Genetic Algorithm and Time Petri net are new Approaches for Path Planning and Co-ordination among Sub-modules in a Mobile Robot

Hussein A. Lafta **Hamed K. Shemran**
Department of Computer Science, College of Sciences for Women, University of Babylon
Ali Abass
College of Engineering , University of Babylon

Abstract

A mobile robot has to generate a navigational plan in a given environment between pre-defined starting and goal points. The robot's environment includes many obstacles and thus finding the shortest path without touching the obstacles. In many cases is an extremely complex problem. The complexity of the problem is increases further, when the obstacles are dynamic and there exits many spatial constraints for the movement of the robot. This search considers the problems of navigational planning and Co-ordination among Sub-modules in a mobile robot for dynamic environments. Genetic Algorithms are used for path planning optimization and Time Petri nets are used for co-ordination among Sub-modules in a Mobile Robot.

Key words: Genetic, Petri net, Mobile, Robot, Optimization.

الخلاصة

يستخدم جوال الروبوت لتوليد خطة ملاحة في بيئة معينة تم تحديدها مسبقا بين نقاط البدء ونقطة الهدف. تحتوي بيئة الروبوت على العديد من العوائق وبالتالي علينا إيجاد اقصر طريق دون لمس تلك العوائق. نظهر تلك المشكلة في كثير من الحالات المعقدة. وتعقد المشكلة يزداد عندما تكون العوائق متحركة ووجود تقيدات على حركة الربوت.يعالج البحث مشاكل تخطيط الملاحة و تنظيم السير بين الموديلات الجزئية لربوت متحرك.تستخدم الخوارزميات الجينية لتحسين تخطيط المسار.و تستخدم شبكات بتري المتزامنة لعمليات التنظيم بين الموديلات الجزئية للربوت.

Introduction

The elementary model of cognition (Matlin,, 1996) includes three main cycles. Among these, the sensing action cycle. This cycle inputs the location of the obstacles and subsequently generates the control commands for the motors to set them in motion. The second cycle passes through perception and planning states of cognition, while the third includes all possible states including sensing, acquisition, perception, planning and action (Hutchinson and Kak, 1989). Sensing here is done by ultrasonic sensors/camera or by both. The sensed range signals or images are saved for multi-sensory data fusion. The fusion of multi-sensory data is generally carried out by dempster-shafer theory, Bayesian belief networks [Pearl, 1986; Krebs, 1998) and kalman filtering (Nicholas , 1991; Brown, 1997). A robot can also construct high-level knowledge from relatively lower level data and knowledge. This is referred to as perception. Construction of knowledge about its environment from range data and gray images helps it in generating its plan for action. For instance, a robot can plan its trajectory, if it knows its local environment. The local range information may be combined to construct global information about its world. There exists ample scope of building perception for robots, at very little of it could have been realized so far in practical systems.

The task planning by a robot becomes easier, when it has the requisite knowledge about its world. For example, determining the trajectory in a known world map is simpler, in contrast to an unknown environment. In an unknown environment, the robot has to execute only the sensing-action cycle. After the navigational planning is completed, the remaining task is to generate control commands to the actuators and final control elements for the execution of the plans (Borenstain *et al.*,1996)

Among the five states of cognition cycle, AI is required mainly in perception and planning. In this search, we thus mainly concentrate on planning and co-ordination of a robot. It may be added here that the term ‘co-ordination’ refers to co-ordination among sensors and arms of a robot or it may equally include the co-ordination of multiple robots in complex task planning problems.

Co-ordination among Sub-modules in a Mobile Robot

The co-ordination in a mobile robot is required for autonomously handing a set of concurrent as well sequential tasks, pertaining to sensing, motion and control of various subsystems within the robot. A number of models of coordination are available. The finite state machine and time Petri net are few popular models, the time Petri net model for the coordination is presented in this search.

Petri nets are graphical model to study parallism, deadlock management, tokens movement in a data flow graph and reasoning in knowledge base system(Peterson , 1981; Murata, 1989).Currently Petri nests are gaining importance in modeling coordination of a multitasking in mobile robots (Freedman , 1991). .Ideal Petri net suffers from conflict problem. In figure (1) the transition t1 and t2 could fire if, each of them has a token at all its input places. However if t1 fires earlier than t2 then according to the nation of Petri nets p4 will receive one new token and p1 and p2 will lose their tokens. Transition t2 cannot fire, as p2 does not possess a token. Thus, either of these two transitions can fire only in this system. However, none of the transition t1 and t2 will fire unless no additional condition is imposed regarding the priority of the firing of the transition. This problem of waiting in a loop is often referred to as time conflict in Petri nets theory ((Borenstain *et al.*,1996; Patnaik and Konar,1998]. One way to restore this conflict is to match time with the tokens and firing delay with the transitions. Such Petri net are called time Petri nets. Figure (2) describes a time Petri net (Peterson, 1981), where the semantic of the places are presented in the figure itself. The firing conflict in a Petri net with transitions like t1 and t2 of figure (1) is a void by using the following strategy.

If the arrival time of the latest token in the input place of one transition + its firing delay < the arrival time of the latest token in the input place of the second transition (with common place +its firing delay then the former will fire.

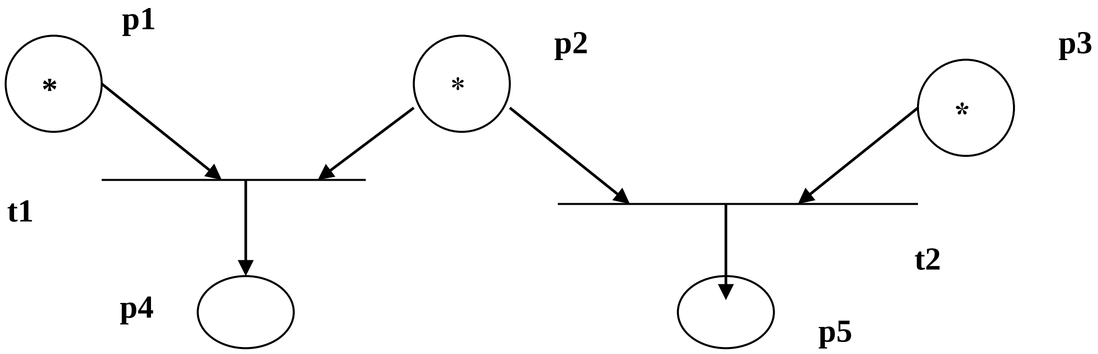

Figure (1): A Peri net model

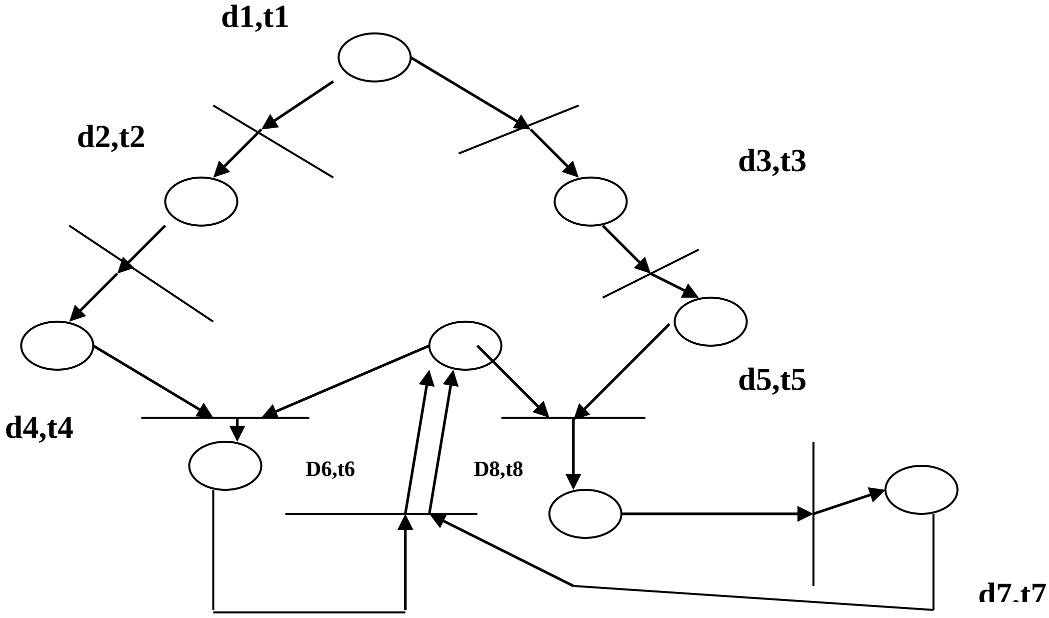

Figure (2): Petri net showing the coordination of multitasking

d1, t1= sense surrounding by sensors and camera at time t1;
d2, t2=build local map of surrounding from sonar sensor and compare with world map at time t2;
d3, t3=image processing and matching with image database at time t3;
d4, t4=plan for obstacle free path at time t4;
d5, t5=object is recognized at time t5;
d6, t6=start from node *ni* at time t6;
d7, t7=stop at node *ni+1* at time t7;
d8, t8=start movement to keep the tool in pre specified place at time t8;
d9, t9=place the tool at proper place at time t9.

Proposed Genetic Algorithm for Path planning and optimization

This algorithm is used for path planning optimization in a dynamic environment. It helps in selection of the next point and the path up to that point only in one genetic evolution. This is extremely fast and can take care of movable obstacles of speed comparable to the robot (Goldberg , 1989).

The first step is to set up the initial population. For this purpose, instead of taking random coordinates, we have taken the sensor information into account and the coordinates obtained from those sensors are used to set up the initial population. With this modification it is assured that all the initial population are feasible, in the sense they are obstacle-free points and the straight line paths between the starting point and the selected next points are obstacles free (Michalewicz, 1986).

Since in one genetic iteration, we plan the path up to the selected next point, the data structure to represent the chromosome becomes extremely simple as presented in figure (3)

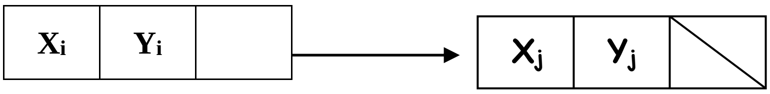

Figure (3): Representation of the chromosome in our simulation.

Here (Xi,Yi) is the starting point and (Xj,Yj) is the one of the two dimensions points , obtained from the sensor information .All these chromosomes form the initial population .The next step is to allow crossover among the initial population .But what about the crossover point? If we choose the cross-site randomly, it is observed that most of the off springs generated in the process of crossover are not feasible, as those paths may fall outside the two dimensions workspace. Therefore, instead of binary crossover, we employed integer crossover. The crossover process is shown below. .Consider the two chromosomes as shown in figure (4) and the crossover point is set between the third and the fourth integer for every chromosome (Krebs, 1998).

After making crossover between all pairs of initial population, we will get the new population.For the new population we will find the feasibility, i.e., they are reachable from the starting point by the straight-line path or not. The next step of the algorithm is making the mutation .This will make fine-tuning of the path, such as avoiding the sharp turns. In this process, we select a binary bit randomly on the bit stream of the sensor coordinates and alter that binary bit value, such that the feasibility should not be lost for that chromosome(Michalewicz, 1986; Filho , 1998).

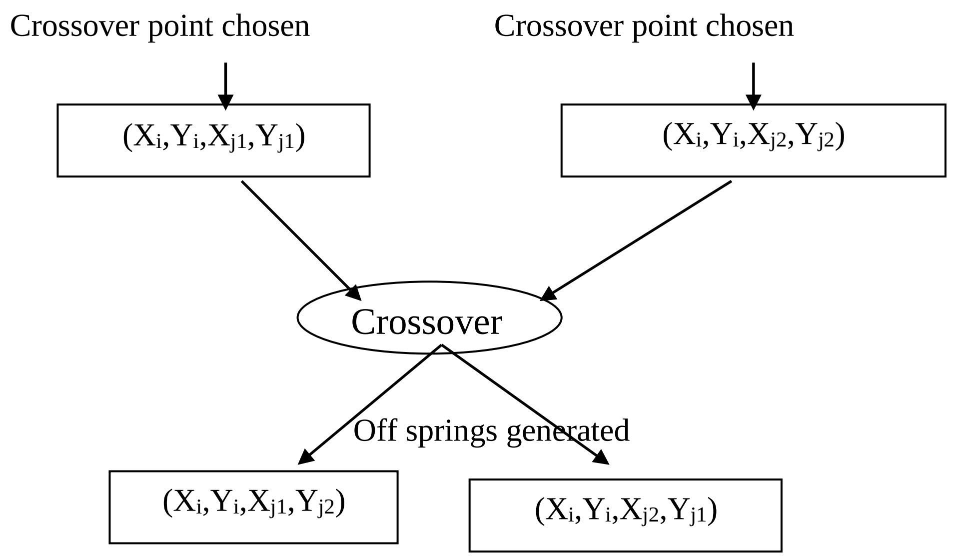

Figure (4): The crossover operation used in the proposed algorithm

Our next task is estimating the fitness used of each chromosome of the total present population (both for the initial and new populations). Calculation of the fitness involves finding the sum of the straight line distance from the starting point (Xi,Yi) to the coordinate (Xj1,Yj1) obtained from the sensor information and the distance from (Xj1,Yj1) to the goal point (Xg,Yg).

Fitness of a chromosome (Xi,Yi,Xj1,Yj1)=
1/{(distance between (Xi,Yi)and (Xj1,Yj1))+
(distance between (Xj1,Yj1)and(Xg,Yg))}

After finding the fitness value of the chromosomes, we will evaluate the best fit chromosome,i.e., for which the fitness is the best .In this case ,the best fit chromosome represents the predicted shortest path from the starting point to the goal . We restrict the number of generations to one, since in the first generation itself we are getting a near optimal intermediate point to move. That third and fourth integer field of the best fit chromosome will become the next intermediate point to move .Then we update the starting point with this better point and the whole process of the genetic algorithm, from setting up the initial population, is repeated, until the best fit chromosome will have its third and the fourth field equal to the x-and y- coordinates of the goal location [Nicholas, 1991; Patnaik and Konar, 1998).
The algorithm is now formally presented below.

Procedure GA-path
// (xi,yi) = starting point ;
(xg,yg)= goal point; //
Add-pathlist (xi,yi);
Repeat
i) Initialization:
Get sensor information in all possible directions
(xj1,yj1) , (xj2,yj2) , …. (xjn,yjn).
Form chromosomes like (xi,yi,xj,yj) ;
ii) Crossover:
Select crossover point randomly on the third and the fourth field of the chromosome.

Allow crossover between all chromosomes and get new population as
(xi,yi,xj1,yj1),(xi,yi,xj2,yj2),(xi,yi,$xj1^{i}$,$yj1^{i}$),(xi,yi,$xj2^{ii}$,$yj2^{ii}$);
iii) Mutation:
Select a mutation point in bit stream randomly and
Complement that bit position for every chromosome.
iv) Selection:
Discard all chromosomes (xi,yi,xj,yj) from population
Whose line segment is on obstacle region
For all chromosomes in population, find fitness using
Fitness (xi,yi,xj,yj)=1/($(xj-xi)^2$ +(yj-yi)+(xg-xj)+$(yg-yj)^2$);
Identify the best fit chromosome (xi,yi,xbf,ybf);
Add-pathlist (xbf,ybf);
Xi=xbf ; Yi=ybf ;
EndFor,
Until (xi=xg) and (yi=yg) ;
End.

A samples execution of the above algorithm with the details of how to execute the algorithm is presented in figures (5, 6, 7, 8, 9). The GA based evolution scheme can identify a path easily, when the obstacles are convex-shaped .For concave shaped obstacles, the evolution has to be continued quite a large number of iterations to find the goal point. This ,of course ,is not the limitation of the algorithm .It happens so for all GA-based path planning .To overcome this problem ,one simple way is to follow the boundary of the concave obstacles , and when there is a 90 degree turn ,again start the evolution . Thus, GA may be used in an interleaved manner .We tested and found this to be satisfactory in most cases.

Implementation of the work
One-The workspace coordinates are (50, 50), and (450, 450)
Enter starting x (50-450):70
Enter starting y (50-450):70
Enter sensing range: 30
Enter robot step size (5 to 30):20

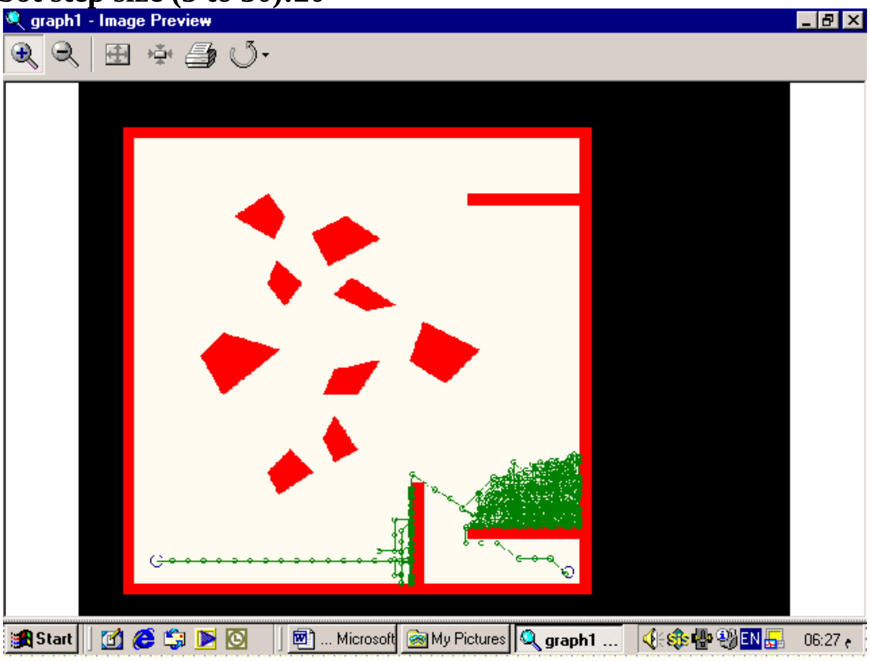

Figure (5): Computer simulation of the proposed GA and Time Petri Net
Time taken to search path=2.747253
Distance covered=6475.417165 units

Two-Workspace coordinate range is between (80, 80) and (400,400).
The obstacles are lying with in the following coordinates i.e. coordinates of diagonal of the square region (x_1, y_1) to (x_2, y_2).
Obstacle 1: (80, 80) to (120,120)
Obstacle 2: (200,120) to (240,160)
Obstacle 3: (240,160) to (320,240)
Obstacle 4: (١٦٠,240) to (240,320)
You have to enter the (x, y) coordinate of the starting position, of the robot within the workspace and should not lie on the obstacle zone
Set goal x-location: 380
Set goal y-location: 380
Set starting x-location: 130
Set starting y-location: 130

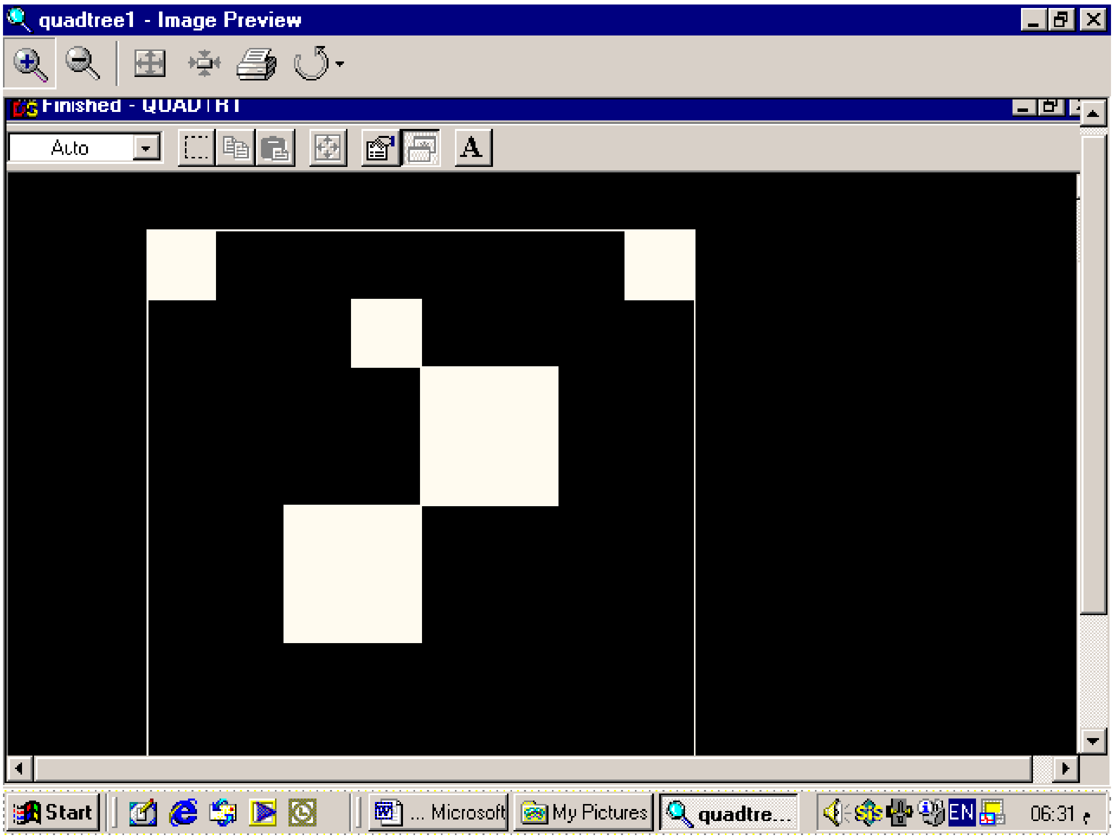

Figure (6): Computer simulation for workspace two
Current node: 120,120 and 160,160

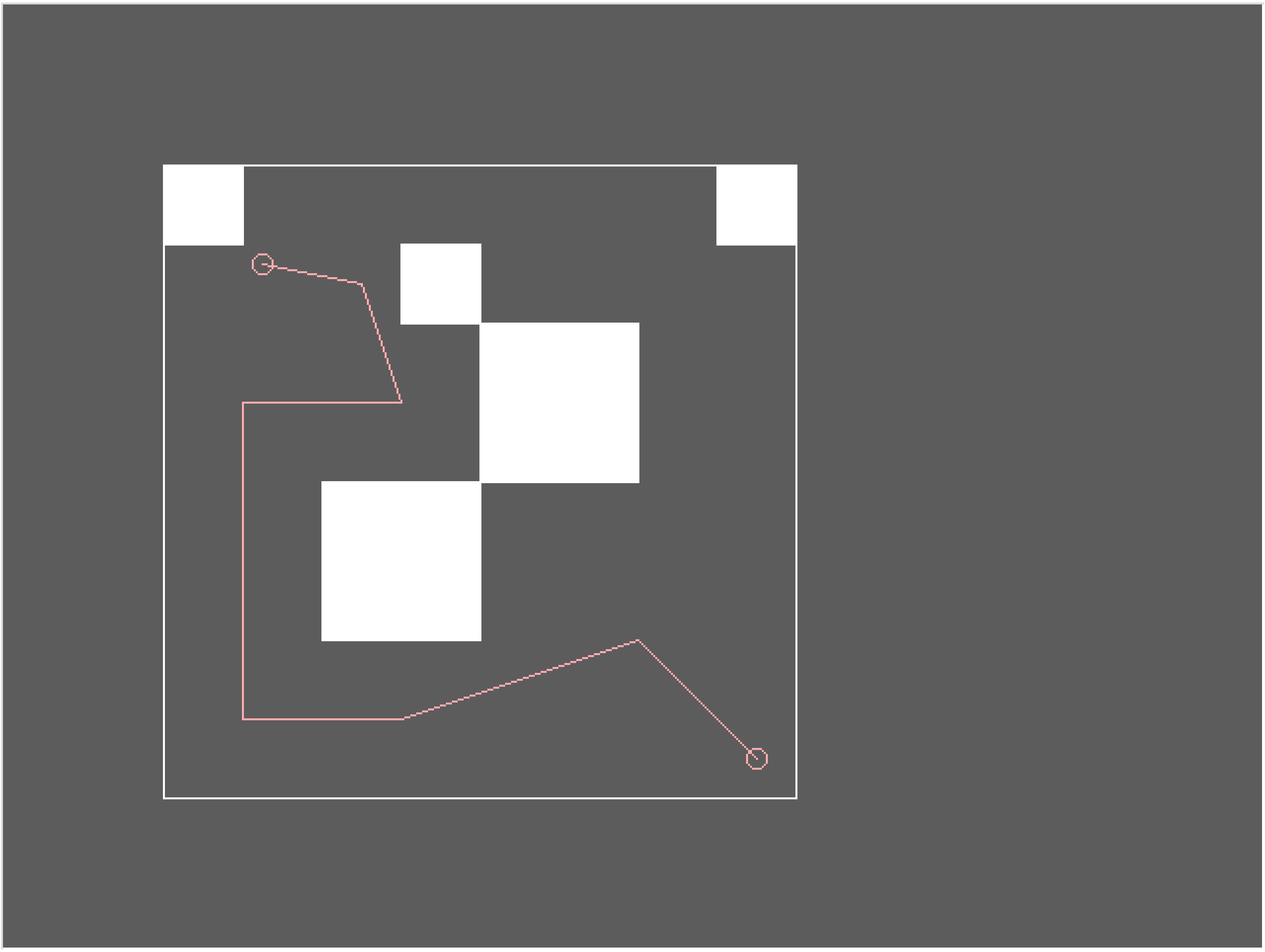

Figure (7): Computer simulation for path node (240,240) and(400,400)

Three-This program builds a map of the obstacles and the room boundaries by first determine the room boundary and then employing the priority of robot movement.

Workspace is rectangular and co-ordinates are (85, 85),(85,359),(395,85) and (395,395)

Starting position of robot x(enter value between 85-395):100

Starting position of robot y(enter value between 85-395):380

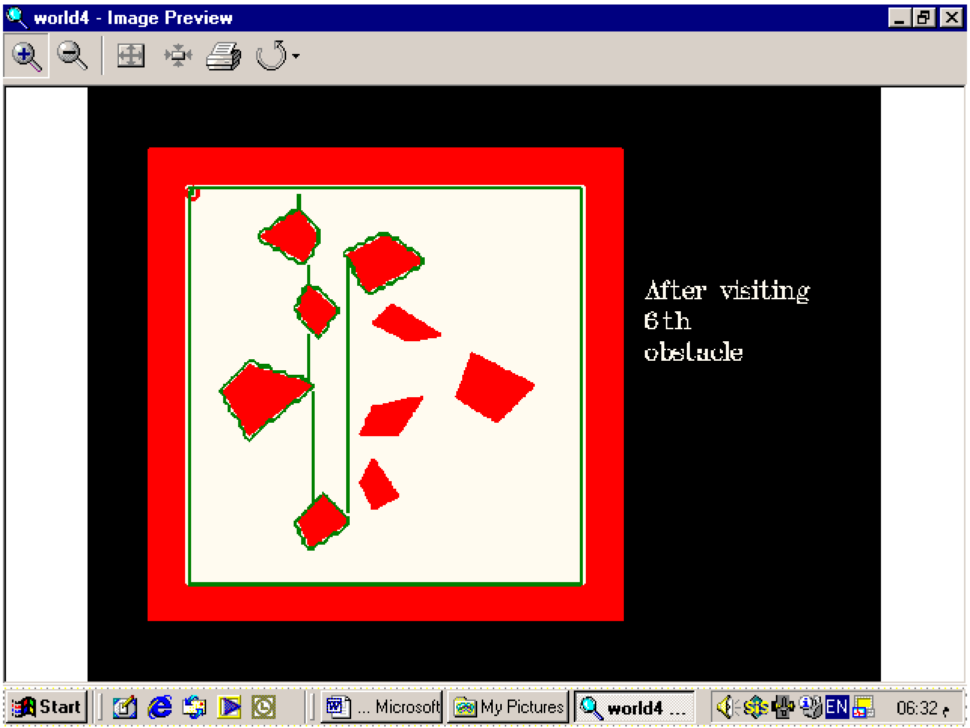

Figure (8): Computer simulation after visiting the six*th* obstacle

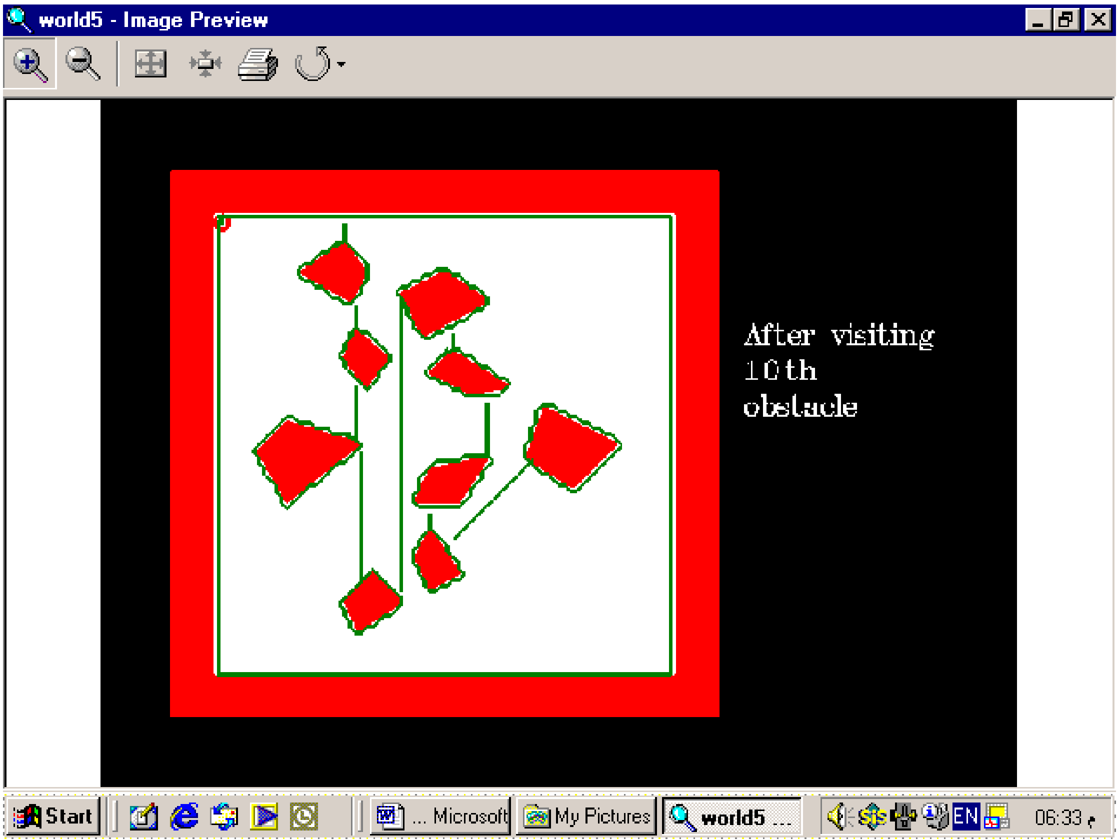

Figure (9): Computer simulation after visiting the ten*th* obstacles

Conclusion and Future Works

Conclusion

The model of cognition finds application in a wide domain of problems under robotics. This algorithm is used for path planning optimization in a dynamic environment in order to reduce the much-wasted time for planning a complete path used in other planning algorithms. It helps in selection of the next point and the path up to that point only in one genetic evolution. This is extremely fast and can take care of movable obstacles of speed comparable to the robot. Automated methods for constructing knowledge by a robot about its environment are therefore a significant research problem in modern robotics. A number of models of coordination are available. The finite state machine and time Petri net are few popular models.

Future works

1-Using quadtree based heuristic search for path traversal problem.
2-Using self-organization map for online navigational planning.
3-Using modular back-propagation neural nets for online navigational planning.
4-Using A Language For Action (ALFA) for coordination among sub-modules for mobile robot.
5-Wire-antenna designs using genetic algorithms.

References

Borenstain, J., Everett, H.R, and FENG, L.,(1996). Navigating Mobile Robots: systems and Techniques, A.,K., Peters, Wellesley.

Brown, R.G., (1997). "Introduction to random signal and applied kalman filtering", John Wiley and Sons, New York.

Filho, J.L.R. (1994). "Genetic algorithm programming environment", IEEE Computer Socity Press, PP.28-43, June.

Freedman, P., (1991).“Time, Petri nets, and robotics, ”IEEE Trans. On Robotics and Automation, Vol.7, No. 4, PP.417-433.

Goldberg, D.E,(1989). Genetic Algorithm in Search Optimization and Machine Learning, Addition Wesley, Reading, MA.

Hutchinson, S.A and Kak, A.C., (1989). ”Planning-sensing strategies in a robot work cell with Multi-sensor capabilities, “IEEE Trans. On Ronoticsans automation, Vol.5, No.6.

Krebs., M., (1998). "A task Driven 3D Object Recognition System Using Bayesian Networks", Proc. of Int.Config. On Computer Vision, PP527-532.

Matlin, W.(1996). Margaret, Cognition, Halt Sounders, Printed and Circulated by Prism Books, India.

Michalewicz, Z., (1986). Genetic Algorithm + Data Structure = Evolution Programs, Springer-Verlag, New York, 3td edition.

Murata. T. (1989). Petri nets:" Properties, analysis and applications", Proceedings of the IEEE, 77(4):541{557, Apr .

Nicholas, A., (1991). "Artificial Vision for Mobile Robot" , The MIT Press, Cambridge ,MA.

Patnaik, S., Konar, A.K.,(1998). “Visual Perception for Navigational Planning and Coordination of Mobile Robots,” Indian Journal of Engineers, Vol.26, Annual Number ’97, PP.21-37.

Pearl, I., (1986). "Fusion, Propagation and structuring in belief networks", Artificial intelligence Elsevier, Vol29, PP.241-288.

Peterson, J.L.,(1981). "Petri net theory and the modeling of systems", (Englewood Cliffs).