

Assignment 1.2: GBF and A*

Aleksandra Ristic

a. Representation

The idea is to create a graph that will have cities as nodes and the weight of the edges represents the distance between the two cities.

Set of variables:

$X = \{x_1, x_2, \dots, x_i, \dots, x_n\}$ - n is number of cities and x_i represents one city ie one node.

Domain of each variable:

$x_i = \{0, 1\}$ - if city i is chosen to be in solution or not

Constrain of the problem:

$C = \{x_1 \neq x_2, x_2 \neq x_3, \dots\}$

The goal is to calculate global minimum of distance.

$$d(x_i, x_j) = \min_{i=1,2,\dots,m} d_i, x_i, x_j \in X$$

$$d_i = \sum_{j=1}^{k_i} v_j$$

m is number of possible path between x_i, x_j

x_i is city we are starting from

x_j is the city we want to get to

k_i number of cities from which are passing i -th path

v_1 is distance between start city and next visited city

v_{nj} is distance of path between end city and previously visited cities

b. Equation GBF

$$f(n) = h(n)$$

$f(n)$ is evaluation function which depends only of heuristic function - $h(n)$. Heuristic function in this approach represent straight-line distance between two cities, current and end city.

n represents our current city.

c. Equation A*

$$f(n) = g(n) + h(n)$$

$f(n)$ represents estimated total cost of path through n to goal

$h(n)$ is heuristic value ie estimated cost from n to goal

$g(n)$ is cost so far to reach n

d. Difference between GBF and A*

Difference between these two algorithms refer to differences in way of calculating $f(n)$.

GBF or Greedy Best First Search algorithm doesn't care about past, in the way that he will decide which node will be expanded only on the basis of distance between current city and end city, and because of that path found may not be the optimal one. While, A* will find the optimal solution.

These two algorithms are type of BFS(Breath First Search) algorithm. That means that we are putting in ordered queue all expanded nodes and we are ordering it by evaluation function. After that we are choosing the best node from queue, trying to use it in order to find solution and in the same time removing from queue, but in queue we will put neighbors of that city (node).