алгоритмами показало, что при меньшем времени работы новый алгоритм дает более качественные решения

Тестирование разработанных алгоритмов КТ и их сравнение с известными производилось на бенчмарках Ex1, Ex3b, Ex3c, Ex4b, Ex5, трудный пример Дойча, По сравнению с существующими алгоритмами достигнуто улучшение результатов на 2-5%.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Лебедев Б.К.* Интеллектуальные процедуры синтеза топологии СБИС. − Таганрог: Изд-во ТРТУ, 2003.
2. *Лебедев Б.К.* Адаптация в САПР. − Таганрог: Изд-во ТРТУ, 1999.
3. *Лебедев Б.К.* Методы поисковой адаптации в задачах автоматизированного проектирования СБИС. − Таганрог: Изд-во ТРТУ, 2000.

УДК 681.31

**Ю.О. Чернышев, П.Г. Белявский, А.Ю. Полуян**

## ЭВОЛЮЦИОННЫЙ ПОДХОД К РЕШЕНИЮ ЗАДАЧИ О НАЗНАЧЕНИИ ЧЕРЕЗ ОПРЕДЕЛЕИЕ КРАТЧАЙШЕГО ПУТИ

**Введение**. Задача о назначении является частным случаем транспортной задачи [1]. Поэтому ее можно решать, например, методом потенциалов, алгоритмом Литла, а также венгерским методом [2, 3]. Указанные алгоритмы определения оптимального назначения не дают заранее временной оценки решения, а некоторые из них не всегда приводят к оптимальным решениям. При практическом увеличении размеров матрицы ограничений применение этих методов также становится нецелесообразным. С целью сокращения вычислительной сложности решения задачи о назначении и получении глобального оптимального решения предлагается свести задачу о назначении к определению кратчайшего пути в графе.

**Математическая модель задачи о назначении**. Для задачи о назначении решение ее представляет собой перестановку ($p_1$, $p_2$, … $p_n$) чисел i=1,…,n, полученную в результате назначений вида $i \to p_i$. Целью является нахождение на конечном множестве указанных перестановок минимума затрат $\sum_i c_{ip}$, i =1,…,n.

Любую перестановку можно интерпретировать как точку в $n^2$ – мерном евклидовом пространстве. При этом ее удобно представить в виде n x n – матрицы X= $//x_{ij}//$, где $x_{ij}$ = I в случае назначения i-го кандидата на j-ю должность и $x_{ij} = 0$ в противном случае. Чтобы не помешать применению программирования, заменим последнее условие другим: $x_{ij} \geq 0$. Минимум суммарных затрат запишется:

$$\min F=\sum_i \sum C_{ij} x_{ij}, \quad i, j =\overline{1,n}.$$

Тогда целочисленные условия недробления должностей и недопустимости совместительства выразятся:

$$\sum_{j} x_{ij} = 1, \qquad j, i = \overline{1, n},$$

$$\sum_{i} x_{ij} = 1, \qquad i, j = \overline{1, n}. \tag{1}$$

**Сведения задачи о назначении к определению кротчайшего пути.** Тривиальным способом решения задачи назначения являет*ся* перебор всех n! возможных вариантов с оценкой эффективности каждого из них. Если построить граф-дерево назначений, то каждый путь в этом дереве будет однозначно соответствовать одному варианту назначений [4].

В итоге решение задачи о назначении может быть сведено к определению кратчайшего пути на графе. Последний можно найти методом параллельного исследования всех возможных путей. Для этого рассмотрим матрицу C = //Cij// при i, j = $\overline{1, n}$, столбцы которой соответствуют выполняемым работам, а строки − исполнителям. Множество *всех* вариантов назначений представляется в виде графа-дерева назначений, ветви которого Cij означают количественную форму эффективности выполнения j-й работы i-м исполнителем.

Дерево назначений строится следующим образом. Поскольку число работ равно числу исполнителей, то каждый из них должен быть назначен только на одну из работ. Поэтому в первой строке матрицы C нужно отметить всего один элемент Cij (j=$\overline{1, n}$). Но из первой строки можно выбрать любой элемент Cij (j=$\overline{1, n}$), поэтому из начальной вершины $\alpha$ (вершины нулевого уровня) дерева назначений будет исходить n ветвей, которое назовем ветвями первой зоны (рис. 1).

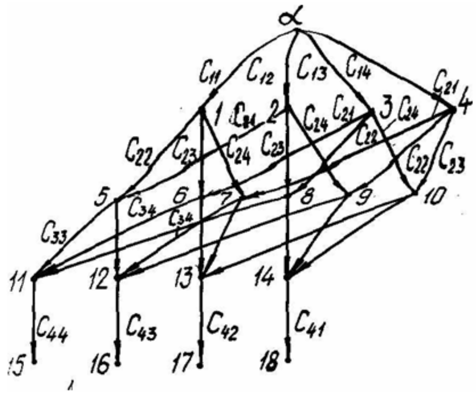


*Рис. 1. Полный граф вариантов решения задачи о назначении*

Под уровнем дерева понимается линия, охватывающая вершины, имеющие равное число исходящих ветвей. Ветви, соединяющие вершины *i*-го уровня с вершинами (*i* +1)-го уровня, назовем ветвями (*i* + 1)-й зоны. Переход с уровня на уровень (*i*+1) соответствует назначению *i*-го исполнителя на одну из j-х работ. Поясним построение ветвей, лежащих во второй зоне. Из каждой вершины первого уровня должна исходить n-1 ветвь. Это объясняется тем, что после рассмотрения первой строки матрицы C переходим к рассмотрению второй строки. Поскольку из n элементов $C_{1j}$ – первой строки уже выбран один, то во второй строке назначение

$C_{2j}$ должно выделяться из (n-1)-го элемента. При построении *i*-й зоны из каждой вершины (*i -1* )-го уровня будет исходить n - *i* ветвей.

Общее количество ветвей, содержащихся в дереве назначений, следующее:

$$m = 2n! + \frac{n!}{(n-k)!}, \quad k = \overline{1, n-2}.$$

Как видно, построение полного дерева назначений связано с большими трудностями. Так, уже для задачи на 7 работ оно будет состоять из 13.699 ветвей. Поэтому большой интерес представляет исследование возможностей сокращения числа ветвей в графе-дереве назначений по аналогии с графом дерева маршрутов для задачи коммивояжера [5].

При анализе полного графа-дерева можно заметить, что, начиная с некоторого уровня, исходящие из него ветви определяют одни и те же варианты последующих назначений. Так, уже на третьем уровне имеются вершины, которые можно объединить по две, на четвертом – по три и т.п., на n-ом − по n-I. После объединения получаем новое унифицированное дерево назначений (рис. 2).

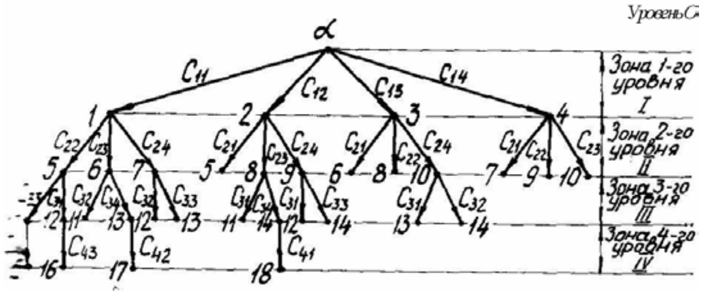


*Рис. 2. Унифицированный граф вариантов решения задачи о назначении*

Если в полном дереве количество ветвей, исходящих из соответствующих уровней, распределяется следующим образом:

$$\begin{aligned} &I - n; \\ &II - n(n-1); \\ &III - n(n-1)(n-2); \\ &IY - n(n-1)(n-2)(n-3); \\ &\ldots \\ &n - n! \end{aligned}$$

то в у инфицированном дереве *с* 3-го уровня будет исходить количество ветвей в 2 раза меньше, чем в полном дереве. Поскольку во 2-й зоне произошло объединение по два, с 4-го уровня − в 2.3 =3! (после объединения во 2-й и 3-й зонах), с 6-го – в 4! и т.д., с n-го уровня − в (n-I)! раз меньше. Количество ветвей в унифицированном дереве распределится в зонах соответствующих уровней следующим образом:

$$\begin{aligned} &I - n; \\ &II - \frac{n(n-1)}{1!} \\ &III - \frac{n(n-1)(n-2)}{2!}; \end{aligned}$$

$$IY - \frac{n(n-1)(n-2)(n-3)}{3!};$$

$$Y - \frac{n(n-1)(n-2)(n-3)(n-4)}{4!};$$

$$\ldots$$

$$n-1 - \frac{n(n-1)(n-2)\ldots4*3*2}{(n-1)} = n.$$

Тогда общее выражение для количества ветвей унифицированного дерева примет вид:

$$m_y = n + \frac{n(n-1)}{1!} + \ldots + \frac{n(n-1)(n-2)\ldots3*2*1}{(n-1)!};$$

$$m_y = n\left[1 + \frac{(n-1)}{1!} + \frac{(n-1)(n-2)}{2!} + \ldots + \frac{(n-1)(n-2)\ldots3*2*1}{(n-1)!}\right]; \quad (2)$$

$$1 + \frac{(n-1)}{1!}X + \ldots + \frac{(n-1)(n-2)\ldots2*1}{(n-1)}X^{n-1} = (1+X)^{n-1}. \quad (3)$$

При X = I в состав выражения (2) входит разложение бинома Ньютона (3). Так как ряд (3) сходится при X = I, то его сумма определяется как

$$\lim_{x\to1}(1+X)^{n-1} = 2^{n-1}.$$

Тогда

$$m_y = n2^{n-1}. \quad (4)$$

Выражением (4) оценивает объем памяти, необходимый для решения задачи назначения n исполнителей на n работ. Для практического решения задачи также важно знать рост количества ветвей в графе при увеличении числа исполнителей и работ на единицу. На основании расчетных данных построены графики (рис. 3): (а) – для унифицированного дереве; (б) – для полного дерева, где по оси абсцисс отложено отношение количества ветвей дерева для *n + 1* – исполнителя к количеству ветвей дерева для n исполнителей, а по оси ординат – число ветвей графа m.

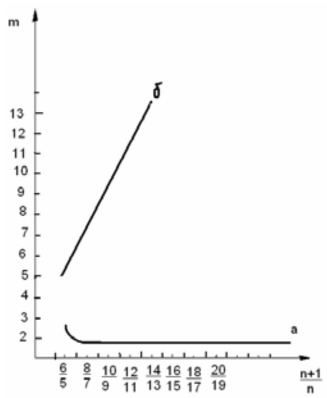


*Рис. 3. График расчетных данных числа ребер графа* m *от числа исполнителей* n *для унифицированного (а) и полного (б) графов вариантов*

Из этих графиков видно, что при анализе n! вариантов на полном дереве при увеличении количества исполнителей и работ в задаче от *n* до *n +1* возрастает количество ветвей m дерева *в n +1* раз, тогда как при анализе того же количества вариантов на унифицированном дереве увеличение количества назначений на единицу вызывает рост количества ветвей в m<3 раз, а при увеличении n,m. Стремимся к двум, т.е.

$$\lim_{n\to\infty}\frac{m_y+1}{m_y}=\lim_{n\to\infty}\frac{(n+1)2^n}{n2^{n-1}}=\lim_{n\to\infty}\frac{n+1}{n}2=2.$$

Таким образом, задача назначения сведена к задаче о кратчайшем пути на графе с числом ветвей $m_y = n2^{n-1}$.

Если при решении задачи на графе найдено несколько равнозначных кротчайших путей, то это свидетельствует о том, что в матрице существует несколько равноценных назначений, для которых выполняется условия из (1).

Данный метод может быть успешно применен и при решение задачи назначения, для которой функция цели должна принять максимальное значение.

При этом критерием оптимальности будет служить не минимум, а максимум, который из (1) запишется:

$$[MAX]F=\sum_{i,j}c_{ij}x_{ij},\ \ \textit{где}\ \ i,j=\overline{1,n}.$$

Для нахождения оптимального назначения такой задачи на графе дерева назначений необходимо выделить длиннейший (критический) путь. Ветви, принадлежащие этому пути, соответствуют оптимальному назначению. При этом соотношения количества ветвей, необходимых для решения задачи, остаются прежним.

Таким образом, используя подход к решению задачи коммивояжера на графе вариантов [5], предложен и математически обоснован новый способ решения задачи о назначении через определение кратчайшего пути, который отличается от известного [1] тем, что в качестве модели используется упрощенный граф вариантов, топологически отображающий матрицу назначений, и позволяющий оценить и сократить объем памяти, необходимый для решения задачи, параллельно формировать оптимальный результат, соответствующий экстремальному пути.

**Структура генетического алгоритма решения задачи о кратчайшем пути в графе.** Предлагается структура генетического алгоритма состоящая из 3-х блоков. В первом реализуется создание одного или некоторого множества начальной популяции на основе различных локальных методов поиска [6].

Второй состоит из четырех этапов:

- выбор представления решения;
- разработка модифицированных генетических операторов: «жадного» оператора кроссинговера (ОК) и оператора мутации (ОМ);
- определение условий выживания решения;
- рекомбинация.

Третий блок реализует принципы эволюционной адаптации к внешней среде лицом принимающим решение (ЛПР) и самоорганизации.

С учетом вышесказанного структуру генетического алгоритма (ГА) для решения задачи о нахождении кратчайшего пути в графе, опишем следующим образом:

*Блок I:*

1. Создание начальной популяции решений задачи (t=0).
2. Определение значений целевой функции для каждой хромосомы и популяции.
3. Реализация операторов репродукции, то есть отбор хромосом (альтернативных решений) для выполнения генераций алгоритма.
4. Реализация модифицированных генетических операторов.
5. Если заданный критерий достигнут, то переход на шаг 10, если нет, то на шаг 6.
6. Рекомбинация родителей и потомков для создания новой генерации.
7. Редукция, т.е. приведение размера популяции к заданному виду.
8. Получена новая популяция альтернативных решений, переход к следующей генерации (t+1=0).

*Блок II:*

9. Реализация новой генерации.
10. Конец работы алгоритма.

Отметим, что в п. 4 подразумевается реализация большого числа внутренних итераций генетических операторов. Модифицированные генетические операторы инверсии и ОМ оказывают большое влияние на получение эффективных результатов. Отметим, что отличительной особенностью рассмотренного генетического алгоритма является способность хорошо работать на популяции с малым числом хромосом, что уменьшает сложность и временную сложность алгоритма.

На селектированные индивидуальности действуют операторы кроссинговера и мутации с вероятностью $p_к$ и $p_м$ соответственно. Затем вновь полученные решения оцениваются, сортируются и строится новая популяция. Заметим, что размер популяции в нашем алгоритме поддерживается постоянным. Критерий остановки алгоритма может быть разным. Можно задать фиксированное число генераций. Можно на каждой итерации полученное решение сравнивать с оптимальным, пока не будет достигнута желаемая точность.

На рис. 4 показан график зависимости ЦФ от числа итераций при P(ОК)=0.85, P(ОМ)=0.2 кол-во вершин графа =100, где: ПГА – простой генетический алгоритм; ГА – генетический алгоритм; мод. ГА – модифицированный ГА.

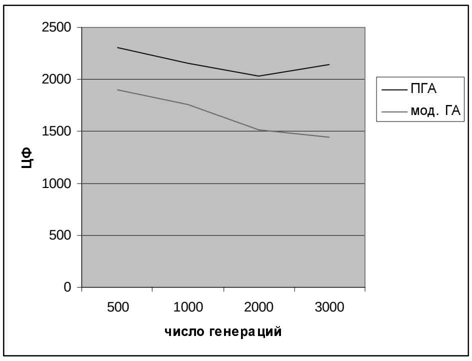


*Рис. 4. График зависимости ЦФ от числа итераций*

**Заключение.** Проведенные вычислительные эксперименты позволили уточнить целесообразность решения задачи о кратчайшем пути в графе. Временная сложность предложенного генетического алгоритма о кратчайшем пути оценивается как 0 ($n^2$), что позволяет сократить время решения задачи о назначении.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Вагнер Г.* Основы исследования операций. − М.: Мир, 1972, т. 1. − 335 с.
2. *Гольштейн Е.Г., Юдин Д.Б.* Задачи линейного программирования транспортного типа. – М.: Наука, 1969. − 382 с.
3. *Кофман А.* Введение в прикладную комбинаторику. − М.: Наука, 1975. − 479 с.
4. *Чернышев Ю.О.* Электронное моделирование задачи о назначении // Однородные цифровые и интегрирующие структуры. − Таганрог: Изд-во ТРТИ, 1977, вып. 8. − С. 99-103.
5. *Чернышев Ю.О., Насекин В.А.*Сведение задачи выбора максимальных интервалов булевой функции к нахождению кратчайшего пути // Известия ВУЗов. Электротехника. − 1974, №3. − С. 235-238.
6. *Чернышев Ю.О., Басова А.В., Полуян А.Ю.* Решение задач транспортного типа генетическими алгоритмами. − Ростов-на-Дону: Изд-во ЮФУГОУ, 2008. − 87 с.

УДК 004.896

**А.Н. Берёза, А.С. Стороженко**

## КОМБИНИРОВАННЫЙ МНОГОПОПУЛЯЦИОННЫЙ МУРАВЬИННЫЙ ГЕНЕТИЧЕСКИЙ АЛГОРИТМ*

**Введение.** Комплексная интеллектуализация систем автоматизированного проектирования в электронике (САПР-Э или ECAD – Electronic Computer Aided Design) предполагает разработку интеллектуальных процедур для всех этапов маршрута проектирования. Одной из таких процедур является оптимизация схемотехнических решений на транзисторном уровне (transistor level). Цель оптимизации заключается в нахождении такой точки пространства допустимых решений, в которой критерии оптимальности принимают экстремальные значения или имеют желаемые характеристики. Создание интеллектуальных процедур параметрической оптимизации позволит повысить качество схемотехнических решений и значительно сократить время их проектирования.

В настоящее время созданы алгоритмы и описаны методы для решения различных оптимизационных задач, как для конкретных случаев, так и в общем виде. Эффективным методом решения оптимизационных задач являются генетические алгоритмы (ГА), основоположником которых считается Джон Холланд (John Holland), описавший ключевые принципы ГА в своей книге «Адаптация в естественных и искусственных системах» (Adaptation in Natural and Artificial Systems).

ГА моделируют процесс эволюции живой природы и на основе эволюционных принципов осуществляют поиск лучших решений [1, 2]. Основным недостатком этих алгоритмов является быстрая сходимость к субоптимальному решению и слабая эффективность поиска при небольшом разнообразии генетического материала. Для обеспечения выхода из локального оптимума применяются различные эвристики [1, 3, 4], одной из которых является *многопопуляционные ГА* (МГА).

Принцип функционирования МГА заключается в создании определенного количества популяций, развивающихся до определенного времени самостоятельно.

* Работа выполнена при поддержке РФФИ (гранты № 07-01-00174, № 08-01-00473).