

УДК 004:338.3

М. А. ВЕРХОТУРОВ

ЗАДАЧА НЕРЕГУЛЯРНОГО РАСКРОЯ ФИГУРНЫХ ЗАГОТОВОК: ОПТИМИЗАЦИЯ РАЗМЕЩЕНИЯ И ПУТИ РЕЖУЩЕГО ИНСТРУМЕНТА

Рассматривается проблема нерегулярного раскроя фигурных заготовок. Ее решение состоит из нескольких этапов, наиболее сложными из которых являются следующие, относящиеся к задачам оптимизационного геометрического моделирования: плотное размещение геометрических объектов в заданных областях и генерация пути режущего инструмента. Для их решения разработаны: класс годограф-ориентированных алгоритмов, использующих метаэвристические методы локального поиска — *НО-МН* (*Hodograph Oriented MetaHeuristics*) и модификация алгоритма *GRASP* (*Greedy Randomized Adaptive Search Procedure*). Приводятся алгоритмы. Рассматривается структура автоматизированной системы раскроя. Описывается вычислительный эксперимент. *Раскрой; упаковка; плотное размещение геометрических объектов*

ВВЕДЕНИЕ

При производстве изделий в машиностроительной, текстильной, кожевенной и других отраслях промышленности приходится раскраивать материал на заготовки произвольной формы. Решение этой задачи состоит из ряда последовательно выполняемых шагов:

- аппроксимации исходных контуров заготовок с учетом технологических параметров режущего инструмента и раскраиваемого материала (диаметра режущего инструмента, толщины материала и т. д.);
- плотного размещения геометрических объектов (ГО), полученных на предыдущем шаге, в заданных областях;
- построения пути режущего инструмента с учетом генерации вспомогательных технологических траекторий и контуров (траекторий подхода/отхода к/от контуру/а, контуров обработки острых углов - петель, соединительных сегментов и т. д.);
- генерации программ для станка с ЧПУ.

К наиболее сложным из перечисленных относятся второй и третий этапы, которые являются задачами оптимизационного геометрического моделирования. С точки зрения комбинаторной сложности, они принадлежат к классу *NP*-трудных. Эти две задачи и обсуждаются в данной работе.

1. ПОСТАНОВКА ЗАДАЧИ

1.1. Плотное размещение геометрических объектов в заданных областях

Задача нерегулярного размещения ГО (*Irregular Cutting Stock Problem, ICSP*) рассматривается в следующей постановке: имеется область упаковки W и множество геометрических объектов $P = \{p_1, \dots, p_n\}$. Произвольную зафиксированную точку c_i объекта p_i назовем *центром* этого объекта. *Картой раскроя* называется изображение области упаковки W с расположенными на ней объектами p_i , $i = 1, \dots, n$, удовлетворяющими условиям: $p_i \cap p_j = \emptyset$, $p_i \cap W = p_i$. Пусть $S(x)$ *площадь* области (объекта) x . Предполагается также, что $S(W) \gg \sum_{i=1}^n S(p_i)$. Условно разделим карту на две части: *деловой остаток* U и *занятую* ее часть Q , так что $Q = W \setminus U$. Распределим *незанятую часть* H *использованной области* упаковки Q между объектами p_i , $i = 1, \dots, n$ и *внешностью* области упаковки Q , $i = 0$ на части y_i , $i = 0, 1, \dots, n$ так, чтобы выполнялись следующие условия: $y_i \cap y_j = \emptyset$, $\bigcup_{i=0}^n y_i = H$, $\bigcup_{i=1}^n p_i \cup \bigcup_{i=0}^n y_i = Q$. Целевая функция записывается следующим образом: $S(H) = S(\bigcup_{i=0}^n y_i) \rightarrow \min$.

Суть постановки задачи — разместить геометрические объекты так, чтобы для полученной карты раскроя минимизировались непокрытые части (остатки) области размещения. Она полностью согласуется с общеизвестной постановкой по минимизации занятой части полосы (листа) (доказательство этого утверждения тривиально), но, в отличие от последней, применима и для областей размещения произвольной формы.

Качество карты раскроя характеризуется величиной коэффициента раскроя K_p , определяемой как $K_p = \frac{S(P)}{S(Q)}$, где $S(P) = \sum_{i=1}^n S(p_i)$.

Даже с введением одного из самых распространенных упрощений — фиксации ориентации ГО, данная задача является чрезвычайно сложной многоэкстремальной задачей математического программирования в пространстве высокой размерности с невыпуклой и несвязной областью допустимых решений.

1.2. Построение пути режущего инструмента

Рассматривается следующая постановка задачи построения пути режущего инструмента (ПРИ): имеется множество геометрических объектов n , состоящих из внутренних и внешних контуров. Каждый контур имеет точку врезки, траектории подхода и отхода.

Пусть l_{jk}^i ($j, k = 1 \div (ic_i + 1)$; $j \neq k$; $i = 1 \div n$) — длина перехода от точки отхода одного внутреннего контура до точки врезки другого, где ic_i — количество внутренних контуров i -го ГО. Последний переход в рамках объекта делается на точку врезки ее внешнего контура. Обозначим через $L_{im_q}^q$ ($i = 1 \div n$; $m_q = 1 \div ic_q$; $q = 1 \div n$; $i \neq q$) длину перехода от точки отхода внешнего контура i -го ГО до m_q -го внутреннего контура q -го объекта, через $L_{0m_q}^q$ — длину перехода из начальной точки расположения режущего инструмента к m_q -контур q -го ГО, L_0^n — длину перехода с внешнего контура n -го объекта в начальную точку расположения режущего инструмента. Необходимо найти такую последовательность объектов и внутренних контуров, что:

$$\min \left(L_{01}^1 + \sum_{i=1}^n \left(\sum_{j=1}^{ic_i+1} l_{jj'}^i + L_{i1}^{i'} \right) + L_0^n \right).$$

В такой постановке задача нахождения кратчайшего ПРИ представляет собой оп-

тимизационный вариант геометрической задачи о коммивояжере. Причем соответствующий граф может являться смешанным и неполным (при наличии у деталей внутренних контуров).

2. МЕТОДЫ РЕШЕНИЯ

Обе перечисленные выше задачи можно рассматривать как задачи дискретной оптимизации. И если задача построения пути режущего инструмента относится к соответствующему классу непосредственно, то задача нерегулярного размещения ГО переводится в этот класс при помощи принципа пообъектного размещения. Последний заключается в том, что на каждом шаге решения оперируют отдельными ГО, т. е. производят некоторые геометрические преобразования (изменение координат размещения и угла поворота ГО в области) каждого из них. В этом случае для решения задачи размещения используются методы моделирования геометрических преобразований, одним из наиболее применяемых является метод, основанный на моделировании движения ГО в области допустимых размещений с учетом их взаимного непересечения. Он базируется на понятии годографа функции плотного размещения [1]. В данной статье также рассматриваются методы решения, использующие это понятие.

Основная проблема при решении задач дискретной оптимизации заключается в том, что одни из используемых методов приводят к большому перебору, другие — останавливаются в точках локального оптимума, а таких точек может оказаться слишком много, и они могут сильно отличаться от глобального оптимума.

В последнее время широкое распространение получили такие эвристические методы, которые подходят для нахождения приближенных решений задач дискретной оптимизации любого типа, т. е. являются инвариантными по отношению к виду задачи. Это — так называемые метаэвристические методы. Суть этих методов часто следует или из разумных умозаключений — метод «поиск с запретами» TS (*tabu search*), или из механизмов функционирования, «скопированных» с живой: генетический алгоритм GA (*genetic algorithm*), алгоритм муравьиной колонии AC (*ant colonies*) или неживой природы: моделирование отжига SA (*simulated annealing*), алгоритм всемирного потока GDA (*great deluge algorithm*) и др. [2].

2.1. Задача нерегулярного размещения геометрических объектов

Рассмотрим основные идеи некоторых из метаэвристик и способы, при помощи которых эти методы можно применить для решения задач нерегулярного размещения геометрических объектов (ICSP).

2.1.1. Годограф-ориентированная адаптация «Генетических алгоритмов» (НО-GA)

Применение концепции эволюционного развития живой природы посредством естественного отбора для задач оптимизации было представлено Холландом [3] в 1975 году. Разработанные и исследованные им алгоритмы были названы генетическими. Систематизированное рассмотрение генетических алгоритмов было дано Голдбергом [4] в 1989 году. Эти алгоритмы получили очень широкое распространение для решения разнообразных задач дискретной оптимизации.

Идея «Генетических алгоритмов» состоит в том, чтобы после предварительного генерирования фиксированного множества решений продолжить поиск, моделируя на этом первоначальном наборе процедуры мутации и скрещивания, и отбрасывая те решения, которые имеют наихудшие значения целевой функции.

Рассмотрим возможную реализацию метода GA для решения ICSP на основе моделирования годографа и метода последовательно-одиночного размещения (ПОР). Отличительной особенностью метода ПОР является использование приоритетного списка (ПС). Адаптация общего инкрементального GA к задаче нерегулярного размещения ГО затрагивает следующие части алгоритма:

- представление задачи, т. е. кодирование решений задачи с помощью хромосом (строк);
- декодирование и оценка, т. е. присвоение решений задачи хромосомам и вычисление значений приспособленности;
- генетические операторы: кроссовер и мутация.

Представление задачи. Решение задачи определяется в терминах генетической теории следующим образом. Хромосома – это вектор размещения $chrom(i)$, $i = 1, \dots, n$, который указывает последовательность размещений ГО и указывает ориентацию (номер поворота) каждого ГО. Элемент $chrom(i)$ вектора размещения – это структура данных, содержащая два элемента данных: индекс it ГО и номер его поворота $rotate$. Значение $ro-$

$tate$ означает ориентацию ГО относительно его исходного положения. Возможные значения углов поворота задаются заранее.

В ICSP в качестве особи a_k^t выступает конкретное решение – размещение ГО, удовлетворяющее условиям взаимного непересечения. Это позволяет интерпретировать процедуру решения экстремальной задачи как эволюционный процесс, связанный с перераспределением ГО в области размещения для отыскания глобального минимума критерия оптимальности. В этом и заключается в данном случае цель эволюционного развития (эволюции) особей.

Декодирование и оценка. Декодирование хромосомы предназначено для генерации соответствующего допустимого решения задачи. Описание процесса декодирования требует точного определения решения из имеющейся хромосомы.

Решение, аналогично хромосоме, определяется вектором размещения $sol(i)$, $i = 1, \dots, n$. Элемент $sol(i)$ также является структурой данных, которая составляется из четырех элементов данных: индекс it ГО, его вариант поворота $rotate$, координат x и y центра ГО $centrx$ и $centry$ в координатах области размещения.

Для получения координат $centrx$ и $centry$ применяется аппарат моделирования годографа. В качестве значений $centrx$ и $centry$ i -го ГО используются координаты точки, выбранной из множества точек годографа, с минимальной абсциссой. Если таких точек несколько, то берется та, которая имеет минимальную ординату. Годограф строится для ГО с углом поворота $rotate$.

Процедура декодирования трансформирует каждую хромосому популяции в одно решение.

Целью оценки является вычисление значения приспособленности, которое связано с каждой хромосомой популяции. В простейшем случае используются соответствующие значения целевой функции, в качестве которой используется длина занятой части области размещения (для полубесконечной полосы).

Генетические операторы. При использовании этих операторов необходимо учитывать, что каждая позиция хромосомы кодирует не только номер ГО, но и вариант поворота.

Кроссовер. Процесс кроссовера может заключаться, например, в следующем. Пусть $ПС_i$, $ПС_j$ – две перестановки пары родителей и $ПС_{inew}$, $ПС_{jnew}$ – две новые перестановки.

новки, созданные скрещиванием. Сначала выбраны два случайных числа p и q , $1 \leq p, q \leq n$. В случайной позиции p скрещивающий оператор копирует q элементов из $ПС_i$, в начало $ПС_{i_{new}}$ и копирует q элементов $ПС_j$ в начало $ПС_{j_{new}}$. Затем $ПС_{i_{new}}$ дополняется отсутствующими в нем элементами из $ПС_j$ в том порядке, в котором они находятся в $ПС_j$, а $ПС_{j_{new}}$ аналогично дополняется другими элементами из $ПС_i$.

Например,

$ПС_i = \{1, 2, 3, 4, 5, 6\}$ и $ПС_j = \{6, 4, 2, 5, 3, 1\}$, если $p = 2$, $q = 3$, тогда:

$ПС_{i_{new}}(1) = ПС_i(p) = ПС_i(2) = 2$,

$ПС_{i_{new}}(2) = ПС_i(p+1) = 3$,

$ПС_{i_{new}}(3) = ПС_i(p+2) = 4$,

$ПС_{i_{new}}(4) = ПС_j(1) = 6$,

$ПС_{i_{new}}(5) = ПС_j(4) = 5$,

$ПС_{i_{new}}(6) = ПС_j(6) = 1$.

Таким образом, $ПС_{i_{new}} = \{2, 3, 4, 6, 5, 1\}$. Используя этот же метод, $ПС_{j_{new}} = \{4, 2, 5, 1, 3, 6\}$.

Мутация. Существует несколько подходов для выполнения оператора мутации [5]. Например, последовательность случайного подмножества из ПС разворачивается в обратном порядке или некоторые элементы обмениваются местами.

2.1.2. Годограф-ориентированная адаптация метода «Муравьиной колонии» (НО-АС)

Алгоритм муравьиной колонии является одним из самых молодых алгоритмов локального поиска. Он был предложен Дориго и Гамбарделла в 1997 году при моделировании муравьиной кучи [6].

Основная идея алгоритма АС состоит в реализации принципа коллективного разума. Для поиска экстремума целевой функции алгоритм использует параллельно несколько агентов (искусственных муравьев), которые в ходе поиска накапливают статистическую информацию. Эта информация аккумулируется в общедоступном банке данных и используется агентами независимо друг от друга. Каждый агент действует по правилам вероятностного алгоритма и при выборе направления ориентируется не только на приращение целевой функции, но и на статистическую информацию, отражающую предысторию коллективного поиска.

Метод АС является итеративным. На каждой его итерации определенное количество агентов строят допустимые решения задачи. Среди этих решений выбирается часть наи-

лучших по целевой функции, и в этой части отыскиваются повторяющиеся компоненты решений (например, дуги гамильтоновых циклов; элементы покрытий; вершины, включенные в клику и т. п.). Полученная информация запоминается, и на следующих итерациях данные компоненты будут иметь большую вероятность войти в решение, чем это было на предыдущих итерациях.

Критерием остановки является либо предельное число итераций, либо требуемая точность. Наилучшее из найденных решений, рекорд $F_{АС}$, является результатом работы алгоритма.

Применение алгоритма АС для ICSP использует аппарат моделирования годографа и метод последовательно-одиночного размещения.

На начальном этапе одним из простейших методов формируется ПС, например, по уменьшению площадей ГО. Затем при помощи аппарата моделирования годографа генерируется карта раскрыя K_0 , являющаяся первым допустимым решением. Для выбора точки годографа, в которой будет размещен ГО, используется локальная функция цели. В качестве целевой функции можно использовать расстояние от левой границы листа до вершин годографа, т.е. Таким образом, будет выбираться точка годографа с минимальной абсциссой.

Для реализации метода АС для ICSP необходимо определить следующие процедуры:

- механизм получения новых допустимых решений для каждого из агентов (способ перехода);
- отыскание повторяющихся компонентов.

Способ перехода. Каждый из k агентов характеризуется новым ПС, который формируется с помощью модификации исходного ПС, например, произвольным перемещением ГО или взаимной заменой двух случайных ГО в ПС. Согласно каждому из k приоритетных списков, генерируются раскройные карты K_i , которые характеризуются значением целевой функцией $C(K_i)$.

Повторяющиеся компоненты. Для отыскания повторяющихся компонентов решений вводится матрица $U = \|u_{ij}\|_{n \times n}$, каждый элемент которой u_{ij} характеризует качество размещения объекта P_i с объектом P_j . Величины u_{ij} могут вычисляться при помощи метода оценок [7]. Повторяющимися считаются те компоненты, которые характеризуются большими значениями величин u_{ij} .

2.1.3. Годограф-ориентированная адаптация метода «Моделирование отжига» (HO-SA)

Метод SA был предложен независимо друг от друга двумя авторами: Киркпатриком (1983) [8] и Церни (1985) [9] как метод минимизации функции многих переменных. Идея была взята из алгоритма Метрополиса (1953) [10], который предложил простой алгоритм моделирования процесса отжига.

Идея метода «Моделирование отжига» заключается в том, что вводится управляющий параметр — температура, которая влияет на величину перемещений между последовательно получаемыми решениями задачи оптимизации. Высокой температуре, с которой начинается процесс решения, соответствуют большие перемещения, а низкой температуре — маленькие перемещения. Большие перемещения определяются как такие изменения решений, которые вносят большие изменения в значения целевой функции, а маленькие перемещения — маленькие изменения в значениях целевой функции. Большие перемещения при высокой температуре означают, что алгоритм избегает локального минимума. Температура в процессе решения медленно уменьшается до минимального значения.

Определение начальной температуры T_0 , приращение ΔT на каждом шаге и определение «Состояния равновесия» называется «режимом охлаждения» (*cooling schedule*). Для определения параметров режима охлаждения применяются различные эвристики.

Адаптация алгоритма SA для ICSP на основе аппарата моделирования годографа и метода последовательно-одиночного размещения заключается в следующем.

Одним из простейших методов формируется ПС, при помощи аппарата моделирования годографа генерируется карта K_0 , которая является входной информацией для алгоритма, реализующего идеи метода SA. Далее для реализации метода SA необходимо определить:

- целевую функцию;
- механизм получения новых допустимых решений для каждого из агентов (способ перехода).

Функция цели. В отличие от известных реализаций метода SA, использующих произвольные движения, метод на основе моделирования годографа не включает в целевую функцию штрафную составляющую, начисляемую за взаимное пересечение ГО. Поэтому в качестве функции цели может быть ис-

пользована минимизация занятой части области размещения (для полубесконечной полосы).

Способ перехода. Для изменения расположения объектов используются три типа перемещений ГО в ПС:

- перемещение объекта на новое место;
- взаимная перестановка двух объектов;
- изменение ориентации объекта.

Основной сложностью при реализации метода SA является определение параметров «режима охлаждения», так как соответствующие им алгоритмы очень чувствительно относятся к изменениям этих параметров. Это характерно для решения большинства задач дискретной оптимизации. Для устранения этого недостатка Дюком [11] был предложен целый ряд алгоритмов, которые являются модификациями метода SA. Это такие алгоритмы, как: «Пороговой допустимости» (*Threshold Accepting — TA*), «Всемирного потопа» (*Great Deluge — GD*), «Перехода от рекорда к рекорду» (*Record-to-Record Travel — RRT*). Перечисленные оптимизационные эвристики имеют то преимущество перед методом SA, что они зависят только от одного параметра.

Применение алгоритмов «Пороговой допустимости» TA, «Всемирного потопа» GD, «Перехода от рекорда к рекорду» — RRT для решения ICSP производится аналогично адаптации метода SA с использованием годографа HO-SA.

2.1.4. Годограф-ориентированная адаптация метода «Поиск с запретами» (HO-TS)

Родоначальником алгоритма поиска с запретами (*tabu search — TS*) (1986) является Фред Гловер, который предложил следующую схему локального поиска [12, 13]. Пусть для задачи дискретной оптимизации

$$\min\{F(x)|x \in D\}$$

удалось найти некоторое допустимое решение x_0 . Рассмотрим окрестность $N(x_0) \subset D$ точки x_0 и найдем решение задачи

$$\min\{F(x)|x \in N(x_0)\}.$$

Обозначим его через x_1 . Рассмотрим теперь окрестность $N(x_1)$, найдем точку x_2 и т. д. до тех пор, пока $x_k \neq x_{k+1}$. Если $x_k = x_{k+1}$, то точка x_k является локальным оптимумом.

Идея алгоритма «Поиска с запретами» состоит в том, чтобы не останавливаться в локальном оптимуме, как это предписано в

алгоритмах локального спуска, а продолжить поиск, руководствуясь теми же правилами, запрещая посещение уже пройденных точек.

На k -м шаге алгоритма точка x_{k+1} находится из решения задачи

$$\min\{F(x) | x \in N(x_k) \setminus \cup_{i=0}^k \{x_i\}\}.$$

Множество $\cup_{i=0}^k \{x_i\}$ выступает в качестве списка запретов, что и послужило поводом для названия алгоритма. Этот список пополняется на каждом шаге новой точкой. Очевидно, что если старая информация не будет удаляться из списка, то работоспособность алгоритма будет падать с ростом числа итераций. Поэтому длина списка ограничивается сверху некоторой константой l , и список запретов содержит только последние l точек. Проверка и пополнение списка может оказаться весьма трудоемкой операцией. Поэтому иногда целесообразно хранить не сами решения x_i , $i = k - l + 1, \dots, k$, а либо функции от них (например, значения целевой функции), либо номера изменяемых координат, атрибуты перехода от x_k к x_{k+1} .

Обозначим через $N^l(x)$ часть окрестности $N(x)$, взятую случайным образом. Тогда алгоритм поиска с запретами можно представить в следующем виде.

Алгоритм TS

1. Выбираем начальное решение x , полагаем $F^* := F(x)$. Формируем пустой список запретов.

2. Находим новое решение z такое, что

а) $z \in N^l(x)$;

б) $F(z) = \min\{F(y) | y \in N^l(x)\}$;

в) переход $x \rightarrow z$ не является запрещенным или $F(z) < F^*$.

3. Переходим в новую точку $x := z$.

Изменяем список запретов.

Если $F(x) < F^*$, то меняем рекорд $F^* := F(x)$.

4. Если выполнен критерий остановки, то STOP, иначе идти на п. 2.

В качестве критерия остановки используется либо остановка по числу итераций, либо требуемая точность по отношению к заданной нижней границе. Начальное решение выбирается с помощью какого-то простого алгоритма. Меняя длину списка запретов, можно управлять процессом поиска. При уменьшении длины, интенсифицируется поиск в текущей области, увеличение длины способствует переходу к другой области. В качестве окрестности можно рассматривать множество всех

решений, которые получаются из текущего решения изменением одной из координат.

В 1992 году Файгл и Керн [14] предложили вероятностную версию метода TS, который при количестве итераций, стремящихся к бесконечности, сходится к глобальному оптимуму, что, впрочем, свойственно большинству вероятностных алгоритмов. Так рандомизация окрестности сокращает трудоемкость алгоритма на шаге 2 и вносит элемент случайности при выборе направления спуска. Если рандомизированная окрестность составляет от 1 до 10% от исходной, то алгоритм не заикливаются даже без списка запретов. Алгоритм TS применялся к широкому кругу задач дискретной оптимизации [15], показал высокую работоспособность и на сегодняшний день является одной из наиболее популярных вероятностных эвристик.

Одно из возможных применений метода «Поиск с запретами» для решения задач нерегулярного раскрыя-упаковки (P-У) предложено в работе [16] польскими авторами Блазевичем, Хаврулюком и Валковьяком.

Рассмотрим один из возможных способов применения метода TS для решения задачи нерегулярного размещения геометрических объектов (ICSP), основанный на применении годографа — HO-TS. Для его реализации используется также метод последовательно одиночного размещения, который подразумевает использование «приоритетного списка» — ПС для определения последовательности размещения геометрических объектов (ГО) в область P-У.

На начальном этапе одним из простейших методов формируется ПС, например, по уменьшению площадей ГО. Затем при помощи аппарата моделирования годографа генерируется карта P-У K_0 , которая является первым допустимым решением и входной информацией для алгоритма, реализующего идеи метода TS. Для выбора точки годографа, в которой будет размещен ГО, используется локальная функция цели. В качестве целевой функции может использоваться расстояние от левой границы листа до вершин годографа. Таким образом, будет выбираться точка годографа с минимальной абсциссой.

Метод «Поиск запретов» используется для улучшения качества решений, полученных после применения вышеописанного эвристического алгоритма. Он реализуется следующим образом — производится модификация приоритетного списка за счет изменения позиций в нем ГО.

Основные моменты метода TS применительно к решению задачи размещения плоских ГО состоят в следующем.

Способ перехода. Первая и наиболее важная проблема при адаптации метода TS для решения любой комбинаторной задачи — это определение способа перехода от одного допустимого решения к другому. В алгоритме TS способу перехода соответствует понятие *перемещение*. Перемещение s определяется в следующем виде: это изменение расположения ГО в ПС.

Список запретов. Список запретов T содержит все перемещения, неразрешенные для изменения ПС на текущей итерации. Это может быть определено следующим образом:

$$T = \{s^{-1} : s = s_h \text{ for } h > k - t\},$$

где k — индекс итерации, s^{-1} — перемещение, обратное перемещению s и s_h — перемещение на итерации h .

Таким образом, список запретов не позволяет изменять размещение тех геометрических объектов, которые перемещались на последних t итерациях процесса.

Целевая функция. В качестве целевой функции C используется длина занятой части полосы.

Годограф-ориентированная адаптация алгоритма TS, может быть представлена в следующем виде.

Алгоритм НО-TS

Переместить в ПС ГО P_i согласно случайному перемещению $s : s \notin T$

Произвести генерацию карты Р-У K_{new} согласно ПС

if $C(K_{old}) > C(K_{new})$ then

{

$K_{old} = K_{new}$

$T = T \cup \{s\}$

};

В случае более сложной реализации НО-TS, в качестве дополнительного списка запретов используются допустимые размещения объекта P_i рядом с объектом P_j . Эти размещения определяются матрицей $U = \|u_{ij}\|_{n \times n}$, каждый элемент которой u_{ij} характеризует качество размещения объекта P_i с объектом P_j . Величины u_{ij} могут вычисляться при помощи метода оценок [7]. Они вычисляются и для тех вновь генерируемых карт Р-У, которые имеют качество хуже, чем лучшее решение, полученное на данный момент времени. Таким образом, такой способ формирования списка запретов позволяет не только не

возвращаться к уже «пройденным» решениям (запоминание положительного опыта), но и накапливать информацию о взаимном размещении объектов (как положительный, так и отрицательный опыт).

2.2. Задача построения пути режущего инструмента

Метод GRASP (Greedy Randomized Adaptive Search Procedure)

Применение метода GRASP (Жадные рандомизированные адаптивные процедуры поиска) [17] для задачи построения пути режущего инструмента обусловлено хорошими результатами, полученными зарубежными учеными при решении различных задач САПР и АСТПП, простотой метода и возможностью получать качественные результаты за приемлемое время для исходных данных, характерных для реальных практических задач.

GRASP является итеративным процессом, в котором каждая итерация состоит из двух стадий: стадии конструкции, в которой строится возможное решение; и стадия локального поиска, в которой ищется локальный оптимум в окрестностях построенного решения. Лучшее решение сохраняется как результат. Псевдокод, расположенный ниже, поясняет процедуру GRASP для процесса минимизации, в которой выполняется n итераций GRASP.

Процедура GRASP ($f(\cdot), g(\cdot), n, x^*$)

$x^* = \infty$;

цикл для $k = 1, 2, \dots, n$ делать

construct ($g(\cdot), \alpha, x$);

local ($f(\cdot), x$);

если $f(x) < f(x^*)$, то

$x^* = x$;

конец если

конец цикла для

конец GRASP.

На стадии конструкции возможное решение строится итерационно, по одному элементу. Эта стадия подобна полужадной эвристической процедуре, предложенной независимо Хартом и Шоганом [18]. На каждой итерации конструкции выбор следующего элемента для добавления определяется порядком всех элементов - кандидатов (то есть тех, которые могут быть добавлены к решению) в списке кандидатов C относительно жадной функции $g : C \rightarrow R$. Эта функция измеряет полезность выбора каждого элемента. Эвристи-

ка адаптивна, потому что полезность каждого элемента модифицируется на каждой итерации стадии конструкции, чтобы отразить изменения, вызванные выбором предыдущего элемента. Вероятностный компонент GRASP характеризуется случайным выбором одного из лучших кандидатов в списке. Список лучших кандидатов назван ограниченным списком кандидатов (*RCL* — *Restricted Candidate List*). Эта методика выбора учитывает различные решения, которые будут получены на каждой итерации GRASP. Допустим $\alpha \in [0, 1]$ некоторый параметр. Рассмотрим основную стадию конструкции GRASP.

Процедура *construct* ($g(\cdot), \alpha, x$)

$x = 0$;

Инициализация множества
кандидатов C ;

цикл пока $C \neq \emptyset$ **делать**

$s_- = \min\{g(t) | t \in C\}$;

$s^+ = \max\{g(t) | t \in C\}$;

$RCL = \{s \in C | g(s) \leq s_- + \alpha(s^+ - s_-)\}$;

Выбрать s случайно из RCL ;

$x = x \cup \{s\}$

Модифицировать множество
кандидатов C ;

конец цикла пока;

конец *construct*.

Параметр α контролирует «жадность» и «случайность» в алгоритме. Так значение $\alpha = 0$ соответствует «жадной» процедуре конструкции, в то время как $\alpha = 1$ соответствует случайной конструкции.

Как и для многих детерминированных методов, решения, сгенерированные GRASP-конструкцией, не гарантируют локальную оптимальность относительно полученной окрестности, соответствующей найденному решению. Следовательно, имеет смысл применить локальный поиск, чтобы улучшить каждое построенное решение. Алгоритм локального поиска работает в итеративном режиме, последовательно заменяя текущее решение лучшими решениями в окрестности текущего решения. Процесс прекращается, когда не найдено лучшего решения в окрестности. Структура окрестностей N для задачи P связывает решение задачи s с подмножеством решений $N(s)$. Решение s называют локально оптимальным, если не имеется никакого лучшего решения в $N(s)$. Эффективность алгоритма локального поиска зависит от выбора структуры окрестности, методов поиска в окрестности и начального решения.

Приведем базовую процедуру локального поиска.

Процедура *local* ($f(\cdot), x$)

$H = \{y \in N(x) | f(y) < f(x)\}$;

цикл пока $|H| > 0$ **делать**

Выбрать $x \in H$;

$H = \{y \in N(x) | f(y) < f(x)\}$;

конец цикла пока;

конец *local*.

Для формирования рационального ПРИ, метод GRASP был модифицирован следующим образом: базовая процедура запускается для всех значений параметра α из интервала $[0, 1]$ с некоторым шагом для определения такого значения α , при котором суммарная длина переходов была минимальна. Далее для этого значения выполняется алгоритм бинарного поиска, т. е. производится уточнение значения α . Если на некотором шаге деления изменение суммарной длины переходов незначительно, то алгоритм прекращает свою работу, а текущее значение длины переходов считается искомым.

3. СТРУКТУРА

АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ РАЗМЕЩЕНИЯ ГЕОМЕТРИЧЕСКИХ ОБЪЕКТОВ

Процесс решения задач, связанных с проблемой упаковки, на внешнем уровне можно представить следующим образом (рис. 1). На вход системы подается информация:

- геометрические параметры раскраиваемых (упаковываемых) объектов;
- геометрические параметры области размещения;
- количество размещаемых объектов;
- количество областей размещения;
- вид оборудования.

Управляющими параметрами являются:

- технологические ограничения;
- параметры делового остатка.

Исполнителями могут являться:

- инженеры отдела САПР;
- инженеры технологического отдела (ОГТ);
- инженеры отдела АСУП.

На выход системы должна выдаваться следующая информация:

- геометрические параметры карт раскроя-упаковки;
- подетальные нормы расхода материала;
- коэффициенты раскроя и использования материала;
- программы для станков с числовым программным управлением (ЧПУ) и т. д.

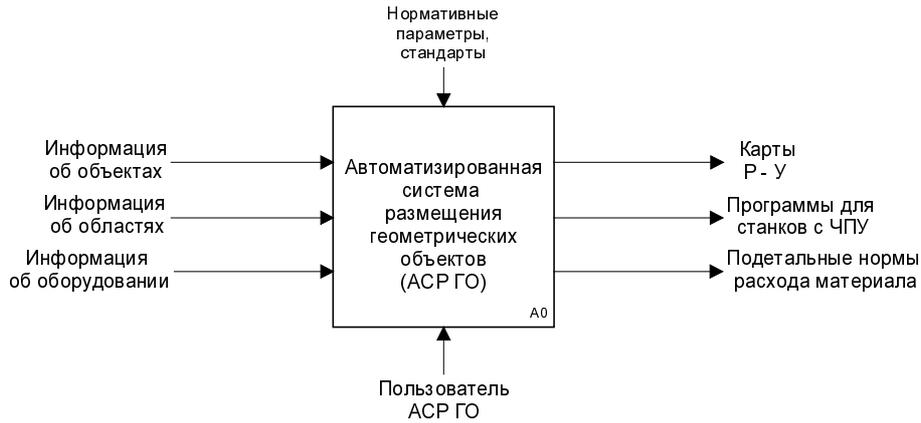


Рис. 1. Представление АСР ГО в методологии IDEF0

Внешние взаимодействия системы определяются спецификой видов информации, входящей в систему (исходной) и исходящей из нее (результурующей).

Внутреннюю организацию предложено представить в виде двух функциональных блоков: оптимизации и обработки данных о геометрии. Это связано с тем, что в задаче, кроме оптимизационной, имеется и проблема, связанная с необходимостью учета того фактора, что в результирующих картах Р-У должны выполняться условия непересечения размещаемых объектов между собой и с границей области, несущих в себе геометрическую сложность. Таким образом, структура АСР представлена на рис. 2.

Для реализации двух составных частей интерфейса раскроя-упаковки (*cutting-packing interface* — *CPI*): входной (*CPI-Input*) и выходной (*CPI-Output*) необходимо рассмотреть более подробно взаимодействие АСР с различными подсистемами, входящими в состав ИСУ.

Для отделения блока оптимизации от блока обработки данных о геометрии введен интерфейс математического программирования (*MPI*).

В связи с тем, что базовым для решения задач размещения является описание геометрии об объектах и области упаковки в некотором формате, то после получения исходной геометрической информации ее необходимо преобразовать к нужному виду.

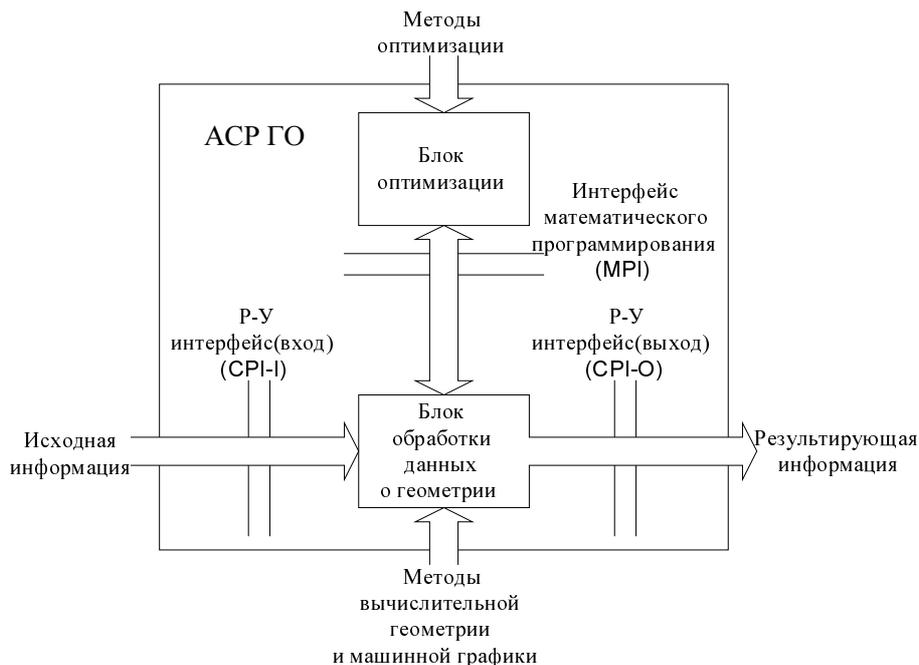


Рис. 2. Структура и интерфейсы АСР ГО

Этот этап присущ любой АСР ГО, так как данные из систем автоматизации проектирования (например, форматы DXF или IGES) требуется преобразовать к внутреннему представлению информации той или иной АСР ГО.

Это преобразование соответствует составляющей «вход» интерфейса раскрыя-упаковки (СРІ-I).

Затем представленные в нужном формате данные вместе с информацией о количестве объектов и областей, а также технологически ограничениями передаются на вход подсистемы, осуществляющей генерирование рациональных карт.

На последнем этапе из информации о картах Р-У получают те данные, которые необходимы различным АС в составе ИСУ:

- карты Р-У для отдела САПР;
- программы для станков с ЧПУ для отделов ОГТ и ОГС (АСТПП);
- подетальные нормы расхода материала, коэффициенты раскрыя-упаковки и использования материала для отдела АСУП.

Это преобразование соответствует составляющей «выход» интерфейса раскрыя-упаковки (СРІ-O).

Таким образом, наиболее целесообразным разбиением автоматизированных систем размещения ГО [19] является выделение следующих подсистем:

- препроцессорный блок;
- расчетное (оптимизирующее) ядро;
- постпроцессорный блок.

Анализ функций перечисленных подсистем приводит к выводу, что такое разбиение делает зависящим от вида конкретной АС только реализацию последнего — постпроцессорного блока.

3.1. Препроцессорный блок АСР ГО

Наполнение препроцессорного блока зависит от внутреннего представления данных в АСР ГО, которое используется для моделирования условий взаимного непересечения. Если, например, учесть, что наиболее широкое распространение в задачах Р-У получило представление объектов и областей в виде многоугольников (многогранников), то тогда его можно описать в виде последовательного выполнения следующих функций:

- выборка и преобразование данных об объектах и областях раскрыя-упаковки из файлов, содержащих информацию в одном из форматов обмена данными между различны-

ми АС в формат, хранящий информацию о геометрии в виде совокупности отрезков прямых и дуг окружностей;

- аппроксимация многоугольниками (многогранниками) объектов и областей, полученных от предыдущего шага.

Выходной информацией этого блока является совокупность разных представлений данных об объектах и областях Р-У, необходимая для моделирования, как раскрыя-упаковки, так и визуализации получаемых карт Р-У.

3.2. Оптимизационное ядро АСР ГО

Оптимизационный блок является ядром всей АСР ГО. Необходимо осуществить его декомпозицию таким образом, чтобы имелась возможность замены его отдельных составляющих без затрагивания остальных его частей. Это является актуальным, как в связи с тем, что существует достаточно большое количество различных методов решения задач размещения ГО, так и с возможным появлением новых методов в дальнейшем.

Исходя из того, что данный класс задач относится к **оптимизационным задачам геометрического моделирования**, то можно выделить, как уже было сказано выше, две основных составляющих этого блока (рис. 3): оптимизационную; геометрическую.

В оптимизационную часть входят реализации различных методов математического программирования, используемых для нахождения оптимальных решений или близких к ним допустимых решений.

Геометрическая часть предназначена для моделирования условий взаимного непересечения ГО между собой и с областью Р-У, она включает в себя реализации различных методов вычислительной геометрии и машинной графики. Между этими блоками связь должна осуществляться согласно интерфейсу математического программирования (*mathematical programming interface* — MPI), который делает независимым использование различных методов математического программирования от специфики описания и обработки геометрических данных.

Реализация MPI зависит от класса методов математического программирования, которые будут использоваться в конкретной АСР ГО, так как в рамках разных классов методов (линейное программирование, нелинейное программирование, дискретная оптимизация) его реальное наполнение значительно отличается одно от другого.

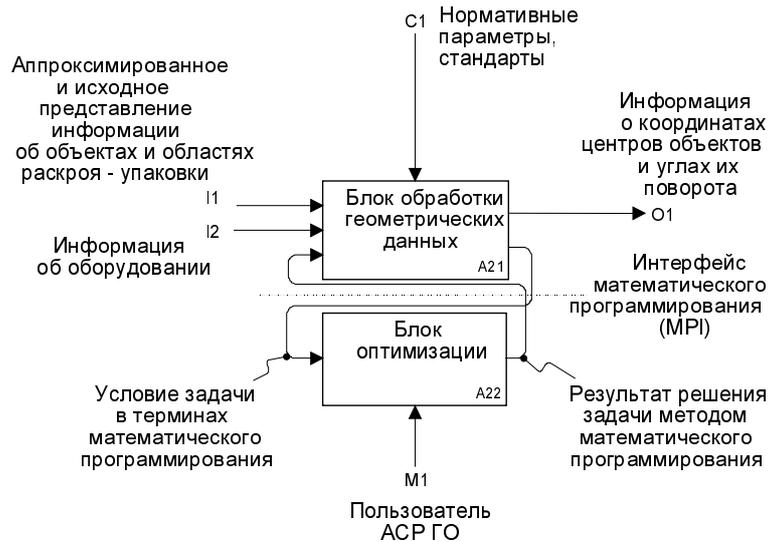


Рис. 3. Декомпозиция оптимизационного ядра АСР ГО

Так, в [20] подробно рассмотрена декомпозиция оптимизационного ядра для случая использования методов дискретной оптимизации, а также соответствующий этому случаю интерфейс DOI (*discrete optimization interface*).

3.3. Постпроцессорный блок АСР ГО

Постпроцессорный блок является интерфейсом, связывающим оптимизационное ядро с различными АС в составе ИСУ.

В связи с тем, что на выходе оптимизационного ядра информацию о картах Р-У наиболее удобно хранить в виде двух наборов данных:

- (первый набор данных) координаты центров размещаемых объектов в системе координат области размещения и соответствующие углы разворота объектов при расположении в этой области;
- (второй набор данных) геометрические описания объектов и областей размещения; то на начальном этапе эта информация объединяется в единый набор данных, хранящий полную информацию о картах Р-У.

На следующих этапах — результирующая информация преобразуется в зависимости от вида той АС ИСУ, в которую она будет передаваться, к соответствующему типу данных (это соответствует интерфейсу СРІ-О).

Так, для отдела САПР информация будет перекодироваться в формат обмена данными между различными АС (IGES, DXF).

Для отделов, связанных с технологией изготовления (ОГТ, ОГС), должны выдаваться соответствующие программы для станков с ЧПУ.

Для отделов, связанных с планированием материалов и оборудования, выводится информация о подетальных нормах расхода материала, коэффициентах раскроя - упаковки, использования материала и т. д.

4. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТА

Тестовый пример, при помощи которого проводилось сравнение, взят из поставки демо-версии системы «AccuFab» — одного из пакетов, распространенных в Северной Америке, представляет собой набор из 14 деталей, различной формы и количества. Детали импортируются из *dxg*-файлов — формата известной САПР «AutoCad».

Количество разрешенных углов поворота деталей зафиксировано и равно четырем: 0, 90, 180 и 270 градусов. Размещение деталей производится на прямоугольных листах длиной 96 и высотой 48 единиц.

Основные параметры размещения: расстояние между деталями на листе не менее 0,25, между деталями и краями листа — 0,5 единицы.

Сравнение проводилось по следующим параметрам: коэффициенты раскроя по каждому листу, длина делового остатка, длина холостого хода режущего инструмента. Время расчета карт раскроя и пути режущего инструмента измеряются от нескольких десятков до нескольких секунд соответственно. Так как расчет примера производился для разных систем на компьютерах различной конфигурации, то приводить точное время (ориентировочно — не более нескольких минут) автор не посчитал необходимым.

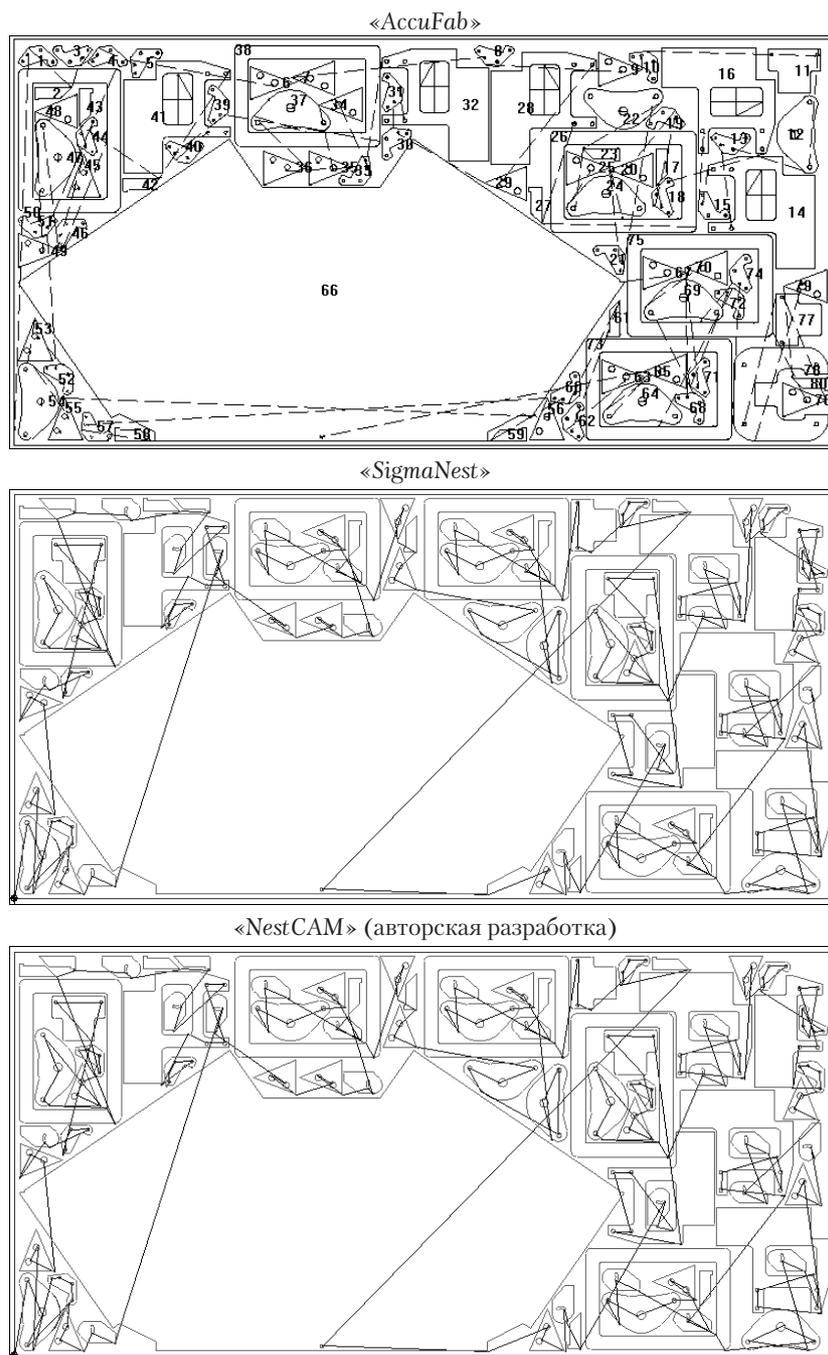


Рис. 4. Примеры карт раскроя, полученные с использованием различных систем

Вычисления производились на машине Celeron 700 МГц, 256 Мб ОЗУ, работающей под управлением Windows 2000. Результаты сравнения приведены в таблице.

Примеры первой карты раскроя с изображенным путем режущего инструмента, полученные для данного тестового примера при помощи систем «AccuFab», «SigmaNest» и авторской разработки, приведены на рис. 4.

Таблица

Сравнение результатов вычислительного эксперимента

Наименование системы	КРА, % по листам			Длина делового остатка, единиц	Длина холостого хода режущего инструмента, единиц
	73,1	60,7	56,2		
AccuFab (США)	73,1	60,7	56,2	6,8	3709
Expert (Испания)	73,4	60,9	59,0	12,2	3621
SigmaNest (США)	74,3	65,9	60,3	20,2	3658
NestCAM (Россия)	74,6	67,2	61,5	21,8	3515

ЗАКЛЮЧЕНИЕ

В современных условиях, когда все более актуальной становится проблема экономии ресурсов, ее решение происходит по всем возможным направлениям науки, техники и производства. К одному из них относится задача нерегулярного раскроя фигурных заготовок из листового материала. Она сочетает в себе как возможность экономии раскраиваемого материала, так и экономию энергии и времени, затрачиваемых на процесс резки. А так как данные задачи относятся к классу NP -трудных, осложненных геометрическими особенностями раскраиваемых объектов, то применение современных метаэвристических методов поиска для их решения является насущной задачей, которая не утратит своей актуальности в ближайшем будущем. С ростом же быстродействия современных вычислительных систем эффективность применения этих подходов будет только возрастать. Соответственно, рассмотренные в вышеизложенной статье методы, алгоритмы и структура автоматизированной системы будут все более востребованы, а в связи с NP -трудностью задачи — претерпевать и дальнейшее совершенствование.

СПИСОК ЛИТЕРАТУРЫ

1. **Стоян, Ю. Г.** Размещение геометрических объектов. Киев : Наук. Думка, 1975.
2. **Aarts, E.** Local Search in Combinatorial Optimization / E. Aarts, J. K. Lenstra (eds.). John Wiley & Sons Ltd, 1997. 315 p.
3. **Holland, J.H.** Adaptation in Natural and Artificial Systems / J. H. Holland. Ann Arbor : The University of Michigan Press, 1975. 96 p.
4. **Goldberg, D.** Genetic Algorithms in Search, Optimization and Machine Learning / D. Goldberg. Addison-Wesley Publ., 1989. 99 p.
5. **Folkenauer, E.** The grouping genetic algorithms for Bin-Packing / E. Folkenauer // JORBEL : Belgian Journal of Operations Research, Statistics and Computer Science. 1995. Vol. 35. P. 64–88.
6. **Dorigo, M.** Ant colonies for the traveling salesman problem / M. Dorigo, L. M. Gambardella // BioSystems. 43. P. 73–81.
7. **Верхотуров, М. А.** Задача нерегулярного раскроя плоских геометрических объектов: моделирование и расчет рационального раскроя / М. А. Верхотуров // Информационные технологии. 2000. № 5. С. 37–42.
8. **Kirkpatrick, S.** Optimization by simulated annealing / S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi // Science. 1983. 220. P. 671–680.
9. **Cerny, V.** Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm / V. Cerny // J. Opt. Theory Appl. 1985. 45. P. 41–51.
10. **Metropolis, N.** Equation of state calculations by fast computing machines / N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller and E. Teller // J. Of Chem. Physics. 1953. 21. P. 1087–1092.
11. **Dueck, G.** New optimization heuristics / G. Dueck // J. Comp. Phys. 1993. 104. P. 86–92.
12. **Glover, F.** Tabu search. P. I / F. Glover // ORSA J. on Computing. 1989. 1 (3). P. 190–206.
13. **Glover, F.** Tabu search. P. II / F. Glover // ORSA J. on Computing. 1990. 2 (1). P. 4–32.
14. **Faigle, U.** Some convergence results for probabilistic tabu search / U. Faigle, W. Kern // ORSA J. on Computing. 1992. 4 (1). P. 32–37.
15. **Glover, F.** Tabu Search in Modern Heuristic Techniques For Combinatorial Problems / F. Glover. Blackwell Publishing, 1992.
16. **Blazewicz, J.** Using a tabu search approach for solving the two-dimensional irregular cutting problem / J. Blazewicz, P. Hawryluk, R. Walkowiak // Annals of OR. 1993. 41 (1–4). P. 313–325.
17. **Feo, T. A.** Greedy randomized adaptive search procedures / T. A. Feo, M. G. Resende // J. of Global Optimization. 1995. 6. P. 109–133.
18. **Hart, J.P.** Semi-greedy heuristics: An empirical study / J. P. Hart, A. W. Shogan // Operations Research Letters. 1987. 6. P. 107–114.
19. **Верхотуров, М. А.** Интегрированная система рационального раскроя в условиях единичного и мелкосерийного производства / М. А. Верхотуров, Э. А. Мухачева // Кузнечно-штамповочное производство. М. :1996. № 5. С. 24–27.
20. **Верхотуров, М. А.** Структура решения задач нерегулярного раскроя-упаковки геометрических объектов / М. А. Верхотуров, Г. Н. Верхотурова, Е. В. Логинов // Моделирование, вычисления, проектирование в условиях неопределенности : межвуз. науч. сб. Уфа : УГАТУ, 2000. С. 375–379.

ОБ АВТОРЕ



Верхотуров Михаил Александрович, проф., каф. выч. матем. и киберн. Дипл. инж.-системотех. по АСУ (УАИ, 1983). Д-р техн. наук по АСУ (УГАТУ, 2000). Иссл. в обл. оптимизации размещения двух- и трехмерн. геометрич. объектов.