

Санкт-Петербургский политехнический университет Петра Великого
Институт машиностроения, материалов и транспорта

Курсовая работа
по теме «А* алгоритм»
по дисциплине «Объектно-ориентированное программирование»

Студент группы 3331506/90401

Кильдибекова А. А.

Преподаватель

Ананьевский М. С.

«_____»_____2022 г.

Санкт-Петербург
2022 г.

Содержание

Введение	3
Описание алгоритма	5
Исследование алгоритма	9
Заключение	10
Список литературы	11
Приложение	12

Введение

A* алгоритм (A-star алгоритм) – это один из наиболее широко используемых алгоритмов поиска кратчайших путей на евклидовом графе. Его частным случаем является алгоритм Дейкстры.

A* получил широкое распространение во многих программных системах, от машинного обучения и поисковой оптимизации до разработки игр. К конкретным примерам можно отнести маршрутизацию телефонного трафика, навигацию по лабиринту, компоновку печатных плат, механическое доказательство теорем и решение задач. Основная область применения алгоритма – нахождение кратчайшего пути на графе, который является моделью карты.

Ограничением по применению A* является необходимость наличия евклидового графа, то есть такого графа, вершины которого являются точками на плоскости и, соответственно, обладают координатами.

A* принято считать самостоятельным алгоритмом, но на самом деле – это семейство алгоритмов, так как он вычисляет маршрут для заданной эвристики. Выбор конкретной эвристической функции определяет конкретный алгоритм из семейства.

Что нужно сделать для зачета по курсовому проектированию:

1. Выбрать один из предложенных алгоритмов и реализовать его;
2. Написать отчет по реализации

Структура отчета:

1. Титульный лист (по форме)
2. Введение -- зачем алгоритм нужен, область применения
3. Описание алгоритма (слова, псевдокод)
4. Исследование алгоритма -- построить графики зависимости скорости выполнения алгоритма от размера входных данных (входные данные могут быть разными -- случайные, наихудший случай, наилучший случай и т.п.)
5. Заключение (рекомендации по применению этого алгоритма)

6. Список литературы (ссылки на литературу, где алгоритм описан)
7. Приложение -- программный код.

Описание алгоритма

A* объединяет два способа поиска пути - математический подход и эвристический подход. Математический подход обычно имеет дело со свойствами абстрактных графов и с алгоритмами, которые предписывают упорядоченное изучение узлов графа для определения пути с минимальными затратами. В рамках этого подхода определяется минимальное расстояние (стоимость перемещения) от исходного (стартового) узла к текущему (рассматриваемому узлу). Принято обозначать его, как g .

Эвристический подход обычно использует специальные знания о предметной области задачи, представленной графом, для повышения вычислительной эффективности решений конкретных задач поиска по графу. В рамках этого подхода используется некая эвристическая функция (её принято обозначать его, как h), которая является оценкой стоимости оптимального пути от текущего узла до предпочтительного целевого узла. Многие задачи, которые могут быть представлены как задача нахождения пути с минимальной стоимостью на графе, содержат некоторую информацию, которая может быть использована для формирования эвристической функции. Выбор $h = 0$ соответствует случаю отсутствия или незнания этой информации. С такой эвристической функцией A* будет представлять собой алгоритм Дейкстры. Поскольку мы, на самом деле, обладаем большей информацией, а именно знаем, как вычисляется расстояние в евклидовом пространстве через координаты двух точек, можно рассчитывать h , как $\sqrt{x^2 + y^2}$ (где x и y - величины различий в координатах x , y вершин). Введение эвристической функции помогает значительно уменьшить количество рассматриваемых узлов.

Два этих подхода объединяются введением функции оценки стоимости оптимального пути f , которая вычисляется как $f = g + h$.

Работа алгоритма:



Перед началом работы создаются два списка open (граница) – список узлов, которые находятся в очереди на посещение и closed – список посещённых узлов

1. Заносят начальный узел в границу (список open)
2. Для него вычисляется f
3. Определяется узел с наименьшим значением f и назначается текущим
4. Проверяют, не является ли текущий узел целевым
5. Текущий узел убирают из границы и заносят его в посещённые узлы (список closed)
6. Для каждого из узлов в границе вычисляется f
7. Для текущего узла находят соседей и заносят их в границу (список open)
8. Повторяют пункты 3-8

Псевдокод:

```
function A* {
    var closed = empty();
    var open = list (start);
    var from = map (null);
    g[start] = 0;
    f[start] = g[start] + h(start, end);
    while (open) {
        curr = min_f(open);
        if (curr == end) {return success;}
        remove(curr, open);
        add(curr, closed);
        for each neighbour in unclosed_neighbour(curr) {
            temp_g = g[curr] + dist(curr, neighbour);
            if (neighbour not in open or temp_g < g[neighbour]) {
                from[neighbour] = curr;
                g[neighbour] = temp_g;
                f[neighbour] = g[neighbour] + h(neighbour, end);
                if (neighbour not in open) {add(neighbour, open);}
            }
        }
    }
    return failure;
}
```

Рассмотрим пример:

Задача: построить кратчайший путь от узла  с координатами (0,0) к узлу  с координатами (4,5), стоимость перемещения между клетками – 1. Граф задан в виде сетки.

На рисунке 1 изображено начало нахождения пути. Текущий узел является начальным. Для него вычисляется оценка общего расстояния $f = g + h = h = (4 - 0) + (5 - 0) = 9$. Затем находятся его соседи (они заносятся в границу), для каждого из которых тоже вычисляется f . Определяется узел с минимальным значением f , он становится текущим.

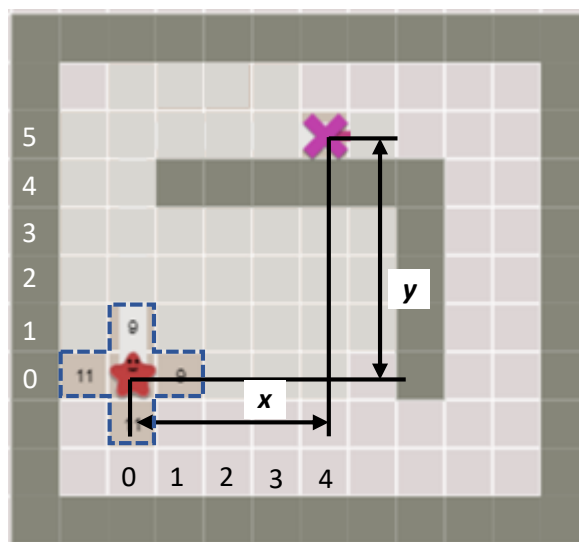


Рисунок 1 – Пример

Рассмотрим подробнее, как вычисляется оценка общего расстояния. На рисунке 2 изображена середина процесса нахождения пути. Допустим, что текущий узел задаётся координатами (1,2). Его соседний узел, занесённый в границу, задаётся координатами (2,2). Минимальное расстояние до этого узла от начального $g = 4$ (количество пройденных клеток – фиолетовая линия). Эвристическая функция задана как сумма разности координат по x и по y . Оценка расстояния от рассматриваемого узла к целевому $h = (4 - 2) + (5 - 2) = 5$. Тогда оценка общего расстояния для рассматриваемого узла $f = 4 + 5 = 9$.

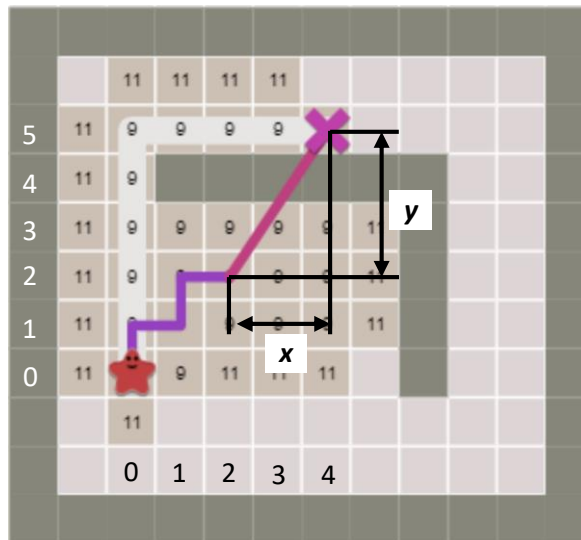


Рисунок 2 – Пример середины процесса

При написании кода был использован ориентированный взвешенный граф (так как это более общий случай), для которого был предварительно составлен массив, содержащий количество шагов между каждым из узлов.

Исследование алгоритма

Заключение

A* алгоритм хорошо подходит для определения кратчайшего пути на графе, который является моделью карты, с минимальными затратами по времени и памяти. Однако, если специальные знания о графе скудные или отсутствуют, этот алгоритм будет менее эффективен. В таком случае предлагается использовать алгоритм Дейкстры.

Список литературы

1. Питер Э. Харт, Нильс Дж. Нильссон и Бертрам Рафаэль. A Formal Basis for the Heuristic Determination of Minimum Cost Paths - IEEE Transactions on Systems Science and Cybernetics , гл. 4, стр. 100-107, июль 1968 г.
2. Седжвик Роберт. Фундаментальные алгоритмы на С. Анализ/Структуры данных/Сортировка/Поиск/Алгоритмы на графах: Пер. с англ./Роберт Седжвик – СПб: ООО «ДиаСофтЮП», 2003. — 1136 с.
3. Люгер Джордж Ф. Искусственный интеллект: стратегии и методы решения сложных проблем, 4-е издание: Пер. с англ. – М: Издательский дом «Вильямс», 2003. – 864 с.
4. Amit Patel. Red Blob Games [Электронный ресурс] // Introduction to the A* Algorithm. URL: <https://www.redblobgames.com/pathfinding/a-star/introduction.html> (даты обращения: 20.02.2022-20.05.2022).

Приложение