

ЗАДАЧА НАВИГАЦИИ НА ОСНОВЕ ДИНАМИЧЕСКОЙ ИНФОРМАЦИИ О ТРАФИКЕ

Р. Ю. Коба

*Белорусский государственный университет, г. Минск
kobaroman1999@gmail.com
науч. рук. – В. А. Образцов, канд. физ.-мат. наук, доц.*

Правильное распределение своего времени – необходимый навык для каждого из нас. К сожалению, людям приходится отводить время для вещей и действий, которые напрямую с их целями не связаны. Таким действием для большинства людей является передвижение на транспорте. В современных реалиях в крупных городах и мегаполисах наблюдаются большие проблемы с трафиком из-за большого количества участников дорожного движения, вследствие чего возникают пробки или аварии.

Поэтому задача навигации в условиях сложной динамической дорожной обстановки приобретает невероятно высокое значение.

В данной работе была поставлена цель разработать собственный навигатор, который будет учитывать дорожную ситуацию и выдавать результаты, максимально приближенные к реальным значениям.

Ключевые слова: навигатор; кратчайший путь; A*; OSM; трафик.

ВВЕДЕНИЕ

Для разработки навигатора, который имеет возможность на высоком уровне обрабатывать дорожную ситуацию, были выделены следующие задачи:

1. Изучение эффективных алгоритмов нахождения кратчайшего пути для дорожного графа;
2. Выбор максимально полного источника геоданных и дорожно-транспортной обстановки;
3. Учёт различных факторов дорожно-транспортной обстановки при реализации алгоритма кратчайшего пути для дорожного графа.

Рассмотрим подробнее каждый этап.

ЗАДАЧА КРАТЧАЙШЕГО ПУТИ НА ДОРОЖНОМ ГРАФЕ

Для задачи кратчайшего пути, которая в базовом случае может быть решена несколькими известными алгоритмами, например, Дейкстры, существует несколько эвристик, применимых к дорожному графу: A*, двунаправленный поиск, построение иерархии дорог, Reach, Arc-Flags, ALT.

Эвристики Reach, Arc-Flags, ALT, рассмотренные в статьях [1, 2], и эвристика построения дорог позволяют незначительно ускорить работу

алгоритма за счёт предобработки дорожного графа, сохраняя полезную информацию для основного этапа работы алгоритма. С другой стороны, они требуют предварительной обработки и сохранения результатов, что тратит время и память, а значит плохо подходят для решения задач, требующих учёта возможности динамического изменения ситуации в дорожной сети в процессе движения.

Поэтому при построении собственной навигационной системы указанные эвристики не использовались.

A*

Алгоритм A* (A star) является модификацией алгоритма Дейкстры. Отличие заключается в том, что A* использует факт географической природы графа: вводится ограничение на распространение “волны” поиска в направлении от целевой точки путём поощрения движения к цели. [3]

Алгоритм сокращает время получения результата и количество рассматриваемых элементов дорожного графа. Первое преимущество необходимо при работе с большими картами или при частом использовании алгоритма. Второе преимущество особо заметно при использовании API. В этом случае алгоритм обращается к меньшему количеству вершин и дуг, что уменьшает количество обращений через API к сервисам геоданных.

При правильной реализации данный алгоритм значительно лучше алгоритма Дейкстры, и при этом он не имеет существенных недостатков. В разрабатываемой навигационной системе был использован именно данный алгоритм построения кратчайшего пути.

ИСТОЧНИКИ ГЕОДАНЫХ И ДОРОЖНОЙ ОБСТАНОВКИ

Из множества источников геоданных, которые можно было бы использовать при разработке собственного навигатора, был выделен проект OpenStreetMap (OSM). В работе он используется в качестве основного источника информации по следующим причинам:

1. Карты OpenStreetMap - это открытый интернет-проект;
2. OSM - это единственные карты, к внутреннему устройству которых у стороннего пользователя есть полный доступ. Для всех остальных карт можно получить только готовую картинку с нарисованной картой;
3. Основная задача проекта - собрать глобальную базу данных, содержащую различную информацию о каждой точке на поверхности Земли, а не построение коммерческой карты. На основе базы собранных данных любой другой проект и сервис может проводить свою работу и создавать собственные карты.

Однако проект имеет и значительный недостаток: среди данных проекта отсутствует информация о динамическом трафике. Единственная информация касательно загруженности дорог находится в теге `maxspeed:practical`, которая не может являться самостоятельным источником данных о трафике по причине своей статичности.

Получение динамической информации оказалось значительным препятствием в разработке программы, ведь существующие навигационные системы не предоставляют её для использования в частных проектах. Для решения вопроса было решено моделировать динамический трафик, поэтому в дополнение к OSM был рассмотрен портал data.cityofchicago.org, предоставляющий информацию об историческом динамическом трафике на дорогах города Чикаго.

УЧЁТ ДОРОЖНО-ТРАНСПОРТНОЙ СИТУАЦИИ

В проекте были рассмотрены два варианта моделирования дорожного трафика:

1. Использование тега OSM `maxspeed:practical`;
2. Моделирование трафика на основе базы данных исторического трафика Чикаго, предоставляемого на сайте data.cityofchicago.org.

В первом случае программа полагается на информацию OSM по фактической скорости на дороге.

Во втором случае была выделена средняя скорость движения в Чикаго для каждого типа дороги с интервалом времени в один час. Эта средняя скорость была применена к соответствующим типам дорог всей рассматриваемой карты.

Также был рассмотрен вариант, который совмещает два предыдущих.

Учёт светофоров и пешеходных переходов при расчёте времени, затраченного на путь, оказался не лёгкой задачей, ведь настройки каждого светофора индивидуальны, а описать каждое устройство невозможно. В разработанном приложении используется фиксированное количество времени на торможение и разгон автомобиля, а для определения вероятности ожидания на светофоре и переходе используется вероятностное распределение Бернулли (с вероятностью 25% водителю необходимо будет остановиться), а время ожидания равновероятно от 0 до 20 секунд.

ОЦЕНКА КАЧЕСТВА РАЗРАБОТАННОЙ СИСТЕМЫ

Была разработана навигационная система для города Минска, вычисляющая кратчайший путь и оценочное время, затраченное на него, учитывающая дорожные перекрёстки, светофоры, пешеходные переходы, максимально допустимую скорость на дороге и трафик.

При моделировании трафика только при помощи данных OSM сколько-нибудь значащих результатов получено не было. Информация по трафику получилась статичной и независимой от времени, а также присутствовала не для каждого участка дороги.

При моделировании трафика на основе динамических исторических данных data.cityofchicago.org были получены завышенные оценки затраченного времени для пути, что, возможно, объясняется большей загруженностью дорог в Чикаго, чем на рассматриваемом участке – городе Минске.

Наилучший результат был получен при совмещении двух указанных способов. Трафик моделировался исходя из отношения скорости на дорогах указанного типа в определённый промежуток времени к средней дневной скорости для дорог данного типа, однако среднее значение скорости движения для каждого типа дороги изымалось из тега `maxspeed:practical`. Дальнейшие результаты получены для данного метода моделирования трафика.

Программа отображает приближенные к реальности значения, проложенный путь в большинстве случаев близок к предлагаемым системами Яндекс.Карты и OSM, но оценочные значения затраченного на дорогу времени на карте города Минска всё ещё имеют некоторое отклонение от значений, предлагаемых в профессиональных навигаторах. Согласно экспериментам, в среднем отклонение составляет не более 10% по длине маршрута и не более 16% по оценочному времени.

Погрешность работы разработанного навигатора объясняется тем, что при работе использовался бесплатный ресурс OSM, заполнение данных в котором производится энтузиастами, что негативно говорит о качестве данных. Также система моделирования дорожной ситуации имеет возможности для улучшения.

При соответствующей доработке результаты могут быть использованы при разработке реальных навигационных систем.

Библиографические ссылки

1. *R. J. Gutman*. Reach-Based Routing: A New Approach to Shortest Path Algorithms Optimized for Road Networks. In Proceedings of the 6th Workshop on Algorithm Engineering, 2004.
2. *Goldberg, A.V., Werneck, R.F.* Computing Point-to-Point Shortest Paths from External Memory. In: Proceedings of the 7th Workshop on Algorithm Engineering and Experiments (ALENEX 2005), pp. 26–40.
3. *P. E. Hart, N. J. Nilsson and B. Raphael*, A Formal Basis for the Heuristic Determination of Minimum Cost Paths, IEEE Transactions on Systems Science and Cybernetic, Vol. 4, No. 2, 1968, pp. 100-107.