

ЭВОЛЮЦИОННЫЙ ПОДХОД К РЕШЕНИЮ ЗАДАЧИ РАСКРОЯ ЛИСТА ДЕРЕВА ГИЛЬОТИННЫМ РАЗРЕЗОМ

Скобцов Ю.А, Фонов А.М.

Донецкий государственный технический университет,
кафедра автоматизированных систем управления

E-mail: fonotov@mail8.dgtu.donetsk.ua, skobtsov@mail8.dgtu.donetsk.ua

Abstract. *Skobtsov Y.A., Fonotov A.M. The evolutionary approach to solution of the task of a page of a tree by a section. In article the opportunity of application of genetic algorithms for the decision of a task of optimization of a cut of a sheet is considered. The analysis of existing methods of the decision of this task is given. The task of creation of genetic algorithm for the decision of a circle of soluble tasks is described.*

Введение. Существует большой круг задач, для которых еще не разработаны достаточно быстрые алгоритмы поиска оптимальных или субоптимальных решений. К этому классу задач относятся и задачи линейной и нелинейной оптимизации. Одной из таких задач является задача оптимального размещения прямоугольных элементов (деталей) на плоскости листа. В условиях нехватки древесины (на Украине эта проблема стоит довольно остро) перед деревообрабатывающей промышленностью стоит задача минимизировать потери дерева при его промышленной обработке.

Производство мебели и других изделий из листового дерева условно можно разделить на две категории: изготовление фигурных прямоугольных. Эти изделия вырезаются из листов прямоугольной формы (заготовок). От вида изделий зависит метод раскроя и распила листового дерева. Сегодня на мебельных предприятиях Украины и зарубежья, для раскроя листа дерева используют станки разных фирм, но имеющие один принцип разреза листа, так называемый гильотинный раскрой. Этот метод заключается в последовательном разрезании прямоугольников на две части. Пример такого разреза приведен на рис. 1. Технологический процесс разреза листа происходит следующим образом: на станок подается разрезаемый лист, а в управляющую станком программу загружается схема раскроя листа (в виде файла в специальном формате). От типа древесины и ее толщины зависит используемая при распиле пила; толщина пилы определяет ширину пропила (т.е. прямые потери дерева при распиле). На рис. 1 область пропила показана пунктирной линией.

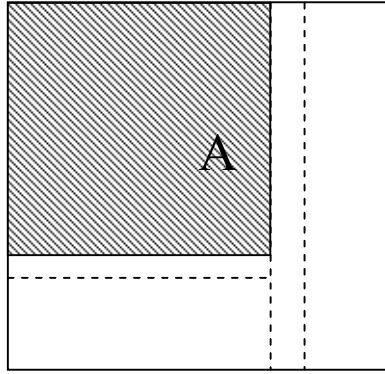


Рисунок 1 — Гильотинный разрез листа

Изготовитель мебели обычно начинает обработки множества требуемых деталей, которые ему были заказаны, и набора листов (заготовок), которые предстоит резать. Потери имеют место, когда лист не покрывается точно требуемыми деталями. Целью данной работы является исследование существующих методов раскроя, анализ их достоинств и недостатков, и на их основе разработка нового метода решения задачи раскроя листа гильотинным разрезом.

Анализ существующих методов решения. Рассматривается задача разделения множества плоских родительских прямоугольников (далее заготовок) $P_i(L_{0i}, W_{0i})$, длиной L_{0i} и шириной W_{0i} ($i=1, \dots, n$), на меньшие части (далее детали) (L_j, W_j) каждый длиной L_j и шириной W_j ($j = 1, \dots, m$). Каждая j -й деталь имеет, в общем случае, не фиксированную ориентацию (то есть может вращаться на 90°); она должна быть вырезана (ширина пропила задается) параллельно к граням родительского прямоугольника (то есть используется гильотинный разрез); число вырезанных деталей каждого типа j , должно удовлетворять условию ($0 \leq N_j \leq M_j$), где N_j — число вырезаемых деталей j -го типа, M_j — число деталей j -го типа которые необходимо вырезать. При этом решается задача максимизации средней плотности заполнения всех листов участвующих в раскрое. Средняя плотность заполнения j -го листа Q_j вычисляется по формуле 1.

$$Q_i = \frac{\sum_{j=1}^{m_i} L_j * W_j}{L_{0i} * W_{0i}}, \quad (1)$$

где $\sum_{j=1}^{m_i} L_j * W_j$ — определяет используемую полезную площадь листа.

$L_{0i} * W_{0i}$ — площадь j -го листа.

Средняя плотность заполнения Q вычисляется по формуле 2.

$$Q = \frac{\sum_{i=1}^{n-1} Q_i}{n-1}, \quad (2)$$

где n — количество использованных листов. Последний лист при расчете средней плотности заполнения, как правило, не учитывается.

Из-за технологических условий распила листа, разрезаемый образец должен также удовлетворять дополнительному набору требований:

- минимальное расстояние между смежными параллельными отрезками должно быть соблюдено, это расстояние определяется шириной пропила;
- края листа не могут использоваться из-за их возможного повреждения, поэтому перед началом раскроя лист обрезается по периметру примерно на 20 мм. С каждой стороны. Ширина пропила зависит от породы дерева, ширины листа, полировки;
- в результате раскроя могут появляться два вида остатков: мерные и не мерные.

Под мерными остатками понимают неиспользованные части листа, удовлетворяющие условиям 3:

$$\begin{aligned} x_i &\geq X_{min}, \\ y_i &\geq Y_{min}, \end{aligned} \quad (3)$$

где x_i и y_i — соответственно ширина и высота остатков,

X_{min} , Y_{min} — минимальная ширина и высота остатка подлежащего повторному использованию.

Из-за приведенных выше ограничений весьма трудно определить некоторую эвристику эффективного раскроя листов.

На сегодняшний день можно выделить четыре основных типа методов поиска оптимальных решений данной задачи [7]:

1. Методы, основанные на решении задач нелинейной оптимизации.
2. Методы перебора.
3. Вероятностные методы.
4. Методы эвристического поиска.

Двумерные задачи раскроя, в различных интерпретациях, широко рассматривались в литературе [1 .. 5], но в них решался только частный случай поставленной задачи. В этих работах задача нахождения оптимального раскроя решается не для всего множества родительских прямоугольников (заготовок), а для каждого из них в частности. Обобщенный алгоритм решения задачи выглядит следующим образом:

1. Задается множество заготовок P и множество деталей D .
2. Из множества заготовок выбирается один лист P_i .
3. Из множества деталей выбирается такое подмножество D_{0i} , что лист P_i раскраивается оптимальным образом.
4. Из множества деталей D исключается множество D_{0i} .
5. Из множества P исключается P_i .
6. Если выполняются условия: множество $D \neq \emptyset$ и множество $P \setminus D \neq \emptyset$ возвращаемся к шагу 2.

Такой подход обеспечивает качественное заполнение первых листов из множества P , но затем плотность заполнения начинает заметно снижаться. В таблице 1 приведены данные, полученные посредством тестирования программы раскроя листов дерева “Астра” [8].

На основании результатов проведенных экспериментов можно утверждать, что при увеличении количества раскраиваемых деталей улучшаются качественные показатели заполнения листов с 84% до 86% при увеличении количества деталей в 5 раз (см. Таблицу 2).

Таблица 1а — Изделие “Шкаф”

Деталь	Размеры детали		Количество деталей, шт.	Площадь	
	Высота, мм.	Ширина, мм		Детали, мм ²	Всего, мм ²
1	2100	650	4	1365000	5460000
2	1800	650	2	1170000	2340000
3	900	620	2	558000	1116000
4	900	750	2	675000	1350000
5	750	650	6	420000	2520000
6	930	720	6	669600	4017600
7	1200	520	1	624000	624000
8	500	780	2	390000	780000
9	2100	456	3	957600	2872800
10	689	456	3	314184	942552
11	980	750	1	735000	735000
Всего					22757952

Таблица 16 — Результаты распила программой “Астра”

Изделие	Шкаф	
Количество деталей	32x3	
Общая площадь деталей	83273856	
Лист для раскроя	3600x1800	
Ширина распила	4	
Количество листов	14	
Средняя плотность заполнения, без последнего	0,84	
Количество остатков	54	
Номер листа	Количество деталей на листе	Плотность заполнения, %
1	7	0,96
2	7	0,96
3	9	0,91
4	9	0,91
5	11	0,9
6	6	0,87
7	10	0,84
8	8	0,78
9	8	0,77
10	8	0,7
11	5	0,58
12	4	0,55
13	4	0,54
14	3	0,36

Таблица 2 — Зависимость плотности заполнения от количества раскраиваемых деталей

Количество деталей	Количество листов	Средняя плотность заполнения
32	4	84,5
64	9	84
96	13	84
160	22	86
320	41	86

В рассмотренных алгоритмах обнаружены следующие недостатки:

1. При оптимизации распределения множества деталей на множестве заготовок рассматривается ограниченная задача, т.е. последовательно для каждого листа из множества заготовок рассматривается задача оптимизации только этого листа.

2. При увеличении количества заготовок существенно увеличивается время решения задачи оптимизации.
3. Рассмотренные алгоритмы не могут предоставить законченного решения до их окончания, и не могут пополнять множество заготовок на этапе поиска решения.

Поэтому необходимо разработать более совершенные методы решения этой задачи. В данной работе представлен новый эволюционный подход к решению задачи раскроя листов дерева.

Генетический алгоритм для раскроя листа. Каждый индивидуум в популяции представляет собой вариант раскроя. Индивидуум кодируется с помощью набора деталей (генов) расположенных по порядку и флага возможности их поворота (вокруг оси на 90°). Метод раскроя использует этот порядок для последовательного размещения элементов на плоскости листов. Кроме того, на этапе получения новой популяции, генотип каждого индивидуума рассматривается двумя способами: а) как последовательность генов, б) как набор генов, каждый раскроенный лист из набора рассматривается как неделимая единица (далее блок генов). Цель ГА состоит в том, чтобы использовать такую лучшую перестановку генов и наборов генов, чтобы алгоритм размещения находил оптимальный разрезаемый образец.

Задачей оптимизации является максимизировать среднюю плотность заполнения набора листов. Оптимизация выполняется одновременно для всех листов, которые рассматриваются алгоритмом размещения.

Алгоритм размещения. Алгоритм размещения — простая процедура, которая рассматривает деталь и пробует разместить её в одну из текущих неиспользованных областей (отходов) листа. При появлении не мерных остатков они неизбежно становятся отходами и больше не рассматриваются. Процедура размещения рассматривает две позиции расположения детали (если поворот детали возможен) рисунок 2.



Рисунок 2 — Способы размещения детали на плоскости листа

Процедура работает таким образом, что как только деталь размещена на листе, выбор никогда не пересматривается. Задачей ГА является упорядочение деталей в таком порядке, что бы далее алгоритм размещения привел к хороше-

му варианту раскроя с минимальными отходами. Для этого, процедура рассматривает все остатки, и находит тот, для которого деталь будет размещаться наилучшим образом.

Фитнесс функция. Фитнесс функция непосредственно принимает во внимание полную область потерь. Область потерь состоит из: потери при распиле листа (ширина пропила), потери немерными остатками, потери мерными остатками. Вследствие того, что мерные остатки (остатки, имеющие размеры соизмеримые с минимальным размером детали) могут использоваться в последующих раскроях, используются фитнес функция, основанная на системе штрафов и поощрений. В качестве штрафа выбирается некоторая функция, принимающая положительные значения на недопустимой области и нулевое значение на области, где выполняются все ограничения.

$$Z = \min(k_1 * O(L(f_j)) + k_2 * P(S_{общ}(f_j)) + k_3 * Q(C(f_j))) \quad (4)$$

где f_j — вариант размещения, k_1, k_2, k_3 — весовые коэффициенты, $O(L(f_j))$ — оценка общей площади используемых листов, $P(S_{общ}(f_j))$ — общая площадь не востребованной площади листа, $Q(C(f_j))$ — оценка общего количества мерных остатков.

Представление хромосомы. Каждый индивидум — упорядоченная последовательность деталей (генов), которые участвуют в раскрое. Индивидум представлен двумя взаимосвязанными хромосомами:

- последовательность геометрических характеристик деталей рис.3а (первая строка — ширина, вторая — высота, третья — возможность поворота: 1 — возможен, 0 — невозможен);
- параметр связности с соседней деталью (−1 — левой, 1 — правой, 0 — не установлен) рис. 3б.

Две соседние детали считаются связными, если один из их геометрических параметров (ширина или высота) равен одному и тому же значению. Связанные детали рассматриваются при раскрое, как одна деталь.

Н1	300	650	450	350	200	400	600
	300	300	350	350	350	350	300
	0	1	0	1	1	1	0

а

Н2	1	−1	1	−1	1	−1	0
-----------	---	----	---	----	---	----	---

б

Рисунок 3 — Представление исходных данных об индивидуме в хромосоме

Генетические Операторы. Репродукция. В качестве оператора репродукции используется процедура, выбирающая случайным образом две особи из популяции. Выборка осуществляется по равномерному закону распределения.

Кроссинговер. Используется два варианта оператора кроссинговера:

1. Случайно выбирается точка кроссинговера. Далее из 1-го родителя копируем участок хромосомы от 1 гена до точки кроссинговера в потомок1. Участок потомка1 после точки кроссинговера заполняется генами 2-го родителя в порядке расположения. Второй потомок получается следующим образом: из 2-го родителя участок хромосомы от 1 гена до точки кроссинговера копируется в потомок2. Участок потомка2 после точки кроссинговера заполняется генами 1-го родителя в порядке расположения. Например:

Родитель1.

Ширина	300	650	450	350	200	400	600
Высота	300	300	350	350	350	350	300
Флаг поворота	0	1	0	1	1	1	0
Параметр связности	1	-1	1	-1	1	-1	0
№	1	2	3	4	5	6	7

Родитель2.

Ширина	650	350	450	600	300	400	200
Высота	300	350	350	300	300	350	350
Флаг поворота	1	1	0	0	0	1	1
Параметр связности	0	1	-1	1	-1	1	-1
№	2	4	3	7	1	6	5

Точкой кроссинговера выбран третий ген.

Примечание: пустая ячейка соответствует месту разреза, жирным выделены гены первой хромосомы, курсивом — гены 2-ой хромосомы. В результате скрещивания получаем две дочерние хромосомы $H1^1$ и $H1^2$.

Потомок1.

Ширина	300	650	450	350	600	400	200
Высота	300	300	350	350	300	350	350
Флаг поворота	0	1	0	1	0	1	1
Параметр связности	1	-1	0	0	0	1	-1
№	1	2	3	4	7	6	5

Потомок2.

Ширина	650	350	450	300	200	400	600
Высота	300	350	350	300	350	350	300
Флаг поворота	1	1	0	0	1	1	0
Параметр связности	0	1	-1	0	1	-1	0
№	2	4	3	1	5	6	7

2. В отличие от первого этот оператор работает с блоками генов. Блоки генов образуются после проведения раскроя, в них входят гены, расположенные на одном листе. В результате этой операции кроссинговера появляется всего один потомок. Процедура кроссинговера работает следующим образом:

Из первого родителя выбирается N блоков (первоначально $N=5-10\%$ от общего числа блоков, и увеличивается до 30% по мере решения), которые имеют лучший показатель плотности заполнения и копируются в начало потомка. Из второго родителя выбирается M блоков (где $M \in [0..N]$) имеющих лучший показатель плотности заполнения, но не включающие в себя тех деталей, которые уже скопированы в потомок. Эти M блоков копируются в потомок и располагаются за блоками, скопированными из первого потомка. Оставшийся не заполненным участок потомка заполняется деталями, не включенными в уже скопированные блоки, в том порядке, в котором они появляются во втором родителе. Пример приведен на рисунке 4.



Рисунок 4 — Второй оператор кроссинговера

Из первого родителя выбирается лист с максимальной плотностью заполнения (указана в скобках после номера листа); это Лист2 с плотностью запол-

нения 86%. Множество деталей из Лист2 первого родителя копируется в потомок с сохранением порядка следования. Из второго родителя выбирается Лист4 с плотностью заполнения 82%, так как Лист2 (плотность заполнения которого больше и равна 84%), содержит деталь №6, которая уже присутствует в потомке. Множество деталей из Лист4 второго родителя копируется в потомок с сохранением порядка следования. Оставшийся незаполненным участок потомка заполняется генами 2-го родителя в порядке их расположения.

Примечание: количество деталей на каждом листе может быть различным, в примере приведено одинаковое число деталей в каждом листе для удобства восприятия.

Мутация. Мутации подвергается 10% особей популяции. В алгоритме используется два равноправных оператора мутации.

1. Первый оператор мутации случайным образом меняет несколько (1-5%) генов местами. Для примера возьмем хромосому H1.

Родитель1.

Ширина	300	650	450	350	200	400	600
Высота	300	300	350	350	350	350	300
Флаг поворота	0	1	0	1	1	1	0
Параметр связности	1	-1	1	-1	1	-1	0
№	1	2	3	4	5	6	7

Так как число генов в хромосоме мало, мутации подвергается пара генов. Эта пара определяется с помощью генерации двух случайных чисел распределенных по равномерному закону распределения (равные права у всех генов). Гены с выпавшими номерами меняются местами. В данном примере это гены 4 и 6.

Потомок.

Ширина	300	650	450	400	200	350	600
Высота	300	300	350	350	350	350	300
Флаг поворота	0	1	0	1	1	1	0
Параметр связности	1	-1	0	0	0	0	0
№	1	2	3	6	5	4	7

При этом параметр связности будет сброшен в 0 для 3,4,5 и 6 гена.

2. Второй оператор мутации предназначен для работы со второй хромосомой индивидуума, и задает параметр связности. Работа второго оператора мутации происходит следующим образом: случайно выбирается хромосома, в ней производится поиск связанных элементов, если такие элементы найдены, устанавливается параметр связности. Например:

Родитель1.

Ширина	300	650	450	350	200	400	600
Высота	300	300	350	350	350	350	300
Флаг поворота	0	1	0	1	1	1	0
Параметр связности	0	0	0	0	0	0	0
№	1	2	3	4	5	6	7

Предок.

Ширина	300	650	450	350	200	400	600
Высота	300	300	350	350	350	350	300
Флаг поворота	0	1	0	1	1	1	0
Параметр связности	1	-1	1	-1	1	-1	0
№	1	2	3	4	5	6	7

Редукция. В алгоритме предусмотрено использование оператора Редукция. В результате в новую популяцию переносится 5 процентов индивидуумов имеющих максимальное значение фитнес функции.

Для постоянного “омолаживания” популяции 5% индивидуумов генерируются заново.

Заключение. Рассмотрена проблема оптимизации размещения прямоугольных объектов на плоскости. Для решения задачи предложен подход, основанный на генетических алгоритмах и предназначенный для поиска подмножества субоптимальных решений.

Разработана модель кодирования информации о размещении в геноме. Найдены, согласованные с данной моделью, модифицированные генетические операторы: мутации, скрещивания и генерации.

Основным преимуществом предложенного решения данной задачи является решение глобальной задачи оптимизации, а не отдельного листа. Предло-

женный алгоритм может динамически расширять область решения задачи (ввод новых данных) по ходу ее оптимизации. Преимуществом является также тот факт, что в любой момент времени можно получить текущее решение для всей задачи.

Предложенный метод можно использовать не только для раскроя листа дерева. Модификации приведенного метода можно использовать для раскроя листа стекла, ткани, одномерного раскроя таких материалов как пластиковые и железные трубы и т.п.

Литература

1. Holland, J.H. (1975) "Adaptation in Natural and Artificial Systems", Ann Arbor, MI: The University of Michigan Press. 2nd edn. (1992) Boston, MA: MIT Press.
2. Д.И.Батищев, С.А.Исаев, Е.К.Ремер "Эволюционно-генетический подход к решению задач невыпуклой оптимизации". — Сборник статей ВГТУ, 1998.
3. Батищев Д.И., Исаев С.А. Оптимизация многоэкстремальных функций с помощью генетических алгоритмов / Межвуз. сб. "Высокие технологии в технике, медицине и образовании" Часть 3. — Воронеж, ВГТУ. 1997.
4. Goldberg, D.E. (1989) "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison-Wesley.
5. Mitchell, M. An Introduction to Genetic Algorithms. MIT Press, Cambridge, MA (1996).
6. Липницкий А.А. "Применение генетических алгоритмов к задаче о размещении прямоугольников", Кибернетика и системный анализ.
7. Методы оптимизации// Сост.Желтов В.П.,Поколев С.С. — ЧГУ, 1994.
8. А.с. ПА № 4052. "Автоматизована система розкрою листових матеріалів "Астра-Д" / Д.В. Удовицький, В.К. Паргас // Державне агентство по реєстрації авторських прав — 21 березня 2001 р.
9. F. Corno, P. Prinetto, M. Rebaudengo, M. Sonza Reorda, "Optimizing Area Loss in Flat Glass Cutting", Bottero SpA, Automazione Vetro Piano, Cuneo, Italy.

Сдано в редакцию: .03.2003г.

Рекомендовано к печати: д.т.н., проф. Скобцов Ю.А.