

УДК 681.3.001.63

О.Б. Лебедев

МУРАВЬИНЫЙ АЛГОРИТМ РАЗБИЕНИЯ*

Предлагаются новые технологии, принципы и механизмы решения задачи разбиения, основанные на моделировании процессов адаптивного поведения муравьиной колонии. Описывается структура графа поиска решений, процедура поиска решений на графе, способы отложения и испарения феромона. В работе используется циклический (ant-cycle) метод муравьиных систем. Рассматривается метод размещения, основанный на разбиении схемы. По сравнению с существующими алгоритмами достигнуто улучшение результатов.

Разбиение; муравьиная колония; оптимизация.

O.B. Lebedev

ANT ALGORITHM OF PARTITIONING

New technologies, principles and mechanisms of the decision of a problem of the partitioning based on modeling of processes of adaptive behaviour of an ant colony are offered. The structure of the graph of search of decisions, procedure of search of decisions on the graph, ways of adjournment and evaporation pheromone is described. In work it is used cyclic (ant-cycle) a method of ant systems. The method of placing based on partitioning of the scheme is considered. In comparison with existing algorithms improvement of results is reached.

Partitioning; an ant colony; optimization.

Введение. Одной из широко востребованных задач целочисленного программирования является задача разбиения, рассматриваемой в комбинаторном направлении теории графов. Современная СБИС может содержать десятки миллионов транзисторов, поэтому в связи с ограниченными возможностями вычислительных средств (память, скорость) не может быть спроектирована топология всей схемы в целом. Нормальным является разбиение схемы группированием компонентов в блоки. В результате разбиения формируется множества блоков и межсоединений между блоками. В очень больших схемах используется иерархическая структура разбиения.

Существующие алгоритмы разбиения делятся на два класса: конструктивные и итеративные [1]. Первый класс характеризуется относительным быстродействием, но в тоже время низким качеством решения. Второй класс является более трудоемким, однако позволяет получать более качественные решения [1].

Среди итеративных алгоритмов можно выделить детерминированные и вероятностные. В детерминированных алгоритмах изменение разбиения (решения) реализуется на основе четкой, детерминированной зависимости от изменяемого решения. Недостатком является частое попадание в локальный оптимум («локальную яму»).

В вероятностных алгоритмах переход к новому решению осуществляется случайным образом. Недостатком алгоритмов, реализующих чисто случайный поиск, является значительная трудоемкость [1].

В последние годы интенсивно разрабатывается научное направление с названием «Природные вычисления», объединяющее математические методы, в кото-

* Работа выполнена при поддержке: РФФИ (грант № 09-01-00509), г/б № 2.1.2.1652.

рых заложены принципы природных механизмов принятия решений [2-4]. К таким методам можно отнести, прежде всего, методы моделирования отжига [5], метод эволюционного моделирования [6], генетические алгоритмы [7-9], эволюционной адаптации [10,11], алгоритмы роевого интеллекта [12,13] и муравьиные алгоритмы (Ant Colony Optimization – ACO) [14]. Идея муравьиного алгоритма моделирование поведения муравьёв, связанного с их способностью быстро находить кратчайший путь от муравейника к источнику пищи. Основу поведения муравьиной колонии составляет самоорганизация, обеспечивающая достижения общих целей колонии на основе низкоуровневого взаимодействия благодаря которому, в целом, колония представляет собой разумную многоагентную систему. Особенности являются наличие непрямого обмена, который и используется в муравьиных алгоритмах. Непрямой обмен – стигмержи (stigmergy), представляет собой разнесённое во времени взаимодействие, при котором одна особь изменяет некоторую область окружающей среды, а другие используют эту информацию позже, когда в неё попадают. Такое отложенное взаимодействие происходит через специальное химическое вещество – феромон (pheromone). Концентрация феромона на пути определяет предпочтительность движения по нему. При своём движении муравей метит путь феромоном, и эта информация используется другими муравьями для выбора пути. Концентрация феромонов определяет желание особи выбрать тот или иной путь. Однако, при таком подходе неизбежно попадание в локальный оптимум. Эта проблема решается благодаря испарению феромонов, которое является отрицательной обратной связью.

В работе используется представление задачи разбиения в виде адаптивной муравьиной системы, на основе сочетания принципов самообучения, самоорганизации и коллективной адаптации.

Постановка задачи. Пусть дан граф $G(X,U)$, где X – множество вершин, $|X| = n$, U – множество ребер. Необходимо разбить множество X на два непустых и непересекающихся подмножества X_1 и X_2 , $X_1 \cup X_2 = X$, $X_1 \cap X_2 = \emptyset$, $X_i \neq \emptyset$. На формируемые узлы (блоки, компоненты) накладываются ограничения: $|X_1| = n_1$, $|X_2| = n_2$, $n_1 + n_2 = n$. Критерий оптимизации – число связей F между X_1 и X_2 . Цель оптимизации – минимизация критерия F .

Механизмы разбиения на основе муравьиной колонии. Для поиска решения задачи используется полный граф решений $R(X,E)$, где E – множество всех ребер полного графа, связывающих множество вершин X . На множестве ребер E будем откладывать феромон. На начальном этапе на всех ребрах графа R откладывается одинаковое (небольшое) количество феромона Q/m , где $m=|E|$. Каждый из агентов формирует множество X_{1k} , где k – номер агента. Формирование множества X_{1k} осуществляется последовательно (пошагово). На каждом шаге t у k -го агента есть список вершин, уже включенных в формируемое множество – $X_{1k}(t)$ и список оставшихся (свободных) вершин $X_{ck}(t)$, $X_{1k}(t) \cup X_{ck}(t) = X$. На первом шаге в каждое формируемое множество $X_{1k}(t)$, где $t = 1$, включается вершина графа G , причем вершины графа G распределяются по узлам равномерно, то есть в каждом узле своя вершина, $(\forall i,j) [X_{1i}(1) \cap X_{1j}(1) = \emptyset]$. Такое распределение необходимо, чтобы все вершины графа G имели одинаковые шансы быть отправной точкой при формировании узла X_1 . В модификациях алгоритма использовались также ln муравьёв, причем каждая группа из l муравьёв используют в качестве начального одно и то же $X_{1i}(1)$. На конечном шаге $t=n_1$ k -м агентом будет сформирован узел $X_{1k}(n_1) = X_{1k}$. $|X_{1i}(n_1)| = n_1$. Множество вершин, помещенных в узел X_{2k} , определится как $X_{2k} = X \setminus X_{1k}$.

Моделирование поведения муравьёв в задаче разбиения связано с распределением феромона на ребрах графа R . При этом вероятность включения вершины $x_j \in G$ в формируемое отдельным муравьём множество $X_{Ik}(t)$, пропорциональна суммарному количеству феромона на ребрах, связывающих вершину x_j с $X_{Ik}(t)$. Количество откладываемого на ребрах графа R феромона обратно пропорционально числу связей между сформированными узлами. Чем меньше число связей между X_{Ik} и X_{2k} , тем больше феромона будет отложено на рёбрах полного подграфа $R_{Ik} \subset R$, построенного на вершинах узла X_{Ik} , следовательно, большее количество муравьёв будет включать вершины узла X_{Ik} в синтез собственных узлов. Для избежания преждевременной сходимости используется отрицательная обратная связь в виде испарения феромона. Процесс поиска решений итерационный. Каждая итерация t включает три этапа. На первом этапе муравей находит решение, на втором этапе откладывает феромон, на третьем этапе осуществляется испарения феромона. В работе используется циклический (ant-cycle) метод муравьиных систем. В этом случае феромоны откладываются агентом на ребрах после полного формирования решения.

На первом этапе каждой итерации каждый k -ый муравей формирует свое собственное множество X_{Ik} . Процесс построения множества X_{Ik} пошаговый. На каждом шаге агент применяет вероятностное правило выбора следующей вершины для включения ее формируемое множество $X_{Ik}(t)$.

Первый этап осуществляется следующим образом. Агент просматривает все свободные на данном шаге вершины $X_{ck}(t)$. Для каждой вершины $x_i \in X_{ck}(t)$ рассчитываются два параметра:

f_{ik} – суммарный уровень феромона на ребрах графа R , связывающих x_i с вершинами узла $X_{Ik}(t)$;

s_{ik} – число связей на графе G между x_i и $X_{Ik}(t)$.

По формуле (1) – при мультипликативной свертке, либо по формуле (2) – при аддитивной свертке определяется потенциальная стоимость F_{ik} связей x_i с $X_{ck}(t)$.

$$F_{ik} = (f_{ik})^\alpha \cdot (s_{ik} + 1)^\beta, \quad (1)$$

$$F_{ik} = (f_{ik})^\alpha + (s_{ik})^\beta, \quad (2)$$

где α, β – управляющие параметры, которые подбираются экспериментально.

Вероятность P_{ik} включения вершины $x_i \in X_{ck}(t)$ в формируемый узел $X_{Ik}(t)$ определяется следующим соотношением:

$$P_{ik} = F_{ik} / \sum_i F_{ik}. \quad (3)$$

Агент с вероятностью P_{ik} выбирает одну из вершин, которая включается в множество $X_{Ik}(t)$ и исключается из множества $X_{ck}(t)$.

При $\alpha = 0$ наиболее вероятен выбор вершины x_i , максимально связанной с вершинами узла $X_{Ik}(t)$, то есть алгоритм становится жадным.

При $\beta = 0$ выбор происходит только на основании феромона, что приводит к субоптимальным решениям.

Поэтому необходим компромисс между этими величинами, который находится экспериментально.

После формирования за n_j шагов муравьями узлов (каждый муравей – свой узел X_{Ik}), на втором этапе итерации, каждый муравей откладывает феромон на рёбрах полного подграфа $R_{Ik} \subset R$, построенного на вершинах узла X_{Ik} .

Количество феромона $\Delta\tau_k(l)$, откладываемое k -ым муравьем на каждом ребре подграфа $R_{l_k} \subset R$, построенного на l -ой итерации, определяется следующим образом:

$$\Delta\tau_k(l) = Q / D_k(l), \quad (4)$$

где l – номер итерации, Q – общее количество феромона, откладываемое k -ым муравьем на ребрах подграфа $R_{l_k} \subset R$, $D_k(l)$ – число связей на графе G между множествами X_{l_k} и X_{2k} , сформированными k -ым муравьем на l -ой итерации.

После того, как каждый агент сформировал решение и отложил феромон, на третьем этапе происходит общее испарение феромона на ребрах полного графа R в соответствии с формулой:

$$f_{ik} = f_{ik}(1 - \rho). \quad (5)$$

После выполнения всех действий на итерации находится агент с лучшим решением, которое запоминается. Далее осуществляется переход на следующую итерацию. Алгоритм разбиения на основе метода муравьиной колонии формулируется следующим образом.

1. В соответствии с исходными данными определяется число агентов. Оно равно числу n вершин графа $G(X, U)$.

2. Формируется граф поиска решений $R(X, E)$, в каждую вершину которого помещается агент.

3. На всех ребрах (или вершинах) графа $R(X, E)$ откладывается начальное количество феромона.

Задаются значения параметров: α , β . Выбирается критерий F_k по формулам (1) или (2).

4. Задается число итераций N_l .

5. На первом этапе каждой итерации на графе поиска решений $R(X, E)$ каждым агентом z_k строится маршрут M_k и формируется решение P^k .

6. Для каждого решения P^k , находится значение целевой функции D_k .

7. На ребрах (или вершинах) всех найденных маршрутов в графе поиска решений $R(X, E)$ агентами откладывается феромон. Количество феромона откладываемого k -ым агентом обратно пропорционально числу связей $D_k(l)$ на графе G между множествами X_{l_k} и X_{2k} , сформированными k -ым агентом на l -ой итерации.

8. Испарение феромона на ребрах (или вершинах) графа поиска решений $R(X, E)$.

9. Выбор лучшего решения, полученного на протяжении всех выполненных итераций.

10. Если все итерации выполнены, то конец работы алгоритма, в противном случае переход к пункту 5 для выполнения очередной итерации.

Временная сложность этого алгоритма на одной итерации определяется как $O(n^2)$, где n – количество вершин графа $G(X, U)$.

В общем случае время работы этого алгоритма зависит от времени жизни колонии l (число итераций), количества вершин графа n и числа муравьев m , помещаемых в каждую вершину графа поиска решений $R(X, E)$.

Размещение, основанное на разбиении схемы. Рассмотренный выше муравьиный алгоритм разбиения используется в качестве базовой процедуры в алгоритмах разбиения и размещения на основе многоуровневой парадигмы. Поскольку количество элементов, которые необходимо разместить может быть слишком велико для алгоритма, выполняющего детальное размещение, применяется иерархический подход. При таком подходе сначала все множество элементов разбивается на блоки, при разбиении минимизируется число внешних связей между блоками.

Размещение через разбиение – это важный класс алгоритмов размещения, основанных на последовательном делении заданной схемы на плотно связанные подсхемы, таким образом, чтобы минимизировать число цепей пересекаемых разрезом. Также, на каждом шаге разбиения схемы, имеющаяся область разделяется попеременно в горизонтальном или вертикальном направлении. Каждая подсхема присваивается одной части разбиения. Если продолжать этот процесс до тех пор, пока каждая подсхема не будет состоять из одного модуля, тогда каждый модуль будет размещен в уникальной позиции в схеме. Большинство алгоритмов размещения, основанных на разбиении, или Min-cut алгоритмы в той или иной степени основаны на рекурсивном разбиении схемы.

По мере выполнения рекурсии, подмножества ячеек уменьшаются, и распределение площади ячеек по размещаемой области становится более равномерным. После того как подмножества элементов становятся достаточно маленькими, к ним применяются специальные завершающие алгоритмы детального размещения. В структуре многоуровневого разбиения (multiway partitioning), промежуточные результаты от разбиения каждого региона используются, чтобы повлиять на окончательное разбиение других. Явное использование многоуровневого разбиения на каждом этапе может в некоторых случаях привести к конфигурации более близкой к глобальному оптимуму, чем та, что получается при использовании одного лишь рекурсивного разбиения.

Многоуровневый алгоритм работает с различными уровнями иерархии. Алгоритмы размещения на основе многоуровневой парадигмы завоевали внимание совсем недавно. Эти методы основаны на укрупнении (coarsening), релаксации (relaxation) и интерполяции (interpolation), которые определяются следующим образом.

- ◆ *Укрупнение.* Иерархии строятся либо снизу вверх путем рекурсивной агрегации, либо сверху вниз путем рекурсивного разбиения.
- ◆ *Релаксация.* Локальные процедуры оптимизации выполняются на каждом уровне агрегации.
- ◆ *Интерполяция.* Промежуточные решения переносятся с каждого уровня агрегации на смежный следующий уровень.

Простейший и наиболее часто применяемый подход это движение сверху вниз, от самого крупного к конечному уровню.

Заключение. Алгоритм разбиения был реализован на языке C++. Экспериментальные исследования проводились на ЭВМ типа IBM PC/AT. Примерно одинаковые по качеству решения можно получить, используя как аддитивную, так и мультипликативную свертку, варьируя управляющие параметры α и β . Исследованию подвергались примеры, содержащие до 1000 вершин. Тестирование производилось на бенчмарках 19s, PrimGA1, PrimGA2. Сравнение с известными алгоритмами показало, что при меньшем времени работы у полученных с помощью муравьиного алгоритма решений значения целевой функции лучше (меньше) в среднем на 6%. В среднем запуск программы обеспечивает нахождения решения, отличающегося от оптимального менее чем на 0,5%. Перспективными путями улучшения муравьиных алгоритмов являются различные адаптации параметров с использованием базы нечетких правил и их гибридизация с эволюционными алгоритмами. Как вариант, такая гибридизация заключается в построении итерационной структуры процесса разбиения графа, которая в рамках общего подхода к снижению размерности, основана на парадигме многошагового разбиения дина-

мически изменяющегося подмножества вершин. Динамическое изменение подмножества вершин, подвергающихся разбиению методом муравьиной колонии, осуществляется с помощью механизмов коллективной адаптации.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Naveed Sherwani*. Algorithms for VLSI physical design automation. Kluwer academic publishers. Boston/Dordrecht/London. 1995.
2. *G. Di Caro, F. Ducatelle, L.M. Gambardella*. AntHocNet: An adaptive nature-inspired algorithm for routing in mobile ad hoc networks. *European Transactions on Telecommunications*, 16(5): 443-455, 2005.
3. *Engelbrecht A.P.* Fundamentals of Computational Swarm Intelligence. John Wiley & Sons, Chichester, UK, 2005.
4. *МакКоннелл Дж.* Основы современных алгоритмов. – М.: Техносфера, 2004.
5. *Wong D.F., Leong H.W., and Lin C.L.* Simulated Annealing for VLSI Design. Boston, MA: Kluwer Academic, 1988.
6. *Емельянов В.В., Курейчик В.М., Курейчик В.В.* Теория и практика эволюционного моделирования. – М.: Физматлит, 2003.
7. *Курейчик В.В., Курейчик В.М., Родзин С.И.* Концепция эволюционных вычислений, инспирированных природными системами // Известия ЮФУ. Технические науки. Тематический выпуск "Интеллектуальные САПР". – 2009. – № 4 (93). – С. 16-24.
8. *Курейчик В.М., Курейчик В.В.* Генетический алгоритм разбиения графа // Известия Академии наук. Теория и системы управления. – 1999. – №4.
9. *Лебедев Б.К., Дуккардт А.Н.* Комплексный гибридный генетический алгоритм разбиения // Известия ЮФУ. Технические науки. Тематический выпуск "Интеллектуальные САПР". – 2008. – № 4 (81). – С. 26-32.
10. *Лебедев Б.К., Лебедев О.Б.* Разбиение на основе гибридной многоуровневой адаптации // Известия ЮФУ. Технические науки. Тематический выпуск "Интеллектуальные САПР". – 2008. – № 9 (86). – С. 52-60.
11. *Курейчик В.М., Лебедев Б.К., Лебедев О.Б.* Поисксовая адаптация: теория и практика. – М.: Физматлит, 2006.
12. *Clerc M.* Particle Swarm Optimization. ISTE, London, UK, 2006.
13. *Poli R.* Analysis of the publications on the applications of particle swarm optimisation. Journal of Artificial Evolution and Applications, Article ID 685175, 10 pages, 2008.
14. *Dorigo M. and Stützle T.* Ant Colony Optimization. MIT Press, Cambridge, MA, 2004.

Лебедев Олег Борисович

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

E-mail: lbk@tsure.ru.

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

Тел.: 88634371743.

Кафедра систем автоматизированного проектирования; доцент.

Lebedev Oleg Borisovich

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

E-mail: lbk@tsure.ru.

44, Nekrasovsky, Taganrog, 347928, Russia.

Phone: 88634371743.

Department of Computer Aided Design; associate professor.