

6. Проблемы разработки перспективных микроэлектронных систем. Сборник научных трудов под ред. А. Стемповского. – М.: ИППМРАН, 2005. – 537 с.
7. Емельянов В.В., Курейчик В.М., Курейчик В.В. Теория и практика эволюционного моделирования. – М.: Физматлит, 2003. – 431 с.
8. Гладков Л.А., Курейчик В.М., Курейчик В.В. Генетические алгоритмы. Учебное пособие. Под ред. В.М. Курейчика. – Ростов-на-Дону: ООО «Ростиздат», 2004. – 400 с.
9. Харари Ф. Теория графов. – М.: Мир, 1973. – 300 с.
10. Зыков А.А. Теория конечных графов. – Новосибирск: Изд-во Наука, Сибирское отделение, 1969. – 543 с.
11. Zebulum R.S., Evolutionary Electronics: Automatic Design of Electronic Circuits and Systems by Genetic Algorithms (CRC Press LLC, New York, 2002).

Б. К. Лебедев, А. Н. Дуккардт

### ГЕНЕТИЧЕСКИЙ АЛГОРИТМ ДЛЯ ЗАДАЧИ РАЗБИЕНИЯ С АНТИМОНОПОЛЬНОЙ ПОЛИТИКОЙ РАЗВИТИЯ ОСОБИ ДЛЯ МИНИМИЗАЦИИ ЧИСЛА МЕЖУЗЛОВЫХ СОЕДИНЕНИЙ И ВРЕМЕННЫХ ЗАДЕРЖЕК\*

**Введение.** В последнее десятилетие появилось много работ по решению задачи разбиения с учетом временных задержек [2,7,8,10]. В большинстве предыдущих подходов минимизация задержек осуществлялась путем внесения изменений в списки цепей с целью удовлетворения временных ограничений. При этом решалась задача минимизации числа разрезов цепей. В результате проведенного анализа предыдущих подходов к задаче разбиения, можно выделить несколько проблем:

- ◆ Используются нереалистичные модели задержек. В основном используется общепринятая модель задержек, в которой всем вентилям присваивается задержка 1, задержка 0 присваивается соединениям внутри одного узла и постоянная задержка присваивается соединениям между различными частями разбиения [2,7,10].
- ◆ Выполняются нереалистичные упрощения. Например, схемы преобразовываются в вентили с двумя входами [7].
- ◆ Время работы алгоритмов даже для схем умеренного размера слишком велико.

В работе целевая функция изменена таким образом, чтобы она учитывала число разрезов цепей и задержку сигнала. С целью достижения более реалистичной модели задержки модель Эльмора используется вместе с различными задержками проводников для каждой цепи.

**Постановка задачи.** В качестве модели схем при разбиении чаще всего используется гиперграф. На основе гиперграфовой модели задача разбиения формулируется следующим образом: дан гиперграф  $G = (X, E)$ , где  $X = \{x_i \mid i = 1, 2, \dots, n\}$  – множество вершин, а  $E = \{e_j \mid e_j \subset X, j = 1, 2, \dots, m\}$  – множество рёбер (каждое ребро – подмножество связываемых им вершин). Вес вершин задаётся множеством  $\Phi = \{\varphi_i \mid i = 1, 2, \dots, n\}$ , а вес рёбер – множеством  $\Psi = \{\psi_j \mid j = 1, 2, \dots, m\}$ . Необходимо сформировать  $K$  – узлов, т.е. множество  $X$  разбить на  $K$  непустых и непересекающихся подмножеств  $X_k$ :

$$X = \cup X_k \quad (\forall i, j), \quad X_i \cap X_j = \emptyset, \quad X_k \neq \emptyset.$$

\* Работа выполнена при финансовой поддержке программы развития научного потенциала высшей школы РНП.2.1.2.2238

На формируемые узлы накладываются ограничения. С помощью вектора  $\mathbf{N} = \{n_v \mid v = 1, 2, \dots, k\}$  — максимально допустимое число вершин, назначенных в  $v$ -й узел. Ограничения на вместимость имеют вид:

$$|X_v| \leq n_v,$$

где  $v = 1, 2, \dots, k$ ;  $C = \{e_j \mid (\forall v) [e_j \cap X_v \neq \emptyset]\}$  — множество рёбер в разрезе.

Цель классического алгоритма разбиения — это минимизация числа разрезов цепей. Существует множество алгоритмов, которые достаточно эффективно решают эту задачу [9]. Многокритериальная задача разбиения графов гораздо более сложная. В нашем случае мы стремимся получить алгоритм разбиения, который минимизирует величину  $C$  — число разрезов и величину  $D$  — задержку. Обобщенная целевая функция будет вычисляться по формуле:

$$F = p_1 C + p_2 D, \text{ где } p_1, p_2 \in [0, 1], p_1 = 1 - p_2, \quad (1)$$

где  $(p_1, p_2)$  — вектор предпочтения.

Компонент задержки  $D$  выражается комбинацией трех факторов, которые напрямую влияют на задержку схемы: задержка критических путей, количество разрезов каждого критического пути и вес всех рёбер, лежащих на критических путях.

$$D = \alpha \sum_i^{|E_k|} \omega_i e_i + \beta \sum_j^K D_j k_j, \quad (2)$$

где  $\alpha$  и  $\beta$  — весовые параметры;  $\omega_i$  — вес ребра;  $e_i$  — 1 или 0 в зависимости от того, разрезается ребро или нет;  $|E_k|$  — количество рёбер в критических путях  $K$ ;  $D_j$  — текущая задержка  $j$ -ого критического пути;  $k_j$  — сколько раз происходило разрезание критического пути  $j$ .

Предлагаемая модель задержки содержит два компонента. Первый это задержка вентиля. Второй компонент — это задержка проводников, которую мы аппроксимируем, используя модель задержки Эльмора. Для ребра  $e$  (ребро соответствует проводнику) задержка определяется по формуле:

$$D(e) = R_e \left( \frac{C_e}{2} + C_i \right), \quad (3)$$

где  $R_e$  — общее сопротивление проводника  $e$ ;  $C_e$  — общая емкость проводника  $e$ ;  $C_i$  — общая емкость истоков каждой цепи. Для вычисления  $R_e$  и  $C_e$  нам необходимо длина каждого ребра. Средняя длина цепи [11], соединяющая  $m$  элементов, определяется по формуле:

$$L_e = (\alpha * m^\gamma - \beta) \frac{a * b}{a + b} + (a + b), \quad (4)$$

где  $\alpha$ ,  $\beta$ ,  $\gamma$  — подгоночные параметры, вычисленные в [11] как  $\alpha = 1.1$ ,  $\beta = 2.0$ ,  $\gamma = 0.5$ . Сопротивление на единицу длины проводника считаем  $r = 0.115$ , емкость на единицу длины проводника  $c = 0.00015$ .

В предлагаемом подходе задача разбиения гиперграфа  $H = (X, E)$  с взвешенными вершинами и рёбрами сведём к задаче о назначении множества гиперрёбер  $E$  в  $K$  узлов при выполнении условия, что каждое гиперребро может быть назначено только в один узел. Будем считать, что гиперребро  $e_j$  назначено в узел  $z_v$ , если всё множество составляющих его вершин назначено в этот узел.

**Описание алгоритма.** В работе предлагается новый генетический алгоритм (ГА) для разбиения графа на подграфы. Он отличается применением нетрадиционной архитектуры генетического поиска, а так же генетических операторов.

В естественной среде характеристики взрослой особи зависят не только от ее генетического кода, но и от среды в которой она выросла. Если две абсолютно одинаковые по генетическому коду особи (близнецы) поместить в различную сре-

ду обитания, то благодаря адаптации к окружающим условиям, они начнут приобретать различные свойства необходимые для выживания в той или иной среде обитания. Но далеко не факт, что приобретенные свойства одной особи помогут ей выжить в другой среде обитания. Поэтому мы решили положить этот процесс в основу предложенного алгоритма. Мы вводим механизм антимонопольного развития генетического материала (особи), тем самым, имитируя процесс развития особи в различных условиях.

Для реализации механизма антимонопольного развития генетического материала в алгоритм было введено несколько операторов декодирования генетического материала. Причем для большего соответствия естественным процессам была введена случайная величина  $n = \{1,2\}$ , определяющая количество используемых операторов декодирования. На рис.1 приведена структурная схема, отображающая механизм антимонопольного развития особи при  $n = 2$ .



Рис.1. Структурная схема, отображающая механизм антимонопольного развития особи при  $n = 2$

Алгоритм работы блока, реализующего предложенный механизм, имеет вид:  
 1<sup>0</sup>. На вход блока реализующего предложенный механизм подается особь, полученная после кроссинговера или мутации.

2<sup>0</sup>. Производится оценка величины  $n$ .

3<sup>0</sup>. Копирование входящей особи  $n$  раз («Клонирование»).

4<sup>0</sup>. Для каждой копии особи вызывается свой ранее определенный оператор декодирования.

5<sup>0</sup>. После декодирования производится оценка полученных решений.

6<sup>0</sup>. Выбор одного лучшего решения.

В начале работы алгоритма случайным образом создается начальная популяция хромосом фиксированного размера. С помощью механизма антимонопольного развития производится оценка ЦФ для всех хромосом в начальной популяции и фиксируется наилучшая ЦФ. Пользователем задается параметр  $N_m$ . Далее случайным образом, параметру  $n$  присваивается натуральное число в диапазоне  $[2;6]$ . Этот параметр управляет количеством потомков получаемых при ОК. Случайным образом производится выбор родительской пары для осуществления ОК.

Осуществляется реализация оператора кроссинговера (ОК). Все некорректные решения отсеиваются из дальнейшего рассмотрения. Некорректным называется такое решение, в котором генетический материал совпадает с материалом любой другой особи популяции. После оператора кроссинговера корректные решения подвергаются оценке механизмом антимонопольного развития. Если значение ЦФ потомка выше, чем ЦФ одного из родителей, то этот потомок заносится в новую популяцию. В противном случае этот потомок мутирует с некоторой априорно заданной вероятностью  $P_{ом}$ . После ОМ производится переоценка потомка, и, если его ЦФ лучше чем ЦФ одного из родителей, то он заносится в новую популяцию, в противном случае он заносится в новую популяцию с некоторой вероятностью  $P_d$ .

По окончании работы ОК, ОМ формируется новая популяция, в которую входит прежняя популяция и хромосомы, полученные в результате работы ГО. После чего выполняется редукция новой популяции на основе элитной селекции, в результате которой в новую популяцию попадает нужное количество хромосом с «лучшими» ЦФ. Таким образом, завершилась одна итерация ГА.

**Кодирование решения, операции декодирования.** В работе предлагается новый подход, при котором на основе информации в хромосоме узлы формируются путем назначения в них цепей. Назначение цепи в узел фактически сводится к назначению всех связанных ею элементов. Это основное отличие предложенного метода разбиения от всех его аналогов. Назначение цепей в узлы помогает решить сразу несколько проблем:

- ◆ Цепь не разрывается, если все элементы укладываются в узел, что влечет за собой минимизацию внешних связей между узлами.
- ◆ Задержка прохождения сигнала между смежными элементами минимизируется.

В предложенном ГА алгоритме решение представлено в виде двух хромосом. Первая хромосома задает порядок назначения цепей в узлы. Структурное представление хромосомы имеет вид  $H1 = \{g_i | i = 1, 2, \dots, n\}$ , значением гена  $g_i \in H1$  является номер цепи.

Вторая хромосома задает порядок формирования самих узлов. Структура второй хромосомы имеет вид  $H2 = \{h_i | i = 1, 2, \dots, k\}$ , значением гена  $h_i \in H2$  является номер узла. Длина обеих хромосом различна. Длина первой равна количеству цепей, а длина второй – количеству узлов, на которые надо разбить схему.

**Операции декодирования.** В предложенном алгоритме введен блок антимонопольного развития особи. Суть его работы была описана выше. В этот блок входят два независимых оператора декодирования (ОД) особи. Задается управляющий параметр  $n$ . Рассмотрим каждый из них более подробно.

**ОД последовательного назначения ребер в узлы.** Из хромосомы  $H1$  выбирается первое, не назначенное ребро с номером, определяемым значением гена  $h1_i - e_i \in E$ . Из хромосомы  $H2$  выбирается первый, не заполненный узел, номер которого определяется значением гена  $h2_j - k_j \in K$ . Производится проверка возможно ли полностью назначить ребро  $e_i$  в узел  $k_j$  ( $|e_i| \leq n_j$ ). Если да, то ребро назначается в узел, помечается, и из хромосомы  $H1$  выбирается следующее, не назначенное ребро. В противном случае в узел  $k_j$  назначается только часть ребра  $e_i$ , т.е. в выбранный узел назначаются только те вершины, инцидентные выбранному ребру, которые помещаются в него и имеют максимальное количество связей с выбранным узлом. После этого из хромосомы  $H2$  выбирается следующий узел и в него назначается оставшаяся часть выбранного ребра по аналогичной схеме. После назначения вершин в узлы, они удаляются из всех списков инцидентности оставшихся ребер. Это необходимо для того, что бы избежать повторного назначения вершин.

**ОД назначения ребер в узлы, основанный на реберной связности.** Этот ОД отличается от предыдущего только тем, что здесь не учитывается последовательность заданная первой хромосомой особи, т.е. последовательность назначения ребер. Рассмотрим более подробно. Из хромосомы  $H1$  выбирается первое, не назначенное ребро с номером, определяемым значением гена  $h1_i - e_i \in E$ . Из хромосомы  $H2$  выбирается первый, не заполненный узел, номер которого определяется значением гена  $h2_j - k_j \in K$ . Производится проверка – возможно ли полностью назначить ребро  $e_i$  в узел  $k_j$  ( $|e_i| \leq n_j$ ). Если да, то ребро назначается в узел и помечается как назначенное. Далее из хромосомы  $H1$  выбирается следующее, не назначенное ребро  $e_i$ , максимально связанное с  $e_i$ , и процесс назначения повторяется. Если такое ребро не одно, то выбирается такое ребро  $e_i$ , которое будет максимально назначено в текущий узел. Если ребро  $e_i$  не помещается, то в узел  $k_j$  назначается только часть ребра  $e_i$ . Оставшаяся часть ребра назначается в следующий узел, выбранный из хромосомы  $H2$ .

После декодирования проводятся оценки полученных решений и выбирается лучшее.

**Генетические операторы.** Так как решение представлено в виде двух хромосом, поэтому предложенный подход требует разработки новых генетических операторов (кроссинговера, мутации, селекции и др.).

**Оператор кроссинговера.** В работе используется комбинированный оператор кроссинговера [7]. Данный оператор кроссинговера реализуется следующим образом. Сначала на выбранной родительской паре  $R1$  и  $R2$  реализуется обмен генами между гомологичными хромосомами родительской пары с заданной вероятностью  $P_{ок1}$ . После проведения первого кроссинговера образуется дочерняя пара  $R1'$  и  $R2'$ . Далее эта пара рассматривается как родительская и в ней применяется обмен хромосомами с некоторой вероятностью  $P_{ок2}$ . В результате проведения оператора кроссинговера над парой  $R1$  и  $R2$ , образуется дочерняя пара  $R1''$  и  $R2''$ . В результате предложенного оператора кроссинговера может получиться до 6 потомков. Количеством потомков управляет параметр  $n$  предложенного алгоритма (рис.1). Представление решения в виде нескольких хромосом дает возможность использовать оператор комбинирования набором хромосом в одном решении, что является приближением к естественной эволюции

**Оператор мутации.** Задается параметр  $P_m$  – вероятность мутации хромосомы. Просматриваются последовательно все хромосомы решения. С вероятностью  $P_m$  осуществляется мутация выбранной хромосомы по нижеописанной схеме. Производится случайный выбор двух генов в хромосоме, причем, при выборе первого гена фиксируется ген, расположенный справа от него, а при выборе второго гена фиксируется ген, расположенный слева от него. Далее гены один и два меняются местами, а ген, стоящий справа от первого гена, меняется с геном, стоящим слева от второго.

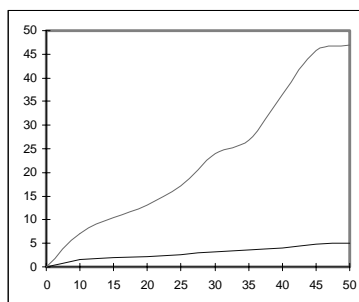
**Экспериментальные исследования.** Тестирование алгоритмов проводилось на IBM совместимом компьютере: CPU AMD Athlon-64 3000+/2x256 DDR DIMM. Основное внимание было уделено тестированию генетических алгоритмов, а именно, ПГА и предложенного алгоритма. В таблице приведены зависимости времени работы ПГА и ГАСМАР от числа вершин и значений целевых функций от числа итераций. При работе алгоритмов, граф разбивался на 3 подграфа, по возможности с равным количеством вершин в каждом узле. При этом число итераций не превышало значение двухсот, что является рекомендуемым значением.

На рис.2,а приведены временные зависимости алгоритмов от числа вершин, а на рис.2,б – значения целевых функций от числа итераций для алгоритмов для графа  $H=(X,E)$ ,  $|X| = 30$ ,  $|E| = 67$ .

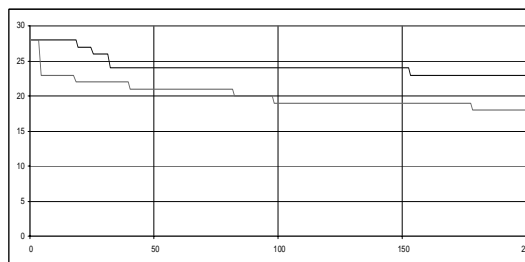
Таблица

**Зависимость времени работы алгоритмов и ЦФ от числа вершин**

Кол-во вершин	ПГА		ГAcMAP	
	Время работы, с	ЦФ	Время работы, с	ЦФ
10	1,7	6	7,05	6
15	2,04	12	10,55	9
20	2,14	17	13,03	12
25	2,64	19	17,15	13
30	3,18	25	23,95	18
35	3,68	31	26,86	22
40	3,96	41	36,45	27
45	4,83	39	45,79	29
50	5,11	46	46,93	29



а



б

*Рис.2. Результаты исследования алгоритмов: а) временные зависимости алгоритмов от числа вершин; б) значения целевых функций от количества итераций (нижний график – ГAcMAP, верхний – ПГА)*

Эксперименты показали, что разработанный алгоритм позволяет получать результаты за приемлемое время. Предложенный алгоритм позволяет получать оптимальные результаты в среднем на 100-130 итерациях в силу его направленного поиска, тогда как ПГА может получить решение на любой итерации совершенно случайным образом. Из таблицы видно, что на большом количестве вершин качество решений получается лучше, чем у ПГА в среднем в 1,7 раза.

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Thang Nguyen Bui, Byung-Ro Moon. "GRCA: A Hybrid Genetic Algorithm for Circuit Ratio-Cut Partitioning". IEEE Transactions on computer-aided design of integrated circuits and systems, vol.17, No.3, March 1998, pp.193-204.
2. Shihliang Ou and Massoud Pedram, "Timing-Driven Bipartitioning with Replication Using Iterative Quadratic Programming", at <http://atrk.usc.edu/~massoud/>, see "Coming Attractions!", 2001.
3. Dominic A. Antonelli, Danny Z. Chen, Timothy J. Dysart, Xiaobo S. Hundrew, B. Kahng Peter, M. Kogge Richard, C., Murphy Michael, T. Niemier, "Quantum-Dot Cellular Automata (QCA) Circuit Partitioning: Problem Modeling and Solutions", DAC, 2004.
4. Букатова И.Л. Эволюционное моделирование: идеи, основы теории, приложения. – М.: Знание, вып. 10, 1981.

5. *Батищев Д.И.* Генетические алгоритмы решения экстремальных задач: Учебное пособие. – Воронеж, 1995. – 69 с.
6. *Дуккардт А.Н.* Методы генетического поиска для мультихромосомных представлений / VII всероссийская научная конференция студентов и аспирантов, Техническая кибернетика, радиоэлектроника и системы управления. – Таганрог: Изд-во ТРТУ, 2004. – С. 108–109.
7. *J. Cong, C. Wu*, “Global Clustering-Based Performance-Driven Circuit Partitioning”, Proc. ISPD, 2002.
8. *W.E. Donath et al*, “Timing Driven Placement Using Complete Path Delays”, Proc. ACM/IEEE DAC, 1990.
9. *G. Karypis, R. Aggarwal, V. Kumar, S. Shekhar*, “Multilevel Hypergraph Partitioning: Application in VLSI domain”, Proc. ACM/IEEE DAC, 1997.
10. *J. Minami, T. Koide, S. Wakabayashi*, “An Iterative Improvement Circuit Partitioning Algorithm under Path Delay Constraints”, IEICE Trans. Fundamentals, Dec. 2000.
11. *P. Zarkesh-Ha, J.A. Davis, J.D. Meindl*, “Prediction of Net-Length Distribution for Global Interconnects in Heterogeneous System-on-a-Chip”, IEEE Trans. VLSI Systems, Dec. 2000.

**Ю.В. Чернухин, А.А. Приемко**

### **АЛГОРИТМ НАВИГАЦИИ АДАПТИВНЫХ МОБИЛЬНЫХ РОБОТОВ ПО АВТОМАТИЧЕСКИ ФОРМИРУЕМОЙ КАРТЕ В УСЛОВИЯХ ДИНАМИЧЕСКИ ИЗМЕНЯЮЩЕЙСЯ ВНЕШНЕЙ СРЕДЫ**

**Введение.** Автоматическое формирование карты внешней среды и навигация по ней в условиях динамических изменений этой среды являются актуальными проблемами современной робототехники. Для их решения необходимо разрешить как минимум две задачи. Первой из них является выработка стратегии исследования среды функционирования робота с одновременным формированием ее карты. Такая стратегия должна обеспечивать полное обследование среды для выявления всех возможных путей движения к цели, так как некоторые из них могут оказаться перекрытыми подвижными препятствиями в процессе навигации. Второй задачей является планирование кратчайшего маршрута движения при помощи созданной карты без столкновений со стационарными и подвижными препятствиями.

Решение этих задач известными методами вызывает значительные трудности, т.к. они ориентированы на применение в условиях статических сред с априори известной конфигурацией элементов [1]. Данное обстоятельство стимулирует разработку новых бионических методов, связанных с имитацией нейросетевым способом алгоритмов решения указанных задач живыми организмами.

С целью решения указанных задач в работе [2] предложен метод формирования и использования карты внешней среды адаптивным мобильным роботом (АМР). При этом предполагается, что АМР имеет в своем составе сенсорную подсистему (СП), содержащую дальнометры для измерения расстояний до объектов среды и видеокамеру для идентификации этих объектов и их классификации по признаку принадлежности к препятствиям либо цели. Определение пути, пройденного АМР, осуществляется с помощью одометрической подсистемы с применением пассивного колеса. Суть метода формирования карты состоит в том, что внешняя среда дискретизируется на участки, соизмеримые с габаритными размерами корпуса робота. Построение карты происходит одновременно с исследованием внешней среды. Участок, на котором в начальный момент времени находился АМР, считается началом координат создаваемой карты. Находясь на начальном участке, АМР определяет координаты соседних участков, связанных с начальным по шаблону четырехсвязности. После этого робот переходит к их исследованию с