IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 20, Issue 5, Ver. I (Sep - Oct 2018), PP 38-45
www.iosrjournals.org

Comparative Analysis of Path Finding Algorithms

Shabina Banu Mansuri[1], Shiv kumar[2]
1M.Tech Scholar, Computer Science & Engineering, Mewar University, Gangrar, Chittorgargh-312901, India.
2Computer Science & Engineering, Mewar University, Gangrar, Chittorgargh-312901, India.
corresponding author: Shabina Banu Mansuri

Abstract:*Searching is the problem-solving technique in artificial intelligence. There are various search algorithms related to search like Dijkstra, Depth first search, Breadth first search, A*,Hill Climbing, Best first search algorithms and their variants. But most famous algorithms are Dijkstra and A* search algorithms. Dijkstra algorithm is used to solve the path finding problem. Almost all application is using Dijkstra till now. As technology grows, the speed of vehicles also increased like rocket. That is why a scholar as well as Microsoft, Google like companies starts working on A* algorithm because A* performance is better than Dijkstra. At present time, even one second has also weightage for Light motor vehicles because of its very high speed. That is why it is necessary to design and develop tools to analysis the performance of Dijkstra, A* and their variants. According to designed or developed tool IDA* performance is better than Dijkstra and A*.*
Keywords: *Light motor vehicles, Path Finding Problem, Dijkstra, A*, IDA*.*

Date of Submission: 05-09-2018 Date of acceptance: 21-09-2018

I. Introduction

Searching is the universal technique of problem solving in AI . There Are Two Kinds of AI Searching Technique Uninformed Search and Informed search

1.1 **Uninformed Search** –uninformed search also called Blind search, is a class of general purpose search algorithm that operate in a brute –force way .Uses no knowledge about problem. BFS(Breadth First Search) ,DFS(Depth First Search) ,Depth limited search Algorithm, Depth First Depending search , Uninformed cost search are example of Uniformed searching algorithm technique[1]

1.2 **Informed Search-** informed search, heuristic is used as a guide that leads to better overall performance in getting to the goal state. Hill Climbing search, Best First Search, A* Algorithm, IDA* Algorithm is example of Informed searching algorithm technique[1]

1.3 **Evolution strategies**

1. Completeness – An algorithm is said to be complete if it provides at least one solution.
2. Optimality – An algorithm is said to be optimal if the solution found guaranteed to be the best or lowest cost.
3. Time Complexity – The upper bound on the time required to find a solution, as a function of the complexity of the problem.
4. Space Complexity – The upper bound on the storage space (memory) required at any point during the search, as a function of the complexity of the problem.

1.4 Dijkstra algorithm is applied to automatically find directions between physical locations, such as driving directions on websites like Map quest or Google Maps. In a networking or telecommunication applications, Dijkstra algorithm has been used for solving the min-delay path problem. For example in data network routing, the goal is to find the path for data packets to go through a switching network with minimal delay. It is also used for solving a variety of shortest path problems arising in plant and facility layout, robotics, transportation, and VLSI design [2]. The algorithm exists in many variants; Dijkstra original variant found the shortest path between two nodes, but a more common variant fixes a single node as the "source" node and finds shortest paths from the source to all other nodes in the graph, producing a shortest path tree.

1.5 A* search, which is the most popular form of best-first search, repeatedly picks the node with the lowest f(n) to expand next. It turns out that if the heuristic function h(n) satisfies certain conditions, then this strategy is both complete and optimal. A* algorithm is used to find a path from a given initial node to a given goal node. It employs a "heuristic estimate" h(x) that gives an estimate of the best route that goes through that node. It visits the nodes in order of this heuristic estimate. It follows the approach of best first search. The secret to its success is that it combines the pieces of information that Dijkstra algorithm uses (favouring vertices that are close to the starting point) and information that Best- First-Search uses (favouring vertices that are close to the goal). In the standard terminology used when talking about A*,

g(n)represents the exact cost of the path from the starting point to any vertex n, and h(n) represents the heuristic estimated cost from vertex n to the goal[4].

f (node) = g(node) + h(node)

g (node) is the cost of the path so far leading up to the node, and h(node) is an underestimate of the distance of the node from a goal state; f is called a path-based evaluation function. When operating A*, f(node) is evaluated for successor nodes and paths extended using the nodes that have the lowest values of f.

1.6 IDA* algorithm

Iterative deepening A* (IDA*) is a graph traversal and path search algorithm that can find the shortest path between a designated start node and any member of a set of goal nodes in a weighted graph. It is a variant of iterative deepening depth-first search that borrows the idea to use a heuristic function to evaluate the remaining cost to get to the goal from the A* search algorithm. Since it is a depth-first search algorithm, its memory usage is lower than in A*, but unlike ordinary iterative deepening search, it concentrates on exploring the most promising nodes and thus doesn't go to the same depth everywhere in the search tree. Unlike A*, IDA* doesn't utilize dynamic programming and therefore often ends up exploring the same nodes many times [3, 4].

II. Literature survey

Harita vamja, et al in 2017,has analysed Dijkstra ,A* and IDA*.They done the comparative analysis between time ,complexity etc. But they has not taken the most important parameter is that execution time [5].

Sharwan. Kr. Sharma, B.L. Pal in 2015, "Shortest path Searching for road network using A* Algorithem "they have complete analiys of Dijikstra and A * algo to find the shortest on the basis of obstacal and without obstacal help of bidirecational .The finaly show the A* best find path[6]

Ankit Bhadoria, et.al in 2014 ," optimized Angular a Star Algorithm for Global Path Search Based on Neighbor Node Evaluation"In this paper we have done analysis on partially know environment situation. Optimal path is planned by new heuristic approach over the A * algorithem in this paper A* algo is better then Dijkstra algo in robotic path findin ang energy constraint.[7]

Abhishek Goyal , et.al in 2014,has compared A* and Dijkastra to find shortest path between source and destination based on execution time and proved that A* solves the problem nearly half time lesser then Dijkstra algorithm. Implementation was based on console [8].

Kairanbay Magzhan , et.al , 2013, has analysed the s Shortest path algorithms like Dijkstra's Algorithm, Floyd-Warshall Algorithm, Bellman-Ford Algorithm, and Genetic Algorithm. Genetic algorithm (GA) provide better result and optimal solution.[9]

Thepparit Sinthamrongruk, et.al , in 2013, has compaired A* Path finding and Waypoint Navigator Algorithm on Android and Ios Operating System based on searching time. According to executed test case, A-star was better than waypoint algorithm [10].

Pawan Jindal ,et. al in 2010 has Analysed the single source shortest path algorithm and all pair shortest path problem like Ball man ford , Dijkastra and Floyd warshall algorithmthay were got time and space complexity[11].

Yuxi Li, et.al, in 2007, has designed "Fast Exact Multi Constraint Shortest Path Algorithms". Has proposed two algorithm A*and Fringe MCSP (multi constraint shortest path)and fringe MCSP based on A* and the MCSP has good performance in both the cases. Both the algorithms were NP hard problem [12].

III. Problem Statement & Objectives

3.1 Problem Statement

Now a day's time is the most important factor for the path searching algorithm due to new technology in market. Memory price are decreasing day to day and LMV's speed increasing day by day. So, it is necessary to analyse which algorithm will provide the better result for searching the path in different - different scenario, basically with obstacle and without obstacle

3.2 Objectives

- To study the search algorithm.
- To compare Dijkstra, A* and IDA* using following parameters
  - Time complexity
  - Space complexity
  - Execution time
  - With obstacle
  - Without obstacle
- To design and develop tool for path finding path

IV. Proposed System

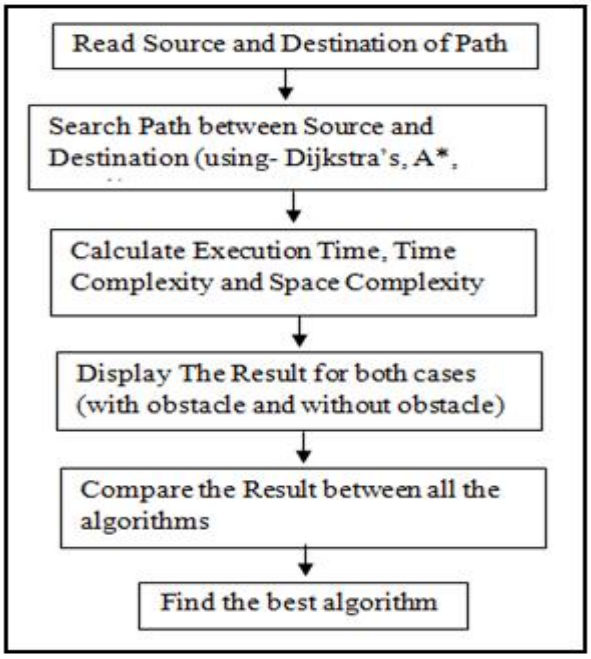


Figure 4.1 Proposed systems

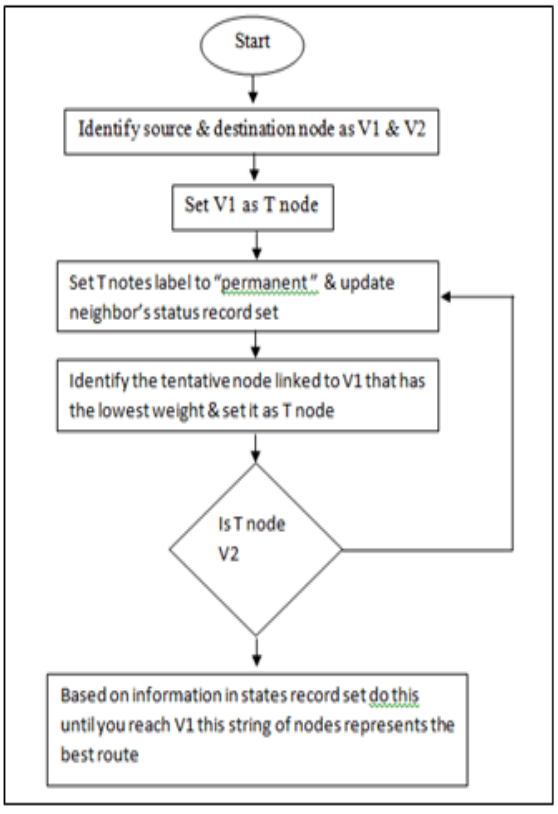


Figure 1.4.1 Fowl chart of Dijkstra Algorithm

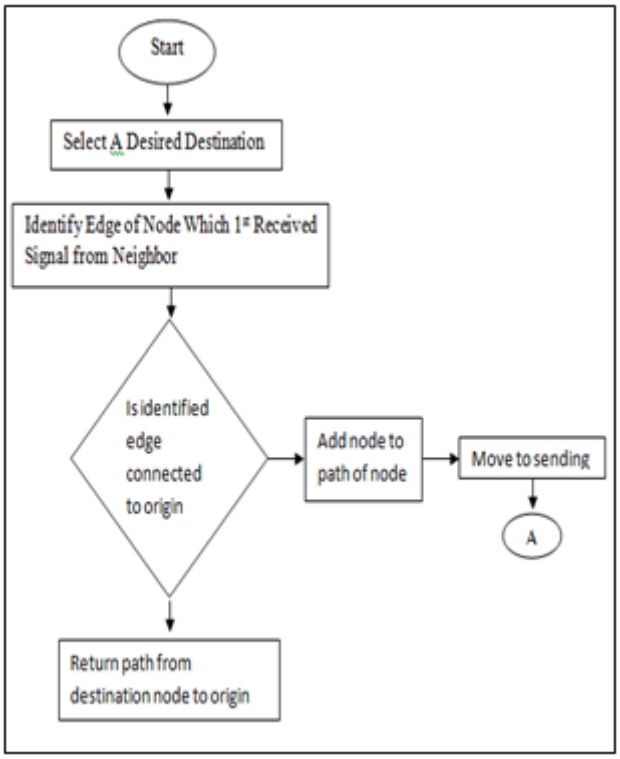


Figure 1.5.1 Flow chart of A* Algorithm

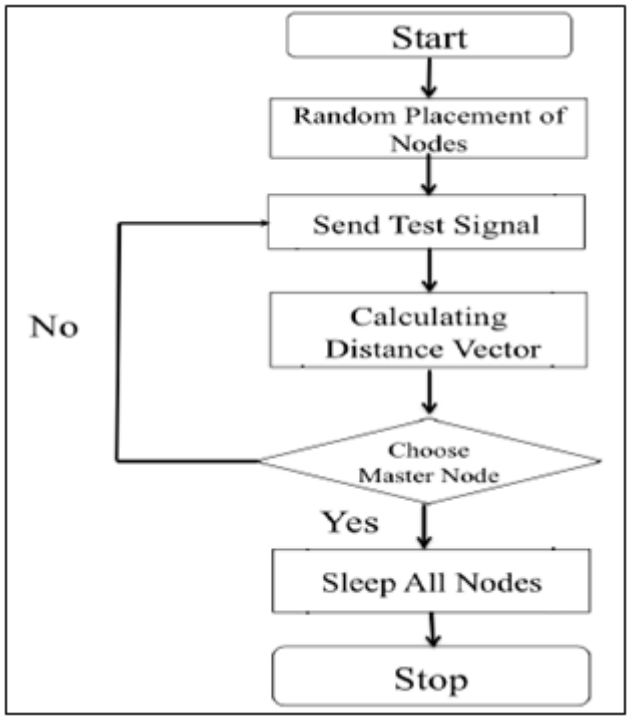


Figure 4.3 Flow chart of IDA* Algorithm

V. Experiment Result And Analysis

Mathematical Formula used in tools are
Length =total number of grid
Time=End time –Start time
Operation=Start grid + number of open grid + number of successor grid + Goal

A) Snapshot of Tool

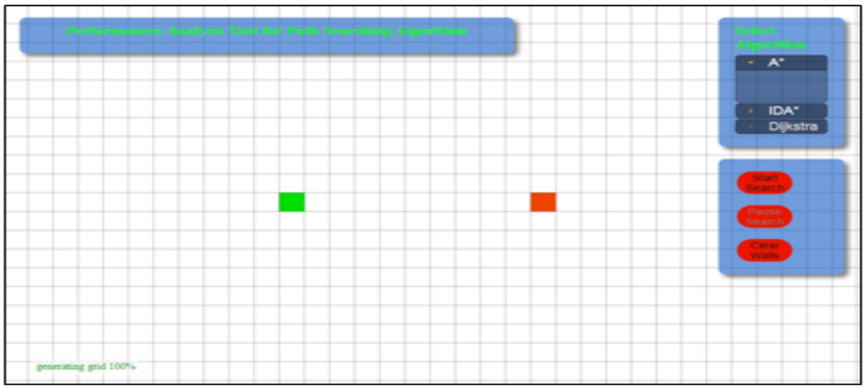


Figure 5.1 UI of Tool

5.1 Snapshot without Obstacle:

B) Snapshot of Dijkstra Algorithm without obstacle

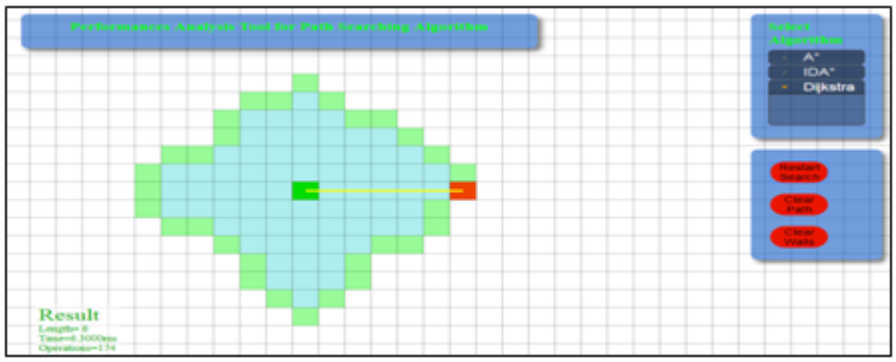


Figure 5.2 Dijkstra algorithm

C) Mathematical analysis of Dijkstra algorithm

Length=length is calculate of total grid used in path made
Time=when we start search the time is count by processor time of computer
Operation=1(start) +32(open) +140(successor) +1(goal) =174

D) Snapshot of A* Algorithm

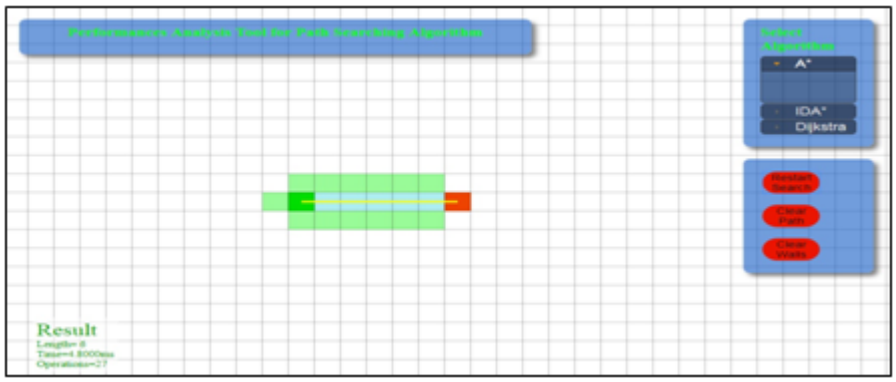


Figure 5.3 A* algorithm

E) Mathematical analysis of A*

Length=length is calculate of total grid used in path made
Time=when we start search the time is count by processor time of computer
Operation=1(start) +18(open) +7(successor) +1(goal) =27

F) Snapshot of IDA* algorithm

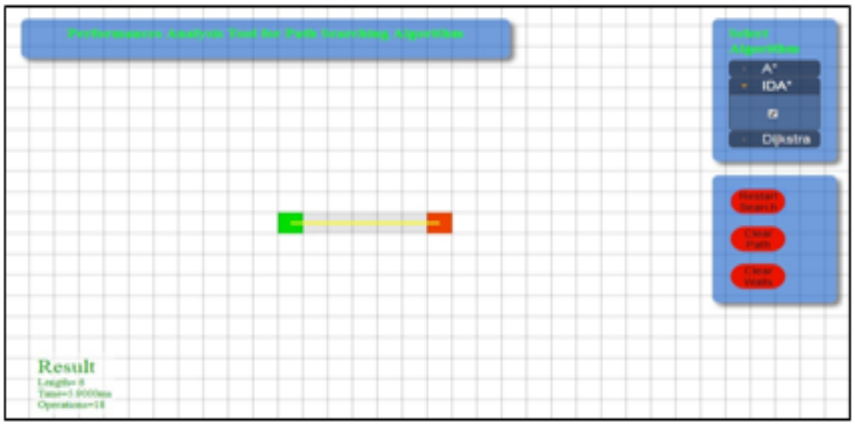


Figure 5.4 IDA* algorithm

G) Mathematical Analysis of IDA* Algorithm

Length=length is calculate of total grid used in path made
Time=when we start search the time is count by processor time of computer
Operation=1(start) +10(open) +6(successor) +1(goal) =18

5.2 Snap shot with obstacle:

A) Snapshot of Dijkstra algorithm with obstacle

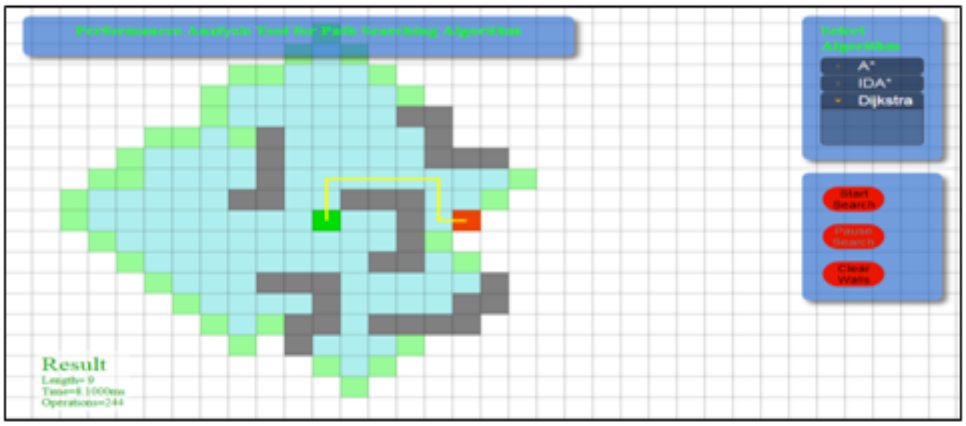


Figure 5.5.Dijkstra algorithms with obstacle

B) Snapshot of A* algorithm with obstacle

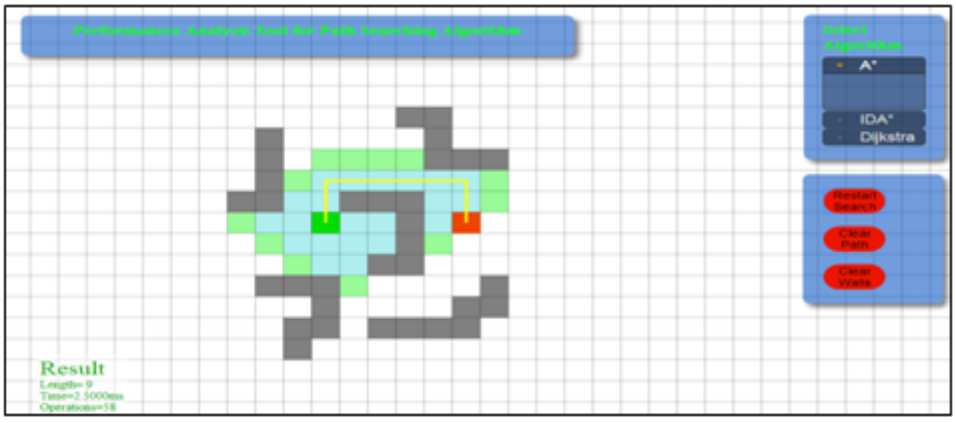


Figure 5.6 A* algorithms with obstacle

C) Snapshot of IDA * with obstacle

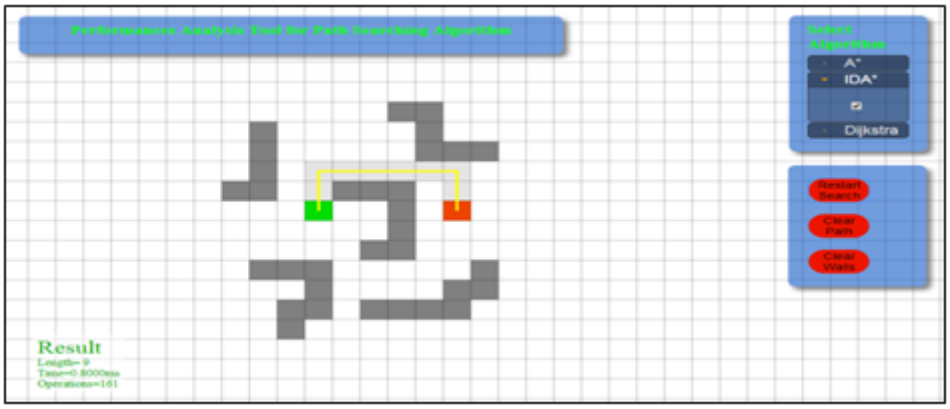


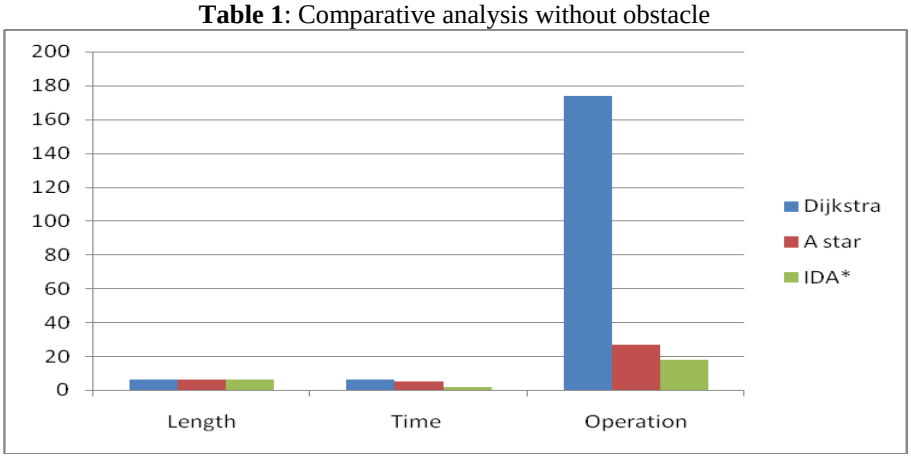
Figure 5.7 IDA* algorithms with obstacle

VI. RESULT ANALYSIS

6.1 Result analysis without obstacle:

Algorithm	Length	Time	Operation
Dijkstra	6	6.30	174
A star	6	4.80	27
IDA*	6	1.70	18

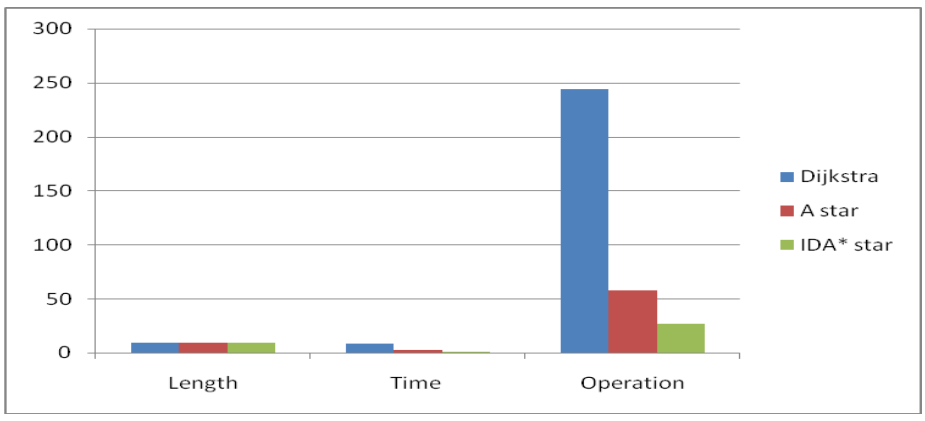
Table 1: Comparative analysis without obstacle



6.2 Result analysis with obstacle:

Algorithm	Length	Time	Operation
Dijkstra	9	8.10	244
A star	9	2.50	58
IDA* star	9	0.90	27

Table 2: Comparative analysis with obstacle



VII. Conclusion

According to executed test cases on design and developed tool IDA* algorithm is performs better then Dijkstra and A* algorithm in both the cases (i.e. with obstacle and without obstacle).

Acknowledgements

Foremost, I would like to express my sincere gratitude to my advisor Shiv Kumar for the continuous support of my study, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this paper. I could not have imagined having a better advisor and mentor for this paper.

References

[1]. Fredman, Michael Lawrence; Tarjan, Robert E. (1984). Fibonacci heaps and their uses in improved network optimization algorithms. 25th Annual Symposium on Foundations of Computer Science. IEEE. pp. 338–346. doi: 10.1109/SF CS.1 984.71 5934

[2]. Disha Sharma, Sanjay Kumar Dubey,"Anytime A* Algorithm – An Extension to A* Algorithm", IJSER 2013, ISSN 2229-5518

[3]. Russell, Stuart J.; Norvig, Peter (2002), "3.4 Uninformed search strategies", Artificial Intelligence: A Modern Approach (2nd ed.), Prentice Hall.

[4]. Phaneendhar Reddy Vanam,"Shortest path using A* Algorithm", December 13, 2011

[5]. Harita vamja ,Ayesha shaika et al" comparative analysis of different path finding algorithms to study limitations and progress",IJETAE ,VOLUME 7 ISSUE 9,SEPT. 2017.

[6]. Shrawan Kr. Sharma, B.L.Pal,Shortest Path Searching for Road Network using A* Algorithm,IJCSMC, Vol. 4, Issue. 7, July 2015, pg.513 – 522

[7]. Ankit Bhadoria,Ritesh kumar singh," Optimized Angular a Star Algorithm for Global Path Search Based on Neighbor Node Evaluation",IJISA, july 2014

[8]. Abhishek Goyal , ,Prateek mogha,Rishabh luthra,Neeti sangwan,"Path finding :A* or Dijkstra's",IJITE,vol 2,issue 1,Jan 2014

[9]. Kairanbay Magzhan , hajar mat jani,"A review and evaluations of shortest path algorithms",IJSTR,vol 2issue 2 , june 2013Thepparit Sinthamrongruk, Krisada Mahakitpaisarn, and Wapee Manopiniwes"A Performance Comparison between A* Pathfinding and Waypoint Navigator Algorithm",IACSIT,vol 5,aug 2013

[10]. Thepparit Sinthamrongruk,Krisada mahakitpaisarm ,wapee manopiniwes, "A performance comparison between A* Path finding and Waypoint Navigator Algorithm on Android and Ios Operating System",IACSIT,vol5 , Aug 2013

[11]. pawan jindal ,amit kumar,shisir kmar,"analysis of shortwst path algorithms",GJCST,vol 10 issue 8 sept 2010

[12]. Yuxi Li Janelle Harms Robert Holte,"Fast Exact MultiConstraint Shortest Path Algorithms",Department of Computing Science, University of Alberta, Edmonton, AB, Canada T6G 2E8,of IEEE Communications Society ICC 2007,1-4244-0353-7/07/$25.00 ©2007 IEEE

Shabina Banu Mansuri "Comparative Analysis of Path Finding Algorithms " IOSR Journal of Computer Engineering (IOSR-JCE) 20.5 (2018):, pp. 38-45