В.М. КУРЕЙЧИК, О.Б. ЛЕБЕДЕВ

Таганрогский технологический институт Южного федерального университета

kur@tsure.ru, lbk@tsure.ru

АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧ НА ГРАФАХ НА ОСНОВЕ МЕТОДА МУРАВЬИНОЙ КОЛОНИИ

Рассматриваются новые технологии, принципы и механизмы решения комбинаторных задач на графах, основанные на моделировании процессов адаптивного поведения муравьиной колонии. Предложен и реализован метод решения задачи нахождения максимального паросочетания в графе и родственных ей задач раскраски графа и выделения клик в графе, основанный на выделении в графе независимого подмножества вершин. По сравнению с существующими алгоритмами достигнуто улучшение результатов.

Ключевые слова: коллективный интеллект, адаптивное поведение, самоорганизация, муравьиная колония, оптимизация

Введение

Среди набора комбинаторно-логических задач на графах важное место занимает проблема определения паросочетаний. Паросочетанием графа G=(X,U) называет подмножество таких рёбер U'⊂ U, что любые два ребра uk, ul ∈ U' не имеют общих вершин, т.е. не смежны. Паросочетание максимальной мощности определяется как паросочетание, включающее максимальное число рёбер [1]. Алгоритмы решения данной задачи применяются при проектировании инженерных сетей, коммуникаций, построения систем поддержки принятия решений в неопределенных условиях, проектировании СБИС и т.п. Задачи такого типа относятся к переборным задачам с экспоненциальной временной сложностью. В этой связи разрабатывают различные эвристики для построения алгоритмов с полиномиальной временной сложностью. Существуют алгоритмы определения паросочетаний в графе, основанные на использовании потоков в сетях [1,2] , имитационного моделирования [3], генетического поиска [4] и других эвристиках, которые обеспечивают приемлемые результаты при решении задач малой и средней сложности. Часто эта процедура используется в итерационных структурах. Это предъявляет повышенные требования к качеству и времени решения задачи нахождения максимального паросочетания. Возникшие потребности в решении задач большой и очень большой размерности является побудительным мотивом исследований и разработок новых эффективных алгоритмов. Анализ литературы показывает, что наиболее успешными в этих условиях являются математические методы, в которых заложены принципы природных механизмов принятия решений [5, 6, 7]. К таким методам можно отнести, прежде всего, методы моделирования отжига [8], метод эволюционного моделирования [9], генетические алгоритмы [10], эволюционной адаптации [11, 12], алгоритмы роевого интеллекта [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.

В работе излагается методика решения задачи нахождения максимального паросочетания в графе и родственных ей задач раскраски графа и выделения клик в графе, основанная на моделировании адаптивного поведения муравьиной колонии.

^ Постановка задачи нахождения паросочетания в графе

Пусть дан граф G=(X,U) (рис. 1). U={ui| i=1,2,…,9}. Паросочетание такого графа определяется как множество рёбер, не имеющих общих вершин. Например: паросочетание P={u1, u4, u7, u9}. Построим граф Gd = (U,V) – двойственный для графа G. Вершины графа Gd – соответствуют рёбрам графа G. Пара вершин (ui, uj) в графе Gd связаны ребром в том и только в том случае, если в графе G пара рёбер (ui, uj) смежны, т.е. инциденты одной вершине.

Множество X0 вершин графа G = (X,U) называется внутренне устойчивым, если любые две вершины xi X0 и xj X0 не являются смежными. Максимальное число вершин во внутренне устойчивом множестве графа G называется числом внутренней устойчивости и обозначается как α(G). Иногда число внутренней устойчивости называют также числом независимости графа G.

Рис. 1. Пример графа
Для примера, приведенного на рис.1, двойственный граф Gd имеет вид, представленный на рис.2.

Рис. 2. Двойственный граф Gd
Подмножество вершин P={u1, u4, u7, u9} является внутренне устойчивым, т.к. любые две вершины подмножества P не смежны. Таким образом, паросочетанию в графе G соответствует внутренне – устойчивое подмножество двойственного графа Gd. Максимальному по мощности паросочетанию в графе G соответствует предельное внутренне – устойчивое подмножество (содержащее наибольшее число вершин) двойственного графа Gd.

Раскраской графа называется такое приписывание цветов его вершинам, что никакие две смежные вершины не получают одинакового цвета. Минимальное число цветов, в которое можно раскрасить граф G, называется хроматическим числом и обозначается χ(G). Если в графе G выделить s непересекающихся друг с другом внутренне – устойчивых подмножеств вершин, то граф можно раскрасить в s цветов. Другими словами, задача раскраски графа сводится к задаче формирования в графе G непересекающихся внутренне – устойчивых подмножеств вершин.

Рассмотрим задачу о клике. Кликой графа G называется максимальное по включению множество X0 вершин графа, любые две из которых являются смежными. Нетрудно видеть, что при переходе от графа G к его дополнению Gк каждая клика в G переходит в независимое множество в Gк. Отсюда следует, что задача выделения клики в графе G сводится к задаче выделения независимого множества вершин в графе Gк, являющегося дополнением графа G.

Таким образом, в основе процедур построения максимального паросочетания, раскраски графа, выделения в графе клик лежит одна общая процедура формирования в графе G(X, U) внутренне – устойчивого множества вершин X1 ⊆ X.
Механизмы выделения в графе независимого подмножества

вершин на основе моделирования адаптивного

поведения муравьиной колонии
Пусть дан граф G(X, U), где X – множество вершин, |X| = n, U – множество ребер. Сформулируем задачу формирования в графе G(X, U) внутренне – устойчивого множества вершин X1 $\subset$ X как задачу разбиения.

Необходимо разбить множество X на два непустых и непересекающихся подмножества X1 и X2, таких, что любые две вершины xi$\in$X1 и xj$\in$X1 не являются смежными, X1$\cup$X2 =X, X1 $\cap$ X2=$\emptyset$. Пусть |X1| = n1, |X2| = n2, n1 + n2= n. Критерий оптимизации – число вершин F= n1 в подмножестве X1. Цель оптимизации максимизация критерия F.

Для поиска решения задачи используется полный граф решений R(X, E), где E – множество всех ребер полного графа, связывающих множество вершин X. В процессе решения участвует коллектив муравьев (агентов). Каждый из агентов формирует свое решение – свое множество X1k, где k – номер агента. Формирование множества X1k осуществляется последовательно (пошагово). На каждом шаге t у k-ого агента есть список вершин, уже включенных в формируемое множество – X1k(t) и список оставшихся (свободных) вершин Xck(t), X1k(t)$\cup$Xck(t)=X. На первом шаге в каждое формируемое множество X1k(t), где t = 1, включается вершина графа G, причем вершины графа G распределяются по всем множествам X1k(1) равномерно, то есть в каждом множестве X1k(1) своя начальная вершина, ($\forall$i,j) [X1i(1)$\cap$X1j(1)=$\emptyset$].Такое распределение необходимо, чтобы все вершины графа G имели одинаковые шансы быть отправной точкой при формировании узла X1. В модификациях алгоритма использовались также γ·n муравьев, причем каждая группа из γ муравьев используют в качестве начального одно и то же X1i(1).На конечном шаге t = n1 k-м агентом будет сформировано множество X1k(n1) = X1k, |X1(n1)| = n1 такое, что любые две вершины xi $\in$ X1k и xj $\in$ X1k в графе G(X, U) не являются смежными. Шаг t = n1 является конечным, если после его выполнения не существует вершины xi $\in$ Xck такой, чтобы xi не была смежной с любой вершиной xi $\in$ X1k.

Моделирование поведения муравьёв в процессе формирования каждым из них своего подмножества X1k связано с распределением и учетом количества феромона на множестве ребер графа R. На начальном этапе на всех ребрах графа R откладывается одинаковое (небольшое) количество феромона ξ/m, где m = |E|. Вероятность включения в формируемое отдельным муравьем множество X1k(t), вершины xj□ Xck(t), не имеющей в графе G связей с вершинами множества X1k(t) , пропорциональна суммарному количеству феромона на ребрах, связывающих вершину xj с X1k(t) в графе R. Количество откладываемого феромона пропорционально числу вершин сформированного узла X1k(t). Чем больше мощность X1k, тем больше феромона будет отложено на каждом рёбре полного подграфа R1к□ R, построенного на вершинах узла X1k следовательно, большее количество муравьёв будет включать вершины узла X1k при синтезе собственного внутренне – устойчивого подмножества. Для избегания преждевременной сходимости используется отрицательная обратная связь в виде испарения феромона.

Процесс поиска решений итерационный. Каждая итерация l 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 ферромоны откладываются агентом на ребрах после полного формирования решения.

На первом этапе каждой итерации каждый k-ый муравей формирует свое собственное подмножество X1k. Процесс построения подмножества X1k пошаговый. На каждом шаге агент применяет вероятностное правило выбора следующей вершины для включения ее формируемое множество X1k(t).

Первый этап осуществляется следующим образом. Формируется подмножество X*ck(t)□ Xck(t), такое, что ни один из элементов xi□X*ck(t) не имеет связей ни с одним из элементов xj□X1k(t). Агент просматривает все вершины X*ck(t). Для каждой вершины xi □ X*ck(t) рассчитываются два параметра:

fik – суммарное количество феромона на ребрах графа R, связывающих xi□X*ck(t) с вершинами подмножества X1k(t);

sik – число связей на графе G между xi и X*ck(t).

По формуле (1) – при мультипликативной свертке, либо по формуле (2) – при аддитивной свертке определяется потенциальная стоимость Fik связей xi□X*ck(t) с X1k(t):

$Fik=(fik)\alpha/(sik+1)\beta$, (1)

$Fik=(fik)\alpha+(1/(sik+1))\beta$, (2)

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

Вероятность Pik(t) включения вершины xi□X*ck(t) в формируемое подмножество X1k(t) определяется следующим соотношением:

$Pik(t) = Fik / \sum i\ Fik$. (3)

Агент с вероятностью Pik(t) выбирает одну из вершин, которая включается в подмножество X1k(t) и исключается из подмножества Xck(t).

При α = 0 наиболее вероятен выбор вершины xi минимально связанной с вершинами подмножества X*ck(t),что косвенно способствует тому, что на последующем шаге среди X*ck(t) будут вершины, не связанные с X1k(t), и следовательно |X1k(t)| может расти.

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

Поэтому необходим компромисс между этими величинами, который находится экспериментально.

После формирования муравьями узлов (каждый муравей – свой узел X1k за nk шагов), на втором этапе итерации, каждый муравей откладывает феромон на рёбрах полного подграфа R1к □ R, построенного на вершинах узла X1k.

Количество феромона □hk(l), откладываемое k-ым муравьем на каждом ребре подграфа R1к □ R, построенного на l-й итерации, определяется следующим образом:

□hk(l)= δ·Dk(l), (4)

где l – номер итерации, δ – общее количество феромона, откладываемое k-м муравьем на ребрах подграфа R1к R, Dk(l) – число вершин подмножества X1k, сформированного k-ым муравьем на l-ой итерации. (Другими словами, целевая функция для данного решения.) Обозначим как ϕij(l) суммарное количество феромона, отложенного на дуге (i,j) всеми муравьями на l-й 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 (5).
hij(t +1) = hij(t)·(1 – ρ)+ ϕij(l), (5)
где hij(t) – уровень феромона на ребре (i, j), ρ – коэффициент обновления. После выполнения всех действий на итерации находится агент с лучшим решением, которое запоминается. Далее осуществляется переход на следующую итерацию.

Временная сложность этого алгоритма зависит от времени жизни колонии l (число итераций), количества вершин графа n и числа муравьев m, и определяется как O(l*n2*m).

Алгоритм выделения в графе независимого подмножества вершин 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 графа R(X,E) откладывается начальное количество феромона.

Задаются значения параметров: α, β. Выбирается формула определения потенциальной стоимости Fik(t) связей ((1) или (2)).

4. Задается число итераций Nl.

5. l=1.

6. На первом этапе каждой итерации на графе поиска решений R(X,E) каждым агентом ak строится маршрут Mk(l) и формируется решение Pk(l).

7. Для каждого решения Pk(l), находится значение целевой функции Dk(l).

8. Каждый муравей откладывает феромон на рёбрах полного подграфа R1к(l)⍰R, построенного на вершинах узла X1k(l).Количество феромона откладываемого k-ым агентом прямо пропорционально числу Dk(l) вершин подмножества X1k, сформированного k-ым муравьем на l-ой итерации.

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

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

Временная сложность этого алгоритма на одной итерации определяется как O(n2), 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).

После формирования в графе G(X, U) внутренне – устойчивого множества вершин X1 ⍰ X.для построения паросочетания или выделения в графе клики осуществляется переход от графа G к исходному графу Gи.

При этом при построения паросочетания граф G рассматривается как двойственный исходному графу Gи, а при выделении клики граф G рассматривается в как дополнительный исходному графу Gи.

При решении задачи раскраски графа подмножество X1 окрашивается в один цвет и исключается из X .Далее выполняются аналогичные действия, пока не будут раскрашены все вершины.

Алгоритм разбиения был реализован на языке Си++. 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. 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 %.

Выводы

На основе сравнительного анализа существующих подходов и методов для решения комбинаторных задач на графах использованы мультиагентные методы интеллектуальной оптимизации, базирующиеся на моделировании адаптивного поведения муравьиной колонии.

В отличие от канонической парадигмы муравьиного алгоритма агент на графе решений R(X,E) совершает перемещение не от вершины к вершине, а от группы вершин к вершине, при этом главная цель заключается в формировании подмножеств вершин (подграфа), а не построение маршрута. Это расширяет область применения и круг решаемых задач.

Использован комбинированный подход к формированию оценки, характеризующей выбор вершины для включения в формируемое подмножество вершин.

Предложенный алгоритм показал свою эффективность.

Программа дальнейших исследований будет нацелена на поиск подходов для преодоления локального барьера, основанных на сочетании различных видов эволюции, в частности, муравьиного и эволюционного алгоритма нахождения максимального паросочетания, рассмотренного в работе [11].

^ Список литературы

Андерсон Д. Дискретная математика и комбинаторика. М.: Вильямс, 2003.

Кормен К., Лейзерсон Ч., Ривест Р. Алгоритмы построение и анализ. – М.: МЦМНО. 2000.

Свами М., Тхуласираман К. Графы, сети и алгоритмы. – М.: Мир. 1984.

Курейчик В.В., Курейчик В.М. Генетический алгоритм определения паросочетаний графа. // Труды 10-й международной конференции “Knowledge-dialogue-solution”, Варна, Болгария, 2003. С. 246–251.

Di Caro G., Ducatelle F., Gambardella L.M. AntHocNet: An adaptive nature-inspired algorithm for routing in mobile ad hoc networks. // European Transactions on Telecommunications, 16(5): 443–455. 2005.

Engelbrecht A.P. Fundamentals of Computational Swarm Intelligence. John Wiley & Sons, Chichester, UK. 2005.

МакКоннелл Дж. Основы современных алгоритмов. Москва, Техносфера, 2004.

Wong D.F., Leong H.W., and Lin C.L. Simulated Annealing for VLSI Design. Boston, MA : Kluwer Academic. 1988.

Емельянов В.В., Курейчик В.М., Курейчик В.В. Теория и практика эволюционного моделирования. М.: Физматлит. 2003.

Mazumder P., Rudnick E. Genetic Algorithm For VLSI Design, Layout & Test Automation. India, Pearson Education. 2003.

Лебедев Б.К., Лебедев О.Б. Эволюционный алгоритм нахождения максимального паросочетания. 3-й Международный НТС “Интегрированные модели и мягкие вычисления в искусственном интеллекте”. М: Изд-во Физматлит. 2005. С. 274–280.

Курейчик В.М., Лебедев Б.К., Лебедев О.Б. Поисковая адаптация: теория и практика. М.: Физматлит. 2006.

Clerc M. Particle Swarm Optimization. ISTE, London, UK. 2006.

Dorigo M. and Stützle T. Ant Colony Optimization. MIT Press, Cambridge, MA. 2004.