

Исследование и применение бионических методов и моделей для автоматизированного проектирования маршрутов обхода геометрических объектов

Г.В. Пушкарёва

Новосибирский государственный технический университет,

Новосибирск, Россия

pushkaryova@sibmail.ru

Аннотация

Полученные результаты исследования позволяют предложить эффективные алгоритмы, основанные на бионических принципах, для решения широкого класса задач дискретно-непрерывной структуры. Проведено системное исследование по направлению интеллектуализации в САПР и разработаны конструктивные методы решения задачи построения гамильтонова контура на множестве геометрических объектов, включая объекты с внутренними контурами. Применение генетических алгоритмов при рассмотрении проблемы автоматизации подготовки управляющих программ для станков тепловой резки металла с ЧПУ позволяет сократить сроки проектирования и улучшить качество проектных решений. Способ решения рассматриваемой задачи основан на интеграции технологий искусственного интеллекта, математического программирования и вычислительно-поисковых процедур.

Ключевые слова: гамильтонов цикл, оптимизационные алгоритмы, маршрутизация, бионические принципы, принятие решений, интеллектуализация в САПР, генетическое программирование

1. ВВЕДЕНИЕ

Генетические алгоритмы, являясь одной из парадигм эволюционных вычислений, представляют собой алгоритмы поиска, построенные на принципах, сходных с принципами естественного отбора и генетики [2]. Они объединяют в себе принцип выживания наиболее перспективных особей-решений и обмен информацией, в котором присутствует элемент случайности, и который моделирует природные процессы наследования и мутации. Дополнительным свойством этих алгоритмов является невмешательство человека в развивающийся процесс поиска. Человек может влиять на него лишь опосредованно, задавая определенные параметры.

Будучи разновидностью вероятностных методов поиска, генетические алгоритмы имеют целью нахождение лучшего, а не оптимального решения задачи. Это связано с тем, что для сложной задачи часто требуется найти хоть какое-нибудь удовлетворительное решение, а проблема достижения оптимума отходит на второй план. При этом другие методы, ориентированные на поиск именно оптимального решения, вследствие чрезвычайной сложности задачи становятся вообще неприменимыми. В этом кроется причина появления, развития и роста популярности генетических алгоритмов.

Хотя, как и всякий другой метод поиска, этот подход не является оптимальным методом решения любых задач.

Известные программные комплексы автоматизированного проектирования управляющих программ в значительной мере используют элементы интерактивной графики, в то время как более перспективным представляется полная автоматизация проектных решений на основе математических моделей и оптимизационных методов.

Траектория движения режущего инструмента состоит из следующих элементов [12]:

- 1) внешних контуров вырезаемых деталей;
- 2) внутренних контуров;
- 3) траекторий, связывающих смежные контуры (вырезаемые с одной врезки);
- 4) траекторий переходов инструмента в выключенном состоянии от одной точки врезки к другой.

При резке тонколистового металла маршруты типа 3, как правило, отсутствуют, поскольку каждая деталь вырезается в этом случае с отдельной врезки. Для толстолистового проката затраты на сквозную пробивку металла оказываются столь значительны, что экономически целесообразным становится обработка деталей без выключения режущего инструмента.

Таким образом, задача состоит в минимизации траекторий активного и холостого хода инструмента, числа врезок и оптимизации переходов от одной врезки к другой. Исходными данными для задачи построения рационального маршрута являются технологические карты раскроя. Конфигурацию плана раскроя образуют внутренние и внешние контуры вырезаемых деталей. Особенность и сложность задачи заключается в том, что она имеет дискретно-непрерывную структуру. При решении данной задачи возникает ряд различных технологических ограничений, связанных со спецификой исполнительного инструмента.

Аналогичные по формальной постановке прикладные задачи могут быть сформулированы при построении рациональных коммуникационных сетей, при решении прикладных задач размещения и обслуживания оборудования, при проектировании сетей связи с кольцевой архитектурой (Token Ring), для автоматизации проектирования технологии обработки деталей, минимизирующей перемещение инструмента с двумя степенями свободы и т.п. [1].

Для таких трудных с вычислительной точки зрения проблем, один из подходов состоит в том, чтобы ослабить требование глобальной оптимальности результата, заменив

исчерпывающий поиск приближенным, что приводит к более эффективным алгоритмам. При этом в большинстве случаев ожидается более чем умеренный проигрыш в качестве полученного результата.

При решении поставленной задачи в основном применяются эвристические методы и комбинированные алгоритмы, так как для её решения на количестве деталей из реальных практических задач не существует точных методов, дающих результат за приемлемое время.

Для нахождения приближённого решения задачи в работе было решено разработать гибридный генетический алгоритм, обладающий устойчивостью к попаданию в точки локальных экстремумов и способностью постоянно увеличивать качество популяции от поколения к поколению. Генетические алгоритмы (ГА) успешно используются при проектировании СБИС в задачах размещения, в задачах двумерной упаковки [4], при решении комбинаторно-логических задач [17], задач канальной трассировки [15,18,19], задачи распределения инвестиций и других экономических задач [8]. Известны научные работы Л.А. Растрюгина по направлению генетических алгоритмов [7].

2. ПОСТАНОВКА ЗАДАЧИ

Имеется множество деталей, состоящих из внутренних и внешних контуров. Каждый контур имеет начальную точку вырезки (x_i, y_i) , принадлежащую i -му контуру ($i=1, 2, \dots, n$). Обозначим расстояние между начальными точками вырезки i -го и j -го контура через L_{ij} ($i, j=0, 1, \dots, n$). Причём, равенство нулю индекса i (или j) означает соответствие началу координатной системы, т. е. точке $(0, 0)$. Необходимо найти кратчайший маршрут k^* из множества K допустимых маршрутов

$$k = (i_1(x_{i_1}, y_{i_1}), i_2(x_{i_2}, y_{i_2}), \dots, i_n(x_{i_n}, y_{i_n})),$$

где (i_1, i_2, \dots, i_n) — произвольная перестановка чисел $1, 2, \dots, n$.

Задача принимает вид

$$F(k^*) = \min (L_{0i_1}(x_{i_1}, y_{i_1}) + \sum_{j=1}^{n-1} L_{i_j i_{j+1}}(x_{i_j}, y_{i_j}, x_{i_{j+1}}, y_{i_{j+1}}) + L_{i_n 0}(x_{i_n}, y_{i_n}))$$

Существует система геометрических и технологических ограничений. В качестве геометрических ограничений рассмотрим контуры вырезаемых деталей — геометрические объекты, являющиеся совокупностями отрезков прямых и дуг окружностей. В системе AutoCAD данные геометрические объекты представляются полилиниями. В частном случае контуры вырезаемых деталей могут быть представлены окружностями.

Контуры вырезаемых деталей можно описать аналитически в виде совокупности параметрических уравнений:

$$x_i = a_i + r_i \cos t_i, \quad y_i = b_i + r_i \sin t_i,$$

где $0 \leq i \leq m-1$ (m — количество окружностей), $0 \leq t_i < 2\pi$, a_i, b_i — координаты центра i -ой окружности, r_i — радиус i -ой окружности;

$$x_{ij} = x_{ij1} + (x_{ij2} - x_{ij1}) t_{ij}, \quad y_{ij} = y_{ij1} + (y_{ij2} - y_{ij1}) t_{ij},$$

где $m \leq i \leq m+n-1$ (m — количество окружностей и n — количество полилиний), $0 \leq j \leq p-1$ (p_i — количество отрезков

в i -ой полилинии), $0 \leq t_{ij} \leq 1$, x_{ij1}, y_{ij1} и x_{ij2}, y_{ij2} — координаты конечных точек j -го отрезка i -ой полилинии;

$$x_{ij} = a_{ij} + (x_{ij1} - a_{ij}) \cos t_{ij} - (y_{ij1} - b_{ij}) \sin t_{ij},$$

$$y_{ij} = b_{ij} + (y_{ij1} - b_{ij}) \cos t_{ij} + (x_{ij1} - a_{ij}) \sin t_{ij},$$

где $m \leq i \leq m+n-1$ (m — количество окружностей и n — количество полилиний), $p_i \leq j \leq p_i + d_i - 1$ (p_i — количество отрезков в i -ой полилинии и d_i — количество дуг в i -ой полилинии), $0 \leq t_{ij} \leq \varphi_{ij}$ при $\varphi_{ij} > 0$ и $\varphi_{ij} \leq t_{ij} \leq 0$ при $\varphi_{ij} < 0$ (φ_{ij} — угол стянутого дугового сегмента и $\varphi_{ij} = 4 \arctg k_{ij}$, где k_{ij} — кривизна этого сегмента), x_{ij1}, y_{ij1} и x_{ij2}, y_{ij2} — координаты концов j -ой дуги i -ой полилинии, a_{ij}, b_{ij} — координаты центра j -ой дуги i -ой полилинии, определяемые следующим образом:

$$a_{ij} = \frac{(x_{ij1} + x_{ij2})(1 - \cos \varphi_{ij}) - (y_{ij2} - y_{ij1}) \sin \varphi_{ij}}{2(1 - \cos \varphi_{ij})},$$

$$b_{ij} = \frac{(y_{ij1} + y_{ij2})(1 - \cos \varphi_{ij}) + (x_{ij2} - x_{ij1}) \sin \varphi_{ij}}{2(1 - \cos \varphi_{ij})}$$

Координаты вершин и кривизна участков полилиний, координаты центров и радиусы окружностей содержатся в подписках с соответствующими кодами в графической базе данных чертежа [5].

Кроме геометрических ограничений, существует ряд технологических ограничений, связанных со спецификой исполнительного инструмента. В частности, в качестве таких ограничений могут являться следующие:

- маршрут начинается и заканчивается в начале координатной системы,
- исключается возможность пересечения резов,
- вырезание некоторых деталей по специальному маршруту из-за термической деформации,
- ограничение возможностей движения оборудования,
- внешний контур детали обрабатывается только после того, как обработаны все ее внутренние контуры.

Для реализации последнего технологического ограничения формируется матрица структуры технологической карты раскроя $V_{n \times n} = (v_{ij})_{n \times n}$, где n — количество контуров деталей ($i, j=0, 1, 2, \dots, n-1$). Элемент матрицы $v_{ij} = 1$, если i -ый контур принадлежит внутренней области j -го контура, иначе $v_{ij} = 0$. Любое решение данной задачи должно соответствовать сформированной матрице $V_{n \times n}$, то есть в случае $v_{ij} = 1$ необходимо, чтобы i -ый контур вырезался ранее j -го контура.

Процесс формирования оптимального проектного решения можно разбить на три стадии:

- генерация порядка обработки контуров деталей технологической карты раскроя с созданием набора переходов инструмента между контурами;
- формирование полного пути режущего инструмента как результирующей суммы переходов между контурами и, собственно, пути по самим обрабатываемым контурам;
- выбор в качестве оптимального проектного решения пути режущего инструмента с минимальной длиной и наименьшим количеством врезок.

3. МЕТОДЫ РЕШЕНИЯ ОПТИМИЗАЦИОННЫХ ЗАДАЧ МАРШРУТИЗАЦИИ

Для приближенного поиска оптимальной топологии обхода плана раскроя могут быть применены различные эвристические методы решения, причем, для каждого из них известна только верхняя оценка качества приближения, но неизвестна какая-либо зависимость качества решения от исходных данных. Наилучшее приближение для конкретных исходных данных (конкретной карты раскроя) может быть найдено путем последовательного применения различных эвристических методов, используя для сравнительной оценки качества приближения длину полученного маршрута. Наиболее часто используются три следующих эвристических алгоритма решения сформулированной задачи [1,11]:

- 1) метод ближайшего соседа (Nearest Neighbor);
- 2) метод включения ближайшего города (Nearest Town);
- 3) метод самого дешевого включения (Most Cheap Inclusion).

В методе ближайшего соседа, пункты обхода плана последовательно включаются в маршрут, причем, каждый очередной включаемый пункт должен быть ближайшим к последнему выбранному пункту среди всех остальных, еще не включенных в состав маршрута.

Метод ближайшего города на каждом шаге алгоритма проводит допустимый маршрут по текущему подмножеству пунктов обхода уже включенных в маршрут, добавляя к нему новый пункт из числа еще не включенных в маршрут, для которого найдется ближайший сосед из числа пунктов уже принадлежащих маршруту обхода плана. Новый пункт включается в маршрут после своего соседа, которому он обязан включением. Звенья маршрута модифицируются так, чтобы расширенный маршрут являлся допустимым.

Метод самого дешевого включения на каждом шаге алгоритма проводит допустимый маршрут по текущему подмножеству пунктов обхода, уже включенных в маршрут, добавляя к нему новый пункт, включение которого между некоторыми смежными пунктами приводит к минимальному увеличению стоимости (длины) маршрута.

Эвристические методы перечислены в порядке улучшения верхней оценки качества приближенного решения и, соответственно, увеличения вычислительной трудоемкости. Алгоритмы этой группы в метрическом пространстве имеют относительную погрешность 2 и время работы $O(n^2)$, где n — число пунктов обхода плана [3].

Трудности получения точного решения задач этого класса обусловлены большой размерностью поискового пространства. Возможность практической реализации исчерпывающего поиска для сложных карт раскроя существенно ограничена имеющимися вычислительными ресурсами. Ограниченность возможностей точных методов решения задач дискретной оптимизации естественным образом приводит к идее разработки комбинированных алгоритмов, в частности, комбинированных алгоритмов ветвей и границ. Метод ветвей и границ широко используется как для нахождения точного, так и для приближенного решения [9].

Практика применения метода ветвей и границ показала, что в случае сложных планов требуется построение и

динамическое размещение в области данных задачи сильно разветвленного дерева поиска, объем которого может превосходить наличные ресурсы памяти. При распространении дерева поиска за границу памяти выполнение алгоритма ветвей и границ должно автоматически блокироваться. Это свидетельствует о том, что оптимальный цикл обхода данного плана не может быть построен из-за недостатка вычислительных ресурсов. В такой ситуации рекомендуется предварительная ϵ -модификация базового алгоритма ветвей и границ для снижения точности оценки нижних границ или минимизация верхней границы с помощью эвристических методов поиска локально-оптимального маршрута.

Кроме различных модификаций метода ветвей и границ, в настоящее время широко применяется метод динамического программирования [13,14]. Оба этих метода, основываясь на общих идеях последовательного анализа вариантов, используют различные подходы для отсева подмножеств, не содержащих оптимальных решений. В методе динамического программирования отсев производится на основании принципа оптимальности. В методе ветвей и границ отсев производится с применением нижних оценок на подмножествах. Представляется естественным совместное применение обеих процедур отсева, т.е. построение комбинированного правила отсева.

Особого внимания заслуживают вероятностные методы. Эти методы, использующие элементы случайности, стали появляться относительно недавно, по мере того как становились очевидными недостатки методов, основанных на математических вычислениях, и перечислительных методов [2].

Следует заметить, что сейчас при решении очень сложных задач основной целью является поиск уже не оптимального, а более «хорошего» решения по сравнению с полученным ранее или заданным в качестве начального. Здесь вероятностные методы получают определенное преимущество перед остальными. Однако даже с такими допущениями непосредственный случайный поиск является малоэффективным. Исследования показали, что внесение в такие методы элементов детерминированности дает значительное улучшение показателей. Одним из типов таких вероятностно-детерминированных методов являются эволюционные вычисления.

Эволюционные вычисления — термин, обычно используемый для общего описания алгоритмов поиска, оптимизации или обучения, основанных на некоторых формализованных принципах естественного эволюционного процесса. Основное преимущество эволюционных вычислений в этой области заключается в возможности решения многомодальных (имеющих несколько локальных экстремумов) задач с большой размерностью за счет сочетания элементов случайности и детерминированности точно так, как это происходит в природной среде.

Детерминированность этих методов заключается в моделировании природных процессов отбора, размножения и наследования, происходящих по строго определенным правилам. Основным правилом при этом является закон эволюции: "выживает сильнейший", который обеспечивает улучшение находимого решения. Другим важным фактором эффективности эволюционных вычислений является моделирование размножения и наследования.

Рассматриваемые варианты решений могут по определенному правилу порождать новые решения, которые будут наследовать лучшие черты своих "предков".

В качестве случайного элемента в методах эволюционных вычислений может использоваться, например, моделирование процесса мутации. В этом случае характеристики того или иного решения могут быть случайно изменены, что приведет к новому направлению в процессе эволюции решений и может ускорить процесс выработки лучшего решения.

История эволюционных вычислений началась с разработки ряда различных независимых моделей эволюционного процесса. Среди этих моделей можно выделить три основные парадигмы: 1)генетические алгоритмы, 2)эволюционные стратегии, 3)эволюционное программирование. Особый интерес представляют генетические алгоритмы.

Основное отличие генетических алгоритмов заключается в представлении любой альтернативы решения в виде строки фиксированной длины, манипуляции с которой производится в отсутствие всякой связи с ее смысловой интерпретацией. То есть в данном случае применяется единое универсальное представление любой задачи.

Парадигму генетических алгоритмов предложил Джон Холланд, опубликовавший в начале 60-х годов ее основные положения. А всеобщее признание она получила после выхода в свет в 1975 году его классического труда "Адаптация в естественных и искусственных системах".

4. ГИБРИДНЫЙ ГЕНЕТИЧЕСКИЙ АЛГОРИТМ

Основная особенность генетического алгоритма состоит в том, что анализируется не одно решение, а некоторое подмножество квазиоптимальных решений, называемых хромосомами и состоящих из генов [6]. Это подмножество носит название "популяция". В данной реализации начальная популяция формируется случайным образом. Для увеличения скорости сходимости генетического алгоритма в неё включается хромосома, описывающая путь по "жадному" алгоритму. Для рассматриваемой задачи хромосома описывает порядок вырезания контуров деталей с указанием координат начала вырезки каждого контура, поэтому имеет дискретно-непрерывную структуру.

Для хромосомы вычисляется целевая функция $F(k)$, называемая эволюционной, где k - маршрут, описываемый хромосомой. Такие функции вычисляют относительный вес каждой хромосомы. В данном случае целевая функция представляет собой длину траектории движения режущего инструмента. Каждый ген в хромосоме состоит из порядкового номера вырезаемого контура и координат начала вырезки этого контура.

Любое решение, закодированное в хромосоме, должно удовлетворять матрице структуры технологической карты раскроя $V_{n \times n} = (v_{ij})_{n \times n}$, где n — количество контуров деталей ($i, j = 0, 1, 2, \dots, n-1$). Поэтому все вновь созданные хромосомы проверяются на допустимость применения в качестве решения.

В рассматриваемом генетическом алгоритме реализована стратегия элитизма, при которой несколько лучших индивидуумов переходят в следующее поколение без изменений. Количество элитных индивидуумов KI определяется по формуле:

$$KI = (1 - SO) * RP,$$

где SO — степень обновления популяции, RP — размер популяции.

После формирования начальной популяции, осуществляется процесс синтеза новых решений (поколений) задачи посредством кроссовера и мутации. Исходными данными для него являются хромосомы текущей популяции. Исследуемая в некоторый момент времени популяция называется текущей. В начале работы алгоритма текущая популяция совпадает с начальной.

Данный генетический алгоритм можно назвать гибридным, так как в нём реализовано целенаправленное изменение хромосом с целью улучшения значений целевой функции $F(k)$. Для этого предлагается ликвидировать имеющиеся пересечения в маршрутах, используя оператор инверсии, а также применить к каждой хромосоме операцию разнообразия. Операция разнообразия вносит некоторые изменения в отдельную хромосому, не меняя порядка вырезаемых контуров. Эти изменения относятся к координатам начальных точек вырезки контуров деталей.

Гибридный генетический алгоритм поддерживает такую архитектуру генетического поиска, при которой целенаправленное изменение применяется ко всем вновь созданным хромосомам.

После скрещивания и мутации размер популяции увеличивается. Однако для последующих преобразований необходимо сократить число хромосом текущей популяции. Такая процедура носит название селекции. В текущей популяции, состоящей из родителей и потомков, производится отбор лучших решений, т.е. хромосом с наилучшим значением fitness-функции (целевой функции). Эта функция показывает, насколько исследуемая хромосома близка к оптимальному решению.

Для текущей популяции повторяются все описанные процедуры. Процесс продолжается до тех пор, пока не будет обработано заданное число поколений. При этом каждая последующая популяция должна быть лучше, чем предыдущая. Решению задачи соответствует хромосома с наилучшим значением fitness-функции.

Таким образом, для гибридного генетического алгоритма выделяется четыре основных этапа: 1)формирование начальной популяции; 2)синтез новых хромосом (операторы скрещивания и мутации); 3)целенаправленное изменение вновь полученных хромосом (операторы инверсии и разнообразия); 4)селекция текущей популяции.

На основе тестовых данных в ходе экспериментальных исследований выбираются параметры гибридного ГА согласно приведённой далее таблице.

Как было указано выше, генетический алгоритм обрабатывает популяцию решений, закодированных в хромосомы. В процессе обработки популяции, к ней последовательно применяются различные генетические операторы, такие как скрещивание, мутация с заданными вероятностями (PS и PM соответственно) и другие операторы. Затем проводится селекция увеличившейся популяции для отбора лучших решений, которые составят следующее поколение, после чего цикл (генерация) повторяется. Число таких циклов называется числом генераций TG .

Параметры гибридного генетического алгоритма

№ п/п	Наименование параметра	Обозначение	Рекомендуемое значение
1	Размер популяции	RP	20-100
2	Число генераций	TG	20-100
3	Вероятность скрещивания	PS	0.7-0.9
4	Вероятность мутации	PM	0.05-0.1
5	Степень обновления популяции	SO	0.95-1.0
6	Количество попыток	KP	3-5
7	Максимальная длина активного хода исполнительного инструмента (в мм)	MD	200-300

Количество поколений, которое требуется для нахождения кратчайшего маршрута, зависит также от начальной генетической информации в первом поколении. Поэтому оно меняется от попытки к попытке. Для получения наилучшего результата работы генетического алгоритма рекомендуется сделать несколько попыток (3-5).

Общую схему реализованного генетического алгоритма можно представить следующим образом:

Шаг 1. Построение матрицы вложенности контуров.

Шаг 2. Применение "жадного" алгоритма для построения маршрута.

Шаг 3. Ввод параметров расчёта.

Шаг 4. Порядковый номер попытки $i=1$ ($i=1, 2, \dots, KP$).

Шаг 5. Формирование начальной популяции i -ой попытки.

Шаг 6. Целенаправленное изменение хромосом начальной популяции.

Шаг 7. Порядковый номер генерации i -ой попытки $j=1$ ($j=1, 2, \dots, TG$).

Шаг 8. Выделение элиты и формирование текущей j -ой популяции.

Шаг 9. Скрещивание хромосом в j -ой популяции.

Шаг 10. Мутация хромосом в j -ой популяции.

Шаг 11. Целенаправленное изменение новых хромосом.

Шаг 12. Добавление новых хромосом к j -ой популяции.

Шаг 13. Селекция в j -ой популяции.

Шаг 14. Добавление элиты к j -ой популяции.

Шаг 15. Переход к следующей популяции: $j=j+1$.

Шаг 16. Если $j \leq TG$, то переход к шагу 8, иначе определение наилучшего маршрута i -ой попытки.

Шаг 17. Переход к следующей попытке: $i=i+1$.

Шаг 18. Если $i \leq KP$, то переход к шагу 5, иначе определение наилучшего маршрута за время работы алгоритма.

Шаг 19. Вывод результирующего маршрута.

Чтобы уменьшить затраты на сквозную пробивку металла, участки траектории движения, не пересекающие контуры

деталей, могут быть пройдены активным ходом исполнительного инструмента.

В каждой популяции хромосомы могут подвергаться действиям различных операторов. При этом происходят процессы, аналогичные действиям, которые случаются в естественной генетике. К основным операторам относят: оператор скрещивания, оператор мутации, оператор инверсии, оператор разнообразия и оператор селекции.

5. ХАРАКТЕРИСТИКА ГЕНЕТИЧЕСКИХ ОПЕРАТОРОВ

Оператор скрещивания представляет собой процесс, в ходе которого выбирают двух подходящих индивидуумов и скрещивают их. При этом потомство получит некоторые черты от обоих родителей. Один индивидуум выбирается с наилучшим значением целевой функции в текущей популяции, а другой индивидуум выбирается из оставшихся хромосом в текущей популяции поочередно. Обмен значениями случайно выбранного гена в хромосомах происходит только в том случае, если вероятность обмена меньше заданной вероятности скрещивания PS . В данной реализации оператор скрещивания является модифицированным, так как в нём предусмотрен механизм, исключающий возникновение "нелегальных" решений. Такими решениями считаются хромосомы, не отвечающие условию задачи.

Общую схему реализованного оператора скрещивания можно представить следующим образом:

Шаг 1. Выбрать хромосому с лучшим значением целевой функции $F(k)$ в текущей популяции, как первого родителя.

Шаг 2. Порядковый номер итерации $i=1$.

Шаг 3. Сгенерировать случайное число rnd , причём $0 \leq rnd < 1$.

Шаг 4. Если $rnd < PS$, то выбрать i -ую хромосому в текущей популяции в качестве второго родителя, применить операцию скрещивания к выбранным хромосомам и запомнить получившихся потомков.

Шаг 5. Переход к следующей итерации: $i=i+1$.

Шаг 6. Если $i < RP$, то перейти к шагу 3.

Оператор мутации также служит для продолжения эволюции. Однако, вместо комбинирования родительских качеств, мутация вносит изменение в одного индивидуума путем случайного изменения одной из хромосом. Мутация очередной хромосомы текущей популяции происходит только в том случае, если вероятность изменения хромосомы меньше заданной вероятности мутации.

В хромосоме, подвергающейся мутации, случайным образом выбирают два гена, которые меняются значениями между собой. Общую схему реализованного оператора мутации можно представить следующим образом:

Шаг 1. Порядковый номер итерации $i=0$.

Шаг 2. Сгенерировать случайное число rnd , причём $0 \leq rnd < 1$.

Шаг 3. Если $rnd < PM$, то применить операцию мутации к i -ой хромосоме текущей популяции и запомнить получившуюся хромосому.

Шаг 4. Переход к следующей итерации: $i = i + 1$.

Шаг 5. Если $i < RP$, то перейти к шагу 2.

Оператор селекции формирует новое поколение из хромосом с лучшими значениями целевой функции $F(k)$. Он уничтожает большую часть популяции и освежает генетический материал, пополняя популяцию большим количеством новых членов. В результате выполнения оператора селекции размер популяции нового поколения вновь становится равным RP .

Оператор инверсии изменяет характер связей между компонентами хромосомы. Он берёт хромосому, случайным образом выбирает в ней две точки разрыва и располагает в обратном порядке элементы, попавшие между точками разрыва [8].

В данном случае оператор инверсии имеет целенаправленный характер и предназначен для ликвидации имеющихся пересечений в маршрутах, описываемых хромосомами. Таким образом, точки разрыва в хромосоме выбираются не случайным образом, а целенаправленно.

Оператор разнообразия также вносит изменения в отдельного индивидуума, но это очень небольшие изменения в каждой хромосоме, а не сильное изменение хромосомы, как происходит при мутации. Они относятся к координатам начальных точек вырезки контуров деталей.

С целью улучшения значения целевой функции каждая хромосома обрабатывается итерационным алгоритмом оператора разнообразия, основанным на методе циклического координатного спуска. Идея метода заключается в том, что его итерация (цикл) состоит из последовательности шагов.

В ходе шага работы итерационного алгоритма смежные участки траектории движения попарно рассматриваются и заменяются отрезком или ломаной линией, минимально возможной длины, с учётом принадлежности начальных точек вырезки соответствующим контурам деталей. Поэтому в процессе реализации каждого шага значения целевой функции $F(k)$ улучшаются, так как минимизируется компонента целевой функции, зависящая от координат начальной точки вырезки соответствующего j -го контура ($j=1, 2, \dots, n$).

Для минимизации количества врезок исполнительного инструмента приоритетным является выбор начальной точки вырезки контура, совпадающей с начальными точками вырезки смежных контуров.

Если изменения целевой функции $F(k)$ для двух последних итераций не превысили заданную точность ε , то итерационный алгоритм оператора разнообразия прекращает свою работу.

6. РЕАЛИЗАЦИЯ И ВНЕДРЕНИЕ РАЗРАБОТАННОГО АЛГОРИТМА

Программное обеспечение гибридного ГА представляет собой FAS-проект для системы AutoCAD2000 (и выше), разработанный в среде Visual LISP, и три DCL-файла для обеспечения интерфейса с пользователем. Приложение системы AutoCAD работает на IBM-совместимых PC.

В разработанном программном обеспечении использовались функции расширения AutoLISP для дополнительных

операций с кривыми (в том числе операций доступа к параметрическому заданию кривых).

Входной информацией для программы является технологическая карта раскроя, содержащаяся в документе с расширением *dwg*. Карта раскроя включает внешние и внутренние контуры деталей и ограничена прямоугольником, который обозначает границу листа материала. Контуров деталей могут быть представлены на чертеже в виде полилиний или, в частном случае, окружностей. Между ними должно быть некоторое расстояние, обусловленное шириной реза.

Для проверки работоспособности программного обеспечения было подготовлено и просчитано двести технологических карт раскроя. Один документ с тестовыми примерами представлен на рисунке 1.

Результатом работы программы является построенный маршрут обхода контуров с минимальной длиной и количеством врезок.

Маршрут содержит переходы между контурами (линии голубого цвета); некоторые контуры могут быть вырезаны с одной врезки и активный ход инструмента между ними обозначен красной линией; синие точки соответствуют точкам врезок в металл и начальным точкам вырезки контуров, которым они принадлежат; а красные точки — это начальные точки вырезки смежных контуров (вырезаемых с одной врезки).

В директории системы AutoCAD формируется отчётный файл «Отчёт.txt», определяющий порядок обхода и координаты начальных точек вырезания контуров деталей (в координатной системе листа раскроя).

Практическое внедрение разработанного программного обеспечения осуществлено в составе программного комплекса «Техтран – фигурный раскрой» на ОАО «Ижорские заводы» и ООО «НИИП-Информатика».

7. ЗАКЛЮЧЕНИЕ

Дальнейшее исследование продолжается с целью повышения эффективности разработанного генетического алгоритма. Два фактора: скорость и устойчивость - и определяют эффективность генетического алгоритма для решения каждой конкретной задачи.

Основным способом повышения скорости работы генетических алгоритмов является распараллеливание [16]. Причем этот процесс можно рассматривать с двух позиций. Распараллеливание может осуществляться на уровне организации работы генетического алгоритма и на уровне его непосредственной реализации на вычислительной машине.

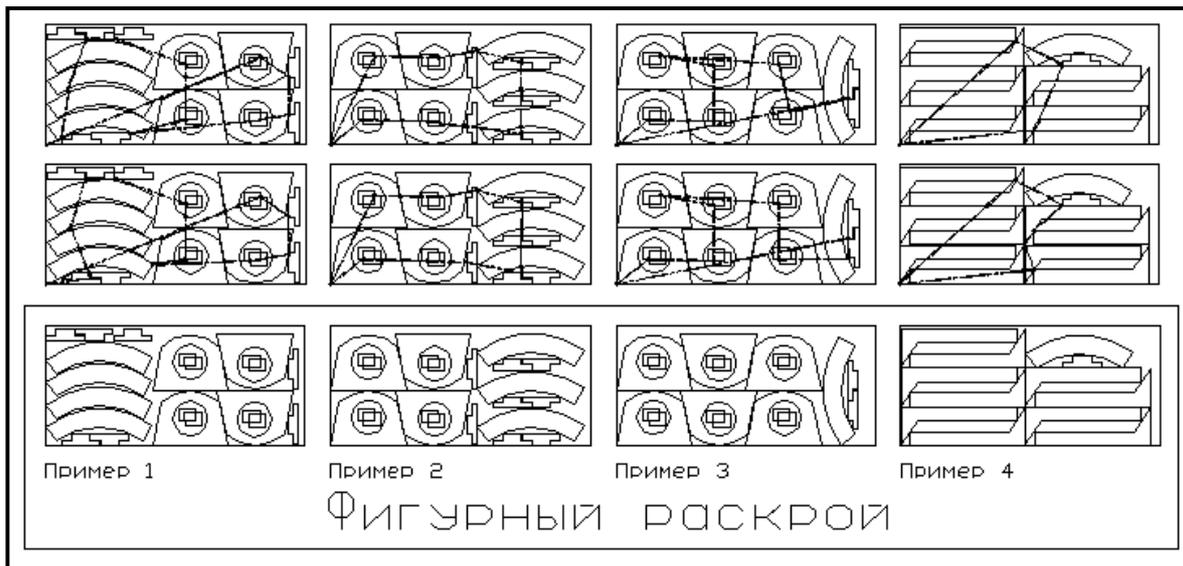


Рис. 1. Тестовые примеры фигурного раскроя с маршрутами обхода

Во втором случае используется следующая особенность генетических алгоритмов. В процессе работы многократно приходится вычислять значения целевой функции для каждой особи, осуществлять преобразования оператора скрещивания и мутации для нескольких пар родителей и т.д. Все эти процессы могут быть реализованы одновременно на нескольких параллельных системах или процессорах, что пропорционально повысит скорость работы алгоритма.

В первом же случае применяется структурирование популяции решений на основе одного из двух подходов:

1. Популяция разделяется на несколько различных подпопуляций (демосов), которые впоследствии развиваются параллельно и независимо. То есть скрещивание происходит только между членами одного демоса. На каком-то этапе работы происходит обмен частью особей между подпопуляциями на основе случайной выборки. И так может продолжаться до завершения работы алгоритма. Данный подход получил название концепции островов.

2. Для каждой особи устанавливается ее пространственное положение в популяции. Скрещивание в процессе работы происходит между ближайшими особями. Такой подход получил название концепции скрещивания в локальной области.

Оба подхода, очевидно, также могут эффективно реализовываться на параллельных вычислительных машинах. Кроме того, практика показала, что структурирование популяции приводит к повышению эффективности генетического алгоритма даже при использовании традиционных вычислительных средств.

Еще одним средством повышения скорости работы является кластеризация [2]. Суть ее состоит, как правило, в двухэтапной работе генетического алгоритма. На первом этапе генетический алгоритм работает традиционным образом с целью получения популяции более "хороших" решений. После завершения работы алгоритма из итоговой популяции выбираются группы наиболее близких решений. Эти группы

в качестве единого целого образуют исходную популяцию для работы генетического алгоритма на втором этапе. Размер такой популяции будет, естественно, значительно меньше, и, соответственно, алгоритм будет далее осуществлять поиск значительно быстрее. Сужения пространства поиска в данном случае не происходит, потому что применяется исключение из рассмотрения только ряда очень похожих особей, существенно не влияющих на получение новых видов решений.

В последнее время в области исследований, направленных на повышение эффективности генетических алгоритмов, большое значение приобрели идеи создания адаптивных генетических алгоритмов, которые могут изменять свои параметры в процессе работы. Они стали продолжением развития идеи поколенческих алгоритмов, которые в процессе работы изменяют размер популяции. Адаптивные алгоритмы способны изменять не только этот параметр, но и суть генетических операторов, вероятность мутации и даже генотип алгоритма. Как правило, данные изменения происходят путем выбора параметров из нескольких вариантов, определенных перед началом работы алгоритма.

Идея адаптивных генетических алгоритмов получила свое воплощение в концепции *nGA*, представляющей многоуровневые генетические алгоритмы. Нижний уровень такого алгоритма непосредственно выполняет задачу улучшения популяции решений. Верхние уровни представляют собой генетические алгоритмы, решающие оптимизационную задачу по улучшению параметров алгоритма нижнего уровня. При этом в качестве целевой функции используется обычно скорость работы алгоритма нижнего уровня и скорость улучшения им популяции от поколения к поколению.

Таким образом, генетические алгоритмы представляют собой одну из важных и активно развивающихся парадигм обширной области алгоритмов поиска оптимальных решений. И в последнее время, с развитием методов компьютерной поддержки принятия решений, они приобретают все большее значение.

8. БИБЛИОГРАФИЯ

- [1] Комплекс программ "Маршрут" (руководство пользователя) / Моск. гос. техн. ун-т им. Н.Э.Баумана, каф. Системы автоматизированного проектирования. - <http://vikt.mega.ru/route.html>.
- [2] Корнеев В.В., Гареев А.Ф., Васютин С.В., Райх В.В. Базы данных. Интеллектуальная обработка информации. – М.: Нолидж, 2000. – 352 с.
- [3] Костюк Ю.Л., Жихарев С.А. Эффективный алгоритм приближённого решения метрической задачи коммивояжера // Дискретный анализ и исследование операций. – Январь-июнь 2000. - Серия 2. - Том 7, №1. – С. 65-74.
- [4] Мухачева А.С. и др. Задачи двумерной упаковки: развитие генетических алгоритмов на базе смешанных процедур локального поиска оптимального решения // Информационные технологии. Приложение. – 2001. №9. – С. 14-24.
- [5] Полещук Н.Н. VisualLISP и секреты адаптации AutoCAD. - СПб.: БХВ-Петербург, 2001. – 576 с.
- [6] Пушкарёва Г.В. Генетическое программирование при автоматизированном проектировании управляющих программ для систем ЧПУ // Сборник научных трудов НГТУ. - 2004. №1. – С. 67-72.
- [7] Растрингин Л.А. Адаптация сложных систем. Методы и приложения. – Рига: Зинатне, 1981. – 394 с.
- [8] Романов В.П. Интеллектуальные информационные системы в экономике: Учебное пособие / Под ред. Н.П. Тихомирова. – М.: Экзамен, 2003. – 496 с.
- [9] Сигал И.Х. Последовательность применения алгоритмов приближённого решения в комбинированном алгоритме решения задачи коммивояжера // Журнал вычислительной математики и математической физики. – 1989. Т.29. №11. – С.1714-1721.
- [10] Сигал И.Х. Приближённые методы и алгоритмы в дискретной оптимизации: Учебное пособие. - М.: МИИТ, 2000. – 107 с.
- [11] Сухарев А.Г., Тимохов А.В., Федоров В.В. Курс методов оптимизации. - М.: Наука, 1986. – 328 с.
- [12] Фроловский В.Д. Моделирование и алгоритмизация процессов геометрического проектирования изделий из листового материала: Диссертация на соискание уч. степени д.т.н. Новосибирск, 2001. - 340с.
- [13] Bard J.F. and Feo N.F. Operations Sequencing in Discrete Parts Manufacturing // Management Science.- 1989.- 35:249-255.
- [14] Bellman R. Dynamic Programming // Princeton University Press, USA, 1957.
- [15] Davidenko V.N., Kureichik V.M., and Miagkikh V.V. Genetic Algorithm for Restrictive Channel Routing Problem // Proc. of the 7th International Conf. on Genetic Algorithms.- M. Kaufmann Publisher.- San Mateo, California, 1997.- P.636-642.
- [16] Goldberg D.E. Genetic Algorithms in Search, Optimization and Machine Learning // Adison Wesley, Reading, MA, 1989.
- [17] Kureichik V.M. et all. Some New Features in Genetic Solution of the Traveling Salesman Problem // Proc. of the Second Intl. Conf. Adaptive Computing in Engineering, Design and Control.- Plymouth, UK, 1996.- P. 294-296.
- [18] Lieng J., Thulasiraman K. A Genetic Algorithm for Channel Routing in VLSI Circuits // Evolutionary Computation.- MIT, 1994, 1(4). – P. 293-311.
- [19] Liu X., Sakamoto A., Shimamoto T. Restrictive Channel Routing with Evolution Programs // Trans. IEICE.- 1993.- Vol. E76-A.- 10:1738-1745.

Об авторе

Пушкарёва Галина Витальевна — к.т.н., старший преподаватель кафедры Автоматизированных систем управления Новосибирского государственного технического университета.

Адрес: Новосибирск, 630092, пр-т К. Маркса, 20, НГТУ.

Телефон: 461559.

E-mail: pushkaryova@sibmail.ru

Research and Application of Bionical Methods and Models for Automated Design of the Round Routes of Geometrical Objects

Abstract

The results obtained allow to offer the effective algorithms based on the bionics principles for solving the wide class of discrete-continuous structure problems. The application of the methodology of genetic programming to a computer-aided design of the control programs for CNC (computer numerical control) machine tools for thermal metal cutting is considered. The way of deciding the considered problem is based on integration of artificial intelligence technologies with methods of mathematical programming and computing - retrieval procedures.

Keywords: Hamilton cycle, optimization algorithms, routing, bionics principles, decision making, intellectualization in CAD, genetic programming

About the author

Galina Pushkaryova is high teacher at Novosibirsk State Technical University, Department of Automated Control Systems. Her contact email is pushkaryova@sibmail.ru.