АЛГОРИТМ ДЕЙКСТРЫ ПРИ ПРОЕКТИРОВАНИИ АВТОМАТИЗИРОВАННЫХ СИСТЕМ

Акулич В. Н., Шилин Л. Ю., Хмыз Д. Д.
Кафедра информационных технологий автоматизированных систем,
Белорусский государственный университет информатики и радиоэлектороники
Минск, Республика Беларусь
E-mail: uladzislau.akulich@gmail.com, dekfitu@bsuir.by

Разработана методика применения алгоритма Дейкстры для оптимизации автоматизированных систем. Разработана автоматизированная система с использованием алгоритма Дейкстры для нахождения кратчайшего пути и визуализации его для пользователей.

Введение

Применение алгоритмов теории графов при проектировании автоматизированных систем позволяет реализовывать новые функции и возможности систем. Использование систем, использующих алгоритмы теории графов, является значительным шагом вперёд для многих организаций и пользователей. Алгоритм Дейкстры является важным инструментом для оптимизации и планирования в различных областях, включая разработку автоматизированных систем.

Применение алгоритма Дейкстры

Алгоритм Дейкстры – это метод, который находит кратчайший путь от одной вершины графа к другой. Граф – структура из точек-вершин, соединенных ребрами. Алгоритм Дейкстры работает для графов, у которых нет ребер с отрицательным весом, то есть таких, при прохождении через которые длина пути уменьшается.

В отличие от похожих методов, алгоритм Дейкстры ищет оптимальный маршрут от одной заданной вершины ко всем остальным, при этом высчитывая длину пути — суммарный вес ребер, по которым проходит при этом маршруте.

Применение алгоритма Дейкстры при проектировании автоматизированных систем является эффективным, так как алгоритм может быть использован для решения следующих задач:

- автоматическое построение маршрута на карте;
- моделирование движения робота;
- маршрутизация движения данных в компьютерной сети;
- обработка транспортных потоков.

Рассмотрим алгоритм Дейкстры для построения кратчайшего пути и визуализации его на карте.

Задача о кратчайшем пути – задача поиска самого короткого пути между двумя вершинами на графе, в котором минимизируется сумма весов рёбер, составляющих путь. В качестве вершин на карте выступают перекрестки, здания и другие объекты, а дороги, которые лежат между ними, являются рёбрами. Если сумма длин дорог между перекрёстками минимальна, тогда найденный путь самый короткий.

Задан ориентированный граф G(V,E), каждой дуге (u,v) ставится в соответствие число L(u,v) – длина дуги (расстояния, стоимости). В общем случае возможно L > 0, L < 0, L = 0.

Под длиной пути понимаем сумму длин дуг, составляющих путь.

Главная задача: найти длины кратчайших путей и сами пути от фиксированной вершины s до всех остальных вершин графа v_i.

Алгоритм решает задачу в случае L>=0. Алгоритм основан на приписывании вершинам v_i временных меток $d(v_i)$. Метка вершины даёт верхнюю границу длины пути от s к этой вершине. Величины меток постепенно уменьшаются, и на каждом шаге итерации одна из временных меток становится постоянной. Это означает, что метка даёт точную длину кратчайшего пути от s к рассматриваемой вершине.

Согласно алгоритму Дейкстры процесс построения дерева начинается с заданной вершины. Положить d(s)=0, считать метку постоянной. $d(v_i) = \infty, i = 1...n$, считать метки временными. p=s.

В дальнейшем на каждом шаге к дереву присоединяется одно новое ребро (и одна вершина). Это ребро выбирается из подходящих рёбер, причем подходящим считается ребро, соединяющее вершину дерева с вершиной, ему не принадлежащей. Среди подходящих ребер выбирается ребро наименьшего веса.

Повторяется n раз, пока не будут упорядочены все вершины. Временная метка $d(v_i)$ всякой неупорядоченной вершины v_i, в которую входит дуга, выходящая из p, считается по формуле:

$$d(v_i) = \min(d(v_i), d(p) + l(pv_i))$$

Выбирается вершина с $min(d(v_i))$ Если их несколько, то любую. Пусть это w. Метку d(w) считать постоянной. p=w.

Доказательство корректности самого алгоритма основывается на следующем утверждении. После того, как какая-либо вершина v становится помеченной, текущее расстояние до неё d_v уже является кратчайшим, и, соответственно, больше меняться не будет.

Реализация алгоритма Дейкстры в автоматизированной системе, в которой автоматически вычисляется и отображается кратчайший путь на карте, состоит из следующих этапов:

1. Инициализация – все вершины графа обозначаются как недостижимые, а расстояние от начальной вершины до всех остальных вершин устанавливается как бесконечность, кроме начальной вершины, для которой устанавливается расстояние 0.
2. Создание пустого множества, которое будет содержать посещенные вершины.
3. Выбор начальной вершины, из которой будет начинаться процесс поиска кратчайших путей.
4. Обновление расстоянийобновление расстояний до соседних вершин - для каждой соседней вершины, которая еще не была посещена, вычисляется новое расстояние, если оно оказывается меньше, чем текущее расстояние до этой вершины, расстояние обновляется.
5. Пометка текущей вершины как посещенной.
6. Поиск следующей вершины для посещения - среди всех непосещенных точек на основе их расстояний от начальной вершины выбирается вершина с наименьшим расстоянием.
7. Повторение этапов с выбором вершин и расчетом расстояния, пока все вершины не будут посещены или пока не будет найден кратчайший путь до заданной конечной вершины.
8. Формирование кратчайшего пути.

Рассмотрим пример задачи для нахождения кратчайшего пути в ориентированном графе. Ориентированный граф представлен на рисунке 1.

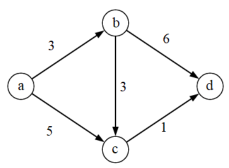

Рис. 1 – Ориентированный граф

В условиях решения задач автоматизированных систем с использованием онлайн-карт рёбра ab, bc, ac, bd, db будут выступать в качестве дорог, а вершины a, b, c, d – в качестве зданий или других объектов.

Первый шаг алгоритма определит, что кратчайший путь от вершины a будет проходить в вершину b и зафиксирует кратчайший путь.

$$l_{ab} = 3;$$

$$l_{ac} = 5;$$

Второй шаг рассмотрит все возможные варианты $A[v] + l_{vw}$. Получим значения:

$$l_{ab} + l_{bc} = 3 + 3 = 6;$$

$$l_{ab} + l_{bd} = 3 + 6 = 9;$$

Оптимальный шаг двигаться по ребру ac, так как $l_{ac} < l_{ab} + l_{bc}$ и $l_{ac} < l_{ab} + l_{bd}$.

Добавляется длина кратчайшего пути от вершины a до вершины c.

На третьем шаге находим $l_{ac} + l_{cd}$.

$$l_{ac} + l_{cd} = 3 + 1 = 4;$$

Таким образом, благодаря алгоритму Дейкстры длина всех путей будет просчитана и будет выбран кратчайший путь до любой из вершин.

Одним из главных преимуществ алгоритма Дейкстры является его значительно низкая сложность, которая является почти линейной, поэтому его внедрение оптимизирует систему.

I. Заключение

Алгоритма Дейкстры в автоматизированных системах применяется для решения разных задач, связанных с построением кратчайшего пути, что позволяет определить оптимальные маршруты с учетом самых различных факторов. Использование алгоритма Дейкстры является эффективным в транспортных и геоинформационные системах, сетях связи, играх с искусственным интеллектом, а также в других сферах применения автоматизированных систем.

Список литературы

1. Рафгарден, Т. Совершенный алгоритм. Графовые алгоритмы и структуры данных. / Т. Рафгарден. – СПб. : Питер, 2019. – 256 с.
2. Карпов, Д. В. Связность графов : / Д. В. Карпов. – СПб.: Санкт-Петербургское отделение Мат. инст. им. В. А. Стеклова РАН, 2018. — 183 с.