DOI: 10.17117/na.2016.01.01.477

Поступила (Received): 25.01.2016

<http://ucom.ru/doc/na.2016.01.01.477.pdf>**Попов Д.А.****Поиск кратчайших маршрутов на графах дорожной сети.
Определение методов ускорения поиска****Popov D.A.****Shortest path graph of the road network. Determination of
methods to accelerate the search**

В статье рассматриваются вопросы поиска оптимальных кратчайших путей в уличных дорожных сетях. Рассматриваются вопросы использования классических алгоритмов Дейкстры, Флойда и Беллмана-Форда для поиска путей на взвешенных графах. Рассмотрены пути оптимизации алгоритмов для ускорения их работы. Приводится сравнительный анализ алгоритмов

Ключевые слова: алгоритм, путь, оптимизация, Дейкстры

The article deals with the search for optimal shortest paths in street road networks. The issues of the use of classical Dijkstra, Floyd and Bellman-Ford to find ways on weighted graphs. The ways of optimization algorithms to speed up their work. The comparative analysis of algorithms

Key words: path, optimization, algorithm, Dijkstra

Попов Дмитрий Александрович

Инженер-программист

Открытый код

г. Самара ул. Ярмарочная, 55

Popov Dmitriy Aleksandrovich

Software Engineer

Open code

Samara, Yarmarochnaya st., 55

В настоящее время одним из актуальных направлений в алгоритмике является вычисление оптимального кратчайшего маршрута в уличных дорожных сетях.

Можно использовать алгоритм Дейкстры – классическое решение задач в теории графов. Но для больших дорожных уличных сетей работа этого алгоритма слишком медленна. Уличные дорожные сети весьма обширны и охватывают тысячи дорожных развязок.

Таким образом, существует значительный теоретический и практический интерес к оптимизации алгоритмов поиска путей на графах и разработке методов нацеленных на ускорение вычислений. Также необходимы новые методы, которые сократят время процессов первичной обработки и генерации вспомогательных данных используемых для последующих запросов в режиме реального времени на планирование и поиск кратчайших маршрутов. Программные комплексы, выполняющие логистическое планирование, симуляцию и прогнозирование загруженности дорожного трафика с последующим формированием

управленческих решений, нуждаются в разрешении проблемы обработки огромного количества таких запросов.

Вычисление оптимальных, кратчайших маршрутов и логистическое планирование, чаще всего опирается на аппроксимационные алгоритмы, что приводит к снижению точности получаемых результатов. С другой стороны, использование агрессивных эвристических алгоритмов также приводит к погрешностям в результатах. Применение же переборных алгоритмов без оптимизации для решения задач поиска кратчайших маршрутов, ведёт к увеличению времени вычислений. Причиной тому, упрощённый подход к выбору алгоритмов и программной реализации, что приводит к непозволительным временным задержкам в обработке запросов. Если программные продукты рассчитывающие кратчайшие маршруты, реализуют такой подход, то это вызывает оправданное недовольство пользователей, вынужденных ждать ответа из-за долгих расчётов. Такой подход к алгоритмам также становится экономически неоправданным, из-за того, что расчёты требуют больших вычислительных мощностей. Разработчикам программных продуктов сложно балансировать между скоростью вычислений и субоптимальностью вычисленных маршрутов. По этим причинам, существует заинтересованность в разработке более эффективных и точных методов планирования и поиска кратчайших маршрутов в дорожной сети.

Дорожная сеть легко представляется в виде графа $G(V,E)$, то есть в виде набора узлов V (переходы, перекрёстки) и рёбер E (дорожные сегменты), где каждое ребро соединяет два узла. Каждому ребру назначен вес, например, длина пути или оценка времени, необходимого для преодоления пути по дороге между смежными узлами. В теории графов, поиск кратчайших путей между двумя узлами является классической задачей. Различают несколько вариантов этой задачи:

- один ко многим (point-to-many): для данного исходного узла s множества V , вычислить длину кратчайшего пути до всех узлов v множества V ;
- один к одному (point-to-point): вычислить длину кратчайшего пути от исходного узла s множества V к целевому узлу t множества V ;
- многие ко многим: для данного узла устанавливаются S, T множества V , вычислить длину кратчайшего пути для каждой пары узлов (s, t) в множестве S, T ;
- все пары: специальный случай многие ко многим варианте с $S:=T:=V$.

Стандартным решением задачи поиска кратчайшего пути графа $G(V,E)$ в варианте один ко многим – это алгоритм Дейкстры. Данный алгоритм поддерживает неубывающую очередь с приоритетами Q и инвариант определяющий равенство $Q=V-S$. Атрибуты $u.d$ для u в множестве V имеют значение равное оценочному расстоянию от исходной вершины. Алгоритм инициализирует пустое множество вершин S , $u.d$ вершин u в множестве V присваивается значением ∞ , за исключением $d(s,s) = 0$ вершины s . Исходная вершина s добавляется в S . В каждой итерации, извлекается вершина u из множества Q с наименьшей оценкой расстояния и добавляется в S , тем самым соблюдается инвариантность. Производится релаксация всех рёбер $a=(u,v)$ из множества E инцидентных к u . Для

каждого такого ребра, определяется расстояние до v через ребро a путем вычисления $d(s,u) + l(a)$. Если это значение меньше расстояния $d(s,v)$, то $v.d = d(s,u) + l(a)$. После релаксации ребер значение расстояния $d(s,u)$ считается верным. Согласно данному описанию алгоритма, каждая вершина извлекается из Q и добавляется в S только по одному разу. Соответственно количество итераций составляет $|V|$.

Алгоритм Дейкстры работоспособен для поиска кратчайшего пути вида один к одному. Работу алгоритма можно остановить, как только он отсканирует целевую вершину t . Множество вершин S определяется алгоритмом как область поиска.

Основное внимание при оптимизации алгоритма Дейкстры концентрируется на том, как минимизировать или избежать накладных расходов на поиск вершины с наименьшим расстоянием в очереди. Оригинальная версия алгоритма Дейкстры работает в $O(n^2)$. Эта оценка может быть улучшена, время работы $O((V+A) \log V)$ с бинарными кучами и $O(A+V \log V)$ с кучей Фибоначчи.

Другой практический подход заключается в сужении поискового пространства, путём использования двунаправленного поиска, который одновременно работает как поиск вперед от начальной вершины s и обратный поиск от целевой вершины t . Алгоритм останавливается, как только их пересечение областей поиска доказуемо содержит вершину x кратчайшего пути из s в t . Для дорожной сети, двунаправленный поиск посещает примерно вдвое меньше вершин, в отличие от однонаправленного подхода.

Минусом алгоритма Дейкстры является невозможность обработки графов, с ребрами отрицательного веса, т. е. если, например, некоторая система предусматривает убыточные маршруты, то для работы системы следует воспользоваться алгоритмом отличным от алгоритма Дейкстры.

Альтернативой алгоритму Дейкстры является алгоритмом вычисления кратчайшего пути Беллмана-Форда. Алгоритм похож на алгоритм Дейкстры, но его преимущество в сложности. Оно составляет $O(VE)$, что делает алгоритм Беллмана-Форда конкурентоспособным с алгоритмом Дейкстры. Кроме того, как преимущество, алгоритм Беллмана-Форда работает на графах с отрицательными весами ребер.

Так же альтернативой алгоритму Дейкстры рассматривается алгоритм Флойда-Уоршала. Алгоритм вычисляет длины кратчайших путей между всеми парами вершин в графе. Главное преимущество алгоритма в простоте реализации, но недостатком является его кубическую сложность $O(V^3)$.

Экспериментальное измерение времени исполнения – самый простой и надёжный путь подобрать алгоритм для конкретного приложения. Если количество дуг в графе меньше, чем n^2 , тогда, рекомендуется применять версию алгоритма Дейкстры со списками смежности. В этом случае время решения общей задачи нахождения кратчайших путей имеет порядок

$O(nm \log n)$, что значительно лучше алгоритма Флойда, по крайней мере, для больших разреженных графов. А так, как транспортная сеть это скорее разреженный граф то из представленных алгоритмов эффективнее будет применять алгоритм Дейкстры.

Список используемых источников:

1. Томас Х.К. Алгоритмы. Вводный курс. М.: Вильямс, 2014. С. 96-117.
2. Томас Х.К., Чарльз И.Л., Рональд Л.Р., Клиффорд Ш. Алгоритмы: построение и анализ. М.: Вильямс, 2006. 296 С.
3. Гасникова А.В. Введение в математическое моделирование транспортных потоков. М.: МФТИ, 2010. 361 с.

© 2016, Попов Д.А.

Поиск кратчайших маршрутов на графах
дорожной сети. Определение методов ускорения
поиска

© 2016, Popov D.A.

Shortest path graph of the road network.
Determination of methods to accelerate the search