

Алгоритм поиска оптимальных маршрутов в транспортных сетях, учитывающий их перколяционные свойства и неопределенности состояния узлов

С. А. Лесько¹, А. С. Алёшкин²

Московский технологический университет, Россия, Москва (МИРЭА)

¹lesko@testor.ru, ²antony@testor.ru

Аннотация. В работе описан разработанный авторами алгоритм поиска оптимальных маршрутов в транспортных сетях с учетом их перколяционных свойств. При описании алгоритма рассматриваются атрибуты маршрутов движения транспортных средств, а также перколяционные свойства транспортных систем (зависимость вероятности блокирования сети в целом от её плотности и вероятности блокирования отдельных узлов). Учет неопределенности состояния узлов транспортной сети (возможность или не возможность перемещения), при выборе проходящего через них маршрута в рассматриваемом алгоритме, осуществляется посредством средней величины вероятности его блокирования. Которая может быть определена, как на основе методов теории перколяции (статическая оптимизация выбора маршрута), так и использования модели Вайдемана совместно с реальными данными для транспортных сетей. Во втором случае она зависит от времени, что позволяет проводить динамическую оптимизацию выбора маршрута.

Ключевые слова: транспортная сеть; поиск маршрута; оптимизация маршрута; алгоритмы поиска и оптимизации маршрута; атрибуты описания маршрутов; неопределенность состояния узлов

I. ВВЕДЕНИЕ

На сегодняшний день существует множество приложений, позволяющих автоматически прокладывать оптимальные маршруты с минимальным участием со стороны пользователей, например такие как: Google.Maps, Яндекс.Карты, Яндекс.Метро, Rusavtobus и другие. Решаемые с их помощью задачи могут быть отнесены к управлению сложными объектами в условиях неопределенности, которая обусловлена множеством факторов, в том числе поведением пользователей и исторически сложившейся, в значительной степени хаотической, транспортной инфраструктурой. Следует отметить, что существующие для управления потоками в транспортных сетях системы имеют ряд недостатков. Одна из проблем, с которой приходится сталкиваться — это отсутствие возможности внесения пользователем индивидуальных данных, с которыми работает алгоритм

управления. Второй проблемой является существующая неопределенность в возможном состоянии узлов маршрута. Никогда нельзя точно утверждать, что узел маршрута не окажется заблокированным при прибытии в него пользователя.

II. ОПИСАНИЕ СУЩЕСТВУЮЩИХ АЛГОРИТМОВ ПОИСКА И ОПТИМИЗАЦИИ МАРШРУТОВ

Проблема поиска оптимальных путей в настоящее время решается с помощью разных алгоритмов, выбор которых зависит от области применения и типа графа. Для этого могут быть, например, использованы модифицированные алгоритмы Дейкстры, алгоритмы поиска в ширину (Breadth-First Search, BFS), а так же разновидности алгоритма поиска в глубину (Depth-First search (DFS) и ряда других. Модифицированный алгоритм Дейкстры предназначен не только для поиска кратчайшего пути между парой вершин, но и для поиска набора оптимальных путей между ними. Данный алгоритм обладает высокой скоростью работы, но не позволяет определить все возможные оптимальные пути в графе. Модифицированный алгоритм поиска в глубину (BFS) позволяет найти все оптимальные маршруты между двумя вершинами, но требует ограничений, которые не позволяют алгоритму уйти в бесконечный цикл. В связи с тем, что оригинальный алгоритм BFS, так же как и его модификация, потребляют большое количество памяти, поиск оптимальных путей может потребовать в большом графе может потребовать слишком много ресурсов. Разновидности алгоритма поиска в глубину, а именно алгоритмы поиска с ограничением в глубину Depth Limited Search (DLS) и поиска в глубину с итеративным углублением Iterative Deepening Depth-First Search (IDDFS), позволяют найти кратчайший путь между парой вершин опираясь на глубину поиска в графе. Алгоритмы DLS и IDDFS можно эффективно использовать при работе с большими графами, так как они могут быть ограничены по глубине поиска в нём. Алгоритмы DLS и IDDFS, в отличие от BFS, требуют гораздо меньше памяти устройства и работает быстрее на больших и сложных графах. Рассмотрим некоторые их современных

исследований алгоритмов поиска и оптимизации маршрутов в транспортных сетях.

В статье [1] авторы рассмотрели вариант проблемы маршрутизации транспортных средств с многосекционной моделью. Исследование основано на результирующей проблеме многопролетной маршрутизации транспортных средств (MCVRP). Основная задача MCVRP – выстраивание логистики совместной доставки мультиконтейнерного груза, по определённым причинам, который должен храниться в отдельных отсеках. В данном случае минимизируется общее время движения. Устанавливается однократное посещение промежуточных точек за маршрут, ограничивается поток, общая длина, гарантируется доставка за один рейс. Для решения данной задачи, авторами был использован модифицированный алгоритм поиска с запретами.

В статье [2] авторы исследовали концепцию оптимальной стратегии движения маршрутов в рамках статического частотного распределения общественного транспорта.

В работах [3, 4] авторы также провели исследования кратчайших маршрутов, из которых следует, что на любой промежуточной остановке, пассажиры, направляющиеся к конкретному месту назначения, будут рассматривать только подмножество привлекательных линий, которые используются для минимизации времени в пути к месту назначения.

В статье [5] автор разработал генетический алгоритм вычисления кратчайших маршрутов, основанный на метаэвристическом методе для получения приемлемого и оптимального решения с меньшим временем вычисления.

Исследования, описанные в работе [6] основаны на модифицированном ant-алгоритме движения муравьёв, который является одним из наиболее эффективных полиномиальных алгоритмов поиска решений для решения проблем оптимизации.

Перспективным направлением развития систем управления транспортной сетью является создание удобных инструментов помогающих пользователям строить не только оптимальные, но и удобные маршруты, а для этого необходимо разработать новые, более эффективные модели и алгоритмы оптимизации маршрутов в транспортной сети, учитывающие неопределенность в состоянии её узлов.

III. ПЕРКОЛЯЦИОННЫЕ СВОЙСТВА ТРАНСПОРТНЫХ СЕТЕЙ И МОДЕЛИРОВАНИЕ ИХ СТРУКТУРЫ

A. Моделирование структуры транспортных систем

Моделирование поведения транспорта на дорогах и выбор оптимальных маршрутов перемещения необходимо проводить с учетом структуры реальных сетей. Для реализации целей моделирования может быть выбран открытый, общедоступный онлайн ресурс географических карт Open Street Map (OSM), который содержит данные о

дорогах, улицах, объектах инфраструктуры, вокзалах и многих других объектах по всему миру. Все данные Open Street Map можно условно разделить на три основные группы: типы данных, описывающие в виде иерархических связей сам объект, как некую пространственную сущность; информационная — описание объекта, не имеющее к его географической структуре прямого отношения (название, физические и прочие свойства); служебные атрибуты объекта, необходимые для организации процесса хранения и обработки информации в виде набора данных, такие как уникальный идентификатор, состояние объекта в базе, время последней правки объекта в базе и т.д.

B. Нахождение порога перколяции в сетях

Между узлами сети (перекрестками) по связям (дорогам) перемещаются автомобили, потоки которых регулируются светофорами. Они открывают на некоторое время, то или иное направление движения. Когда интенсивность движения увеличивается, то автомобили начинают скапливаться и образуется очередь. Если число машин в очереди достигает для данного i - го направления на j - ом перекрестке некоторый критический порог (обозначим его $L_{i,j}$) возникает пробка (узел транспортной сети блокируется). Величину этого порога можно рассматривать как критерий потери проводимости данного узла. Например, можно ввести понятие вероятности блокирования данного узла как отношение числа машин стоящих в очереди к величине этого критического порога. Существует теория перколяции, которая позволяет связать вероятность блокирования отдельных узлов с потерей проводимости всей сетью в целом (достижение порога перколяции). Значение величины порога перколяции транспортной сети, а также вероятности блокирования отдельных узлов может быть использовано, как один из критериев для поиска в ней оптимального маршрута. Одним из основных вопросов, на которые отвечает теория перколяции, — при какой доле p_c проводящих узлов возникает проводящая цепочка, соединяющая противоположные стороны сетки? Величина этой доли называется порогом перколяции. Для изучения случайных сетей с множеством связей аналитических моделей описания перколяционных процессов не существует и их исследование возможно только методами численного моделирования [7]. Для этого необходимо сначала выбрать в сети пару различных произвольных узлов. Затем необходимо задать вероятность блокирования отдельного узла сети и в соответствии с величиной выбранной вероятности случайным образом блокировать в сети данную долю узлов от их общего числа. После этого необходимо определить есть ли в сети хотя бы один свободный путь между выбранными узлами. Затем, аналогичным образом эта процедура проводится при заданной вероятности для других произвольных пар узлов. После этого проводится статистическое усреднение результатов по отдельным экспериментам (по всем рассмотренным парам узлов). Далее, с определенным шагом, изменяется величина вероятности блокирования отдельного узла сети, и блокируется случайным образом ещё некоторое число узлов, а процедура поиска свободного пути повторяется. И т.д. до тех пор, пока не будут

блокированы все узлы. После проведения всех экспериментов проводится усреднение и определяется среднее значения порога перколяции сети (значения вероятности блокировки отдельных узлов, при которой наблюдается скачек вероятности блокирования для всей сети в целом). Задача нахождения порога перколяции при блокировке узлов в случайной сети была решена нами ранее с помощью численного моделирования в работе [9]. Полученные данные линеаризуются (со значением коэффициента корреляции, равным 0,98), в координатах: $\ln P(x)$ в зависимости от $z=1/x$ (натуральный логарифм порога перколяции – величина обратная среднему числу связей x , приходящихся на один узел), что позволяет использовать для описания зависимости порога перколяции от плотности сети уравнение: $y = -1,65z + 0,02$ (где y это $\ln P(x)$).

IV. АЛГОРИТМ ПОИСКА ОПТИМАЛЬНЫХ МАРШРУТОВ ДВИЖЕНИЯ С УЧЕТОМ ПЕРКОЛЯЦИОННЫХ СВОЙСТВ СЕТИ

Для поиска оптимального маршрута в графе транспортной системы необходимо ещё учитывать перколяционные свойства сети (вероятности блокирования отдельных узлов и величину порога перколяции всей сети в целом). Для этого можно использовать следующий подход:

- Строим на основе OpenStreetMap (OSM) граф реальной транспортной сети.
- Далее можно осуществить поиск оптимальных маршрутов в графе транспортной системы, используя два варианта выбора критерия учета неопределенности в состоянии узлов транспортной сети.

Статическая оптимизация. При выборе маршрута для каждого узла находим число его связей, и на основании описанной выше теоретической зависимости порога перколяции сети от среднего числа связей, приходящихся на узел (плотности сети), находим “идеальное” значение максимально допустимой вероятности блокировки для данного узла. При выборе оптимального маршрута будем учитывать среднее значение этих “идеальных” значений вероятности блокировки (усреднение рассчитывается как сумма по всем узлам выбранного маршрута, деленная на число узлов в маршруте). Это значение можно рассматривать как среднюю величину вероятности блокирования всего рассматриваемого маршрута. Она может быть использована в качестве параметра описывающего неопределенность состояний узлов транспортной сети. Одним из критериев оптимальности выбора маршрута будем считать минимальное значение из всех средних вероятностей его блокирования. В данном случае мы будем иметь дело со статическим критерием, величина которого не зависит от времени.

Динамическая оптимизация. Используя модель Вайдемана для описания движения на перекрестках, и реализовав её, например, в среде имитационного моделирования PTV Vissim, находим величину критического порога ($L_{i,j}$) для данного перекрестка и

выбранного направления движения на нем. Далее используя данные с общедоступных ресурсов типа: Google.Maps, Яндекс.Карты, Яндекс.Метро, Rusavtobus и других получаем реальные данные об очередях на узлах предполагаемых маршрутов. Затем определяем отношение числа машин стоящих в очереди к величине критического порога, т.е. вероятность его блокирования в режиме реального времени (динамически изменяющийся параметр). Далее, можно, как и в предыдущем случае рассчитать среднюю величину вероятности блокирования всего рассматриваемого маршрута. Однако здесь будет преимущество, суть которого состоит в том, что выбираемый маршрут можно динамически перестраивать в процессе движения. В результате исследования проблемы поиска оптимальных маршрутов, с учетом перколяционных свойств транспортной системы, был разработан алгоритм, который основывается на базе классического алгоритма поиска в глубину Depth Limited Search (DLS). Одним из минусов классического алгоритма Depth Limited Search является длительное время его выполнения, так как происходит поиск всех возможных путей из точки отправления в точку прибытия. По этой причине поиск маршрута в графе нужно оптимизировать, а алгоритм ограничить минимально возможным количеством пересадок. Кроме того, нужно ввести условие, предотвращающее петли (циклы) поиска маршрутов. Результатом работы алгоритма является выбор набора оптимальных путей $R = \{\{L_1, C_1\}, \dots, \{L_k, C_k\}\}$, содержащих путь L и критерии его оптимальности C . Для нахождения наиболее оптимального пути нужно отсортировать полученный в результате работы алгоритма набор близких к оптимальному путей, по нужному критерию. При использовании данного подхода транспортную сеть можно представить в виде графа: $G=(V, E, O)$, где G – граф транспортной сети, $V=\{i_1, \dots, i_n\}$ – множество вершин i , E – множество дуг (i, j) с длиной $c_{ij} \geq 0$, O – множество не оптимизированных вершин $\{i_1, \dots, i_m\}$. В качестве критериев оптимальности C маршрута можно указать:

- Время, затрачиваемое на маршрут (T).
- Стоимость маршрута (S).
- Количество пересадок на маршруте (N).
- Минимальное среднее значение “идеальной” вероятности блокировки для всех узлов маршрута, найденной с помощью методов теории перколяции (статический параметр), или зависящая от времени средняя вероятность блокировки для всех узлов маршрута, найденной с помощью модели Вайдемана (динамический параметр).

Для работы алгоритма требуется указать атрибуты маршрута: вершины отправления и прибытия (i, j) , а так же задать максимально допустимое число пересадок, N_{\max} (вершины i и j помечаются как не оптимизируемые, $O = \{i, j\}$); начальные критерии оптимальности $C = \{0,0,0,0\}$; последнюю посещенную вершину, $d = i$. Работа алгоритма состоит из следующих шагов:

1. Если $d = j$, сохранить результаты $\{L, C\}$ в R и завершить итерацию или алгоритм поиска.
2. Проверить каждую дугу, соединяющую вершины d и b , если таковых нет, перейти на шаг 10.
3. Если вершина b уже была посещена на этом маршруте L , перейти к шагу 2.
4. Если количество пересадок достигло заданного максимума, $N = N_{\max}$, а вершина b принадлежит отличному от точки прибытия или маршрута типу j , то необходимо перейти к шагу 2.
5. Если вершина имеет связи с уже посещенными вершинами, перейти к шагу 2.
6. Если вершина b имеет отрицательную задержку до следующего транспорта, перейти к шагу 2.
7. Дублировать путь L и критерии оптимальности C , добавить в путь L_2 новую вершину b .
8. Обновить значения критериев оптимальности C_2 в соответствии с новым путём (d, b) и вершиной b .
9. Запустить новую итерацию с $d = b$.
10. Завершить итерацию или алгоритм поиска.

После выполнения алгоритм возвращает: набор оптимальных путей (R) и оптимизированный граф G . Так как алгоритм обхода в глубину работает медленно, то для увеличения производительности нужно прибегнуть к оптимизации графа, а именно сокращению количества вершин и путей которые подходят под критерии оптимизации. Оптимизация графа начинается с создания копии всего графа. Далее в цикле выполняется процедура оптимизации с условием, что оптимизация возможна.

Критериями оптимизации, под которые попадают вершины, являются:

- существуют две вершины i и j , для которых выполняется условие: $(i, j) \in E$;
- каждая из вершин $\{i, j\}$ должна иметь не более двух дуг;
- вершины $\{i, j\}$ принадлежат одному маршруту;
- все связанные с $\{i, j\}$ вершины должны принадлежать одному маршруту.

При выполнении данных критериев вершины и соединяющие их дуги могут быть объединены в одну вершину, которая будет сочетать все показатели оптимальности объединенных вершин, такие как время, затраченное на их преодоление и их стоимость.

Если для работы алгоритма требуется вершина, которая находится в объединенной вершине, объединенная вершина разбивается на вершины и дуги, которые были включены в неё при оптимизации. После разбиения, нужная для работы алгоритм вершина помечается как не оптимизируемая и проводится новый цикл оптимизации.

Процедура оптимизации состоит из нахождения пары вершин, которые соответствуют критериям оптимизации.

Если пара вершин удовлетворяющих критериям была найдена, осуществляется процедура изменения графа:

1. Новая вершина $in+1$, которая обладает суммой критериев оптимальности вершин $\{i_1, i_2\}$, а так же дуги (i_1, i_2) .
2. Вершина $in+1$ добавляется в граф G , а вершины i и j удаляются из него, $i, (n+1) \in V, \{i_1, i_2\} \notin V$.
3. Дуги, соединяющие смежные вершины с вершинами $\{i_1, i_2\}$, меняются на дуги, соединяющие смежные вершины с вершиной $in+1$.
4. Если алгоритму требуется вершина, которая была сокращена при оптимизации, выполняется процедура её восстановления:
5. В графе G производится поиск оптимизированной вершины i , в которую при оптимизации попала искомая вершина j .
6. В граф G добавляются все вершины и дуги, включенные в оптимизированную вершину i .
7. Все дуги, соединяющие вершину i и смежные с ней вершины, меняются на те, что существовали в графе G до оптимизации.
8. Из графа G исключается оптимизированная вершина i и её дуги.
9. Вершина j помечается как неоптимизированная, $j \in O$.
10. Выполняется процедура оптимизации графа.

СПИСОК ЛИТЕРАТУРЫ

- [1] Silvestrin Paulo Vitor, Marcus Ritt. An Iterated Tabu Search for the Multi-compartment Vehicle Routing Problem. // Computers & Operations Research. 2017. Issue 81. P. 192-202.
- [2] Trozzi V., Kaparias I., Bell M., Gentile G. A dynamic route choice model for public transport networks with boarding queues. // Journal Transportation Planning and Technology. 2013. Issue 36. P. 44-61.
- [3] Nguyen S., Pallottino S. Equilibrium traffic assignment for large scale transit networks. // European Journal of Operational Research. 1988. № 37(2). P. 176-186.
- [4] Spiess H., Florian M. Optimal strategies: A new assignment model for transit networks. // Transportation Research Part B: Methodological. 1989. № 23(b). P. 83-102.
- [5] Noraini Mohd Razali An efficient genetic algorithm for large scale vehicle routing problem subject to precedence constraints. // Social and Behavioral Sciences. 2015. Issue 195. P. 1922-1931.
- [6] Dorigo M., Maniezzo V., Colomi A. Ant system: optimization by a colony of cooperating agents. // IEEE Transactions on Systems, Man, and Cybernetics. 1996. Part B. 26 (1). P. 29-41.
- [7] Percolation models of information dissemination in social networks. / Lesko, S.A., Zhukov, D.O. // Proceedings 2015 IEEE International Conference on Smart City. / 19 - 21 Dec 2015, Chengdu, China. P. 213-216.