

УДК 681.3

В.М. Курейчик, А.А. Кажаров**ПРИМЕНЕНИЕ ПЧЕЛИНОГО АЛГОРИТМА ДЛЯ РАСКРАСКИ ГРАФОВ***

Работа посвящена решению NP-трудной графовой задачи о раскраске. Предлагается решить проблему на подходе, основанном на роевом интеллекте – пчелиный алгоритм. Основная идея алгоритма – моделирование естественного роя пчел. Разработана программная среда, реализующая предложенный алгоритм, а также другие методы для сравнений. Экспериментальные исследования доказали эффективность пчелиного алгоритма при решении задачи о раскраске. Приемлемое решение для графов размерностью до 200 вершин находится в течение нескольких секунд на ЭВМ с тактовой частотой 2.5 ГГц.

Пчелиные алгоритмы; роевой интеллект; задача раскраски; NP-задача; графовая задача.

V.M. Kureichik, A.A. Kazharov**GRAPH COLORING SOLUTION BASED ON BEE COLONY OPTIMIZATION**

This paper is dedicated to the solving of NP-complete task – graph coloring problem. It is proposed to solve the problem on the approach based on Swarm Intelligence – bee algorithm. The main idea of the algorithm – modeling of natural swarm of bees. A software environment that implements the proposed algorithm, as well as other methods for comparison. Experimental studies have proven the effectiveness of bee algorithm for solving the problem of coloring. An acceptable solution is found for 200-vertexes graph within a few seconds on modern computers.

Bee Colony Algorithms; swarm intelligence; the problem of graph coloring; NP task; graph problem.

Введение. Разнообразные задачи, возникающие при планировании производства, составлении графиков осмотра, хранения и транспортировке товаров и т.д., могут быть представлены часто как задачи теории графов, тесно связанные с так называемой «задачей раскраски» [1]. Данная задача относится к классу NP-трудных задач. Данный класс задач известен тем, что нельзя найти точного решения за разумное время, так как пространство поиска решений увеличивается в экспоненциальной зависимости от входных данных. В связи с этим были разработаны целый ряд алгоритмов, которые находят приближенные к глобальному оптимуму решения. Одним из перспективных направлений в решении NP-трудных задач являются роевые алгоритмы. К таким алгоритмам относятся муравьиные, роя частиц, пчелиные и т.д.

Постановка задачи. Граф G называют l -хроматическим, если его вершины могут быть раскрашены с использованием l цветов (красок) так, что не найдется двух смежных вершин одного цвета. Наименьшее число l такое, что граф G является χ -хроматическим, называется хроматическим числом графа G и обозначается $\chi(G)$. Задача нахождения хроматического числа графа называется задачей о раскраске графа. Соответствующая этому числа раскраска вершин разбивает множество вершин графа на l подмножеств, каждое из которых содержит вершины одного цвета. Эти множества являются независимыми, поскольку в пределах одного множества нет двух смежных вершин [1].

Пример раскраски графа представлен на рис. 1.

В данном примере вершины раскрашены четырьмя цветами, где краски обозначены буквами A, B, C, D .

* Работа выполнена при поддержке: РФФИ (грант № 09-01-00492), г/б № 2.1.2.1652.

Раскраской вершин графа называется разбиение множества вершин X на l непересекающихся классов (подмножеств) [2]:

$$X_1, X_2, \dots, X_l; X = \bigcup_{i=1}^l X_i; X_i \cap X_j = \emptyset; i, j \in I = \{1, 2, \dots, l\}, \quad (1)$$

таких, что внутри каждого подмножества X_i не должно содержаться смежных вершин.

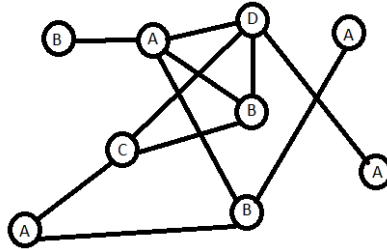


Рис. 1. Пример раскраски графа

Если каждому подмножеству X_i поставить в соответствие определенный цвет, то вершины этого подмножества можно окрасить в один цвет, вершины другого подмножества X_j – в другой цвет, и т.д. до раскраски всех подмножеств. В этом случае граф называется l -раскрашиваемым. Наименьшее число подмножеств, на которое разбивается граф при раскраске, называется хроматическим числом $\chi(G)$. Хроматическое число обычно определяют аналитически с помощью методов линейного программирования [2].

Последовательный алгоритм. Определим хроматическое число и раскраску графа изображенного на рис. 2.

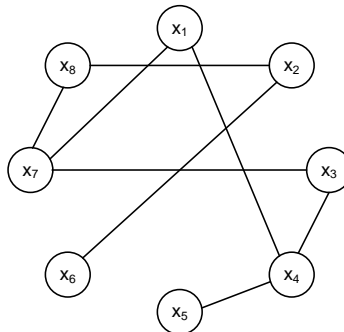


Рис. 2. Граф G

Воспользуемся последовательным алгоритмом раскраски графа [2]:

1. Подсчитываем локальные степени вершин графа G и составляем список вершин в порядке убывания их локальных степеней:

$\rho(x_i)$	3	3	3	2	2	2	2	1
x_i	4	7	8	1	2	3	5	6

2. Выбираем из списка вершину x_4 : $\Pi_1 = \{x_4\}$.

3. Просматриваем список на предмет нахождения несмежных вершин. Выбираем ближайшую несмежную вершину. Это вершина x_7 : $\Pi_1 = \{x_4, x_7\}$.

4. Продолжаем процесс просмотра. Выбираем вершину x_2 - несмежную вершинам x_4 и x_7 , включенным в подмножество Π_1 . Теперь $\Pi_1 = \{x_4, x_7, x_2\}$.

5. Просматриваем список дальше, и, так как в списке больше нет вершин, несмежных вершинам подмножества Π_1 , окрашиваем вершины этого подмножества в первый цвет и удаляем их из списка. После удаления список примет вид:

$$\begin{array}{c|c|c|c|c|} \rho(X_i) & 3 & 2 & 2 & 1 \\ \hline X_i & 8 & 1 & 3 & 5 & 6 \end{array}$$

6. Выбираем вершину x_8 : $\Pi_2 = \{x_8\}$.

7. Выбираем вершину x_1 : $\Pi_2 = \{x_8, x_1\}$.

8. Выбираем вершину x_3 : $\Pi_2 = \{x_8, x_1, x_3\}$.

9. Выбираем вершину x_6 : $\Pi_2 = \{x_8, x_1, x_3, x_6\}$.

10. В списке нет больше вершин, несмежных вершинам подмножества Π_2 . Окрашиваем вершины этого подмножества во второй цвет и удаляем их из списка.

11. В списке остается только одна вершина – x_5 . Эта вершина окрашивается в третий цвет.

Таким образом, для раскраски графа нам потребовались три краски, следовательно, хроматическое число: $\chi(G) = 3$. Алгоритм реализует жадную стратегию. Существуют различные эвристические подходы [2]. Однако с увеличением размерности задачи их эффективность в плане качества решения значительно ухудшается.

Пчелиный алгоритм. Пчелиный алгоритм относится к классу роевых алгоритмов, который, в свою очередь, является подклассом биоинспирированных алгоритмов [3]. Данный алгоритм моделирует поведение пчел в естественной среде. Идея пчелиного алгоритма заключается в том, что все пчелы на каждом шаге будут выбирать как элитные участки для исследования, так и участки в окрестности элитных, что позволит, во-первых, разнообразить популяцию решений на последующих итерациях, во-вторых, увеличить вероятность обнаружения, близких к оптимальным, решений [4].

Приведем основные понятия пчелиного алгоритма:

- 1) источник нектара (цветок, участок);
- 2) фуражиры (рабочие пчелы);
- 3) пчелы-разведчики.

Источник нектара характеризуется значимостью, определяемой различными параметрами. Фуражиры закреплены за отдельным участком. На более значимых участках работают пчелы-разведчики. Количество всех пчел в «элитных» источниках нектара больше, чем на остальных. Среднее количество разведчиков в рое составляет 5-10 %. Вернувшись в улей, пчелы обмениваются информацией посредством танцев на, так называемой, закрытой площадке для танцев [5].

Пусть решение представляет собой вектор H . Областью поиска нектара для пчел будет являться пространство поиска решений, размерностью $n!$ (количество всех возможных перестановок вектора H). Расположение источника нектара характеризуется конкретной перестановкой H , решением. Таким образом, координатами источника является решение H . Количество нектара на источнике обратно пропорционально целевой функции (ЦФ). Участок имеет размеры, где размер – количество решений «близких» к H . Близость между векторами определяется значением расстояния Хемминга. К примеру, решения $\{5,2,7,3,4,1,6\}$ и $\{5,4,7,3,2,1,6\}$ являются «близкими», т.е. в пространстве поиска они располагаются рядом, находясь на одном «участке».

i	1	2	3	4	5	6	7	8
Решение 1	1	8	2	7	3	5	4	6
Решение 2	1	3	2	7	8	5	4	6

Приведём описание алгоритма пчёл [5].

1. Создание популяции пчёл.
2. Оценка ЦФ пчёл в популяции.
3. Выбор участков для поиска в их окрестности.
4. Отправка пчёл-разведчиков.
5. Выбор пчёл с лучшими значениями ЦФ с каждого участка.
6. Отправка рабочих пчёл, осуществляющих случайный поиск и оценка их ЦФ.
7. Формирование новой популяции пчёл.
8. Если условие останова не выполняется, то п. 2.
9. Конец работы алгоритма.

Временная сложность пчелиного алгоритма: $O(n^2)$.

Рассмотрим все эти определения на примере задачи поиска глобального минимума функции

$$f(x, y) = x^2 + y^2,$$

модель которого приведена на рис. 3.

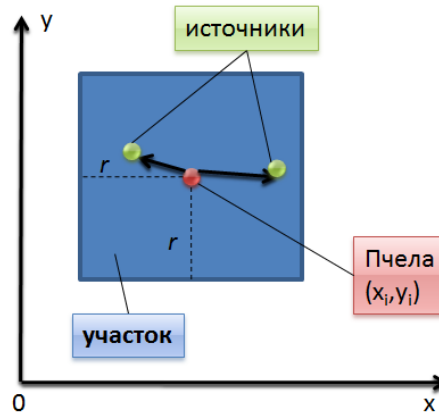


Рис. 3. Модель участка в пространстве поиска

При решении задач нахождения глобальных экстремумов сложных многомерных функций границы участка определяются как

$$[x_1-r; x_1+r][x_2-r; x_2+r] \dots [x_n-r; x_n+r],$$

где x_i – координата пчелы (параметр функции);

r – радиус (размер) участка;

n – количество параметров функции, в данном случае – 2.

В данном примере границы определяются как $[x-r; x+r][y-r; y+r]$. Пусть пространство поиска представляет собой двумерное пространство. Как видно из рисунка и ограничений, пчела может перемещаться в другие источники нектара только в рамках одного участка, ограниченного радиусом. Количество нектара обратно пропорционально значению функции в координатах x, y . Другими словами, агент пытается найти другое решение не намного отличающееся от предыдущего.

Представим решение задачи раскраски в виде вектора H длиной n , где n – число вершин в графе. Областью поиска нектара для пчел будет являться пространство поиска решений, размерностью $n!$ (количество всех возможных перестановок вектора H). Алгоритм декодирования решения использует «жадную» эвристику, т.е. основан на локально-оптимальных шагах. В «жадном» алгоритме (greedy algorithm) всегда делается выбор, который кажется самым лучшим в данный момент, т.е. производится локально оптимальный выбор в надежде, что он приведет к оптимальному решению глобальной задачи [7].

Рассмотрим алгоритм.

1. $r = 0, i = 1$.
2. $j = 1$.
3. Если краска j не занята смежными к i вершинами,
 - а) то $P_i = j$;
 - б) иначе $r = r + 1, P_i = r$.
4. $j = j + 1$.
5. Если $j \leq r$, то переход к п. 3.
6. $i = i + 1$.
7. Если $i \leq n$, то переход к п. 2.
8. Конец.

В алгоритме: n – число вершин в графе, r – целевая функция или количество красок, P_i – цвет i -й вершины.

Пусть расположение источника нектара характеризуется некоторой перестановкой H . Тогда координатами источника является само решение H . Количество нектара на источнике обратно пропорционально ЦФ, т.е. чем путь коммивояжера, тем больше нектара находится на источнике, кодирующий это решение. Участок имеет размеры, где размер – количество решений «близких» к H . Близость между векторами определяется значением расстояния Хемминга между ними. К примеру, решения $\{4,2,5,6,3,1\}$ и $\{4,6,5,2,3,1\}$ являются «близкими», т.е. в пространстве поиска они располагаются рядом, находятся на одном «участке» [5]. Рассмотрим пример перемещения пчелы для данной задачи на рис. 4.

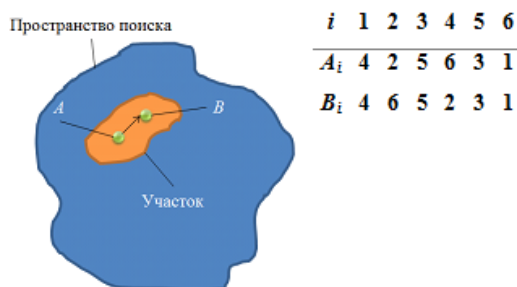


Рис. 4. Модель участка в пространстве поиска для задачи раскраски графа

Приведём описание алгоритма пчёл.

1. Генерация участков для поиска нектара.
2. Оценка полезности участков.
3. Выбор участков для поиска в их окрестности.
4. Отправка фуражиров.
5. Поиск в окрестностях источников нектара.
6. Отправка пчёл-разведчиков.
7. Случайный поиск.
8. Оценка полезности новых участков.

9. Если условие останова не выполняется, то переход к п. 2.
10. Конец работы алгоритма.

Таким образом, ключевой операцией алгоритма пчёл является совместное исследование перспективных областей и их окрестностей. В конце работы алгоритма популяция решений будет состоять из двух частей: пчёлы с лучшими значениями ЦФ элитных участков, а также группы рабочих пчёл со случайными значениями ЦФ [5].

Зависимость временной сложности пчелиного алгоритма от числа вершин – $O(n^2)$.

Представленное кодирование позволяет пчелиному алгоритму решать широкий класс графовых задач [8].

Экспериментальные исследования. В процессе экспериментальных исследований был реализован пчелиный алгоритм. Тестирование разработанного алгоритма проводилось на различных графах размерностью до 1000 вершин. Сравнения проводились с генетическим [9] и эволюционным алгоритмами [10,11]. На тестах с графами размерностью до 80 вершин генетический и пчелиный алгоритмы показывали схожие результаты. На больших размерностях предложенный метод находил лучшие решения, чем генетический и эволюционный.

Заключение. В ходе проделанной работы был реализован пчелиный алгоритм, который находит «хорошее» решение за приемлемое время. Алгоритм легко адаптирован к задаче раскраски и может быть также применен для других графовых задач. Планируется разработать пчелиные алгоритмы для решения других наиболее популярных NP-трудных графовых задач.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Кристофидес Н.* Теория графов. Алгоритмический подход. – М.: Изд-во «Мир», 1978.
2. *Гладков Л.А., Курейчик В.В., Курейчик В.М.* Дискретная математика: Теория графов. – Таганрог: Изд-во ТТИ ЮФУ, 2010. – 162 с.
3. *Курейчик В.М.* Биоинспирированный поиск с использованием сценарного подхода // Известия ЮФУ. Технические науки. – 2010. – № 7 (108). – С. 7-12.
4. *Кажаров А.А.* Решение задачи разбиения графа на основе биоинспирированных алгоритмов // Труды международных научно-технических конференций «Интеллектуальные системы» (AIS'10) и «Интеллектуальные САПР». – М.: ФИЗМАТЛИТ, 2010.
5. *Субботин С.А., Олейник Ан.А., Олейник Ал.А.* PSO-метод // Интеллектуальные мультиагентные методы (Swarm Intelligence). – 2006. – № 3. – С. 55-70.
6. *Курейчик В.В., Полупанова Е.Е.* Эволюционная оптимизация на основе алгоритма колонии пчёл // Известия ЮФУ. Технические науки. – 2009. – 12 (101). – С. 41-46.
7. *Cormen T., Leiserson Ch., Rivest R., Stein C.* Introduction to algorithms. Second edition. The MIT Press edition, Cambridge, Massachusetts, London, England, 2002.
8. *Кирсанов М.Н.* Графы в Maple. Задачи, алгоритмы, программы. – М.: ФИЗМАТЛИТ, 2007.
9. *Гладков Л.А., Курейчик В.М., Курейчик В.В.* Генетические алгоритмы. – Ростов-на-Дону: ООО «Ростиздат», 2004.
10. *Holland John H.* Adaptation in natural an artificial systems. The MIT Press edition, Massachusetts, London, England, 1992.
11. *Курейчик В.М., Курейчик В.В., Родзин С.И.* Концепции эволюционных вычислений, инспирированных природными системами // Известия ЮФУ. Технические науки. – 2009. – № 4 (93). – С. 16-24.

Курейчик Виктор Михайлович

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

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

E-mail: kur@tsure.ru.

Тел.: 88634393260.

Заместитель руководителя по научной и инновационной деятельности; профессор.

Кажаров Аскер Артурович

E-mail: persianland@mail.ru.

Тел.: +79064233689.

Кафедра систем автоматизированного проектирования; студент.

Kureichik Victor Michylovich

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

E-mail: kur@tsure.ru.

44, Nekrasovskiy, Taganrog, 347928, Russia.

Phone: +78634393260.

The Deputy the Head on Scientific Work and Innovations; Professor.

Kazharov Asker Arturovich

E-mail: persianland@mail.ru.

Phone: +79064233689.

Department of Computer Aided Design; Student.