

DOI 10.37882/2223–2966.2021.11.03

РАЗРАБОТКА АЛГОРИТМОВ ДЛЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ ПОСТРОЕНИЯ ОПТИМАЛЬНОГО МАРШРУТА ДОСТАВКИ¹

DEVELOPMENT OF ALGORITHMS FOR THE INFORMATION SYSTEM FOR CONSTRUCTING THE OPTIMAL DELIVERY ROUTE

D. Brykin
S. Pestova

Summary. The article presents the results of the study of the speed and optimality of the algorithms for solving the traveling salesman problem for companies that need to optimize their own delivery service. Five algorithms were analyzed and three of them were compared. For the subsequent work, it was decided to use an exhaustive search algorithm for short routes and a combination of simulated annealing and genetic algorithms for longer routes. Algorithm tests have shown their strengths and weaknesses. The algorithms were evaluated for their speed and the resulting route length.

The genetic algorithm and the simulated annealing algorithm have shown good efficiency if the number of points in the route is large. For short routes, a brute-force algorithm is better suited. It was decided to use their combinations depending on the calculated route.

Keywords: logistics, routing, optimization, traveling salesman problem, metaheuristic algorithms.

Брыкин Дмитрий Олегович
Омский государственный технический
университет
dezorb@gmail.com

Пестова Светлана Юрьевна
К.п.н., Сибирский государственный автомобильно-
дорожный университет, г. Омск
pestova_sv@mail.ru

Аннотация. В статье представлены результаты выполнения исследования скорости и оптимальности работы алгоритмов по решению задачи коммивояжера для фирм, которым необходимо оптимизировать собственную службу доставки. Проведен анализ 5 алгоритмов и были сравнены 3 из них. Для последующей работы решено использовать алгоритм полного перебора для коротких маршрутов и комбинацию алгоритмов имитации отжига и генетического для маршрутов большей протяженности. Тесты алгоритмов показали их сильные и слабые стороны. Алгоритмы были оценены по их скорости работы и получившейся длинны маршрута.

Генетический алгоритм и алгоритм имитации отжига показали хорошую эффективность, если количество точек в маршруте велико. Для коротких маршрутов лучше подходит алгоритм полного перебора. Было решено использовать их комбинации в зависимости от рассчитываемого маршрута.

Ключевые слова: логистика, маршрутизация, оптимизация, задача коммивояжера, метаэвристические алгоритмы.

Введение

В последние годы, рынок доставки неуклонно растет. По оценке РБК, рост в 2019 году составил 22%, а за 2020 год оказался уже более 50% [4]. Однако, такой быстрый рост создает свои проблемы. На плечи логиста ложится все больше нагрузки, а компании не спешат расширять штат доставок.

Если взглянуть на организацию процесса доставки, то она являет собой многоступенчатый обширный процесс, состоящий из таких вещей, как: общая логистика маршрута, предотправная подготовка товаров к отгрузке, планирование и осуществление работ на складе, упаковку, распределение товаров, осуществление

процесса страхования перевозок, а так же, в некоторых случаях, осуществление таможенных услуг. Одной из основных компонент организации грузоперевозок, является выбор подходящих для ваших задач транспортных средств и построение их маршрута.

Конечным продуктом будет программное обеспечение для построения оптимального маршрута доставки с возможностью обмена данными о заказах с другими информационными системами. Можно выделить задачи, которые необходимо решить для достижения цели работы: исследование алгоритмов поиска оптимального маршрута и выбор оптимального для реализуемой задачи, доработка алгоритмов под предметную область; создание MVP (прототипа) программного ком-

¹ Исследование выполнено при финансовой поддержке Фонда Содействия инновациям в рамках договора № 15663ГУ/2020.

плекса для настольных и мобильных ОС; тестирование и отладка программного обеспечения.

В своей работе я постарался рассмотреть оптимизацию некоторых из этих аспектов, а именно построение оптимального маршрута для доставки и кластеризацию точек доставки для более эффективного использования транспортных средств и курьеров. Разработка будет рассмотрена в следующей статье.

Разработка рабочей версии алгоритма поиска оптимального маршрута

Рассматривая задачу, обозначенную в рамках исследования, можно выделить задачу коммивояжера, алгоритм решения которой должен быть оптимальным. В обычной постановке задачи коммивояжеру необходимо посетить N точек по замкнутому маршруту, посетив их все единожды, то есть, чтобы общая длина его пути была минимально возможной. Если рассматривать данную задачу по простому, то есть перебором всех возможных путей, которые объединяют точки, то придется просчитать все $(N-1)!/2$ возможные маршруты — следовательно, простой метод перебора всех доступных путей абсолютно не подходит при большом N ($N > 10$). Подходящим рабочим решением, то есть подразумевающим получение ответа за T , где T органично полиномом от размерности задачи. [10] Данную задачу решают логисты в службах доставки ежедневно.

Несмотря на большой возраст задачи, до сих пор нет алгоритмов, которые признанные оптимальными для данной задачи. Ниже приведено исследование по сравнению данных алгоритмов в рамках задачи по гранту.

Генетический алгоритм

Это алгоритмический процесс, основанный на биологических особенностях жизни организмов: популяция развивается в течении многих поколений, подталкиваемая законами естественного отбора по принципу «выживает сильнейший». В данном случае, приспособленность определяется минимизацией функцией расстояния всего маршрута.

Для нашей задачи, цепочка городов будет ассоциироваться с цепочкой генов. Каждый ген — набор точек маршрута, без повторений. На первом этапе надо задать начальные параметры алгоритма. Такие как, размер «популяции» (набор генов, т.е. набор различных вариантов решений), вероятность возникновения мутации и условия останова. В конце необходимо определить случайным образом первичное поколение.

Критерии останова, которые используются в генетическом алгоритме.

- ◆ нахождение глобального, либо локального минимума (менее предпочтительно);
- ◆ выходом на «плато», когда дельта функции приспособленности перестает меняться с каждым следующим поколением или меняется незначительно;
- ◆ окончание количества поколений, количество которых было задано при инициализации алгоритма;
- ◆ окончание времени, выделенного на эволюцию;

Данный процесс продолжается несколько жизненных циклов (поколений). Критерием остановки является дельта между новыми расстояниями. Задача алгоритма в том, чтобы найти точку экстремума функции. А мутации позволяют избегать локальных минимумов, за счет случайных рекомбинаций генов.

Данный алгоритм осуществляется следующим образом:

1. Методом случайного распределения создается первичный набор пробных решений (популяция):

$$P^1 = \{p_1^1 \dots p_n^1\}, p_i^1 \in X, \tag{1}$$

где X — первое поколение, n - размер популяции).

2. Оценка приспособленности текущего поколения:

$$F^k = \{f_1^k \dots f_n^k\}, f_i^k = W(p_i^k) \tag{2}$$

3. Завершение алгоритма, если выполняется критерий останова, иначе пункт 4.

4. Создание нового поколения популяции с помощью операторов селекции S , скрещивания C и мутаций M , которые описаны в формуле 3. После нее производится переход к пункту 2. [1]

$$p^{k+1} = M * C * S(p^k, f^k) \tag{3}$$

Селекция — это процесс отбора определенного количества особей от популяции, которой предстоит «выжить» на текущем этапе эволюции. Этот процесс необходим, ибо множество потомков и мутантов имеют пониженный уровень «выживаемости», и их необходимо отделить от основной популяции в процессе естественного отбора. По итогу селекции остаются лишь отобранные особи, которым предстоит участвовать в дальнейшем развитии популяции. Главный критерий отбора — значение функции приспособленности. Вся популяция фильтруется по данному критерию, после

чего берется только определенный начальный%. Создание нового поколения в популяции осуществляется с помощью кроссинговера и мутации. Кроссинговер — это процесс создания новой особи на основе родительской с учетом ограничений задачи, а именно через 1 точку нельзя проходить более 1 раза, и это необходимо предусматривать. Хромосомы разрезаются в случайной точке и обмениваются частями без повторений, с дальнейшим сдвигом и добавлением недостающих генов.

Большим преимуществом данного алгоритма является высокая вероятность покидания локальных точек экстремума, а также относительно быстрая скорость решения. Так, на тестовой виртуальной машине (4 ядра, i9-9880H, 6GB RAM), даже решение задачи с 10 точками укладывается в ± 10 секунд машинного времени, тогда полный перебор требует большее 440 секунд. На большем количестве точек разница будет еще более заметна. Недостатком же является большое количество начальных параметров, от правильной настройки которых очень зависит эффективность работы алгоритма.

Алгоритм имитации отжига

Данный метод берет за основу реальное физическое явление. Во время остывания вещества, которое вызывает кристаллизацию за счет смены агрегатного состояния, происходит создание кристаллической решетки. Это явление можно наблюдать, к примеру, при отжиге металлов. Атомы металла стремятся занять оптимальное положение с минимальным уровнем энергии в его формирующейся кристаллической решетки, и меняют свое положение в соответствии с этой целью. Однако, при понижении температуры, вероятность смены места в кристаллической решетке понижается. Имитацию данного природного процесса можно использовать с целью нахождения минимума целевой функции.

Основные шаги алгоритма

1. Выбор первичного решения и начальной температуры
2. Оценка целевой функции первичного решения
3. Основной раздел алгоритма
 1. Случайное изменение текущего решения
 2. Оценка измененного решения
 3. Применение критерия допуска
4. Уменьшение температуры и, если температура больше минимального порога, то переход к основному шагу

Первичное решение обычно генерируется случайным образом. После вычисления целевой функции, происходит переход к новому решению. Для задачи коммивояжера обычно производится перестановка

местами двух соседних точек в маршруте. В результате, появляется два решения. Текущее и измененное. Если новое решение имеет лучшее значение целевой функции, оно принимается. Иначе, новое решение принимается с вероятностью $P = \exp(-\delta E/T)$, где:

- ◆ P — вероятность принять новое решение,
- ◆ δE — величина разности целевой функции нового и предыдущего решения,
- ◆ T — текущая температура. [5]

С каждой итерацией, температура понижается. Это понижает вероятность принятия нового менее оптимального решения. Следовательно, понижается разброс значений целевой функции следующих итераций. Мною был использован метод постепенного понижения температуры с каждой итерацией, ведь главным условием является монотонное убывание температуры к нулю. Скорость убывания зависит от общего максимального количества итераций, которые были определены при запуске алгоритма.

Алгоритм Прима

Данный алгоритм используется для построения минимального остовного дерева взвешенного связного неориентированного графа. Алгоритм был придуман еще в 1930 году и показывает высокую эффективность. Сложность алгоритма составляет $O(V^2)$. Данный алгоритм может использоваться для решения задачи коммивояжера, но только при допущении, что матрица расстояний будет симметричной. Его тестирование показало хорошие результаты, однако, в большинстве реальных задач, матрица симметричной не является, что приводит к бессмысленности его реализации в рамках данной программы.

Муравьиный алгоритм

Все действия муравья являются реакцией на изменение окружающей среды или собратьев. Однако, множество муравьев в колонии формируют так называемый «роевой интеллект», что позволяет им анализировать обстановку и быть успешным видом живых существ. Одну из особенностей их поведения можно взять в качестве алгоритма для решения задачи коммивояжера.

Муравьи после себя оставляют след из феромонов, каждый раз, когда они проходят путь от муравейника до пищи и, так же, в обратном направлении. Другие муравьи чувствуют подобные следы — инстинктивно они отправятся по данному следу. В свою очередь, отправившиеся по следу муравьи, усилят его интенсивность, тем самым делая его заметнее для остальных муравьев. Так же, чем меньше путь следа до конечной точки, тем

Таблица 1. Скорость работы алгоритмов

Алгоритм	Количество точек	Время (мс)	Длина маршрута (м)
Генетический	2	73	5351
Имитации Отжига	2	3236	5351
Полный перебор	2	1	5351
Генетический	4	956	6439
Имитации Отжига	4	3753	6439
Полный перебор	4	5	6439
Генетический	6	2854	8321
Имитации Отжига	6	4275	8878
Полный перебор	6	113	8321
Генетический	8	1256	17529
Имитации Отжига	8	5486	19487
Полный перебор	8	6821	17529
Генетический	10	5598	26094
Имитации Отжига	10	5562	24890
Полный перебор	10	691133	24890

меньше тратится времени на его прохождение, следовательно, сила следа будет усиливаться с каждым новым его прохождением.

Каждый муравей запоминает свой маршрут и не возвращается в точки, которые он уже посетил. Это формирует своеобразный список запретов на переход в следующую точку. Так же, муравей ориентируется на длину маршрута и силу феромона. Феромон так же, как и в природе, со временем испарится, если по ним не проходят муравьи, которые могли бы их усилить.

Алгоритм решения имеет три стадии.

1. Генерация муравьиных особей

- ◆ Отправная точка обусловлена ограничениями задачи. Либо все особи помещаются в одну точку, либо в разные. Так же, определяется начальный показатель феромона.

2. Поиск решения

- ◆ Вероятность очередного перехода из вершины *i* в вершину *j* определяется по формуле 4

$$P_{ijk} = \frac{[\tau_{ij}(t)]^\alpha * [\eta_{ij}]^\beta}{\sum_{i \in J_{ik}} [\tau_{ij}(t)]^\alpha * [\eta_{ij}]^\beta} \tag{4}$$

где $\tau_{ij}(t)$ — уровень феромона, η_{ij} — эвристическое расстояние, α, β — константные параметры.

3. Обновление феромона

- ◆ Уровень феромона обновляется в соответствии с формулой 5

$$T_{ij}(t + 1) = (1 - p)\tau_{ij}(t) + \sum_{i \in J_{ik}} \frac{Q}{L_k(t)} \tag{5}$$

где p — интенсивность испарения, $L_k(t)$ — цена текущего решения для *k*-ого муравья, а Q — параметр,

имеющий значение порядка цены оптимального решения, то есть $Q/L_k(t)$ — феромон, откладываемый *k*-ым муравьём, использующим ребро (*i, j*).

Опять же, оптимальные значения параметров можно определить только экспериментально, они разнятся в зависимости от специфики решаемой задачи.

Данный алгоритм является сложным в реализации, и в тоже время показывает наилучшие результаты на максимальном количестве точек в маршруте. Его реализация была сочтена в данной программе излишней.

Полный перебор

Является популярным методом решения математических задач, любую NP-полную задачу можно решить, перебрав все возможные варианты решения. Однако, его сложность является экспоненциальной, т.е. быстро растет в зависимости от количества возможных комбинаций. Для нахождения оптимального маршрута на 11 точках требуется уже несколько часов работы, т.к. это означает больше 39 миллионов вариантов решений.

Таким образом, подходят для реализации алгоритмы полного перебора, имитации отжига и генетический алгоритм. Для коротких маршрутов будет использоваться алгоритм полного перебора, так как его скорость будет все еще приемлема. Для более длительных маршрутов будут запускаться генетический алгоритм и имитации отжига одновременно с целью выявления оптимального решения. Возможен запуск несколько раз подряд для больших маршрутов, для большей точности рабо-

ты. Алгоритм Прима хоть и работает быстро, но не подходит из-за специфики задачи (матрица расстояний не является симметричной). Муравьиный алгоритм сложен в реализации, его реализация оставлена до будущих версий программы.

Оценка эффективности алгоритмов, сравнение их комбинаций, разработка финальной версии алгоритма

Используя написанную программу, было произведено сравнение скорости работы алгоритмов. Постепенно повышалось количество точек маршрута, который необходимо просчитать. Это приводило к увеличению длины маршрута и увеличению времени расчетов. Расстояние считалось по дорогам г. Омска, а не «по воздуху». Матрица расстояний получена по API Google Maps.

Ниже приведена сравнительная таблица скорости работы алгоритмов на виртуальной машине (4 ядра i9-9880H, 6GB RAM). Показано среднее время из 3 замеров.

Как можно заметить, значительно превосходство метаэвристических алгоритмов начинается с 8 точек в маршруте. Особенно разница видна на 10 точках. Там три прогона одним алгоритмом происходят быстрее, чем полный перебор. А это означает, что их можно запускать одновременно и несколько раз, для выбора оптимального варианта из серии запусков. Для графов большей размерности, применение полного перебора абсолютно нецелесообразно, т.к. сложность растет экспоненциально. Другие два алгоритма справляются намного быстрее, а их допущение является приемлемым, учитывая возможность многократного запуска. Это так же открывает возможности использовать большее

количество итераций для данных алгоритмов при условии увеличения количества точек в маршруте.

В сравнении с точными алгоритмами, такими как метод ветвей и границ и метод ветвей с отсечениями, генетический алгоритм и алгоритм имитации отжига показывают превосходство, как в скорости выполнения, так и в качестве получаемых решений, т.к. не перебирают все возможные комбинации решений. При этом данные алгоритмы показывают более высокую точность, чем метод ближайшего соседа, и не имеют ограничений, которые есть, например, у алгоритма Прима. Главным минусом данных алгоритмов является зависимость от управляющих параметров, подбирать которые надо экспериментальным путем.

Заключение

Для разработки рабочей версии алгоритма поиска оптимального маршрута в ходе работы была исследована задача коммивояжера и методы ее решения с помощью эвристических и метаэвристических методов: перечислить. Также был подробно рассмотрен генетический алгоритм и алгоритм имитации отжига.

Были произведены тесты алгоритмов, с целью выявления их сильных и слабых сторон. Оценка эффективности алгоритмов проходила по критериям скорости работы и получившейся длины маршрута для различного числа точек маршрута.

Генетический алгоритм и алгоритм имитации отжига показали хорошую эффективность на большом количестве точек. Алгоритм полного перебора хорошо подходит для коротких маршрутов. Было решено использовать их комбинации в зависимости от параметра маршрутов.

ЛИТЕРАТУРА

1. Генетический алгоритм [Электронный ресурс]: — URL: http://www.machinelearning.ru/wiki/index.php?title=%D0%93%D0%B5%D0%BD%D0%B5%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC (дата обращения: 25.11.2020)
2. Гиндуллин, Р.В. Оптимизация маршрута доставки однородного груза от множества производителей множеству потребителей: дис. ... канд. ф.-м. наук / Р.В. Гиндуллин. — Уфа, 2014. — 147 с.
3. Информационная модель [Электронный ресурс]. — URL: <http://www.yaklass.ru/p/informatika/11-klass/informatcionnaia-kultura-obshchestva-i-lichnosti-13421/informatcionnye-modeli-12825/re-9caa0260-f48e-4e91-9822-b1e77826fbdf> (дата обращения: 22.01.2021)
4. Как коронокризис «вырастил» рынок доставки еды в России [Электронный ресурс]. — URL: <https://pro.rbc.ru/demo/5fbd1e0c9a79472d51fe79e8> (дата обращения 15.09.2020)
5. Пожидаев, М.С. Алгоритмы решения задачи маршрутизации транспорта: дис. ... канд. тех. наук / М.С. Пожидаев. — Томск, 2010. — 136 с.
6. Практическое применение механизма WEB-сервисов в системе «1С: Предприятие» [электронный ресурс]. — <https://infostart.ru/public/327963/> (дата обращения: 30.11.2020)
7. Самойленко Н.И. Транспортные системы большой размерности: монография / Н.И. Самойленко, А.А. Кобец, под ред. Н.И. Самойленко. — Х.: НТМТ, 2010. — 212 с.

8. Хрусталева, Е.Ю. Разработка интерфейса прикладных решений на платформе «1С: Предприятие8»/Е.Ю. Хрусталева.— М.: ООО «1С-Публишинг», 2018.— 902 с.
9. Хрусталева, Е.Ю. Технологии интеграции «1С: Предприятия 8.3» / Е.Ю. Хрусталева.— М.: ООО «1С-Публишинг», 2020.— 310 с.
10. Щербина, О.А. Метаэвристические алгоритмы для задач дискретной оптимизации / О.А. Щербина // ТВИМ.— 2014.— №1 (24).— С. 56–72

© Брыкин Дмитрий Олегович (dezor6@gmail.com), Пестова Светлана Юрьевна (pestova_sv@mail.ru).

Журнал «Современная наука: актуальные проблемы теории и практики»

