Optimization Strategy

Heuristic 1

Suppose there is a graph consisting of ‘n’ nodes, say A_1, A_2 up to so on A_n and $x, y \in A$, where A is a set of nodes. In this, if we add a new edge, say (u, v), then if the following condition satisfies then we update the path between x and y.

$$d(x, u) + d(u, v) + d(v, y) < d(x, y)$$

where d(x, y) is shortest distance between nodes x and y

We do this because this saves time which is required in recomputing the distance between two nodes of which only a few nodes are newly added in a graph on which Dijkstra Algorithm has already been run.

Heuristic 2

One more heuristic which we use to optimize our algorithm further is that if any number of new nodes and edges are added to an unweighted graph, then only the shortest distance between those two nodes can change whose earlier shortest distance is greater than or equal to 3 units.

In other words, let $x, y \in A$, where A is a set of nodes and $d(x, y)$ be the shortest between the nodes. If $d(x, y) <= 2$, then the shortest distance between x and y can never change.

We have implemented our optimized Dijkstra in using the above two heuristics together in order to get the most optimized result.