

2. *О.В. Коновалов.* Проблемы формирования трехмерных векторных структур. Известия ТРТУ №2, тематический выпуск «Интеллектуальные САПР» Материалы Всероссийской научно-технической конференции с участием зарубежных представителей «Интеллектуальные САПР-97». Таганрог: ТРТУ, 1998.
3. *О.В. Коновалов.* Моделирование объектов в трехмерном пространстве. Сборник «Тезисы докладов РиЭвНХ». М.: МЭИ, 1998.
4. *О.В. Коновалов.* Построение алгоритма обучения сетей радиального базиса при помощи градиентного правила. Труды конференций IEEE AIS'02 CAD-2002. М.: Физматлит, 2002.
5. Вычислительные комплексы, системы и сети / А.М.Ларионов, С.А.Майоров, Г.И.Новиков: Учебник для вузов. Л.: Энергоиздат. Ленингр. отделение, 1987.
6. *Смирнов А.Д.* Архитектура вычислительных систем: Учеб. пособие для вузов. М.: Наука. 1990. 320 с.
7. Системы параллельной обработки: Пер. с англ./ Под ред. Д.Ивенса. М.: Мир, 1985.
8. *Головкин Б.А.* Параллельные вычислительные системы. М.: Наука, 1980.
9. *Ферради Д.* Оценка производительности вычислительных систем. М.: Мир, 1981.
10. Параллельные вычислительные системы с общим управлением/ И.В. Прангишвили, С.Я. Виленкин, И.Л.Медведев. М.: Энергоатомиздат, 1983.
11. *A.Barak, O.La'adan,A.Shiloh.* Scalable Cluster Computing with MOSIX for LINUX. The Hebrew University of Jerusalem, 1998.
12. *A.Barak, O.La'adan.* The MOSIX Multicomputer Operating System for High Performance Cluster Computing. Journal of Future Generation Computer System, 13, 1998.
13. *A.Barak, S.Guday, R.G.Wheeler.* The MOSIX Distributed Operating System, Load Balancing for UNIX. In Lecture Notes in Computer Science, Springer-Verlag, 1993.
14. *A.Barak and A.Braverman.* Memory Ushering in Scalable Computing Cluster. Journal of Microprocessors and Microsystems, 22, 1998.
15. *A.Barak, A.Braverman, I.Gilderman and O.Laden.* Performance of PVM with the MOSIX Preemptive Process Migration. In Proc. Seventh Israeli Conf. on Computer Systems and Software Engineering, 1996.
16. *Y.Amir, B.Averbuch, A.Barak, R.S. Borgstrom and A.Keren.* An Opportunity Cost Approach for Job Assignment and Reassignment in a Scalable Computing Cluster. In PDCS '98, 1998.
17. *А.Ф. Колчин, А.Ф. Стрекалов, М.В. Овсянников, С.В. Сумароков.* Управление жизненным циклом продукции. М.: Анахарсис, 2002.

УДК 681.3:536.2.072

В.Б. Лебедев

НЕЧЕТКОЕ ПЛАНИРОВАНИЕ СБИС НА ОСНОВЕ ЭВОЛЮЦИОННОЙ АДАПТАЦИИ*

Планирование это ранняя фаза проектирования чипа. Оно дает информацию о приблизительных площади, задержке, мощности и других рабочих характеристиках. Во многих приложениях, хотя и требуется хорошая топологическая структура, но не вся информация о всех модулях имеется в наличии, или, что еще хуже, часть этой информации неточна. Примеры таких приложений – это проектирование больших систем, где топологическая схема требуется уже на ранних стадиях проектирования, но еще не были спроектированы все модули. Планирование с учетом неопределенности - это проблема получения хорошей топологии, когда информа-

* Работа выполнена при поддержке Мин. образования, грант № T02-02.3-491

ция о размерах модулей не является полной. В этой работе рассматривается проблема планирования с учетом неопределенности. Предлагается эффективный метод работы с нечеткими данными на основе генетических процедур.

В традиционной проблеме планирования входными данными является набор модулей (блоков), их ширин и высот и информация о соединениях. Пытаются найти топологический план такой, что общая площадь или задержка, или мощность, или комбинация этих параметров минимизирована [1]. В данной работе мы рассматриваем планирование с нечеткими данными. Данные с нечеткостью будут состоять из списков распределения длины $W_i (1 \leq i \leq n)$ и списков распределения высоты $H_i (1 \leq i \leq n)$, где n – число модулей [2]. Необходимо построить план, причем такой, чтобы математическое ожидание площади описывающего прямоугольника (площади плана) было минимальным. Каждый список распределений содержит пары чисел: ширина (или высота) модуля и её вероятность.

$$W_i = \{(w_{i1}, p(w_{i1})), (w_{i2}, p(w_{i2})), \dots, (w_{im_{w_i}}, p(w_{im_{w_i}}))\}, \sum_{j=1}^{m_{w_i}} p(w_{ij}) = 1,$$

$$H_i = \{(h_{i1}, p(h_{i1})), (h_{i2}, p(h_{i2})), \dots, (h_{im_{h_i}}, p(h_{im_{h_i}}))\}, \sum_{j=1}^{m_{h_i}} p(h_{ij}) = 1.$$

Наибольшее распространение получил подход, при котором топологический план представляют в виде иерархического дерева. Каждый лист дерева соответствует блоку, а каждая внутренняя вершина соответствует составному блоку на топологическом плане вверх. Общий план определяется с помощью последовательной свертки блоков.

В традиционном планировании, когда два блока с размерами (w_1, h_1) и (w_2, h_2) группируются вертикально в общий блок, размеры блока могут быть рассчитаны по следующим уравнениям:

$$w_{1,2} = w_1 + w_2; \quad h_{1,2} = \max(h_1, h_2).$$

Когда те же блоки (куски) группируются горизонтально:

$$w_{1,2} = \max(w_1, w_2); \quad h_{1,2} = h_1 + h_2.$$

Различная группировка модулей приводит к различным топологическим планам.

Учет неопределенности при группировке модуля 1, определяемого распределениями W_1 и H_1 , с модулем 2, определяемым распределениями W_2 и H_2 , даёт следующее распределение ширины и высоты для модулей 1,2:

$$W_{1,2} = W_1 \oplus W_2; \quad H_{1,2} = H_1 \otimes H_2.$$

Операции \oplus и \otimes называемые соответственно *сложением распределений* и *максимум распределений* определяются следующим образом:

$$D_1 \oplus D_2 = \{(d_{1i} + d_{2j}, p(d_{1i}) * p(d_{2j})) | (d_{1i}, p(d_{1i})) \in D_1 \text{ и } (d_{2j}, p(d_{2j})) \in D_2\}$$

$$D_1 \otimes D_2 = \left\{ \left(d_{1i}, p(d_{1i}) * \sum_{j: d_{2j} \leq d_{1i}} p(d_{2j}) \right) | (d_{1i}, p(d_{1i})) \in D_1 \text{ и } (d_{2j}, p(d_{2j})) \in D_2 \right\} \cup$$

$$\bigcup \left\{ \left(d_{2j}, p(d_{2j}) * \sum_{i: d_{1i} < d_{2j}} p(d_{1i}) \right) | (d_{1i}, p(d_{1i})) \in D_1 \text{ и } (d_{2j}, p(d_{2j})) \in D_2 \right\}.$$

С помощью этих выражений путем последовательной свертки блоков рассчитываются списки распределений высоты и ширины плана и их математическое ожидание.

Для оценки решения задачи используется три метода

Оптимистичный метод: для каждого модуля выбираем минимальное значение ширины из списка ширин и минимальное значение высоты из списка высот и используем их как ширину и высоту модуля.

Консервативный метод: для каждого модуля выбираем максимальное значение из списка ширин и максимальное из списка высот и используем их как ширину и высоту модуля.

Метод ожидаемого значения: для каждого модуля рассчитываем математическое ожидание распределения ширины и математическое ожидание распределения высоты и используем их как ширину и высоту.

Интуитивно, оптимистичный и консервативный методы не дадут реалистичные результаты и разработчики не смогут полагаться на них для создания высокоуровневых решений. Оптимальный топологический план (оптимальный топологический план, сгенерированный после завершения разработки всех модулей) будет скорее всего иметь большую площадь, чем при оптимистическом методе и меньшую чем при консервативном.

Оптимистический и консервативный методы не способны генерировать надежный топологический план для конечного продукта, потому что они не способны хорошо оценить конечные размеры модулей.

Можно прогнать как оптимистический, так и консервативный методы на задаче и посмотреть на результаты обоих методов для того, чтобы принять проекторочные решения. Разница между размерами двух топологических планов может быть настолько значительной, что невозможно будет судить, основываясь на этих результатах.

Топологический план, сгенерированный методом ожидаемого значения (вероятностным), является оптимальным топологическим планом в конце проектирования.

Использование ожидаемого значения ширины и высоты кажется более разумным.

В данной работе рассматриваются генетические процедуры для решения задачи планирования, отличительными чертами которых является то, что решение для более полного учёта специфики представляется тремя хромосомами $R=\{H1,H2,H3\}$, что увеличивает возможности целенаправленного комбинирования [3]. С другой стороны, гены в одних и тех же локусах хромосом являются гомологичными, что упрощает выполнение генетических операторов кроссинговера и мутации. Хромосома *H1* содержит информацию о структуре дерева. Хромосома *H2* несёт информацию о разметке множества вершин *E*. Хромосома *H3* содержит информацию о типах разрезов (*H* - горизонтальный или *V* - вертикальный). В результате декодирования хромосомы *H1* определяется польское выражение, по которому строится дерево разрезов. После разметки вершин дерева разрезов в соответствии с *H2* и *H3* методом последовательной бинарной свертки [3] строится план, для которого рассчитываем математическое ожидание распределения ширины и математическое ожидание распределения высоты. Представление решения набором из 3-х хромосом дает возможность организации поиска решений в раз-

личных постановках, оставляя отдельные виды хромосом неизменными в процессе генетического поиска.

Например, при фиксированных H_1 , H_2 искать оптимальное решение только лишь за счет изменения H_3 , т.е. типов разрезов (Н или V).

Очевидно, что фиксация отдельных хромосом в некоторой постановке приводит к сужению пространства поиска, но при этом возможна потеря оптимальных решений. В этой связи представляется целесообразным комбинирование отдельными постановками при поиске оптимального решения. В общем случае возможны три подхода к комбинированию постановок: последовательный, параллельный и параллельно-последовательный. Комбинированный кроссинговер реализуется следующим образом. Сначала на выбранной родительской паре R_1 и R_2 реализуется кроссинговер K_1 , т.е. осуществляется обмен генами. Образуется дочерняя пара R_1' и R_2' . Далее эта пара рассматривается как родительская и к ней применяется кроссинговер K_2 , т.е. осуществляется обмен хромосомами. В результате последовательного применения к родительской паре R_1 и R_2 кроссинговеров K_1 и K_2 образуется дочерняя пара R_1'' и R_2'' .

Исследования показали, что при фиксированном значении управляющих параметров, трудоёмкость алгоритма имеет линейную зависимость и пропорциональна $O(N)$, где N – число блоков.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Naveed Sherwani. Algorithms for VLSI physical design automation. Kluwer academic publishers. Boston /Dordrecht/ London. 1995.
2. K. Bazargan, S. Kim, and M. Sarrafzadeh, Nostradamus: a Floorplanner of Uncertain Designs.// IEEE Transaction on Computer Aided Design of Integrated Circuits and Systems, vol. 18, no.4, April 1999.
3. Лебедев В.Б. Планирование СБИС методом генетического поиска. // Известия ТРТУ. Тематический выпуск "Интеллектуальные САПР". Таганрог: Изд-во ТРТУ. 1999. №3. С. 119-126.

УДК 621.3.06

В.А. Литвиненко, В.А. Калашников

АЛГОРИТМ АДАПТАЦИИ ПРОЕКТНОЙ ОПЕРАЦИИ ОПРЕДЕЛЕНИЯ КЛИК ГРАФА

Введение. Адаптация в технических системах - это способность системы изменять свое состояние и поведение (параметры, структуру, алгоритм, функционирование) в зависимости от условий внешней среды путем накопления и использования информации о ней [1]. Классификация методов адаптации рассмотрена в [1,2]. Одним из методов адаптации является априорная параметрическая адаптация, когда приспособление объекта адаптации к внешним условиям производится с помощью параметров адаптации, которые выбираются на основе заранее полученных адаптирующих воздействий, соответствующих внешним условиям [1,2].

В работе [8,10] рассмотрен состав адаптивного программного модуля проектной операции определения клик графа [3,4], построенного на основе библиотеки альтернативных программных модулей (ПМ), выбор которых осуществляется на