Chaotic Bee Swarm Optimization Algorithm for Path Planning of Mobile Robots

Jiann-Horng Lin and Li-Ren Huang

Department of Information Management
I-Shou University, Taiwan
jhlin@isu.edu.tw

Abstract: - This paper is based on swarm intelligence and chaotic dynamics for learning. We address this issue by considering the problem of path planning for mobile robots. Autonomous systems assume intelligent behavior with ability of dealing in complex and changing environments. Path planning problem, which can be studied as an optimization problem, seems to be of high importance for arising of intelligent behavior for different real-world problem domains. In recent years, swarm intelligence has gained increasingly high interest among the researchers from different domains such as commerce, science and engineering. Bees' warming about their hive is an example of swarm intelligence. It's particularly fitting apply methods inspired by swarm intelligence to sundry optimization problems, and chiefly if the space to be explored is large and complex. In this paper, we propose a new approach to the problem of path planning for mobile robots based on an improved artificial bee colony optimization combined with chaos. In artificial bee colony optimization, chaos is hybridized to form a chaotic bee swarm optimization, which reasonably combines the population-based evolutionary searching ability of artificial bee colony optimization and chaotic searching behavior. The track of chaotic variable can travel ergodically over the whole search space. In general, the chaotic variable has special characters, i.e., ergodicity, pseudo-randomness and irregularity. Generally, the parameters of the artificial bee colony optimization are the key factors to affect the convergence of the artificial bee colony optimization. In fact, however, it cannot ensure the optimization's ergodicity entirely in phase space because they are absolutely random in the traditional artificial bee colony optimization. Therefore, this paper provides a new method that introduces chaotic mapping with certainty, ergodicity and the stochastic property into artificial bee colony optimization so as to improve the global convergence.

Key-Words: path planning problem, artificial bee colony, chaotic dynamics.

1 Introduction

One of the final goals in Robotics is to create autonomous robots. Such robots will accept high-level descriptions and tasks and will perform them without further human intervention [1]. This work presents a approach where the user specifies to the robot only what is wanted for robot to do (find collision free path through space containing obstacles) rather than saying anything about how to do it. The path planning problem is a well studied problem of robot intelligence, to which different approaches were applied, for example: Neural Networks, Potential Fields, Genetic algorithms, Particle Swarm Optimization method and other. It expresses at the same time the basic problem of robot intelligence providing convenient model for evaluation of novel approaches in artificial intelligence.

Optimization methods can usually be classified as follows:

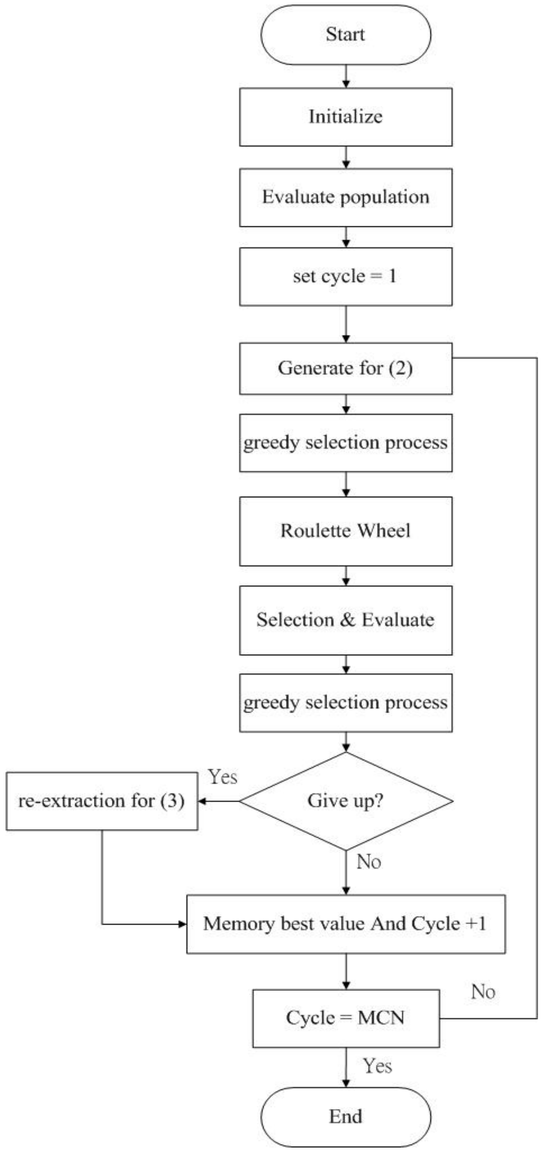
1. Direct search: direct search methods use only objective function and constraint values.
2. Gradient based search: Gradient based search techniques require derivative information of the function and constraints.

Since derivative information is not used, these methods generally require huge number of iterations for convergence, but are at the other hand applicable to very wide problem space. Swarm intelligence is an emerging area of artificial intelligence. It is concerned with modeling of social interactions between social beings, bees, primarily ants, and birds in the recent time [2]. The path planning problem has a important task for mobile robots is autonomous navigation, where the robot travels between a starting point and a goal point without the need for human intervention [1]. While basic information may be available to the robot about the navigation area bound, unknown obstacles may exist within the navigation area. There are common several kinds of the path planning problems as follow:

a) A grid robot is a virtual agent that is located in an environment made of square grids. In cellular representation for path planning problems is constructed of single descriptors [3,4]. The simple path planning problem can refer to some simple rules to produce a map. They are used to construct the path planning problem according to the following rules.
1) Each single descriptor places its first obstruction at (x, y) and follows its pattern thereafter.
2) If an obstruction's position is outside the grid it is not placed and placement of obstructions for the current single descriptor are terminated.
3) The amount of obstructions placed for a single descriptor may not exceed is own maximum number of obstructions.
4) If placement of an obstruction is attempted where one already has been placed then processing of the obstructions for the current single descriptor not only continues but that obstruction is not counted against the single descriptor's total maximum amount of blocks.
5) If the amount of obstructions placed reaches the maximum permitted for a Path planning problem then obstacle placement terminates [5].
b) In the environment, the second problem is a map of each frame to set a weight [0,-1], If weight higher representative the navigation is hard, again penetrates fitness function to appraise.
c) This paper specifies that how third method is to be applied.
The remainder of the study is organized as follows. Section 2 gives a representation for Artificial Bee Colony algorithm used to compute the fitness of the Path Planning Problems. Section 3 specifies the evolvable grid robot path planning problems.

2 Artificial Bee Colony Algorithm

The Artificial Bee Colony, introduced by Dervis Karaboga in 2005[6]. The Artificial Bee Colony can solve Constrained Optimization Problem. Engineering design, structural optimization, economics, VLSI design, allocation and location problems are just a few of the scientific fields in which Constrained Optimization problems are frequently met [7,8]. The colony of artificial bees consists of three groups of bees: employed bees, onlookers and scouts in the Artificial Bee Colony.
Employed bees: a bee going to the food source visited by itself previously is named an employed bee.
Onlookers: A bee waiting on the dance area for making decision to choose a food source.
Scout bees: A bee carrying out random search is called a scout.
The one section of the employed artificial bees and another section includes the onlookers. The number of employed bees is equal to the number of food sources. If food source is exhausted by the employed bees, then onlooker bees becomes a scout. Detailed flow chart of the ABC algorithm is given below:



In artificial bee colony algorithm, the position of a food source represents a possible solution to the optimization problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. The number of the employed bees or the onlooker bees is equal to the number of solutions in the population. The Artificial Bee Colony generates a randomly distributed initial population of *TS* solutions. That "i" is a representation of ($i=1,2,\ldots,TS$) solution and "j" is a ($j=1,2,\ldots,ND$) D-dimensional . The each solution is a x_i ($i=1,2,\ldots,TS$) D-dimensional vector. The *TS* denotes the size of population. First step, After initialization, the population of the positions (solutions) is subjected to repeated cycles of the (*cycle=1,2,…,MC*), search processes of the employed bees, the onlooker bees and scout bees. The *MC* is amount of cycles. An artificial employed or onlooker bee probabilistically produces a modification on the position (solution) in its memory for finding a new food source and tests the nectar number of (fitness value) of the new source (new solution). If nectar number of the new one is higher than that of the previous one, the bee forgets the old one and memorizes the new position. Otherwise, in her memory, she keeps the position of the previous one. They share the nectar information of the food sources (solutions) and their position information with the onlooker bees on the dance area, After all employed bees complete the search process. As in the case of the employed bee, it produces a modification on the position (solution) in its memory and checks the nectar amount of the candidate source (solution), and onlooker bee evaluates the nectar information taken from all employed bees and then chooses a food source with a probability related to its nectar amount. If nectar number of the new one is higher than that of the previous one, the bee forgets the old one and memorizes the new position. An artificial onlooker bee chooses a food source depending on the probability value of roulette wheel method associated with that food source, P_i, calculated by the following expression (1):

$$P_i = \frac{f_i}{\sum_{1}^{TS} f_n} \qquad (1)$$

That f_i is the fitness value of the solution I evaluated by its employed bee. It's proportional to the nectar amount of the food source in the position *i*. The *TS* is the amount of food sources which is equal to the number of employed bees. In the memory, so as to produce a candidate food position from the old one, the Artificial Bee Colony uses the following expression (2):

$$V_{i,j} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \qquad (2)$$

In the expression, $k \in \{1, 2,...,TS\}$ and $j \in \{1, 2,...,ND\}$ are indexes of randomly chosen. This k is *SN* of randomly chosen, but it has to be different from *i*. Where Φ_{ij} is random number between [-1,1]. It controls the production of neighbor food sources nearby x_{ij} and represents the comparison of two food positions visually by a bee. If we searched approaches to the optimum solution in the search space, then step length is adaptively reduced. If its value (Φ_{ij}) produced by this operation exceeds its predetermined limit ([-1,1]), the parameter can be set to an acceptable value and the value of the parameter is set to its limit value. By the scouts, the food source of which the nectar is abandoned by the bees is replaced with a new food source. This is simulated by producing a position randomly and replacing it with the abandoned one in the Artificial Bee Colony. In Artificial Bee Colony, if a position can not be improved further through a redetermined number of cycles, then food source is assumed to be abandoned. The value of predetermined number of cycles is an important control parameter in the Artificial Bee Colony algorithm. This is called "limit" for abandonment. Assume that the abandoned source is x_{ij} *(i=1,2,...,TS , j=1,2,…,ND)* and scout discovers a new food source to be replaced with x_i. We use the following expression (3):

$$x_i^j = x_{\min}^j + \phi \times (x_{\max}^j - x_{\min}^j) \qquad (3)$$

After each candidate source position V_{ij} is produced, then evaluated by the artificial bee, its performance is compared with that old one. Where Φ_{ij} is random number between [0,1]. If the new food has an equal or better nectar than the old one, it is replaced with the old one in the memory. Otherwise, the old one is retained in the memory. This greedy selection mechanism is employed as the selection operation between the old and the candidate one, that is to say. In any case, It's clear from the above explanation that there are four sections used in the Artificial Bee Colony: The number of food sources which is equal to the number of employed or onlooker bees (*TS*), the value of limit, the maximum cycle (*MC*).

3 Chaotic Bee Swarm Optimization

Nonlinear dynamics of complex systems is concerned primarily with making predictions about the behavior of systems which evolve in time, as parameters which control the system, and the initial state of the system itself, are varied. The dynamics of numerous natural and artificial systems is nonlinear, thus often give rise to chaos. The existence of chaos in these nonlinear dynamical systems is a well established fact. It has been recognized that the complicated chaotic behaviors may result from nonlinear systems even with simple structures. A chaotic system tracks a trajectory that is quite complex but not entirely random. Chaos is the apparently irregular motion that is, in reality, nonlinear but deterministic. Sensitive dependence of a dynamical system's temporal evolution to perturbations of the initial conditions is ubiquitous in nonlinear dynamics. Two identical chaotic systems starting at nearly the same point follow trajectories that divert rapidly from each other and become quickly uncorrelated. Here we propose an improved Artificial Bee Colony optimization combined with chaos. Chaos is a kind of characteristic of nonlinear systems, which is a bounded unstable dynamic behavior that exhibits sensitive dependence on initial conditions and includes infinite unstable periodic motions. Although it appears to be stochastic, it occurs in a deterministic nonlinear system under deterministic conditions. Due to the easy implementation and special ability to avoid being trapped in local optima, chaos has been a novel optimization technique and chaos based searching algorithms have aroused intense interests. Firstly, adaptive weight factor is introduced in Artificial Bee Colony optimization to efficiently balance the exploration and exploitation abilities. Secondly, Artificial Bee Colony optimization with adaptive weight factor and chaos are hybridized to form a chaotic bee swarm optimization, which reasonably combines the population-based evolutionary searching ability of Artificial Bee Colony optimization and chaotic searching behavior. The track of chaotic variable can travel ergodically over the whole search space. In general, the chaotic variable has special characters, i.e., ergodicity, pseudo-randomness and irregularity. Generally, the parameters of the Artificial Bee Colony optimization are the key factors to affect the convergence of the Artificial Bee Colony optimization. In fact, however, it cannot ensure the optimization's ergodicity entirely in phase space because they are absolutely random in the traditional Artificial Bee Colony optimization. Therefore, this paper provides a new method that introduces chaotic mapping with certainty, ergodicity and the stochastic property into Artificial Bee Colony optimization so as to improve the global convergence. To enrich the searching behavior and to avoid being trapped into local optimum, chaotic dynamics is incorporated into the parameters of the Artificial Bee Colony optimization. In this paper, the well-known logistic map which exhibits the sensitive dependence on initial conditions is employed to generate the chaos sequence for the parameters of Artificial Bee Colony optimization:

$$\phi(t+1) = 4.0 \times \phi(t) \times (1-\phi(t)), \quad 0 \le \phi(0) \le 1. \qquad (4)$$

Furthermore, simulated annealing has been applied with success to many numerical and combinatorial optimization problems in recent years. Simulated annealing has a rather slow convergence rate, however, on some function optimization problems. In this paper, by introducing chaotic dynamics to simulated annealing, we propose an improved Artificial Bee Colony optimization with chaotic annealing technique. The distinctions between chaotic annealing and simulated annealing are chaotic initialization and chaotic sequences replacing the Gaussian distribution. The key idea of chaotic annealing is to take full advantages of ergodic property and stochastic property of chaotic system and replace the Gaussian distribution by chaotic sequences in simulated annealing.

4 Path Planning Problem

The path planning problem is of major interest for industrial robotics. It can be seen as a vehicle that needs the power of generating collision free paths that take the robot from a starting position to a final position. It needs to avoid obstacles present in the environment. It has to understand enough relevant information of his current position relative to starting position and final position, and of the state of the environment or terrain that surrounds it. It's obvious that one of the most important motives is generate an appropriate path for a robot to follow. It's to help it avoid obstacles along the way, for this motive an appropriate representation of the terrain is needed generating a sufficiently complete map of the given surroundings that the robot will encounter along its route[9]. Such robots will accept high-level descriptions and tasks and it will execute them without further human intervention. Collision free path planning is a well studied problem of robot intelligence, to which different approaches were applied, e.g. Genetic algorithms[10,11], Neural Networks, Potential Fields[12], Particle Swarm Optimization[13] method and other in recent years. A easy version of the problem, that of planning the motion of a point robot among 3-dimensional polyhedral obstacles. It has been proved to be NP-complete[14]. In general, the complexity of the problem is exponential in the amount of degrees of freedom of the robot, and polynomial in the number of obstacles in the environment. Therefore, finding a path for a robot with many degrees of freedom(more than five) in an environment with several obstacles is a very difficult problem. Unlucky, many realistic industrial problems deal with robots of at least six degrees of freedom and hundreds of obstacles. Even worse, usually the environment is dynamic in the sense that some of the obstacles may move, therefore further requiring that new paths be found in very short computing times[15]. The basic systems that operate in an Autonomous Mobile robot are: Vehicle Control, Sensor and Vision, Navigation and Path Planning. The path planning problem when analyzed with the point-to-point (starting position to a final position) technique. It comes down to finding a path from one point to another. Literature is rich in approaches to solve mobile robots trajectory planning in presence of still and/or animated obstacles[16,17,18]. One of the most common planning methods is the artificial potential area[19]. However, this method gives only one trajectory solution that may not be the slighter trajectory in a still environment. The main difficulties in determining the optimum trajectory are due to the fact that analytical methods are very complex to be used in real time, and the searching enumerative methods are excessively unaffected by the size of the searching space. In this paper, the trajectory planning is the main aspect in the movement of a mobile robot. The problem of a mobile robot trajectory planning is typically defined as follows: given a robot, and the environment depiction, a trajectory is planned between two definite locations which is free of collisions and is suitable in a certain performance criteria [20]. In the present study, it's considered a 2-D mobile robot trajectory planning problem, in which the position of the mobile robot R is represented by Cartesian coordinates (x, y) in the xy plan. The start and final points of the robot are (x_0, y_0) and (x_{fp}, y_{fp}), where fp is a design parameter. The start point is always (x_0, y_0). Only the trajectory planning problem is sympathized in this paper, the robot control problem is not the focus of this paper. Nevertheless, details of the robots movement equations can be found[21]. It's assumed that the obstacles are circular in the robot's moving plan. otherwise, the hypothesis that the free 2-D space is connected and the obstacles are limited in size and doesn't overlap the destiny point is true. The optimization problem formulated make up a discrete optimization problem, where the objective function f(x, y), which is the connection among the technique used for optimization and the environment, aims to minimize the total trajectory percurred by the mobile robot. It's ruled by :

$$f(x,y) = w_1 \times E_{obj} + w_2 O_n \tag{5}$$

$$E_{obj} = \sum_{i=0}^{f_p} \sqrt{(x(i+1)-x(i))^2 + (y(i+1)-y(i))^2} \tag{6}$$

where w_1 and w_2 are weighted factors, and E_{obj} represents the Euclidian distance between the start and the final point, O_n denotes the number of obstacles prevented by the robot movement following the planned trajectory, and fp is the number of points where a trajectory change occurs (project parameter in this article). It's noticed by the equation (5) that a w_1 term exists, it's an weighting (penalty) term for impractical solutions, meaning, the trajectory that intercepts obstacles. In this case, the fitness function to be evaluated by optimization approaches of this paper aims to maximize

$$fitness = \frac{fix_c}{f(x,y) + fix_\varepsilon} \tag{7}$$

Where fix_c and fix_ε are scale invariable.

5 Simulation Results

The simulated cases and the results achieved by the ABC. The environment used for the trajectory planning is a 100x100 meters field. The search interval of the parameters is x_i [0,100] meters and y_i [0,100] meters, where $i=1,..,fp$. About the fitness it's adopted w_1=1, w_2=200, fix_c =100 and fix_ε=1x10-6. The Artificial Bee Colony parameters are population size 100, maximum number of iterations 100.

In Table 1 are presented the positions of the centers (x, y) of the circular obstacles and their respective radius (in meters) of case. The results obtained with the bacteria colony are restricted to p=3. (x_0, y_0) is the starting point and (x_{100}, y_{100}) is terminal point, and seven circles are obstacles in the Fig. 1.

The experiment has achieved a feasible solution, the best trajectory was achieved by Artificial Bee Colony. In Fig. 2 was best results achieved by Artificial Bee Colony. Fig. 3 showed the evolution of fitness function.

The best result was achieved for study case by the coordinates: P1 = (19.1891, 22.7910); P2 = (73.2502, 46.1828); P3 = (73.9433, 47.0942).

Table 1: Definition of obstacles for the case study.

Obstacle number	Position (X,Y)	radius
1	(6,30)	10
2	(19,9)	8
3	(50,1)	8
4	(93,37)	9
5	(58,62)	17
6	(38,94)	9

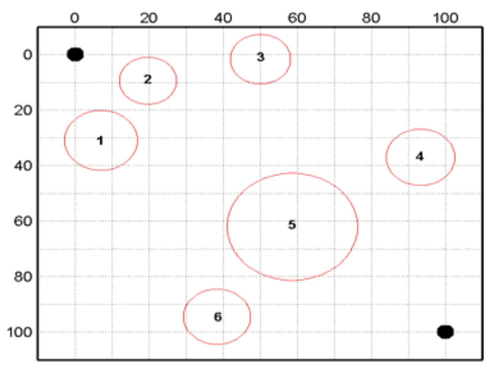


Fig 1. Chart of Initial Environmental.

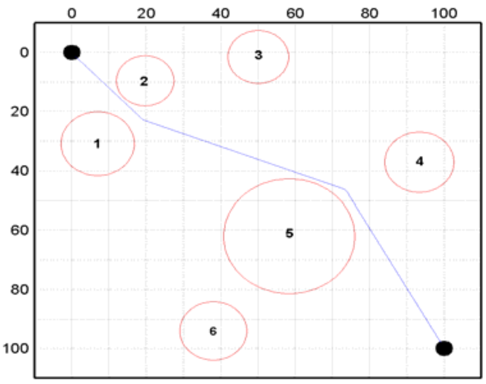


Fig 2. Best result achieved by ABC.

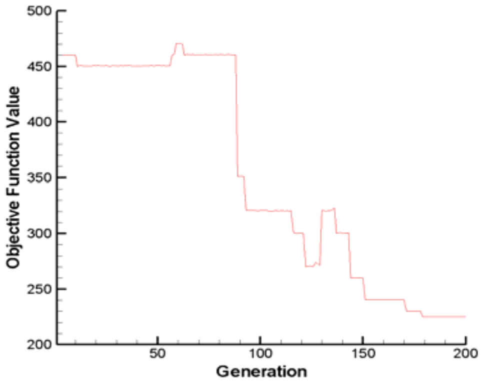


Fig 3. Fitness function evolution

6 Conclusion

Recently, the Artificial Bee Colony algorithm was just developed gradually, and it can be used to solve several optimal problems. The Artificial Bee Colony algorithm for path planning problems has been introduced and its performance has been compared with that of the state-of-art algorithms[5]. It has been concluded that the Artificial Bee Colony algorithm can be efficiently used for solving path planning problems in this paper. The performance of the Artificial Bee Colony algorithm can be also tested for real robotics problems existing in the literature. For some Artificial Intelligence issues, it is even more efficient than other algorithms such as Genetic Algorithm, and Particle Swarm Optimization. To sum up, We think Artificial Bee Colony algorithm will be applied in more domains in the future.

References:

[1] J. C , Latombe, Robot motion planning, Kluwer Academic Publishers, Boston. 1991.

[2] P. Curkovic, B. Jerbic, Honey-bees optimization algorithm applied to path planning problem, International Journal of Simulation Modelling, pp. 137-188, 2007.

[3] Gruau, F., Neural Network Synthesis using Cellular Encoding and the Genetic Algorithm. PhD thesis, 1994.

[4] Gruau, F., Automatic definition of modular neural networks. Adaptive Behaviour, 3(2):151–183, 1995.

[5] Daniel, A., Ashlock, Theodore, W., Manikas, Kaveh Ashenayi., Evolving A Diverse Collection of Robot Path Planning Problems, IEEE Congress on Evolutionary Computation, pp. 1837-1844, 2006.

[6] D. Karaboga, An Idea Based On Honey Bee Swarm For Numerical Optimization, Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[7] D. Karaboga, B. Basturk, Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems, IFSA , LNCS: Advances in Soft Computing: Foundations of Fuzzy Logic and Soft Computing, Vol: 4529/2007, pp: 789-798, 2007.

[8] D. Karaboga, B. Basturk, A Powerful and Efficient algorithm for numerical function optimization: Artificial Bee Colony (ABC) algorithm, Journal of Global Optimization, November, Volume:39, Issue:3,pp:459-171, 2007.

[9] O. Castillo. L, Trujillo. Multiple objective optimization genetic algorithms for path planning in autonomous mobile robots, International journal of computers, Vol. 6, No. 1, 48-63, 2005.

[10] Xiao, J.; Michalewicz, Z.; Zhang, L.; Trojanowsky, K., Adaptive evolutionary planner/navigator for mobile robots, IEEE Transactions on Evolutionary Computation, Vol. 1,18-28, 1997.

[11] Castillo, O.; Trujillo, L., Multiple objective optimization genetic algorithms for path planning in autonomous mobile robots, International journal of computers, systems and signals,Vol. 6, No. 1, 48-63, 2005.

[12] Bekey, G. A.; Goldberg, K. Y., *Neural networks in robotics*, Kluwer Academic Publishers, Boston, 1993.

[13] Chen, M.; Yangmin, L., Smooth formation navigation of multiple mobile robots for avoiding moving obstacles, International Journal of Control, Automation and Systems, Vol. 4,No. 4, 466-479, 2006.

[14] Canny, J., The Complexity of Robot Motion Planning. MIT Press, 1988.

[15] Mazer, E., Ahuactzin, J.M., and Bessiere, P., The Ariadne's Clew Algorithm, Journal of Artificial Intelligence Research, pp. 295-316 , 1998.

[16] Bennewitz, M., Burgard, W., and Thrun, S., “Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots,” Robotics and Autonomous Systems, vol. 41, no. 2, pp. 89-99. 2002.

[17] Melchior, P., Orsoni, B., Lavaialle, O., Poty, A., and Oustaloup, A., “Consideration of obstacle danger level in path planning using A* and fastmarching optimization: comparative study,”Signal Processing, vol. 83, pp. 2387-2396. 2003.

[18] Tu, J., and Yang, S.X., “Genetic algorithm based path planning for a mobile robot,” Proceedings of the IEEE International Conference on Robotics & Automation, Taipei, Taiwan, pp. 1221-1226. 2003.

[19] Tsuji, T., Tanaka, Y., Morasso, P. G., Sanguineti, V., and Kaneko, M., “Bio-mimetic trajectory generation of robots via artificial potential field with time base generator,” *IEEE Transactions on Systems, Man and Cybernetics - Part C*, vol. 32, no. 4, pp. 426- 439, 2002.

[20] Xiao, J., Michalewicz, Z., Zhang, L., and Trojanowski, K., “Adaptive evolutionary planner/navigator for robots,” IEEE Transactions on Evolutionary Computation, vol. 1, no. 1, pp. 18-28, 1997.

[21] Fujimori, A., Nikiforuk, P.N., and Gupta, M.M., “Adaptive navigation of mobile robots with obstacle avoidance,” IEEE Transactions on Robotics and Automation, vol. 13, no. 4, pp. 596- 602, 1997.