

Lebedev Vladimir Borisovich

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

E-mail: lbk@tsure.ru.

44, Nekrasovsky, Taganrog, 347928, Russia.

Phone: +78634371743.

The Department of System Analysis and Telecommunications; Associate Professor.

УДК 658.512

И.Н. Ерошенко

**ОБЗОР СОВРЕМЕННЫХ МОДЕЛЕЙ ЭВОЛЮЦИОННЫХ ВЫЧИСЛЕНИЙ
ДЛЯ РЕШЕНИЯ ЗАДАЧИ ПЛАНИРОВАНИЯ СБИС***

Рассматривается задача планирования сверхбольших интегральных схем. Планирование является одним из ранних этапов конструкторского проектирования СБИС. Данная задача относится к классу NP. В работе указаны недостатки традиционных алгоритмов поиска субоптимального плана СБИС. Приведены описания более современных моделей эволюционных вычислений: дифференциальной эволюции, муравьиного алгоритма, метода роя частиц, меметического алгоритма, который является гибридным генетическим алгоритмом. Для перечисленных алгоритмов приведены описания представления плана в виде польской записи, O-дерева, B-дерева.*

Планирование СБИС; эволюционные вычисления; оптимизация.

I.N. Yeroshenko

**REVIEW OF MODERN EVOLUTIONARY COMPUTATION MODELS
FOR VLSI FLOORPLANNING**

The article is dedicated to the problem of VLSI floorplanning. Floorplanning refers to the early stage of physical VLSI design. VLSI floorplanning is an NP-hard problem. In this paper the disadvantages of the traditional floorplanning algorithms for searching suboptimal solution are mentioned. The description of the more modern evolutionary computation models is given (differential evolution, ant colony optimization, swarm optimization, memetic algorithm which is a hybrid genetic algorithm). Floorplan representations Polish expression, O-tree, B-tree are briefly described for the mentioned algorithms.*

VLSI floorplanning; evolutionary computation; optimization.

Введение. Планирование является важным этапом конструкторского проектирования сверхбольших интегральных схем. Оно влияет на производительность, размеры чипа.

Планирование заключается в размещении модулей таким образом, чтобы они не пересекались, при этом целевая функция минимизировалась. Целью оптимизации могут быть минимизация площади кристалла, длины межсоединений и др. При качественном размещении не только уменьшается площадь микросхемы, но и уменьшается стоимость внутренних соединений. Задача планирования относится к классу NP.

Планирование СБИС и представления плана. При планировании СБИС важно выбрать соответствующее представление плана. Обычно выделяют два типа

* Работа выполнена при частичной поддержке РФФИ (проект № 09-01-00492).

представлений: гильотинные и негильотинные. Гильотинный план получается путем рекурсивных гильотинных разрезов прямоугольника на части. План, который не может быть получен путем таких разрезов, называется негильотинным. В данной статье кратко описаны только те представления, которые связаны с рассматриваемыми метаэвристическими алгоритмами.

Гильотинный план описывается польской записью, которая состоит из имен модулей, операторов вертикального и/или горизонтального разрезов (рис. 1).

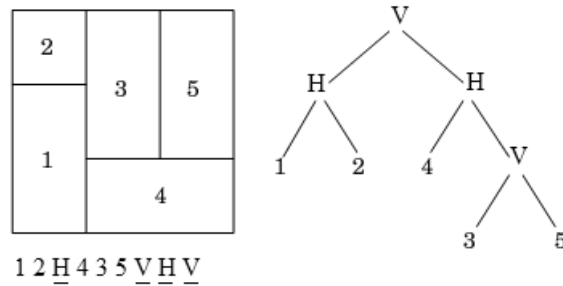


Рис. 1. Гильотинный план, польская запись и бинарное дерево разрезов

Негильотинные планы могут быть описаны с помощью множества разных представлений. Рассмотрим два схожих представления: О-дерево и В*-дерево.

План с n прямоугольными модулями представляется горизонтальным (вертикальным) О-деревом из $n+1$ узлов, причем n узлов соответствуют n модулям, а один узел соответствует левой (нижней) границе плана (рис. 2). О-дерево кодируется кортежем (T, π) , где T – битовая строка, состоящая из $2n$ бит, для определения структуры дерева, а π задает размещение модулей. План называется LB-компактным, если ни один из модулей не может быть перемещен влево и вниз без необходимости сдвига других модулей.

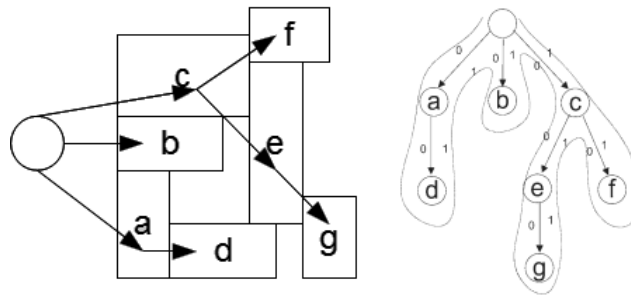


Рис. 2. План и соответствующее О-дерево

Основанное на бинарном дереве представление «В*-дерево» предложено для LB-компактного размещения. В В*-дереве вершина обозначает модуль. Левая дочерняя вершина обозначает самый нижний модуль справа от модуля, представленного родительской вершиной (рис. 3). Правая дочерняя вершина обозначает модуль выше модуля, представленного родительской вершиной, с той же х-координатой.

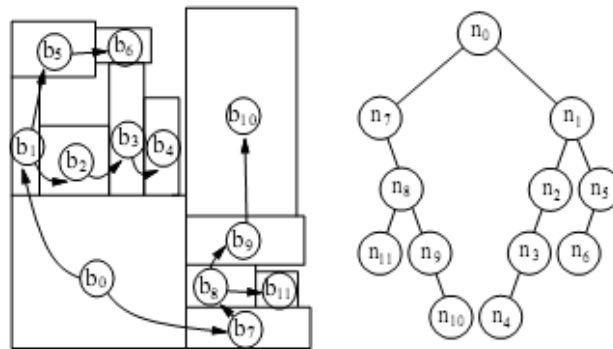


Рис. 3. План и соответствующее V^* -дерево

Эволюционные вычисления. Термин «эволюционные вычисления» (ЭВ) подразумевает использование для решения разнообразных задач проектирования, оптимизации, прогнозирования и управления совокупности 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. В течение последних 50 лет были предложены разнообразные схемы ЭВ: генетические алгоритмы, генетическое программирование, эволюционные стратегии, эволюционное программирование, дифференциальная эволюция, модели поведения роя пчел, стаи птиц и колонии муравьев, моделирование отжига или потока и другие конкурирующие эвристические алгоритмы [1].

Эти методы относятся к междисциплинарному направлению биоинформатики. Основное преимущество таких методов – возможность распараллеливания процесса поиска и получения множества альтернативных квазиоптимальных решений [2].

Долгое время для планирования СБИС наиболее популярными алгоритмами были моделирование отжига и генетические алгоритмы.

Моделирование отжига. Данный алгоритм был предложен в первой половине 80-х гг. Моделирование отжига имеет наибольшую степень сходства с моделями эволюционных стратегий и эволюционного программирования. Однако имеются и отличия: размер популяции решений равен единице; поиск решений ведется локально от точки к точке; мутация является единственным оператором поиска, а селекция проводится по специальным правилам [1].

Генетические алгоритмы. Генетический алгоритм (ГА) – метаэвристический алгоритм поиска, имитирующий природную эволюцию и работающий с популяцией решений. ГА относится к классу эволюционных алгоритмов, был предложен еще в 1975 году Джоном Холландом. Эволюционный процесс представляется как способность лучших особей оказывать большее влияние на состав новой популяции на основе длительного выживания из более многочисленного потомства [3]. Основным блоком в ГА является блок генетических операторов [4]. Важную роль в ГА играет кроссинговер – оператор скрещивания, благодаря которому осуществляется рекомбинация решений.

Недостатки традиционных подходов. К недостаткам моделирования отжига можно отнести сильную зависимость качества конечного решения от качества начального решения, отсутствие учета специфики задачи планирования. В работе [2] отмечено, что в рамках ГА качество получаемых решений для задачи планирования оставляет желать лучшего, так как не учитывается топологическая инфор-

мация. Вычислительная сложность у алгоритма моделирования отжига велика. ГА являются затратными в плане используемой памяти. Рассмотрим более современные модели ЭВ.

Дифференциальная эволюция. Дифференциальная эволюция (ДЭ) – это поисковый эволюционный алгоритм, основанный на популяции (как и генетический алгоритм), использует операторы кроссинговера, мутации и селекции. Метод дифференциальной эволюции был разработан Рэйнером Сторном и Кеннетом Прайсом в 1997 г. Самое важное отличие от ГА заключается в том, что главным оператором ДЭ является не кроссинговер, а мутация. ДЭ использует мутацию в качестве поискового механизма и селекцию для исследования перспективных областей. Кроссинговер использует компоненты существующих членов популяции для построения пробных векторов.

Задача оптимизации, состоящая из D параметров, может быть представлена D -размерным вектором. В ДЭ сначала создается случайным образом популяция NP векторов.

Мутация. Для каждого целевого вектора $x_{i,G}$ создается мутантный вектор в соответствии с формулой:

$$v_{i,G+1} = x_{r1,G} + F * (x_{r2,G} - x_{r3,G}) \quad (1)$$

Индексы $r1, r2, r3$ выбираются случайным образом на отрезке $[1, NP]$. Они должны отличаться друг от друга и от текущего индекса i . Поэтому число векторов в популяции должно быть не меньше четырех. F – один из параметров метода, некоторая положительная действительная константа на отрезке $[0,2]$.

Кроссинговер. Компоненты целевого вектора $x_{i,G}$ смешиваются с компонентами мутантного вектора $v_{i,G+1}$ для получения пробного вектора. Элементы мутантного вектора попадают в пробный вектор с вероятностью CR .

Целевой вектор сравнивается с пробным вектором. Вектор с наименьшим значением ЦФ сохраняется.

В работе [6] в качестве представления плана для ДЭ используется V^* -дерево.

Муравьиный алгоритм. Идея муравьиного алгоритма (ACO – ant colony optimization) заключается в моделировании поведения муравьев, который связан с их умением быстро находить наиболее краткий маршрут от муравейника к источнику пищи. Колония муравьев является разумной многоагентной системой. ACO был предложен Марко Дориго в 1992 г.

Муравьи достигают источника пищи и возвращаются обратно, оставляя феромонный след. Если другие муравьи находят такие феромонные тропы, то они с большой долей вероятности пойдут по ним. Феромон со временем испаряется. На коротком пути плотность феромонов высокая. Испарение феромона также позволяет избегать попадания в локальные оптимумы.

В качестве представления плана в работе [7] выбрана польская запись. Модули могут менять ориентацию.

Пусть n_M – число модулей, n_R – число операторов в польской записи. Для дерева разрезов соблюдается равенство $n_M = n_R + 1$. Поиск решений осуществляется на полном графе поиска решений $G=(X,U)$, где $X = X_1 \cup X_2 \cup X_3 \cup X_4$. Вершины множества $X_1 = \{x_{1i} \mid i=1,2,\dots, n_M\}$ соответствуют модулям m_{1i} , размещаемым в первой ориентации. Вершины множества $X_2 = \{x_{2i} \mid i=1,2,\dots, n_M\}$ соответствуют модулям m_{2i} , размещаемым во второй ориентации. Вершины множества $X_3 = \{x_{3i} \mid i=1,2,\dots, n_R\}$ соответствуют горизонтальным разрезам H . Вершины множества $X_4 = \{x_{4i} \mid i=1,2,\dots, n_R\}$ соответствуют вертикальным разрезам V .

Поиск решения задачи планирования осуществляется коллективом муравьев $Z = \{z_k \mid k=1,2,\dots,1\}$. На каждой итерации муравьиного алгоритма каждый муравей

zk строит индивидуальное решение задачи планирования. Решением является маршрут в графе $G=(X,U)$, включающий n_M вершин, принадлежащих множествам X_1 или X_2 , n_R вершин, принадлежащих множествам X_3 или X_4 .

Моделирование поведения муравьев в задаче планирования связано с распределением феромона на ребрах графа G . На начальном этапе на всех ребрах графа G откладывается небольшое количество феромона. Процесс поиска решений итерационный. Каждая итерация включает три этапа. На первом этапе муравей находит решение, на втором этапе откладывает феромон, на третьем этапе осуществляется испарение феромона. В работе [7] используется циклический (ant-cycle) метод муравьиных систем. В этом случае феромон откладывается агентом на ребрах после полного формирования решения (плана СБИС).

Метод роя частиц. Рой частиц можно рассматриваться как многоагентная система, каждый агент действует независимо по простым правилам. Совокупное поведение системы получается разумным. Оптимизация методом роя частиц (particle swarm optimization – PSO) была предложена Кеннеди и Эберхартом в 1995 г.

На каждой итерации частицы перемещаются в новые позиции. Новая позиция определяется по формуле:

$$x_i(t+1) = x_i(t) + v_i(t+1), \quad (2)$$

где $v_i(t+1)$ – скорость перемещения частицы из позиции $x_i(t)$ в позицию $x_i(t+1)$. Начальное состояние определяется как $x_i(0)$, $v_i(0)$. Формула показана в векторном виде, для отдельного измерения j формула примет вид:

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1). \quad (3)$$

Пусть $f_i(t)$ – текущее значение ЦФ в позиции частицы i в позиции $x_i(t)$, $x_i^*(t)$ – лучшая позиция частицы i , которую она посещала с начала первой итерации, $f_i^*(t)$ – значение ЦФ в этой позиции, $F(t)$ – лучшее значение ЦФ среди всех частиц в момент времени t , $x(t)$ – позиция с этим значением.

Скорость частицы на i -м шаге вычисляется как:

$$v_{ij}(t+1) = w * v_{ij}(t) + k_1 * rnd(0,1) * (x_{ij}^*(t) - x_{ij}(t)) + k_2 * rnd(0,1) * (x_j(t) - x_{ij}(t)), \quad (4)$$

где $rnd(0,1)$ – случайное число на интервале $(0,1)$; w – инерционный коэффициент, k_1, k_2 – некие коэффициенты ускорения.

Формула расчета скорости состоит из трех слагаемых. Предыдущая скорость частицы служит инерционным компонентом. Второе слагаемое прямо пропорционально текущему расстоянию частицы от ее наилучшей позиции, которая была найдена с первой итерации. Второе слагаемое является когнитивным компонентом. Третье слагаемое, называемое социальным компонентом, прямо пропорционально текущему расстоянию частицы от наилучшей позиции роя в текущий момент времени t . Благодаря социальному компоненту, частица может передвигаться в оптимальные позиции, найденные другими частицами.

После случайной инициализации популяции частиц для каждой из них вычисляется значение ЦФ $f_i(t+1)$. Если оно окажется лучше, чем $f_i^*(t)$, то $f_i^*(t+1) = f_i(t+1)$, иначе $f_i^*(t+1) = f_i^*(t)$. Среди $f_i(t+1)$ выбирается лучшее решение $F(t)$, после чего вычисляются новые значения скоростей и новые позиции частиц. Итерационный процесс повторяется.

В работе [8] в качестве представления плана для PSO выбрана польская запись, в работе [9] применяется V^* -дерево.

Меметический алгоритм. Меметические алгоритмы (МА) – одно из интенсивно развивающихся направлений в области эволюционных вычислений. Термин «меметический алгоритм» был впервые представлен Москато еще в 1989 г. Под

МА понимают гибридизацию эволюционного алгоритма с индивидуальным обучением или процедурами локального поиска.

В работе [5] МА используется для минимизации площади охватывающего прямоугольника и длины проводников в рамках задачи планирования СБИС. В качестве представления выбрано О-дерево. Каждая особь в популяции МА – допустимый план, представленный О-деревом.

Целевая функция в данном МА представлена в виде формулы:

$$\text{cost}(F) = w_1 * \frac{\text{Area}(F)}{\text{Area}^*} + w_2 * \frac{\text{Wirelength}(F)}{\text{Wirelength}^*}. \quad (5)$$

В формуле Area^* , Wirelength^* представляют минимальную площадь и минимальную длину проводников соответственно. Так как на практике данные значения неизвестны, то используются оценочные значения. w_1 и w_2 – коэффициенты для ЦФ площади и ЦФ длины проводников, $0 \leq w_1 \leq 1$, $0 \leq w_2 \leq 1$, $w_1 + w_2 = 1$. Пригодность особи вычисляется по следующей формуле:

$$f((T, \pi)) = \frac{1}{\text{cost}(F_{(T, \pi)})}. \quad (6)$$

В данном МА используется метод локального поиска (МЛП), основанный на детерминированном алгоритме. При заданном начальном плане, закодированном с помощью О-дерева (T, π) МЛП находит локальное оптимальное решение посредством проверки тех О-деревьев, которые могут быть получены путем удаления и вставки модулей. Когда модуль вставляется, его ориентация может измениться, если это приведет к улучшению решения.

Псевдокод МА:

- 1) $t := 0$;
- 2) создать начальную популяцию $P(t)$ размера PopSize;
- 3) рассчитать фитнес особей в $P(t)$ и запомнить лучшее решение в best;
- 4) Пока не выполнено условие останова:
 - a) $t := t + 1$;
 - b) Для каждой особи в $P(t)$:
 - i. данная особь становится первым родителем p_1 ;
 - ii. выбрать методом рулетки родителя p_2 ;
 - iii. с определенной вероятностью применить кроссинговер для создания дочерней особи c_1 ;
 - iv. если $\text{fitness}(c_1) \geq \text{fitness}(v)$, то оптимизировать c_1 , используя метод локального поиска;
 - v. если $\text{fitness}(c_1) \geq \text{fitness}(p_1)$, то $p_1 := c_1$;
 - vi. если $\text{fitness}(c_1) \geq \text{fitness}(\text{best})$, то $\text{best} := c_1$;
 - vii. С заданной вероятностью выполнить операторы мутации над особью c_1 для создания особи f ;
 - viii. если $\text{fitness}(f) \geq \text{fitness}(v)$, то оптимизировать f , используя метод локального поиска;
 - ix. если $\text{fitness}(f) \geq \text{fitness}(p_1)$, то $p_1 := f$;
 - x. если $\text{fitness}(f) \geq \text{fitness}(\text{best})$, то $\text{best} := f$.
- 5) Вывести лучшее решение best.

Заключение. Анализ публикаций по теме планирования СБИС и конструкторского проектирования СБИС в целом показал, что модели эволюционных вычислений успешно применяются для поиска субоптимального решения. Для того чтобы повысить качество получаемых решений, авторы используют различные адаптивные механизмы, заимствованные у природы. Важную роль играет гибридизация метаэвристик. В данной работе это представлено на примере меметиче-

ского алгоритма. Успешное использование меметических алгоритмов показало, что полный отказ от традиционных моделей эволюционных вычислений нецелесообразен, можно использовать гибридный подход, который позволяет задействовать преимущества отдельных метаэвристик и учитывать специфику задачи оптимизации.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. 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.
2. Курейчик В.М. Бионспирированный поиск с использованием сценарного подхода // Известия ЮФУ. Технические науки. – 2010. – № 7 (108). – С. 7-13.
3. Курейчик В.М. Гибридные генетические алгоритмы // Известия ЮФУ. Технические науки. – 2007. – № 2 (77). – С. 5-12.
4. Курейчик В.М. Модифицированные генетические операторы // Известия ЮФУ. Технические науки. – 2009. – № 12 (101). – С. 7-14.
5. Tang, Maolin and Yao, Xin. A memetic algorithm for VLSI floorplanning // IEEE Transactions On Systems, Man, And Cybernetics–Part B: Cybernetics. – 2007. – № 37 (1).
6. D. Jackuline Moni and S. Arumugam and D. Gracia Nirmala Rani. VLSI Floor Planning relying on Differential Evolution Algorithm // ICGST International Journal on Artificial Intelligence and Machine Learning. – 2007. – Vol. 7, № 1. – P. 62-67.
7. Лебедев О.Б. Планирование СБИС на основе метода муравьиной колонии // Известия ЮФУ. Технические науки. – 2010. – № 7 (108). – С. 67-73.
8. Лебедев Б.К., Лебедев В.Б. Планирование на основе роевого интеллекта и генетической эволюции // Известия ЮФУ. Технические науки. – 2009. – № 4 (93). – С. 25-33.
9. Tsung-Ying Sun, Sheng-Ta Hsieh, Hsiang-Min Wang, Cheng-Wei Lin. Floorplanning based on particle swarm optimization // ISVLSI '06 Proceedings of the IEEE Computer Society Annual Symposium on Emerging VLSI Technologies and Architectures. – 2006. – P. 5-10.

Статью рекомендовал к опубликованию д.т.н., профессор Ю.О. Чернышев.

Ерошенко Илья Николаевич

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

E-mail: lutaries@rambler.ru.

г. Таганрог, ул. Розы Люксембург, 114.

Тел.: 88634644031.

Кафедра систем автоматизированного проектирования; аспирант.

Yeroshenko Ilya Nikolaevich

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

E-mail: lutaries@rambler.ru.

114, Rosa Luxemburg Street, Taganrog, Russia.

Phone: +78634644031.

The Department of Computer Aided Design; Postgraduate Student.