

УДК 656.1.022:004.421

Максимова Елена Ивановна, студент кафедры вычислительной техники Института кибернетики ТПУ.

E-mail:

YelenaMaksimova@yandex.ru

Область научных интересов: исследование задач и алгоритмов теории графов, алгоритмы обработки изображений.

СРАВНЕНИЕ КАЧЕСТВА РЕЗУЛЬТАТОВ АЛГОРИТМА «A STAR» И ЕГО МОДИФИКАЦИИ ДЛЯ ДОРОЖНОЙ СЕТИ ПРИ ВЫБОРЕ МАРШРУТА С УЧЕТОМ НАПРАВЛЕНИЯ ДВИЖЕНИЯ НА ПЕРЕКРЕСТКЕ

Е.И. Максимова

Томский политехнический университет

E-mail: YelenaMaksimova@yandex.ru

Задача нахождения оптимального маршрута является актуальной для реализации систем автомобильной навигации. Одним из недостатков общеизвестного алгоритма «A star», представляющего собой основу подобных систем, является невозможность учитывать направление движения на перекрестке. Предложена модификация алгоритма «A star», способная учитывать направление движения на перекрестке. Для тестирования алгоритмов были реализованы генератор и визуализатор карт дорожной сети. В результате сравнения данных алгоритмов было установлено, что предложенная модификация алгоритма «A star» находит минимальный путь по времени прохождения, а не по его длине, в отличие от оригинального алгоритма «A star».

Ключевые слова:

Алгоритм, «A star», граф, маршрут, направление, перекресток, модификация, тестирование, Дейкстра, сравнение.

Введение

Все чаще в современном мире используются системы поиска оптимального маршрута для дорожных сетей. Такие системы нашли широкое применение среди водителей транспортных средств и применяются для нахождения маршрута до некоторой конечной точки, требующего минимального количества времени на его преодоление. Большинство алгоритмов нахождения оптимального пути не являются общедоступными. Однако существуют и общеизвестные алгоритмы, на которых основывается большинство подобных систем. Одним из них является алгоритм «A star».

Алгоритм «A star»

Алгоритм «A star» используется для поиска маршрута наименьшей стоимости от начальной вершины до выбранной конечной во взвешенном графе.

Для вычисления оптимального пути «A star» использует два параметра рассчитываемых для каждого узла: наименьший путь $G(v)$ от заданной вершины до вершины v и эвристическую оценку расстояния $H(v)$ от текущей вершины до конечной. Порядок обхода вершин в графе определяется итоговой функцией $F(v) = G(v) + H(v)$, поведение которой зависит от выбранной эвристической оценки $H(v)$. Для планарной дорожной сети в качестве $H(v)$ выбирается евклидово расстояние.

Стоит отметить, что при эвристической оценке $H(v) = 0$ данный алгоритм полностью совпадает с алгоритмом Дейкстры [1].

В реальной дорожной сети в зависимости от направления движения на перекрестке может потребоваться различное количество времени на преодоление этого перекрестка. Одним из недостатков данного алгоритма в таком случае является невозможность учитывать направление движения на перекрестке.

Модификация алгоритма «A star» для учета направления движения на перекрестке

В ходе работы алгоритма «A star» учитываются лишь наименьшие расстояния до каждого из перекрестков. В данном алгоритме не учитывается, по какой дороге достигнут текущий перекресток [2]. Казалось бы, возможна простая модификация алгоритма, при которой запоминалось бы не только минимальное время пути до каждого из перекрестков, но и дорога, по которой был проложен этот маршрут. Однако подобная модификация не является корректной. На практике возможна ситуация, при которой можно достичь некоторого перекрестка за минимальное время в одном из направлений, но затем потребуется осуществить поворот налево, что является более затратным по времени, нежели движение прямо или поворот направо. Подобная ситуация изображена на рис. 1. Перекресток В можно проехать в различных направлениях. Если двигаться по указанному маршруту, то для достижения перекрестка А потребуется меньше времени. Однако для дальнейшего движения потребуется повернуть налево.

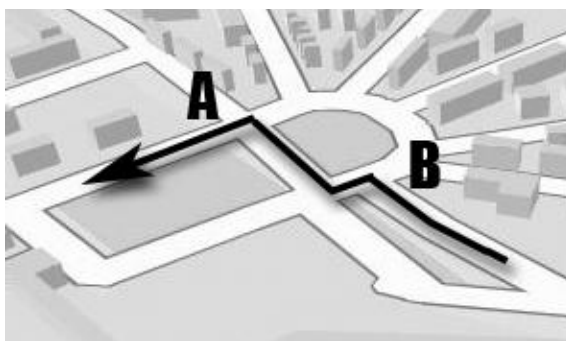


Рис. 1. Пример маршрута с левым поворотом на перекрестке А

Добраться до перекрестка А можно, используя другой путь. При движении в таком направлении может потребоваться больше времени, но для продолжения пути не придется осуществлять поворот налево. Такая ситуация представлена на рис. 2. Если преодолеть перекресток В в указанном направлении, то на перекрестке А не потребуется осуществлять поворот, следовательно, на проезд данного перекрестка уйдет значительно меньше времени.

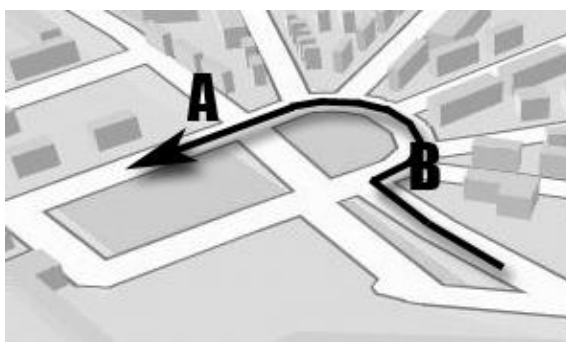


Рис. 2. Пример маршрута без левого поворота на перекрестке А

Как можно заметить, в такой ситуации недостаточно учитывать только наименьшее время, за которое можно достичь того или иного перекрестка. Важно также учитывать время, затрачиваемое на достижения перекрестка в каждом из возможных направлений. Для этого следует несколько модифицировать граф, которым задается дорожная сеть. Каждая вершина в таком графе теперь должна задавать не только некоторый перекресток, но и направление движения, по которому он достигается.

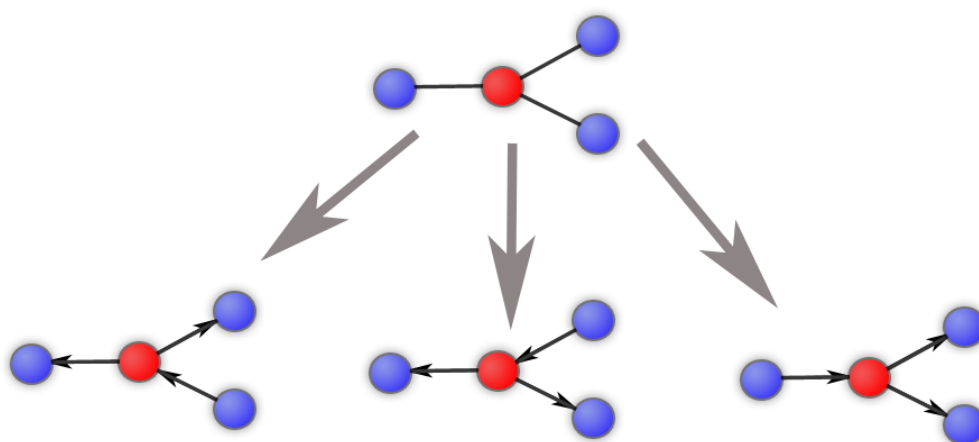


Рис. 3. Пример модификации вершины в графе

То есть каждая вершина графа будет представлять собой пару {идентификатор перекрестка; направление движения до этого перекрестка}. При такой модификации графа количество вершин, как и количество ребер, будет существенно увеличено. Это можно продемонстрировать на примере простого планарного графа (рис. 4)

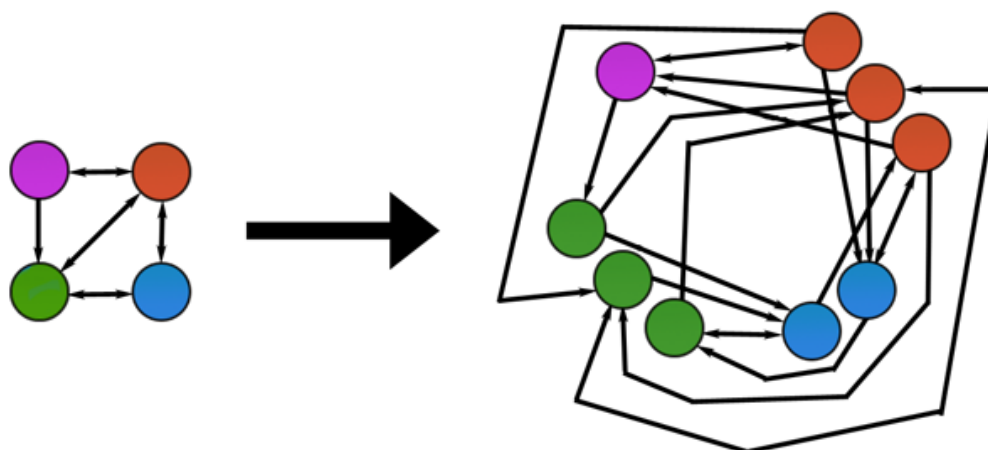


Рис. 4. Визуализация процесса модификации планарного графа с целью учета направления движения до перекрестка

При использовании графа с такой структурой кратчайшее расстояние будет рассчитываться не просто до всех перекрестков, а для каждого направления подъезда к каждому перекрестку. Иллюстрация подобной модификации графа представлена на рис. 4. В ходе работы описанного алгоритма интересные нас перекрестки рассматриваются столько раз, сколько дорог инцидентно данному перекрестку, что и приводит к модификации графа. Так, например, на рис. 4 левому нижнему и правому верхнему перекресткам инцидентно по три дороги, соответственно, каждый из них рассматривается три раза с различными вариантами направления движения до них.

Данная модификация не приведет к потере возможности нахождения оптимального маршрута, ведь теперь каждому возможному положению транспортного средства (с учетом текущего направления движения) на дорожной сети в графе соответствует единственная вершина. До любой вершины, а соответственно, и любого положения будет найден кратчайший путь. Следовательно, такой подход позволяет найти оптимальный маршрут с учетом направления движения на перекрестках.

Очевидно, что для перекрестка существует столько различных направлений движения до него, сколько участков дороги ему инцидентно. Каждому участку дороги инцидентно ровно два перекрестка, следовательно, в новом графе будет всего $2 \cdot M$ вершин, где M – количество

участков дороги между двумя перекрестками в исходной дорожной сети. Количество ребер в полученном графе можно оценить, как $P_a \cdot M$, где P_a – среднее количество участков дорог, инцидентных перекрестку в исходной дорожной сети. Для реальных дорожных сетей значение P_a находится в интервале от 3 до 4, что говорит о том, что количество ребер в новом графе будет больше, чем в исходном, не более чем в 4 раза.

Более того, для реализации алгоритма «A star» на таком графе не обязательно заранее задавать структуру графа в явном виде. Достаточно рассматривать только те пары {идентификатор перекрестка; направление движения до этого перекрестка}, которые будут посещены в ходе работы данного алгоритма. Таким образом, количество вершин в полученном графе для реальной дорожной сети будет существенно меньше, чем $2 \cdot M$. Поэтому количество ребер, которые будут рассмотрены в ходе работы алгоритма, также будет намного меньше приведенной ранее оценки.

Очевидно, что для реализации данного алгоритма необходим способ определения типа маневра (движение прямо, поворот налево или поворот направо) для каждого возможного направления движения. Для этого достаточно учесть координаты трех перекрестков: текущего перекрестка C_1 , предыдущего перекрестка C_2 и перекрестка C_3 , в направлении которого выполняется дальнейшее движение. Рассмотрим два вектора (C_2, C_1) и (C_1, C_3) , анализируя их взаимное расположение можно определить тип маневра. В случае, когда перекресток имеет четыре инцидентных ему участка дороги, среди всех возможных направлений дальнейшего движения можно выбрать то, в котором векторное произведение данных векторов является наименьшим. Такое движение можно считать поворотом налево. Аналогично, если выбрать направление, для которого векторное произведение имеет наибольшее значение, то такой маневр является поворотом направо. Оставшееся направление соответствует движению прямо. В случае, если перекрестку инцидентно три участка дороги, требуется определить направление, для которого угол отклонения первого вектора от второго наименьший. Такое направление соответствует движению прямо. Оставшийся тип маневра определяется в зависимости от векторного произведения вектора для движения прямо и вектора этого направления. Можно предположить, что если оба угла отклонения больше некоторого порогового значения, то ни один из маневров не является движением прямо.

Разработка средств для тестирования и визуализации работы алгоритмов «A star» и его модификации

Для тестирования и визуализации работы алгоритмов «A star» и его модификации на языке C++ были реализованы генератор карт дорожной сети и визуализатор, который отображает найденный кратчайший маршрут между двумя выбранными пользователем перекрестками, а также его длину и время прохождения.

Генератор карт дорожной сети представляет собой консольное приложение. Значения входных данных передаются через параметры вызова из командной строки. Входными данными являются значения количества перекрестков и плотности дорожной сети. Плотность дорожной сети равняется среднему количеству дорог, инцидентных перекрестку, и находится в диапазоне от 2,5 до 3,5. Сгенерированная карта сохраняется в указанную директорию в виде файла формата .xml.

Визуализатор реализован в виде оконного приложения. Данное приложение имеет возможность загрузки карты дорог из xml-файла с описанием дорожной сети, а также отображения кратчайшего маршрута и его длины между двумя перекрестками (рис. 3).

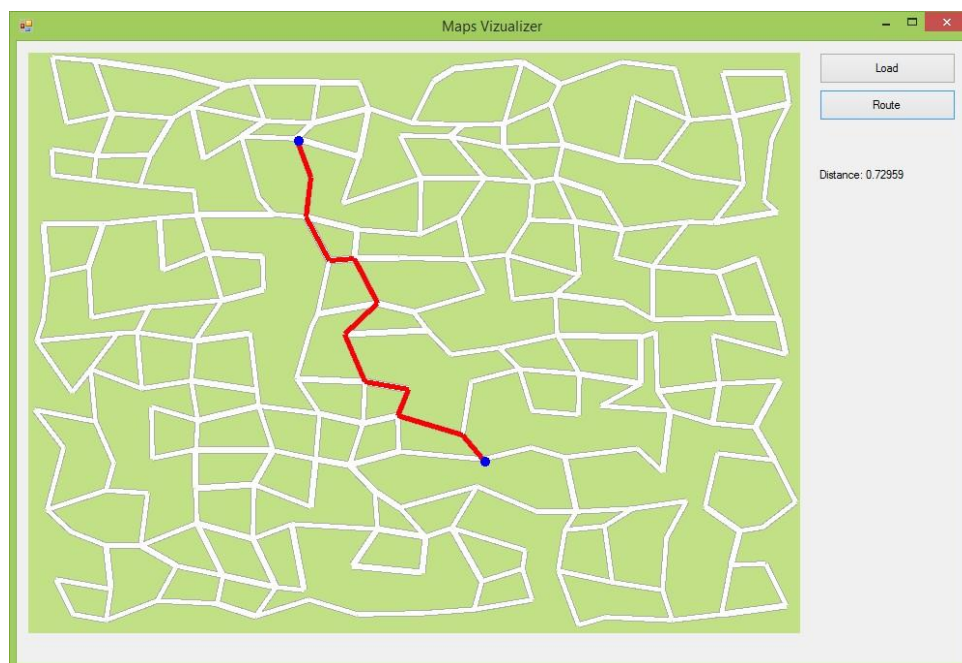


Рис. 5. Визуализация оптимального маршрута на дорожной сети

Результаты сравнения алгоритма «A star» и его модификации для дорожной сети при выборе маршрута с учетом направления движения на перекрестке

Для представления результатов эксперимента была введена фиктивная единица длины, которую автомобиль, двигаясь со среднестатистической скоростью, преодолевает за один час. Время прохождения маршрута представлено в часах.

Таблица 1. Результаты экспериментов

Алгоритм «A star»		Модифицированный алгоритм «A star»	
Длина маршрута	Время прохождения маршрута, ч	Длина маршрута	Время прохождения маршрута, ч
0,49305	0,73305	0,49305	0,73305
0,65371	1,34371	0,68378	1,19378
1,03522	2,11522	1,07138	1,79138
1,33209	2,50209	1,56848	2,34848
1,38463	3,03463	1,55647	2,48647
1,53833	2,40833	1,59250	2,31250

Как можно заметить в табл. 1, при поиске оптимального маршрута между двумя перекрестками длина маршрута, найденного алгоритмом «A star», оказывается меньше длины маршрута, найденного модифицированным алгоритмом «A star». Однако стоит отметить, что время, затраченное на прохождение маршрута в первом случае, существенно превышает время, требующееся для преодоления маршрута, найденного модифицированным алгоритмом «A star». При этом чем больше расстояние между перекрестками, тем больше разница во времени прохождения маршрута между рассмотренными алгоритмами, так как количество маршрутов, а, следовательно, и различных маневров на перекрестках возрастает при увеличении расстояния между начальными точками.

Подобный результат можно объяснить тем, что для реальных дорожных сетей время преодоления какого-либо маршрута зависит не только от его длины, но и от количества и типов маневров, совершаемых при движении. Как уже было отмечено, такой маневр, как поворот налево, занимает несколько больше времени, чем оставшиеся маневры.

Оценка вычислительной сложности алгоритма

Вычислительная сложность алгоритма «A star» совпадает с вычислительной сложностью алгоритма Дейкстры, оценка которой $O(E \cdot \log E)$, где E – количество ребер в графе [3]. Для достижения такой асимптотической оценки в реализации необходимо использовать бинарную кучу (очередь с приоритетами). Ранее было отмечено, что количество ребер в новом графе равняется $P_a \cdot M$, что в P_a раз больше, чем в исходном графе. Стоит отметить, что количество ребер умножается на некоторую константу P_a , поэтому итоговая асимптотическая оценка не изменяется. Следовательно, для дорожной сети предложенная модификация алгоритма «A star» также имеет вычислительную сложность $O(M \cdot \log M)$, как и аналогичный алгоритм без модификации.

Заключение

В результате проделанной работы была предложена модификация алгоритма «A star» для дорожных сетей, позволяющая учитывать направление движения на перекрестке. Было установлено, что предложенная модификация обладает той же вычислительной сложностью, что и оригинальный алгоритм, и в условиях реальных дорожных сетей находит маршрут, на прохождение которого требуется наименьшее время.

Стоит отметить, что с увеличением расстояния между начальными перекрестками увеличивается и разница во времени прохождения маршрута, найденного алгоритмом «A star» и его модификацией, хотя в общем случае длина маршрута, определенного оригинальным алгоритмом, меньше.

СПИСОК ЛИТЕРАТУРЫ

1. Кристофидес Н. Теория графов: алгоритмический подход. – 2-е изд., испр. – М.: Мир, 1978. – 430 с.
2. Алгоритмы: построение и анализ / Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн. – М.: Вильямс, 2011. – 1293 с.
3. Engineering route planning algorithms / D. Delling, P. Sanders, D. Schultes, D. Wagner // Algorithmics of large and complex networks. – Springer, 2009. – 376 p.

Поступила 06.10.2014