

6. Colomi A., Dorigo M., Maniezzo V., "Distributed Optimization by Ant Colonies," Proceedings of the First European Conference on Artificial Life, Paris, France, F.Varela and P.Bourgin (Eds.), Elsevier Publishing, 134–142, 1991.
7. Colomi A., Dorigo M., Maniezzo V., "The Ant System: Optimization by a colony of cooperating agents," Tech.Rep.IRIDIA/94-28, Université Libre de Bruxelles, Belgium, 1996.
8. Балюк Л.В. Возможности комбинированных стратегий решения задачи размещения разногабаритных элементов СБИС // Труды Международных научно-технических конференций «Интеллектуальные системы» (AIS'06) и «Интеллектуальные САПР» (CAD-2006). Научное издание в 3-х томах. – М.: Изд-во «Физико-математической литературы», 2006, Т.1. – С. 552-557.

Б.К. Лебедев, С.А. Степаненко

МЕХАНИЗМЫ ГЕНЕТИЧЕСКОГО ПОИСКА ПРИ РАЗМЕЩЕНИИ*

Введение. Задача размещения является одним из наиболее важных шагов в процессе проектирования СБИС, поскольку качество полученного решения в значительной степени влияет на выполнение последующей трассировки соединений, которые, в свою очередь, сильно влияют на производительность схемы [1-2]. Задача размещения относится к классу NP-полных, поэтому не существует алгоритма, позволяющего получать ее решение за полиномиальное время. Проведенные исследования имеющихся подходов к ее решению показывают, что существующие алгоритмы выдают решения, которые далеки от оптимальных [1,3,4]. Следовательно, эта задача остается по-прежнему актуальной. В работе рассматривается структура хромосом, принципы кодирования и декодирования, а также модифицированные операторы, применяемые в генетическом алгоритме решения задачи размещения.

Постановка задачи. Задача размещения формулируется следующим образом. Имеется множество элементов $M = \{m_i \mid i = 1, \dots, N\}$ с фиксированными размерами и множество цепей $C = \{c_i \mid i = 1, 2, \dots, K\}$, связывающих элементы множества M . Необходимо произвести размещение элементов на плоскости таким образом, чтобы отсутствовали перекрытия элементов, и при этом минимизировалась целевая функция $F(x) \rightarrow \min$. В качестве критериев оптимизации используются: общая площадь схемы, суммарная длина проводников, временные задержки.

Для представления относительного расположения элементов на плоскости будем использовать метод пары последовательностей (Sequence-Pair), который впервые был предложен Murata и др. в 1996 году [5]. Представление плана топологии с помощью пары последовательностей состоит из двух перестановок целых чисел $\langle 1, 2, \dots, N \rangle, \langle 1, 2, \dots, N \rangle$. Каждый элемент последовательности соответствует номеру прямоугольного элемента, расположенного на плоскости без перекрытий с другими элементами, где общее число элементов равно N .

Эта пара последовательностей определяет относительное расположение элементов в пространстве, но для упаковки элементов необходимо также знать их пространственную ориентацию. Пространственная ориентация элементов задается вектором $O = \{o_i \mid i = 1, 2, \dots, N\}$, $o_i \in \{1, 2, 3, 4\}$, $1 \leq i \leq N$. Таким образом, для каждого элемента существует четыре возможных ориентации (North = 1, East = 2, South = 3, West = 4).

* Работа выполнена при поддержке РФФИ (грант № 06-01-81018, № 05-08-18115) и программ развития научного потенциала высшей школы 2006-2008 гг. (РНП.2.1.2.3193, РНП 2.1.2.2238).

Разработка генетического алгоритма включает три основных компонента: разработка структуры, принципов кодирования и декодирования хромосом; разработка основных генетических операторов; разработка общей структуры генетического поиска [6].

Кодирование решения. Основным недостатком представления решения в виде пары последовательностей является то, что хромосомы в таком виде являются не гомологичными. В этом случае, при синтезе новых хромосом возникает проблема получения легальных решений. Это связано с наложением ограничений на значения генов. В частности, налагаемые ограничения заключаются в том, что в хромосоме не должно быть генов с одинаковыми значениями. Это требует дополнительных временных затрат, связанных с контролем и с использованием более трудоемких методов кроссинговера и мутации, не допускающих получение нелегальных решений [9]. Поэтому требуется закодировать последовательность пар в виде гомологичной хромосомы. Рассмотрим подробно процесс кодирования решения в виде гомологичной хромосомы.

Хромосома представляет собой массив генов, число которых на единицу меньше числа размещаемых элементов $H = \{g_i \mid i = 1, 2, \dots, (N-1)\}$. Ген g_i может принимать значения, лежащие в интервале $1 \leq g_i \leq N - i + 1$, т.е. чем выше порядковый номер i , тем меньшее значение он может принимать.

Процедура перехода от гомологичных хромосом к паре последовательностей пар. Рассмотрим процесс перехода от гомологичной хромосомы $H = \{g_i \mid i = 1, 2, \dots, (N-1)\}$ к последовательности целых чисел $S = \{s_i \mid i = 1, 2, \dots, N\}$ (последовательность пар состоит их двух таких последовательностей $\langle S_1, S_2 \rangle$). Для этого нам понадобится опорный вектор номеров элементов $B = \{b_i \mid i = 1, 2, \dots, N\}$, $1 \leq b_i \leq N$, $b_i \neq b_j$, $\forall i \neq j$, $1 \leq i, j \leq N$.

1. Имеется гомологичная хромосома $H = \{g_i \mid i = 1, 2, \dots, (N-1)\}$ и опорный вектор $B = \{b_i \mid i = 1, 2, \dots, N\}$. На первом шаге $i = 1$.
2. Выбирается ген хромосомы с номером i : $k = g_i$.
3. Определяется значение элемента опорного вектора B с номером $k = g_i$: b_k . Это значение присваивается i -ому элементу получаемой в процессе декодирования последовательности S : $s_i = b_k$.
4. Элемент b_k удаляется из опорного вектора B .
5. Если $i < N - 1$, то $i = i + 1$ и переход к пункту 2, иначе – переход к пункту 6.
6. К этому моменту опорный вектор B состоит из одного элемента b_1 , этот элемент добавляется в конец последовательности S : $s_{i+1} = b_1$. Переход к пункту 7.
7. Конец работы алгоритма.

Гомологичные хромосомы хорошо подходят для применения стандартной операции кроссинговера, поскольку гены, расположенные в одинаковых локусах являются гомологичными и обмен местами генов из разных хромосом не приводит к получению нелегальных решений. Однако наряду со стандартными генетическими операторами, которые носят случайный характер поиска, в генетическом алгоритме применяются также более эффективные операторы, использующие знания о

задаче размещения и эти операторы работают не с гомологичной хромосомой, а с парой последовательностей. В результате на выходе получается не гомологичная хромосома, а пара последовательностей и, чтобы вернуть это решение в популяцию, необходимо применить обратную процедуру перехода от пары последовательностей к гомологичной хромосоме, которая описывается далее.

Процедура перехода от пары последовательностей к гомологичной хромосоме. Рассмотрим теперь процесс перехода от последовательности пар к гомологичной хромосоме. При этом также используется опорный вектор номеров элементов $B = \{b_i \mid i = 1, 2, \dots, N\}$.

1. Имеется последовательность целых чисел $S = \{s_i \mid i = 1, 2, \dots, N\}$ и опорный вектор номеров элементов $B = \{b_i \mid i = 1, 2, \dots, N\}$. На первом шаге $i = 1$.

2. Выбирается элемент последовательности с номером i : s_i .

3. Просматривается опорный вектор B и определяется, какую позицию в нем занимает элемент со значением s_i , пусть это будет позиция с номером k .

4. Гену g_i хромосомы H присваивается значение k : $g_i = k$.

5. Элемент b_k удаляется из опорного вектора B .

6. Если $i < N$, то $i = i + 1$ и переход к пункту 2, иначе – переход к пункту 7.

7. Конец работы алгоритма.

Трудоёмкость декодирования хромосом имеет оценку $O(n)$. Достоинствами предложенной структуры хромосом являются:

- 1) Отсутствие сложных элементов.
- 2) Линейная оценка временной сложности декодирования.
- 3) Гены, расположенные в одних и тех же локусах, являются гомологичными.

Отметим, что процесс декодирования хромосомы опирается на вектор B . Таким образом, одному и тому же генотипу H , в зависимости от вектора B , соответствуют различные фенотипы.

Генетические операторы. Кроссинговер. В алгоритме используются стандартный многоточечный оператор кроссинговера [6-8]. Оператор кроссинговера выполняется с заданной вероятностью P_C . При этом для каждой пары генов хромосом, расположенных в одном локусе, генерируется случайным образом число в интервале $[0;1]$ и, если это число меньше заданной вероятности кроссинговера P_C , производится обмен этих генов в потомках. Поскольку код одного решения представляется парой хромосом $\langle H_1, H_2 \rangle$, то кроссинговер может применяться как к хромосомам в пределах одного решения, так и к хромосомам из разных решений.

Рекомбинация. Оператор рекомбинации производит обмен целыми хромосомами для выбранной пары решений.

Мутация. Оператор мутации применяется к одной выбранной хромосоме. Данный оператор работает следующим образом: выбирается случайно некоторый ген g_i в хромосоме, генерируется случайное число в интервале $[0;1]$ и, если это число меньше заданной вероятности мутации P_M , производится случайное изменение гена в интервале $[1; N - i + 1]$ (длина хромосомы равна $N - 1$).

Рассмотренные выше операторы носят чисто случайный характер, поскольку они не используют знания о решаемой задаче. Более эффективными являются опе-

раторы, использующие так называемые «жадные эвристики», которые пытаются изменить закодированное решение таким образом, чтобы улучшить целевую функцию. Рассмотрим ориентированные на знания операторы, разработанные специально для задачи размещения. Данные операторы работают не с гомологичной хромосомой, а с парой последовательностей, поэтому необходимо выполнить преобразование гомологичной хромосомы к последовательности пар, используя описанный выше алгоритм.

Оператор сближения элементов цепей. Данный оператор пытается уменьшить длину проводников цепей. Оператор работает с первой последовательностью S_1 в последовательности пар $\langle S_1, S_2 \rangle$. Поскольку первая последовательность задает порядок укладки элементов, интуитивно понятно, что чтобы уменьшить длину проводников цепи, нужно чтобы элементы, принадлежащие одной цепи, были расположены как можно ближе в последовательности S_1 .

В исходных данных задачи размещения задан список цепей, задающих связи между элементами: $C = \{c_i \mid i = 1, 2, \dots, K\}$. Каждая цепь (элемент списка c_i) содержит список элементов, связанных ею: $Net = \{b_i \mid i = 1, 2, \dots, V\}$, где $b_i \in M$, M – множество элементов, $|Net| = V$ – степень цепи (количество контактов цепи).

Оператор сближения элементов цепей работает отдельно с каждой цепью. Выбор цепи осуществляется случайным образом или на основании некоторого критерия, например, чем больше количество элементов цепи, тем большую вероятность выбора она имеет. Оценка выбранной цепи Net производится следующим образом: для каждого элемента цепи определяется его порядковый номер в последовательности S_1 . Далее определяется элемент с минимальным (а) и максимальным (б) порядковым номером в последовательности ($S_1 = \langle \dots, s_a, \dots, s_b \rangle$ на отрезке $[a; b]$ расположены все элементы связанные цепью Net). Длина отрезка равна $l = b - a + 1$. Возможны два случая:

1) Если длина этого отрезка больше числа элементов связанных цепью Net : $l > |Net|$, то внутри отрезка содержатся также элементы не связанные цепью Net (точное число элементов равно $k = l - |N|$) и можно длину этого отрезка уменьшить за счет обмена местами элементов цепи с остальными элементами.

2) В противном случае: $l = |Net|$ и на отрезке $[a; b]$ расположены только элементы цепи Net . В этом случае длину проводников цепи можно попытаться уменьшить за счет изменения горизонтальных или вертикальных ограничений между элементами, связанными цепью.

Сближение элементов цепи можно проводить разными способами:

1) Расположение всех элементов цепи в том порядке, в котором они расположены на отрезке $[a; b]$, начиная с позиции a , вытесняя все остальные элементы.

2) Обмен крайних элементов с номерами a , b с какими-либо элементами внутри отрезка, тем самым, «притягивая» крайние элементы цепи.

Оператор разбивания критических путей в графах горизонтальных и вертикальных ограничений. Данный оператор также работает с последовательностью пар и пытается уменьшить общую площадь схемы, что достигается путем разбиения критических путей в графах ограничений, которые состоят из элемен-

тов, размеры которых входят в выражения, по которым рассчитывается ширина и высота схемы.

Пусть имеется последовательность пар $\langle S_1, S_2 \rangle$, по имеющейся последовательности пар построены графы ограничений $H(X, E_H)$ и $V(X, E_V)$, выполнена процедура упаковки элементов и определены пути максимальной длины, состоящие из элементов, размеры которых вносят вклад в определение ширины и высоты схемы: $R_H = \langle r_i \mid i = 1, \dots, L \rangle$ (путь в графе горизонтальных ограничений) и $R_V = \langle r_j \mid j = 1, \dots, T \rangle$ (путь в графе вертикальных ограничений), $1 \leq i, j \leq N$.

Производится выбор одного из путей (R_H или R_V) одним из следующих способов: если требуется уменьшить ширину схемы - выбирается путь R_H , если же требуется уменьшить высоту схемы – выбирается путь R_V . Далее производится выбор некоторого множества паросочетаний элементов пути. Затем выполняется обмен местами элементов каждой пары в последовательности S_1 . Например, пусть выбрана пара элементов $\langle a, b \rangle$, если до выполнения операции обмена элементы были расположены в последовательности S_1 таким образом: $S_1 = \langle \dots, a, \dots, b, \dots \rangle$, то после выполнения операции обмена их положение в последовательности S_1 станет таким: $S_1 = \langle \dots, b, \dots, a, \dots \rangle$.

Рассмотренные выше операторы позволяют производить модификацию решения с целью уменьшения размеров схемы и суммарной длины проводников. Экспериментальные исследования подтверждают их большую эффективность по сравнению со стандартными операторами кроссинговера, мутации и рекомбинации.

Заключение. В данной статье рассмотрена задача размещения разногабаритных элементов, в качестве критериев оптимизации использовались размеры схемы и суммарная длина проводников. Для представления относительное размещение элементов на плоскости использовался метод пары последовательностей (Sequence-Pair). Этот метод с успехом применялся в алгоритмах размещения основанных на моделировании отжига, он также хорошо подходит и для генетических алгоритмов. В предложенной статье были рассмотрены структуры хромосом, принципы кодирования и декодирования, а также модифицированные операторы, применяемые в генетическом алгоритме решения задачи размещения. Экспериментальные исследования подтверждают эффективность предложенных методик.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Jason Cong, Joseph R., Shinnerl и Min Xie (UCLA Computer Science), Tim Kong (Magma Design Automation) и Xin Yuan (IBM Corporation, Microelectronics Division). Large-Scale Circuit Placement. ACM Transactions on Design Automation of Electronic Systems, Vol. 10, No. 2, April 2005.
2. Adya S. N., Yildiz M., Markov I. L., Villarrubia P. G., Parakh P. N., Madden P. H. Benchmarking for large-scale placement and beyond. In Proceedings of the International Symposium on Physical Design. ACM, Monterey, 95–103., 2003.
3. Cong J., Kong T., R J. Shinnerl, Xie M., and Yuan X. Large-scale circuit placement: Gap and promise. in Proc. of the International Conference on Computer-Aided Design, November 2003.
4. Chang C.C., Cong J. и Xie M. 2003. Optimality and scalability study of existing placement algorithms. In Proceedings of the Asia South Pacific Design Automation Conference. 621–627.

5. Hiroshi Murata, Kunihiro Fujiyoshi, Shigetoshi Nakatake, Yoji Kajitani. VLSI Module Placement Based on Rectangle-Packing by the Sequence-Pair. IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, VOL. 15, NO. 12, DECEMBER 1996.
6. Батищев Д.И. Генетические алгоритмы решения экстремальных задач: Учебное пособие. – Воронеж, 1995. – 69 с.
7. Курейчик В.В. Эволюционное моделирование: Учебное пособие. – Таганрог: Изд-во ТРТУ, 2003. – 76с.
8. Гладков Л.А., Курейчик В.М., Курейчик В.В. Генетические алгоритмы. – Таганрог: Изд-во ТРТУ, 2004. – 400 с.
9. Лебедев Б.К., Лебедев О.Б. Методы размещения. – Таганрог: Изд-во ТРТУ, 2006. – 108 с.

Р.В. Потарусов

ГИБРИДНЫЙ ГЕНЕТИЧЕСКИЙ ПОИСК ДЛЯ ЗАДАЧИ УПАКОВКИ БЛОКОВ*

Введение. Задача упаковки блоков (Bin Packing Problem (BPP)) – хорошо известная NP-сложная комбинаторная оптимизационная задача. В BPP целью является скомбинировать (объединить) элементы в блоки определенного объема так, чтобы минимизировать общее количество блоков [1,3,5-7].

BPP является интересной темой для исследования по следующим причинам [2,4,8,9]:

- 1) BPP встречается во многих отраслях промышленности таких, как производство стали, стекла и бумаги;
- 2) существует много других промышленных задач, которые выглядят различными, но имеют очень сходную структуру такие, как составление сметы капитала, планирование, распределение времени процессора и дизайн СБИС;
- 3) сложность проблемы BPP – NP-полная задача.

В данной работе предлагается алгоритм гибридного генетического поиска (АГГП). Экспериментально показано, что предлагаемый метод на определенных экземплярах задачи упаковки лучше, чем самый известный и успешный на сегодняшний день гибридный группирующий генетический алгоритм (ГГГА) Эммануэля Фалкенауэра [3,9].

1. Формулировка задачи. BPP описывается следующим образом [1,3,5-10]. Дан определенный набор элементов $I=\{i_1, \dots, i_n\}$, $|I| = n$ с весами $W=\{w_1, \dots, w_j\}$, $|W| = j$ такими, что $0 \leq w_i \leq c$, где c – объем каждого блока и набор блоков $B=\{b_1, \dots, b_N\}$, $|B| = N$. Разбить I на N подмножеств так, чтобы сумма весов в каждом подмножестве была не более c и N было минимальным.

Другими словами: даны n элементов (товара, изделий) и N блоков (рюкзаков, ящиков или бинов (bin)) с w_j – весом элемента j , c – грузоподъемностью (емкостью) каждого блока. Необходимо назначить каждый элемент одному блоку так, чтобы общий вес элементов в каждом блоке не превышал c и число использованных блоков было минимально.

2. Целевая функция. Целевая функция (ЦФ) описывается следующим образом [2,3,6,7,9]:

* Работа выполнена при поддержке РФФИ (грант № 07-01-00511, № 05-08-18115) и программ развития научного потенциала высшей школы 2006-2008 гг. (РНП.2.1.2.3193, РНП 2.1.2.2238).