Ю. П. Титов, Модификации метода муравьиных колоний для решения задач разработки авиационных маршрутов, *Автомат. и телемех.*, 2015, выпуск 3, 108–124

Автоматика и телемеханика, № 3, 2015

Системный анализ и исследование операций

© 2015 г. **Ю.П. ТИТОВ** (kalengul@mail.ru)
(Московский авиационный институт)

МОДИФИКАЦИИ МЕТОДА МУРАВЬИНЫХ КОЛОНИЙ ДЛЯ РЕШЕНИЯ ЗАДАЧ РАЗРАБОТКИ АВИАЦИОННЫХ МАРШРУТОВ

Исследуется метод муравьиных колоний разработанный для решения задачи коммивояжера. Рассмотрена возможность применения метода муравьиных колоний к другим задачам на графах: к задаче сбора ресурсов при различных ограничениях и условиях и к задаче маршрутизации нескольких транспортных средств, с возможностью выбора места расположения этих средств.

Разработан алгоритм, позволяющий реализовать данный метод. Проведена оценка эффективности алгоритма при решении различных задач.

Алгоритм показал быструю сходимость и близость найденного решения к оптимальному. Данный метод можно рекомендовать для решения большинства задач на графе.

1. Введение

Большинство задач маршрутизации решаются на графах. Графы для задач маршрутизации авиационных средств являются сильносвязанными или полносвязанными. Время перемещения между пунктами определяется в основном расстоянием между ними. Количество дуг в полном графе очень велико поэтому применение жадного алгоритма не представляется возможным.

В последнее время при решении задач комбинаторной оптимизации широко используются методы метаэвристики, в основе которых лежат процедуры случайного поиска, позволяющие находить решения, близкие к оптимальному. Метаэвристически (от англ. metaheuristic, meta — "за пределами" и heuristic — "найти") представляют собой алгоритмы, не имеющие в большинстве случаев строгого доказательства сходимости, но опирающиеся на естественные правила выбора, существующие в объектах живой и неживой природы.

Среди широкого класса метаэвристических методов особое место занимает метод оптимизации муравьиными колониями (ant colony optimization, ACO), предложенный итальянским исследователем Марко Дориго (Marco Dorigo) в 1992 г. [1].

Алгоритм муравьиных колоний можно отнести к многоагентным алгоритмам. Муравьи в данном случае являются агентами, так как они отвечают характеристикам агентов:

1) автономность: агенты, хотя бы частично, независимы;

2) ограниченность представления: ни у одного из агентов нет представления о всей системе или система слишком сложна, чтобы знание о ней имело практическое применение для агента;

3) децентрализация: нет агентов, управляющих всей системой.

Каждый муравей решает только задачу прохождения по выбранному пути. В целом многоагентная система муравейника за счет взаимодействия муравьев-агентов в форме оставления феромонов находит кратчайший путь к источнику пищи.

Ряд исследователей отмечает [2, 3], что на сегодняшний день муравьиные алгоритмы весьма конкурентоспособны по сравнению с другими метаэвристиками и для некоторых задач дают наилучшие результаты.

На практике существует большое количество задач, отличных от задачи коммивояжера. Эти задачи имеют множество особенностей и параметров. В данной статье рассматриваются задачи сбора ресурсов в различных интерпретациях. Большинство задач данного типа решаются с помощью эвристических алгоритмов, соответствующих предметной области. Предложенный ниже алгоритм на основе алгоритма муравьиных колоний может решать подобные задачи в общей форме и легко модифицироваться под конкретные задачи.

2. Описание алгоритма муравьиных колоний для задачи коммивояжера

Перемещения муравьев (агентов) по графу описывается правилом: чем больше феромона на дуге графа, тем больше вероятность агента выбрать этот путь [1].

Вероятность выбора агентом дуги i–j считается по формуле

$$P_{ij,k}(t) = \frac{\tau_{ij}^{\alpha}(t)\cdot\mu_{ij}^{\beta}(t)}{\sum\limits_{l\in J_{i,k}}(\tau_{il}^{\alpha}(t)\cdot\mu_{il}^{\beta}(t))}, \quad j\in J_{i,k} \tag{1}$$
$$P_{ij,k}(t) = 0, \quad j\notin J_{i,k},$$

где i – вершина, в которой находится агент; t – момент времени, в который агент пришел в вершину i; j – следующая на пути агента вершина; k – номер агента; $J_{i,k}$ – множество вершин, в которые агент номер k может попасть из вершины i; $\tau_{ij}(t)$ – количество феромона на дуге i–j в момент времени t; μ_{ij} – длина дуги i–j; α – коэффициент феромонов; β – коэффициент длины; $0\leqslant P_{ij,k}(t)\leqslant 1$.

Количество феромона на дугах постоянно меняется. Агенты, определив решение, оставляют на всех дугах, по которым прошли, количество феромона, которое вычисляется по формуле

$$\Delta\tau_{ij,k}(t) = \frac{Q}{L_k(t)}, \quad (i,j)\in T_k(t), \tag{2}$$
$$\Delta\tau_{ij,k}(t) = 0, \quad (i,j)\notin T_k(t),$$

где $\Delta\tau_{ij,k}(t)$ – изменение феромона на дуге i–j; Q – общее количество феромона у муравья; $L_k(t)$ – длина пройденного k-м агентом пути; $T_k(t)$ – набор дуг, пройденных k-м агентом.

После окончания итерации (т.е. после того, как все агенты завершили проход по графу) определяется количества феромона, которые оставят на своих путях агенты. Количество феромона на дуге i–j вычисляется по формуле

$$\tau_{ij}(t+1) = p \cdot \left(\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij,k}(t) \right), \tag{3}$$

где $\tau_{ij}(t)$ – текущее количество феромона на дуге i–j; p – коэффициент испарения феромона ($0 \leqslant p \leqslant 1$); $\Sigma\Delta\tau_{ij,k}(t)$ – общее количество феромона, которое добавили агенты на дугу i–j на текущей итерации.

При решении задачи коммивояжера необходимо учитывать условие, запрещающее агенту более одного раза заходить в одну и ту же вершину. Выполнение этого условия обеспечивается наличием у агента “памяти”, т.е. каждый агент помнит, по каким вершинам он прошел. Если агент, попав в вершину i, далее не может никуда идти, но он не прошел все вершины, то агент “умирает”, т.е. не оставляет феромона на графе.

Существуют модификации алгоритма муравьиных колоний. Они основаны на увеличении количества оставляемого феромона у агентов, которые нашли самые короткие пути. Если увеличивается феромон только у одного агента, такой алгоритм называется элитным, в случае, когда агентов несколько, ранжированным [4, 5].

3. Алгоритм муравьиных колоний для задачи коммивояжера в псевдокоде

Агентов представим в виде объектов исходя из терминологии объектно-ориентированного программирования. У каждого муравья будем хранить список пройденных им вершин. Отдельно будет храниться указатель на вершину, в которой находится агент.

Тогда алгоритм можно реализовать следующим образом.

Основная программа:

Строка № 01. **Начало**

Строка № 02. **Цикл № 1: Пока** не нашли оптимальный путь, **выполнить**

Строка № 03. Создание популяции агентов

Строка № 04. Установка агента в начальную вершину

Строка № 05. **Цикл № 2:** по всем агентам

Строка № 06. **Цикл № 3: Пока** агент не закончит перемещение, **выполнить**

Строка № 07. Процедура: передвинуть агента на следующую вершину

Строка № 08. **Конец Цикла № 3**

Строка № 09. **Конец Цикла № 2**

Строка № 10. **Если** агент прошел не по всем вершинам, **то**

Строка № 11. Удалить агента

Строка № 12. **Конец Если**

Строка № 13. **Цикл № 4** по всем агентам

Строка № 14. Добавление феромона на дуги графа

Строка № 15. **Конец Цикла № 4**

Строка № 16. Испарение феромона со всех дуг графа

Строка № 17. **Конец Цикла № 1**

Строка № 18. Вывод оптимального пути

Строка № 19. **Конец**

Процедура перемещения агента на следующую вершину

Строка № 01. **Начало процедуры**

Строка № 02. Получение списка вершин, в которые агент может попасть из той, в которой он находится

Строка № 03. Удаление вершин, в которых уже побывал агент

Строка № 04. **Если** список вершин не пустой, **то**

Строка № 05. Вычисляем общее количество феромонов на дугах возможных переходов

Строка № 06. Распределяем вероятности по дугам

Строка № 07. Выбираем переход

Строка № 08. Перемещаем агента на данную вершину

Строка № 09. Увеличиваем общую длину пройденного агентом пути на величину перехода

Строка № 10. **Конец Если**

Строка № 11. **Конец процедуры**

4. Применение метода муравьиных колоний

Метод муравьиных колоний можно применять не только для решения задачи коммивояжера, но и для других прикладных задач маршрутизации авиационных средств [6]. Для того чтобы алгоритм муравьиных колоний решал другие задачи маршрутизации, необходимы изменения алгоритма. Разделим весь алгоритм муравьиных колоний на две части: в первой части определяется целевая функция – количество феромонов, а во второй определяются ограничения – набор вершин, куда агент может переходить из той, в которой находится. Изменение этих двух частей позволяет применять алгоритм муравьиных колоний к другим задачам.

Для демонстрации работы алгоритма возьмем полный граф из 30 вершин. Этот граф разработан для тестирования применения алгоритма при решении задач малой авиации. Вершинами графа могут служить близко расположенные населенные пункты, сельскохозяйственные поля, зоны мониторинга пожарной опасности местности и т.д. Граф генерировался автоматически, длины дуг определялись геометрическим расстоянием между вершинами в условных единицах. Для большой авиации граф имеет большую размерность, поэтому наглядное представление работы алгоритма затруднено. Жестко фиксированной начальной вершины нет, так как все вершины графа в примере идентичны. Все агенты выходят из одной вершины, она же начальная, так как в предложенных далее задачах идеологически невозможно посещение всех вершин. Условием остановки является близость множества найденных маршрутов к минимальному по длине за итерацию:

$$\frac{M - \mathrm{Min}}{\mathrm{Min}} < 0{,}1, \tag{4}$$

где M – среднее количество собранных ресурсов на итерации; Min – минимальное количество несобранных ресурсов на итерации.

5. Задача сбора ресурсов

Рассмотрим задачу следующего вида: Существует множество городов с аэропортами. Каждый город производит некоторое количество ресурсов. Авиационное средство перевозит ресурсы. При доставке ресурсов из определенного города авиакомпания получает прибыль. Для каждого города выгода от доставки ресурсов различна. У авиационного средства есть запас топлива, который ограничивает его перемещения между аэропортами. Будем считать, что единица топлива расходуется на единицу пройденного расстояния.

Целевая функция:

$$R = \max(R_i), \tag{5}$$

ограничения:

$$L_i \leqslant L, \tag{6}$$

где R – прибыль, которую получит компания; R_i – величина прибыли, которую собрал агент i; L_i – длина пути, который прошел агент i; L – ограничение на длину пути агента.

Значение целевой функции определит количество феромонов, которые каждый агент заносит на граф. Количество феромона, который агент заносит на пройденную дугу, вычисляется по формуле

$$\begin{gathered} \Delta\tau_{ij,k}(t) = \frac{Q}{R\max - R_k(t)}, \quad (i,j) \in T_k(t), \\ \Delta\tau_{ij,k}(t) = 0, \quad (i,j) \notin T_k(t), \end{gathered} \tag{7}$$

где Rmax – суммарное количество ресурсов на графе; $R_k(t)$ – количество ресурсов, собранное агентом k; Q – количество феромонов; $\Delta\tau_{ij,k}(t)$ – изменение феромона на дуге i–j; $T_k(t)$ – набор дуг, пройденных агентом k.

В результате больше феромонов будет на тех дугах, путь по которым обеспечивает наименьшее число несобранных ресурсов, т.е. наибольшее число собранных ресурсов. Для выполнения ограничений агент будет иметь возможность перемещаться только на дуги, которые будут удовлетворять ограничениям (6) по длине пути, пройденного агентом.

Изменения реализации стандартного алгоритма муравьиных колоний заключаются в следующем.

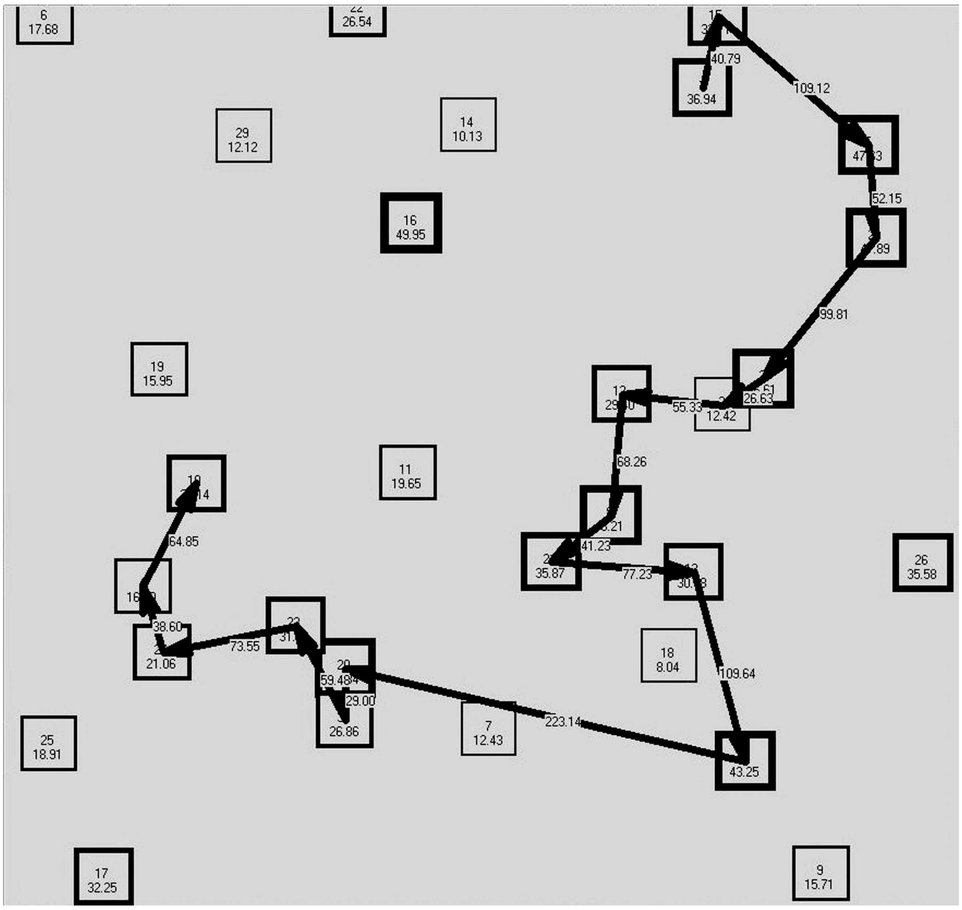

Рис. 1. Маршрут авиационного средства, в случае решения задачи сбора ресурса.

1. Перед началом поиска оптимального пути необходимо вычислить общее количество ресурсов на графе.

2. Перед добавлением феромонов на дуги графа (строка № 14) изменить формулу вычисления количества феромонов.

3. В процедуре перемещения агента на следующую вершину после удаления всех вершин, в которых уже побывал агент (строка № 03), необходимо удалить вершины, переход в которые не удовлетворяет ограничениям, а после увеличения длины пути, пройденного агентом (строка № 09), увеличить количество ресурсов, которые агент собрал.

Пусть авиационное средство находится в аэропорту города 1. Поставим ограничение по топливу, равное 1200 длины. Для нахождения решения будем использовать 18 муравьев. Количество феромонов (Q) примем равным 500, коэффициент испарения (p) — 0,5, $\alpha = 0{,}5$, $\beta = 1$. Начальной вершиной для всех агентов является вершина № 1. В результате оптимальный маршрут авиационного средства представлен на рис. 1. Для экономии места выведем

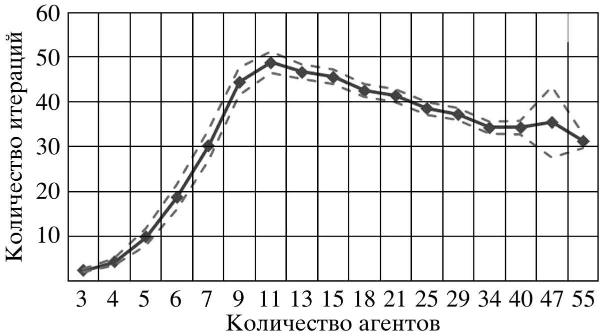

Рис. 2. Зависимость количества итераций от количества агентов в алгоритме сбора ресурсов.

дуги только найденного пути. Числа на дугах обозначают их длину. Толщина линий квадратов определяет количество ресурса в вершине.

В данном решении видно, что алгоритм муравьиных колоний прошел мимо вершин, в которых находится меньше ресурсов, для того чтобы успеть посетить вершины, где ресурса больше.

Исследуем скорость сходимости алгоритма (количество итераций) и рациональность найденного решения (количество собранного ресурса) в зависимости от различных параметров для выработки рекомендаций. На графике (рис. 2) приведено математическое ожидание оценок исследуемых характеристик по результатам 500 независимых реализаций. Так как по центральной предельной теореме ошибка наблюдений сходится к нормально распределенной величине, то построим доверительный интервал (на графиках пунктир) при доверительной вероятности, равной 0,999 [6]. На графиках черной линией представлено общее количество ресурсов на графе, максимально возможное количество собранных ресурсов.

Будем изменять количество агентов (N). Количество феромонов (Q) примем равным 500, коэффициент испарения (p) — 0,5. Ограничение на время сбора ресурса (T) установим равным 3500, $\alpha = 1$, $\beta = 0$. Коэффициент “зрения” был выбран равным 0 с целью исследовать поведение части алгоритма, основанной на распределении феромонов.

Из графика видно, что при увеличении количества агентов, участвующих в процессе сбора ресурсов, увеличивается количество итераций. При достижении количества агентов некоторой величины количество итераций начинает уменьшаться в силу того, что большое количество агентов находит более рациональные маршруты, чем малое, и в результате алгоритм сходится к этим рациональным маршрутам быстрее. Но время работы алгоритма не уменьшается из-за того, что уменьшение количества итераций не может компенсировать временные затраты, вызванные увеличением числа агентов.

При увеличении коэффициента испарения начинает расти количество найденных путей. В результате увеличивается количество собранных ресурсов и количество итераций.

Количество феромонов у агента слабо влияет на точность найденного решения и на количество итераций, так как является относительной величиной.

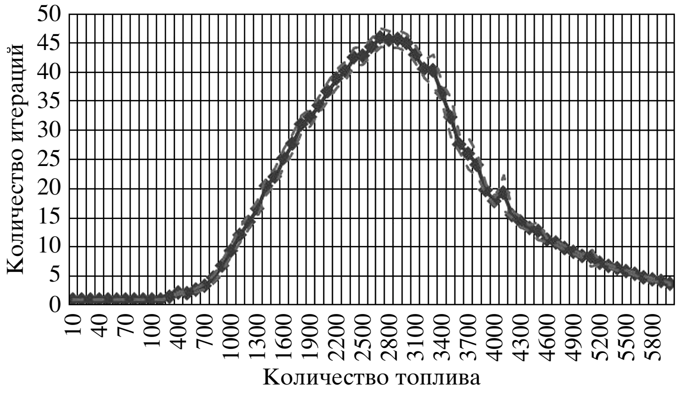

Рис. 3. Зависимость количества итераций от количества топлива в алгоритме сбора ресурсов.

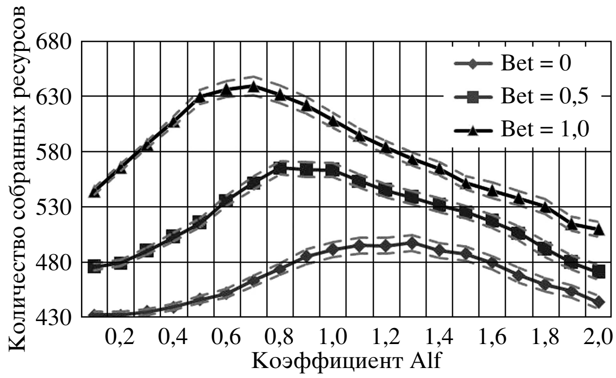

Рис. 4. Зависимость количества собранных ресурсов от коэффициентов α и β в алгоритме сбора ресурсов.

Будем изменять величину ограничения на время сбора ресурса (T). $N = 18$, $p = 0{,}5$, $Q = 500$, $\alpha = 1$, $\beta = 0$. Результаты приведены на рис. 3.

При величине ограничения (количестве топлива) на длину пути, меньшей, чем минимальная длина дуги, выходящей из начальной вершины, количество итераций равно 0. При увеличении значения ограничения количество итераций растет, так как растет число возможных путей. Количество итераций достигает экстремума в момент, когда ограничение на длину пути позволяет собрать все ресурсы, которые имеются на графе. Если дальше увеличивать время сбора ресурса, увеличится количество путей, которые обеспечивают сбор всех ресурсов на графе, и в результате количество итераций будет уменьшаться.

Исследуем влияние коэффициента α и β. $N = 8$, $p = 0{,}5$, $Q = 500$, $T = 3500$. Результаты приведены на рис. 4.

Можно сделать вывод, что максимальное количество собранных ресурсов достигается при коэффициенте α, близком к 0,8. Также прослеживается линейная зависимость увеличения количества собранных ресурсов с возрастанием числа итераций. При низком (меньше 0,8) коэффициенте α основную роль в поиске маршрута играет параметр зрения. Количество итераций резко увеличивается при увеличении коэффициента. При коэффициенте α больше 0,8 количество феромонов, оставляемых на дугах, начинает играть основную роль в выборе маршрута. Поэтому уменьшается количество собранных ресурсов и количество итераций. Чем выше коэффициент β, тем больше итераций требуется для нахождения маршрута. На представленном графике количество собранных ресурсов не достигает максимума в силу малочисленного ($N = 8$) количества агентов.

Анализируя результаты проведенных исследований, видим, что следует выбирать наибольший коэффициент β. Однако при больших коэффициентах β алгоритм "вырождается". Резкое уменьшение числа итераций при большом коэффициенте β связано с возможностью собрать все ресурсы, находящиеся на графе, что и приводит к "вырождению" алгоритма до A*. Коэффициент α рекомендуется выбирать равным 0,5.

Когда количество ресурсов в вершине зависит от времени перемещения агента, алгоритм посещает вершины с большим количеством ресурсов, не обращая внимания на вершины с небольшим количеством ресурсов. В случае наличия вершин "дозаправки", в которых уменьшается время перемещения агента, алгоритм старается посетить как можно больше таких вершин. Алгоритм также работает в условиях, когда "дозаправка" и количество собранных с вершины ресурсов зависят от уже имеющихся у агента ресурсов и от маршрута агента.

6. Задача сбора ресурсов несколькими авиационными средствами

Обычно в авиакомпании количество авиационных средств больше одного. Алгоритм муравьиных колоний легко модифицируется для решения данной задачи и не требует отдельной кластеризации вершин графа.

Пусть по графу перемещаются одновременно несколько агентов, составляющих группу. Все агенты из группы проходят по своему пути в графе. Эти пути не пересекаются, т.е. один агент из группы не может посетить вершину, в которой побывали другие агенты из той же группы. Количество ресурсов, которое собрала группа, вычислим как сумму ресурсов, всех ее агентов. Все агенты из группы преследуют общую цель по максимизации собранного группой ресурса. Количество агентов в группе будет равно числу авиационных средств. Для группы агентов используется один список пройденных вершин. При этом алгоритм поведения агента не изменится.

Постановка задачи в данном случае будет выглядеть так.

Целевая функция:

$$R = \max\left(\sum_{j=1}^{N} R_{i,j}\right), \tag{8}$$

ограничения:

$$L_{i,j} \leqslant L, \tag{9}$$

где R – прибыль, которую получит компания; $R_{i,j}$ – величина прибыли, которую собрал агент j из группы агентов i; N – количество агентов в группе; $L_{i,j}$ – длина пути, который прошел агент j из группы агентов i; L – ограничение на длину пути.

В зависимости от количества собранного группой ресурса будет вычисляться количество феромона, занесенного на каждую дугу, по которой прошли агенты из данной группы. Количество феромона, который каждый агент из группы заносит на пройденную им дугу, равно

$$\begin{gathered}\Delta\tau_{ij,k}(t) = \frac{Q}{R\max - \sum\limits_{s=1}^{N} R_{k,s}}, \qquad (i,j) \in T_k(t), \\ \Delta\tau_{ij,k}(t) = 0, \qquad (i,j) \notin T_k(t),\end{gathered} \tag{10}$$

где $R\mathrm{max}$ – суммарное количество ресурсов на графе; $R_{k,s}$ – количество ресурса, которое собрал агент s из группой агентов k; N – количество агентов в группе; Q – количество феромонов; $\Delta\tau_{ij,k}(t)$ – изменение феромона на дуге i–j группой агентов k; $T_k(t)$ – набор дуг, пройденных группой агентов k.

Так как у агентов есть общий список пройденных путей, то необходимо определить в какой последовательности агенты из группы заполняют этот список, т.е. какой агент будет перемещаться по графу в данный момент времени. Пусть агент, который прошел минимальный по длине путь, переместиться в следующую вершину. На первой итерации номер агента не имеет принципиального значения, так как все агенты в группе идентичные.

Для реализации решения данной задачи необходимо изменить алгоритм следующим образом.

1. Вместо муравьев будем работать с группой муравьев.

2. До добавления феромонов (строка № 14) добавится процедура вычисления количества ресурсов, собранных группой.

3. Перед процедурой Передвижения выбранного агента (строка № 7) необходимо выбрать агента из группы, который и будет перемещаться в соседнюю вершину.

Пусть у авиакомпании есть три летательных аппарата. Количество топлива у каждого авиационного средства равно 500. Проведем опыт с различными начальными вершинами у агентов из одной группы, см. рис. 5.

Из рисунка видно, что агенты стараются посетить группой все вершины с большим количеством ресурса, чтобы суммарная прибыль компании была максимальна. В случае разных аэродромов базирования авиационные средства наиболее эффективным образом используют свой запас топлива.

Исследуем влияние параметров α (параметр феромонов) и β (параметр длины) на скорость сходимости алгоритма и на величину найденного решения. Агентов в группе – 3, число групп $N = 10$, $p = 0{,}5$, $Q = 500$, $T = 500$, начальные вершины 1, 2, 7 (рис. 6).

Из графика можно сделать рекомендации по применению параметров $\alpha = 0{,}5$ и $\beta = 1$. Так же отчетливо видно, что характер кривых изменился. При увеличении коэффициента α количество итераций растет незначительно, а

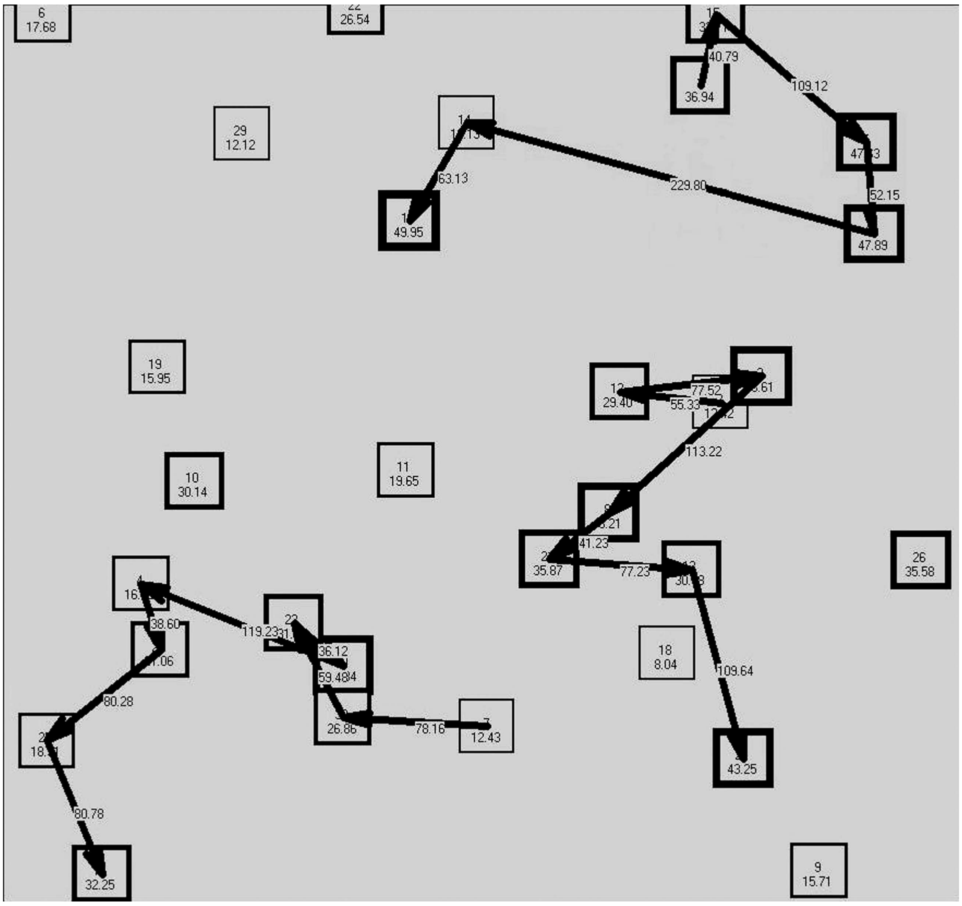

Рис. 5. Маршрут трех агентов на графе из начальной вершины №1, 2, 7.

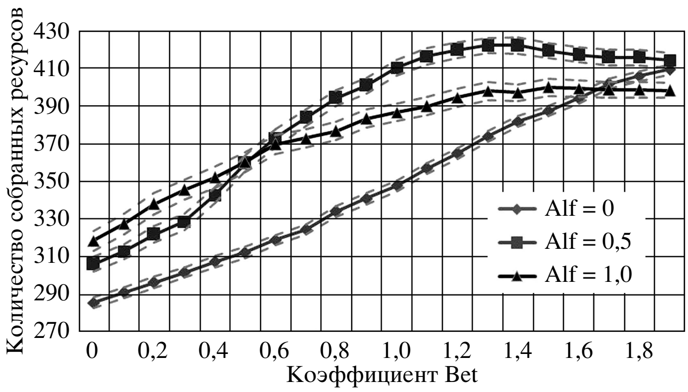

Рис. 6. Зависимость количества собранных ресурсов от коэффициентов α и β в алгоритме сбора ресурсов группой агентов.

экстремум количества собранных ресурсов сдвинулся влево к коэффициенту $\alpha = 0{,}5$. Данное изменение произошло благодаря тому, что одновременно три агента перемещались по графу и за одну итерацию заносили большее число феромонов. При изменении коэффициента β характер графиков остался прежним, но спада количества итераций при большом значении коэффициента β не наблюдается.

Следует отметить, что итерации для данного алгоритма выполняются дольше, чем итерации алгоритма сбора ресурса. Одна итерация данного алгоритма может быть оценена как три итерации алгоритма сбора ресурсов в условиях количества топлива, равного 500.

7. Задача сбора ресурсов несколькими различными авиационными средствами

Наиболее часто авиакомпании пользуются не однотипными транспортными средствами. Авиационные средства различаются по скорости и грузоподъемности.

В такой постановке задачи целевая функция останется прежней. Появится ограничение на количество ресурсов, которые агент может собрать в данной вершине, и изменится ограничение на длину пройденного пути. Ограничения примут вид

$$\begin{gathered} L_{i,j} \leqslant L_j, \\ R_j(t) \leqslant R\max_j, \end{gathered} \tag{11}$$

где $L_{i,j}$ – длина пути, который прошел агент j из группы i; L_j – ограничение на длину пути агента j; $R_j(t)$ – количество ресурсов, которые собрал агент j при посещении вершины в момент времени t; $R\max_j$ – максимальное количество ресурсов, которые может собрать агент j из одной вершины.

Для выполнения ограничения на длину пути агента введем коэффициент скорости, заданный в относительных единицах. Он задается пропорционально скоростям авиационных средств. Коэффициент скорости самолета (*Kspeed*) будет влиять на длину дуг для конкретного агента в соответствии с формулой

$$\mu_{ij,k} = Kspeed_k \cdot \mu_{ij}, \tag{12}$$

где $\mu_{ij,k}$ – длина дуги i–j для агента k; $Kspeed_k$ – коэффициент скорости агента k; μ_{ij} – длина дуги i–j.

Грузоподъемность задается в единицах прибыли таким образом, чтобы ограничивать максимальное количество ресурсов, собранных агентом из вершины. При различной грузоподъемности имеет смысл использовать для каждого агента из группы свой список пройденных вершин. Тогда авиационные средства смогут посещать одну и ту же вершину вместе и тем самым собрать весь ресурс из этой вершины. Но использование большого числа списков сильно замедляет работу алгоритма.

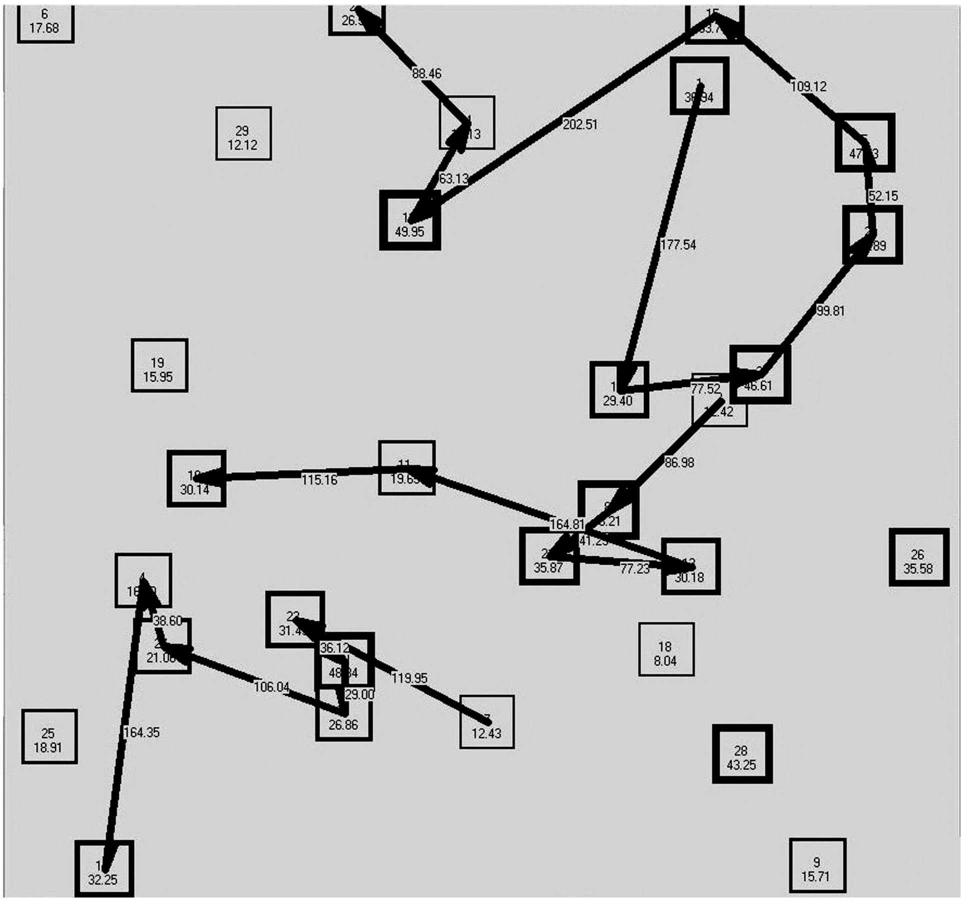

Рис. 7. Маршрут трех агентов из вершин 1,2 и 7. Один агент в два раза быстрее остальных.

Для реализации решения данной задачи потребуются небольшие изменения в алгоритме.

1. В процедуре перемещения агента на следующую вершину после удаления всех вершин, в которых уже побывал агент (строка № 03), необходимо удалить вершины, переход в которые не удовлетворяет ограничениям с учетом скорости данного агента.

2. В процедуре перемещения агента на следующую вершину после увеличения длины пути, пройденного агентом (строка № 09), увеличить количество ресурсов, которые агент собрал с учетом ограничения на максимальное количество собранных ресурсов. После этого уменьшить количество ресурсов в вершине на эту величину.

3. После нахождения решения агентами (строка № 08) нужно восстановить ресурсы в вершинах.

Для нахождения пути при условии различной скорости авиационных средств расположим трех агентов в вершинах 1, 2 и 7 (рис. 7). Пусть первый агент из группы будет передвигаться в 2 раза быстрее остальных.

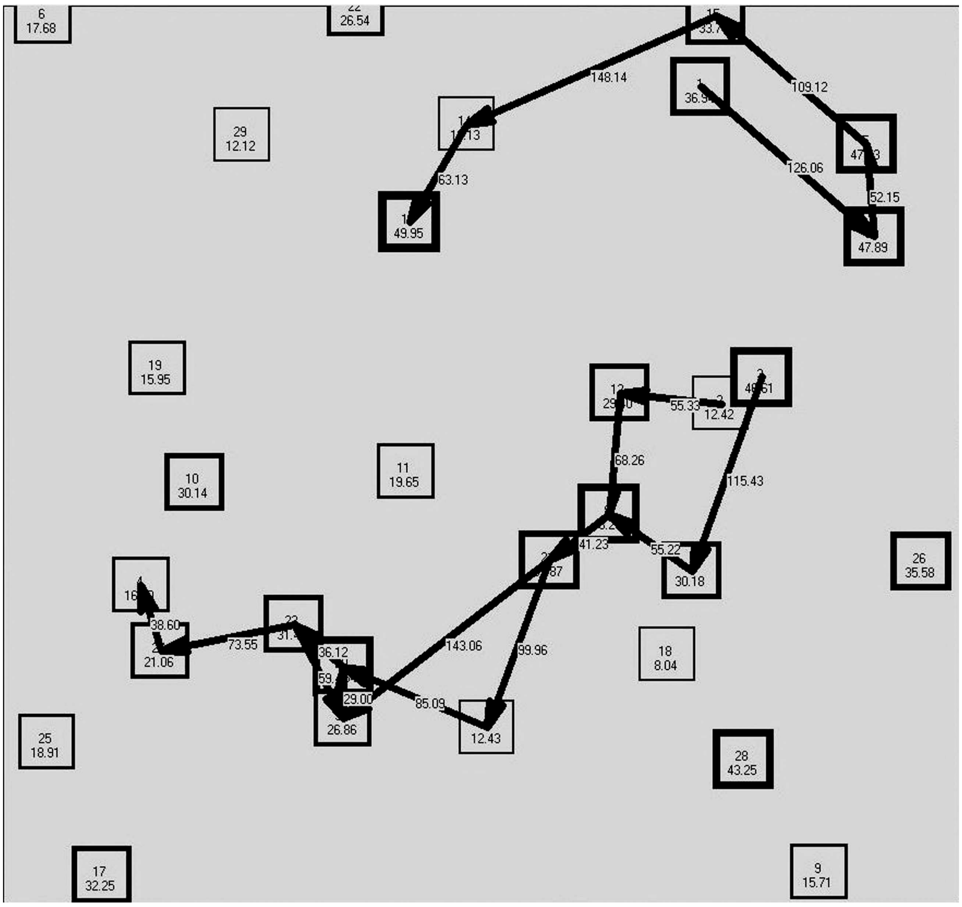

Рис. 8. Маршрут трех агентов из вершин 1,2 и 7 с ограничениями на грузоподъемность.

Для нахождения пути в условиях ограничений на грузоподъемность разместим транспортные средства в городах 1, 2 и 3. Установим ограничения на грузоподъемность у второго – 20, у третьего – 25, а первому ограничение не установим (рис. 8).

Увеличение скорости 1-го агента изменило маршрут 2-го. Из рис. 8 видно, как агенты заходят в одинаковые вершины, в которых количество ресурсов велико. Маршруты отличаются вершинами, где количество ресурса невелико и не выгодно заходить в эту вершину двумя агентами.

8. Задача сбора ресурсов несколькими авиационными средствами с автоматическим определением мест базирования

Как видно из рис. 7, 8 на величину приносимой прибыли влияет начальное расположение авиационных средств. Возникает задача оптимального размещения авиационных средств по аэродромам для получения наибольшей прибыли для авиакомпании.

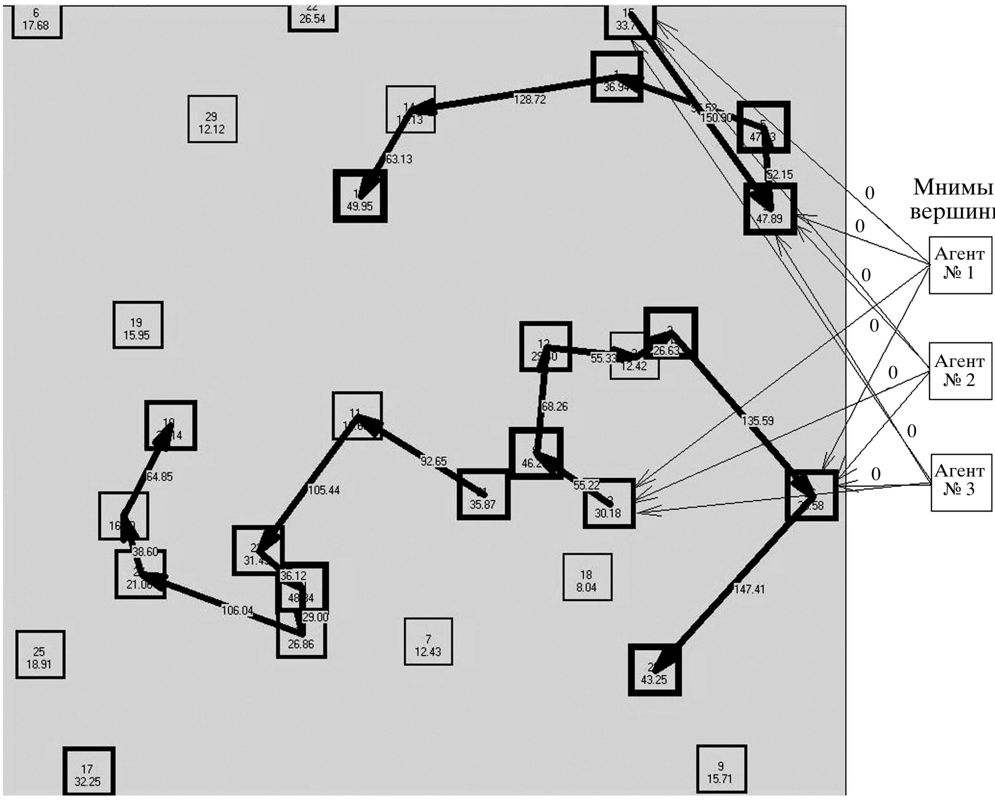

Рис. 9. Маршрут трех агентов при возможности выбора любой начальной вершины.

В данном случае поиск решения разбивается на два этапа: определение оптимального расположения авиационных средств и построение оптимальных маршрутов для получения наибольшей прибыли. Целевая функция и ограничения при этом останутся прежними.

Для решения данной задачи изменим структуру графа. В граф добавим "мнимые" вершины (аэропорты), с которых будут стартовать агенты (авиационные средства). Таких вершин будет столько, сколько агентов в одной группе (см. рис. 9). В каждой вершине будет располагаться только один конкретный агент из группы, каждая вершина будет начальной только для одного агента. Каждая новая ("мнимая") вершина соединяется дугой с каждой вершиной графа, рассматриваемой как возможный аэродром для авиационного средства. Длина таких дуг будет равна 0. После нахождения решения вершина, в которую агент переместиться из "мнимой", будет начальной вершиной для данного агента в найденном решении.

Так как алгоритм выбирает начальные вершины одновременно с поиском рационального решения, то полученные в результате начальные вершины будут удовлетворять целевой функции, а найденный путь – ограничениям.

Изменим алгоритм для решения подобных задач.

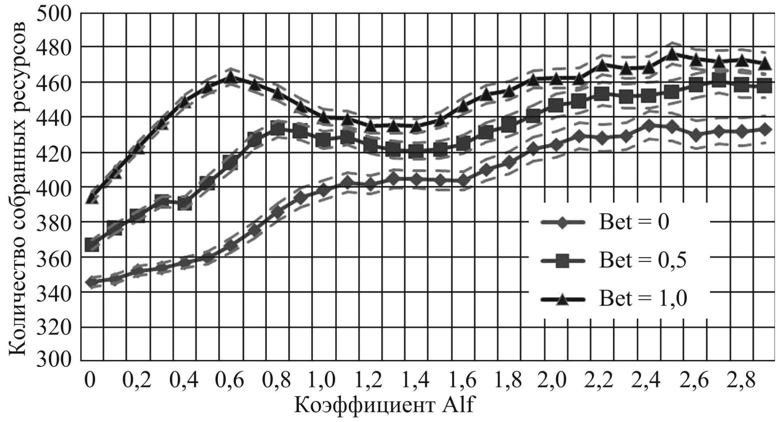

Рис. 10. Зависимость количества собранных ресурсов от коэффициентов α и β в алгоритме сбора ресурсов группой агентов при выборе начальных вершин.

1. Перед выполнением алгоритма и после выбора списка возможных начальных вершин необходимо создать дополнительные вершины (“мнимые”) и дуги на графе, а после завершения работы удалить их.

2. Закрепить за каждым агентом из группы соответствующую “мнимую” вершину.

Пусть группу агентов составляют три агента, которые выбирают любую начальную вершину. Тогда найденное решение будет выглядеть так, как показано на рис. 9 (показаны не все дуги из “мнимых” вершин).

Наибольшую прибыль принесет расположение авиационных средств в начальных вершинах 13, 15 и 21.

Исследуем скорость сходимости алгоритма и рациональность найденного решения в зависимости от различных параметров. При самостоятельном выборе агентами начальной вершины количество собранных ресурсов больше, чем при фиксированных вершинах, но количество требуемых итераций для нахождения рационального решения существенно выше за счет резкого увеличения количества вариантов путей. Время выполнения итерации немного увеличивается: когда агент выбирает начальную вершину, размерность графа увеличивается на число агентов в группе. В данном случае имеем полный граф с 33 вершинами.

Исследуем влияние коэффициентов α и β (рис. 10). $N = 10$, агентов в группе — 3, $p = 0{,}5$, $Q = 500$, $T = 500$.

Графики влияния коэффициента α и β схожи с аналогичными графиками для задачи без выбора мест базирования. Увеличение количества итераций обусловлено увеличением размерности графа. Благодаря возможности выбора начальной вершины, увеличивается количества собранных ресурсов при увеличении коэффициента α (агенты имеют возможность улучшать решение за счет выбора начальных вершин).

9. Выводы и рекомендации

В данной статье рассматривались модификации алгоритма муравьиных колоний для решения различных задач на графах. Представление алгоритма муравьиных колоний как многоагентного алгоритма позволило легко модифицировать его. В данном случае необходимо изменить поведение агента (или муравья с точки зрения метаэвристических алгоритмов), а не состояние системы в целом. Для решения различных задач маршрутизации предложено разбиение поведения агента на две части. В первой части определялась целевая функция, а во второй ограничения поведения агента. В такой постановке описание алгоритма сводится к стандартной формализации задач маршрутизации. В ходе разработки алгоритмов был написан программный продукт на языке программирования Borland Delphi 7.0, на котором проведены стендовые испытания по применению данных алгоритма к различным задачам на реальных графах различной размерности. В частности, рассматривались реальные задачи маршрутизации группы инкассаторов по Москве и решение задачи нахождения рациональных маршрутов интермодальных перевозок по миру [7], т.е. перевозок в которых используются два и более вида транспорта.

СПИСОК ЛИТЕРАТУРЫ

1. *Colorni A., Dorigo M., Maniezzo V.* Distributed Optimization by Ant Colonies // Proc. First Eur. Conf. Artific. Life, Paris, France, F. Varela and P. Bourgine (Eds.), Elsevier Publishing. 1992. P. 134–142.
2. *Colorni A., Dorigo M., Maniezzo V.* An Investigation of some Properties of an Ant Algorithm // Proc. Parallel Probl. Solving from Nature Conf. (PPSN 92), Brussels, Belgium, R. Männer and B. Manderick (Eds.), Elsevier Publishing. 1992. P. 509–520.
3. *Штовба С.Д.* Муравьиные алгоритмы // Exponenta Pro. Математика в приложениях. 2003. № 4 (4). С. 70–75.
4. *Курейчик В.М., Кажаров А.А.* О некоторых модификациях муравьиного алгоритма // Изв. ЮФУ. Технические науки. 2008. № 4 (81).
5. *Павленко А.И., Титов Ю.П.* Сравнительный анализ модифицированных методов муравьиных колоний // Научно-практ. журн. “Прикладная информатика” 2012. № 4 (40). С. 100–112.
6. *Венцель Е.С.* Теория вероятностей. 11-е изд. М.: КРОНУС, 2010.
7. *Милославская С.В., Плужников К.И.* Мультимодальные и интермодальные перевозки. Уч. пособие. М.: РосКонсульт, 2001.

Статья представлена к публикации членом редколлегии П.Ю. Чеботаревым.

Поступила в редакцию 09.07.2012