J. Basic. Appl. Sci. Res., 2(4)4122-4129, 2012

ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

A new approach for robot Path Planning with Genetic Algorithms

Hamid Yazdani, Ali Fallah, Seyed Mahmod Hoseini

Islamic Azad University, Nour Branch, Iran

ABSTRACT

Path planning has important applications in many areas, for example industrial robotics, autonomous systems, virtual prototyping, and computer-aided drug design. This paper examines robot path planning. This is the task of planning the motionsof a robot so that it avoids collisions with objects in the workspace.The method is applied to a twolink planar robot manipulator using a two dimensional conguration Space representation Parallelism within the method is identified and reductions in execution time are achieved. The basic concept of the Tangent graph is extended to develop a new path planner. The second new planner is developed based around a Genetic Algorithm. Thisprovides a probabilistic approach to the search for a path. This is also demonstrated using a simulation of a three link planar manipulator andagain parallelization issues are discussed. The implications of incorporating the new path planners into a robot system areconsidered. A practical testing is described which integrates a vision systempath planner trajectory planner and manipulator controller.

KEYWORDS: Robot, Path Planning, Genetic Algorithms, Tangent graph.

1- INTRODUCTION

In its most basic form, robot path planning is about finding a collision free motion from one position to another. Efficient algorithms for solving problems of this type have important applications in areas such as: industrial robotics, computer animation, drug design, and automated surveillance. It is therefore not surprising that the research activity in this field has been steadily increasing over the last two decades. In the first part of this paper, we study how object-oriented design methods can be of use in the context of path planning. The result is an object-oriented framework that makes it easy to develop and compare path planning algorithms. Using this framework, two new algorithms have been developed; one for the basic form of the path planning problem, and one aimed for pick-and-place tasks.

Apart from the obvious application areas in industrial robotics and autonomoussystems in general, there are also other useful application areas for path planning. In this section we list some interesting examples togetherwith relevant references.

Industrial and Service Robotics In industrial applications, therobot motion has to be carefully programmed for each new task. Asthis programming can be both laborious and time consuming, there ismuch to win if this process could be made semi-autonomous. That is, apath planning algorithm could be used to give suggestions on collision free motions. The robot programmer could then choose to accept or to modify the generated motions, before making them part of the robot program.

Unlike industrial robots, service robots have to operate in unpredictable and unstructured environments. Such robots are constantly faced with new situations for which there are no preprogrammed motions. Thus, these robots have to plan their own motions. Path planning for service robots are much more difficult due to several reasons. First, the planning has to be sensor-based, implying incomplete and inaccurate world models. Second, real-time constraints means limited resources for planning. Third, due to incomplete models of the environment, planning could involve secondary objectives, with the goal to reduce the uncertainty about the environment. Navigation for mobile robots is closely related to sensor-based path planning in 2D, and can be considered as a mature area of research [1,2]. Sensor-based planning for manipulators, on the other hand, is still a very open research problem. One of the first systems capable of sensor based planning for manipulation was the Handey system [3,4]. Based on laser range data, this system could plan pick-and-place tasks for a manipulator equipped with a parallel-jaw gripper. For two recent examples, see ref. [5], and ref. [6]. Computer Animation There is a growing integration of computer animation with artificial intelligence to create virtual actors [7, 8]. Autonomous or semi-autonomous virtual actors lead to higher productivity as animators can now use high-level commands instead of specifying long sequences of key-frames. Furthermore, with physics-based simulation, a higher degree of realism can be achieved [9]. he type of motion used could be determined from motion capture data, allowing for realistic animations. The techniques used

***Corresponding Author**: Hamid yazdani, Islamic Azad University, Nour Branch, Iran.
Email: eng.hamid.yazdani@gmail.com

in [10] and [11] can also be used in computer games to create more convincing and more skilled computer controlled characters. Virtual Prototyping Virtual prototyping involves modeling and simulation of all important aspects of a prototype, e.g., mechanical design, kinematics, and dynamics, accompanied by realistic visualization. For a mechanical assembly like an engine, one important aspect of the final product is maintainability: How easy is it to reach and replace a certain part in the assembly? Without a physical prototype, such questions can be difficult to answer, and path planning algorithms could thus be a useful tool. For efficient manufacturing, we are interested in assembly strategies that are as efficient as possible.

A somewhat unexpected application area for path planning algorithms is within the fields of computational biology and chemistry. In these fields path planning algorithms have been used to study flexible ligand docking [13], and protein folding pathways [14]. Ligand docking is important in the area of computer-aided drug design, where the goal is to find a small molecule (the ligand) that is able to dock with a target protein (the receptor). A potential docking configuration must not only correspond to a configuration of low potential energy; it must also be accessible to the ligand from an outside location. The found docking configurations were close to the real ones. Protein molecules are long chains of amino acids. These molecule chains will, under normal circumstances, fold themselves into a closepacked, low-energy configuration. This folding process is important to understand for several reasons: The protein's function is determined by its three-dimensional structure, and disturbed protein folding is related to diseases, such as cystic fibrosis and Alzheimer's disease [15]. The folding process is hard to capture experimentally because it happens so quickly [16], and therefore simulation is a necessary tool.

2- MATERIALS AND METHODS

The Genetic Algorithm GA is a probabilistic search process that mimics themechanisms of evolution and natural genetics. The GA operates on a populationof individuals that undergoes a process of simulated evolution. The individualsare each an encoded representation of a potential solution to the search problemthe equivalent of the genetic material of a natural individual At each generationthe most promising or test of the potential solutions are selected and bredtogether to generate new individuals As the GA progresses through severalgenerations the solutions are reined and will ideally converge on the optimal solution. The work described in this chapter is an exploration of the possibilities for usingGAs to solve the Path Planning problem. Many deterministic path plannershave been demonstrated to work affectively in low i.e. two or three dimensionalspaces. There are few planners that can produce reasonable results in high i.e. sixor seven dimensional spaces. Of these the most effective employ some elementof randomness in conjunction with a deterministic search. This research aims to explore the potential of a GA for combining directed search andrandomness in an effective fashion an important feature of GAs as described later is their potential for parallel implementation. It is unlikely that any useful real time path planner is realizable on current single processor systems so the potential parallelism of any method must be considered The GA was rest proposed by Holland in the concepts being brought together in his book Adaptation in Natural and Artificial Systems Another approach to optimization based on the operations of evolution was independently investigated by Scwefel and Rechenberg, These models are commonly termed Evolution Strategies and Evolutionary Programming GAs tend to rely on crossover as the mechanism for information exchange whereas Evolutionary Strategies rely primarily on mutation All these methods and their variations are generally referred to as Evolutionary Computation In this chapter a new method of path planning is proposed which uses a GA to search Cspace and generate solution paths Evolutionary computation and robotics have been brought together by many researchers though the motion planning problem has received relatively little attention Bessier suggests a path planner that uses a local GA to done Manhattan motions to a series of sub goals or landmarks A second GA planner then attempts to connect the landmarks with the goalIn a system with N degrees of freedom.

A Manhattan motionM consists of moving each degree of freedomif successively once by %i This method seems to be effective attending legal paths but the inherent stepped nature of Manhattan motions leads to paths that are not very smooth and far from optimal Page divides the search space into a series of regular cells and employs an EP method to search for paths which are dennedas lists of adjacent cells This requires a tradeofbetween cell size and accuracy small cells that give sufficient resolution to navigate obstacles generate large lists for paths and could lead to jaggeddeviations from more optimal paths Solano searches for paths by progressing through the space in a stepwise fashion using a GA search to select the next step from a circular area surrounding the current location. This method takes a very local view of the search space causing generated paths to move toward local minima and then have to back out before continuing to the goal.

Other researchers have touched on a motion planning aspectMcDonnell and Chen use EP and a GA respectively for configurationoptimization of mobile manipulator taking into account a collision avoidance criterion

Ram uses a GA to develop a reactive behavior for an autonomous vehicle traversing obstaclespaces Davidor applies GAs to trajectory generation searching for inverse kinematic solutions to denned end effectors paths.

A new path planner for a point robot moving among circular obstacles in a two dimensional space is presented_ This is a relatively simple problem that would be most effectively solved by using a tangent graph but it allows the development of GA elements that may then be extended to the general n dimensional Cspace case Figure 1 shows an example of a typical problem. The GA uses a population of potential solution paths each individual representing a path between the start and goal configurations. The GA evaluates paths based on their legality and their length the more promising paths are bred together so that after several generations a reasonable solution path will evolve.

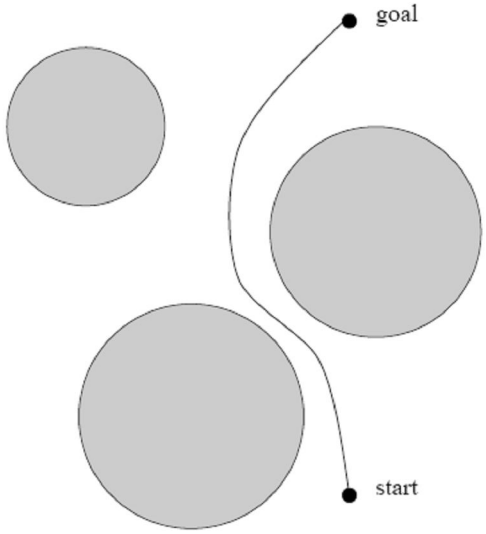


Figure 1: The problem finding a path from start to goal in a space occupiedby circular obstacles

An overview of the path planning GA is given below. It has the same structure as the SGA but with various enhancements which are described in the following section. Genetic algorithms [17] are a class of adaptivemethods that can be used to solve search and optimizationproblems involving large search spaces. The search isperformed using the idea of simulated evolution (survival ofthe fittest). These algorithms maintain and manipulate“generations” of potential solutions or “populations”. Witheach generation, the best solutions (as determined by aproblem specific fitness function) are geneticallymanipulated to form the solution set for the followinggeneration. As in nature, solutions are combined (viacrossover) and/or undergo random mutation.The following are general specifications for our GA-basedlocal path-planning approach:

1) A map of the room in which the path planning takes place is known. The path planner will determine the length and the width of the search space and then apply a grid system to the room, similar to a chessboard. Thus, the room is divided into rows and columns. In our approach we assume the number of rows is equal to the number of columns. The locations of known obstacles are marked as “occupied cells” in the grid.

2) The row and column coordinates of the start-point and the end-point of the desired robot’s movement are also known.

3) The robot is allowed to move on all “free” cells, where the center of the robot moves along an imaginary line from the center of one cell to the center of another cell.

B. Different Types of Robot Movement

Assume a robot is required to navigate from the upper-left corner of a room to the lower-right corner, as shown in Figure 1. In order for the robot to do this task, generally, there are two types of robot movements: Row-Wise and Column-Wise.

B.1) Row-Wised Movement: In a row-based movement, the robot starts moving row by row from the start-point to the end-point. In other words, any horizontal line in the search space will meet the path only once. Therefore, in this movement, the robot always has to go forward and it does not have the capability of going back (up) to the previous row.

B.2) Column-Wised Movement: In a column-based movement, the robot will start moving toward its destination column by column to the right. In other words, any vertical line in the search space will meet the path only once. Therefore, in this movement, the robot always has to move from left to right, and it does not have the capability of moving back to the left.

C. Encoding Technique

The chromosome structure must have sufficient information about the entire path from the start point to the end-point in order to be able to represent it. Next, another member of our research group, [18] modified the genotype by introducing a new instruction flag for each path, called Path-Flag. This Flag instructs the next movement type for each step of the movement. Therefore, this genotype allowed the robot to plan either a row-wise or a column-wise movement according to the search space arrangements. But, neither of these two previous structures was able to combine both row-wise and column-wise paths while planning for a single path. This caused the robot to fail

The mobile robot path planning problem is typically formulated as follows: given a mobile robot and a description of an environment, we need to plan a path between two specified locations, a start and end point. The path should be free of collision and satisfies certain optimization criteria (i.e., shortest path)[19]. According to this definition, path planning problem is categorized as an optimization problem. Researchers distinguish between various methods used to solve the path planning problem according to two factors,

(1) the environment type (i.e., static or dynamic,

2) the path planning algorithms (i.e., global or local). The static environment is defined as the environment which doesn't contain any moving objects other than a navigating robot; while the dynamic is the environment which has dynamic moving objects (i.e., human beings, moving machines and moving robots).

The global path planning algorithms requires a complete knowledge about the search environment and that all terrain should be static. On the other hand, local path planning means that path planning is being implemented while the robot is moving; in other words, the algorithm is capable of producing a new path in response to environmental changes.

Proposed algorithms: To use GAs for solving the path planning problem, we considered number of steps.

These steps are:

First: Convert the search environment to a grid graph (i.e., node). Thus, the robot shall move in a step fashion on the proposed grid as they appear in the real environment

Second: Specify the staring and ending point where the path need to be established

Third: Defining the static obstacles locations on each node of the grid

Path planning technique: In this study, we present the main steps for the proposed path planning techniques.

Initialization: Create an initial population with a predefined population size. The population contains number of individuals (i.e., chromosomes). Each individual represents a solution for the problem under study. In our case, each solution is in fact a path between the start and end point in the search space. The initial population with size n can be presented as:

$$Initial\ Population = < p_1, p_2, \ldots, p_n >$$

Each structure pi is simply an integer string of length L, in general. Each structure pi represents a vector of node numbers in the grid which can take values of 1, 2, …, L (i.e., search space).

Normally, GAs individuals can contain any point value between the starting and ending point. Thus, the individual generated by GAs is in the form of:

$$< c_1, c_2, \ldots, c_l >$$

where, l is the number of visited node in the search space. The starting and ending point will not be shown in this individual. This is why we need to make some modification to the individual structure so that we can add the starting and ending point. The modified individual representation will be:

$$< c_{start}, c_1, c_2, \ldots, c_l, c_{end} >$$

Fitness function: Fitness function represents an important part of any evolutionary process using GAs. Appropriate selection of the fitness function will lead the search towards the optimal solution. The optimal path, in our case, is the shortest path between the starting and ending point. Thus, the fitness function is responsible on finding this path. The shortest path helps computing the total number of steps the mobile robot need to take to reach the ending point. Consequently, the fitness value for a complete solution will be computed as:

$$p_i = d_0 + d_1 + \cdots + d_m$$

$$d_0 = The\ distance\ between\ c_{start}, c_1$$

$$d_1 = The\ distance\ between\ c_1, c_2$$

$$d_2 = The\ distance\ between\ c_2, c_3$$

dm = The distance between $c1, cent$

Fitness computation: To compute the fitness function for an individual, we should have the coordinates of each point in the individual. Thus, we can compute the distance between any two points in the search space (i.e., environment of the robot).

Assume we have two points in the search space $Xcurrent, Xnext$.

Absolute value is important since distance is a quantity value.

Experiments: In this study, we present our development experiments in with two types of environment (i.e., obstacle free and obstacle environment).

Obstacle free environment: We used Genetic Algorithms to search a space of 10□10 and 100□100 nodes to find an optimal path for a mobile robot to move from a start to end points. In our experiment we used an individual structure of 7 elements and 18 elements, respectively. This means that the mobile robot could visit the same point number of times. This is why GAs has to pick up the best path which avoids this type of problems.

We ran GAs with various population sizes 10, 20 and 50, respectively. The goal is to investigate about the behavior of GAs in each case. This will also help in showing that GAs will converge to the optimal solution (i.e., optimal path) in each run.

3- RESULTS AND DISCUSSION

The proposed algorithm has been tested on a wide range of geometries with good results. To give an indication of the required planning time and the type grasps produced by the planner, three examples are presented here. In all examples, the contour was given as a spline curve, which was adaptively approximated by a polygon.For all the examples, only the best grasp is shown, but the output is actually an ordered set of Ng distinct (i.e., well separated thumb positions)grasps. For the examples we used Ng = 10. The timing results were obtained using a Sun blade 100 computer. The time needed for the spline-to-polygon conversion is included in the total planning time.

In the first example, see Figure 1, gravity is directed along the view plane normal. Thus, we are planning a vertical grasp. The height of the cylinder is so large that no grasp hypo paperviolated any depth constraints. Most of the good grasps for this geometry had $\phi \approx 60^{\circ}$, simply because the admissible object weight reaches its theoretical maximum if all three contact forces converge at a single point..7 that the planner also emphasizes grasp robustness: The planned grasp can tolerate relative large perturbations without losing its stability. The polygon curve has 174 vertices and the required planning time was 0.38 seconds. The second example is a vertical grasp on an eccentric ellipse. Here the strong eccentricity of the object (e = 16) is used to initially bias the thumb position and the spread angle such that the grasps wrap around the minor axis of the object. The best grasp is centered over the centroid and achieves a large contact area between the palm and the object, thereby making it a very secure grasp. The polygon curve has 60 vertices and the required planning time was 0.24 seconds.

The last example is a horizontal grasp, with gravity directed downwards in the view plane. As the intent was to implement this planner on a (research) service robot, we chose an object more appropriate to that context, namely an iron. Because the iron is resting on a support plane, a constraint box is put around its lower parts, which can be seen as the dash-dot box. This is a much harder problem compared to the previous ones because much of the contour is not accessible. Obviously one or two fingers5 must get under the handle of the iron, whichis not easy considering the minimum clearance needed for the fingers,

Even though the algorithm assumes cylindrical objects, this is an example where the planner does well even on objects that do not fulfill this assumption. We can see that the palm has contact with the object. If the real robot would execute this grasp, the hand would be moved forward until the tactile sensor in the palm senses a contact. When closing the fingers, they would, due to the clutch mechanism, wrap around the handle and secure the object. This is an additional reason why grasps with palm contacts are preferred by the planner.

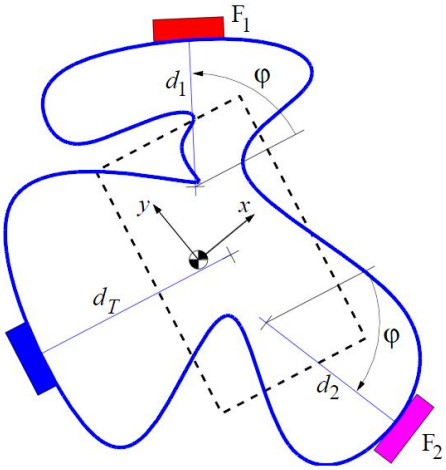


Figure 2: The polygon has 174 vertices and required planning time was 0.38 seconds.

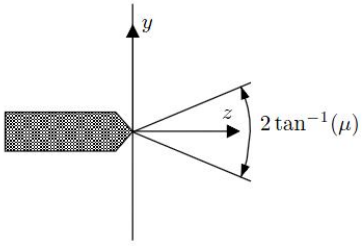
It is assumed that the grasped object is rigid and that the grasp consists of any number of point contacts with friction. The point contact assumption might seem limiting but, as was pointed out by Nguyen [18], any planar polygonal contact can be represented as the convex sum of point contacts placed at the vertices of the contact polygon. Attached to the object is a reference frame, to which all contacts and forces are related.

Each contact will have its own reference frame, with the z-axis pointing in the direction of the inward surface normal, see Figure 3 (a). Because of friction being present, the contact force can deviate from the z-axis. If the contact forces obey the Coulomb friction model, then the space of all admissible contact forces forms a circular cone with opening angle $2\tan-1(\mu)$, where $\mu$is the coefficient of friction. This cone, called the friction cone, will impose nonlinear constraints on the contact force components.

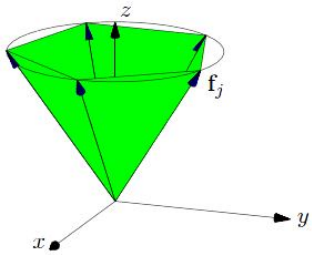
In literature, the circular friction cone is often approximated with an n-sided pyramid, see Figure 3 (b). By doing this, we can write the contact force as a positive linear combination of the force vectors spanning the pyramid:

$$f = \sum \alpha_j f_j, \alpha_j \geq 0.$$

Note that by choosing the vectors $\{f_j\}$to have unit z-component, the normal component of the contact force is easily obtained as$\sum \alpha_j$



(a) Point contact



(b) Friction cone approximation

Figure 3: For nonslipping contacts that obey the Coulomb friction model, the contact forces must be inside the friction cone. (a) A side view of a point contact together with its coordinate system. (b) An example of a friction cone approximated by a five-sided pyramid.

It is often convenient to concatenate force and torque vectors, F and T, into a wrench, defined as $W = (F_T, T_T)_T$. A wrench is thus a six-dimensional column vector. Each force $f_j$will result in an object wrench , which can be computedif the position and the orientation of the contact relative the object frame is known. Let the $w_j$from all contacts be the columns of a $6 \times mn$matrix G, where $m$is the number of contacts. This matrix is called the grasp matrix. Summing up the contributions from all contacts, the total wrench exerted by the grasp on the object, W, can be written as

$$W = Gx, x_k \geq 0, k = 1, \ldots, mn$$

wherex is a vector containing the $\alpha_j$for all contacts. See the book by Murray et al. [19] for more details on how to construct the grasp matrix. When analyzing a grasp, it is of interest to know the space of wrenches that can be applied to the object by the grasp.

This space is equal to the convex hull of G, which can be efficiently computed using the Quickhull algorithm, see Barber et al. [12].An important class of grasps are those that have force closure. This means that the grasp can counteract any external wrench acting on the body by adjusting the contact forces properly. If a grasp has force closure, then the convex hull of G must contain a neighborhood of the origin [107]. The converse is also true: If the convex hull contains a neighborhood of the origin, then the grasp has force closure.

4- Conclusion

This paper investigates path planning strategies for repeated traversal in largedynamic partially unknown environments. The aim of the approach was tominimize collision risk and speed up the mission by adapting to the changes in the dynamic environment.

The advantages of the novel path selection algorithm for generating innovative paths between predefined target points are demonstrated. Over 600 test runs are conducted using the research robot Khepera (all descriptions of the experiments and experimental results are represented additionally.the behaviour of the robot is verified against the shortest path following strategy in various complex environments (varying from static to dynamic as well as from unknown to partially and totally known).

The experimental results lead to the following general conclusions. Path planning approach presented in this paper can be used even if very little is known about the environment or when the environment is completely restructured during the mission. The path selection algorithm will efficiently cover the whole space even if the environment is large. This approach helps to reduce time, risk of collisions and increases the predictability of robot's behaviour.

To optimize travel time, distance, energy consumption, collision risk or deviation from the original path, unexpected events should be decreased as changes in the former parameters depend on the last one.

In an uncertain environment the trajectory of the robot is very difficult to predict and control because the deviation from the planned path is weakly correlated to the accuracy of the world-model.

Optimal (shortest) path planning is not a relevant problem in partially unknown environments. The behaviour of the robot is influenced by the knowledge it has about the environment but does not depend on the path planning strategy. In order to increase the reliability of mobile robot applications, much more attention should be paid on modelling the environment and its changes than an optimisation of path planning algorithms.

Gaining as accurate as possible knowledge about the surrounding is not necessary beneficial if the mission time is limited in a large hazardous environment. Mission-oriented exploration heuristics could be considered in mobile robot applications that are time-critical, where the robot is operating in a large unknown environment and when this environment is dangerous.

REFERENCES

1. A. Elshamli, H. A. Abdullah and S. Areibi, "Genetic algorithm for dynamic path planning". In: Proc. Canadian Conf. Elect. andComput. Eng., Niagara Falls, Vol.2, May 2-5, 2004, pp: 677-680,

2. Y. Davidor, "Robot programming with a genetic algorithm,".In Proc. 1999 IEEE Int. Conf. Comput. Syst. and Software Eng., Tel Aviv, Israel, pp: 186-191.

3. K. H. Sedighi, K. Ashenayi, T. W. Manikas, R. L. Wainwright, and H. M. Tai, "Autonomous Local Path Planning for a Mobile Robot Using a Genetic Algorithm," in Congr. Evol.Comput. 2004. CEC2004., Vol. 2, pp. 1338-1345.

4. A. Sheta and H. Turabieh, (2006, January). A comparison between genetic algorithms and sequential quadratic programming in solving constrained optimization problems.The Int. J. of Artificial Intell. and Mach. Learning 6(1), pp. 67-74.

5. A. Ramirez-Serrano, H. Liu and G. C. Pettinaro.(2008). Mobile robot localization in quazi-dynamic environments.Int. J. on Ind. Robot, pp.246-258.

6. B. Graf, J. M. HostaletWandosell, "Flexible path planning for nonholonomic mobile robots,". In Proc. 4th European workshop on advanced MobilemRobots (EUROBOT'01).,Fraunhofer Inst. Manufact. Eng. Automat.(IPS). Lund, Sweden, Sept. 19-21, 2001, pp. 199-206.

7. J. C. Latombe, Robot Motion Planning. 1st ed, Boston, UK, MA: Kluwer Academic Publishers (Published by Springer), 1991, ISBN: 0-7923- 9129-2.

8. T. Lozano-Perez and M. Wesely, (1979, October).An algorithm for planning collision-Free paths among polyhedral obstacles.Commun. Of the ACM [Online], 22(10), pp. 560-570.

9. D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning. 1st ed. ISBN: 0-201-15767-5, Boston, MA: Addison-Wesley Longman publishing Co., Inc., 1989.

10. C. E. Thomas, M. A. C. Pacheco, M.M. and B.R. Vellasco, "Mobile Robot Path Planning Using Genetic Algorithms,". In foundations and tools for neural modeling, vol. 1606/1999, Springer Berlin/ Heidelberg, ISBN: 3-540-66069-0, 1999, pp. 671- 679.

11. J. Brank, "Evolutionary approaches to dynamic optimization problems-introduction and recent trends," [Online] In Proc. GECCO Workshop on Evol. Algorithms for Dynamic Optimization Problems, Chicago, USA, 2003, pp. 2

12. J. Lu and D. Yang, "Path planning based on double-layer genetic algorithm," in 3rd Int. Conf. on Natural Computation (ICNC 2007) [Online], Hangzhou, Haikou, China, Aug. 24-27, 2007, Vol. 4, pp: 357-361.

13. A. Ramirez-Serrano and M. Boumedine, "Real time navigation in unknown environments using fuzzy logic and ultrasonic sensing," in Proc. 1996 IEEE Int. Symp.In Intell. Control, Dearborn, MI, ISBN: 0-7803-2978-3, Y de Estudiossupeiores de Monterrey, 15-18 Sept, 1996, pp: 26-30, DOI: 10.1109/ISIC.1996.556172.

14. I.K. Jung, K.B. Hong, S.K. Hong and S.C. Hong, "Path planning of mobile robot using neural network," in Proc. IEEE Int. Symp. on Ind. Electron,. Sch. Elect. Eng., 1999 ISIE Bled Slovenia, Vol. 3, 12-16 July, 1999, pp. 979-983, DOI: 10.1109/ISIE,1999.796750.

15. J.J. Grefenstette, "A User's guide to GENESIS," Technical Report, CS-84-11, Department of Computer Science, Vanderbilt University, Nashville, TN.

16. A. Zelinsky. Using Path Transforms to Guide the Search for Findpath in 2D. *The Int. Journal of Robotics Research,* 3(4):315-325, August 1994.

17. A.Elfes. Using occupancy grids for mobile robot perception and navigation.*IEEE Computer*, 22(6):46-57, 1989.

18. A.Howard, H.Seraji. Vision-Based Terrain Characterization and Traversability Assessment.*Journal of Robotic Systems*, 18(10):577-587, Wiley periodicals, 2001.