

УДК 658.512

И.Н. Ерошенко**РАЗРАБОТКА ГЕНЕТИЧЕСКОГО АЛГОРИТМА КЛАСТЕРНОГО ПЛАНИРОВАНИЯ СБИС***

Рассматривается проблема планирования сверхбольших интегральных схем. Во введении обоснована актуальность задачи. Приведено описание представления плана в виде обобщенной польской записи. Также дано описание структур хромосом, принципов их кодирования/декодирования. Разработаны генетические операторы и схема поиска. Приведен псевдокод генетического алгоритма планирования.

Планирование СБИС; генетический алгоритм; обобщенная польская запись; кластеризация.

I.N. Yeroshenko**THE DEVELOPMENT OF GENETIC ALGORITHM FOR CLUSTERED FLOORPLANNING**

The article is dedicated to the problem of VLSI floorplanning. The description of floorplan in the form of generalized polish expression (GPE) is given. Genetic algorithm for GPE has been chosen as a search engine. The principles of chromosomes coding and decoding, genetic operators, genetic search scheme are proposed in this paper. The principles of clustering are mentioned. The pseudo-code of genetic algorithm is given.

VLSI floorplanning; genetic algorithm; generalized polish expression, clustering.

Введение. Планирование СБИС заключается в размещении на поле кристалла непересекающихся блоков, имеющих форму прямоугольников. Основной целью оптимизации является минимизация общей площади кристалла.

Данная задача является одним из ранних этапов конструкторского проектирования СБИС и относится к классу NP. Планирование влияет на размеры, производительность чипа.

В течение последних лет были предложены различные подходы к решению проблемы планирования. Они отличаются постановками и моделями. В качестве поискового механизма при планировании чаще всего используют метаэвристики, основанные на моделировании отжига (simulated annealing, сам планировщик при этом называется аннилером), и генетические алгоритмы (ГА). Аннилеры работают с одним решением (результат во многом зависит от качества начального решения), а ГА с популяцией решений. Структуры данных, используемые в качестве представлений плана, зачастую сложны для обработки. Поисковые механизмы, задействованные в алгоритмах планирования, не всегда учитывают специфику задачи. Поэтому построение более совершенных моделей является актуальным. В данной статье представлен генетический алгоритм, в основу которого положен механизм многоуровневой эволюции, описанный в работе [1], а кластеризация применяется при входных данных большой размерности.

Итак, при создании планировщика СБИС необходимо учитывать следующие аспекты:

1. Представление плана (обобщенная польская запись использована в качестве структуры данных).
2. Поисковый механизм (генетический поиск взят за основу).

* Работа выполнена при поддержке: РФФИ (гранты № 10-01-00115, № 10-01-90017-Бел_а), г/б № 2.1.2.1652.

Рассмотрим их подробнее.

Представление плана «обобщенная польская запись». В литературе представлены два класса планов: гильотинные и негильотинные.

План гильотинного типа получается путем рекурсивного деления прямоугольника с помощью вертикальных и/или горизонтальных гильотинных разрезов.

Такой план удобно отображать в виде бинарного дерева разрезов, у которого внутренние вершины представляют разрезы, а листья – модули. Дерево кодируется с помощью польской записи (PE – Polish Expression).

В общем случае, негильотинное представление имеет большее пространство решений и позволяет добиться более компактного расположения модулей по сравнению с гильотинным планом. В работе [2] представлена схема кодирования GPE (аббревиатура для Generalized Polish Expression). Она является обобщением для польской записи. GPE позволяет эффективно задействовать некоторую область, которая не может быть использована при наличии операторов вертикального и горизонтального разрезов, определенных в польской записи. Благодаря введению нового оператора, возможно представлять план СБИС с негильотинным разрезом. Помимо реляционных операторов V и H в GPE внедрен третий тип оператора – @ (угловое отношение). Угловой оператор @ поместит модуль или супермодуль в угол, сформированном другими модулями. Запись вида @_(b,c) означает, что нужно поместить текущий модуль/супермодуль в угол справа от b и выше c.

Рассмотрим подробнее схему декодирования GPE. Процедура декодирования GPE показана на рис. 1.

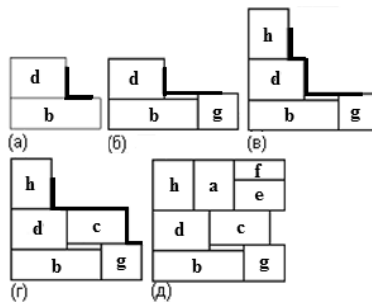


Рис. 1. Формирование плана для $GPE = \{bd + g * h + c @ aef + * @\}$

GPE обрабатывается последовательно слева направо.

Сначала, для $\{b \ d \ \underline{H}\}$, модуль d будет помещен над модулем b, сформируется первый элемент списка углов – $\{d-b\}$. Для $\{b \ d \ \underline{H} \ g \ \underline{V}\}$ модуль g помещается справа от супермодуля $\{b \ d \ \underline{H}\}$. Список углов модифицируется (замена $\{d-b\}$ на $\{d-g\}$, так как для экономии памяти и эффективности расчетов должна поддерживаться выпуклость углов). Затем для $\{b \ d \ \underline{H} \ g \ \underline{V} \ h \ \underline{H}\}$ модуль h располагают над супермодулем $\{b \ d \ \underline{H} \ g \ \underline{V}\}$. Список углов становится $\{h-d-g\}$. Когда для $\{b \ d \ \underline{H} \ g \ \underline{V} \ h \ \underline{H} \ c \ @\}$ встречается оператор @, необходимо выбрать один подходящий угол для модуля c на основе следующей эвристики: (1) выбрать такой угол, в который модуль может быть полностью помещен или (2) выбрать такой угол, у которого помещение модуля не приведет к увеличению площади плана. Если вставка модуля в каждый угол может увеличить план, то тогда выбирается любой угол. В данном примере выбирается угловое ограничение (d,g) для первого оператора @, угловой список становится $\{h-c-g\}$. Процесс свертки продолжается схожим образом до очередного оператора @. Снова необходимо выбрать эвристически один угол из списка $\{h-c-g\}$. Для второго оператора @ выбирается ограничение (h,c). Процесс декодирования завершен.

В качестве представления плана была выбрана структура GPE, так как она относительно проста (связана с бинарным деревом), при этом может отображать планы негильотинного типа.

Генетический алгоритм. В данном разделе рассматриваются процедуры генетического поиска для решения задачи планирования на основе обобщенной польской записи. В генетических алгоритмах используется эволюционный подход, они относятся к классу эволюционных вычислений (ЭВ).

ЭВ представляют собой совокупность алгоритмических, программных, аппаратных средств и приближенных эвристических методов, основанных на имитации механизмов эволюции для синтеза структур обработки данных, а также на статистическом подходе к исследованию ситуаций и итерационном приближении к искомому решению [3]. Идеи эволюции и самоорганизации оказались плодотворными на практике. В моделях и алгоритмах ЭВ ключевым элементом конструкции являются построение начальной модели и правил, по которым она может изменяться (эволюционировать) [4].

Генетические алгоритмы оперируют с популяцией решений. Гибкость структуры генетических алгоритмов, возможность настройки множества параметров позволяют получить более высокие результаты по сравнению с другими методами.

Для более полного учета специфики планирования в данной работе решение представлено не одной, а несколькими хромосомами: $S = \{H1, H2, H3, H4\}$. Хромосома H1 кодирует разметку модулей, H2 определяет структуру бинарного дерева, а H3 содержит информацию о типах внутренних вершин дерева (операторы разрезов и угловой оператор), H4 описывает ориентацию модулей.

Пусть n – число модулей. Тогда размеры хромосом H1, H2, H3 равны $n-1$, а размер хромосомы H4 равен n .

Каждый ген g_i хромосомы H1 принимает значение от 1 до $(n+1-i)$. Пример: $n = 4$, тогда $1 \leq g_1 \leq 4$, $1 \leq g_2 \leq 3$, $1 \leq g_3 \leq 2$.

Декодирование хромосомы H1 производится путем последовательного анализа генов хромосом, выбора элементов опорного вектора V , записи в порядке выборки в итоговый вектор разметки Q .

Пример: пусть для $n = 4$ имеется хромосома вида $H1 = \langle 2, 1, 2 \rangle$, опорный вектор $V^1 = \langle 1, 2, 3, 4 \rangle$, вектор разметки Q (первоначально пустой). $g_1 = 2$, тогда в V^1 выбирается 2-й элемент, записывается $q_1 = b^1_2 = 2$ в Q . Элемент b^1_2 удаляется, получается вектор $V^2 = \langle 1, 3, 4 \rangle$. $g_2 = 1$, тогда в V^2 выбирается 1-й элемент, записывается $q_2 = b^2_1 = 1$ в Q . Элемент b^2_1 удаляется, получается вектор $V^3 = \langle 3, 4 \rangle$. $g_3 = 2$, тогда в V^3 выбирается 2-й элемент, записывается $q_3 = b^3_2 = 4$ в Q . Элемент b^3_2 удаляется, получается вектор $V^4 = \langle 3 \rangle$. Наконец, $q_4 = b^4_1$.

Каждый ген g_i хромосомы H2 принимает значение от i до $(n-1)$.

Пусть дан алфавит $A = \{X, *\}$. Структуру бинарного дерева можно задать, используя на базе алфавита A польское выражение для бинарного дерева, где X соответствует листьям дерева (областям), а $*$ – внутренним вершинам дерева (операторам V , H , $@$). Польское выражение для дерева, представленного на рис. 2, имеет следующий вид: $XX*XX*X**$.

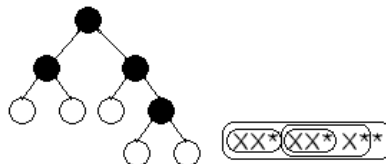


Рис. 2. Бинарное дерево

Число X-элементов ОПЗ обозначим через n_x , а через n_* – число *– элементов. Для бинарного дерева выполняется равенство: $n_x = n_* + 1$. Первый знак * в ОПЗ (при просмотре слева направо) может появиться только после двух знаков X. Пронумеруем позиции между знаками X, как показано на рис. 3.

$$X X _ X^1 _ X^2 _ X^3 _ X^4 _ \dots X^{n_x-1} _$$

Рис. 3. Схема ОПЗ

Максимальное число знаков *, которое может появиться в позиции, равно номеру позиции. Если ОПЗ соответствует перечисленным свойствам, то ему соответствует бинарное дерево.

Польская запись строится в результате декодирования хромосомы H2. Значение гена указывает номер позиции в ОПЗ, в которую необходимо поместить знак *.

Рассмотрим пример. Пусть имеется для $n_*=4$ (число модулей $n=5$) хромосома вида $H2 = \langle 2, 2, 4, 4 \rangle$.

ОПЗ будет иметь вид (рис. 4):

$$X X _ X^1 _ X^{2*} _ X^3 _ X^{4*} _$$

Рис. 4. ОПЗ хромосомы H2

С помощью хромосомы H3 задаются типы операторов (H, V, @). Значением гена является 0, 1 или 2 (флаг 0 – H-разрез, флаг 1 – V-разрез, флаг 2 – угловой оператор @). Разметка внутренних вершин осуществляется последовательно в порядке расположения знаков * в ОПЗ.

Хромосома H4 определяет ориентацию модулей. Если $g_i = 0$, то высота прямоугольного модуля больше его ширины, при $g_i = 1$ высота меньше, либо равна ширине.

Декодирование хромосом имеет оценку трудоемкости $O(n)$, где n – число модулей. Пространственная сложность для одного решения имеет оценку $O(n)$.

В работе используются два типа оператора кроссинговера: K1 и K2 [1]. При помощи K1 можно получить новые решения путем комбинирования составами хромосом родительской пары. Для этого необходимо задать специальный параметр – P_{k1} (вероятность кроссинговера). Осуществляется последовательный просмотр хромосом H1-H4, и с вероятностью P_{k1} производится обмен гомологичными хромосомами родительской пары.

С помощью K2 новые решения образуются путем обмена гомологичных хромосом гомологичной пары. Задается параметр P_{k2} – вероятность кроссинговера. Последовательно просматриваются локусы хромосом. С вероятностью P_{k2} производится обмен гомологичными генами.

Комбинированный кроссинговер реализуется следующим образом. Выбирается первичная родительская пара R1 и R2. Производится кроссинговер типа K2. Образуется дочерняя пара R1' R2'. Получившуюся пару следует рассматривать как вторичную родительскую, к ней применяется оператор кроссинговера K1 (обмен на уровне целых хромосом). В итоге появляется пара R1'' и R2'' как результат последовательного использования операторов кроссинговера K2 и K1.

Оператор мутации выполняется следующим образом. Задается параметр P_m – вероятность мутации. Просматриваются последовательно локусы хромосом решения. С вероятностью P_m осуществляется мутация гена в текущем локусе. Мутация заключается в принятии геном случайного значения из заданного диапазона значений для гена в данном локусе.

В основу селекции для выбора родительской пары хромосом положен «принцип рулетки». В соответствии с этим принципом для всех решений рассчитывается вероятность выбора на основе значений фитнеса.

На каждом i -м этапе осуществляется генетический поиск путем модификации хромосом, входящих в заданный для этого этапа набор NH_i модифицируемых типов хромосом (рис. 5). Это означает, что в полном объеме используется кроссинговер $K1$, заключающийся в комбинировании наборов хромосом, входящих в решение, а кроссинговер $K2$ и мутация применяются только к тем хромосомам, которые входят в набор типов модифицируемых хромосом.

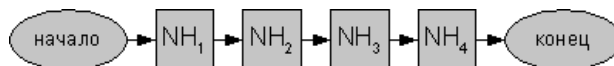


Рис. 5. Последовательный подход

На рис. 6,а представлен псевдокод алгоритма планирования, реализующий последовательный подход.

С помощью процедуры *НАЧ_ГЕНЕРАЦИЯ(задача, генетика)* осуществляется генерация популяции P_n . На каждом шаге производится обработка особей популяции (всего S шагов). В соответствии с текущим набором типов модифицируемых хромосом происходит эволюционирование популяции.

На рис. 6,б приведен псевдокод алгоритма генетического поиска, реализованного в каждом узле последовательной структуры.

<pre> Алгоритм ПЛАНИРОВАНИЕ; begin задача := ИСХОДНЫЕ_ДАННЫЕ; генетика := НАСТРОЙКА; попул := НАЧ_ГЕНЕРАЦИЯ(задача,генетика); С := ЧИСЛО_ШАГОВ; SEQN := ДЛИНА_ПОСЛЕДОВАТЕЛЬНОСТИ; К := 1; фитнес := РАСЧЕТ(попул, генетика); while К <= С do { I := 1; наборы := ВЫБОР_НАБОРОВ(SEQN); while I <= SEQN do { ЭВОЛЮЦИЯ (попул,наборы[I]); I := I + 1; }; }; К := К + 1; }; end. </pre>	<pre> Процедура ЭВОЛЮЦИЯ (попул,наборы[I]) begin Т := ЧИСЛО_ПОКОЛЕНИЙ; while Т > 0 do { кросс_попул := 0; мут_попул := 0; L := число_кросс; while L > 0 do { род_пара := ВЫБОР(попул, фитнес, генетика); доч_пара := КРОСС_2(род_пара,наборы[I],генетика); доч_пара := КРОСС_1(доч_пара,генетика); кросс_попул := ВКЛЮЧИТЬ(кросс_попул, доч_пара); L := L - 1; }; тек_попул := ОБЪЕДИНИТЬ(попул, кросс_попул); мут_попул := МУТАЦИЯ(тек_попул,наборы[I],генетика); тек_попул := ОБЪЕДИНИТЬ(тек_попул, мут_попул); фитнес := РАСЧЕТ(тек_попул, генетика); лучш_реш := ВЫДЕЛИТЬ(тек_попул, фитнес); попул := СЕЛЕКЦИЯ(тек_попул, фитнес); Т := Т - 1; }; end. </pre>
---	--

а

б

Рис. 6. а – псевдокод алгоритма планирования; б – псевдокод процедуры ЭВОЛЮЦИЯ

На каждой генерации (число генераций T) последовательно реализуются операторы кроссинговера и мутации. Популяция становится временно расширенной, а после редукции уменьшается до начального размера.

Операторы кроссинговера $K2$ и $K1$ выполняются L раз. Каждый раз выбирается родительская пара (по принципу рулетки). Сначала выполняется оператор $K2$, учитывающий типы модифицируемых хромосом в текущем наборе. Потом пара дочерних особей обрабатывается оператором $K1$ на хромосомном уровне. В итоге в коллекцию *кросс_попул* попадает пара особей после последовательного выпол-

нения операторов K2 и K1. По окончании кроссоверного цикла популяции *попул* и *кросс_попул* объединяются в *тек_попул*.

Затем хромосомы подвергаются мутации с заданной степенью вероятности, образуются мутанты. Процедура *ОБЪЕДИНИТЬ*(*тек_попул*, *мут_попул*) мутанты присоединяются к *тек_попул*. Процедура *РАСЧЕТ*(*тек_попул*, *генетика*) определяет фитнес для новых особей. Запоминается лучшее решение.

В конце реализуется процесс «естественного отбора» (редукция популяции).

Кластеризация. Для того, чтобы повысить эффективность планирования, можно воспользоваться принципом "разделяй и властвуй". Исходные модули группируются в кластеры. Каждый кластер имеет заданный размер (максимальное число объектов, которое он должен объединять).

Кластеры могут интегрироваться в кластеры более высокого уровня. ГА применяется к каждому кластеру. Кластеризация может быть простейшей (когда не учитываются характеристики модулей), а также направленной (когда модули сортируются на основе их геометрических параметров. Группировка модулей, у которых разница по площади невелика, является эффективной при кластерном планировании (тогда мертвая зона может получиться меньше). Важно отметить, что эффективность кластеризации лучше проверять на задачах большой размерности.

Заключение. Благодаря совместному использованию подходов, предложенных в работах [1,2], создан прототип кластерного планировщика, который использует механизм генетического для поиска субоптимального плана, представленного в виде довольно простой и эффективной структуры GPE.

Разработанные многохромосомные структуры отличаются простотой декодирования и гомологичностью, что облегчает выполнение генетических операторов и упрощает организацию поиска.

Необходимо отметить, что при генетическом поиске реализуются случайные изменения, а для нахождения глобального оптимума нередко надо определить цепочку «осмысленных» изменений. Поэтому повысить качество получаемых решений можно за счет внедрения дополнительных эвристик.

Предлагаемый в данной статье алгоритм планирования СБИС на основе эволюционного подхода находится в стадии разработки.

Предполагается дальнейшее развитие (учет дополнительных физических ограничений) и реализация данного алгоритма (как программы-планировщика), а также проведение серии тестов для изучения характеристик и экспериментального определения его временной сложности.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Лебедев В.Б.* Планирование СБИС на основе многоуровневой эволюции // Известия ТРТУ. – 2001. – № 4 (22). – С. 153-163.
2. *Lin, C. -T., Chen, D. -S., and Wang, Y. -W.* GPE: A New Representation for VLSI Floorplan Problem // Proceedings of the International Conference on Computer Design, Freiburg, Germany. – 2002. – P. 42-44.
3. *Курейчик В.М., Родзин С.И.* Эволюционные алгоритмы: генетическое программирование (обзор) // Известия РАН. Теория и системы управления. – 2002. – № 1. – С. 127-137.
4. *Курейчик В.В., Курейчик В.М., Родзин С.И.* Концепция эволюционных вычислений, инспирированных природными системами // Известия ЮФУ. Технические науки. – 2009. – № 4 (93). – С. 16-24.

Ерошенко Илья Николаевич

Технологический институт федерального государственного автономного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г. Таганроге.

E-mail: lutaries@rambler.ru.

347928, г. Таганрог, пер. Некрасовский, 44.

Тел.: 88634644031.

Yeroshenko Ilya Nikolaevich

Taganrog Institute of Technology – Federal State-Owned Educational Establishment of Higher Vocational Education “Southern Federal University”.

E-mail: lutaries@rambler.ru.

44, Nekrasovsky, Taganrog, 347928, Russia.

Phone: +78634644031.

УДК 658.512.2.011.5

М.В. Лисяк

ПРИМЕНЕНИЕ ГЕНЕТИЧЕСКОГО ПРОГРАММИРОВАНИЯ ДЛЯ РЕШЕНИЯ ЗАДАЧИ СИМВОЛЬНОЙ РЕГРЕССИИ*

Рассматривается приложение метода генетического программирования для решения задачи символьной регрессии. Входными данными задачи являются множество переменных и констант (аргументов функции), множество элементарных функций, набор фитнес-тестов (таблица эталонных значений искомой функции при различных значениях аргументов). Результатом решения задачи является символьная запись математического выражения, наилучшим образом описывающего заданную фитнес-тестами функциональную зависимость. Приводится описание задачи и метода её решения, способ кодирования решений и оценки их качества, используемый алгоритм генетического поиска, модификации стандартных генетических операторов, параметры алгоритма поиска. Приводится пример работы программы, реализующей описанный метод.

Генетическое программирование; генетический алгоритм; символьная регрессия; оценки фитнеса; репродукция; кроссинговер; многократная селекция.

M.V. Lisyak

APPLICATION OF GENETIC PROGRAMMING FOR DECISION OF SYMBOLIC REGRESSION PROBLEM

There is considered the decision of symbolic regression problem by means of genetic programming method. The problem input data include the set of independent variables and constants (arguments of function), the set of elementary functions, list of fitness-cases (table containing reference values of unknown function at different values of the arguments). Result of the problem decision is mathematical expression in symbolic form, which fits the functional dependence, described by fitness-cases, in the best way. There are described the problem and the method of decision, the way of alternative decisions representation and quality measuring, algorithm of genetic search, modifications of standard genetic operators, parameters of search algorithm. The result of running a program, which realizes described method, is considered.

Genetic programming; genetic algorithm; symbolic regression; fitness measures; reproduction; crossover; over-selection.

Введение. Символьной регрессией называется процесс поиска математического выражения, наилучшим образом описывающего функциональную зависимость, заданную некоторым набором числовых данных. Поиск осуществляется путем перебора произвольных композиций элементарных функций из заданного множества.

Исходными данными задачи являются множество независимых переменных, множество элементарных математических функций, которые могут входить в состав искомой функции, а также таблица значений искомой функции при различ-

* Работа выполнена при поддержке: РФФИ (гранты № 09-07-00318, № 10-07-00538), г/б № 2.1.2.1652.