

Mobile Robot Path Planning with η^3 -Splines Using Spatial-Fitness-Sharing Variable-length Genetic Algorithm

Jiun-Hau Wei, Jing-Sin Liu, *Member, IEEE*

Abstract—This paper presents a novel evolutionary G^3 -continuous (continuous-differentiable curvature) path planner for nonholonomic wheeled mobile robots. The evolutionary path planner generates intermediate configurations connected via η^3 -splines that yields a collision-free G^3 -continuous path, which each point on the path has a closed-form expression. The path planner is implemented as architecture of island parallel genetic algorithm (IPGA) running a variable-length genetic algorithm in each island. The techniques of spatial fitness-sharing in search space and a crowded measure of generated paths are integrated in the planner to implement a high-diversity evolutionary optimizer of paths that could self-adjust the spacing and number of intermediate configurations. The experimental result demonstrates the robustness and self-adjusting capability of evolutionary path planner in discovering shorter and smoother composite η^3 -splines paths in complex environments.

I. INTRODUCTION

THERE are various path planning for nonholonomic mobile robots utilizing different path primitives and planning schemes in a given static environments published recently, as in the literature [1]-[10], [12]-[14]. Approaches based on introducing randomness into the path planner, e.g. PRM[18], RRT[19], adaptive random walk [13], and simple genetic algorithm [1,3,5,9] or parallel genetic algorithm [14],[7],[10], show the effectiveness in searching a feasible path in difficult environments.

In addition to nonholonomic constraints, path planning problems for the practical autonomous mobile robots must consider other constraints. The criterion of minimal length path is favored for fuel and time efficiency. Smoothness of the path yields non-stop motion suitable for high speed motion to reduce the traversal time. A continuous and bounded curvature path is necessary for smooth wheel motions. Furthermore, the continuity of the derivatives of curvature also helps to eliminate the jerk effects generated from the sudden changes of acceleration. These requirements of practically useful paths, therefore, are directly reflected on the geometric features of the chosen path primitive.

For a smooth control of mobile robots, path planners usually choose one or more parametric curves as path primitive connected into a composite path with sufficient degrees of parametric or geometric continuity. Numerous attempts of blending novel parametric curves (e.g. [2], [8], [13], [17]) and

various splines (e.g. [12]) are recently adopted for smooth path generation. For example, a continuous-curvature steer for a car-like mobile robot proposed in [16] was made up of line segments, circular arcs and clothoid arcs. Some parametric curves, e.g. clothoid or cubic spiral, are described by the curvature function parameterized by arc length along the curves. The robot can be guided easily by controlling its steering angle as following the curve. However, these curves have no closed-form expression of the position of each point on the curves, which can only obtained by numerical integration. The η^3 -splines published by Piazzini *et al* [8] is a parametric curve $[x(t),y(t)]$ composed of seventh order polynomials. It has been shown that the η^3 -splines can represent any seventh-order polynomial curve with third order geometric continuity, denoted as G^3 continuity [8] (continuous in position, tangent, curvature, and the derivative of the curvature), in addition to the advantage of easily calculating the coordinates of the point on the curve. More importantly, η^3 -splines reveals its superior ability to approximate a rich class of commonly-used path primitives such as straight line segment, circular arc, clothoid (Conru spiral), and cubic spiral [2]. Such superior features of η^3 -splines are very suitable for smooth point-to-point path planning of wheeled mobile robots.

For path planning problem, to improve the efficiency of evolution toward feasible paths, parallelism of genetic algorithm [11], [14] was proposed as a viable means. Parallelism can significantly improve the efficiency and solution quality of simple genetic algorithm by encouraging a good tradeoff between exploration and exploitation of the search space. In particular, we design and implement the Island parallel genetic algorithm (IPGA) [11], [7] as a path computing framework which allows more flexibility in searching for useful solutions, since the architecture includes several design parameters that can be tuned to improve actual and statistical performance. This work implements a variable-length genetic algorithm for composite η^3 -splines path planning, which adopts variable chromosome size within a given range for initial population. The generated G^3 -continuous path guarantees that smooth control of heading and turning angle and rate, collision-free and shorter motion for nonholonomic mobile robot. Another major advantage is that there is no need to manually pre-specify the number of path primitives for different environments. Deciding a priori adequate or optimal number of waypoints connecting the path primitives and geometric arrangement of these waypoints on the free space of the map that achieve a good trade-off between path length and avoidance of obstacles are in general hard by

This work was supported by National Science Council of ROC by contract NSC 96-2221-E-001-018-MY2.

Jiun-Hau Wei and Jing-Sin Liu are with the Institute of Information Science, Academia Sinica, Nangang, Taipei, Taiwan 115, ROC.

manually setting based on heuristics, especially in large-scale complicated environments. In addition, the extensive simulations performed on a fixed number of waypoints scheme in our previous work [10] shows that there is a minimal number of waypoints that provides a minimally necessary degrees of freedom to ensure the existence of a feasible path. However, it is computationally expensive to determine an appropriate number of waypoints for a shorter collision-free path. This paper employs a variable-length genetic algorithm so that a chromosome can vary its size during evolution. Once the range of number of waypoints of initial population is set, the path planner will be self-adjusting in accordance with the complexity of environment, so that the waypoints insertion/removal and their spacing/location could be evolved to achieve the continuous shape, curvature and the derivate of curvature. Thus, the proposed path planner is very robust and could effectively decrease the amount of manual efforts involved in smooth path generation.

This paper is organized in five sections; section II introduces the primitive curve segments, η^3 -splines, used in this path planner. Section III shows the detail and the implementation of proposed variable-length genetic algorithm. Section IV presents the simulation results of the evolutionary path planner in complex environments and demonstrates its robustness. Section V is the conclusion.

II. THE η^3 -SPLINES

The η^3 -splines was firstly developed by Piazzzi *et al* [8]; By specifying the position, direction, curvature, and derivative of curvature at two terminals, a seventh-order polynomial representing a G^3 -interpolating curve is generated. Besides the given spatial configurations, this curve can still have additional six degrees of freedom, which is stated through the η vector:

$$\vec{\eta} = [\eta_1, \eta_2, \eta_3, \eta_4, \eta_5, \eta_6]^T.$$

The configurability of the splines is very flexible but results in a huge search space, causing unacceptable executing time.

In this paper, simplification is made to reduce the degrees of freedom by enforcing

$$\eta_1 = \eta_2 = \|(x_A - x_B, y_A - y_B)\|$$

the Euclidean distance between two terminal configurations,

$$\eta_3 = \eta_4 = \eta_5 = \eta_6 = 0,$$

and make the curvature and the derivative of it zero at both terminals. Following the standard formulae given by Piazzzi [8], an advantage of the η^3 -splines is we can easily calculate the Cartesian coordinates of the point on the curve via

$$\begin{aligned} \alpha_0 &= x_A, \\ \alpha_1 &= \eta_1 \cos \theta_A, \quad \gamma_1 = \eta_2 \cos \theta_B, \\ \alpha_2 &= \alpha_3 = 0, \\ \alpha_4 &= 35(x_B - x_A) - 20\alpha_1 - 15\gamma_1, \\ \alpha_5 &= -84(x_B - x_A) + 45\alpha_1 + 39\gamma_1, \\ \alpha_6 &= 70(x_B - x_A) - 36\alpha_1 - 34\gamma_1, \\ \alpha_7 &= -20(x_B - x_A) + 10\alpha_1 + 10\gamma_1, \end{aligned}$$

and similar formulae for the simplified y-coordinate coefficients are

$$\begin{aligned} \beta_0 &= y_A, \\ \beta_1 &= \eta_1 \sin \theta_A, \quad \delta_1 = \eta_2 \sin \theta_B, \\ \beta_2 &= \beta_3 = 9, \\ \beta_4 &= 35(y_B - y_A) - 20\beta_1 - 15\delta_1, \\ \beta_5 &= -84(y_B - y_A) + 45\beta_1 + 39\delta_1, \\ \beta_6 &= 70(y_B - y_A) - 36\beta_1 - 34\delta_1, \\ \beta_7 &= -20(y_B - y_A) + 10\beta_1 + 10\delta_1. \end{aligned}$$

This leads to a family of the η^3 -splines which has been numerically demonstrated its capability in its effect of minimizing the maximal curvature and maximal curvature derivatives [8].

III. THE VARIABLE-LENGTH GENETIC ALGORITHM

A. Individual representation

It is natural to represent a path composed of connected η^3 -splines as an ordered sequence of N control points; i.e. a $N+1$ connected segments of η^3 -splines can be fully defined by N intermediate control points between two boundary points, including start and end terminal. Obviously, an individual p can be represented as

$$p = [\vec{r}_1, \theta_1; \vec{r}_2, \theta_2; \dots; \vec{r}_N, \theta_N],$$

where \vec{r}_i , θ_i are the position vector and the tangent direction of the path at the control points, respectively, $N \in [N, N_u]$ is the number of control points and is not fixed but within a range.

B. Mutation, Crossover, and Selection

We employ the crossover operator reported in [9] and [15]. As shown in Fig. 1, the crossover operator is defined as the random combination between two individuals. A random point in each parent path is selected, and then two descendents are obtained by recombining the segments with different permutation. If the size of chromosomes in two individual are M and N (two terminals included), respectively, then that of the resulting descendents is between $M+N-2$ and 2 (two terminals included), resulting a variable-length crossover operation. Finally, the mutation operator is defined as the random perturbation of parameters within its allowable range [7], and the mutation rate are defined as the mutation probability per gene; i.e. it means several mutations can occur in the same individual. Finally, the selection method adopts the rank-based selection by giving the feasible paths higher priority. Feasible paths are sorted according to the path length; while infeasible paths are sorted according to the path penalty introduced in the following subsection.

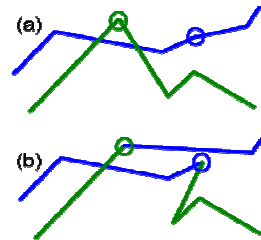


Fig. 1 (a) Two chromosomes before the crossover (b) Generated offspring after the crossover. The offspring contains information from both parents but may have different number of waypoints.

C. Infeasible Path Penalty Method

In order to accelerate the convergent process, proper collision penalty algorithms can be used to provide an improvement direction of the evolution. For example, the modified intrinsic cost presented in [7] as an improvement of intrinsic cost [4] adopted the penetration depth on infeasible paths, giving a measure of the difficulty in making the path collision free. Basically, this method finds a nearest vertex of the intersected polygonal obstacle to penetrated segments and to measure the distance between them and the vertex, representing the distance to move the segment out of the obstacle. In Fig. 2, however, it demonstrates a situation that the method fails: a path intersects a concave-shaped obstacle. In this situation, the vertices used to calculate the modified intrinsic cost may not provide a direction to guide the colliding path to become collision-free. In order to guide a colliding path around a concave polygons correctly, a more efficient infeasibility penalty method similar to [9] is proposed. As demonstrated in Fig. 3, the shortest path between two intersections along the contour of an obstacle is taken, and the length of this contour segment, which points out the extent of the infeasibility or how deep the path segments intersect an obstacle, is employed as a measure of the cost of collision.

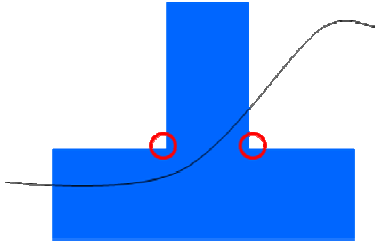


Fig. 2 Malfunction of the modified intrinsic cost may happen in case of concave-shaped obstacle. The points with red circle will be selected for computing modified intrinsic cost

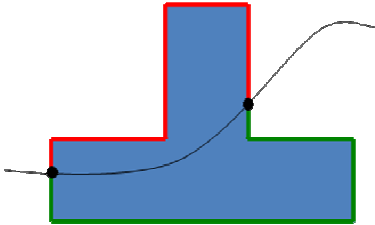


Fig. 3 The proposed path penalty method; the shortest contour segment (red) is chosen.

D. Spatial Fitness Sharing

Fitness-sharing method is widely used to improving the diversity of the evolution process. This method firstly measures the closeness between individuals, and then adjusts the fitness according to the measurement. The distance usually defined as the difference between solutions on the Pareto Front of the objective space; the main idea is to let nearby individuals compete limited resources. However, the measurement of closeness is usually defined on the fitness function and is not reflected in the actual search space. That is, the nearby individuals on the Pareto Front are not necessarily nearby in the search space, so the diversity cannot be enhanced efficiently. Therefore, for path-planning problem, the

search space is the map itself; it is trivial to define the closeness by the spatial distribution of paths. Since each path is unique determined by a set of ordered sequences of control points, we can calculate the distance efficiently by only using these points. We define a neighborhood function which measures the closeness of a specific individual p_i relative to all individuals in a generation, which is

$$neighborhood(p_i) = \sum_k correlation(p_i, p_k) \geq 1,$$

$$correlation(p_i, p_k) = f_{th} \left(\frac{1}{\sqrt{2}} \sqrt{\left(\frac{E[(X_i - \bar{X}_i)(X_k - \bar{X}_k)]}{\sigma_x \sigma_{x_k}} \right)^2 + \left(\frac{E[(Y_i - \bar{Y}_i)(Y_k - \bar{Y}_k)]}{\sigma_y \sigma_{y_k}} \right)^2} \right),$$

$$f_{th}(x) = \begin{cases} x & \text{if } x > 0.5, \\ 0 & \text{otherwise,} \end{cases}$$

where the subscript denotes the specified individual, σ_x and σ_y are the standard deviations in two orthogonal directions, X and Y are the position of an arbitrary control point, and \bar{X} and \bar{Y} are the mean position of all control points. Note that we assume the probability of every control point is the same so the expectation and the standard deviation of them can be easily obtained. By using this new measuring function, the fitness sharing is now reflected in the actual search space of the map, not in the objective space.

E. Crowd Measurement among Individuals

The spatial-fitness-sharing method can also be utilized to measure the extent of how crowded the paths in a generation or, equivalently, the inverse of diversity in search space. Using the same concept of spatial fitness sharing, the crowded factor (CF) can be defined as

$$CF = \sum_{i,k} correlation(p_i, p_k) / N^2,$$

$$1 \geq CF \geq 1/N,$$

where N is the total population size. As will be demonstrated in simulations later, crowded factor can provide a deeper understanding to the evolution process, and this information, of course, can be used as a help function to design an adaptive evolution process [15].

F. Parallelism of Genetic Algorithm

In this paper, we use the implementation of island parallel genetic algorithm (IPGA) [7], [11]. The parallel evolutionary feature can simultaneously improve the diversity and convergent speed. Furthermore, the divide-and-conquer-like scheme can also reduce the execution time and provide the possibility of parallel computation among multiple processors.

IPGA begins with several isolated subpopulations, called islands; each evolves independently with different user-setting crossover and mutation rates. A small portion of individuals will exchange among these islands periodically via a common pool serving as a migration center. The parameters among the islands are usually quite different in order to form high-diversity and high-convergent-speed islands. While these islands exchange their individuals via migration, the ultimately better individuals during evolution via separate genetic parameters in each island are preserved and migrated into whole population. Obviously, the performance is affected by four factors: the number of migrants, the migration

interval, the selection and the replacement algorithm. Too large migration size and too small migration interval will break the isolation among islands, implicating poor diversity.

IV. EXPERIMENTAL RESULT

To demonstrate the capability of the path planner based on island parallel variable-length IPGA, three maps with different size and topology feature, which includes a channel/narrow corridor of varying width, maze-like, and a mediate-scale environment consisting of a higher density of mixed convex and concave polygons, are examined to characterize the behavior of the variable-length GA. It is worth noting that sorts of difficulties can occur in path planning in each of these three testing maps. An evolutionary path planner like the one proposed in this paper may not be complete, i.e. it is likely that the planner could not discover an acceptable path in one run even if such a path does exist. However, an evolutionary path planner is possible to discover more than one alternative path. The path planner is programmed in the programming language C++, and the dumped text-based result file is used for analysis and illustration. The IPGA is composed of three equal-sized islands, which differ in settings of mutation rate and crossover rate; it migrates three percent of the total population size every ten generation. Equivalently, with sub-population size of 100 individuals in each island, a total of 9 individuals, 3 for each island, migrates in the given migration period. Also, in order to preserve diversity, three islands are configured as high-mutation-rate, high-crossover-rate, and balanced one respectively. Here some selected simulation results in a success run are presented in Fig. 4 to Fig. 6, where the shown paths may be of different number of intermediate nodes. In addition, the black line shown in these figures represents the shortest path composed of only straight line segments, which can be easily obtained from traditional deterministic path planners. Note that no attempts are made to look for the best combinations of parameters in our presented simulations. The simulation results of the proposed path planner are summarized below:

A. Evolution Characteristics

For generation of initial population, a number (between 4 and 10) of waypoints is randomly generated and linked by η^3 -splines. During evolution, the chromosome can vary its size using the crossover operation. As demonstrated in Fig. 7, for simpler maps, e.g. map I, the average of converged gene length is between five and six; too high or too low initial gene length setting in these maps will cause longer converging process or worse—unable to find feasible paths.

We denote $min(M)$ as the minimum path penalty of all paths, $avg(L)$ as the average path length of all paths, $min(FL)$ as the minimum path length of “feasible” paths, and $avg(N)$ as the average gene length of all paths. Note that the generation number at which $min(M)$ firstly reaches zero indicates the speed of finding at least one collision-free composite η^3 -splines. Very often, the feasible paths found will be preserved to evolve to shorter length and thus $min(M)$ is zero afterwards. Simulation results of three maps in 50 generations are depicted in Fig. 7 to Fig. 9.

An interesting phenomena can be observed; the evolution process of map II and map III both shows a sudden rise of $avg(L)$ just before any feasible path is found. The evolution characteristics of the proposed variable-length genetic algorithm in success runs can be stated as following.

Firstly, the path penalty is high in the earlier generations due to the random distribution of chromosomes. Then a significant evolutionary improvement is observed at the first few generations; all $min(M)$, $avg(L)$, and $avg(N)$ are reduced to much smaller value as the evolution progresses. However, for complex map like Map II and Map III, finding a collision-free path may not be easy. Before obtaining a feasible path, the evolution may temporarily reach a local minimum and possibly no improvement can be made in next few generations until leaving the local minimum. If the evolution stays on the local minimum, it means the most portion of paths collide to some obstacle, so it cannot easily get rid of them by crossovers. Although it can still have little possibility to escape via mutation, before escaping, all chromosomes will eventually become similar due to the rank-based selection and crossovers; the indistinguishable population will lead to a random crossover. Since the fitness value is only related to the path-penalty value before obtaining a feasible solution, individuals with similar fitness value may have extremely different gene length and path length. The random crossover will randomly combine any two individuals, resulting offspring with larger or smaller gene length and path length. Trivially, there must be a lower bound of gene length and path length but no upper bound of them. This asymmetry will eventually increasing the gene length and path length. Also, the longer gene length provides more flexibility, thus helping escaping. Finally, the rising will end once escaping the local minimum, then gradually evolving towards another minimum. Note the similar mechanism can happen more than once until any feasible solution is acquired.

This phenomenon can be observed from the simulation results. For example, in Fig. 7, the solution quality is improved in the first few generations. After that, the path-penalty value sticks before the generation 10, and then a rising of both $avg(N)$ and $avg(L)$ is exhibited, meaning the evolution is trying to escaping the local minimum. Finally, at generation 11, a feasible path is found, and the local minimum has been escaped. Afterwards, the solution quality is gradually improved, meaning the evolution is moving toward another minimum.

B. Robustness

The path planning in a map via variable-length genetic algorithm should not vary too much despite the variety in initial setting of range of gene length. In order to test this robustness property for the proposed path planner, map II is chosen to be demonstrated with three different setting of range of initial gene length as follows:

S1: 3~4 (small range)

S2: 4~10 (mediate range)

S3: 20~30 (wide range, not covering the convergent gene length).

The range setting represents an initial estimate of the number of control points used for the initial population. The

convergence of average gene length of feasible paths during the evolution is shown in Fig. 10. At earlier generations, the gene length is large for exploring locally and globally. As alternative feasible paths are found, they are more likely to be selected for crossover to generate another feasible path. Thus, the gene length of feasible paths will alternate within a very limited numbers. The average of gene length of feasible paths will converge to a value that is close to the number of way-points that most feasible paths will have. It can be observed that the range of initial gene length is not a critical parameter but affects diversity and convergent speed of the evolution process. The convergent gene length depends only on the map feature, thus revealing the robustness.

By comparison, initialization is crucial for the iterative gradient descent method for path optimization task to converge to a better solution. This shows the advantage of evolutionary path planner.

C. Crowd Measurement

As an example, the evolution of crowded factor in Map I is shown in Fig. 11, where the effect of spatial fitness sharing is compared. Noticeably, both evolutions find feasible solutions around generation 30. In addition, this illustrates the increasing trend of crowded factor during the evolution process. As expected, the evolution curve without fitness sharing has a higher crowding factor, or worse diversity.

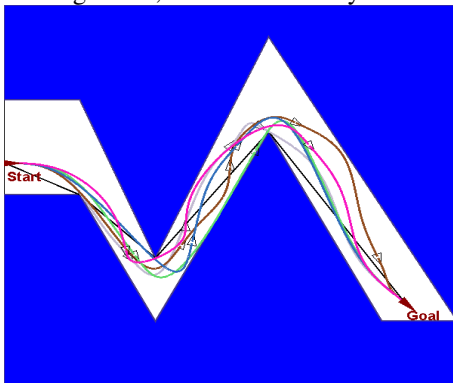


Fig. 4 G^3 -continuous path planning and the shortest path in a channel or narrow corridor of varying width (Map I, 600x600). The maximum length generated is 939, the minimum one is 912, and black line length is 827 (pixels).

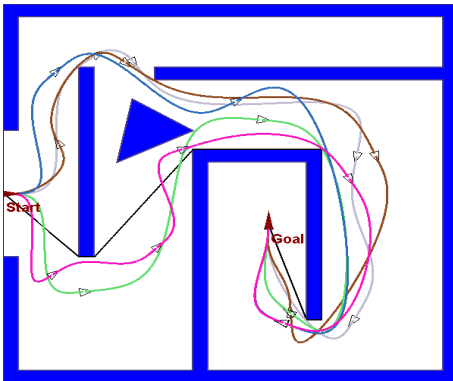


Fig. 5 G^3 -continuous path planning and the shortest path in a maze-like environment (Map II, 600x600). The maximum length generated is 1273, the minimum one is 1240, and black line length is 993 (pixels).

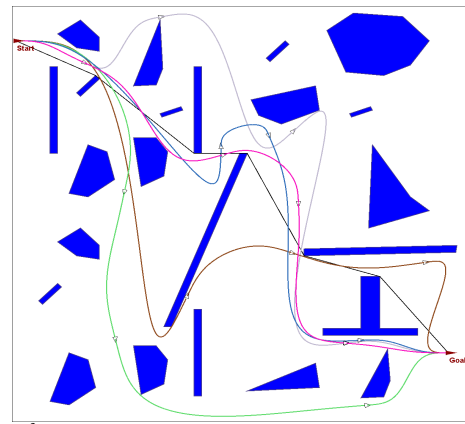


Fig. 6 G^3 -continuous path planning and the shortest path in a cluttered environment occupied with 24 mixed convex or concave obstacles (Map III, 1200x1200). The maximum length generated is 2247, the minimum one is 1744, and black line length is 1540 (pixels).

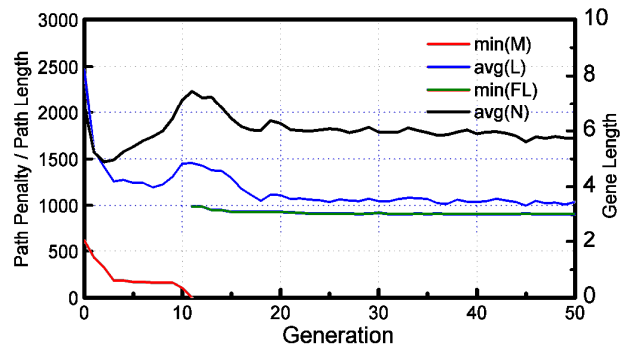


Fig. 7 Evolution of costs for Map I

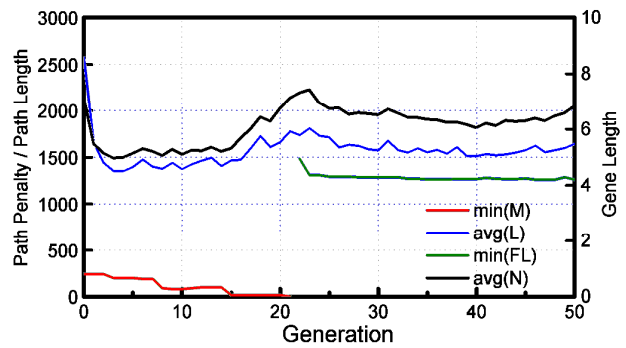


Fig. 8 Evolution of costs for Map II

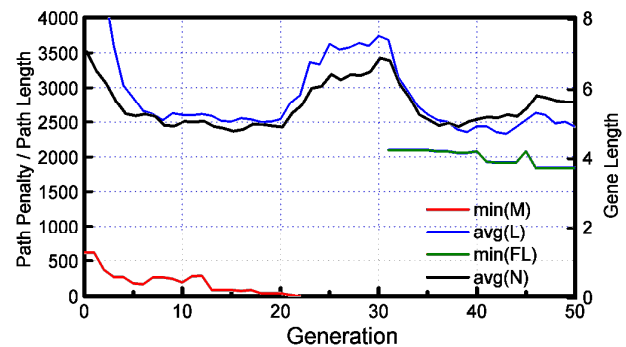


Fig. 9 Evolution of costs for Map III

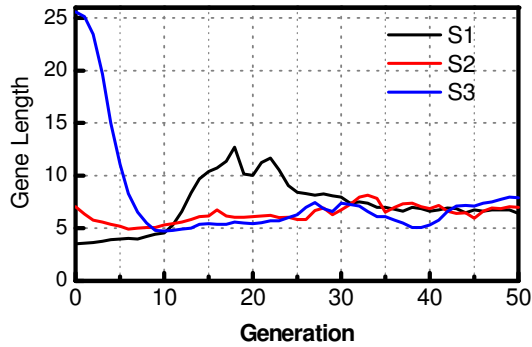


Fig. 10 Convergence of gene length for Map II

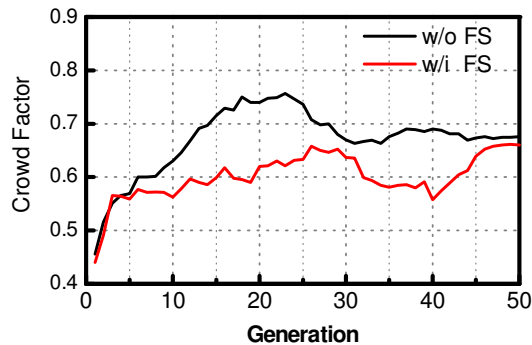


Fig. 11 Crowd Measurement of Map I

V. CONCLUSION

We propose a diversity-enhancing evolutionary path planner implementing island parallel genetic algorithm with variable-length initial chromosome and the spatial-fitness-sharing technique. The planner could automatically select near-optimal number and locations of intermediate nodes of composite η^3 -Spline via evolution to successfully generate multiple reasonably good G^3 -continuous collision-free shorter paths for wheeled mobile robots in a variety of potentially complex environments. The self-adjusting of gene length via evolution renders our path planner robust to variations in manual setting of the range of initial number of waypoints, which is critical for iterative gradient descent method for path optimization task, to converge to multiple better solutions instead of being stuck at a local minimum.

REFERENCES

- [1] K. Sugihara and J. Smith, "Genetic algorithms for adaptive motion planning of an autonomous mobile robot," *IEEE International Conference on Computational Intelligence in Robotics and Automation*, pp.138-146, 1997.
- [2] Y.J. Kanayama, B.I. Hartman, "Smooth local path planning for autonomous vehicles," *Int. J. Robotics Research*, Vol.16, no.3, 1997, pp 263-283.
- [3] J. Xiao, Z. Michalewicz, L. Zhang, and K. Trojanowski, "Adaptive Evolutionary Planner/Navigator for Mobile Robots," *IEEE Transactions on Evolutionary Computation*, Vol.1, No.1, pp 18-28, 1997.

- [4] K. Konolige, "A gradient method for real-time robot control," *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp.639-646, 2000.
- [5] M. Gemeinder, and M. Gerke, "GA-based path planning for mobile robot systems employing an active search algorithm," *Applied Soft Computing*, Vol.3, pp149-158, 2003.
- [6] T.C. Liang, J.S. Liu, G.T. Hung, and Y.Z. Chang, "Practical and flexible path planning for car-like mobile robot using maximal-curvature cubic spiral," *Robotics and Autonomous System*, Vol. 52, pp. 312-335, 2005.
- [7] K.T. Hung, J.S. Liu, and Y.Z. Chang, "A comparative study of smooth path planning for a mobile robot by evolutionary multi-objective optimization," *IEEE Int. Symposium on Computational Intelligence in Robotics and Automation*, Jacksonville, Florida, June 2007.
- [8] Aurelio Piazzi, Corrado Guarino Lo Bianco, and Massimo-Romano, " η^3 -Splines for the Smooth Path Generation of Wheeled Mobile Robots," *IEEE Trans. Robotics*, Vol. 23, No. 5, pp. 1089-1095, 2007.
- [9] G. Erinc and S. Carpin, "A genetic algorithm for nonholonomic motion planning," *IEEE International Conference on Robotics and Automation*, pp. 1843-1849, 2007.
- [10] H.-C. Chang and J.-S. Liu, "High-quality Path Planning for Autonomous Mobile Robots with η^3 -Splines and Parallel Genetic Algorithms," *IEEE Int. Conf. Robotics and Biomimetics*, Bangkok, Thailand, Dec. 2008.
- [11] D. Henrich, "Fast motion planning by parallel processing-A review," *Journal of Intelligent and Robotic Systems*, Vol. 20, pp 45-69, 1997.
- [12] V. Munoz, A. Ollero, M. Prado, A. Simon, "Mobile robot trajectory planning with dynamic and kinematic constraints," *IEEE International Conference on Robotics and Automation*, pp.2802-2807, 1994.
- [13] S. Carpin, G. Pilloentto "Motion planning using adaptive random walk," *IEEE Trans. Robotics*, vol.21, no.1, pp. 129-136, 2005.
- [14] J.M. Ahuactzin, E.-G. Talbi, P. Bessiere and E. Mazer, "Using genetic algorithms for robot motion planning," *European Conference on Artificial Intelligence 92*, Vienna, Austria, 1992.
- [15] M. Tarokh, "Hybrid intelligent path planning for articulated rovers in rough terrain," *Fuzzy Sets and Systems*, vol.159, pp.2927-2937, 2008.
- [16] T. Fraichard and A. Scheuer, "From Reeds and Shepp's to continuous-curvature paths," *IEEE Trans. On Robotics*, vol. 20, no. 6, pp.1025-1035, 2004.
- [17] A. Kelly and B. Nagy, "Reactive nonholonomic trajectory generation via parametric optimal control," *International Journal of Robotics Research*, vol.22, no.7-8, pp.583-601, 2003.
- [18] L. Kavraki, P. Svestka, J. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE transactions on Robotics and Automation*, vol. 12, pp. 566-580, 1996.
- [19] S. LaValle and J. Kuffner Jr, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, p. 378, 2001. (RRT)