Proceedings of the 2004 IEEE
International Conference on Robotics & Automation
New Orleans, LA • April 2004

A Knowledge Based Genetic Algorithm for Path Planning of a Mobile Robot

Yanrong Hu and Simon X. Yang
Advanced Robotics and Intelligent Systems (ARIS) Lab
School of Engineering, University of Guelph
Guelph, Ontario, Canada N1G 2W1
Email: syang@uoguelph.ca

Abstract— **In this paper, a knowledge based genetic algorithm (GA) for path planning of a mobile robot is proposed, which uses problem-specific genetic algorithms for robot path planning instead of the standard GAs. The proposed knowledge based genetic algorithm incorporates the domain knowledge into its specialized operators, where some also combine a local search technique. The proposed genetic algorithm also features a unique and simple path representation and a simple but effective evaluation method. The knowledge based genetic algorithm is capable of finding an optimal or near-optimal robot path in both complex static and dynamic environments. The effectiveness and efficiency of the proposed genetic algorithm is demonstrated by simulation studies. The irreplaceable role of the specialized genetic operators in the proposed GA for solving robot path-planning problem is demonstrated by a comparison study.**

I. INTRODUCTION

Path planning is an important issue in mobile robotics. In an environment with obstacles, path planning is to find a suitable collision-free path for a mobile robot to move from a start location to a target location. Very often this path is highly desirable to be optimal or near-optimal with respect to time, distance or energy. Distance is a commonly adopted criterion. Robot path planning has been an active research area, and many methods have been developed to tackle this problem [1], such as global C-space methods [2], [3], potential field methods [4], [5], and neural networks approaches [6], [7]. Each method has its own strength 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.

There is no doubt that path planning can be viewed as an optimization problem (e.g., shortest distance) under certain constraints (e.g., the given environment and collision-free condition). Since the appearance of genetic algorithms (GAs) in 1975 [8], GAs have been used in solving many optimization problems successfully. GAs are stochastic search techniques analogous to natural evolution based on the principle of survival of the fittest [9], [10]. 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 fitter 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 feature of parallel search and the ability of quickly locating high performance region [9] contribute to the success of GAs on many applications.

It is not surprising that researchers applied GAs on path planning for mobile robots [11]–[13]. 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. Sugihara *at al.* [11] proposed a genetic algorithm for path planning with fix-length binary string chromosomes based on cell representation of mobile robot environment. 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. A similar approach was also proposed by Tu *et al* [14], where no obvious improvement is made in spite of using variable-length chromosomes instead of fix-length chromosomes. It takes hours to evolve a solution due to its inefficiency.

The classical GAs use 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 GAs [10], [15]. 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. Shibata *et al.* [16], [17] proposed a genetic algorithm based on MAKLINK graph environment representation [18]. 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, which can be very time consuming. Both Hocaolu *et al* [19] and Xiao *et al* [20]designed specialized genetic operators with some heuristic knowledge. In [19], a multi-paths planning algorithm based on an iterative multi-resolution path representation was proposed. A path is represented by a hierarchically ordered set of vectors that define path vertices generated by a modified Gram-Schmidt orthogonalization process [21], [22]. Xiao *et al.* [20] proposed an evolutionary planner for both on-line and off-line planning. However, both approaches are relatively complicate on problem representation, evaluation, or GA structure.

In this paper, a knowledge based genetic algorithm is proposed. It uses a simple yet effective path representation that combines grids and coordinates representations. Unlike other grid methods, the grids adopted here do not limit movement of the path, but simplify the chromosome structure and genetic operation by discretizing the environment. This approach makes it possible to have one number for each gene and to use integer numbers instead of real numbers in chromosomes. The proposed GA also has six knowledge based genetic operators. Problem-specific genetic operators are not only designed with domain knowledge, but also incorporate small-scale local search that improves efficiency of the operators. A relatively simple but effective evaluation method is applied to both feasible and infeasible solutions. The proposed GA is suitable for both static and dynamic environments. The effectiveness of the knowledge based GA for mobile robot path planning is demonstrated by simulation studies.

II. The Proposed Knowledge Based GA for Path Planning

The proposed genetic algorithm features its simple and unique problem presentation, its effective evaluation method and its knowledge based genetic operators. The detailed description of each element is presented below.

A. Problem Representation

Representation is a key issue in the work of GAs. The proposed GA uses a simple yet effective path representation. The mobile robot environment is represented by orderly numbered grids, each of which represents a location in the environment. The boundary of obstacles is formed by their actual boundary plus minimum safety distance considering the size of the mobile robot, which makes it possible to treat the mobile robot a point in the environment. As an example, Fig. 1 shows an environment with 10×10 grids applied on. A potential path is formed by line segments connecting the nodes falling on the grids with different numbers. Thus a path is encoded as a sequence of grid numbers starting from the source and ending at the target with a various number of intermediate nodes (Fig. 2). A feasible path is a collision free path, i.e. no nodes fall on any obstacle, or no any of line segments of a path intersects an obstacle. The length of a chromosome is variable and between 2 and maximum length N_{max}. Such a grid representation is different from the one that usually uses grids to limit the movement of a path to be one of its eight adjacent cells and uses relative directions to represent a path [11]. The proposed path representation is more like a coordinate representation, but differs by discretization and using integer numbers instead of coordinates (x, y).

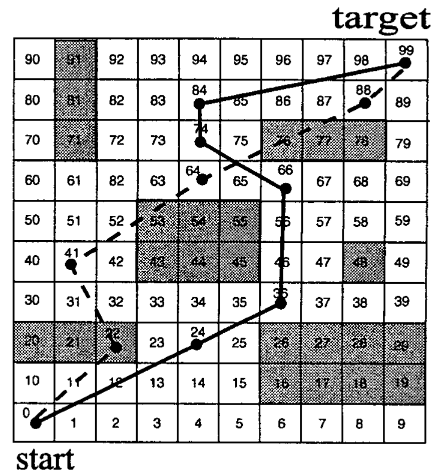

Fig. 1. Mobile robot environment and path representation. Solid line: a feasible path; dashed line: an infeasible path.

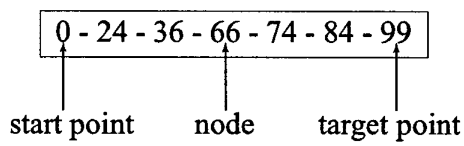

Fig. 2. A sample chromosome: a path represented by nodes falling on grids with different numbers.

B. Evaluation

A path can be either feasible (collision free) or infeasible because intermediate nodes can fall on any of the grids. The evaluation should be able to distinguish feasible and infeasible paths and tell the difference of path qualities within either category. The evaluation function is presented below:

$$F_{\text{cost}} = \sum_{i=1}^{N}(d_i + \beta_i C), \tag{1}$$

where N is the number of line segments of a path, d_i is the Euclidean distance of the two nodes forming the line segment, C is a constant, β_i is the coefficient denoting depth of collision, and its definition is given as

$$\beta_i = \begin{cases} 0 & \text{if the } i\text{th line segment is feasible} \\ \sum_{j=1}^{M} \alpha_j & \text{if the line segment intersects obstacle(s)} \end{cases}, \tag{2}$$

where M is the number of obstacles the line segment intersects. α_j is determined by considering how deep a line segment intersects an obstacle j. It is defined as the shortest moving distance for escaping the intersected obstacle. This evaluation gives penalty to infeasible paths, but still keep them in the population because they might become good feasible solutions after certain genetic transformations. Besides, this evaluation allows some overlap between fitnesses of feasible and infeasible solutions because a very poor feasible path is not necessarily better than a very good near-feasible path in the sense of evolving solutions. It is beneficial to give more chance to some good infeasible solutions that are easily to be evolved to good solutions. To save computational time, some information obtained by the evaluation needs to be recorded so that later on it can be used by some specialized genetic operators as heuristic knowledge without re-calculation. The information includes feasibility (feasible or infeasible, node-infeasible or line-infeasible), number of infeasible nodes or line segments, and which obstacle(s) a path intersects.

C. Genetic Operators

Two classical genetic operators: crossover and mutation are not applicable for the problem here. They have to be tailored for the path planning problem and adopted problem representation. Besides, to make the genetic algorithm more effective, four specialized operators are designed to make use of available problem-specific knowledge including knowledge of the environment. These six operators are introduced as following and depicted in Fig. 3.

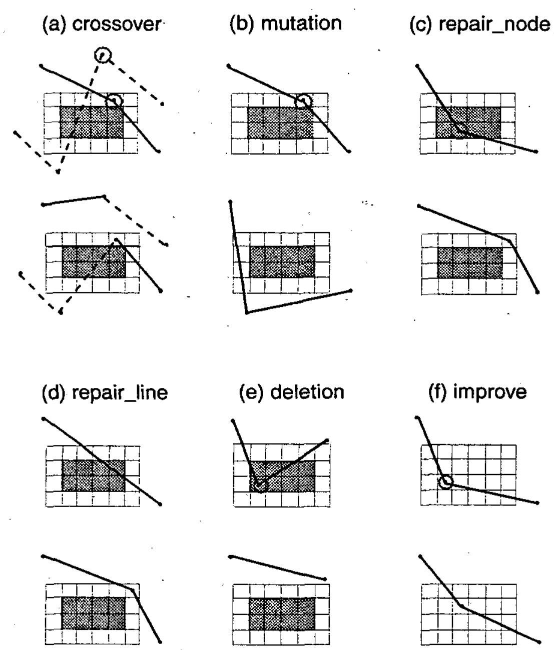

Fig. 3. Six specialized genetic operators that incorporate problem-specific knowledge

Crossover is the operator that randomly choose a node from Parent 1 and the other node from Parent 2. Exchange the part after these two nodes. Check the two offspring, and delete the part between two same nodes if it happens. The choice of different crossover sites in different parents can increase the variability of chromosome length, which benefits exploration of the solution space.

Mutation is that randomly choose a node and replace it with a node that is not included in the path. Mutation is served as a key role to diversify the solution population. Therefore, it is not necessary that a solution is better after it is mutated.

Node_repair is used to move a node falling on an obstacle out of the obstacle and to a best grid around the obstacle. To locate the best grid, a small-scaled local search in the neighbors of the obstacle is applied.

Line_repair is used to repair an infeasible line segment by inserting a suitable node between the two nodes of the segment. Again, to locate a best node, similar local search is applied in the all of neighboring grids of the intersected obstacle.

Deletion is applied to both feasible and infeasible path. Randomly choose one node, check its two adjacent nodes, and connected segments, if the deletion of the chosen node is beneficial (turn the infeasible to the feasible, reduce the cost), delete the node.

Improvement is designed for feasible solutions. Randomly chose one node, do a local search in the neighboring grids of the node, move to a best grid. This operator is used for fine tuning of a feasible solution.

These operators are very necessary to evolve feasible and good quality solutions. The firing of these operators depends on two criteria: probability and heuristic knowledge (e.g., if *feasible* then *improve*). The important role of these operators is discussed later.

D. Outline of the Knowledge Based Genetic Algorithm

An outline of the proposed knowledge based genetic algorithm is given in Fig. 4. Initial solutions are generated randomly and are evaluated by the fitness function in Eqn. (1). Two parents are selected according to some selection mechanism. Then one or more genetic operators are selected and applied to the parents according to some probabilities and heuristic knowledge. The whole generation is replaced by children. The best solution so far is updated in each generation, and it will be the final solution when some stop criterion is satisfied. The stop criterion can either be that the preset maximum generation is exceeded, or that the best solution remains unchanged for certain generations. The algorithm is also suitable for dynamic environment. It checks sensing information 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. To increase diversity of the population at the moment, mutation with higher probability is applied to the current population.

III. Simulations

The effectiveness of the proposed genetic algorithm is demonstrated by simulations. In the following simulations, parameter setting for the genetic algorithm is: population size = 50, probability for mutation per chromosome is 0.2, and 0.9 for all the rest operators. Tournament selection and elitism are applied. For simplicity, 16×16 grids is applied to the environments. All the simulations are conducted on a a Pentium III PC (933MHz).

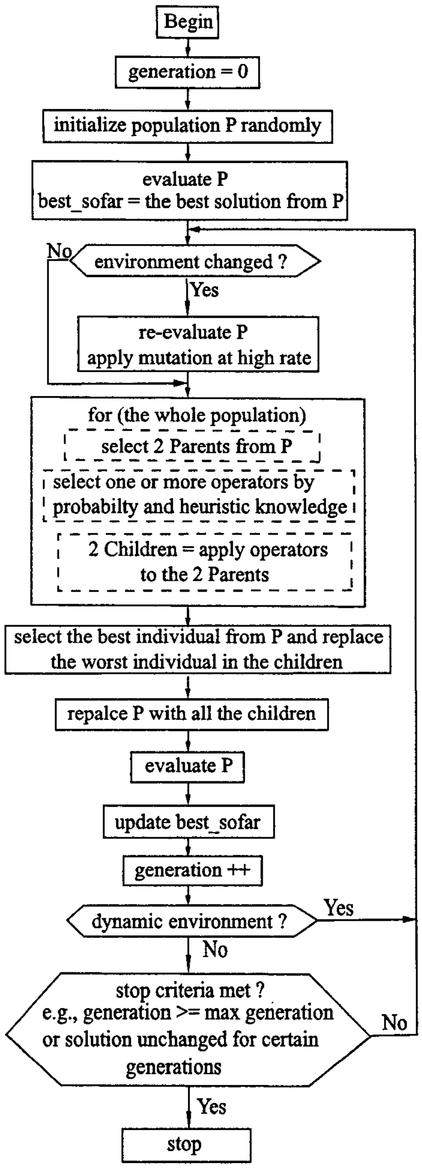

Fig. 4. Outline of the knowledge based genetic algorithm for path planning of a mobile robot.

A. Path Planning in an Environment with a U-shape Obstacle

The proposed genetic algorithm can easily deal with U-shape obstacles. Fig. 5 shows an example. The best solution in the initial population in Fig. 5(a) is far from feasible. Fig. 5(b) shows the best solution after eight generation's evolution, which is feasible but not optimal. The genetic algorithm continues to evolve better solutions (Fig. 5(c)) until the optimal solution (Fig. 5(d)) is found in generation 30.

B. Path Planning in Complex Environments

In this simulation, the proposed genetic algorithm is applied to different mobile robot environments with different obstacle layouts. Fig.s 6 and 7 show that the GA is capable of obtaining an optimal/near-optimal collision free path from randomly generated paths. For the environment in Fig. 6, for 20 runs, the average cost is 30.85 with 0.67 Standard Deviation, and the average generation number for obtaining the results in the 20 runs is 257 with 123 Standard Deviation. Fig. 7 shows the result of the GA on a more challenging mobile robot environment.

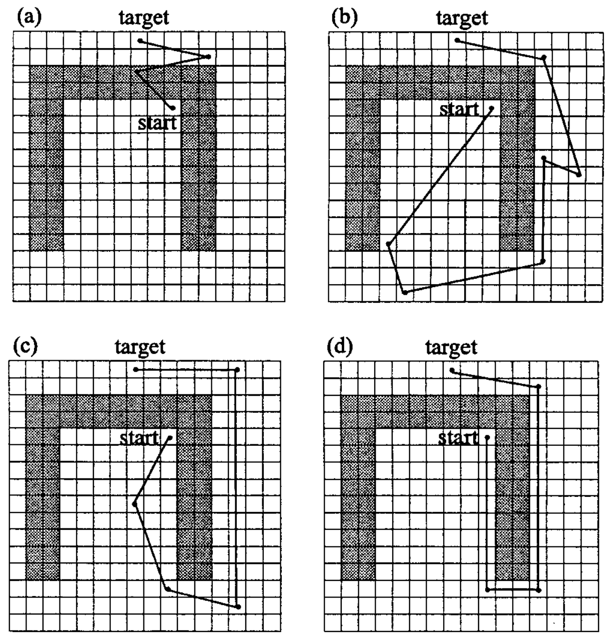

Fig. 5. One typical run of path planning in a U-shape obstacle environment. (a) The best initial path (cost: 80.59); (b) The best solution in generation 8 (cost: 42.02); (c) The best solution in generation 22 (cost: 33.98); (d) The optimal path (cost: 29.10) obtained in generation 30.

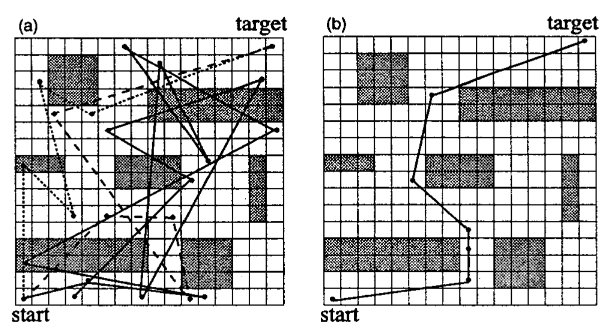

Fig. 6. Path planning in a complex environment. (a) Three randomly generated initial paths; (b) The path obtained by the GA in one typical run.

C. Path Planning in a Dynamic Environment

The proposed genetic algorithm can not only deal with complex static environment, but also be suitable for dynamic

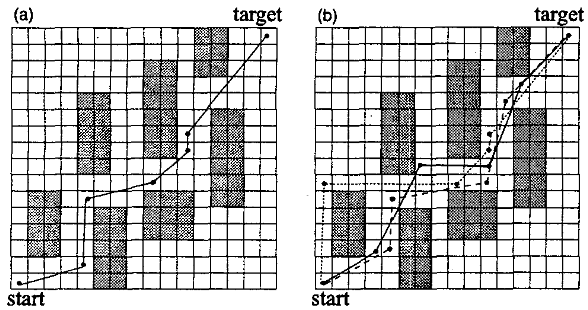

Fig. 7. Path planning in another complex environment. (a) The path obtained by the GA in one typical run; (b) Three alternative paths obtained by the GA from different runs.

environment. When the environment changes, information of the obstacles is updated. The algorithm re-evaluates the current population according to the new information. Therefore, the costs of the solutions in the current population are updated. The quality and feasibility of a path is determined according to the new environment. The original best individual and the better ones may become solutions having bad quality. The GA starts to evolve the population until a new best solution is obtained according to the new environment.

A simulation is presented below to show the adaptability of the proposed genetic algorithm to changing environment. Fig. 8 shows the result of one typical run. In Fig. 8(a), the GA finds a best solution (with cost 28.80) at generation 125 in the environment. Then a new obstacle appears (Fig. 8(b)), and the original solution is no longer feasible. The GA starts to evolve better solutions according to the new environment. After 346 generations, it finds a best path (with cost 29.83) shown in Fig. 8(b). When the obstacle is removed again, after 206 generations the GA evolves an even better new path (with cost 27.77) shown in Fig. 8(c).

IV. DISCUSSION

The above simulation results demonstrate the capability of the proposed GA of evolving satisfactory paths in complex environment. This is mainly contributed by the specialized genetic operators that incorporate heuristic knowledge. To show the contribution of these operators, a comparison study between the GA with and without the specialized operators is conducted as following. We apply the comparison on the same environment as shown in Fig. 6. To see the performance of the GA without the developed specialized operators, we only keep crossover and mutation operators, and shut off all the other operators. Every specialized operator can be viewed as a special mutation operator. Simply shutting off operators makes the two sides of the comparison have different mutation rates. Therefore, we set a best mutation rate for the GA with only crossover and mutation operators. By running the GA for 100 times at different mutation rates, a mutation probability of 0.5 is selected as the best value. Then the GA with and without specialized operators is run for 20 times respectively.

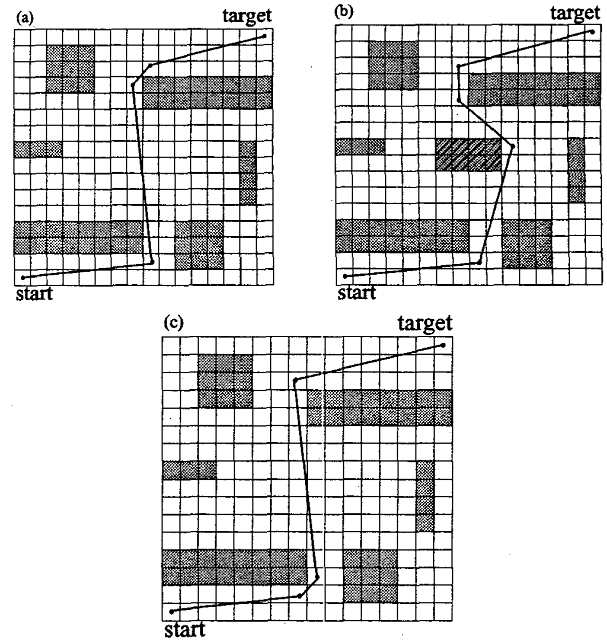

Fig. 8. One typical run of path planning in a dynamic environment. (a) Path obtained in the original environment; (b) Path after adding an obstacle; (c) Path after removal of the obstacle.

Statistic analysis shows that the specialized operators improve the performance of the GA significantly, which is evidently shown in Table I.

TABLE I

Comparison of the GA with and without specialized operators (SD: standard deviation)

Specialized operators		with	without
number of runs		20	20
Best found path cost	Mean	30.85	61.41
	SD	0.67	13.54
Number of generations	Mean	257	799
	SD	123	470

Figure 9(a) shows the best path found in 20 runs by the GA with only crossover and mutation operators. The cost is 39.15, at is found at generation 690. Fig. 9(b) displays the other 3 paths evolved by undergoing only crossover and mutation transformations. Comparing to Fig. 6, it is clear that without the knowledge based operators, the quality of solutions deteriorates dramatically, and for most of the times, it even cannot find feasible path when the environment is complex. The figures and the statistical analysis in Table I prove that the knowledge based genetic operators are essential for solving the path planning problem.

Simulation study also indicates that the proposed knowledge based genetic algorithm is of practical use because the required computational time is quite reasonable. As introduced before, grids are applied to the robot environment to obtain a relatively

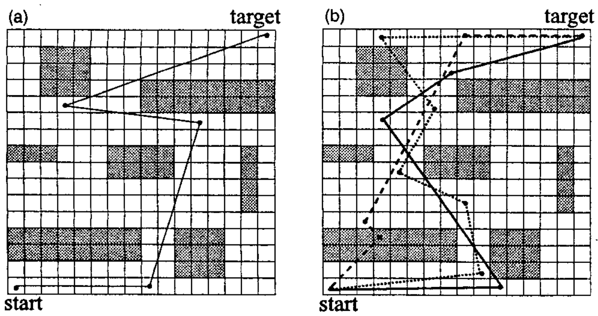

Fig. 9. The path obtained by the GA with only crossover and mutation operators. (a) The best path found in 20 runs; (b) Three other obtained paths.

simple problem representation and to simplify the involved operations of the GA. Obviously, resolution is a main factor affecting the computational time. However, simulation results show that the computational time does not increase exponentially as the resolution increases. For the environment shown in Fig. 6, when resolution increases from 16×16, to 24×24, and to 32×32, the computational time are 5.20,7.44, and 8.20 respectively, which only increases slightly. Comparing to most of the GAs used for path planning, the proposed GA is fast. Notice that the stop criterion adopted in this simulation is that the best result remains unchanged for certain generations. 100 generations is used here and also counted into computational time.

V. Conclusion

In this paper, a knowledge based genetic algorithm for path planning of a mobile robot is proposed. The GA uses a simple and unique robot path representation that combines grids and coordinates environment representations. The genetic algorithm incorporates the domain knowledge into its problem-specific genetic operators. The developed GA also features its efficient evaluation method that is greatly beneficial for evolving good solutions from infeasible solutions. The effectiveness of the knowledge based operators is demonstrated by simulation studies. The simulation results also show that the proposed genetic algorithm is effective in both complex static and dynamic environments. The efficiency of computational time makes the proposed knowledge based genetic algorithm be able to be applied to real applications.

In this proposed GA, domain knowledge is incorporated into its genetic operators. There are many other ways to utilize additional knowledge besides designing specialized genetic operators. The GA will work better by using the domain knowledge to generate more feasible solutions for initial population. This will be future work. In addition, when the GA deals with dynamic environments, it would be beneficial that some new solutions based on knowledge about the environment change can be injected into the population. Furthermore, the genetic algorithm includes more genetic operators, firing of which depends on their respective probability. On-line tuning of these probabilities is much desirable, which would be another important future work.

Acknowledgment

This work was supported by Natural Sciences and Engineering Research Council (NSERC) and Materials and Manufacturing Ontario (MMO) of Canada.

References

[1] J. C. Latombe, *Robot motion planning*. Boston: Kluwer Academic Publisher, 1991.

[2] T. Lozano-Perez, "Spatial planning: A configuration approach," *IEEE Trans. on Computers*, vol. C-32, no. 2, pp. 108–120, Feb 1983.

[3] M. Sharir, "Algorithmic motion planning in robotics," *Computer*, vol. 22, no. 3, pp. 9–19, March 1989.

[4] P. Khosla and R. Volpe, "Superquadric artificial potentials for obstacle avoidance and approach," in *Proc. of IEEE Intl. Conf. on Robotics and Automation*, vol. 3, Philadelphia, PA, USA, April 1988, pp. 1778–1784.

[5] E. Rimon and D. E. Doditschek, "Exact robot navigation using artificial potential fields," *IEEE Trans. on Robotics and Automation*, vol. 8, pp. 501–518, 1992.

[6] S. X. Yang and M. Meng, "An efficient neural network approach to dynamic robot motion planning," *Nerual Networks*, vol. 13, pp. 143–148, 2000.

[7] ——, "An efficient neural network method for real-time motion planning with safety consideration," *Robotics and Autonomous Systems*, vol. 32, pp. 115–128, 2000.

[8] J. Holland, *Adaptation in natural and artificial systems*. Ann Arbor: University of Michigan Press, 1975.

[9] L. D. (Editor), *Genetic algorithms and simulated annealing*. Los Altos, California: Morgan Kaufman Publishers, 1987.

[10] Z. Michalewicz, *Genetic algorithms + Data structures = Evolution programs, 2nd extended edition*. Springer-Verlag, 1994.

[11] K. Sugihara and J. Smith, "Genetic algorithms for adaptive motion planning of an autonomous mobile robot," in *Proc. of IEEE Intl. Symposium on Computational Intelligence in Robotics and Automation*, 7 1997, pp. 138–143.

[12] I. Ashiru, C. Czarnecki, and T. Routen, "Characteristics of a genetic based approach to path planning for mobile robots," *J. Network and Computer Applications*, vol. 19, pp. 149–169, 1996.

[13] D. Gallardo, O. Colomina, F. Florez, and R. Rize, "A genetic algorithm for robust motion planning," in *11th Intl. Conf. on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, vol. 2, Benicassim, Spain, June 1998, pp. 115–121.

[14] J. Tu and S. Yang, "Genetic algorithm based path planning for a mobile robot," in *Proc. of IEEE Intl. Conf. on Robotics and Automation*, Taiwan, September 2003, accepted.

[15] J. J. Grefenstette, "Incorporating problem specific knowledge into genetic algorithms," in *Genetic Algorithm and Simulated Annealing*, L. Davis, Ed. Los Altos, California: Morgan Kaufman Publishers, 1987, pp. 42–60.

[16] T. Shibata and T. Fukuda, "Intelligent motion planning by genetic algorithm with fuzzy critic," in *Proc. of Intl. Symposium on Intelligent Control*, Aug. 1993, pp. 565–570.

[17] T. Shibata, T. Fukuda, K. Kosuge, and F. Arai, "Selfish and coordinative planning for multiple mobile robots by genetic algorithm," in *Proc. of 31th Conf. on Decision and Control*, 12 1992, pp. 2686–2691.

[18] M. K. Habib and H. Asama, "Effecient method to generate collision free paths for autonomous mobile robot based on new free space structuring approach," in *Proc. of IEEE/RSJ Intl. Workshop on Intelligent Robotics and Systems IROS'91*, vol. 2.

[19] C. Hocaoglu and C. Sanderson, "Planning multiple paths with evolutionary speciation," *IEEE Trans. on Evolutionary Computation*, vol. 5, no. 3, pp. 169–191, 6 2001.

[20] J. Xiao, Z. Michalewicz, L. Zhang, and K. Trojanowski, "Adaptive evolutionary planner/navigator for mobile robots," *IEEE Trans. on Evolutionary Computation*, vol. 1, no. 1, pp. 18–28, 4 1997.

[21] C. Hocaoglu and A. C. Sanderson, "Evolutionary supposition using minimal representation size clustering," in *Evolutionary Programming IV. Proc. of the 4th Annual Conf. on Evolutionary Programming*, San Diego, CA, USA, March 1995, pp. 187–203.

[22] ——, "Multimodal function optimization using minimal representation size clustering and its application to planning multipaths," *J. of Evolutionary Computation*, vol. 5, no. 1, pp. 81–104, 1997.