

О модификации муравьиного алгоритма для решения задачи коммивояжёра

О.С. Нургаянова
Факультет информатики и робототехники
Уфимский государственный авиационный
технический университет
Уфа, Россия
e-mail: onurgayanova@yandex.ru

И.И. Янбаев
Факультет информатики и робототехники
Уфимский государственный авиационный
технический университет
Уфа, Россия
e-mail: iyanbaev@mail.ru

Аннотация¹

В статье рассматриваются вопросы модификации муравьиного алгоритма для решения задачи коммивояжёра. Внесение изменений в привычную структуру алгоритма позволяет заметно сократить временные издержки, либо повысить точность получаемого ответа путём увеличения входных параметров. Увеличение числа задействованных муравьёв, либо числа итераций, позволяет повысить точность ответа ввиду использования сильной стороны представленного алгоритма – рассмотрение большего числа граней, за которыми может скрываться более короткий маршрут.

Муравьиный алгоритм, задача коммивояжёра, маршрут, расстояние, метаэвристика, транспортная логистика, временные издержки.

1. Введение

Задача построения минимальных по длине маршрутов не является новой и оригинальной и существует ряд известных алгоритмов и программных продуктов, позволяющих успешно её решать. Однако при этом складывается ситуация, при которой процесс оптимизации транспортной логистики более доступен крупным компаниям, при этом вопрос снижения затрат для малых предприятий зачастую является более важным в высококонкурентной среде большей ограниченности ресурсов. В связи с этим актуальной является задача поиска алгоритма, который будет работать эффективно и при этом не будет затратным в отношении потребления ресурсов и временных издержек.

Специфика малых компаний включает в себя такие требования, как: необходимость быстрого расчёта; повышенная потребность в надёжном алгоритме, который не может выродиться – бизнес-процессы должны быть максимально безопасны. В принципе, эти особенности характерны для компании любого

Труды Шестой всероссийской научной
Конференции "Информационные технологии
интеллектуальной поддержки принятия
решений", 28-31 мая, Уфа-Ставрополь, Россия,
2018

уровня, однако у небольших компаний отсутствуют временной и финансовый резервы и меньший запас прочности как таковой. Вторая группа особенностей заключается в том, что в маршруте заложено относительно небольшое число (в большинстве случаев этот показатель меньше 25) точек, откуда следует необходимость в получении более точных решений. Небольшое число точек в маршруте, кроме того, что в целом облегчает задачу, также снижает привлекательность эвристических методов решения задачи из-за малой разницы в решениях.

2. Описание задачи и алгоритма решения

Для решения задачи коммивояжёра известны две группы методов: точные и приближённые. Точные методы предоставляют оптимальное решение, однако показывают не самые лучшие показатели в плане времени выполнения. Большая часть исследований посвящена улучшению неточных методов решения задачи коммивояжёра, что является следствием отнесения задачи нахождения кратчайшего маршрута к категории трансвычислительных и накладывает свои ограничения [1].

Муравьиный алгоритм является полиномиальным алгоритмом для нахождения приближённых решений задачи коммивояжёра, а также решения аналогичных задач поиска маршрутов на графах, обладающий достаточно высокой эффективностью в своём классе. В основе муравьиного алгоритма лежит вероятностный подход к поиску оптимального пути. Преимуществами алгоритма являются невысокая погрешность найденного решения, низкие временные затраты при работе с графами большой размерности, модифицируемость алгоритма и возможность распараллеливания. В алгоритме представлена многократность: одновременный итерационный поиск муравьями оптимального маршрута, каждый из которых выступает как независимый коммивояжёр и за одну итерацию алгоритма совершает обход всех вершин графа [2].

В качестве положительной обратной связи в задаче выступает маркирование феромонного следа. Вероятность включения вершины в маршрут муравья прямо пропорциональна количеству феромона, из чего следует случайность: чем короче маршрут, тем

большее количество феромона будет оставлено на нём. Отрицательной обратной связью является время испарения феромонов: время не должно быть слишком большим, чтобы поиск маршрута не свёлся к единственному варианту, но и слишком малое время испарения приведёт к некооперированному поведению муравьёв, и оптимальный путь будет иметь значительную погрешность.

Для перехода муравья из одного города в другой необходимо учитывать три составляющие: память муравья, видимость и феромонный след.

Память муравья (список табу) – это список уже посещенных муравьем городов. При использовании списка табу муравей гарантированно не попадет в один и тот же город дважды. Этот список возрастает при совершении маршрута и обнуляется в начале каждой итерации алгоритма. Пусть $J_{i,k}$ — список городов, которые еще предстоит посетить муравью k , находящемуся в городе i . Список $J_{i,k}$ является дополнением к памяти.

Видимость – это обратная расстоянию величина $\eta_{ij} = 1 / D_{ij}$, где D_{ij} – расстояние между городами i и j . Видимость представляет собой локальную информацию, определяющая стремление муравья посетить город j из города i – чем выше этот показатель, тем ближе город и тем сильнее «желание» муравья его посетить.

Феромонный след можно охарактеризовать как подтвержденное опытом желание муравья посетить город j из города i . В отличие от видимости, феромонный след постоянно обновляется на протяжении работы алгоритма, отображая полученный муравьями опыт. Количество феромона на ребре (i,j) на итерации t обозначается как $\tau_{ij}(t)$.

Вероятность перехода муравья k в город j из города i определяется следующей формулой:

$$P_{ij}(t) = \frac{\tau_{ij}(t)^\alpha \left(\frac{1}{d_{ij}}\right)^\beta}{\sum_{l \in J_{i,t} \setminus j} \tau_{il}(t)^\alpha \left(\frac{1}{d_{il}}\right)^\beta} \quad (1)$$

Параметры α и β задают значимость уровня феромона и видимости города при выборе следующего города. При $\alpha=0$ будет выбран ближайший город, что соответствует «жадному» алгоритму. Если $\beta=0$, тогда работает лишь феромонное усиление, что влечет за собой быстрое вырождение маршрутов к одному субоптимальному решению. Непосредственный выбор следующего города осуществляется в произвольном порядке: для каждого муравья генерируется маршрут движения из текущего местонахождения случайным образом, с учётом вероятностей перехода. По завершении маршрута каждый муравей k откладывает на ребре (i,j) некоторое количество феромона.

Введём обозначение: CLN_{ij} – список, колония муравьёв, использующих ребро (i,j) . Тогда

обновление феромона на ребре (i, j) представляется в данном алгоритме следующим образом:

$$\tau_{ij}(t + 1) = (1 - \rho)\tau_{ij}(t) + \sum_{k \in CLN_{ij}} \frac{q}{L_k} \quad (2)$$

При реализации в алгоритм были внесены изменения. В формуле (1), при вычислении вероятности выбора рассматриваемого ребра, вычислялась сумма предпочтительности остальных доступных вершин, что оказало значительное влияние на расход ресурсов: при увеличении входных параметров в каждой итерации часть вычислений повторяется множество раз, в особенности при увеличении феромонов на определённых гранях и более высокой вероятности их поиска. Общая схема работы муравьиного алгоритма приведена на рисунке 1.



Рис. 1. Общая схема муравьиного алгоритма

Расчёт вероятности перехода по определённой грани представлен на рисунке 2.



Рис. 2. Расчёт вероятности перехода по грани

С учётом этих особенностей было принято решение при начале каждой итерации вносить в память результаты вычислений данной суммы. Модификация алгоритма представлена на рисунке 3.



Рис. 3. Расчёт вероятности перехода по грани с использованием результатов прошлых вычислений

3. Заключение и выводы

В качестве входных параметров муравьиного алгоритма были использованы так называемые параметры Дориго [3]. Марко Дориго в своём исследовании обнаружил параметры, которые в совокупности являются более предпочтительными. Перечень параметров был взят следующий:

$$\begin{aligned}\alpha &= 1 \\ \beta &= 5 \\ \rho &= 0,99\end{aligned}$$

Было произведено сравнение работы модифицированной версии алгоритма с классической исходной. Алгоритмы были реализованы на языке Javascript. По окончании проведения тестирования на пяти тысячах случайных маршрутах с числом точек от 15 до 35 были получены следующие результаты:

1. Маршрут подвергается сокращению в среднем на 20,51%.
2. Модификация алгоритма была задействована в среднем в 80% итераций от общего их количества и позволила снизить среднее время нахождения маршрута более чем на 14%.
3. Случаи вырождения алгоритма и возвращения некорректных результатов отсутствуют.

4. Случаи непредвиденного завершения работы алгоритма отсутствуют.

5. Модификация алгоритма не потребовала дополнительных больших объёмов оперативной памяти. При указанных параметрах объём используемой оперативной памяти не превысил 200 Мбайт.

6. Время поиска при выше обозначенных входных параметрах не превысило 45 секунд (для маршрутов, состоящих из 50 точек).

Стоит отметить, что преимущества использования модифицированного алгоритма становятся ещё более очевидными при увеличении размерности задачи.

В рамках проведения вычислительных экспериментов была реализована также версия алгоритма с использованием элитных муравьёв. Были опробованы варианты использования одной четвертой части колонии муравьёв в качестве элитных, одной трети и половины подобных муравьёв от общей популяции. По результатам проведения тестирования на 2,5 тысячах случайных маршрутах различной размерности (от 15 до 40 городов в маршруте) существенного влияния на итоговый результат выявлено не было. Имеющиеся различия незначительны и объясняются использованием вероятностной модели выбора вершин.

Как показали результаты тестирования, модификация оказалась достаточно эффективной и может оказаться полезной при практическом применении.

Исследование частично поддержано грантом РФФИ 18-07-00193-а.

Список используемых источников

1. Введение в исследование операций: учеб. пособие. /А.И. Ерзин. - Новосиб.: Ун-т. Новосибирск, 2006. – 100 с.
2. Математические модели и методы принятия решений: учеб. пособие. /Э. Х. Гимади, Н. И. Глебов. - Новосиб.: Ун-т. Новосибирск, 2008. – 163 с.
3. Marco Dorigo, Gianni Di Caro, and Luca M. Gambardella. Ant algorithms for discrete optimization. *Artificial Life*, 5(2):137–172, 1999