УДК 519.173

ИСПОЛЬЗОВАНИЕ АЛГОРИТМА МУРАВЬИНОЙ КОЛОНИИ ДЛЯ РЕШЕНИЯ ЗАДАЧИ ШТЕЙНЕРА

А.И. Ольшевский, М.Ю. Лефаров

Донецкий национальный технический университет
кафедра программного обеспечения интеллектуальных систем

В работе рассматриваются вопросы решения задачи Штейнера с помощью алгоритма муравьиной колонии. Для анализа времени построения маршрута исследуются зависимости настройки параметров системы и выбора стратегии поиска. Описываются предложенные подходы и приводятся результаты численных экспериментов при задании различных исходных данных.

Введение

С появлением коммутируемых сетей появилась проблема маршрутизации в них. Для минимизации суммарной длины ребер графа в кратчайших связывающих деревьях предложено при соединении множества деревьев использовать дополнительные точки (вершины). Задачу построения минимального дерева при помощи введения дополнительных точек называют задачей Штейнера(ST – Steiner Treeproblem) [1].

В связи с возможностью быстрого получения набора различных деревьев Штейнера наиболее эффективным представляется применения методов, в которых заложены принципы природных механизмов принятия решений. Среди так называемых «Soft computing techniques» выделяются муравьиные алгоритмы (Ant Colony Optimization – ACO, Ant Systems–AS) моделируют поведение муравейника.

Особенно эффективны муравьиные алгоритмы при динамической оптимизации процессов в распределенных нестационарныхсистемах,например,трафиковвтелекоммуникационны хсетях.

1 Математическая постановка задачи

В данной работе рассматриваются связные неориентированные графы без петель и кратных ребер.

Дана сеть, представленная в виде ненаправленного графа $G = (V, E)$, где V - набор узлов или вершин графа, а E - набор отрезков, соединяющих вершины, это ребра или связи графа. Граф G является конечным, так как число его вершин конечно. Две точки v_i, v_j называются смежными, если между ними есть ребро. Маршрут в графе – это чередующаяся последовательность вершин и ребер, в которой любые два соседних элемента смежны. Длина маршрута – это количество ребер в маршруте (с повторениями). Связный граф – это такой граф, в котором все вершины связаны. Информация, содержащаяся в некотором графе, может быть представлена в алгебраическом виде посредством матриц.

Матрица стоимости – это такая матрица W порядка (NЧN), где N – это количество узлов в графе, а w_{ij} показывает стоимость использования связи $(i, j) \in E$.

Матрица стоимости показывает структуру графа. Значения в матрице на пересечениях столбцов и строк при $i = j$ равно нулю, что обусловлено отсутствием в графе петель. Вершины графа G различаются: вершина-центр $S \in E$, вершина- участник $D \in E$ и нейтральная вершина или вершина Штейнера.

Задача построения оптимальной топологии для графа такого вида сводится к нахождению такого дерева T графа G с корнем в S, соединяющего всех членов набора D так, что маршрут дерева T будет минимальным.

2 Адаптация метода к решаемой задаче синтеза сети

Представим компьютерную сеть как окрестность, в которой взаимодействуют муравьи, при этом узлы этой сети будут являться источниками пищи для муравьев. Узлы соединены линиями связи, на которых муравьи могут оставлять след феромона. Каждая связь в сети характеризуется двумя значения, представляющие интерес при поиске: количество феромонов на ребре и стоимость связи представленной матрицей стоимости. При инициализации муравьи размещаются по узлам-участникам с повторением. Каждый муравей движется от одного узла сети к другому, при этом собирая информацию о ребрах, соединяющих узлы. Муравей следует по маршруту до тех пор, пока не пройдет все вершины-участники и вершину-центр. Для начала, муравей проверяет, есть ли смежные не

пройденные вершины-участники. Если такие были найдены муравей переходит к поиску той вершины, вероятность перехода в которую максимальна. В случае, если встречаются вершины с одинаково малой вероятностью, муравей выбирает в какую переходить случайным образом.

Если не было найдено смежных не пройденных вершин-участников, начинается поиск смежных не пройденных нейтральных вершин.

Если не было найдено смежных не пройденных вершин-участников и нейтральных вершин, поиск следующей вершины осуществляется вышеописанным способ только для пройденных вершин. При этом проверяется, чтобы вершина не образовывала цикл. Это сделано, для исключения ситуации перехода муравья по одному и тому же маршруту.

Каждый муравей продолжает поиск, пока не будет достигнуто условие окончания поиска:

1) муравей прошел все вершины-участники;

2) муравей прошел определенное количество итераций.

После завершения поиска минимального пути муравей проходит проверку на то, по какому условию закончен поиск. В случае, если муравей закончил поиск по второму условию, он помечается как бракованный и для поиска лучшего решения в колонии не учитывается.

Когда все муравьи прошли описанные этапы, среди не бракованных муравьев ищется тот, у которого минимальный маршрут. После этого все муравьи удаляются и на ребрах испаряется значение феромона. Алгоритм переходит к следующей итерации. Снова создаются муравьи и размещаются по вершинам-участником. Алгоритм работает заданное пользователем количество итераций [2].

3 Реализация алгоритма муравьиной колонии

Обобщенный алгоритм муравьиной колонии для данной сети представлен на рисунке 1. Переменной It обозначено количество итераций алгоритма (задается пользователем). NAnt - это количество муравьев, Sbest - это лучшее решение за всю работу алгоритма, а S - это лучшее решение текущей популяции муравьев.

Для того, чтобы определить, прошел ли муравей все вершины была введена функция allnodes. В качестве параметра a передается индекс проверяемого муравья, а в качестве параметра b - количество вершин-участников сети. Функция возвращает значение true, если все вершины-участники пройдены муравьем.

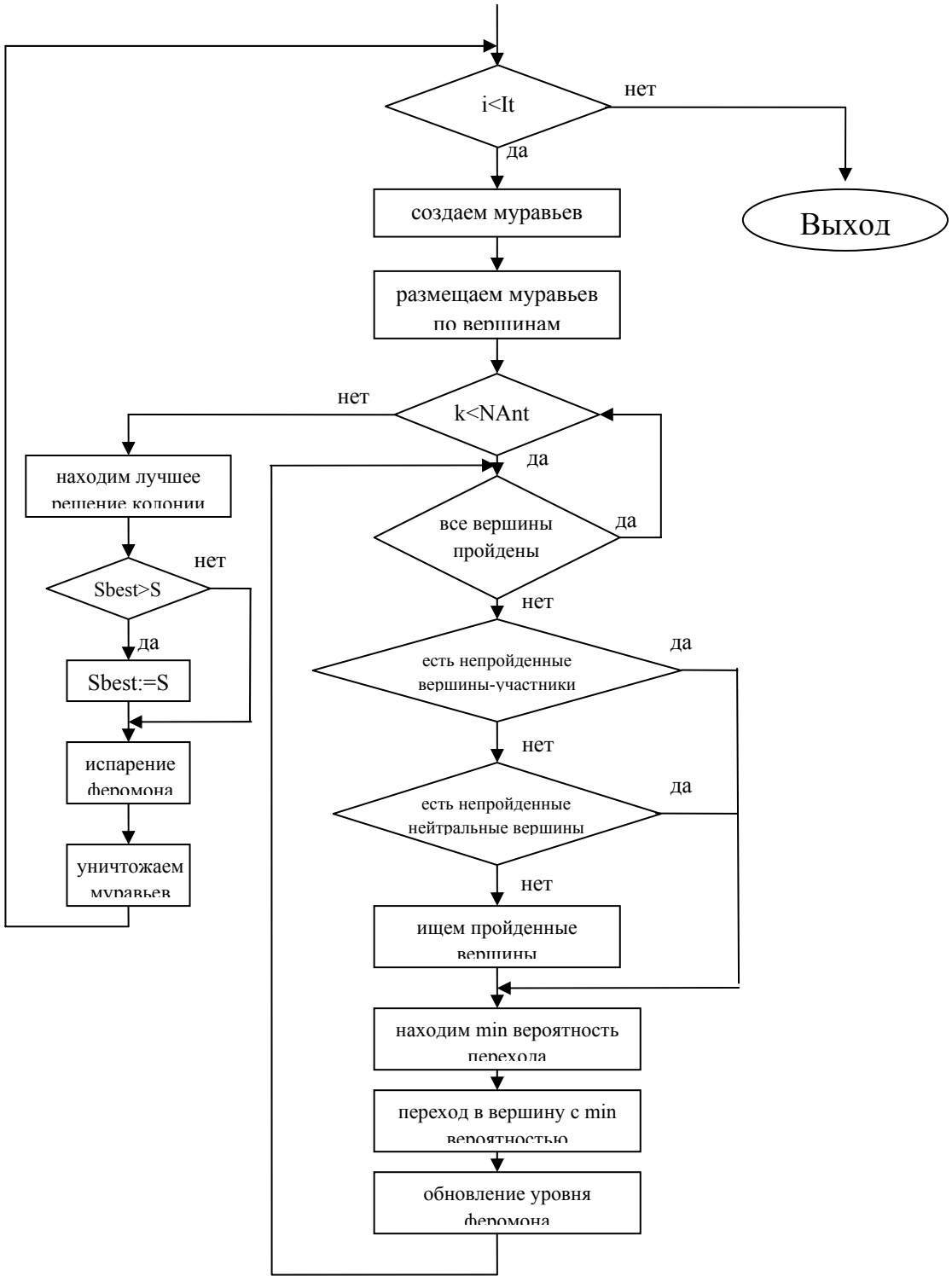

Рисунок 1 – Обобщенный алгоритм муравьиной колонии

Функция onenodeway облегчает переход в висячую вершину-участник. В качестве параметра a передается индекс муравья, параметра b - текущая вершина, т.е. вершина из которой будет

осуществляться переход, параметр c - задаваемое пользователем значение Q, d - количество вершин-участников сети.

Функция wasin используется для определения была ли вершина посещена ранее. В качестве параметра a передается индекс текущего муравья, а параметра b - индекс проверяемой вершины.

Для исключения перехода в нейтральную висячую вершину используется функция no tone neit. В качестве параметра a передается текущая вершина, а параметр b передает индекс проверяемой вершины. Функция возвращает значение true, если проверяемая нейтральная вершина является висячей.

Для проверки, что вершина, в которую может перейти муравей, не создает циклом в его маршруте используется функция cycle. В качестве параметра a передается индекс текущего муравья, а параметра b - индекс проверяемой вершины. Функция возвращает значение true, если проверяемая вершина образует цикл в пути муравья.

При подсчете вероятности перехода необходимо находить сумму ряда. Для этого была введена функция sum_func. Параметры c и f передают индекс текущей вершины и значение параметра Q, а параметры d и e передают значения коэффициентов a и b . Функция возвращает результат суммирования.

4 Результаты численных экспериментов

В качестве среды разработки программного продукта была выбрана среда Borland Delphi, а реализованный алгоритм муравьиной колонии используется как части программного комплекса в «Системе проектирования топологии и оптимизации сетей» [3].

Окно для редактирования параметров алгоритма муравьиной колонии и настроек системы представлено на рисунке 2.

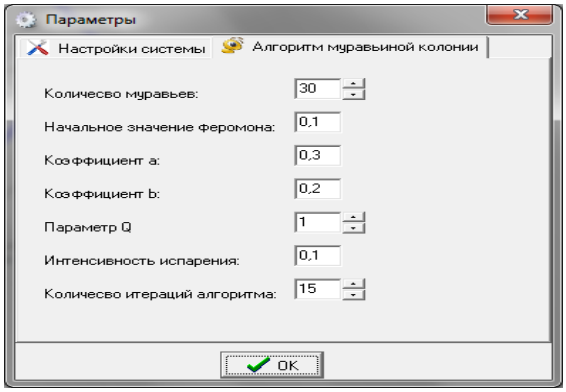

Рисунок 2 – Окно «Параметры»

Программный продукт тестировался на функциональность построения оптимальной топологии сети. Для этого была выбрана модель сети, состоящая из 26 узлов, 9 из которых вершины - участники и один узел-центр. Для этой сети был применен алгоритм муравьиной колонии. Результаты алгоритма тестирования представлены на рисунке 3.

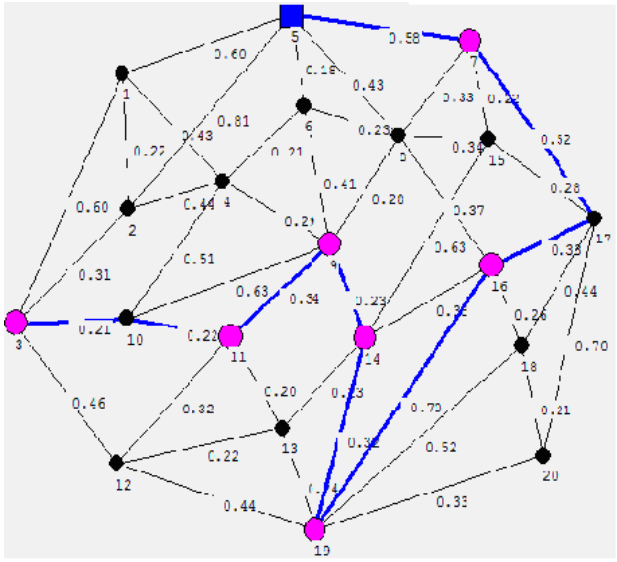

Рисунок 3 –Графический результат тестирования алгоритма

Одним из важных компонентов системы является вывод информации в текстовый процессор MSWord, представленный на рисунке 3.

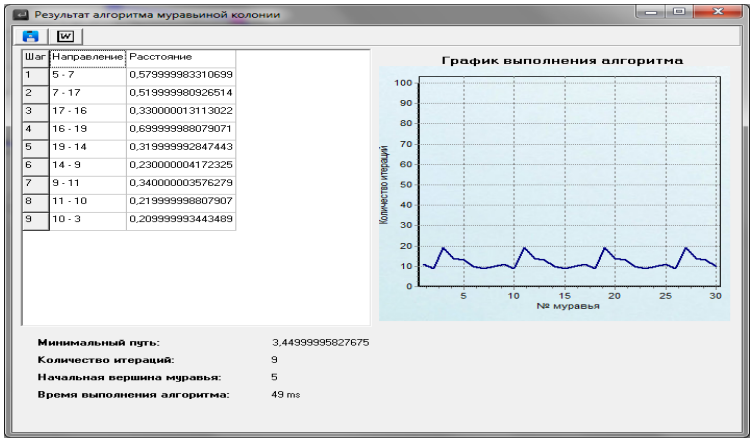

Рисунок 6 – Табличный результат тестирования алгоритма

Выводы

Приведенные исследования показывают, что при решении задачи Штейнера с целью повышения эффективности алгоритма муравьиной колонии особое внимание при программной реализации необходимо уделять настройке системы (выбору параметров и стратегии поиска).

Для решения проблемы маршрутизации сети автором был модифицирован алгоритма муравьиной колонии, который позволил находить близкое к оптимальному решение, с учетом неравнозначности узлов сети.

Практическая значимость работы определяется программной реализацией алгоритма с возможностью анализа результата в графическом и текстовом виде.

Список литературы

1.Guo-Quing Hu. Forest build tree algorithms for multiple destinations // The Potential. №3, 1998.-с. 13-16.

2.Чураков Михаил. Муравьиные алгоритмы /Михаил Чураков, Андрей Якушев. / [Электронный ресурс]. – Режим доступа: http://rain.ifmo.ru/cat/data /theory/unsorted/ant-algo-2006/article.pdf.

3.Ольшевский А.И. Интеллектуальная система проектирования информационных сетей дистанционного обучения на базе ДонГИИИ // Искусственный интеллект.-2007.-№1.- с. 244-249.