

## АЛГОРИТМЫ ПОИСКА ОПТИМАЛЬНОГО ПУТИ ДЛЯ ЗАДАЧИ ПОСТРОЕНИЯ МАРШРУТА В ТРАНСПОРТНОЙ СЕТИ

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Новаковская А.И.

Сиротко С.И. – канд. физ-мат. наук, доцент

При проектировании и разработке приложений, которые помогают ориентироваться в городе, основной задачей является обеспечение оперативной и достоверной информации, причем объем данных, необходимых для построения оптимального маршрута, может быть достаточно велик. Качество построения маршрута и скорость при обработке картографической информации достигаются применением особых алгоритмов, закладываемых в программу. Именно запрограммированные алгоритмы являются важнейшей частью рассматриваемого приложения.

Рассмотрим способы и примеры хранения картографической информации. Картографическая информация города достаточно объемна и разнообразна, она сочетает в себе: дороги, здания, остановки маршрутно-транспортных средств и т.д. Для хранения информации о дорогах в большинстве своем используются графы. Дорожный граф — это сеть дорог. Он содержит в себе множество фрагментов, которые связаны друг с другом. Каждый из них хранит информацию о своём участке дороги: среднюю скорость, координаты, направление движения, плотность потока в разное время и другие параметры. Также, каждый элемент содержит данные о том, какие связи он имеет с соседними элементами — в какие стороны можно поворачивать, можно ли там развернуться или разрешено ехать только прямо. Подобным образом хранят данные в yandex и google картах.

Еще одной важной частью приложения является алгоритм нахождения оптимального пути. Именно от него и его реализации зависит качество результата и быстрота его получения. Существует много алгоритмов для построения оптимального маршрута, такие как алгоритмы Дейкстры, Беллмана–Форда, Флойда–Уоршелла и другие. Рассмотрим некоторые из них.

1. Алгоритм Дейкстры используется во многих приложениях. Он ищет самые короткие пути от заданной вершины графа до других. Алгоритм работает только, если графы не имеют отрицательного веса. В самом простом варианте, когда при поиске вершины с минимальным весом просматривается всё множество вершин и величины хранятся в массиве - сложность алгоритма составляет  $O(n^2 + m)$ , где  $n$  — вершины,  $m$  — ребра. Если не пройденные вершины хранить в фибоначиевой куче, то сложность уменьшится и составит  $O(n \log n + m)$ .

2. Алгоритм Беллмана–Форда был изобретен Р. Беллманом и Л. Фордом. В отличие от предыдущего, алгоритм работает с ребрами, имеющими отрицательный вес. Другое преимущество алгоритма — его меньшая сложность, которая составляет лишь  $O(n * m)$ , где  $n$  — вершины,  $m$  — ребра.

3. Алгоритм Флойда–Уоршелла был разработан Р. Флойдом и С. Уоршеллом в 1962 году. Он достаточно прост в реализации программы по сравнению с алгоритмами Дейкстры и Беллмана–Форда, но демонстрирует высокую сложность при выполнении —  $O(n^3)$ , где  $n$  — вершины графа.

4. Алгоритм бэктрекинга был изобретен несколькими независимыми исследователями. Цель алгоритма в нахождении не просто одного кратчайшего пути, а всех возможных (альтернативных) вариантов путей от вершины А к вершине Б. Эффективность алгоритма заключается в том, что найденные возможные варианты можно сохранить и затем их можно использовать в дальнейших расчетах в программе. Сложность алгоритма  $O(n * m)$ , где  $n$  — вершины графа.

Исходя из достоинств и недостатков перечисленных алгоритмов, наиболее оптимальным для определения маршрута на общественном транспорте представляется алгоритм бэктрекинга. Данный алгоритм быстрее почти всех известных, кроме разве что алгоритма Флойда–Уоршелла, но выигрывает у него простотой реализации. Кроме того, алгоритм бэктрекинга позволяет сохранять возможные варианты и в дальнейшем использовать их, что позволяет существенно уменьшить время работы.

Список использованных источников:

1. YandexRouteTechnologies [Электронный ресурс]. — Электронные данные. — Режим доступа: <https://yandex.ru/company/technologies/routes/>
2. Алгоритм Дейкстры [Электронный ресурс]. — Электронные данные. — Режим доступа: <https://en.wikipedia.org/wiki/Dijkstra>
3. Алгоритм Беллмана -Форда [Электронный ресурс]. — Электронные данные. — Режим доступа: [https://en.wikipedia.org/wiki/Bellman%E2%80%93Ford\\_algorithm](https://en.wikipedia.org/wiki/Bellman%E2%80%93Ford_algorithm)
4. Алгоритм Флойда -Уоршелла [Электронный ресурс]. — Электронные данные. — Режим доступа: [https://en.wikipedia.org/wiki/Floyd%E2%80%93Warshall\\_algorithm](https://en.wikipedia.org/wiki/Floyd%E2%80%93Warshall_algorithm)
5. Алгоритм бэктрекинга [Электронный ресурс]. — Электронные данные. — Режим доступа: <https://en.wikipedia.org/wiki/Backtracking>