International Journal of the Physical Sciences Vol. 6(20), pp. 4721-4731, 23 September, 2011
Available online at http://www.academicjournals.org/IJPS
DOI: 10.5897/IJPS11.573
ISSN 1992 - 1950 ©2011 Academic Journals

Full Length Research Paper

Path planning for a mobile robot in dynamic environments

P. Raja* and S. Pugazhenthi

School of Mechanical Engineering, SASTRA University, Thanjavur – 613401, Tamilnadu, India.

Accepted 9 August, 2011

Path planning for mobile robots has been gaining extensive research attention recently. Optimal path planning increases the effectiveness of a mobile robot. There are many algorithms to solve the path planning problems overcoming obstacles. However most of the algorithms are applicable to specific shapes or suitable for static environments. This paper introduces a new method of global path planning for a robot moving in an environment cluttered with obstacles which have arbitrary shape, size and location. The proposed algorithm is applicable to static, partially dynamic as well as dynamic environments containing obstacles. Simulation of variety of cases are presented and discussed.

Key words: Mobile robot, path planning, dynamic environment, static and dynamic obstacles.

INTRODUCTION

An important research theme in mobile robotics is the design of path planning system. Path planning is an important problem in navigation of autonomous mobile robots, which is to find an optimal collision-free path from a starting point to a goal in a given environment according to some criteria such as distance, time or energy while distance or time being the most commonly adopted criterion (Raja and Pugazhenthi, 2009).

There are many path planning algorithms such as potential field methods (Ge and Cui, 2000), visibility graph methods (Li et al., 2002) and grid methods (Boschian and Pruski, 1993). Potential field method is widely used for mobile robot path planning for both static and dynamic environments. Its main part is an artificial potential field, which 'attracts' the robot to the target and 'repulses' it from the obstacles. The potential field approach can be used as a global motion planning algorithm when the environment is relatively uncluttered and the obstacles are convex (Azariadis and Aspragathosc, 2005). However, it exhibits problems of local minima, no passage for the robot between obstacles and goals non-reachable with obstacles nearby (Ge and Cui, 2000; Koren and Borenstein, 1991). The visibility graph method constructs a graph of vertices of polygons representing obstacles. It means that two vertices are connected in the graph if they are mutually visible (Saska et al., 2006). A shortest path is then determined using standard Dijkstra's algorithm (Jorgen and Gutin, 1979).

The main problem of this method is that, they have more complicated search paths and lower search efficiency. In the grid methods, where grids are used to form the map of the environment, the main problem is how to determine the size of grids, the smaller the size of grids, the more precise the representation of the environment. However, using smaller grids will result in exponential increase in memory space and search range (Zheng et al., 2007).

The cell decomposition algorithm explicitly computes the configuration space of the mobile robot, decomposes the resulting space into cells, and then searches for a route in the free space cell graph (Latombe, 1991). However, the algorithm suffers from the drawback of high time complexity. Path planning problem can also be solved by vector field approach (Arkin and Murphy, 1990). In a given environment the vector field must be frequently recalculated, however this need not be too computationally intensive in an environment with few obstacles. The main disadvantage with vector field based approaches is that they are ideal for static obstacles but tend to produce poor performance in dynamic environments.

There are many other intelligent algorithms for path planning such as genetic algorithm (Xiong et al., 2004; Liu et al., 2004), ant colony algorithm (Bell and McMullen,

*Corresponding author. E-mail: raja_sastra@yahoo.com.

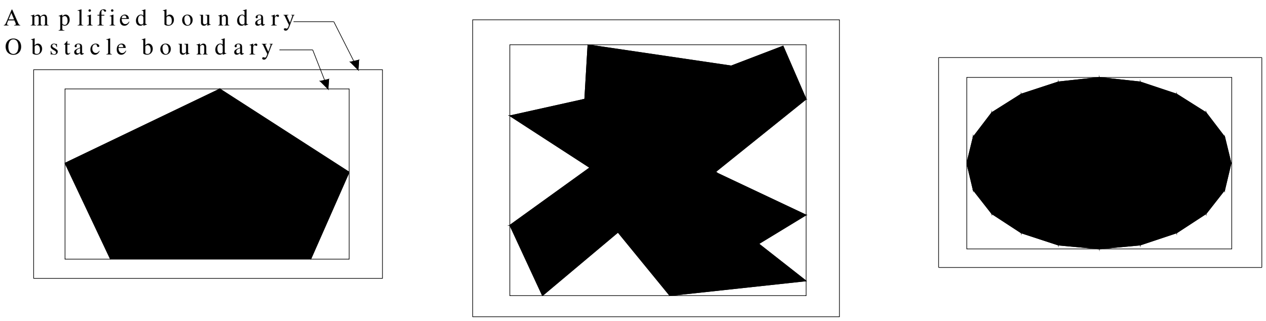

Figure 1. Example of obstacle amplification for path negotiation.

2004), neural networks (Bin et al., 2004), fuzzy logic (Banga et al., 2011) and so on. However, these algorithms cannot reach an ideal solution individually in complex dynamic environment (Mei et al., 2006). Other than traditional and non-traditional techniques, some of the authors presented their own methods of path planning algorithms. Storer and Reif (1994) claimed that, their practical algorithm for finding minimum length paths reduces considerably the complexity of the problem but it works with only polygonal obstacles and not for curved obstacles (Azariadis and Aspragathosc, 2005). Dougall and Archibald (2006) presented an algorithm which is based on a recursive method for path planning and gives an optimum path, in sparse environments while also providing smooth paths in densely populated environments. However, this algorithm is applicable only for circular obstacles, but not applicable for polygonal obstacle. Raja and Pugazhenthi (2008) developed an algorithm capable of negotiating convex polygonal and curved obstacles with an objective of minimizing traveling distance and computational time in static environment. However, the algorithm is not applicable for concave polygonal obstacles and environment containing moving obstacles.

The literature (Sugihara and Smith, 1997; Shahidi et al., 2004; Gerke, 1999; Gao and Tian, 2007; Zhao and Yan, 2005) shows most of the non-traditional algorithms are applied only for regular convex polygonal obstacles, not for irregular convex, concave polygonal or curved obstacles. The above cited literatures are applicable only for static environment. Moreover, curve obstacles increases the resolution of the grid, ultimately increasing the computation time as most of the evolutionary algorithms makes use of grid as the environment representation. Sipahioglu et al. (2008) proposed real-time tour construction for a mobile robot in a dynamic environment. In this study, a heuristic-based traveling salesman problem is applied along with savings algorithm and Dijsktra's algorithm is used to determine the feasible tour. However, the algorithm does not deal with collision avoidance of moving obstacles. Numerous technologies are being explored to develop a solution for collision avoidance system (Albaker and Rahim, 2011; Nakhaeinia et al., 2011).

Hence, it is worth to investigate new ideas and new directions towards solving the basic motion-planning problem, taking into account moving obstacles. The objective of this work is to develop an efficient path planning algorithm for mobile robot avoiding known obstacles in the environment and yet arriving at the shortest path. In this paper, we introduce the notion of 'Direction concept' and 'Waiting time concept' to resolve the problem of motion planning for a robot moving in a planar terrain. These obstacles are generally considered as prohibited areas, which have arbitrary size and shape (convex or non-convex, polygonal or curve) and are randomly distributed within the robot environment. Given such an environment, 'direction concept' algorithm is used to negotiate static obstacles and 'Waiting time concept' algorithm is used for moving (dynamic) obstacles. Simulation results are presented for static, partially-dynamic and dynamic environments.

Proposed algorithm

Assumptions

The complexity of the general motion planning problem is enormous, so a variety of simplifications have been suggested to reduce the complexity (Azariadis and Aspragathosc, 2005). In this paper, the following assumptions are made:

1. Robot moves on a flat terrain and environment is global (position and velocity of obstacle are known).
2. The given obstacle is enclosed by a rectangular or square boundary and then amplified by a value, taking into consideration of physical dimensions of the robot, and the maneuverable safe distance from the obstacles, without collision as shown in Figure 1. As the obstacle is enclosed by rectangular or square boundary, obstacles of any shape like convex, concave and curved obstacles can be negotiated by the proposed algorithm.

Many obstacles of any irregular shape can be modeled by enclosing rectangle or circle as it reduces the computational burden. Furthermore, this type of approximation is standard in robot collision avoidance literature (Chakravarthy and Ghose, 1998; Chazelle, 1987). Also, it is trivial to allow the envelope of an

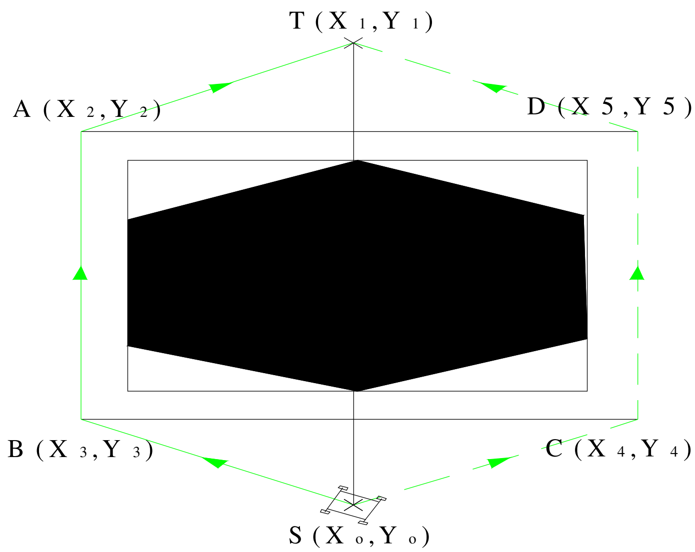

Figure 2. Path negotiation for interfering convex polygonal obstacle.

Table 1. Identification of 'pass through' and 'no pass through' edges.

Line	Assigned sign convention	Determination of 'pass through' and 'no pass through' edges	Inference
SA	Left (-)	Edge AB = SA*SB = +	AB is 'pass through' edge
SB	Left (-)	Edge BC = SB*SC = −	BC is 'no pass through' edge
SC	Right (+)	Edge CD = SC*SD = +	CD is 'pass through' edge
SD	Right (+)	Edge DA = SD*SA = −	DA is 'no pass through' edge

obstacle to be represented by union/intersection of several circles. The envelopes could also be polygonal. Mathematically, circular envelopes can be represented by second order inequalities while polygonal envelopes can be described by first-order linear inequalities (Luh and Liu, 2007).

In real world, a mobile robot may encounter obstacles of any nature such as static, moving (dynamic) or partially dynamic (an environment comprising of both static and dynamic obstacles).

Path negotiation under static environment

To maneuver the robot in the static environment, algorithm commences by creating a straight line from initial position to the target position (the shortest path). To reach the destination, algorithm searches for existence of any obstacle along the shortest path. Based on the obstacle vertex position, 'Irrelevant' obstacle is identified and discarded and 'relevant' (interfering) obstacle is negotiated.

Identification of 'irrelevant' obstacles: If there is an obstacle, it creates a number of lines equal to the number of amplified boundary vertices of the polygonal (for example, line SA, SB, SC and SD in Figure 2). Each vertex position (either left or right direction) of obstacle is defined with respect to line joining (x_o, y_o) and (x_1, y_1). Specifically, an obstacle whose vertex positions are either completely left or right to the shortest path is identified as an 'Irrelevant' obstacle. This analogy can be applied to any obstacle, which is 'irrelevant' to the shortest path. Suppose, if any obstacle obstructs the shortest path, identification of maneuverable and non-maneuverable edges is the major task.

Path negotiation for interfering obstacle: In obstacle-interfering environment, the proposed algorithm segregates the 'pass through' and 'no pass through' edges based on the 'direction concept'. The amplified obstacle edge, which can be maneuvered by the robot, is designated as the 'Pass through' edge and the non-maneuverable edge as the 'No pass through' edge. The 'direction concept' refers to identification of vertices either left $(-)$ or right $(+)$ to the shortest path. These vertices are considered as maneuverable points. An edge is said to be maneuverable if the product of its vertices yields a positive sign, a negative sign identifies a non-maneuverable edge. Joining all the maneuverable edges on the left or on the right gives an obstacle-free path. The shortest path is selected based on the Euclidean distance.

For example, for the polyhedral obstacle as shown in Figure 2 the lines SA and SB are to the left of the shortest path ST and hence are $(-)$ assigned asign. The product of SA and SB yields a $(+)$ sign, which means that the edge AB is maneuverable and hence is the 'Pass through' edge. Similarly, for the rest of the edges 'pass through' and 'no pass through' edges are identified as given in Table 1.

From the Figure 2 and Table 1, two alternate paths through AB

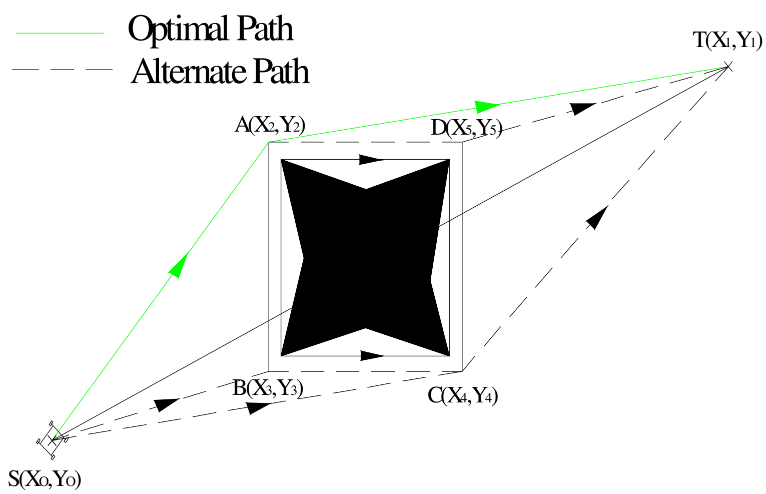

Figure 3. Path negotiation for interfering concave obstacle.

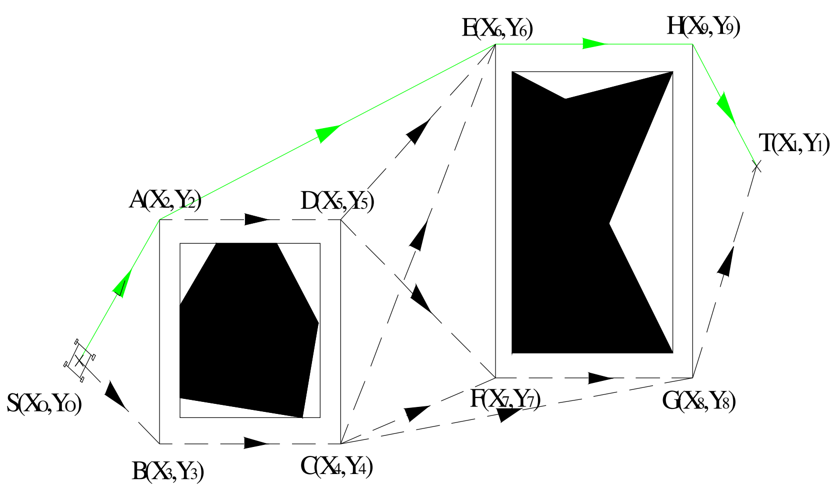

Figure 4. Optimal path for multiple obstacles.

and CD are possible. Euclidean distance (ξ_d) is calculated to the possible paths.

ξ_d along path1 = $\xi_{SB} + \xi_{BA} + \xi_{AT}$

ξ_d along path 2 = $\xi_{SC} + \xi_{CD} + \xi_{DT}$

The shortest path is chosen based on the smallest Euclidean distance. The proposed 'direction concept' algorithm can be applied to any polygonal obstacles for path negotiation as shown in Figure 3. When number of obstacles is more than one, the last maneuvered points of the first obstacle are considered for alternate paths. For example, for the environment shown in Figure 4 the new start points, after the negotiation of first obstacle are $A(x_2,y_2)$, $B(x_3,y_3)$, $C(x_4,y_4)$ and $D(x_5,y_5)$. By using 'direction concept' algorithm, 'pass through' and 'no pass through' vertices and edges are identified. The shortest path $SA - AE - EH - HT$ is selected. Similarly, the algorithm can be extended to any number of polygonal obstacles for path negotiation.

Path negotiation under dynamic environments

Apart from avoiding stationary obstacles, a mobile robot is sometimes required to avoid moving obstacles also. Figure 5 shows an example obstacle with known robot and obstacle velocities and designated start and target locations. To negotiate this moving obstacle following procedure is adopted.

Identification of 'irrelevant' moving obstacles: Step 1- Connect robot start and target points $((x_o, y_o)$ and $(x_1, y_1))$ and obstacle start and target positions $((x_{o11}, y_{o11})$ and $(x_{111}, y_{111}))$ by straight lines respectively.

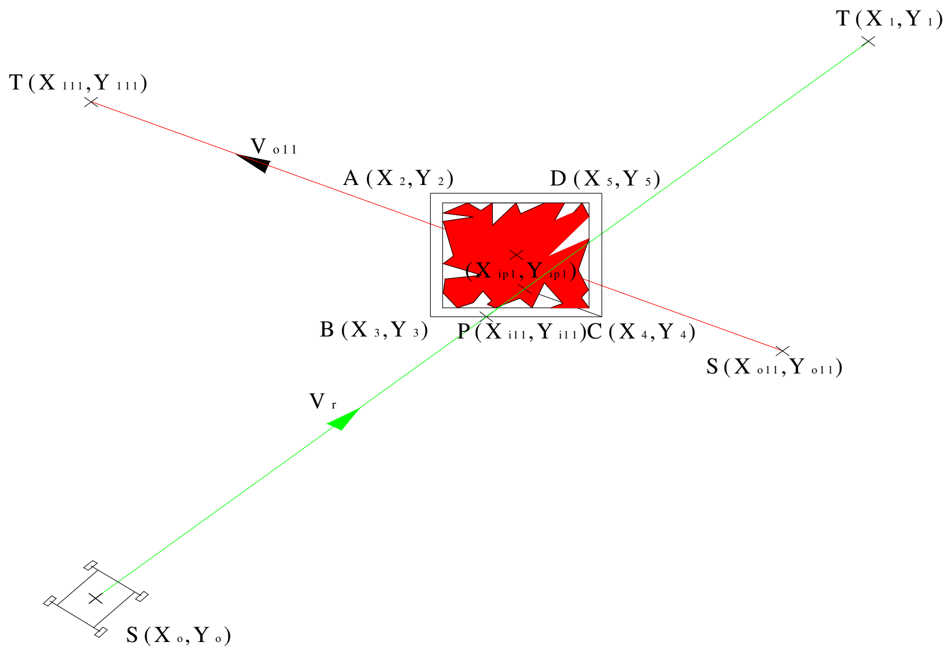

Figure 5. Path negotiation for interfering moving obstacle.

Step 2- Calculate imaginary intersection points of above two lines
Step 3- To reach this intersection point, calculate the time required by the robot
Step 4- At this calculated time, find the position and hence vertices of the obstacle

Depending upon the position of the obstacle with respect to the shortest path an 'irrelevant' moving obstacle is discarded and 'relevant' moving obstacle is negotiated.

To identify and discard the 'irrelevant' obstacles the following condition has to be satisfied, at the time, when the robot reaches imaginary intersection point.

If the vertex position of the obstacle does not interfere with the shortest path line
Then the path is not interfered by the obstacle and the robot can maneuver straight

The analogy can now be applied to any moving obstacle which is 'irrelevant' to the shortest path. In case, if any obstacle obstructs the shortest path, the negotiation of this moving obstacle is the major task. The problem can be solved by 'waiting time concept' algorithm (Figure 5).

Path negotiation for interfering moving obstacle: 'Waiting time concept' algorithm finds the required waiting position and time of the robot till the moving obstacle crosses the robot. The algorithm procedure is explained in the following steps:

Step 5: From $C(x_4, y_4)$ draw a line parallel to the obstacle path $((x_{o11}, y_{o11})$ and $(x_{111}, y_{111}))$.

$C(x_4, y_4)$ is said to be the 'relevant vertex'. The 'relevant vertex' is found by calculating the Euclidean distances of all the vertices of the relevant obstacle with respect to the obstacle's target. The vertex having the longest Euclidean distance is considered as the 'relevant vertex'

Step 6: Find the intersection point (x_{ip1}, y_{ip1}) of line drawn to the shortest path
Step 7: Calculate the corresponding time required and the same will be the waiting time of the robot
Step 8: Find the first intersection point (x_{i11}, y_{i11}) of robot shortest path with current obstacle position and the same will be for the waiting position of the robot
Step 9: Calculate Euclidean distance (ξ_d) and time taken by the robot (T_r) to reach the target.

$$\xi_d = \xi_{SP} + \xi_{PT};$$

$$T_r = T_{SP} + T_{PT};$$

Step 10: End.

The above discussed 'waiting time concept' algorithm can then be applied to any moving obstacles for path negotiation. Similarly, the proposed algorithm can be applied to partially dynamic environment as next explained.

Path negotiation under partially dynamic environment: Partially dynamic environment consists of both moving and stationary obstacles. In such case, the last maneuvered points of the first obstacle are considered as new start points for alternate paths. For example, Figure 6 shows an environment comprising of a moving obstacle with given start and targeted points and a stationary obstacle. At the time when the robot reaches the imaginary intersection point, the position of the moving obstacle in the environment is shown in Figure 6. The new start point, after the

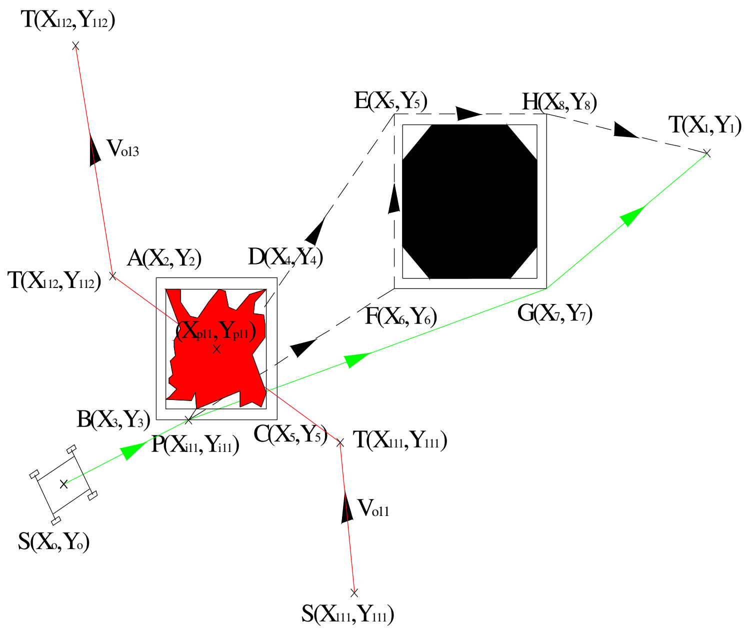

Figure 6. Path negotiation for partially-dynamic obstacles.

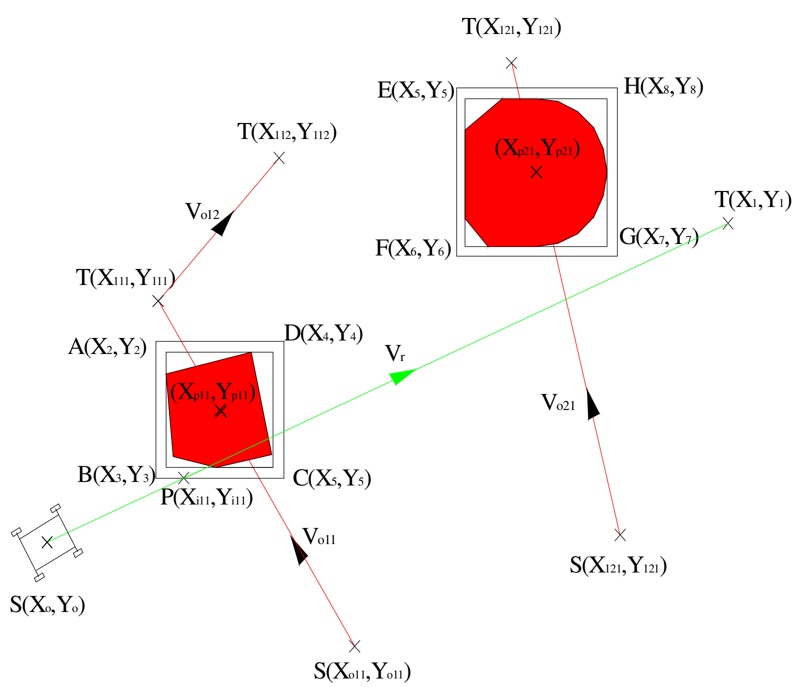

Figure 7. Path negotiation for multiple moving obstacles.

negotiation of first obstacle is $P(x_{i11}, y_{i11})$. 'Irrelevant' obstacle is discarded and 'Relevant' obstacle is negotiated by using 'direction concept' algorithm, by identifying 'pass through' and 'no pass through' vertices and edges. The shortest path $SP - PT - PG - GT$ is selected. Similarly, the algorithm can be extended to any number of for path negotiation.

Path negotiation for multiple moving obstacles: When number of moving obstacles is more than one, the last maneuvered points of the first obstacle are considered as new start points. For example, Figure 7 shows an environment comprising of two moving obstacles with required start and target points. At the time when the robot reaches the imaginary intersection point, the position of the obstacles are as shown in Figure 7. The new start point, after the

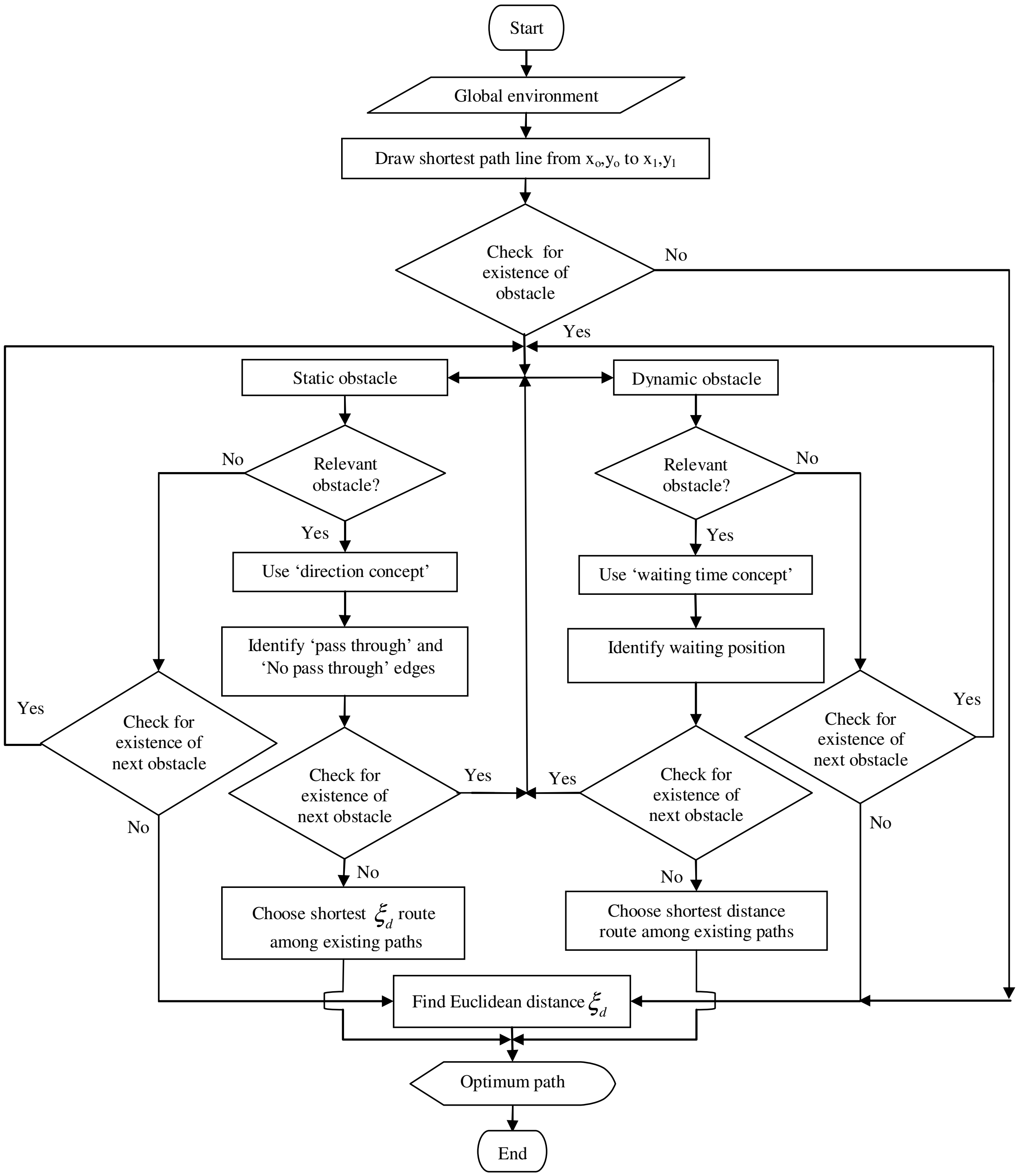

Figure 8. Flowchart of the proposed algorithm for path planning.

negotiation of first obstacle is $P(x_{i11}, y_{i11})$. 'Irrelevant' obstacle is discarded and 'relevant' obstacle is negotiated by using 'waiting time concept' algorithm. The shortest path $SP-PT$ is selected. Similarly, the algorithm can be extended to any number of polygonal obstacles for path negotiation.

The overall algorithm for avoiding both stationary and moving obstacles is depicted through the flowchart given in Figure 8.

The survival of the robot is assured by selecting the waiting position for the robot. However if obstacles inevitably trap a robot, Ge and Cui (2002), recommended following decision making steps:

i) The first step is to keep the robot to wait for the obstacles to change their path. Thus, the waiting method is frequently adopted (which is as explained in this work).

ii) The second step is if subsequent to a definite period's

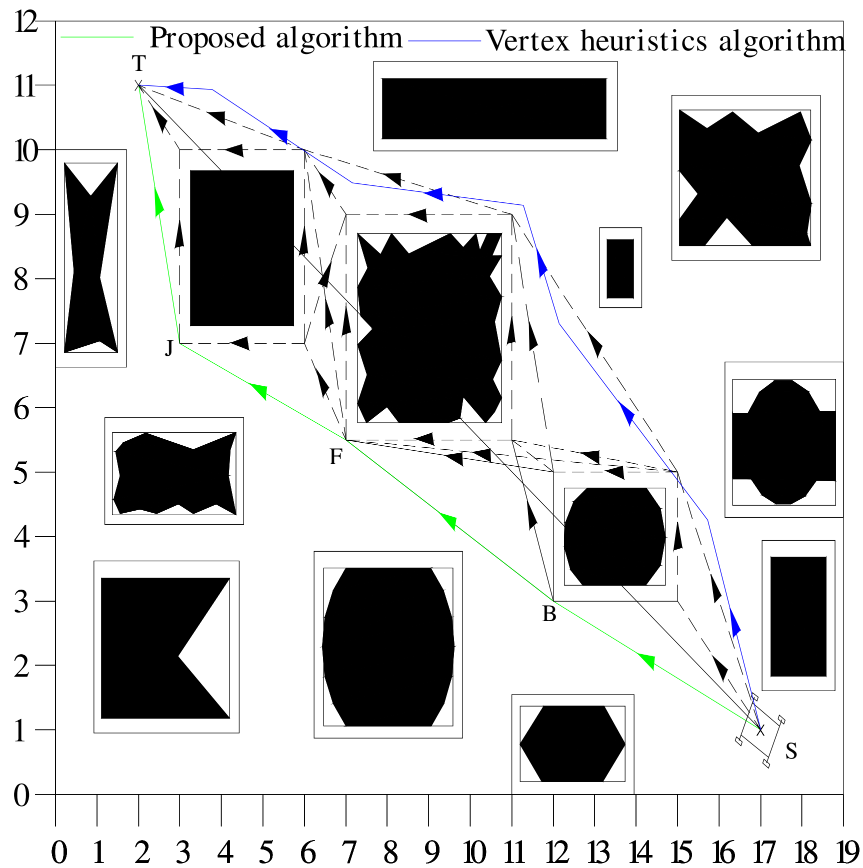

Figure 9. Path planning for static environment.

waiting, the pattern of the obstacles is still unaffected and the robot is still trapped then the robot has to take other methods. Conventional recovery methods such as wall following method (Borenstein and Koren, 1989) can be used. The method considers each instantaneous dynamic situation as static. The method is adopted till the vicinity to the target.

SIMULATION RESULTS

The effectiveness of the proposed algorithm is verified by simulations through a personal computer with Intel core solo 1.86 GHz processor. The test program are written in MATLAB 7.0. Experiments are conducted by considering a terrain of 19×12 as the workspace. A map is created where blocks S and T are assigned as the start and target points respectively. The mobile robot is required to move from S to T while avoiding any obstacles. Experiments are performed in environments of static, partially dynamic and environments. Simulated environments contain convex, concave polygonal and curved obstacles of different size and shape.

The first set of experiments was performed for static obstacles. The environment contains convex, concave polygonal and curved obstacles. Figure 9 shows an environment comprising of thirteen obstacles of different size and shape. Velocity of the robot is assumed to be 0.6 m/s. After identifying 'relevant' and 'irrelevant' obstacles through 'direction concept' algorithm, the alternate valid paths are shown in Figure 9. Table 2 provides path distance and computation time of the algorithm. Shortest path is chosen among the alternate valid paths. For clarity in the figure, the selected shortest path is shown in green.

The second set of experiments was performed for partially dynamic environment. The environment contains concave, convex and curved obstacles of different size and shape. Figure 10 shows an environment comprising of eight obstacles, out of which five are static and three are moving obstacles. Static obstacles are represented by black and moving by red respectively. The desired start and target points, moving obstacles path and speed are shown in the environment. After identifying 'relevant' and 'irrelevant' static and moving obstacles through 'direction concept' and 'waiting time' concept algorithm, the alternate valid paths are shown in Figure 10. The required 'Waiting position' is shown as W. Table 2 provides path distance, time taken to reach the

Table 2. Simulation results for static, partially-dynamic and dynamic environments.

Environment type	Robot positions (units)		Shortest path (m)		Improvement (%)	Shortest time (s)		Improvement (%)	Computation time (ms)		Improvement (%)
	Start	Target	Proposed algorithm	Vertex heuristics		Proposed algorithm	Vertex heuristics		Proposed algorithm	Vertex heuristics	
Static	17,1	2,11	19.37	30.2	35.8	32.28	41	21.2	3.5	29	87.9
Partially dynamic	6,2	17,10.5	14.12	22.8	38	21.76	33	34	4.1	40	89.7
Dynamic	2,2	17,8	16.15	22.4	27.9	29.82	39.5	24.5	6.2	51	87.8

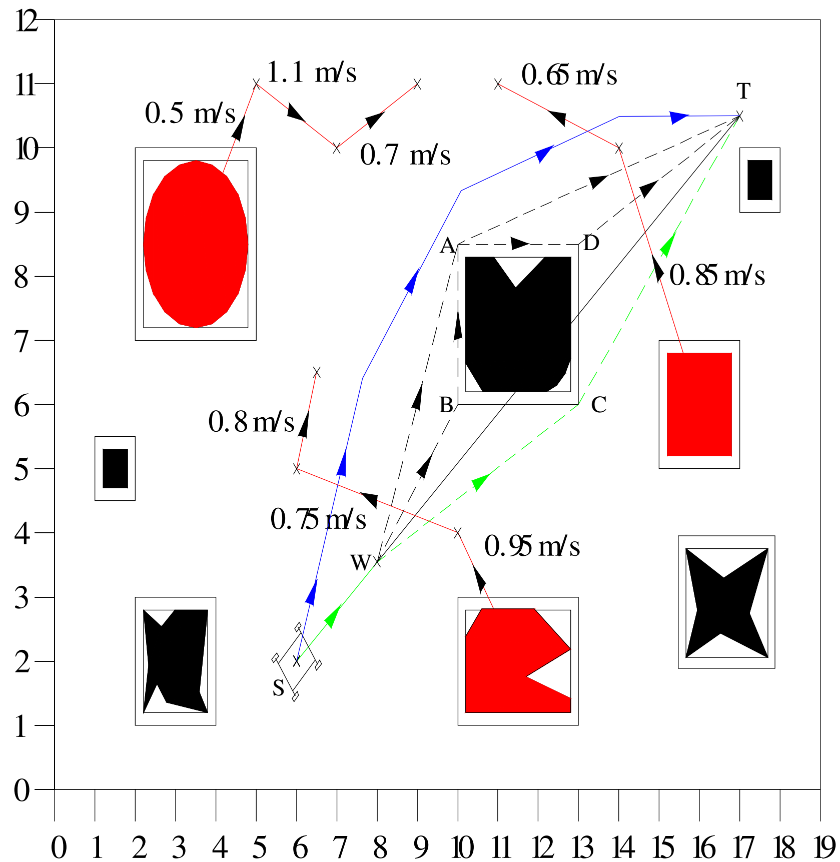

Figure 10. Path planning for partially dynamic environment.

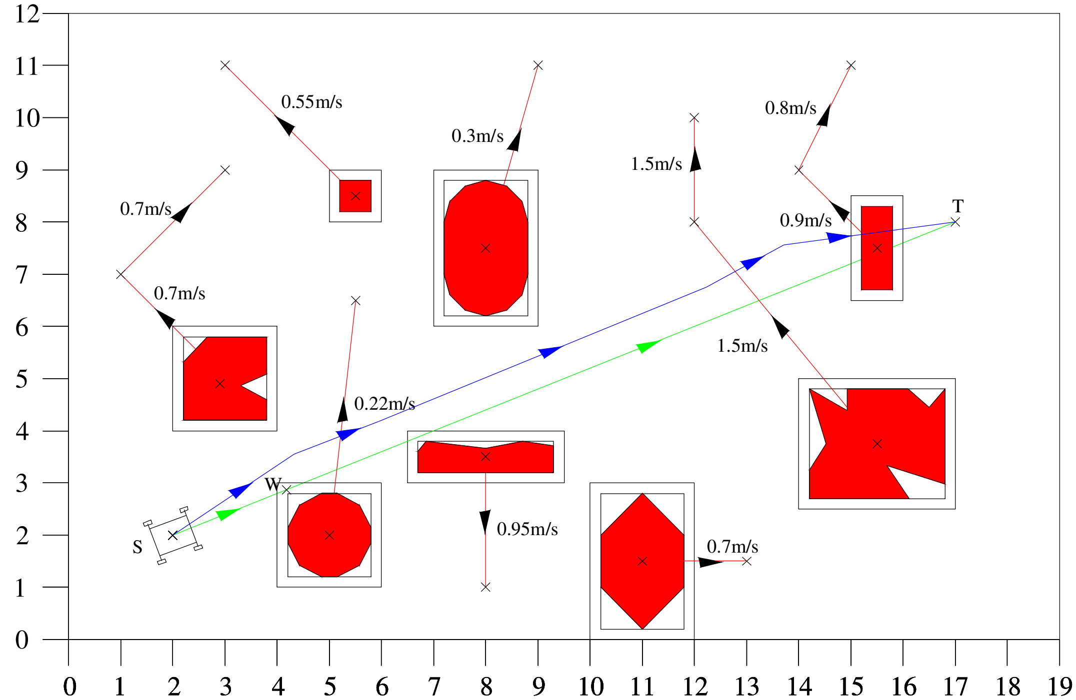

Figure 11. Path planning for dynamic environment.

destination and computation time of the algorithm. For clarity in the Figure 10, the selected shortest path is shown in green.

The third set of experiments was performed for dynamic environment. The environment contains concave, convex and curved obstacles of different size and shape. Figure 11 shows an environment comprising of eight moving obstacles. Moving obstacles are represented by red. The desired start and target points, moving obstacles path and speed are shown in the environment. After identifying 'relevant' and 'irrelevant' moving obstacles through 'waiting time' concept algorithm, the path traversed by the robot is shown in Figure 11. Table 2 provides path distance, time taken to reach the destination and computation time of the algorithm. For clarity in the Figure 11, the selected shortest path is shown in green.

Comparative performance of the algorithm

For comparison and validation purpose, the above three environments discussed previously have been subjected to famous visibility graph based vertex heuristics algorithm (Wang et al., 2007) which make use of complex mathematical model for path evaluation and binary coded genetic algorithm for path optimization. Table 2 compares the results of the proposed algorithm with the vertex heuristics algorithm for static, partially-dynamic and dynamic environments. An average of 33.9, 26.5 and 88.4 improvement in percentage was obtained over vertex heuristics algorithm for shortest path, shortest time and computation time respectively. Also, the proposed algorithm has reduced path segment characteristics (compared to vertex heuristics) which allow easier maneuverability to the mobile robot. By comparison, it is evident that, performance of the proposed algorithm is superior to vertex heuristics algorithm. The shortcomings of the vertex heuristics algorithm can be attributed to the following three reasons:

i) Opening search space is restricted to only few vertices of the amplified obstacles as it make use of fixed length chromosomes.
ii) Infeasible paths (which interferes with the obstacles) are also considered during evaluation, which results in increased execution time.
iii) Utilization of binary encoding and special genetic operators also increase the complexity of the functioning of the algorithm which results in increased path length.

Conclusions

This paper presents an algorithm capable of negotiating obstacles in static, partially dynamic and dynamic environments. Static obstacles are negotiated by using 'direction concept' and moving obstacles by 'waiting time concept' algorithm. Through 'direction concept' algorithm, 'pass through' and 'no pass through' edge concepts are established to identify maneuverable and non-maneuverable obstacle edges. Through 'waiting time concept' algorithm, required waiting position and time, for the robot are found. Unlike the other graphical approaches, only relevant static and moving obstacles are negotiated and irrelevant obstacles are discarded. This leads to the reduced start to destination path distance and computation time of the processor. Unlike the other algorithms cited in the literature the proposed algorithm is applicable for convex as well as concave polygonal obstacles. The algorithm is also valid for curved obstacles. In fact, the algorithm can be applied to environments having any irregular shape obstacles, as the given obstacle is enclosed by the rectangular or square boundary. In addition, the algorithm is applicable to partially dynamic and dynamic environments besides static environment. Also, the performance of the proposed algorithm is found to be both efficient and effective, in terms of shortest path, time and lowest execution time compared to vertex heuristics algorithm.

The present work dealt with path planning for a mobile robot in global environments with known information about obstacles. Essentially a mobile robot needs to switch over to local mode when it discovers new changes in obstacle scenario. Hence, the future scope of this research is to develop a mathematical model considering the parameters of unknown obstacles which are within the sensor range of mobile robot. Further, this work can be extended for environments with multiple robots.

REFERENCES

Albaker BM, Rahim NA (2011). Autonomous unmanned aircraft collision avoidance system based on geometric intersection. Int. J. Phys. Sci., 6(3): 391-401.

Arkin R, Murphy R (1990). Autonomous Navigation in a Manufacturing Environment. IEEE T. Robot. Autom., 6(4): 445-454.

Azariadis PN, Aspragathosc NA (2005). Obstacle representation by Bump-surfaces for optimal motion-planning. Robot. Auton. Syst., 51: 129–150.

Banga VK, Kumar R, Singh Y (2011). Fuzzy-genetic optimal control for robotic systems. Int. J. Phys. Sci., 6(2): 204-212.

Bell JE, McMullen PR (2004). Ant colony optimization techniques for the vehicle routing problem. Adv. Eng. Inf., 18: 41–48.

Bin N, Xiong C, Liming Z, Wendong X (2004). Recurrent Neural Network for Robot Path Planning. Proceedings of the Fifth International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT 2004), LNCS 3320, pp. 188–191.

Borenstein J, Koren Y (1989). Real-time obstacle avoidance for fast mobile robots. IEEE T. Syst. Man Cy., 19(5): 1179-1187.

Boschian V, Pruski A (1993). Grid modeling of robot cells: A memory-efficient approach. J. Intell. Robot. Syst., 8(2): 201-223.

Chakravarthy A, Ghose D (1998). Obstacle avoidance in a dynamic environment: a collision cone approach. IEEE T. Syst. Man Cy. A., 28(5): 562-574.

Chazelle B (1987). Approximation and decomposition of shapes. Adv. Robotics. I: Algorithm. Geom. Aspects Robotics. Schwartz JT, Yap CK, Eds. Hillsdale, NJ: Lawrence Erlbaum, pp. 145–185.

Gao M, Tian J (2007). Path Planning for Mobile Robot Based on Improved Simulated Annealing Artificial Neural Network. Proc. Third Int. Conf. Nat. Comput., 3: 8-12.

Ge SS, Cui YJ (2000). New potential functions for mobile robot path planning. IEEE T. Robot. Autom., 16(5): 615- 620.

Ge SS, Cui YJ (2002). Dynamic motion planning for mobile robots using potential field method. Auton. Robot, 13: 207-222.

Gerke M (1999). Genetic Path Planning for Mobile Robots. Proceedings of the American Control Conference, San Diego, California, pp. 2424-2429.

Jorgen BJ, Gutin G (1979). Digraphs: Theory, Algorithms and Applications, Elsevier North Holland, New York.

Koren Y, Borenstein J (1991). Potential field methods and their inherent limitations for mobile robot navigation. Proceedings of the IEEE Conference on Robotics and Automation, Sacramento, California, pp 1398–1404.

Latombe JC (1991). Robot Motion planning, Kluwer Academic Publisher, Boston.

Li L, Ye T, Tan M (2002). Present state and future development of mobile robot technology research. Robot., 24(5): 475-480 (in Chinese).

Liu S, Tian Y, Liu J (2004). Multi Mobile Robot Path Planning Based on Genetic Algorithm. Proceedings of the 5th World Congress on Intelligent Control and Automation, pp. 4706-4709.

Luh GC, Liu WW (2007). Motion planning for mobile robots in dynamic environments using a potential field immune network. IMechE J. Syst. Control Eng., 221: 1033-1046.

Mei H, Tian Y, Zu L (2006). A Hybrid Ant Colony Optimization Algorithm for Path Planning of Robot in Dynamic Environment. Int. J. Inf. Technol., 12(3): 78-87.

Nakhaeinia D, Tang SH, Mohd Noor SB, Motlagh O (2011). A review of control architectures for autonomous navigation of mobile robots. Int. J. Phys. Sci., 6(2): 169-174.

Raja P, Pugazhenthi S (2008). Path Planning for a Mobile Robot to Avoid Polyhedral and Curved Obstacles. Int. J. Assist. Robot. Mechatron., 9(2): 31-41.

Raja P, Pugazhenthi S (2009). Path Planning for Mobile Robots in Dynamic Environments using particle swarm optimization. IEEE International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom 2009), Kottayam, India, pp. 401-405.

Saska M, Macas M, Preucil L, Lhotska L (2006). Robot Path Planning using Particle Swarm Optimization of Ferguson Splines. Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation, Prague, pp. 833-839.

Shahidi N, Esmaeilzadeh H, Abdollahi M, Lucas C (2004). Memetic Algorithm Based Path Planning for a Mobile Robot. Int. J. Inf. Technol., 1: 174-177.

Sipahioglu A, Yazici A, Parlaktuna O, Gurel U (2008). Real-time tour construction for a mobile robot in a dynamic environment. Robot. Auton. Syst., 56: 289-295.

Storer JA, Reif JH (1994). Shortest paths in the plane with polygonal obstacles. J. Assoc. Comput. Machinery, 41(5): 982–1012.

Sugihara K, Smith J (1997). Genetic algorithms for adaptive motion planning of an autonomous mobile robot. Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation, pp. 138-143.

Wang Y, Sillitoe IPW, Mulvaney DJ (2007). Mobile robot path planning in dynamic environments. Proceedings of IEEE International Conference. on Robotics and Automation, Roma, Italy, pp. 71-76.

Xiong L, Xiao-ping F, Sheng Y, Heng Z (2004). A Novel Genetic Algorithm for Robot Path Planning in Environment Containing Large Numbers of Irregular Obstacles. Robot, p. 26.

Zhao Q, Yan S (2005). Collision-Free Path Planning for Mobile Robots Using Chaotic Particle Swarm Optimization. Proceedings of the first International Conference on Natural Computation, LNCS 3612: 632–635.

Zheng TG, Huan H, Aaron S (2007). Ant Colony System Algorithm for Real-Time Globally Optimal Path Planning of Mobile Robots. Acta. Automatica Sinica, 33(3): 279-285.