Исследование метаэвристических методов решения задач маршрутизации на плоскости

А.И.Дегтярева, Н.Д. Ганелина
Новосибирский государственный технический университет,
Новосибирск, Россия
alena03@ngs.ru, natalie_ganelina@ngs.ru

Аннотация

Статья посвящена исследованию метаэвристических методов решения задач маршрутизации на плоскости. В статье описано, как алгоритм построения муравьиной колонии (ACS) может быть успешно применен для решения таких задач. Рассматривается модель задачи проектирования оптимального маршрута обхода геометрических объектов на плоскости, описывается алгоритм ACS применительно к поставленной задаче. Приводятся результаты проведенных экспериментов.

Ключевые слова: *маршрут, колония муравьев, феромон.*

1. ВВЕДЕНИЕ

Рассматривается задача проектирования оптимального маршрута обхода геометрических объектов на плоскости. Необходимо обойти все объекты на плоскости, осуществляя обход каждого объекта лишь один раз таким образом, чтобы минимизировать общий пройденный путь. Причем, путь должен начинаться и заканчиваться в начале координатной системы.

При решении поставленной задачи, в основном, применяются эвристические методы и комбинированные алгоритмы, так как для её решения на количестве объектов из реальных практических задач не существует точных методов, дающих результат за приемлемое время.

Целью этого исследования является подтверждение того, что метаэвристический алгоритм построения муравьиной колонии (ACS) может быть успешно применен для проектирования оптимального маршрута обхода геометрических объектов. Метаэвристическая технология является итеративной процедурой для быстрого и эффективного нахождения решения сложных оптимизационных задач. Алгоритм ACS основан на наблюдении за действиями муравьев, которые находят кратчайший путь от гнезда до места расположения пищи, оставляя дорожку феромона.

Алгоритм ACS, являющийся частным случаем алгоритма оптимизации передвижений в муравьиной колонии, может быть успешно применен для решения сложных комплексных задач оптимизации. Цель решения сложных комплексных задач оптимизации – поиск и определение наиболее подходящего решения для оптимизации (нахождения минимума или максимума) целевой функции (цены, точности, времени, расстояния и т.п.) из дискретного множества возможных решений. Типичный пример решения подобной задачи – задача о назначениях, задача календарного планирования, задача маршрутизации транспорта. Эти задачи возникают в бизнесе, инженерии, производстве и многих других областях.

Второй раздел статьи содержит описание модели задачи. В третьем разделе приводится обоснование выбора метода решения. Алгоритм ACS, используемый для решения поставленной задачи, описывается в четвертом разделе. Пятый раздел посвящен проведенным экспериментам, результаты которых показывают, что применение алгоритма ACS для построения оптимального маршрута обхода геометрических объектов является эффективным.

2. МОДЕЛЬ ЗАДАЧИ

Рассматриваемая задача заключается в проектировании оптимального маршрута обхода геометрических объектов на плоскости. Имеется двумерный чертеж, на котором изображены объекты, представленные многоугольниками. Многоугольники не должны пересекаться. Они могут быть выпуклыми или невыпуклыми, могут находиться внутри друг друга. Вершины многоугольников представляют собой их вероятные точки обхода.

Необходимо найти маршрут, траектория которого должна проходить только через одну точку обхода каждого многоугольника ровно один раз, начинаясь и заканчиваясь в начале координатной системы. Оптимальный маршрут должен иметь минимальную длину среди множества возможных маршрутов. При этом возможным является проведение маршрута через внутренние точки многоугольников. Траектория оптимального маршрута составляет результат проектирования — оптимальное проектное решение. Особенность и сложность задачи заключается в том, что задача имеет дискретно-непрерывную структуру.

Имеется множество геометрических объектов на плоскости. Контур каждого объекта имеет начальную точку обхода (x_i,y_i), принадлежащую i-му контуру (i = 1,2,…,n). Обозначим расстояние между начальными точками обхода i-го и j-го контура через L_{ij} (i,j=0,1,…,n). Причём, равенство нулю индекса i (или j) означает соответствие началу координатной системы, т. е. точке (0, 0). Необходимо найти кратчайший маршрут k^* из множества K допустимых маршрутов $K = (i_1(x_{i_1}, y_{i_1}), i_2(x_{i_2}, y_{i_2}),..., i_n(x_{i_n}, y_{i_n}))$,

где (i_1, i_2, …, i_n) — произвольная перестановка чисел 1, 2,…, n.

Задача принимает вид

$$F(k^*) = \min(L_{0i_1}(x_{i_1}, y_{i_1}) + \sum_{j=1}^{n-1} L_{i_j i_{j+1}}(x_{i_j}, y_{i_j}, x_{i_{j+1}}, y_{i_{j+1}}) + L_{i_n 0}(x_{i_n}, y_{i_n}))$$

Существует система геометрических и технологических ограничений. Геометрические ограничения для чертежа определяются размещением объектов, условиями их взаимного непересечения, а также способами размещения объектов на чертеже. В качестве геометрических ограничений для поставленной задачи рассмотрим контуры объектов, являющиеся совокупностями отрезков прямых. Контуры обходимых объектов можно описать аналитически в виде совокупности параметрических уравнений:

$$x_{ij} = x_{ij1} + (x_{ij2} - x_{ij1})t_{ij},$$
$$y_{ij} = y_{ij1} + (y_{ij2} - y_{ij1})t_{ij},$$

где m ≤ i ≤ m+n-1 (m — количество отрезков и n — количество многоугольников), 0 ≤ j ≤ p-1 (p_i — количество отрезков в i-ом многоугольнике), $0 \le t_{ij} \le 1$, x_{ij1}, y_{ij1} и x_{ij2}, y_{ij2} — координаты конечных точек j-го отрезка i-ого многоугольника.

Кроме геометрических ограничений, существует ряд технологических ограничений, связанных со спецификой исполнительного инструмента. В частности, в качестве таких ограничений могут являться следующие:

- маршрут начинается и заканчивается в начале координатной системы,
- исключается возможность пересечения отрезков маршрута,
- ограничение возможностей движения оборудования и др.

Таким образом, процесс формирования оптимального проектного решения заключается в генерации порядка обхода контуров с созданием траектории переходов от точки обхода одного контура к точке обхода другого и выборе в качестве оптимального проектного решения траектории с минимальной длиной.

3. МЕТОД РЕШЕНИЯ

Все известные алгоритмы, которые могут быть применены для получения глобально оптимального решения данной задачи, требуют исчерпывающего поиска маршрута в объеме экспоненциальном относительно числа пунктов обхода плана. Возможность практической реализации исчерпывающего поиска для сложных планов существенно ограничена наличными вычислительными ресурсами. Для нахождения приближенного решения задачи в работе было решено провести исследование на основании метаэвристического метода – муравьиной колонии (Ant Colony Optimization).

Муравьиная колония разрабатывает сеть оптимальных маршрутов к источникам пищи, используя вероятностный метод проб и ошибок. Муравьи отмечают свои маршруты, откладывая химическое вещество, называемое феромоном. Другие муравьи, находя феромоновый след, начинают следовать ему и закрепляют его добавочным феромоном.

Такое поведение является формой автокаталитического действия: чем больше муравьев следуют по маршруту, тем более привлекательным он становится. Поведение муравьев вызвало появление алгоритма оптимизации муравьиной колонией (ACO), разработанного Marco Dorigo из IRIDIA и успешно применяемого для решения многих комплексных задач оптимизации. Основная идея, лежащая в основе алгоритма ACS заключается в использовании механизма положительной обратной связи, который помогает найти наилучшее возможное решение в сложных задачах оптимизации. Результаты, получаемые при применении алгоритмов, основанных на действиях муравьиной колонии, так же хороши, как и при применении алгоритмов, основанных на метаэвристических алгоритмах общего назначения.

Алгоритм ACS, являющийся частным случаем ACO, использует колонию виртуальных муравьев, ведущих себя как кооперативные агенты в математическом пространстве, в котором они могут искать и подтверждать найденные пути (решения) в целях поиска одного наилучшего. Эти пути могут содержать в себе очень много информации. Искусственные муравьи имеют некоторую память и не действуют вслепую. Кроме того, принимается, что время дискретно, а не непрерывно. В рамках алгоритма ACS, где каждый муравей двигается по замкнутому маршруту, количество феромона вдоль маршрута увеличивается в зависимости от качества решения, найденного муравьем.

4. АЛГОРИТМ ACS

Рассмотрим алгоритм ACS применительно к поставленной задаче. Вершины многоугольников будем называть “городами”. Будем считать, что d_{ij} - расстояние между городами i и j, которое рассчитывается как Эвклидово расстояние: $d_{ij} = sqrt((x_i - x_j)^2 + (y_i - y_j)^2)$,

а задача задана полносвязным графом G(N, E), где N - набор городов (|N| = n), E - дуги их соединяющие, а $d_{ij} = d_{ji}$.

Предположим, что для решения задачи используем m муравьёв. В начале каждой итерации муравьи располагаются случайным образом по n городам, а в конце итерации муравьи "умирают". Каждый муравей за время одной итерации должен сделать количество ходов равное числу объектов, для того, чтобы построить полный путь. Ход муравья можно описать следующими правилами:

- он выбирает город, в который двинуться, с вероятностью, которая является функцией от расстояния между текущим городом и выбираемым, а также количества феромона на этом пути;
- для контроля правильности решения, найденного муравьём, используется список уже посещённых городов (память муравья). Муравей выбирает направление движения только среди городов, не принадлежащих уже посещенным хотя бы в одной вершине многоугольникам;
- при глобальном обновлении феромона (по завершению каждой итерации), его добавление происходит только к дугам, принадлежащим глобальному кратчайшему пути;
- пока муравьи ищут решение, происходит локальное обновление феромона (аналогия с настоящими муравьями, которые откладывают феромон в процессе своего движения).

Изначально все дуги (все возможные пути) покрываются начальным значением феромона τ_0 (достаточно маленькое число). Пусть $\tau_{ij}(t)$ - величина феромона на пути между городами i и j на t-ой итерации. В конце каждой итерации происходит глобальное обновление феромона: со всех дуг испаряется определённое количество феромона, а к дугам, принадлежащим глобальному кратчайшему пути, добавляется некоторое его количество, причем, чем короче путь, тем больше феромона добавляется. Глобальное обновление феромона происходит по следующей формуле:

$$\tau_{ij}(t+1) = (1-g) * \tau_{ij}(t) + g * \Delta\tau_{ij},$$

где g – коэффициент глобального испарения, $\Delta\tau_{ij}$ – величина добавления феромона, которая рассчитывается по формуле:

$$\Delta\tau_{ij} = \begin{cases} L_{gb}^{-1}, (i,j) \in global-best-tour \\ 0, otherwise \end{cases},$$

где L_{gb} – длина наименьшего пути, найденного со времени начала поиска.

Таким образом, только на лучшем и имеющем кратчайшую длину пути будет накапливаться феромон, а параметр g предназначен для усиления поисков вокруг этого пути.

В процессе построения решения муравьи одновременно делают ход, в конце которого изменяют количество феромона на дугах, по которым они прошли – происходит локальное обновление феромона. При этом к этим дугам добавляется некоторое количество феромона, часть которого тут же испаряется. Затем, одновременно совершая следующий ход, они для выбора нового города пользуются локально обновленным количеством феромона. Локальное обновление феромона происходит по следующей формуле:

$$\tau_{ij}(t) = (1-\rho) * \tau_{ij}(t) + \rho * \Delta\tau_{ij},$$

где ρ - коэффициент локального испарения феромона, $\Delta\tau_{ij}$ - величина добавления феромона. В данной реализации алгоритма величина добавления феромона при локальном обновлении равна начальному его значению: $\Delta\tau_{ij} = \tau_0$.

Согласно локальному обновлению, уровень феромона на дуге конструируемого пути в высокой степени зависит от значения параметра ρ. При большом значении ρ уровень феромона снижается, а это, в свою очередь, уменьшает шанс того, что другой муравей выберет то же самое решение и того, что последовательность поиска будет более разнообразной.

Рассмотрим функцию выбора нового города. Находясь в городе i, муравей выбирает город s по следующей формуле:

$$s = \begin{cases} \arg\max\limits_{j \in allowed_k} \left\{ [\tau_{ij}(t)]^{\alpha} * [\eta_{ij}]^{\beta} \right\}, q < q_0 \\ r, otherwise \end{cases},$$

где $allowed_k$ - список городов, ещё не пройденных k-ым муравьём и не принадлежащих многоугольникам, уже посещенным k-ым муравьём хотя бы в одной вершине, η_{ij} - так называемая "видимость" - величина обратно пропорциональная расстоянию между городами ($\eta_{ij} = 1/d_{ij}$), α и β - параметры, контролирующие относительный приоритет феромона на пути или видимости следующего города, q - случайное число на отрезке [0, 1], q_0 - параметр баланса между использованием накопленных знаний и исследованием новых решений ($0 \le q_0 \le 1$), r - случайный город, выбранный на основе вероятностей, посчитанных по следующей формуле:

$$p_{ij}^k(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum\limits_{m \in allowed_k} [\tau_{im}(t)]^{\alpha} \cdot [\eta_{im}]^{\beta}}, j \in allowed_k \\ 0, otherwise \end{cases}.$$

Как уже отмечалось выше, у каждого искусственного муравья есть своя память. В этой памяти муравей хранит, так называемый, список "табу" - список пройденных им уже городов ($tabu_k$).

Параметры α и β определяют важность феромона и видимости пути. Если α = 0, то феромон не играет никакой роли, и на вероятность выбора следующего города влияет только видимость (близлежащие города выбираются с большей вероятностью). Если β = 0, то для выбора города используется только феромон.

Параметр q_0 играет следующую роль: если случайное число q меньше q_0, то происходит использование накопленных знаний - выбирается более близкий город, на пути к которому больше феромона. В противном случае ($q \ge q_0$) - происходит случайный выбор города на основе вероятностей их выбора (поиск новых решений).

В данной реализации алгоритма итерации проходят до тех пор, пока их количество не будет равно NC_{max}. В качестве лучшего решения используется путь с наименьшей длиной - $\min(L_k)$. Для его отображения используется список табу муравья, который нашёл это решение. Так как список табу заполняется городами в порядке их посещения, он содержит корректный путь.

5. ПРОВЕДЕНИЕ ЭКСПЕРИМЕНТОВ

В этом разделе оценивается эффективность алгоритма ACS и влияние параметров на качество получаемого решения. Эксперименты проводились на примере, изображенном на рисунке 1.

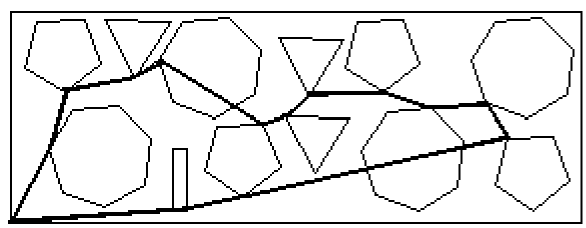

Рис. 1. Задача на многоугольниках

Для оценки эффективности алгоритма ACS проведён сравнительный анализ его с другими методами, результаты анализа представлены в следующей таблице:

Метод	Лучшее решение	Среднее решение	σ, %
ACS	86.7303	88.6459	2
Метод «ближайшего соседа»	91.862	93.2048	7
Симуляция отжига (Simulated Annealing)	93.6139	94.2630	11

В таблице представлены следующие данные:

- лучшее решение, найденное методом;
- среднее решение, равное среднему значению решений, найденных за 10 испытаний;

- среднее отклонение от рекорда задачи в процентах, которое рассчитано по формуле:

$$\sigma = \frac{L_{cp.} - L^*}{L^*} \cdot 100\% ,$$

где $L_{cp.}$ — средняя длина маршрутов для определённого метода, L^* — рекорд задачи (длина наилучшего маршрута для всех методов).

Среднее отклонение от рекорда задачи σ будем считать показателем эффективности метода. При этом, чем меньше σ, тем лучше будет решение, найденное методом, и, следовательно, тем эффективнее будет используемый метод.

Таким образом, алгоритм ACS является эффективнее методов «ближайшего соседа» и «симуляции отжига» для решения рассматриваемой задачи. Причем рекорд задачи был найден именно алгоритмом ACS, что подтверждает сделанные выводы.

Сходимость алгоритма ACS можно рассматривать как переход итерационного процесса в устойчивое состояние и получение эффективного решения за определённое количество итераций. Как видно из экспериментальных данных (см. рисунок 2), с увеличением номера итерации стабильно улучшается качество решения. Это обусловлено механизмом работы алгоритма ACS: текущее решение качественно лучше предыдущего.

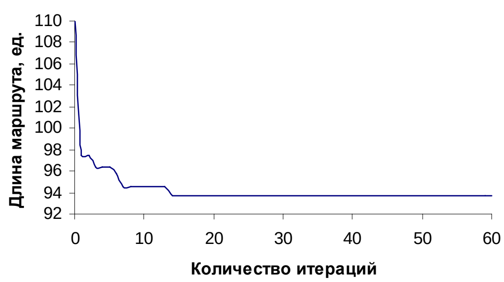

Рис. 2. Динамика изменения длины маршрута

Развивая процесс поиска, необходимо аккуратно выбирать параметры при применении алгоритма ACS для проектирования оптимального маршрута обхода геометрических объектов. Это параметры, относящиеся к размерности и типу задачи, конструктивные и управляющие параметры.

Конструктивные элементы описывают задачу в рамках ACS. Они содержат:

- матрицу расстояний (расстояние между городами);
- число муравьев m и их начальное расположение (число муравьев может быть задано равным числу объектов, числу городов или быть произвольным, муравьи могут быть расположены случайным образом или в некоторой начальной позиции);
- перечень запретов (связанных с каждым муравьем и запрещающих посещать один и тот же объект дважды).

Управляющие параметры обуславливают работу собственно алгоритма ACS и в основном связаны с информацией о феромонах. Они содержат:

- управляющий приоритетом феромона параметр α,
- управляющий приоритетом видимости пути параметр β,
- коэффициент глобального испарения g ,
- коэффициент локального испарения ρ,
- критерий останова (в данной реализации алгоритма прекращающего процесс после заданного числа итераций),
- начальное значение феромона $\tau_{0,}$
- параметр баланса между использованием накопленных знаний и исследованием новых решений q_0 .

Длина лучшего пути, найденного метаэвристическим ACS методом, оказалась лучше, чем решение найденное методами «ближайшего соседа» и «симуляция отжига». Процесс вычислений по алгоритму ACS был прерван после достижения критерия останова при количестве итераций NC_{max} = 20. Управляющие параметры имели следующие значения: α = 0.5, β = 5, g =0.5, ρ = 0.1, τ_0 = 0.00001, q_0 = 0.3. Число муравьев было равно количеству объектов: m = 12. При этом было получено решение $\min(L_k)$ = 86.7303. Лучший путь согласно списку табу изображен на рис. 3.

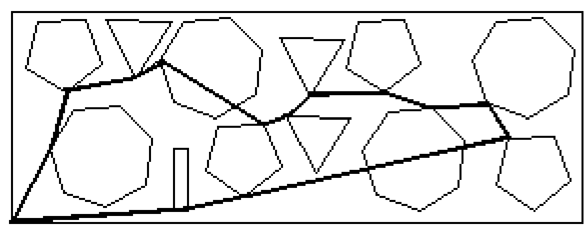

Рис. 3. Лучший путь, найденный ACS, при NC_{max}= 20, α= 0.5, β = 5, g =0.5, ρ = 0.1, τ_0 = 0.00001, q_0 = 0.3, m = 12

Для параметра ρ наилучшие результаты работы алгоритма были получены при значении ρ = 0.1, это означает, что необходимо использовать значительный уровень разнообразия поисков. При большом значении ρ уровень феромона снижается, что, в свою очередь, уменьшает шанс того, что последовательность поиска будет более разнообразной.

Параметр g используется при глобальном обновлении для задания направления поиска, наилучшие результаты работы ACS были получены при его значении менее 0.6. С другой стороны, большие значения g уводили процесс поиска в сторону от наилучшего решения.

Экспериментальным путем получено, что для ACS наилучшим сочетанием параметров α и β соответственно являются следующие сочетания: (3,1), (3,3), (0.5,5), (1,5) (см. рис. 4).

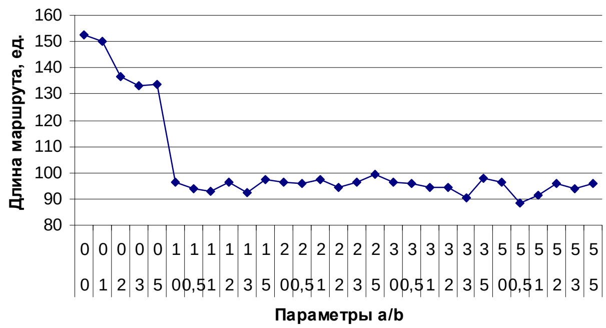

Рис. 4. Влияние параметров α и β на длину маршрута

В соответствии с рисунком 5 можно сделать вывод о том, что использование накопленных знаний может привести не только к улучшению решения, но и к его ухудшению. Оптимальными значениями исследуемого параметра являются: (0.3, 0.7).

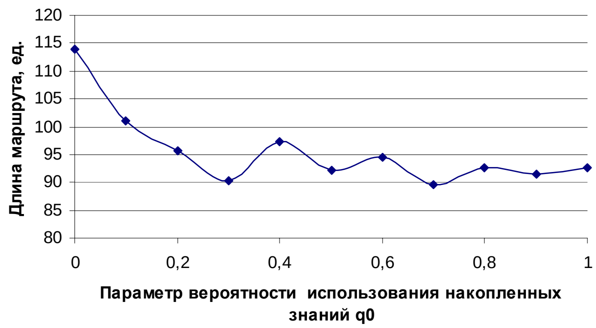

Рис. 5. Зависимость длины маршрута от уровня используемых знаний

6. ЗАКЛЮЧЕНИЕ

Метаэвристический алгоритм построения муравьиной колонии (ACS) был успешно применен для проектирования оптимального маршрута обхода геометрических объектов на плоскости. Основная идея, лежащая в основе ACS, заключается в использовании механизма положительной обратной связи, который помогает найти наилучшее возможное решение в сложных задачах оптимизации. То есть, если в данном городе муравей должен выбрать между различными вариантами и если фактически выбранные результаты будут хорошими, то в будущем такой выбор будет более желателен, чем предыдущий. Этот подход является многообещающим из-за его общности и эффективности в обнаружении очень хороших решений сложных проблем.

Исследование параметров, проведенное в рамках настоящей работы, позволяет подобрать значения исходных параметров для наилучшей работы алгоритма. Эти рекомендации справедливы для всего класса задач. Однако результаты, полученные экспериментальным путем, на настоящий момент не позволяют построить какой-либо функциональной зависимости между параметрами и качеством решений.

Последующие исследования должны включать в себя анализ параметров для задач ещё большей размерности и добавление объектов произвольной формы. Другое важное направление исследований в этой области – добавление различных локальных стратегий поиска к основной программе для получения более качественных результатов.

7. БИБЛИОГРАФИЯ

[1] Marco Dorigo, Vittorio Maniezzo, Alberto Colorni. *The Ant System: Optimization by a colony of cooperating agents* URL: http://ftp://iridia.ulb.ac.be/pub/mdorigo/journals/IJ.15-BIOSYS97.pdf

[2] Marco Dorigo, Luca Maria Gambardella. *Ant colonies for the traveling salesman problem* URL: ftp://iridia.ulb.ac.be/pub/mdorigo/journals/IJ.10-SMC96.pdf

[3]Hussian Aziz Saleh. *Поведение муравьев можно успешно использовать для разработки съемочных сетей GPS* /GPS World, Sept. 2002. Пер. с англ Навгеоком, 2003 URL: http://www.navgeocom.ru/projects/ants/

8. ОБ АВТОРАХ

Алена Ивановна Дегтярева – магистрант кафедры АСУ Новосибирский Государственный Технический Университет. Адрес: Новосибирск, 630092, ул. Немировича-Данченко, 136, НГТУ, 7-й учебный корпус, факультет АВТ, кафедра АСУ. E-mail: alena03@ngs.ru

Наталья Давидовна Ганелина – аспирант кафедры АСУ, Новосибирский Государственный Технический Университет. Адрес: Новосибирск, 630092, ул. Немировича-Данченко, 136, НГТУ, 7-й учебный корпус, факультет АВТ, кафедра АСУ. E-mail: natalie_ganelina@ngs.ru

Metaheuristic Approach for Routing problems in the Plane

Alena I. Degtyareva, Natalie D. Ganelina
Novosibirsk State Technical University,
Novosibirsk, Russia
alena03@ngs.ru, natalie_ganelina@ngs.ru

Abstract

The paper is devoted to the research of metaheuristic methods for routing problems in the plane. It's shown that Ant colony method can be successfully applied to such a problem. A problem of constructing the optimal route of traversal of geometrical objects in the plane is considered. Ant colony algorithm applied to the problem stated is described. The experimental results are presented.

Keywords: *route, ant colony, pheromone.*

About the authors

Alena I. Degtyareva is a MS student, Automated Control Systems Department, Novosibirsk State Technical University. His contact email is alena03@ngs.ru

Natalie D. Ganelina is a PhD student, Automated Control Systems Department, Novosibirsk State Technical University. His contact email is natalie_ganelina@ngs.ru.