

C++ Optimisation Task

Naim Sen 9638390

Dec 2016

This program looks at the optimum position of two delivery hubs (one in the North, one in the South) vs one delivery hub. The cost of each hub is calculated as well as the payback time of two hubs (since startup cost of two hubs will always be more expensive than one). There are various parameters that can be adjusted which will change the resulting position and cost of the hubs. These include:

- `dividingLine` : determines the latitude limit up to which the two hubs service.
- `ppm` : determines the cost per mile per day of running a truck. It is assumed that all places are visited each day.
- `hub_cost` : the cost for setting up one hub that serves the entire UK. It is assumed that when setting up two hubs, each hub is two thirds of the cost since the hubs serve fewer places.
- `running_cost` : the cost per day for running a hub.

Other approximations:

- The distance is population weighted such that populations greater than 250,000 require 1 additional truck, 500,000 require 2 additional trucks etc. up to 1,000,000.
- The great circle distance approximation is improved by assuming that larger distances deviate from the great circle distance more than smaller distances. Hence, the function `total_distance` includes a check for distances greater than 150 & 250 miles and multiplies by 1.3 & 1.1 respectively.

performance improvements:

- the repeated hillclimb is split into two for each hub (North, South and Global) and each hillclimb is dealt with asynchronously by separate threads where available to reduce runtime.
- the repetitions of the hillclimb and resolution of the hillclimb can be modified for the performance of the user's system by adjusting `NRand_evals` and `step`.