A Method for the Shortest Path Search by Extended Dijkstra Algorithm

Group-8

S.Pavan Sai Krishna - 2022H1030061H
Soumyadip Guha - 2022H1030092H
Havisteja Dantu - 2022H1030083H

1.Introduction

- The path search problem involves finding the optimum path between the present location and the destination under given conditions.

- This problem arrives in networks such as the highway system, railroads, and communication networks, and cover a wide range of applications. In particular, car navigation systems.


0.3mi
South 35 Wichita
35 SOUTH
Wichita
W 106th St
LIMIT
65
63 m h
Speed
I-83
3:45 PM
Arrival

- A number of path search methods have been proposed for use in car navigation systems, but while all of those methods have their respective merits and demerits, no exceptionally good method has yet been established.
- The three main types of algorithms that are currently being used or studied for use in path search problems are
  1)A* algorithm
  2)Genetic algorithm
  3)Dijkstra's algorithm

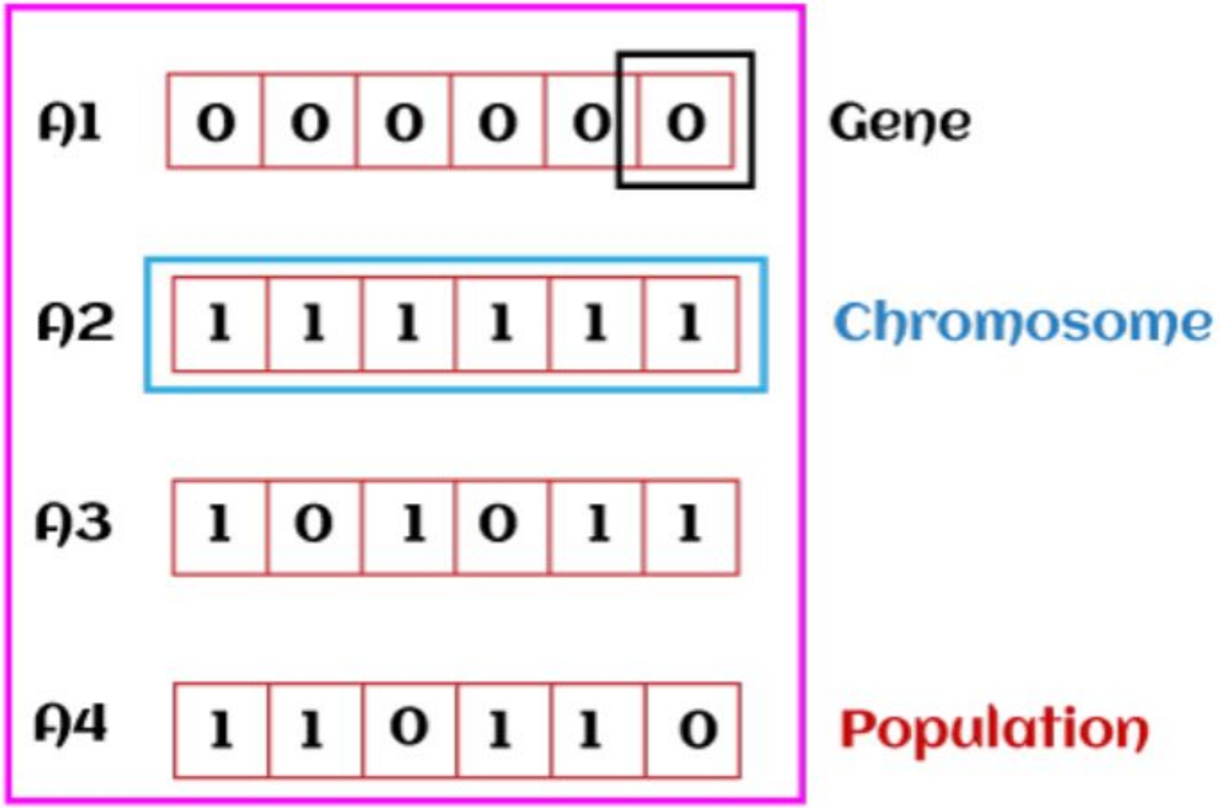
1. A* Algorithm

- A* was initially designed as a graph traversal problem, to help build a robot that can find its own course. It still remains a widely popular algorithm for graph traversal.
- It searches for shorter paths first, thus making it an optimal and complete algorithm. An optimal algorithm will find the least cost outcome for a problem, while a complete algorithm finds all the possible outcomes of a problem.

- It is the sum of two variables values that determines the node it picks at any point in time.
- At each step, it picks the node with the smallest value of ‘f’ (the sum of ‘g’ and ‘h’) and processes that node/cell. ‘g’ and ‘h’ is defined as simply as possible below:
- ‘g’ is the distance it takes to get to a certain square on the grid from the starting point, following the path we generated to get there.
- ‘h’ is the heuristic, which is the estimation of the distance it takes to get to the finish line from that square on the grid.

2.Genetic algorithm

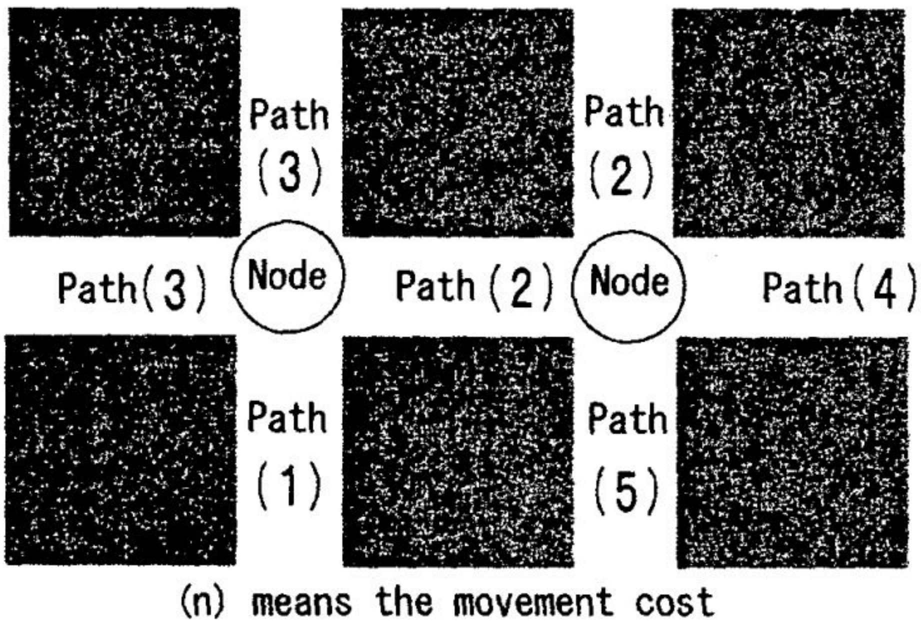
- A genetic algorithm is an adaptive heuristic search algorithm inspired by "Darwin's theory of evolution in Nature.
- Basic Terminology:
  - Population
  - Chromosome
  - Gene
  - Allele
  - Fitness Function
  - Genetic operators
  - Selection


A1
0 0 0 0 0 0
Gene
A2
1 1 1 1 1 1
Chromosome
A3
1 0 1 0 1 1
A4
1 1 0 1 1 0
Population

3. Dijkstra's Algorithm

This is an algorithm for finding the optimum path. Because this algorithm searches for the minimum-cost path among all paths in order, beginning from the starting point, the search region expands concentrically. This method thus has the disadvantages of poor search efficiency and a long search time when the distance to the destination is large.

We describe the algorithm of the Dijkstra method. First, we consider road intersections as nodes, the roads that connect nodes as paths (routes), and the length and ease of transit of a path as the movement cost (cost)

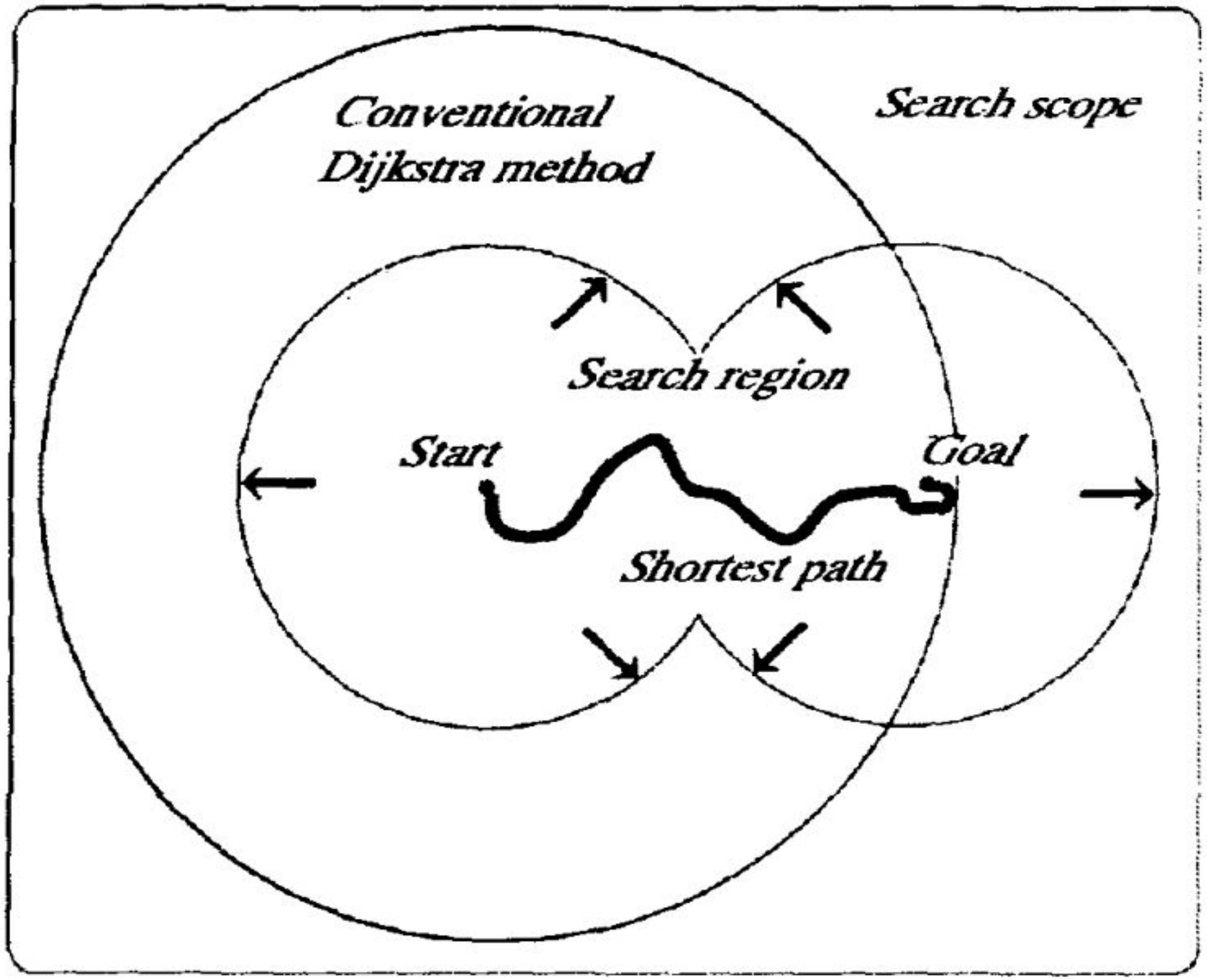


Algorithm for Dijkstra

- Step 0: Mark the starting point.
- Step 1: Calculate the movement cost for movement from the starting point to each node connected to the starting point, and mark the node for which that value is the smallest.
- Step 2: Calculate the movement cost for movement between the starting point and each node connected to the marked node and mark the node for which that value is the smallest.
- Step 3: Repeat Step 2 until the destination is marked.

Extended Dijkstra's Algorithm

- In the conventional Dijkstra method, the search region generally expands concentrically.
- This method has the disadvantage that the solution cannot be obtained in real time if the distance from the starting point to the destination is long.
- Therefore, we propose a new algorithm in which the Dijkstra method is applied from both directions, which is to say beginning from the starting point and also beginning from the destination.
- In this way, the concentric expansion of the search region is restricted and the number of nodes to be searched can be reduced.


Conventional
Dijkstra method
Search scope
Search region
Start
Goal
Shortest path

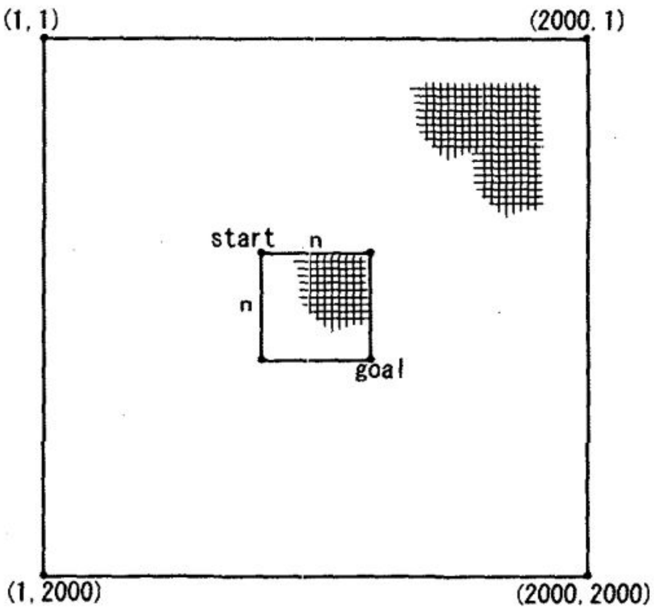
Algorithm for Extended Dijkstra

- Step 0: Mark the starting point and destination point.
- Step 1: Calculate the movement cost for movement from the starting point to each node connected to the starting point and for movement from the destination to each node connected to the destination, and mark the nodes for which those values are the smallest.
- Step 2: Mark the nodes that have the smallest movement cost from the unmarked nodes that are connected to the marked nodes to either the starting point or the destination.
- Step 3: Repeat Step 2 until the search from the starting point and the search from the destination overlap and then end.

Simulation

- We created a. 2000 x 2000 node grid pattern of paths to serve as the search region.

- The path movement costs were assigned with a uniform random distribution in five levels, 1 to 5



- We can change the distance between source and destination by changing the size of the square.
- We performed 10 simulation for each distance of total 17 distance ranged from 10 to 450.

Simulation Results

Distance	Conventional Dijkstra method Search time (s) (time1)	Number of node (count1)	Mean cost (cost1)	Extended Dijkstra method Search time (s) (time2)	Number of node (count2)	Mean cost (cost2)
10	0.03	645	38.1	0.01	305	38.1
20	0.25	2671	74.5	0.05	1252	74.8
30	0.93	6132	110.0	0.18	2929	110.3
40	2.34	11106	147.0	0.44	5289	147.3
50	4.43	16744	178.7	0.84	8079	178.9
60	7.65	23832	211.2	1.46	11518	211.4
70	12.10	32133	244.6	2.31	15528	244.9
80	18.63	42496	280.4	3.50	20348	280.5
90	26.89	54011	316.0	5.11	26022	316.1
100	37.36	66890	350.7	7.18	32455	350.8
150	127.06	148891	521.0	24.82	72558	521.6
200	311.55	267064	694.2	61.18	130238	694.6
250	616.41	415855	863.0	122.62	203524	863.2
300	1087.29	598716	1031.9	218.37	293502	1031.9
350	1771.71	816690	1203.1	358.87	401059	1203.3
400	2692.92	1062340	1370.7	552.75	523268	1370.9
450	3926.72	1344920	1540.2	811.82	663627	1540.4

	Time (%)	Count (%)	Cost (%)
10	12.82	47.27	100.00
20	17.58	46.88	99.60
30	18.81	47.77	99.73
40	18.69	47.63	99.80
50	19.00	48.25	99.89
60	19.02	48.33	99.91
70	19.06	48.32	99.86
80	18.79	47.88	99.96
90	19.00	48.18	99.97
100	19.21	48.52	99.97
150	19.54	48.73	99.88
200	19.64	48.77	99.94
250	19.89	48.94	99.98
300	20.08	49.02	100.00
350	20.26	49.11	99.98
400	20.53	49.26	99.99
450	20.67	49.34	99.99

Observations

- With the extended Dijkstra method, the number of nodes checked is reduced by about one-half compared to the conventional Dijkstra method.
- By reducing the number of nodes checked, the search time was shortened to about 1/5 the original time.
- The conventional and extended Dijkstra method both give the optimum solution and the minimum movement cost.
- We got difference between the movement cost between two algorithm in 44 out of 250 runs, which is about 18%.

Conclusion

- Extended Dijkstra algorithm can give near optimum solutions and its confirmed by the simulation.
- The result can show that time can be greatly reduced without much impact on movement cost.
- This new method can be applied in the car navigation system and other such path search problem here real-time solution required.
- As this is based on Dijkstra so problem of long search time for broad search scope is remained.
- In future we can optimize this algorithm further using genetics algorithm.