

УДК 681.3.06

МУРАВЬИНЫЕ АЛГОРИТМЫ: ТЕОРИЯ И ПРИМЕНЕНИЕ

© 2005 г. С. Д. Штовба

Винницкий национальный технический университет

21021 Украина, Винница, Хмельницкое шоссе, 95

E-mail: shtovba@ksu.vstu.vinnica.ua, shtovba@svitonline.com

Поступила в редакцию

В статье дается обзор теории и применения муравьиных алгоритмов – нового метода дискретной оптимизации, основанного на имитации самоорганизации колонии биологических муравьев. Колония муравьев может рассматриваться как многоагентная система, в которой каждый агент – муравей функционирует автономно по очень простым правилам. В противовес почти примитивному поведению агентов, поведение всей системы получается на удивление разумным. Муравьиные алгоритмы серьезно исследуются европейскими учеными с середины 90-х годов. Муравьиные алгоритмы с успехом применяются для решения таких сложных комбинаторных задач оптимизации, как: задача коммивояжера, задача маршрутизации транспорта, задача раскраски графа, квадратичная задача о назначениях, задача оптимизации сетевых графиков, задача календарного планирования, и других. Особенно эффективны муравьиные алгоритмы для он-лайн оптимизации процессов в распределенных нестационарных системах, например, трафиков в телекоммуникационных сетях.

ВВЕДЕНИЕ

Цель настоящей статьи состоит в изложении теоретических основ и примеров практического применения муравьиных алгоритмов – нового перспективного подхода к оптимизации, базирующегося на имитации поведения колонии муравьев. Колония муравьев может рассматриваться как многоагентная система, в которой каждый агент – муравей функционирует автономно по очень простым правилам. В противовес почти примитивному поведению агентов, функционирование системы получается на удивление разумным: “...гнезда многих видов [муравьев] поражают своими размерами, сложной и рациональной архитектурой. Дороги, тоннели, разбросанные по территории убежища для тлей и червецов, грибные сады... Разнообразные способы запасаения и хранения пищи, фактическое приручение ряда видов насекомых...” [1].

Интересным результатом кооперативного поведения биологических муравьев является нахождение кратчайшего маршрута от источника пищи к гнезду. Алгоритмы оптимизации, имитирующие такое поведение муравьев, предложены в начале девяностых в Италии [2]. В международном журнале первая статья по муравьиным алгоритмам [3] была опубликована в 1996г., и уже через несколько лет сформировалось отдельное научное направление “Роевой интеллект и муравьиные алгоритмы”, в котором интенсивно работают европейские ученые. Раз в два года в Бельгии проводится международный научный семинар по муравьиным алгоритмам и роевому интеллекту – International Workshop on Ant Colony Optimization and Swarm Intelligence. Организовываются специальные “муравьиные” секции и семинары на конгрессах и крупных конференциях; издаются спецвыпуски международных научных журналов.

Хорошие результаты муравьиной оптимизации получены для таких сложных комбинаторных задач, как: задача коммивояжера, задача маршрутизации транспорта, задача раскраски графа, квадратичная задача о назначениях, оптимизация сетевых графиков, задача календарного планирования, и других. Знаковым со-

бытием в признании перспективности исследований по муравьиной оптимизации можно считать решение Европейской комиссии о присуждении в 2003 г. Marie Curie Excellence Award (Премии Фонда Марии Кюри за выдающиеся научные достижения) в размере 50000 евро доктору Марко Дориго, изобретателю муравьиных алгоритмов. Несмотря на стремительные успехи муравьиных алгоритмов, большинство русскоязычных специалистов по математическому программированию не знакомо с этим научным направлением. Первая статья по муравьиной оптимизации в международном журнале опубликована российскими учеными только в 2004 г. [4].

Предлагаемая вниманию читателя статья состоит из четырех разделов: в первом разделе излагаются принципы самоорганизации социальных насекомых и объясняется, как муравьи находят кратчайший маршрут, во втором – на примере задачи коммивояжера показывается, как внедрить кооперативное поведение муравьев в алгоритмы комбинаторной оптимизации, в третьем – анализируются методы улучшения муравьиных алгоритмов, в четвертом – дается обзор применения муравьиных алгоритмов. При написании первого и второго разделов использовались источники [5–7].

1. ПРИНЦИПЫ ПОВЕДЕНИЯ МУРАВЬЕВ

Муравьи относятся к *социальным* насекомым, живущим внутри некоторого коллектива – семьи или колонии. Около двух процентов насекомых являются социальными, половину из них составляют муравьи. Число муравьев в одном социуме колеблется от 30 штук до десятков миллионов. Муравьи занимают доминирующее положение в бассейне Амазонки, в лесах которого на них приходится более 30% биомассы фауны. Поведение муравьев при транспортировании пищи, преодолении препятствий, строительстве муравейника и других действиях зачастую приближается к теоретически оптимальному. Принципы поведения муравьев выдержали испытания на протяжении 100 миллионов лет – именно столько времени назад муравьи “колонизировали” Землю. Живучесть му-

равьиных колоний удивительная – удаление до 40% насекомых практически не сказывается на функционировании всего социума [8]. При массовой гибели муравьев, например, после химической обработки территории, насекомые из соседних муравейников объединяются в одну семью для сохранения социума [1].

Основу социального поведения муравьев составляет *самоорганизация* – множество динамических механизмов, обеспечивающее достижение системой глобальной цели через низкоуровневое взаимодействие ее элементов. Принципиальной особенностью такого взаимодействия является использование элементами системы *только локальной информации*. При этом исключается любое централизованное управление и обращение к глобальному образу, репрезентирующему систему во внешнем мире. Самоорганизация является результатом взаимодействия следующих четырех компонентов:

- многократности повторения;
- случайности;
- положительной обратной связи;
- отрицательной обратной связи.

Муравьи используют два способа передачи информации: прямой – обмен пищей, мандибулярный, визуальный и химический контакты, и непрямой – стигмержи (*stigmergy*). *Стигмержи* – это разнесенный во времени тип взаимодействия, когда один субъект взаимодействия изменяет некоторую часть окружающей среды, а остальные используют эту информацию позже, когда находятся в ее окрестности. Биологически, стигмержи осуществляется через *феромоны* (*pheromone*) – специальный секрет, откладываемый как след при перемещении муравья. Чем выше концентрация феромонов на тропе, тем больше муравьев будет по ней двигаться. Со временем феромоны испаряются, что позволяет муравьям адаптировать поведение под изменения внешней среды. Распределение феромонов является своего рода динамически изменяемой глобальной памятью муравейника. Любой муравей в фиксированный момент времени может воспринять и изменить лишь одну локальную ячейку этой глобальной памяти.

На примере экспериментов с асимметричным мостом разберем, как кооперативное поведение муравьев позволяет найти кратчайший маршрут к источнику пищи. Асимметричный мост (рис. 1) соединяет гнездо муравьев с источником пищи двумя ветками разной длины. Эксперименты [9] проводились с лабораторной колонией аргентинских муравьев (*Iridomyrmex humilis*), которые откладывают феромоны по дороге, как от гнезда, так и к гнезду. Эксперименты проводились по следующей схеме:

- 1) строился мост А-В-С-Д;
- 2) открывались ворота в точке А;
- 3) фиксировалось количество муравьев, выбравших длинную (А-С-Д) и короткую (А-В-Д) ветки моста.

В начале экспериментов муравьи выбирали обе ветки одинаково часто. Через некоторое время почти все муравьи двигались по кратчайшему маршруту А-В-Д, что объясняется следующим образом. В начале экспериментов на ветках моста феромоны отсутствовали, поэтому муравьи одинаково часто выбирали маршруты А-С-Д и А-В-Д. Муравьи, выбравшие короткий маршрут А-В-Д-В-А, быстрее возвращались с пищей в гнездо, оставляя феромонные следы на короткой ветке моста. При следующем выборе маршрута, муравьи отдавали предпочтение короткой ветке моста, так как концентрация феромонов на ней выше. Поэтому феромоны начинают быстрее накапливаться на ветке А-В-Д, привлекая муравьев к выбору кратчайшего маршрута.

2. МУРАВЬИНЫЙ ПОДХОД К ЗАДАЧЕ КОММИВОЯЖЕРА

Задача коммивояжера состоит в поиске кратчайшего замкнутого маршрута, проходящего через все города ровно один раз. Выбор задачи коммивояжера для иллюстрации идей муравьиных алгоритмов обусловлен следующим:

- 1) задача наглядно интерпретируется в терминах поведения муравьев – перемещения коммивояжера и муравьев интуитивно сопоставимы;
- 2) это NP-сложная задача;

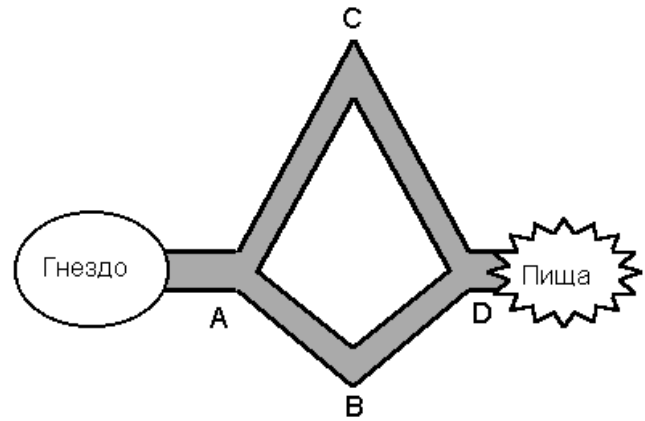


Рис. 1. Асимметричный мост (по материалам [9]).

- 3) это традиционный тестовый полигон (benchmark problem) для методов комбинаторной оптимизации. Существует обширная библиотека тестовых задач коммивояжера и методов их решения, что позволяет сравнить эффективность муравьиных алгоритмов оптимизации с другими подходами;
- 4) это дидактическая задача, для которой можно без злоупотребления техническими деталями алгоритма объяснить процесс поиска оптимума;
- 5) это первая комбинаторная задача, решенная муравьиными алгоритмами.

Рассмотрим, как реализовать четыре составляющие самоорганизации муравьев при оптимизации маршрута коммивояжера.

Множественность взаимодействия реализуется итерационным поиском маршрута коммивояжера несколькими муравьями одновременно. Каждый муравей рассматривается как отдельный, независимый коммивояжер, решающий свою задачу. За одну итерацию алгоритма муравей совершает полный маршрут коммивояжера.

Положительная обратная связь реализуется имитацией поведения муравьев типа “оставление следов – перемещение по следам”. Чем больше следов оставлено на тропе (ребре графа в задаче коммивояжера), тем больше муравьев выберут ее. При этом на тропе появляются новые следы, которые на последующих итерациях алгоритма привлекают допол-

нительных муравьев. Для задачи коммивояжера положительная обратная связь реализуется следующим стохастическим правилом: вероятность включения ребра графа в маршрут муравья пропорциональна количеству феромонов на нем. Такое вероятностное правило реализует и другую составляющую самоорганизации – *случайность*. Чем короче путь “источник пищи – муравейник”, тем чаще биологический муравей может пройти его за фиксированный промежуток времени, откладывая при каждом проходе некоторое количество феромонов. Для имитации такого поведения муравьев объем виртуальных феромонов, откладываемых на ребре графа, принимают обратно пропорциональным длине маршрута. Чем короче маршрут, тем больше феромонов будет отложено на соответствующих ребрах графа, и тем чаще муравьи будут использовать их при синтезе новых маршрутов.

Использование только положительной обратной связи приводит к стагнации – к случаю, когда все муравьи двигаются одним и тем же субоптимальным маршрутом. Для избежания этого вводится *отрицательная обратная связь* через испарение феромонов. Интенсивность испарения не должна быть слишком высокой, чтобы не заузить область поиска. С другой стороны, испарение не должно быть и слишком быстрым, чтобы не привести к преждевременному “забыванию” – потере памяти колонии, и, следовательно, к некооперативному поведению муравьев.

Для каждого муравья переход из города i в город j зависит от трех составляющих: табу-списка (tabu list), видимости и следа виртуальных феромонов.

Табу-список (память муравья) – это список посещенных муравьем городов, заходить в которые еще раз нельзя. Он возрастает при совершении маршрута и обнуляется в начале каждой итерации алгоритма. Обозначим через J_{ik} – список городов, которые еще необходимо посетить муравью k , находящемуся в городе i . Ясно, что J_{ik} является дополнением к табу-списку.

Видимость – это величина, обратная расстоянию: $\eta_{ij} = 1/D_{ij}$, где D_{ij} – расстояние между городами i и j . Видимость – это локальная ста-

тическая информация, выражающая эвристическое желание попасть в город j из города i : чем ближе город, тем сильнее желание посетить его.

След виртуальных феромонов на ребре $(i - j)$ представляет подтвержденное опытом колонии желание посетить город j из города i . В отличие от видимости, распределение феромонов меняется после каждой итерации алгоритма, отражая приобретенный муравьями опыт. Количество виртуальных феромонов на ребре $(i - j)$ на итерации t обозначим через $\tau_{ij}(t)$.

Вероятность перехода k -го муравья из города i в город j на t -ой итерации рассчитывается следующим вероятностно-пропорциональным правилом:

$$P_{ij,k}(t) = \begin{cases} \frac{(\tau_{ij}(t))^\alpha \cdot (\eta_{ij})^\beta}{\sum_{l \in J_{ik}} (\tau_{il}(t))^\alpha \cdot (\eta_{il})^\beta}, & \text{если } j \in J_{ik}, \\ 0, & \text{если } j \notin J_{ik}, \end{cases} \quad (1)$$

где $\alpha \geq 0$ и $\beta \geq 0$ – два регулируемых параметра, задающие важность следа феромонов и видимости при выборе маршрута. При $\alpha = 0$ выбирается ближайший город, что соответствует жадному алгоритму в классической теории оптимизации. Если $\beta = 0$, то работает лишь феромонное усиление, что приводит к быстрому вырождению маршрутов к одному субоптимальному решению. Для обеспечения хорошей динамики оптимизации в [3] рекомендуется установить $\beta \geq \alpha$.

Обратим внимание, что правило (1) определяет вероятности выбора того или иного города. Собственно выбор города осуществляется по принципу “колеса рулетки” – каждый город на ней имеет свой сектор с площадью, пропорциональной вероятности (1). Для выбора города нужно бросить шарик на рулетку – сгенерировать случайное число и определить сектор, на котором он остановится.

После завершения маршрута k -ый муравей откладывает на ребре (i, j) такое количество феромонов:

$$\Delta\tau_{ij,k}(t) = \begin{cases} \frac{Q}{L_k(t)}, & \text{если } (i, j) \in T_k(t), \\ 0, & \text{если } (i, j) \notin T_k(t), \end{cases}$$

```

%Базовый муравьиный алгоритм решения задачи коммивояжера:
<Ввод матрицы расстояний D>
<Инициализация параметров алгоритма  $\alpha, \beta, Q, p, \tau_0$ >
m=n %Количество муравьев равно числу городов
For i=1:n %Для каждого ребра
  For j=1:n
    If i~=j
       $\eta(i,j)=1/D(i,j)$  %Видимость
       $\tau(i,j)=\tau_0$  %Феромон
    Else  $\tau(i,j)=0$ 
    End
  End
End
End
For k=1:m
  <Разместить муравья k в случайно выбранный город>
End
<Выбрать условно-кратчайший маршрут  $T^+$  и рассчитать его длину  $L^+$ >
%Основной цикл
For t=1:tmax %tmax - количество итераций алгоритма
  For k=1:m %Для каждого муравья:
    <Построить маршрут  $T_k(t)$  по правилу (1)>
    <Рассчитать  $L_k(t)$  - длину маршрута  $T_k(t)$ >
  End
  If "Лучшее решение найдено?" <Обновить  $T^+$  и  $L^+$ > End
  For i=1:n
    For j=1:n %Для каждого ребра
      <Обновить следы феромона по правилу (2)>
    End
  End
End
End
<Вывести кратчайший маршрут  $T^+$  и его длину  $L^+$ >

```

Рис. 2. Базовый муравьиный алгоритм оптимизации маршрута коммивояжера.

где

$T_k(t)$ – маршрут муравья k на итерации t ;

$L_k(t)$ – длина маршрута $T_k(t)$;

$Q > 0$ – регулируемый параметр.

Для исследования всего пространства решений необходимо обеспечить испарение феромонов. Обозначим коэффициент испарения через $p \in [0, 1]$, тогда правило обновления феромонов примет вид:

$$\tau_{ij}(t+1) = (1-p) \cdot \tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij,k}(t), \quad (2)$$

где m – количество муравьев в колонии. В начале оптимизации количество феромонов на ребрах принимается равным небольшому положительному числу τ_0 .

Общее количество муравьев в колонии остается постоянным. Очень многочисленная колония приводит к быстрому усилению субоптимальных маршрутов. Когда же муравьев мало, возникает опасность потери кооперативности через ограниченное взаимодействие и быстрое испарения феромонов. Обычно число муравьев назначают равным количеству городов. При этом каждый муравей начинает маршрут со своего города.

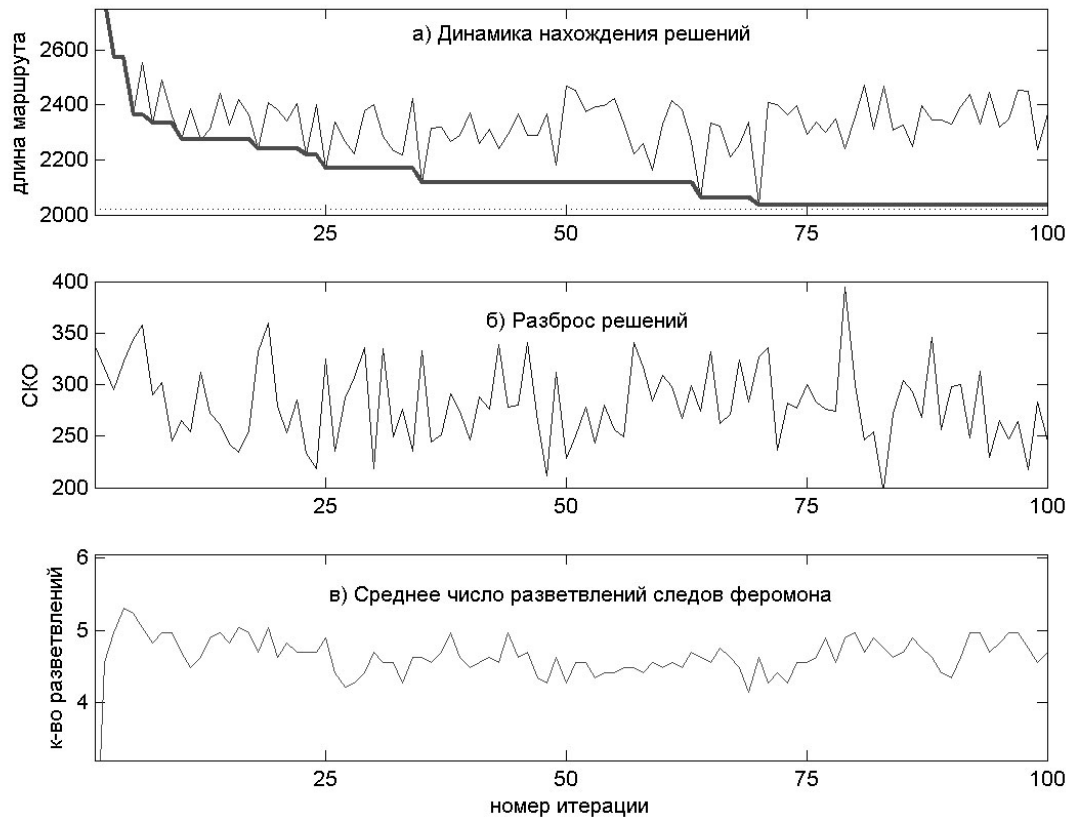


Рис. 3. Исследование процесса решения задачи коммивояжера Bays29 базовым муравьиным алгоритмом (при $\alpha = 0.5$, $\beta = 0.5$, $Q = 2000$, $p = 0.5$).

В отличие от биологических муравьев виртуальные агенты помнят список посещенных городов и живут в пространстве с дискретным временем. Кроме того, они не полностью “слепы”, так как выбирают маршрут не только по концентрации феромонов, но и используют эвристики [3]. Эти отличия обусловлены тем, что виртуальные муравьи используются для решения задач оптимизации, а не для моделирования колонии насекомых. На рис. 2 приводится базовый муравьиный алгоритм оптимизации маршрута коммивояжера, реализующий рассмотренные принципы самоорганизации.

В наших экспериментах на тестовой задаче с 29 населенными пунктами в Баварии Bays29 [10] базовый муравьиный алгоритм после 100 итераций нашел оптимальный маршрут длиной 2020 в двух экспериментах из 10. Для гарантированного нахождения оптимума число итераций алгоритма необходимо увеличить до 1–2 тысяч. При выполнении алгоритма популяция решений

не вырождается к одному, общему для всех муравьев маршруту. На рис. 3 а тонкой линией показаны лучшие решения, найденные на каждой итерации муравьиного алгоритма. Жирной линией показаны наилучшие решения, найденные с начала работы алгоритма. Линии не совпадают, следовательно, муравьиный алгоритм генерирует новые решения на каждой итерации. Об этом свидетельствует и рис. 3 б, на котором показано среднеквадратическое отклонение длин маршрутов, найденных муравьями на текущей итерации алгоритма. На рис. 3 в показано среднее по городам число разветвлений следов феромонов. Оно получено подсчетом ребер, инцидентных вершине графа, феромоны на которых превышают некоторый порог. На протяжении всей работы алгоритма в любом городе существует около пяти перспективных альтернатив продолжения маршрута.

По сравнению с точными методами, например, динамическим программированием или методом ветвей и границ, муравьиный алгоритм

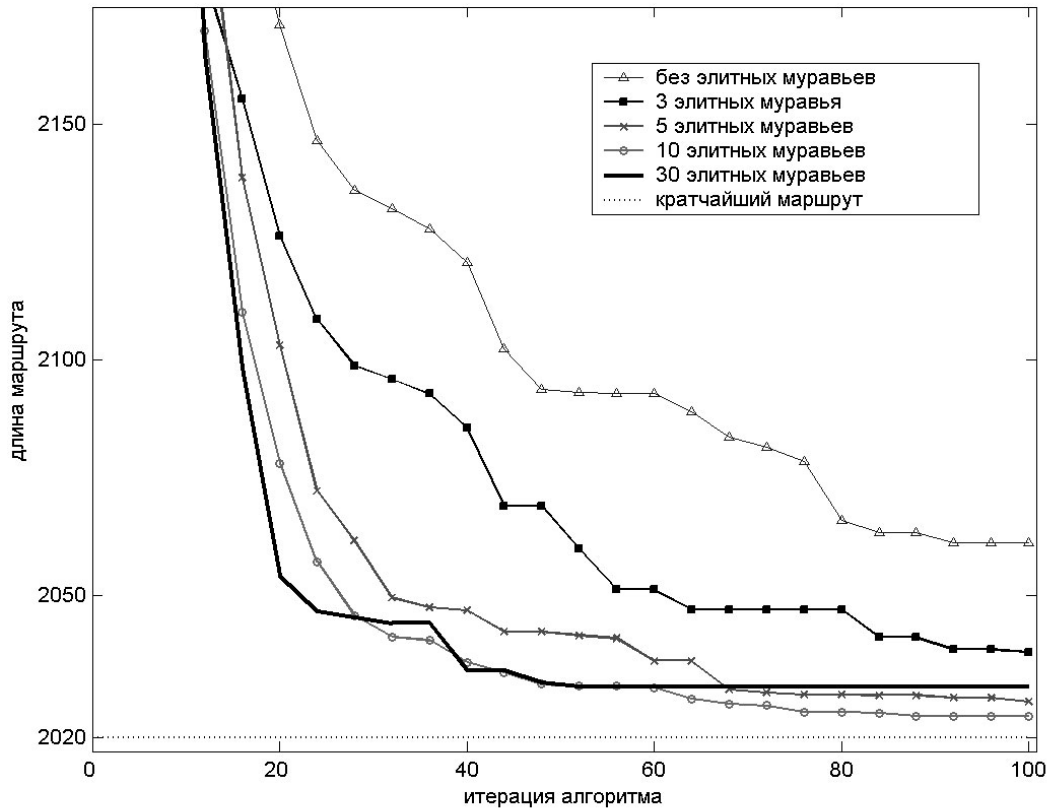


Рис. 4. Сравнение динамик алгоритмов с разным числом элитных муравьев для задачи Vays29.

значительно быстрее находит близкие к оптимальному решению даже для задач небольшой размерности. Время оптимизации муравьиным алгоритмом является полиномиальной функцией от размерности — $O(t_{max}, n^2, m)$, тогда как для точных методов зависимость экспоненциальная.

3. МЕТОДЫ УЛУЧШЕНИЯ БАЗОВОГО МУРАВЬИНОГО АЛГОРИТМА

Базовый муравьиный алгоритм (Ant System) решает задачи коммивояжера небольшой размерности (с числом городов до 75) с точностью на уровне других неспециализированных эвристических методов, таких, как генетические алгоритмы (Genetic Algorithms) и имитированный отжиг (Simulated Annealing) [3]. Для задач большой размерности простейший муравьиный алгоритм — не конкурент современным специализированным методам оптимизации маршрута коммивояжера. Недостаточная эффективность базового муравьиного алгоритма связана с:

- 1) возможностью потери наилучшего найденного решения через вероятностное правило выбора маршрута;
- 2) низкой сходимостью вблизи оптимума через приблизительно равный вклад как лучших, так и худших решений в обновление феромонов;
- 3) хранением в памяти колонии заведомо неперспективных вариантов, что для задач большой размерности сильно расширяет область поиска.

Аналогичные проблемы возникают и в генетических алгоритмах с селекцией по принципу колеса рулетки, когда площади секторов рулетки для хороших и плохих хромосом практически равны [11]. Чтобы дистанцировать хорошие решения от плохих, в генетических алгоритмах применяются адаптивная фитнес-функция, ранжирование хромосом, усеченная и турнирная селекций. Для сохранения наилучшего решения используется селекция с элитиз-

мом, когда в новую популяцию вначале включается наилучшая хромосома, а затем выбираются все остальные. Для ускорения сходимости в окрестности оптимума применяются методы локального поиска. Для сокращения пространства поиска используются “строительные блоки”. Ниже рассматриваются схожие способы улучшения муравьиных алгоритмов.

3.1. Элитные муравьи

В окрестности оптимума длины маршрутов муравьев отличаются незначительно, поэтому, в соответствии с (2), вклад как лучших, так и худших решений в обновление феромонов будет почти одинаковым. Это обуславливает медленное приближение к оптимуму в его окрестности. Первое улучшение муравьиных алгоритмов состояло в усилении наилучшего решения через элитных муравьев [3]. Элитные муравьи откладывают феромоны только на ребрах наилучшего маршрута T^+ , найденного с начала работы алгоритма.

Для задачи коммивояжера количество феромонов, откладываемое элитным муравьем на каждом ребре маршрута T^+ , принимается равным Q/L^+ , где L^+ – длина маршрута T^+ . Идея элитизма состоит в том, что увеличенное количество феромонов будет притягивать больше муравьев, подталкивая их к исследованию решений, содержащих несколько ребер наилучшего на данный момент маршрута T^+ . Если в муравейнике есть e элитных муравьев, то ребра маршрута T^+ получают дополнительное усиление:

$$\Delta\tau_{ij,e} = e \cdot Q/L^+, \quad \forall(i, j) \in T^+. \quad (3)$$

На рис. 4 приведены усредненные за 10 прогонов динамики решения задачи Bays29 алгоритмами с различным числом элитных муравьев. При большом числе элитных муравьев алгоритмы очень быстро, за 30–40 итераций, находят субоптимальные маршруты длиной 2033, 2028, 2026. Однако после этого надолго застревают в локальных оптимумах, так как элитные муравьи сильно их усиливают. В проведенных нами 10 экспериментах за 100 итераций алгоритмы с тремя и с пятью элитными муравьями нашли оптимальный маршрут по 3 раза, с десятью элитными муравьями – 6 раз, а с тридцатью – только дважды.

Идеи элитизма развиваются в ранговых муравьиных алгоритмах (Rank-Based Ant Systems) [12], алгоритмах муравьиной колонии (Ant Colony Systems) [13], макс-минных муравьиных алгоритмах (MAX-MIN Ant Systems) [14] и лучших-худших муравьиных алгоритмах (Best-Worst Ant Systems) [15]. Эти алгоритмы оптимизируют быстрее за счет увеличения вероятностей выбора хороших фрагментов маршрутов.

3.2. Ранговый муравьиный алгоритм

В ранговых муравьиных алгоритмах найденные на каждой итерации решения ранжируют, и только $(w-1)$ лучших муравьев и один элитный муравей откладывают феромоны. Таким образом, плохие маршруты не запоминаются. Количество феромонов зависит от ранга муравья.

Для задачи коммивояжера правило обновления феромонов (2) принимает вид:

$$\begin{aligned} \tau_{ij}(t+1) = & (1-p) \cdot \tau_{ij}(t) + \\ & + \sum_{r=1, \dots, w-1} (w-r) \cdot \Delta\tau_{ij,r}(t) + \\ & + w \cdot \Delta\tau_{ij,e}(t), \end{aligned} \quad (4)$$

где

$$\Delta\tau_{ij,e} = \begin{cases} Q/L^+, & \text{если } (i, j) \in T^+, \\ 0, & \text{если } (i, j) \notin T^+ \end{cases}$$

– феромоны элитного муравья;

$$\Delta\tau_{ij,r} = \begin{cases} Q/L^r(t), & \text{если } (i, j) \in T^r(t), \\ 0, & \text{если } (i, j) \notin T^r(t) \end{cases}$$

– феромоны муравья с рангом r ;

$T^r(t)$ – маршрут муравья с рангом r на итерации t ;

$L^r(t)$ – длина маршрута $T^r(t)$.

Правило (4) в терминах базового муравьиного алгоритма интерпретируется так: w муравьев двигаются наилучшим маршрутом T^+ , $(w-1)$ муравьев двигаются лучшим текущим маршрутом $T^1(t)$, $(w-2)$ муравьев двигаются вторым по рангу маршрутом $T^2(t)$ и т.д. Количество феромонов, откладываемых на ребрах двух маршрутов с почти одинаковой длиной, отличается существенно, как минимум на $\frac{100}{w-1}\%$. Поэтому, в

Таблица 1. Решения задач коммивояжера различными муравьиными алгоритмами [14]

Задача	Eil51	Kroa100	D198	Ry48p	Ft70	Kro124p	Ftv170
Оптимум	426	21282	15780	14422	38673	36230	2755
MMAS+pts	427.1	21291.6	15956.8	14523.4	38922.7	36573.6	2817.7
MMAS	427.6	21320.3	15972.5	14553.2	39040.2	36773.5	2828.8
ACS	428.1	21420	16054	14565.4	39099	36857	2826.5
ASR	434.5	21746	16199.1	14511.4	39410.1	36973.5	2854.2
ASR+pts	428.8	21394.9	16025.2	14644.6	39199.2	37218	2915.6
ASE	428.3	21522.8	16205	14685.2	39261.8	37510.2	2952.4
ASE+pts	427.4	21431.9	16140.8	14657.9	39161	37417.7	2908.1
AS	437.3	22471.4	16702.1	15296.4	39596.3	38733.1	3154.5

окрестности оптимума, когда длины маршрутов почти одинаковые, ранжирование сильно ускоряет поиск наилучшего решения.

3.3. Алгоритм муравьиной колонии

В алгоритмах муравьиной колонии усиливается роль эксплуатации наилучшего решения. На каждой итерации феромоны обновляются только на ребрах наилучшего маршрута.

Для задачи коммивояжера правило обновления феромонов (2) принимает вид:

$$\tau_{ij}(t+1) = (1-p) \cdot \tau_{ij}(t) + p \cdot \Delta\tau_{ij,\epsilon}(t), \quad (5)$$

где (i, j) – ребро наилучшего маршрута, которым может быть как лучший на текущей итерации, так и наилучший маршрут с начала работы алгоритма. Для задач большой размерности хорошие результаты дает обновление феромонов по маршруту T^+ .

Правило выбора продолжения маршрута (1) изменено так: k -ый муравей с вероятностью q_0 переходит из города i в наиболее привлекательный город $z \in J_{ik}$, а с вероятностью $(1-q_0)$ выбирает город j по правилу (1). Чем больше q_0 , тем сильнее эксплуатируется опыт муравьиной колонии при синтезе новых маршрутов. Наиболее привлекательный город определяется так:

$$z = \arg \max_{j \in J_{ik}} ((\tau_{ij}(t))^\alpha \cdot (\eta_{ij})^\beta).$$

Введенные правила заставляют муравьев искать оптимум в узкой окрестности предыдущего наилучшего решения. Для поддержания корректного баланса между эксплуатацией наилучшего решения и исследованием новых областей

в алгоритмы муравьиной колонии введено следующее правило локального обновления феромонов. На каждой итерации алгоритма муравей при переходе из города i в город j “съедает” некоторое количество феромонов с ребра $(i-j)$. Это ребро теряет привлекательность для других муравьев, которые на этой итерации будут исследовать другие маршруты из городов i и j . Решения становятся разнообразнее, так как на протяжении каждой итерации распределение феромонов динамически обновляется, отражая выполненные муравьями перемещения.

3.4. Макс-минный муравьиный алгоритм

Макс-минный алгоритм отличается от базового муравьиного алгоритма следующими тремя правилами:

- 1) на каждой итерации феромоны добавляются только на ребра наилучшего маршрута по правилу (5);
- 2) количество феромонов на ребре графа ограничено диапазоном $[\tau_{min}, \tau_{max}]$;
- 3) в начале работы алгоритма количество феромонов на каждом ребре графа принимается равным τ_{max} .

Ограничения на количество феромонов позволяют разнообразить решения и тем самым избежать стагнации. Для расширения области поиска новых решений в макс-минных муравьиных алгоритмах также применяется механизм сглаживания следов (trail smoothing mechanism), со-

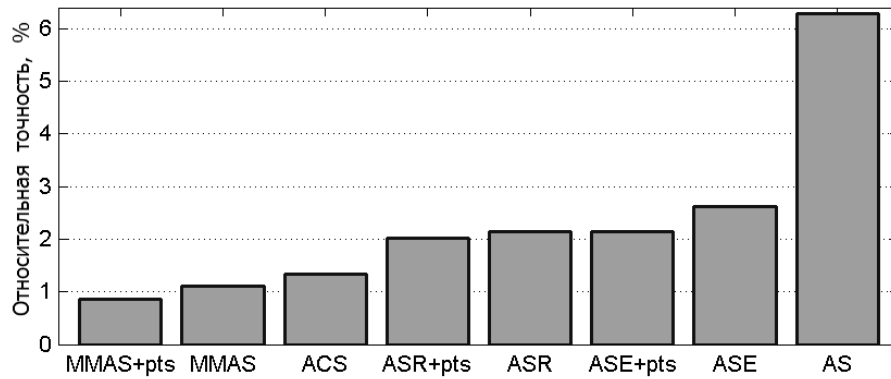


Рис. 5. Сравнение муравьиных алгоритмов по относительной точности по данным таблицы 1.

гласно которому откладываемое количество феромонов $\Delta\tau_{ij}(t)$ на ребре $(i-j)$ пропорционально $(\tau_{max} - \tau_{ij}(t))$.

Компьютерные эксперименты с задачами коммивояжера [14] показывают, что среднее время оптимизации удается значительно сократить, если феромоны обновлять не на маршруте T^+ , а на ребрах лучшего на текущей итерации маршрута. Для задач большой размерности время оптимизации сокращается при смешанной стратегии, когда на некоторых итерациях феромоны откладываются по маршруту T^+ , а на некоторых – по лучшему текущему маршруту. При этом частота обновления феромонов по маршруту T^+ должна увеличиваться во время выполнения алгоритма [14].

В таблице 1 сравниваются результаты решения задач коммивояжера из библиотеки [10] следующими алгоритмами: AS – базовый муравьиный алгоритм; ASE – базовый муравьиный алгоритм с элитными муравьями; ASR – ранговый муравьиный алгоритм; ACS – алгоритм муравьиной колонии; MMAS – макс-минный муравьиный алгоритм. Символы “+pts” означают применение механизма сглаживания следов. Все алгоритмы синтезировали одно и то же количество маршрутов: $10000 \cdot n$ для симметричных задач Eil51, KroA100, D198 и $20000 \cdot n$ для асимметричных задач Ry48p, Ft70, Kro124p и Ftv170. Числа в названии задач указывают количество городов (n). Числа в ячейках таблиц – длины найденных кратчайших маршрутов коммивояжера, усредненные за 25 прогонов. Полужирным шрифтом выделены лучшие решения.

На рис. 5 сравниваются муравьиные алгоритмы по критерию средней относительной точности:

$$\varphi = \frac{1}{7} \sum_{i=1, \dots, 7} \frac{L_i - opt_i}{opt_i} \cdot 100\%,$$

где

opt_i – длина оптимального маршрута для i -ой тестовой задачи из таблицы 1;

L_i – усредненная длина маршрутов, найденных муравьиным алгоритмом для i -ой тестовой задачи.

Из рис. 5 видно, что лучшую эффективность демонстрируют макс-минные алгоритмы, затем идут алгоритмы муравьиной колонии. Заметим, что по данным [14] для двух симметричных задач большой размерности Att532 и Rat783 [10] усредненные длины маршрутов, найденных MMAS, были короче, чем длины даже наилучших маршрутов, найденных ACS.

3.5. Лучший-худший муравьиный алгоритм

Лучший-худший алгоритм отличается от базового муравьиного алгоритма следующими тремя правилами:

- 1) феромоны добавляются на ребра наилучшего маршрута T^+ по правилу (5);
- 2) на данной итерации t феромоны испаряются только с наихудшего текущего маршрута $T^-(t)$:

$$\tau_{ij}(t+1) = (1-p) \cdot \tau_{ij}(t), \\ (i, j) \in T^-(t) \text{ и } (i, j) \notin T^+;$$

Таблица 2. Сравнение метаэвристических методов оптимизации маршрута коммивояжера [6]

Тестовая задача	Eil50	Eil75	KroA100
Имитированный отжиг (Simulated Annealing)	443	580	нет данных
Генетические алгоритмы (Genetic Algorithms)	428	545	22761
Эволюционное программирование (Evolutionary Programming)	426	542	нет данных
Алгоритмы муравьиной колонии (Ant Colony System)	425	535	21282

3) феромонный след на ребре $(i - j)$ с вероятностью p_{mut} подвергается мутации:

$$\tau_{ij}(t+1) = \begin{cases} \tau_{ij}(t) + \Delta\tau_{mut}, & \text{если } a = 0, \\ \tau_{ij}(t) - \Delta\tau_{mut}, & \text{если } a \neq 0, \end{cases}$$

где $i, j = 1, \dots, n, i \neq j$, $\Delta\tau_{mut}$ – случайное число из диапазона, зависящего от номера итерации и среднего количества феромонов на ребрах маршрута T^+ ; $a \in \{0, 1\}$ – случайное число.

Первое правило усиливает вклад наилучшего решения. Второе правило ослабляет вклад наихудшего текущего решения. Третье правило является аналогом операции мутации в генетических алгоритмах. Оно введено для диверсификации решений через расширение области поиска. При приближении стагнации, когда текущие лучшее и худшее решения отличаются лишь несколькими дугами, происходит “перезагрузка” – количество феромонов на всех ребрах принимается равным $\tau_{ij} = \tau_0, i, j = 1, \dots, n$. В работе [16] экспериментально установлено, что среди рассмотренных выше алгоритмов при решении квадратичных задач о назначениях наилучшую производительность демонстрирует лучший-худший муравьиный алгоритм.

3.6. Список кандидатов

Для задач большой размерности целесообразно использовать список кандидатов. Он представляет собой небольшой список предпочти-

тельных вершин, в которые может перейти муравей из данной вершины. Список кандидатов формируется на основе априорных знаний о задаче или на базе динамически обновляемой в процессе решения информации. Муравей выбирает вершину не из списка кандидатов только в том случае, когда он уже прошел его полностью. Список кандидатов позволяет исключить заведомо неперспективные варианты и направить муравьев на исследование наиболее обещающих маршрутов, тем самым существенно сократив область поиска.

Для задачи коммивояжера в список кандидатов включают города, расположенные по соседству. Например, для задачи с 2392 городами Pr2392 [10] оптимум можно найти, если исследовать продолжения маршрутов в 8 ближайших городов [17]. Список кандидатов может использоваться со всеми рассмотренными модификациями муравьиных алгоритмов. Впервые список кандидатов был внедрен в муравьиные алгоритмы в работе [18].

3.7. Гибридизация муравьиных алгоритмов

Наиболее часто гибридизацию муравьиных алгоритмов выполняют добавлением процедур локального поиска. На каждой итерации алгоритма методы локального поиска пытаются улучшить найденные муравьями решения. Обычно применяют итеративные процедуры, которые циклически улучшают решение, пока не будет достигнут локальный оптимум. Удачный выбор метода локального поиска существенно ускоряет решение задачи оптимизации. Для задачи коммивояжера часто используют про-

Таблица 3. Сравнение метаэвристических методов решения квадратичной задачи о назначениях [6]

Тестовая задача	Nugent (12)	Nugent (15)	Nugent (20)	Nugent (30)	Elshafei (19)	Krarup (30)
Имитированный отжиг (Simulated Annealing)	578	1150	2570	6128	17937024	89800
Табу-поиск (Tabu Search)	578	1150	2570	6124	17212548	90090
Генетические алгоритмы (Genetic Algorithms)	588	1160	2688	6748	17640584	108830
Эволюционные стратегии (Evolution Strategies)	598	1168	2654	6308	19600212	97880
Муравьиные алгоритмы (Ant System)	578	1150	2598	6232	18122850	92490
Муравьиные алгоритмы с локальным поиском (Ant System with local search)	578	1150	2570	6128	17212548	88900

цедуры локального поиска 2-opt, 3-opt и Лин-Кернигхана (Lin-Kernighan), которые улучшают маршрут заменой двух, трех и переменного количества дуг, соответственно.

В последнее время исследуются возможности гибридизации муравьиных алгоритмов с другими метаэвристическими методами оптимизации и природными вычислениями, прежде всего с генетическими алгоритмами. Существует два основных направления такой гибридизации: островная схема и использование генетических операций в муравьиных алгоритмах. Островная схема представляет собой параллельное решение задачи генетическим и муравьиным алгоритмами с обменом решениями через определенное время. На сегодня предприняты попытки решения задачи коммивояжера и задачи маршрутизации транспорта (Vehicle Routing Problem) по островной муравьино-генетической схеме [19–21], однако делать какие-либо обобщения еще рано. Второе направление муравьино-генетической гибридизации реализовано лучшим-худшим муравьиным алгоритмом [15, 16], в котором количество феромонов на ребрах графа изменяется посредством мутации.

Интересным является использование нечетких <если-то> правил при выборе маршрутов виртуальными муравьями. Такая гибридизация

муравьиных алгоритмов обеспечивает составление хороших транспортных расписаний при нечетких исходных данных [22].

4. ОБЗОР ПРИМЕНЕНИЯ МУРАВЬИНЫХ АЛГОРИТМОВ ОПТИМИЗАЦИИ

Приведенный в статье муравьиный алгоритм оптимизации маршрута коммивояжера после незначительных модификаций может использоваться для решения различных комбинаторных задач: квадратичной задачи о назначениях (Quadratic Assignment Problem), задачи маршрутизации транспорта (Vehicle Routing Problem), задачи календарного планирования (Job-Shop Schedule Planing), задачи раскраски графа (Graph Coloring Problem), задачи о кратчайшей общей суперпоследовательности (Shortest Common Supersequence), многомерной задачи о рюкзаке (Multiple Knapsack) и др. Для решения этих задач муравьиными алгоритмами необходимо: (1) свести их к поиску кратчайшего пути на некотором графе; (2) определить механизмы инициализации и обновления феромонов; (3) назначить эвристические правила выбора маршрута.

Муравьиные алгоритмы решают задачи дискретной оптимизации не хуже других метаэвристических технологий и некоторых проблемно-ориентированных методов. При этом обеспечивается хороший баланс между точностью реше-

Таблица 4. Сравнение метаэвристических методов маршрутизации транспорта [23]

Число заказчиков	Ранговый муравьиный алгоритм с декомпозицией задачи и локальным поиском (D-Ant)			Генетический алгоритм (Genetic Algorithm)		Гранулированный табу-поиск (Granular Tabu Search)	
	Среднее решение за 10 прогонов	Лучшее решение	Время решения, мин	Найденное решение	Время решения, мин	Найденное решение	Время решения, мин
200	6460.98	6460.98	7.13	6460.98	1.04	6697.53	2.38
255	589.28	586.87	139.27	596.89	14.32	593.35	11.67
300	1007.81	1007.07	32.55	1018.74	39.33	1016.83	21.45
399	932.58	927.27	158.93	933.74	78.50	936.04	33.12
420	1836.87	1834.79	239.47	1846.55	210.42	1915.83	43.05
480	13958.68	13816.98	240.00	13728.8	187.6	14910.62	15.13

ния и временем оптимизации. В качестве примера сравниваются различные метаэвристические методы оптимизации маршрута коммивояжера (таблица 2) и решения квадратичной задачи о назначениях (таблица 3). Числа в ячейках таблиц – значения критерия оптимальности для решений, найденных соответствующими методами. В таблице 4 сравниваются три метода маршрутизации транспорта для задач большой размерности. Время оптимизации пересчитано под производительность процессора Pentium-900MHz. Полужирным шрифтом выделены наилучшие на сегодня решения.

Муравьиные алгоритмы применимы и для стохастической комбинаторной оптимизации. Теоретическая сходимость к глобальному оптимуму стохастических муравьиных алгоритмов продемонстрирована в [24].

Среди прикладных работ выделим исследования по применению муравьиных алгоритмов:

- *в инженерии*: многокритериальное проектирование водных ирригационных сетей [25], оптимизация водоснабжения [26], оптимизация структуры съемочных геодезических сетей [27], оптимизации надежности с помощью резервирования [28], эргономическое проектирование компьютерных клавиатур [29], размещение данных в памяти суперкомпьютера [30], динамическая оптимизация химических процессов [31];
- *в менеджменте*: составление университетских расписаний [32], оптимизация размещения автобусных остановок [33], согласование транспортных расписаний [22];

- *в биологии*: предсказание структуры протеина по его аминокислотным цепочкам [34];
- *в искусстве*: составление музыкальных произведений [35] и написание картин [36].

Хорошие результаты получены при использовании муравьиных алгоритмов для обучения Байесовских сетей [37], классических логических правил [38] и нечетких баз знаний [39], а также для экстракции нечетких правил из экспериментальных данных [40]. Корпорация Siemens на основе муравьиной оптимизации, нечеткой логики и роевой кластеризации разработала гибридный метод управления логистикой. Пилотное использование этого метода на мюнхенских складах корпорации снизило задержки по доставке товаров на 44% [41].

Высокую эффективность демонстрируют муравьиные алгоритмы при оптимизации распределенных нестационарных систем. Примером может служить нахождение муравьиными алгоритмами оптимальных трафиков в телекоммуникационных сетях [6, 42]. В таблице 5 приведены результаты маршрутизации в американской сети NSFNET, содержащей 14 узлов с 21 двунаправленной линией связи. Сравнивались следующие алгоритмы: AntNet – муравьиный алгоритм; OSRF – официальный интернетовский алгоритм маршрутизации; SRF – алгоритм, использующий динамическую метрику при расчете стоимости соединений; Daemon – аппроксимация идеального алгоритма маршрутизации;

Таблица 5. Сравнение алгоритмов маршрутизации для сети NSFNET [42]

Алгоритм	AntNet	OSRF	SRF	Daemon	BF
Средняя задержка передачи сообщения, с	0.93 (0.2)	5.85 (1.43)	3.58 (0.83)	0.10 (0.03)	4.27 (1.22)
Пропускная способность, $\times 10^7$ бит/с	2.392 (0.011)	2.100 (0.002)	2.284 (0.003)	2.403 (0.010)	1.410 (0.047)

BF – алгоритм Беллмана-Форда. В таблице 5 приведены средние значения времени задержки и пропускной способности при интенсивной загрузке сети. Числа в скобках – значения среднеквадратических отклонений при 10-кратном прогоне алгоритмов. Информацию о других применениях муравьиных алгоритмов можно найти в обзорных статьях [43, 44] и в книгах [6, 45].

5. ВЫВОДЫ

В последнее время биологические науки оказывают значительное влияние на математику и компьютерные технологии – зарождается технобиология – наука, использующая биологические принципы для совершенствования техники и информационных процессов [8]. К технобиологии можно отнести и муравьиные алгоритмы, которые основаны на механизмах самоорганизации социальных насекомых. Муравьиные алгоритмы предложены в начале девяностых годов и за 10 лет из “игрушечных” демонстраций переросли в важное направление в оптимизации.

В статье на примере задачи коммивояжера показано, как решать задачи комбинаторной оптимизации с помощью муравьиных алгоритмов. Рассмотрены основные методы улучшения муравьиных алгоритмов, использующие элитных муравьев, процедуры локального поиска, механизмы сглаживания следов и списки кандидатов. Проанализированы современные модификации муравьиных алгоритмов: ранговый алгоритм, макс-минный алгоритм, алгоритм муравьиной колонии и лучший-худший алгоритм.

Муравьиные алгоритмы применимы к задачам оптимизации, которые сводятся к поиску кратчайшего пути на графе с некоторыми ограничениями. Виртуальные муравьи выбирают маршруты на графе по вероятностному прави-

лу, учитывая количество феромонов и эвристические методы решения конкретного типа задачи. Компьютерные эксперименты свидетельствуют, что муравьиные алгоритмы обеспечивают хороший баланс между точностью решения и временем оптимизации. Особенно хорошие результаты получаются при муравьиной оптимизации динамических систем с нестационарными параметрами, например, телекоммуникационных и компьютерных сетей. Важным свойством муравьиной оптимизации является неконвергентность – даже после большого числа итераций одновременно исследуется множество решений, поэтому алгоритмы надолго не застревают в локальных экстремумах. Все это позволяет рекомендовать муравьиные алгоритмы для решения сложных комбинаторных задач оптимизации.

СПИСОК ЛИТЕРАТУРЫ

1. *Захаров А.А.* Муравей, семья, колония. М.: Наука, 1978.
2. *Dorigo M.* Optimization, Learning and Natural Algorithms. PhD Thesis. Dipartimento di Elettronica, Politecnico Di Milano. Italy. 1992. 140 p.
3. *Dorigo M., Maniezzo V., Colormi A.* The Ant System: Optimization by a Colony of Cooperating Agents // IEEE Trans. on Systems, Man and Cybernetics. Part B. 1996. V. 26. № 1. P. 29–41.
4. *Леванова Т.В., Лореш М.А.* Алгоритм муравьиной колонии и имитации отжига для задачи о р-медиане // Автоматика и Телемеханика. 2004. № 3. С. 80–88.
5. *Штовба С.Д.* Муравьиные алгоритмы // Exponenta Pro. Математика в приложениях. 2003. № 4. С. 70–75.
6. *Bonavear E., Dorigo M.* Swarm Intelligence: from Natural to Artificial Systems. Oxford University Press, 1999. 307 p.

7. *Dorigo M.* Swarm Intelligence, Ant Algorithms and Ant Colony Optimization // Reader for CEU Summer University Course "Complex System". Budapest, Central European University. 2001. P. 1–38.
8. *Швецова Н.* Эволюционная биология и высокие технологии: симбиоз будущего. CNews.ru. <http://www.cnews.ru/newcom/index.shtml?2002/09/27/136108>.
9. *Goss S., Aron S., Deneubourg J.L., Pasteels J.M.* Self-Organized Shortcuts in the Argentine Ant // *Naturwissenschaften*. 1989. № 76. P. 579–581.
10. TSPLIB. <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>.
11. *Gen M., Cheng R.* Genetic Algorithms and Engineering Design. John Wiley & Sons, 1997. 352 p.
12. *Bullnheimer B., Hartl R.F., Strauss C.* A New Rank-Based Version of the Ant System: A Computational Study // *Central European Journal for Operations Research and Economics*. 1999. V. 7. № 1. P. 25–38.
13. *Dorigo M., Gambardella L.M.* Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem // *IEEE Trans. on Evolutionary Computation*. 1997. V. 1. № 1. P. 53–66.
14. *Stutzle T., Hoos H.H.* MAX-MIN Ant System // *Future Generation Computer Systems*. 2000. V. 16. № 8. P. 889–914.
15. *Cordon O., Fernandez de Viana I., Moreno L.* A New ACO Model Integrating Evolutionary Concepts: The Best-Worst Ant System. Proc. of ANTS2000 – From Ant Colonies to Artificial Ants: A Series of Inter. Workshops on Ant Algorithms. Brussels, Belgium. 2000. P. 22–29.
16. *Cordon O., Fernandez de Viana I., Herrera F.* Analysis of the Best-Worst Ant Systems and Its Variants on the QAP. Proc. of the Third International Workshop on Ant Algorithms "ANTS 2002" // *Lecture Notes in Computer Science 2463*. Springer-Verlag, 2002. P. 228–234.
17. *Reinelt G.* The Traveling Salesman: Computational Solutions for TSP Applications // *Lecture Notes in Computer Science*. V. 840. Berlin: Springer-Verlag, 1994.
18. *Gambardella L.M., Dorigo M.* Solving Symmetric and Asymmetric TSPs by Ant Colonies. Proc. of the IEEE Conference on Evolutionary Computation "ICEC'96". USA, Piscataway. 1996. P. 622–627.
19. *Reimann M., Shtovba S., Nepomuceno E.* A Hybrid Ant Colony Optimization and Genetic Algorithm Approach for Vehicle Routing Problems Solving // *Student Papers of Complex Systems Summer School-2001*. Hungary, Budapest. P. 134–141.
20. *Pilat M., White T.* Using Genetic Algorithm to Optimize ACS-TSP. Proc. of the Third International Workshop on Ant Algorithms "ANTS 2002" // *Lecture Notes in Computer Science 2463*. Springer-Verlag, 2002. P. 282–287.
21. *Acan A.* GAACO: A GA + ACO Hybrid for Faster and Better Search Capability. Proc. of the Third International Workshop on Ant Algorithms "ANTS 2002" // *Lecture Notes in Computer Science 2463*. Springer-Verlag, 2002. P. 300–301.
22. *Lucic P.* Modeling Transportation Problems Using Concepts of Swarm Intelligence and Soft Computing. PhD Thesis. Civil Engineering Department. Virginia Polytechnic Institute and State University. Virginia, USA. 2002. 141 p.
23. *Reimann M.* Ant Based Optimization in Good Transportation. PhD Thesis. University of Vienna. Vienna, Austria. 2002. 149 p.
24. *Gutjahr W.J.* A Converging ACO Algorithm for Stochastic Combinatorial Optimization. Proc. of SAFA-2003 – Stochastic Algorithms: Foundations and Applications // *Lecture Notes in Computer Science 2827*. Springer-Verlag, 2003. P. 10–25.
25. *Mariano C.E., Morales E.* MOAQ: An Ant-Q Algorithm for Multiple Objective Optimization Problems. Proc. of Genetic and Evolutionary Computation Conference (GECCO-99). USA, San-Francisco. 1999. V. 1. P. 894–901.
26. *Maier H.R., Simpson A.R., Zecchin A.C., Wai Kuan Foong, Kuang Yeow Phang, Hsin Yeow Seah, Chan Lim Tan.* Ant Colony Optimization for Design of Water Distribution Systems // *Journal of Water Resources Planning and Management*. V. 129. Issue 3. P. 200–209.
27. *Hussian Aziz Saleh.* Поведение муравьев можно успешно использовать для разработки съемочных сетей GPS оптимальной структуры. Русский перевод на <http://www.agr.ru/projects/ants/>.
28. *Liang Yun-Chia, Smith A.E.* An Ant System Approach to Redundancy Allocation. Proc. of the Congress on Evolutionary Computation (CEC '99). 1999. V. 2.
29. *Eggers J., Feillet D., Kehl S., Wagner M.O., Yannou B.* Optimization of the Keyboard Arrangement Problem using an Ant Colony Algorithm //

- European Journal of Operational Research. 2003. № 148. P. 672–686.
30. *Rodrigues A.* Application of Ant Colony Optimization to Data Distribution in Memory in Computer Systems. Abstracts of 7th Annual Swarm Researchers Meeting “SwarmFest 2003”. USA, Notre Dame. 2003. Full paper at <http://www.nd.edu/~arodrig6/>.
 31. *Rajesh J., Gupta K., Kusumakar H.S., Jayaraman V.K., Kulkarni B.D.* Dynamic Optimization of Chemical Processes Using Ant Colony Framework // *Computers and Chemistry*. 2001. V. 25. Issue 6. P. 583–595.
 32. *Socha K., Knowles J., Smples M.* A MAX-MIN Ant System for the University Course Timetabling Problem. Proc. of the Third International Workshop on Ant Algorithms “ANTS 2002” // *Lecture Notes in Computer Science 2463*. Springer-Verlag, 2002. P. 1–13.
 33. *De Jong J.* Multiple Ant Colony Systems for the Busstop Allocation Problem. Master Thesis. Department of Philology. University of Utrecht. Utrecht, Holland. 2001. 59 p.
 34. *Shmygelska A., Hoos H.* An Improved Ant Colony Optimization Algorithm for the 2D HP Protein Folding Problem. Proc. of the Sixteenth Canadian Conference on Artificial Intelligence (AI’2003). Canada. 2003.
 35. *Gueret C., Monmarche N., Slimane M.* Ants Can Play Music. Proc. of Fourth International Workshop on Ant Colony Optimization and Swarm Intelligence (ANTS 2004) // *Lecture Notes in Computer Science 3172*. Springer-Verlag, 2004. P. 310–317.
 36. *Aupetit S., Bordeau V., Monmarche N., Slimane M., Venturini G.* Interactive Evolution of Ant Paintings. Proc. of IEEE Congress on Evolutionary Computation. Camberra: IEEE Press, 2003. P. 1376–1383.
 37. *De Campos L.M., Gamez J.A., Puerta J.M.* Learning Bayesian Networks by Ant Colony Optimisation: Searching in Two Different Spaces // *Mathware & Soft Computing*. 2002. № 9.
 38. *Raspinelli J.M., Lopes H.S., Freitas A.A.* Data Mining with an Ant Colony Optimization Algorithm // *IEEE Trans. on Evolutionary Computation*. Special issue on Ant Colony Algorithms. 2002. V. 6. № 4. P. 321–332.
 39. *Cassillas J., Cordon O., Herrera F.* Learning Fuzzy Rules Using Ant Colony Optimization Algorithms. Proc. of ANTS2000 – From Ant Colonies to Artificial Ants: A Series of Inter. Workshops on Ant Algorithms. Brussels, Belgium. 2000. P. 13–21.
 40. *Cassillas J., Cordon O., Fernandez de Viana I., Herrera F.* Learning Cooperative Linguistic Fuzzy Rules Using the Best-Worst Ant System Algorithm // *International Journal of Intelligent Systems*. 2005. V. 20. P. 433–452.
 41. Component Transport Follows the Ant Trail. <http://www.siemens.com>. September 23, 2004.
 42. *Caro G.D., Dorigo M.* Anet: a Mobile Agents Approach to Adaptive Routing. Technical Report IRIDA 97-12. IRIDA – Universite Libre de Bruxelles. Brussels, Belgium. 1997. 27 p.
 43. *Dorigo M., Stutzle T.* The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances / Glover F., Kochenberger G. (eds.) *Handbook of Metaheuristics*. Norwell, MA: Kluwer Academic Publishers, 2002.
 44. *Cordon O., Herrera F., Stutzle T.* A Review on the Ant Colony Optimization Metaheuristic: Basis, Models and New Trends // *Mathware & Soft Computing*. 2002. № 9.
 45. *Dorigo M., Stutzle T.* *Ant Colony Optimization*. Bradford Book, 2004.