Computers and Electrical Engineering 37 (2011) 1058–1070

Contents lists available at SciVerse ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Path planning strategy for autonomous mobile robot navigation using Petri-GA optimisation ☆

Jagadish Chandra Mohanta [a,*], Dayal Ramakrushna Parhi [b], Saroj Kumar Patel [b]

[a] Mechanical Engineering Division, C.P.R.I., Bangalore 560 080, India
[b] Department of Mechanical Engineering, N.I.T., Rourkela 769 008, India

ARTICLE INFO

Article history:
Received 19 June 2010
Received in revised form 13 July 2011
Accepted 16 July 2011
Available online 3 September 2011

ABSTRACT

In this paper, a novel knowledge based genetic algorithm (GA) for path planning of multiple robots for multiple targets seeking behaviour in presence of obstacles is proposed. GA technique has been incorporated in Petri-Net model to make an integrated navigational controller. The proposed algorithm is based upon an iterative non-linear search, which utilises matches between observed geometry of the environment and a priori map of position locations, to estimate a suitable heading angle, there by correcting the position and orientation of the robots to find targets. This knowledge based GA is capable of finding an optimal or near optimal robot path in complex environments. The Petri-GA model can handle inter robot collision avoidance more effectively than the stand alone GA. The resulting navigation algorithm has been implemented on real mobile robots and tested in various environments to validate the developed control scheme.

1. Introduction

Motion planning is an important aspect in the field of mobile robotics. Motion planning is to find a suitable collision-free path for a mobile robot to move from a start configuration to target configuration in an environment consisting of obstacles. In course of motion planning, very often this path is highly desirable to be optimal or near-optimal with respect to time, distance, energy and smoothness. Distance is a commonly adopted criterion. Since last few decades robot path planning has been an emerging area, and many techniques have been adopted to tackle this problem [1], such as fuzzy logic techniques [2–4], genetic algorithms, potential field methods [5,6], neural networks approaches [7,8] and even hybrid techniques [9,10]. Each method has its own advantage over others in certain aspects. Generally, the main difficulties for robot-path-planning problem are computational complexity, local optimum and adaptability. Researchers have always been seeking alternative and more efficient ways to solve the problem.

It is obvious that path planning can be viewed as an optimisation problem (e.g., shortest distance) under certain constraints (e.g., the given environment with collision-free motion). Since the appearance of genetic algorithms (GA) in 1975 [11], GAs have been used in solving many optimisation problems successfully. GA is stochastic search technique analogous to natural evolution based on the principle of survival of the fittest. The potential solutions of a problem are encoded as chromosomes, which form a population. Each individual of the population is evaluated by a fitness function. A selection mechanism based on the fitness is applied to the population and the individuals strive for survival. The fittest ones have more chance to-be selected and to reproduce offspring by means of genetic transformations such as crossover and mutation. The process is repeated and the population is evolved generation by generation. After many generations, the population converges to solutions of good quality, and the best individual has good chance to be the optimal or near optimal solution. The

☆ Reviews processed and approved for publication to Wysocki.

* Corresponding author.
E-mail address: jagadish_mohanta@yahoo.co.in (J.C. Mohanta).

0045-7906/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compeleceng.2011.07.007

feature of parallel search and the ability of quickly locating high performance region [12] contribute to the success of GAs on many applications.

Many researchers applied GAs for path planning of mobile robots [13–15]. However, like most early GA applications, most of those methods adopt classical GAs that use fixed-length binary strings and two basic genetic operators, and few modifications were made to the algorithms. Genetic algorithm based path planning with fix-length binary string chromosomes based on cell representation of mobile robot environment has been proposed [16]. Its binary encoding is biased and inefficient. Besides, in order to use the standard GA, the path planning solutions are restricted to X-monotone or Y-monotone. The classical GAs uses binary strings and two basic genetic operators. After encoding solutions to a problem, the classical GAs are more like "blind" search, and perform well when very little prior knowledge is available. However, GAs do not have to be "blind" search, when additional knowledge about problem is available, it can be incorporated into GAs to improve the efficiency of GA [17,18]. Path planning is such a problem that requires knowledge incorporation into the GAs for the problem. Graph technique is a traditional way of representing the environment where a mobile robot moves around. A genetic algorithm based on MAKLINK graph environment representation is proposed by many authors [19–21]. In this genetic algorithm, the path is represented by variable length chromosomes formed by mid-points of the free-links, which is a more natural way of encoding than binary strings. This graph based method needs to form a configuration space before applying the genetic algorithm.

A specialised genetic operators was designed [3,22] with some heuristic knowledge. A path is represented by a hierarchically ordered set of vectors that define path vertices generated by a modified Gram–Schmidt orthogonalization process [23]. An evolutionary planner was proposed using GA for both on-line and off-line planning [3]. However, both approaches are relatively complicated on problem representation, evaluation, or GA structure. A novel Genetic Algorithms (GAs) approach was proposed [24] for a near optimal path planning of a mobile robot in a greenhouse. The chromosome encoding features in inverse proportion between research spaces of GAs and complexity of obstacles. They designed the fitness evaluation for both incomplete and complete paths to guide the evolutional direction. An improved genetic algorithm performance was developed by considering more efficient genotype structure for a known environment with static obstacles [25]. Motion was constrained to only row-wise navigation. The above work was improved and presented results of a genetic algorithm based path-planning model developed for local obstacle avoidance of a mobile robot in a given search space [26]. While a new approach based on evolutionary computations is discussed to solve constrained nonlinear programming problems [27]. Three dominant hybrid approaches to intelligent control are experimentally applied to address various robotic control issues for navigation of mobile robot [28]. The hybrid controllers consist of a hierarchical NN-fuzzy controller applied to a direct drive motor, a GA-fuzzy hierarchical controller applied to position control of a flexible robot link, and a GP-fuzzy behaviour based controller applied to a mobile robot navigation task. The problem of path finding through a maze of a given size has been addressed by [29]. They presented a biologically inspired solution using a two level hierarchical neural network for the mapping of the maze. However, in all such studies, a limited effort was made to find an optimal controller (instead, a GA was designed based on a particular user-defined function and rules) for mobile robots navigation with multiple targets. Moreover, in this literature no way it has been noticed the incorporation of Petri-Net model with GA, which is attempted in the current investigation to make an integrated navigational controller for path planning of mobile robots.

In this paper the motion planning of multiple mobile robots with multiple targets in presence of obstacles in a priori unknown environment using modified knowledge based GA controller is discussed. This task could be carried out by specifying a set of GA rules by taking into account the different situations found by the mobile robots. The approach is to extract a set of GA rules from a set of trajectories provided by human intelligence. Problem-specific genetic operators are not only designed with domain knowledge, but also incorporate small-scale local search that improves efficiency of the operators. For this purpose inputs are left obstacle distances (LOD), front obstacle distances (FOD) and right obstacle distances (ROD) to all the GA controller and output heading angles (HA) are expressed by crisp values in terms of encoded generation function distributions. In order to avoid inter robot collision during navigation each robot incorporates a set of collision prevention rules implemented as a Petri-Net model in its controller. The proposed GA based controller can be suitably used for both static and dynamic environments. The effectiveness and efficiency of the proposed approach are demonstrated through simulation studies as well as real-time experiment in various environments.

2. Design of mobile controller using GA

2.1. Basic approach for obstacle avoidance

As specified earlier a genetic algorithm (GA) based obstacle avoidance scheme has been used here for path planning of multiple robots with multiple targets in presence of obstacles. Genetic algorithms are heuristic optimisation methods whose mechanisms are analogous to biological evolution. The evolutionary procedure employed in the simulations consists in programming a standard genetic algorithm (GA) system. The speed of genetic algorithm depends heavily on the encoding scheme of the chromosomes and on the genetic operators that work on these chromosomes [30,3]. In order to speed up a GA, the chromosome's and gene's structures need to be as simple as possible. In addition, only a few, but very effective, reproduction operators should be applied on the chromosomes. A GA operates on a population of chromosomes, which represent possible solutions for a given problem. This implementation is a new approach to the path-planning and obstacle

avoidance problem, representing each chromosome as a group of basic attitudes. These attitudes define the robot's movements in agreement with the feedback generated by its environment. Each feedback, which is used as input to the system, is based on the sensors reading and on the robot's direction to its goal location. The sensors reading are presented to the GA system in a simplified form. The proposed simulator provides 6 sets of ultrasonic and 4 sets of IR sensors for detecting the obstacles and bearing of the targets. However, the distances are calculated at each instantaneous position of the robots. These distances are calculated based on the position and orientation of the obstacles with respect to robots instantaneous position.

For this purpose the basic inputs to the genetic controller are front obstacle distances (FOD), left obstacle distances (LOD) and right obstacle distances (ROD) to all the GAC and output heading angles (HA) are expressed in terms of encoded generation function distributions by crisp values. Then, these crisps values are converted to binary value in order to facilitate the interface the machine language for further processing of the controller. To visualise the above genetic controller in real sense the problem has been addressed in different stages. The stages are analysed below

2.1.1. Stage 1: Formation of pool set for obstacle avoidance

From the sensors out puts (FOD, LOD and ROD) distances an initial population pool is created 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 terms of a heading angle between the current directions of the robots' steering with respect to targets' directions from its start to end point in the search space. The initial population with size n can be presented as follows:Initial population = $\langle P_1, P_2, \ldots, P_n \rangle$

Each structure have the elements $p_{(i,i)}$ which are simply an integer string of length L, in general.

Each population have 5-sets of chromosomes which are represented by Element numbers 1 to 5.

$$\begin{array}{lllll} \text{Element : 1} & \text{Element : 2} & \text{Element : 3} & \text{Element : 4} & \text{Element : 5} \\ P_1 = \{p_{1,1} & p_{1,2} & p_{1,3} & p_{1,4} & p_{1,5}\} \\ P2 = \{p_{2,1} & p_{2,2} & p_{2,3} & p_{2,4} & p_{2,5}\} \\ \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots \\ P_n = \{p_{n,1,} & p_{n,2,} & p_{n,3,} & p_{n,4,} & p_{n,5}\} \end{array} \quad (1)$$

where,

Element no. 1 ($p_{1,1}$ to $p_{n,1}$) represents the front obstacle distance (FOD).
Element no. 2 ($p_{1,2}$ to $p_{n,2}$) represents the left obstacle distance (LOD).
Element no. 3 ($p_{1,3}$ to $p_{n,3}$) represents the right obstacle distance (ROD).
Element no. 4 ($p_{1,4}$ to $p_{n,4}$) represents the instantaneous heading angle (HA) with respect to target potion.
Element no. 5 ($p_{1,5}$ to $p_{n,5}$) represents the sign conversion ('+ve', 'zero' and '−ve') for clock wise, straight and anti clockwise based on the direction of HA (ϕ) respectively which is shown in Fig. 1.

Beyond 511 mm radius the region is treated as obstacles free and in such case robot considered that, there is no obstacle in the specified direction and starts moving towards target till find any obstacle on the way within the range. In this case heading angle will be zero (Case 5 in Table 1). In case 2 left obstacle distance is 'medium', front obstacle distance is 'near' and right obstacle distance is 'near', so robot will take a left turn of 12° (−ve) in order to avoid obstacle. Similarly in case 8, left obstacle distance is 'near', front obstacle distance is 'very near' and right obstacle distance is 'medium', so robot will

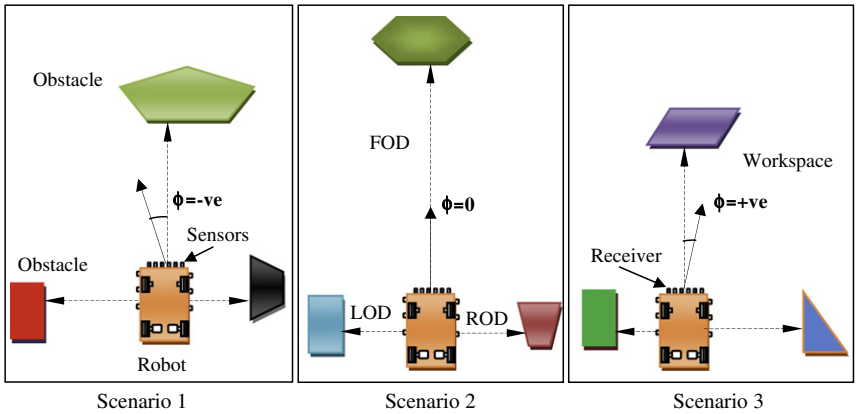


Fig. 1. Sign convention of GA output in terms of heading angle (HA) with respect to obstacle position.

Table 1
Heading angle (HA) with respect to different obstacle positions.

Case	FOD (mm)	LOD (mm)	ROD (mm)	HA (Degree)	Direction (+/−)ve
1	250	200	150	18	−ve
2	300	340	260	12	−ve
3	400	300	150	10	−ve
4	270	280	160	15	−ve
5	600	100	120	0	straight
6	505	400	280	5	−ve
7	480	220	450	14	+ve
8	120	200	360	12	+ve
9	500	100	450	16	+ve
10	320	450	180	10	−ve

take a right turn of 12° (+ve) in order to avoid obstacle. Each degree of rotation heading angle is taken as (511/180)th part for both clock wise and anti clockwise movement of the robot. For simplicity a set of 10 populations has been shown in tabular form (Table 1).

2.1.2. Stage 2: Analysis of fitness function for obstacle avoidance

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 obstacle avoidance, in our case, is the possible collision free motion of robot with optimum heading angle (HA) with respect to target location, thereby optimising the trajectory between the start and end point in the environment. Thus, the fitness function is responsible for optimal obstacle avoidance. The proposed knowledge based GA controller helps in computing the total number of steps (by optimising the HA, i.e. minimum value of (ϕ) gives shorter distance) required by the mobile robot to reach the target. Considering the above constraints, the fitness value for a complete solution is computed as follows:

$$f_{Total} = 0.4(f_1) + 0.15(f_2) + 0.15(f_3) + 0.15(f_4) + 0.15(f_5) \tag{2}$$

where,

$$f_1 = \sqrt{(f_2^2 + f_3^2 + f_4^2)} \tag{3}$$

$$f_2 = |C_{FOD} - p_{ci,1}| \tag{4}$$

$$f_3 = |C_{LOD} - p_{ci,2}| \tag{5}$$

$$f_4 = |C_{ROD} - p_{ci,3}| \tag{6}$$

$$f_5 = |TA - HD| \tag{7}$$

and $(C_{FOD} - P_{ci,1}), (C_{LOD} - P_{ci,2})$ and $(C_{ROD} - P_{ci,3})$ are the best distances (child) obtained from the given pool set of front, left and right obstacles distances from instantaneous obstacle position with respect to initial position. These distances are the output of sensors data. TA and HD are the target angle and heading direction respectively. The coefficients of the fitness function are computed statistically using fuzzy inference technique and are cited in Appendix A.

2.1.3. Stage 3: Crossover of parameters and its analysis

During the operation of reproduction crossover is applied on the chosen parent chromosomes only within a certain probability, i.e. the crossover probability. In the chosen crossover operator, two parent chromosomes are combined to apply a single-cross-point value encoding crossover. The crossover operator has been modified to produce two offspring chromosomes with each crossover operation. This is achieved by using the gene information, which were not used to build offspring one, in order to build a second chromosome. In the proposed controller we used the crossover operators for front, left and right obstacle distances as well as for the heading angle. The function of the crossover operators for first two pool set (shown in Table 1) are illustrated in Fig. 2.

2.1.4. Stage 4: Mutation

For mutation, almost every operation that changes the order of genes within a chromosome or that changes a gene's value (such as location or direction) is a valid mutation operator. The mutation operator has been designed according to the addressed obstacle avoidance problem. The chosen mutation operator checks with a mutation probability for every single gene whether it should be mutated or not. If a gene is to be mutated, a random number between 1 and the total number of population in the search space is assigned to location and a random direction, either clock wise, anticlockwise or straight, is based on reference direction. This mutation variant has the advantage that it gives the opportunity for a chromosome to

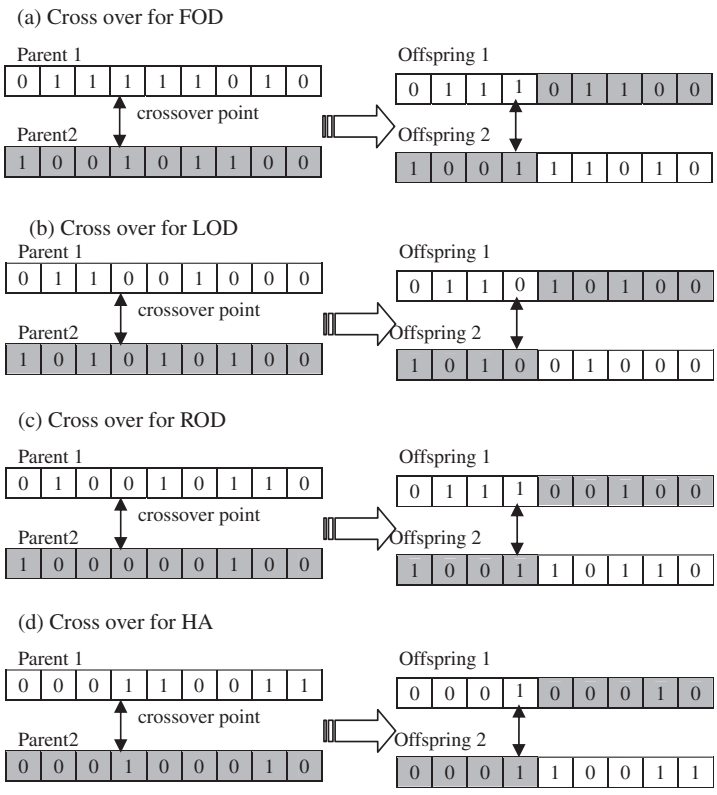


Fig. 2. Single-Cross-Point, value-encoding crossover for FOD, LOD, ROD and HA.

become significantly altered. That means that the complete search space will be explored and it therefore prevents the GA from getting stuck in a local optimum. The fitness of all affected genes (steps) is re-evaluated and stored in the variable feasibility immediately after the changes in location and direction are made. Each step's fitness is therefore always up to date with each instant position of robot.

2.1.5. Stage 5: Evaluation of fittest child according to fitness function

The evaluation of fittest child is computed as per the fitness function described in stage 2. The outline of the schematic diagram showing the flowchart for the proposed knowledge based genetic algorithm is given in Fig. 3 and the detailed working principle is highlighted below.

The GA controller begins its search by randomly creating a number of solutions (equals to the population size) represented by the binary strings and are evaluated by the fitness function derived in Eq. (1). Two parents for FOD, LOD and ROD are selected from the pool set according to fitness function. Once the parents are chosen from the population, they are modified by using three operators—reproduction, crossover and bit-wise mutation. The iteration process involving these three operators followed by the fitness evaluation is called a generation. The generations proceed until a termination criterion is satisfied. In the current approach, the termination criterion can either be that the preset maximum generation is exceeded, or that the best solution remains unchanged for certain generations. Accordingly the best heading angle (HA) will be decided and command execution starts to move the robot towards the target. The proposed algorithm can also be suitably applied for dynamic environment. During navigation it checks sensing about the environment periodically. If the environment is changed, the algorithm will re-evaluate the current population according to the new environment and starts the process to get a new solution. In such cases, in order to increase the diversity of the population mutation with higher probability is applied to the current population. The obstacle avoidance behaviour of robots using genetic algorithm is incorporated through Petri-Net model for successful navigation of the mobile robot. The detail of the proposed Petri-Net model is discussed in the following section.

2.2. Petri-Net model for inter-robot collision and obstacle avoidance

The Petri-Net model was developed to avoid inter-robot collision and obstacle avoidance during navigation [31]. The schematic representation of this model is shown in Fig. 4. Each robot embedded with Petri-net enables it to avoid collision among other robots. This model comprises of seven tasks which are explained briefly.

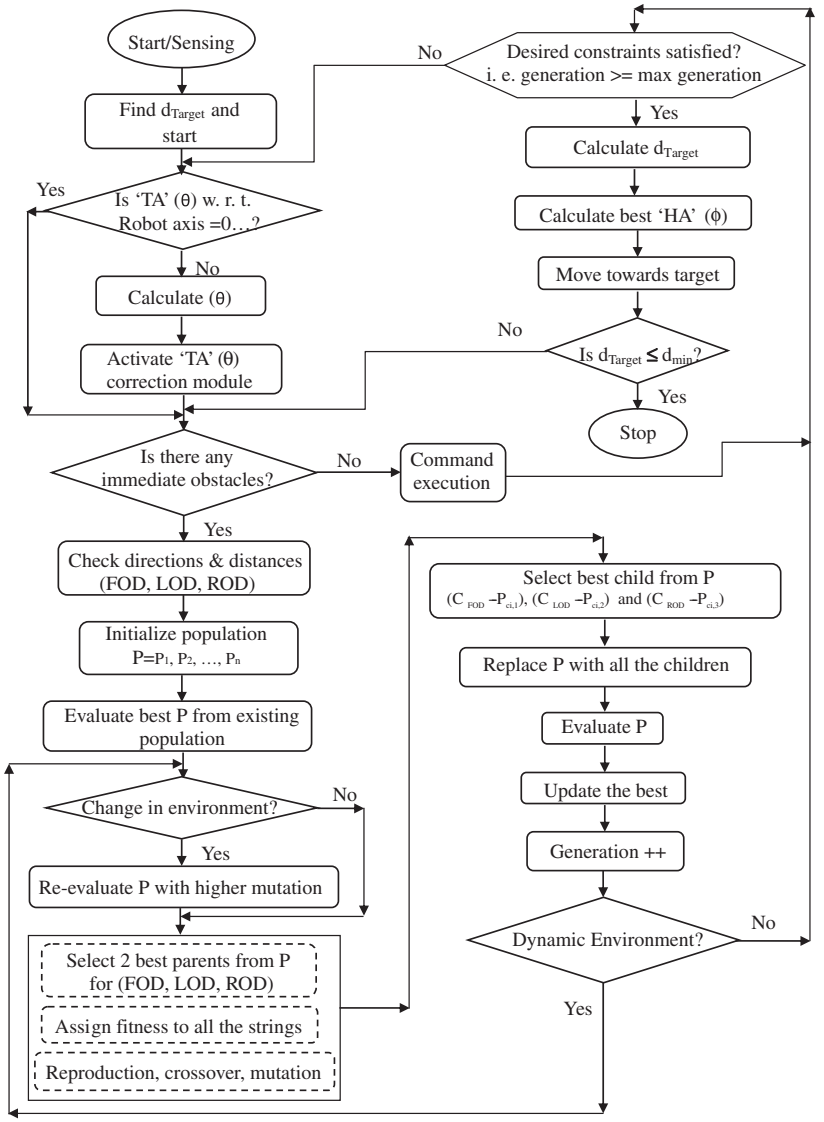


Fig. 3. The outline of schematic diagram showing the flowchart for the proposed motion planning scheme.

It is assumed that initially the robots are placed in a highly cluttered environment without any prior knowledge of one another, targets and obstacles. This means the robot is in state "Task 1" ("Wait for the start signal"). Now the token is in place "Task 1" in Fig. 4.

Once the robots have received a command to start searching for the targets, they will try to locate targets while avoiding collision among themselves and obstacles. The robot is thus in state "Task 2" ("Moving, avoiding obstacles").

During navigation, if the path of a robot is obstructed by another robot, a conflict situation is raised (State "Task 3", "Detecting Conflicts"). Conflicting robots will negotiate with each other to decide which one has priority. The lower priority robot will be treated as a static obstacle and the higher priority robot as a proper mobile robot (state "Task 4", "Negotiating"). As soon as the conflict situation is resolved, the robots will look for other conflicts and if there is no other conflict they will execute their movements (state "Task 5", "Checking for conflict and executing movements"). Finally approaching towards target (state "Task 6", "searching for targets") and wait after reaching the target (state "Task 7", "Waiting").

3. Simulation results

This section presents exercises aimed at illustrating the ability of the proposed control scheme to manage the navigation of mobile robots in different situations. Simulations were conducted with the help of MATLAB software package developed by the author. This generalised programme enables to generate any number of mobile robots, targets and obstacles, and

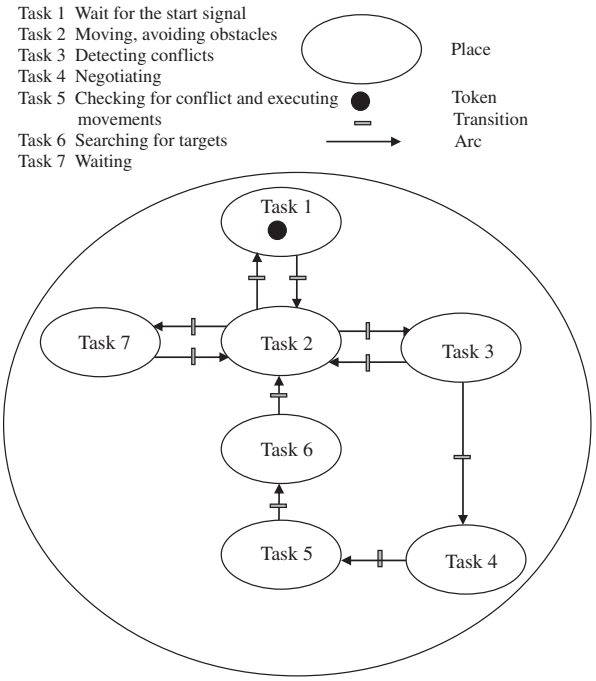


Fig. 4. Petri-Net model for avoiding inter-robot collision.

controls in an artificial simulated environment containing multi targets and obstacles. Three exercises have been designed by arranging obstacles in different fashion to create different environment for the GA based controllers to show the capabilities of the proposed control scheme.

3.1. Collision-free movement, obstacle avoidance and target seeking in highly clutter environment

The navigation environment for collision free motion and obstacle avoidance for three robots with three targets in a highly cluttered environment has been shown in Fig. 5. This exercise is designed to demonstrate that each robot reach their

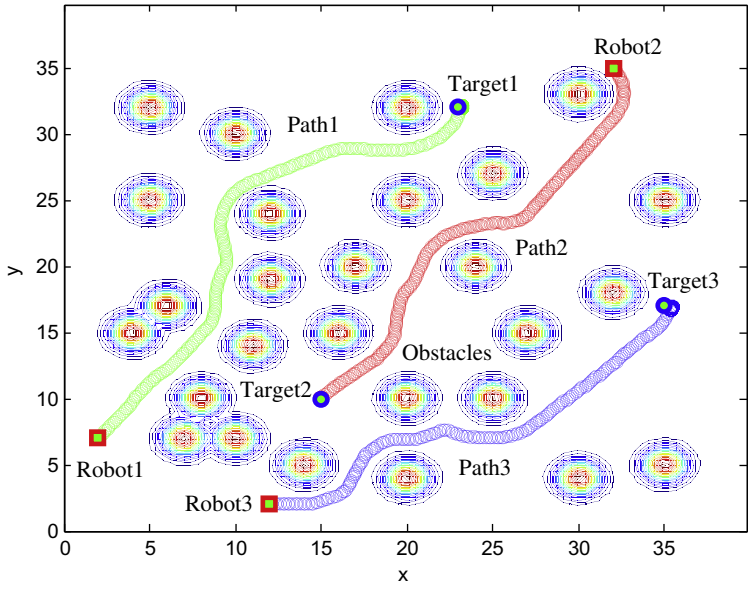


Fig. 5. Navigation environment for collision avoidance by three robots with three targets in highly cluttered environment.

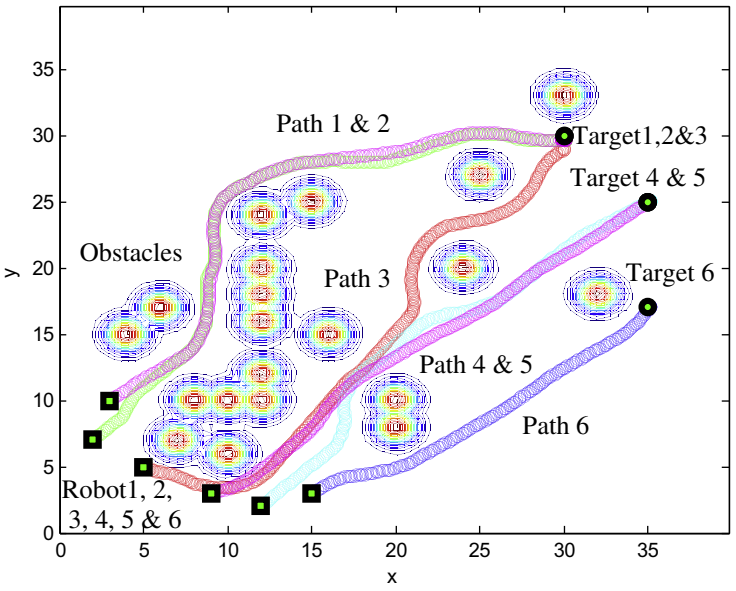


Fig. 6. Navigation environment for target seeking behaviour for collision free motion and obstacle avoidance by multiple robots with multiple targets.

targets without colliding with other robots while avoiding the obstacles. Robots choose their path by its own to reach the target assigned to them by following the shortest trajectories. It can be noted that the robots stay well away from the obstacles and move in smooth path from its start location to end location and found their targets efficiently.

3.2. Obstacle avoidance and target seeking by several robots

This exercise involves six mobile robots, twenty obstacles and three targets initially assembled in a cluttered environment. In this simulation, each robot has reached their nearest target in an efficient manner without any collision between themselves and obstacles in a cluttered priori unknown environment (see Fig. 6).

3.3. Wall following and target seeking behaviour

The wall following and target seeking behaviour has been shown in Fig. 7. This exercise involves the scenario having four robots and two targets. In the present scenario the obstacles are arranged in a particular fashion so that they act like a wall between the robots and the targets. As the robots starts searching for their targets, they find the wall along which they continue to move by applying the wall following rules. In the initial position the robot is heading towards the dead-end. This is due to the additional context information provided by the proposed controller, allowing the robot to perceive the dead end. The controller initiates a turning manoeuvre, which lasts until the robot is heading away from the dead-end. Afterwards, the normal wall-following behaviour guides the robot to the exit of the corridor while keeping a safe distance between the walls. Finally the all the robots reach their targets efficiently. This shows the robot trajectory from start to target configuration without suffering from "dead cycle" problem.

In the above simulation, it can be seen that each robot has reached their nearest target in an efficient manner without any collision between themselves and obstacles in different environments. Further the robots well compromised between themselves in the cluttered environment during navigation using the proposed control scheme. The same controller has been demonstrated for other cases of simulation in the succeeding sections.

4. Approach for design validation with other model

In this section a comparison has been made with [32] and results from current control scheme in simulation and experimental mode. The performance of the two methods is mainly evaluated on the basis of path length.

The results from [32] shown in Figs. 8 and 9(a) are compared with the results obtained from current study (Figs. 8 and 9(b)) for similar environments as has been tabulated in Table 2.

In the first case, both the controllers are applied to a situation with complex maze with different shapes of boundary that would cause the existence of "dead cycle" problems. In this case the robot cannot see the target directly due to the wall in between them. It can be observed in the environment shown in Fig. 8(a) presented by [32] model that the robot is trapped at each steps throughout the maze due to the existence of local minima and finally able to reach the target by following a longer

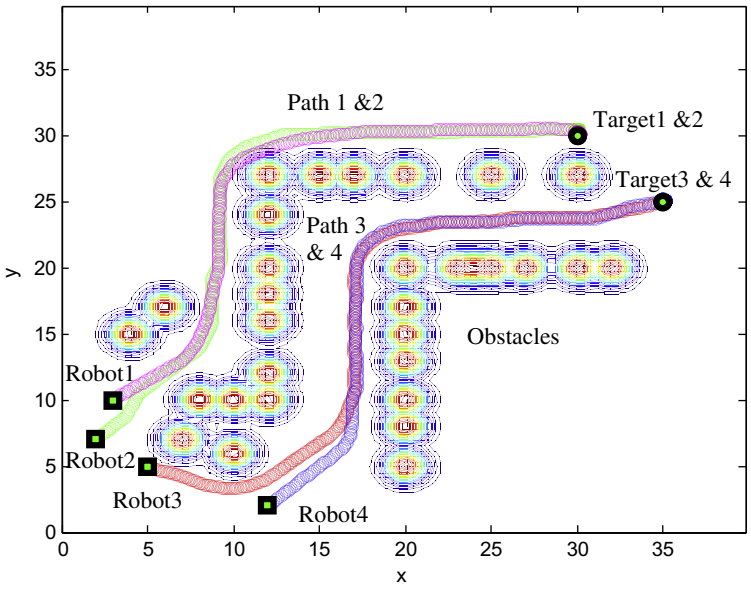


Fig. 7. Navigational scenario for wall following and target seeking behaviour of multiple robots.

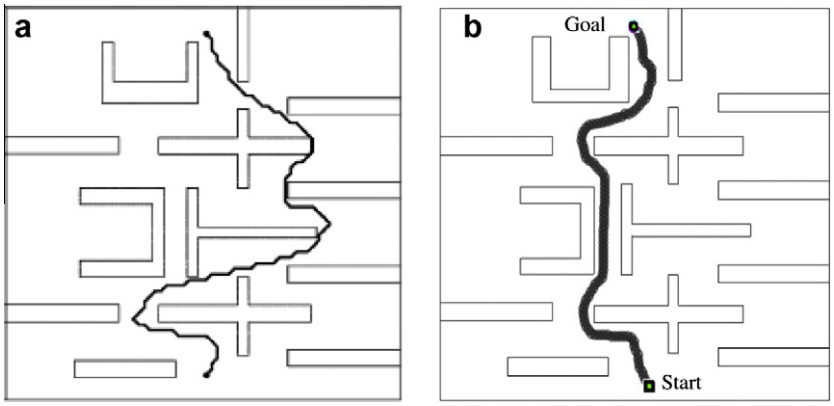


Fig. 8. Comparison of results from [32] and the current investigation (Scenario 1).

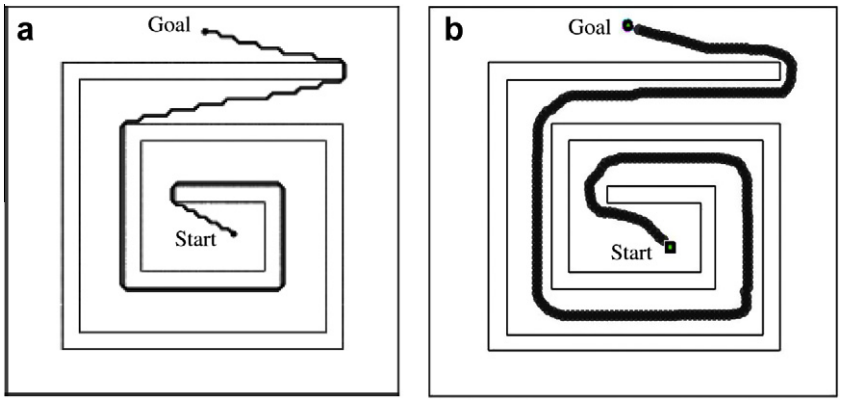


Fig. 9. Comparison of results from [32] and the current investigation (Scenario 2).

Table 2
Comparison of results with [32] model and the current investigation.

S. no.	Environmental types	Path length of [32] model, in 'cm'	Path length from current investigation in 'cm'
1	Complex maze with different shape of rectangular obstacles (Fig. 8(a) and (b))	11.6	8.8
2	Closed aisle (Fig. 9(a) and (b))	20.4	19.2

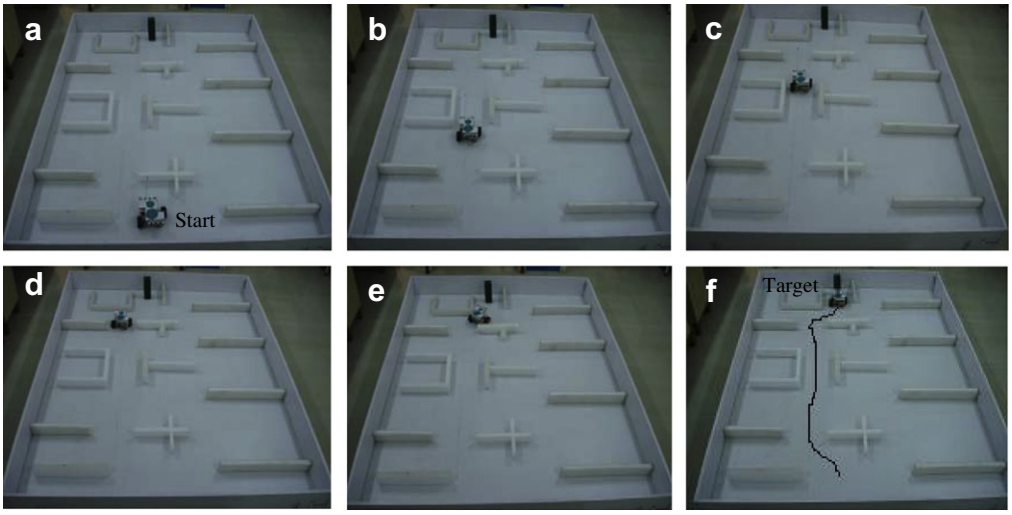


Fig. 10. Experimental results for navigation of mobile robot in the similar environment shown in Fig. 8(b).

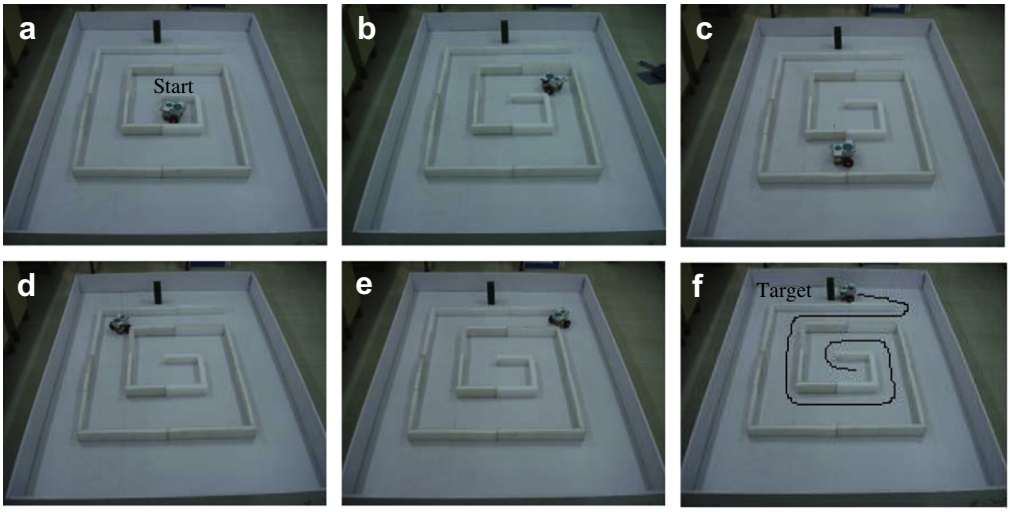
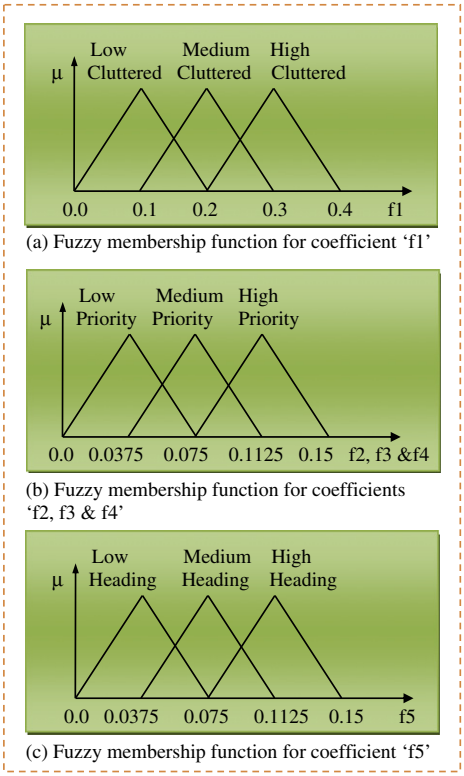


Fig. 11. Experimental results for navigation of mobile robot in the similar environment shown in Fig. 9(b).

trajectory. In Fig. 8(b) the robot can avoid the maze made with similar boundary found on its way towards the target efficiently by the proposed Petri-GA model. In this case robot will first takes left turn due to the obstacle (boundary) in front of it, then sense the target and finally follows the walls in order to reach the target successfully by receiving systematic information from the sensors through the state memory strategy.

In the second case, the robot supposed to find the target approximately in a complex situation like closed aisle for a dead cycle problem. In Fig. 9(a) proposed by [32] the robot got trapped in the U-shaped area first, and then it was escaped from the trap by taking a loop and eventually reaches the target by following a zigzag motion. Fig. 9(b) shows a simulation of the robot behaviour adapted by means of the proposed method. In the initial position the robot is heading towards the dead-end.



(a) Fuzzy membership function for coefficient 'f1'

(b) Fuzzy membership function for coefficients 'f2, f3 & f4'

(c) Fuzzy membership function for coefficient 'f5'

Fig. A1. Fuzzy membership function used for evaluation of different fitness function coefficient using proposed motion planning scheme.

During course of motion robot gets additional context information from the proposed controller to perceive the dead-end. The controller initiates a turning manoeuvre, which lasts until the robot is heading away from the dead-end. Afterwards, the normal wall-following behaviour guides the robot to the exit of the corridor by keeping a safe distance to the left wall.

In both the scenarios of model [32], it can be seen that the path of robot has sudden change in direction with some greater steering angles. This has been taken care of in the present investigation. The performance of the two models was mainly evaluated on the basis of path length of trajectory which is shown in Table 2. From the above simulation results it is clear that, the developed algorithm can efficiently drive the robot in different cluttered environment. Experimental verification of the above simulation results have been shown in Figs. 10 and 11.

5. Experimental results and discussions

To demonstrate the effectiveness of the above control system and validity of the algorithm, a variety of experiments using prototype robots were conducted. In this section we present the simulation results from our motion planner, which was operated in an environment with rectangular aisle of different shapes. The path traced by the robot during motion was marked on the floor by means of a pen attached to the back of the robot frame. The position and posture of the prototype robot can be estimated by dead reckoning using the equations developed and information from the encoders arranged on the wheels and the steering axes. The robot considered for experiment is a differential drive robot with an on-board PC and wireless Ethernet. There are six ultrasonic and four infra red sensors mounted around the top of robot (out of which two sensors in each in front and back sides and one in each at the left and right side of the mounting) in order to sense the front, left, right and back obstacle distances. Although the range of the above mentioned sensors are about 1m, but the robot takes a turn when the obstacle distances reaches 10cm. Control card (commands from microprocessor are given in form of voltages through D/A converter using an interface boRIF-01). These voltage signals drive the DC-motors via driver circuits, so that driving torque is occurred. While pulses from the encoders are counted by UPP (Universal Pulse Processor) on the interface board. These counts are transmitted to the microprocessor for further processing.

Two different cases of similar environments as described by [32], which are already verified in simulation mode, have been verified experimentally (Figs. 10 and 11) to show the effectiveness of the developed controller.

In Fig. 10, it is demonstrated a situation where robot and target are placed in opposite corner of the complex maze (created by a variety of rectangular obstacle configurations). Initially robot cannot see the target directly due to the presence of obstacles between robot and target. When robot starts motion it senses the target and speeds up in straight path towards the

targets up to the U-shaped wall and slows down to take right turn to avoid obstacle, then follows a wall following rules to reach the target. The robot autonomously chooses its way in the shortest trajectory to reach the desired destination by getting the optimised heading angle obtained from knowledge based GA controller. For the second robot navigation (Fig. 11f), it can be observed that, the robot follows a straight path except the turning points from its start to the goal position inside the closed aisle. In some cases the robot changes its direction and moves with some rotational motion until it reaches the target. The developed controller takes care to invoke a new path based on available information received by the robot about the environment with ruled based heuristic recovery GA approach.

The experimentally obtained paths follow closely those traced by the robots during simulation. From above experimental results, it can be seen that the robots can indeed avoid obstacles and reach the targets. It has been concluded by comparing the results from both the simulation as well as experiment that, the path followed by the robots using the proposed controller can successfully arrive at the target by avoiding obstacles. The trajectories are smooth and take reasonably efficient paths as compared to [32] paths. More than thirty experiments have been conducted to test the model. The maximum velocity of mobile robot used for navigation is 0.05 m s^{-1}. There are a number of trials with varying complexity to show that the model works for different sizes and numbers of obstacles. The real time simulated results show the effectiveness of the developed controller.

6. Concluding remarks

In this study, the prime objective was to construct the framework of some hierarchical or procedural control structures to implement basic navigation problems for multiple robots based on GA. Firstly, we have proposed a method for adjustment of fitness values to avoid statistical variation due to randomness in initial positions and orientations of obstacles. Then the distances of obstacles from three directions (viz. front, left and right) were evaluated by using suitable fitness function and optimised by the proposed algorithm based upon an iterative non-linear search, which utilises matches between observed geometry of the environment and a-priori map of position locations, there by correcting the position and orientation of the robot to find targets. During navigation of several robots there could be the chances for conflict situations and inter robot collision among them. This has been taken care in the present study by suitably designing of a general-purpose conflict avoidance module based on Petri-Net model and embedded in the controller of each robot to navigate safely in the environment. The Petri-GA model can handle inter robot collision avoidance effectively than the stand alone GA. The proposed method has been compared with other method [32] and the current control scheme is found to be more efficient in terms of path length. The developed strategies have been checked via simulations as well as experiments, which show the ability of proposed controller to solve the multiple robot navigation tasks in an optimised way and to obtain a strategy for this purpose.

Appendix A

The Fuzzy membership functions for different coefficients are shown in Fig. A1. The coefficients for the sub-fitness functions $f_1, f_2, \ldots, f_5$ are calculated using fuzzy inference technique and statistical analysis. The degree of membership functions are detailed as follows.

The maximum values are estimated statistically from the membership functions distribution. In the current case the high intensity factors are considered for navigation of mobile robots. However, the other values of coefficient can be tried in future for alternative navigational performance. It may be noted that the fuzzy inference technique is only used for calculation of the sub coefficients for the fitness function described in Eq. (1).

References

[1] Henrich D, Wurll C, Worn H. On-line path planning by heuristic hierarchical search. IECON'98. In: Proceedings of the 24th annual conference of the IEEE industrial electronics society. New York, NY, USA; 1998. p. 2239–44.
[2] Song KT, Tai IC. Fuzzy navigation of a mobile robot. IEEE/RSI. In: International conference on intelligent robots and systems; 1992.
[3] Xiao J, Michalewicz Z, Zhang L, Trojanowski K. Adaptive evolutionary planner/navigator for mobile robots. IEEE Trans Evol Comput 1997;1:18–28.
[4] Wehn HW, Belanger PR. Ultrasound-based robot position estimation. IEEE Trans on Rob Autom 1997;13.
[5] Arambula Cosio F, Padilla Castaneda MA. Autonomous robot navigation using adaptive potential fields. Math Comput Modell 2004;40:1141–56.
[6] Rimon E, Doditschek DE. Exact robot navigation using artificial potential fields. IEEE Trans Rob Autom 1992;8:501–18.
[7] Muniz F, Zalama E, Gaudiano P, Lopez-Coronado J. Neural controller for a mobile robot in a nonstationary environment. In: Proceedings of 2nd IFAC conference on intelligent autonomous vehicles. Finland: Helsinki; 1995. p. 279–84.
[8] Yang SX, Meng M. An efficient neural network approach to dynamic robot motion planning. IEEE Trans Neural Networks 2000;13:143–8.
[9] Nirmal Baran Hui, Dilip Kumar Pratihar. A comparative study on some navigation schemes of a real robot tackling moving obstacles. Rob Comput Integr Manuf 2009;25:810–28.
[10] Akbarzadeh-T MR, Kumbla K, Tunstel E, Jamshidi M. Soft computing for autonomous robotic systems. Comput Electr Eng 2000;26:5–32.
[11] Holland J. Adaptation in natural and artificial systems. Ann Arbor: University of Michigan Press; 1975.
[12] Editor LD, editor. Genetic algorithms and simulated annealing. Los Altos, California: Morgan Kaufman Publishers; 1987.
[13] Ibikunle Ashiru, Chris Czarnecki, Tom Routen. Characteristics of a genetic based approach to path planning for mobile robots. J Network Comput Appl 1996;19:149–69.
[14] Il-Kwon Jeong, Ju-Jang Lee. Evolving cooperative mobile robots using a modified genetic algorithm. Rob Auton Syst 1997;21:197–205.
[15] Ashiru I, Czamecki C, Routen T. Characteristics of a genetic based approach to path planning for mobile robots. J Network Comput Appl 1996;19:149–69.

[16] Sugihara K, Smith J. Genetic algorithms for adaptive motion planning of an autonomous mobile robot. In: Proceedings of IEEE international symposium on computational intelligence in robotics and automation, vol. 7; 1997. p. 138–43.
[17] Michalewicz Z. Genetic algorithms + data structures = evolution programs, 2nd extended edition. Springer-Verlag; 1994.
[18] Grefenstette JJ. Incorporating problem specific knowledge into genetic algorithms. In: Davis L, editor. Genetic algorithm and simulated annealing. Los Altos, California: Morgan Kaufman Publishers; 1987. p. 42–60.
[19] Shibata T, Fukuda T, Kosuge K, Arai F. Selfish and coordinative planning for multiple mobile robots by genetic algorithm. In: Proceedings of 31th conference on decision and control, vol. 12; 1992. p. 2686–91.
[20] Habib MK, Asama H. Effecient method to generate collision free paths for autonomous mobile robot based on new free space structuring approach. In: Proceedings of IEE/RSJ international workshop on intelligent robotics and systems IROS, vol. 291.
[21] Yuming Liang, Lihong Xu. Global path planning for mobile robot based genetic algorithm and modified simulated annealing algorithm. In: Proceedings of the first ACM/SIGEVO summit on genetic and evolutionary computation; 2009. p. 303–8.
[22] Hocaoglu C, Sanderson C. Planning multiple paths with evolutionary speciation. IEEE Trans Evol Comput 2001;51:69–191.
[23] Hocaoglu C, Arthur C. Sanderson. Multimodal function optimization using minimal representation size clustering and its application to planning multipaths. J Evol Comput 1997;5:81–104.
[24] Xuemei Liu, Jin Yuan, Kesheng Wang. A problem-specific genetic algorithm for path planning of mobile robot in greenhouse. IFIP Int Fed Inf Process 2006;207:211–6.
[25] Geisler T, Manikas T. Autonomous robot navigation system using a novel value encoded genetic algorithm. In: Proceeding of IEEE midwest symposium on circuits and systems. Tulsa; 2002. p. 45–8.
[26] Sedighi KH, Ashenayi K, Manikas TW, Wainwright RL, Tai HM. Autonomous local path planning for a mobile robot using a genetic algorithm. In: Proceeding IEEE congress on evolutionary computation (CEC); 2004. p. 1338–45.
[27] Simionescu PA, Dozier GV, Wainwright RL. A two-population evolutionary algorithm for constrained optimization problems. In: IEEE congress on evolutionary computation; 2006. p. 1647–53.
[28] Akbarzadeh-T M-R, Kumbla K, Tunstel E, Jamshidi M. Soft computing for autonomous robotic systems. Comput Electr Eng 2000;26(1):5–32.
[29] Srinivasan S, Mital DP, Haque SA. Novel solution for maze traversal problems using artificial neural networks. Comput Electr Eng 2004;30(8):563–72.
[30] Thomas Bäck, Ulrich Hammel, Hans-Paul Schwefel. Evolutionary Computation: Comments on the History and Current State. IEEE Transactions on Evolutionary Computation 1997;1(1):3–17.
[31] Peterson JL. Petri Net theory and the modelling of systems. Englewood Cliff, N.J.: Prentice-Hall; 1981.
[32] Gemeinder M, Gerke M. GA-based path planning for mobile robot systems employing an active search algorithm. Appl Soft Comput 2003;3:149–58.

J.C. Mohanta is presently working as an Engineering Officer in Mechanical Engineering Division of Central Power Research Institute, Bangalore. He is a member of The Institution of Engineers (India) and an editorial board member of the International Journal IJAICR. He is actively involved in research work focussing on navigational behaviour of mobile robots using various AI techniques.

D.R. Parhi received his first Ph.D. in Mobile Robotics from Cardiff School of Engineering, UK and second Ph.D. in Vibration Analysis of Cracked structures from Sambalpur University. He has 17 years of research and teaching experience. Presently, he is engaged in Mobile robot Navigation research, and a faculty member in Department of Mechanical Engineering, National Institute of Technology Rourkela.

S.K. Patel is now working as Associate Professor in Mechanical Engineering Department of National Institute of Technology, Rourkela. He graduated in Mechanical Engineering from National Institute of Technology, Rourkela. Initially he served with Hindustan Aeronautics Limited, Koraput Division, Sunabeda. He obtained M. Tech degree from Indian Institute of Technology Madras and Ph.D. degree from Indian Institute of Technology, Kharagpur.