

Алгоритм A^*

Эдуард

26 июля 2019 г.

1 Постановка задачи

Имеем $G = (V, E)$ - неориентированный взвешенный граф с весовой функцией $g : E \rightarrow \mathbf{R}$, вершины которого соответствуют положениям, которые можно занимать; ребра - траектории, по которым проходим, двигаясь между соседними вершинами. Известны стартовая v_S и финишная v_F вершины.

Путь $\pi\{v_S, \dots, v_F\}$ - последовательность вершин и ребер, которая связывает v_S и v_F . Пусть P - множество путей из v_S в v_F .

Вес пути: $g(\pi\{v_S, \dots, v_F\}) = \sum_i g(v_i)$, где $i = S, S + 1, \dots, F - 1, F$.

Задача: Найти путь $\pi_m\{v_S, \dots, v_F\} = \arg \min_{\pi \in P} g(\pi)$

Определим эвристическую функцию $f(v_i) = g(v_i) + h(v_i)$, где

$g(v_i) = g(\pi\{v_S, \dots, v_i\})$,

$h(v_i)$ - оценка веса пути из v_i в v_F . Должна удовлетворять условию:

$h(v_i) \leq h^*(v_i)$,

$h^*(v_i)$ - действительный вес пути из v_i в v_F .

В данной задаче можно использовать $h(v_i) = \rho(v_i, v_F)$ - евклидово расстояние от v_i до v_F .

2 Псевдокод

```
A-star(start, goal)
  OPEN :=  $\emptyset$ ;
  CLOSE :=  $\emptyset$ ;
  OPEN.push(start);
  g[start]=0;
  f[start] = g[start]+h[start];
  while OPEN.size>0 do
    v:=arg minv∈OPEN f(v);
    OPEN.remove(v);
    if v=goal then
      getPathFromParentPointers(v);
      return "path found";
    end
    CLOSE.push(v);
    Expand(v);
  end
  return "no path found"
end

Expand(v)
  SUCC=getSuccessors(v);
  foreach v' ∈ SUCC do
    gNew = g(v) + g({v, v'});
    if v' ∈ CLOSE or gNew ≥ g(v) then
      continue;
    end
    if v ∉ CLOSE or gNew < g(v) then
      parent[v']=v;
      g(v')=gNew;
      f(v)=g(v)+h(v);
      if v' ∉ CLOSE then
        Close.push(v')
      end
    end
  end
end
```

Algorithm 1: A* algorithm

Список литературы

1. A Method of Ship's Path Planning at Sea by Yuanliang Zhang
Sci-Hub
2. A*-based Pathfinding in Modern Computer Games by Xiao Cui and Hao Shi
Paper
3. Path Planning for Ground Simulation Object Based on A* Algorithm
by Zhao Zhiqiang
Sci-Hub
4. Grid-based angle-constrained path planning by Konstantin Yakovlev
(Example of pseudocode)