного плана. При этом начальные подетальные нормы определяются после по-

строения начального плана раскроя, составленного с помощью того или иного алгоритма конструирования линейного раскроя для фиксированного списка π^l подачи материала. Затем с помощью полученных оценок строится новый план, и оценки корректируются. Таким образом, происходит уточнение оценок заготовок по мере получения очередного допустимого плана раскроя (после решения каждой задачи генерации раскроя). Формулы, по которым рассчитываются начальные и текущие оценки заготовок, имеют следующий вид:

$$y_i^0 = \frac{Ll_i}{b_i} \sum_{j=1}^n \frac{a_{ij}}{L - h_j}; \quad (8)$$

$$y_i^{(\nu)} = \frac{1}{b_i} \left[\frac{Ll_i a_{i,\nu-1}}{L - h_j} + y_i^{\nu-1} b_i^{\nu} \right], \quad (9)$$

h_j – длина остатка раскроя с номером j .

Каждый очередной допустимый раскрой генерируется таким образом, чтобы его суммарная оценка заготовок была достаточно высока. Прутки выбираются согласно списка π^l , при этом прутки i используются, пока $b_i \neq 0$. Но следует заметить, что поиск $\max(\alpha, y)$ не дает наилучшее решение в целом, так как очень плотное заполнение первых карт раскроя приводит к ситуации «дефицита хороших комбинаций». Нужно так организовать процесс генерирования раскроев, чтобы оставался необходимый резерв заготовок. Это достигается с помощью ведения барьера, непосредственно связанного с текущими значениями оценок. Величина начального барьера характеризует оценку раскраиваемого материала в среднем и в рассматриваемых ниже алгоритмах подсчитывается следующим образом:

$$\Delta(L) = \frac{f^0(L) + \tilde{f}(L)}{2}, \quad (10)$$

где $f^0(L)$ и $\tilde{f}(L)$ – соответственно верхняя и нижняя граница оценки материала длины L . При этом $f^0(L) = L$, а

$$\tilde{f}(L) = y_{i_0}^* \cdot L. \quad (11)$$

Для включения в раскройный план достаточно найти раскрой, для которого

$$(\alpha, y) \geq \Delta. \quad (12)$$

Если не удалось найти раскрой α , для которого выполняется это условие, то величина барьера снижается.

Барьерная политика позволяет также ограничивать перебор возможных способов раскроя. Генерирование раскроя производится в соответствии с приоритетным списком (π^d), в котором заготовки ранжированы по убыванию их удельных оценок. Уточним, что за удельную оценку принимается величина

$$y_i^* = \frac{y_i}{l_i}. \quad (13)$$

Для каждого очередного раскраиваемого остатка длины L вычисляется текущий барьер путем уменьшения имеющегося к данному моменту барьера на величину суммарной оценки полученных уже заготовок, а также верхняя оценка

$$\tilde{f}_i(L) = y_i^* L \quad (14)$$

этого остатка, где i — номер приоритетной заготовки остатка длины l .

Перспективность раскроя остатка оценивается следующим соотношением

$$\tilde{f}_i(L) > \Delta(L). \quad (15)$$

При условии выполнения (15) рассматриваемая заготовка укладывается в остаток, т. е. производится шаг вперед. В противном же случае делается шаг назад, т. е. заготовка признается неперспективной.

Наличие вектора

$$\beta^{(\nu)} = (\beta_1^{(\nu)}, \beta_2^{(\nu)}, \dots, \beta_m^{(\nu)}), \quad (16)$$

компоненты которого учитывают требуемое количество заготовок после получения ν раскроев, позволяет установить признак конца решения задачи. Если все элементы данного вектора равны нулю, то процесс раскроя заканчивается.

Для оценки эффективности алгоритма решения задач линейного раскроя был проведен численный эксперимент. Исходные данные для задач моделировались с помощью программного датчика случайных чисел с нормальным законом распределения. Всего было просчитано более 2000 различных примеров. Входными параметрами для каждой задачи являлись параметры раскраиваемого материала и заготовок. В качестве критерия оценки эффективности решения принималась величина N^{opt} , характеризующая оптимальное решение. Таким образом, оптимальный план раскроя был известен заранее, и алгоритму предлагалось построить аналогичный (возможно и отличный по структуре, но дающий

в итоге количество раскrojенных прутков, равное N^{opt}) план раскроя.

На основании проведенного эксперимента сделан вывод, что статистическая оценка эффективности алгоритма равна 9/10, т. е. в 9 случаях из 10 был получен оптимальный план раскроя. Далее, рассматривалась случайная величина D_0 , представляющая собой отношение суммарного остатка к общему расходу материала. Таким образом, величина D_0 показывает долю отхода в общем расходе материала. Согласно центральной предельной теореме случайная величина D_0 подчиняется нормальному закону распределения. Используя стандартные приемы нахождения доверительного интервала случайной величины, подчиненной нормальному закону распределения при неизвестной точности измерения (σ) был построен доверительный интервал для D_0 . Для результатов измерений среднее значение и среднее квадратическое отклонение соответственно составляют $\bar{X} = 0,0066$ и $S = 0,02092$. По заданной надежности $\gamma = 0,98$ и числу измерений $n = 2000$ была получена доверительная оценка истинного значения D_0 . Таким образом, с надежностью 0,98 можно считать, что значение D_0 заключено в интервале (0,0006 ; 0,0126).

Среди приближенных методов решения задачи одномерной упаковки в контейнеры наилучшим считается «первый подходящий с упорядочиванием» (FFD). Априорная оценка поведения в худшем случае (V) данного алгоритма равна 1,22. Алгоритм, разработанный в диссертации, имеет гарантированную оценку отклонения $V \leq 1,22$. Это вытекает из того факта, что в указанном алгоритме начальный раскройный план, на основании которого рассчитываются псевдо оценки, находится с помощью алгоритма FFD, и в процессе решения данный план улучшается. По результатам проведенного вычислительного эксперимента можно сделать вывод о том, что алгоритм построения рациональных (оптимальных) планов раскроя линейного материала обеспечивает хорошую точность решения и обладает большой временной эффективностью.

3. ЗАКЛЮЧЕНИЕ

Рассмотрена задача линейного раскроя, возникающая в строительной индустрии при изготовлении окон. Основное ее отличие от традиционных задач раскроя состоит в том, что материал на раскрой поступает в виде хлыстов (прутков) различных размеров.

Предложен для решения задачи эволюционный алгоритм на основе гибридизации метода последовательного уточнения оценок (SVC) и наивного алгоритма $((1 + 1) - EA)$. В первом алгоритме с целью сдерживания его жадности введена оригинальная барьерная политика.

СПИСОК ЛИТЕРАТУРЫ

1. **Мухачева, Э. А.** Математическое программирование / Э. А. Мухачева, Г. Ш. Рубинштейн. Новосибирск : Паука, СО, 1987. 236 с.
2. **Канторович, Л. В.** Расчет рационального раскроя материалов / Л. В. Канторович, В. А. Залгаллер. Л. : Лениздат, 1951. 289 с.
3. **Канторович, Л. В.** Рациональный раскрой промышленных материалов / Л. В. Канторович, В. А. Залгаллер. Новосибирск : Наука, СО, 1971. 299 с.
4. **Gilmore, P. C.** A linear programming approach to the cutting-stock problem / P. C. Gilmore, R. E. Gomory // *Operations Research*. 1961. № 9. P. 849–859.
5. **Belov, G. N.** A cutting plane algorithm for the one-dimensional cutting stock problem with multiple stock lengths / G. N. Belov, G. Scheithauer // *European J. of Operational Research*. 2002. № 141. P. 274–294.
6. **Folkenaucr, E.** A hybrid grouping genetic algorithm for bin packing / E. Folkenaucr // *J. of Heuristics*. 1998. № 2 (1). P. 5–30.
7. **Мухачева, Э. А.** Метод последовательного уточнения оценок: алгоритм и численный эксперимент для задачи одномерного раскроя / Э. А. Мухачева, А. С. Мухачева, Г. Ш. Белов // *Информационные технологии*. 2000. № 2. С. 11–17.

ОБ АВТОРЕ



Тарасов Андрей Евгеньевич, аспирант каф. выч. техн. и защиты информации. Дипл. экон.-матем. (УГАТУ, 2004). Готовит дисс. в обл. автоматизации проектирования раскройно-заготовительного производства светопрозрачных конструкций.