Proceedings of the 2007 IEEE International Symposium on
Computational Intelligence in Robotics and Automation
Jacksonville, FL, USA, June 20-23, 2007

A comparative study of smooth path planning for a mobile robot by evolutionary multi-objective optimization

Kao-Ting Hung[a,b], Jing-Sin Liu[a], and Yau-Zen Chang[b]
[a]Institute of Information Science, Academia Sinica, Nangkang, Taipei, Taiwan 115, ROC
[b]Department of Mechanical Engineering, Chang Gung University, Tao-Yuan, Taiwan 333, ROC
kattin@st.cgu.edu.tw, liu@iis.sinica.edu.tw, zen@mail.cgu.edu.tw

***Abstract*—This paper studies the evolutionary planning strategies for mobile robots to move smoothly along efficient collision-free paths in known static environments. The cost of each candidate path is composed of the path length and a weighted sum of penetration depth to vertices of polygonal obstacles. The path is composed of a pre-specified number of cubic spiral segments with constrained curvature. Comparison of the path planning performance between two Pareto-optimal schemes, the parallel genetic algorithm scheme based on the island method (PGA) and the non-dominated sorting genetic algorithm (NSGA-II), are conducted in terms of success rate in separate runs and path length whenever collision-free paths are found. Numerical simulation results are presented for three types of obstacles: polygons, walls, and combinations of both.**

I. Introduction

GLOBAL or local path planning for a mobile robot between two locations rich in obstacles has been studied by complicated numerical methods such as [2] and [3]. The approaches that apply the genetic algorithms, such as [4]-[7], can find piecewise linear paths for a mobile robot without complicated mathematical formulations. Usually, the evolutionary planners contain specially designed operators that are complex and case-dependent. However, the turning constraint, including smoothness and curvature, for the car-like robot is normally unaccounted for [18].

Besides, a feasible path should also be of minimum length and/or minimum energy, these requirements lead to a multi-objective optimization problem (MOOP) with conflicting objectives [13-17]. In recent years, the idea of Pareto-optimality [16] is introduced to solve MOOP with the advantage that multiple tradeoff solutions can be obtained in a single run.

In our previous work [11], a parallel genetic algorithm (PGA) scheme based on the island model was applied to find collision-fee paths composed of a pre-specified number of cubic spiral segments. The method inherently encodes the curvature-continuity constraint into the candidate paths.

In this paper, we propose a new intrinsic cost for the evolutionary path design strategy that incorporates obstacle penetration-depth to evaluate the degree of collision. The candidate paths are composed of cubic spiral segments. Tradeoff between penetration-depth of obstacles and path length renders the problem two-objective. Search performance of the proposed scheme is compared with the non-dominated sorting genetic algorithm (NSGA-II) [19], a popular Pareto-based evolutionary method.

The remainder of the paper is organized as follows. Section II briefly reviews the cubic spiral method and introduce a new penetration-depth based intrinsic cost for obstacle avoidance. The path searching algorithms based on evolutionary multiobjective optimization with non-smoothness handling, mainly PGA and NSGA-II, are presented in Section III. Comparisons and simulations are presented in Section IV. Finally we make a conclusion in Section V.

II. Preliminaries

Let $q \equiv (x, y, \theta)$ represent a configuration of the mobile robot where (x, y) and θ denotes the position and orientation, respectively. The path followed by a unit-speed mobile robot starting from the initial configuration (x_0, y_0, θ_0) is governed by integrating the nonholonomic kinematic constraints,

$$\begin{cases} \theta(s) = \theta_0 + \int_0^s \kappa(t)dt \\ x(s) = x_0 + \int_0^s \cos(\theta(t))dt \\ y(s) = y_0 + \int_0^s \sin(\theta(t))dt \end{cases} \quad (1)$$

where x, y and θ represent the function of position in x-axis and y-axis and orientation of robot through a path, s is the path length, and is set as 0 at the initial point of robot (x_0 , y_0). κ is the curvature function.

A. Two objectives for path optimization

The cost of travel is path length and the cost of obstacle avoidance, which can be represented by the intrinsic cost by assuming that the robot is a point to avoid time-consuming collision detection between rigid objects [3]. When used in environments of varying geometric shapes and sizes, we see some defects in the original definition of intrinsic cost, which only calculates how many intersections. As Fig. 1 shown, similar paths (represented by solid and dashed lines) cross a thin or a large obstacle (compared to the length of a path segment) will have the same intrinsic cost. Nonetheless, the solid path is visually more proximate to collision-free than

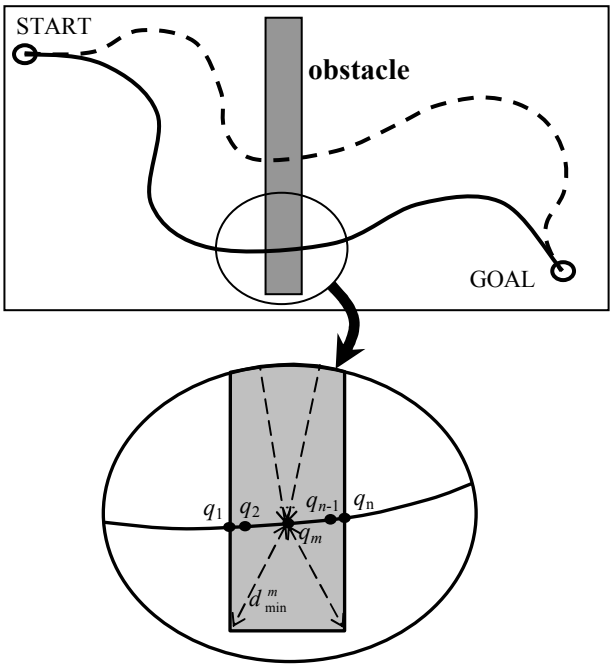


Fig. 1 Definition of penetration-depth based intrinsic cost for obstacle avoidance

the dashed path. To invoke the measure of proximity to collision-free into a path to be evaluated, we design a modified intrinsic cost in the following. Refer to Fig. 1. Suppose there are n sampling nodes $q_{1,\Lambda,}q_n$ of a path segment intersecting a designated obstacle. For convenience, every path segment of different length is sampled as the same number of nodes N_s. The normalized intrinsic cost for i-th path segment is defined as $r_i = n_i / N_S, i = 1,\Lambda\ N_{seg}$, where n_i represents the number of nodes of i-th path segment that are within a designated obstacle, N_{seg} is a prespecified number of path segments. For each segment, we measure the penetration depth as the minimum distance from the middle node q_m to all vertices of the intersecting obstacle, $d^m_{\min}$, as the proximity to collision-free. Note that if the number of intersected nodes is odd, q_m would be unique. On the other hand, if the number is even, there are two median nodes, i.e. q_m and q_{m+1}, to derive two minimum distances, $d^m_{\min}$ and $d^{m+1}_{\min}$. Their arithmetic average is the proximity to collision-free. The general form of newly designed intrinsic cost function can thus be defined as:

$$f_{new} = \sum_{i=1}^{N_{seg}} r_i \cdot D_i$$

where r_i and D_i of i-th path segment are defined as:

$$r_i = n_i / N_S \text{ (normalized intrinsic cost)}$$

$$D_i = \begin{cases} (d^m_{\min})_i & \text{, if } n_i \text{ is odd} \\ \left((d^m_{\min})_i + (d^{m+1}_{\min})_i\right)/2 & \text{, if } n_i \text{ is even} \end{cases}$$

where subscript i denotes i-th path segment.

B. Review of Cubic Spiral Method

For smooth path generation, the path is made up of cubic spiral segments, which is curvature continuous.

1). Cubic Spiral: By definition, cubic spiral is a set of trajectories that the direction function θ is a cubic polynomial of curve length l. Its angle, which describes how much the curve turns from the initial orientation to final orientation, is denoted by

$$\alpha = \theta(l) - \theta(0) \tag{2}$$

From the first equation of (1) and the boundary conditions at s=0 and s=l, we have (Lemma 2, [1]),

$$\kappa(s) = \frac{6\alpha}{l^3} s(l - s) \tag{3}$$

If the length of a cubic spiral is 1, its size is given by (Lemma3, [1])

$$D(\alpha) \equiv 2\int_0^{1/2} \cos\left(\alpha\left(\frac{3}{2} - 2t^2\right)t\right)dt \tag{4}$$

Due to similarity of all cubic spirals, the value $D(\alpha)$ can be computed and then derive the curve's length l by the following equation (Proposition 8, [1]),

$$l = \frac{d}{D(\alpha)} \tag{5}$$

where d is the distance of two configurations.

2). Concept of Symmetric Configurations: For an arbitrary configuration q, [q] denotes its position (x, y), and (q) its direction θ. For a configuration pair (q_1, q_2), the size is the distance between the two points [q_1] and [q_2], and the angle is the deflection angle between the two orientations (q_1) and (q_2). In [1], a symmetric mean q of any configuration pair (q_1, q_2) is a configuration that leads (q_1, q) and (q, q_2) are both symmetric pairs. All symmetric means of a configuration pair (q_1, q_2) forms a circle if (q_1) ≠ (q_2) or a line connecting q_1 and q_2 if (q_1) = (q_2) (Proposition 3, [1]). It is noted that the symmetric property is very important in this method because a cubic spiral can connect two symmetric configurations.

3). Original cubic spiral path planning method

The cubic spiral method can connect two given configuration q_1 and q_2 according to the following steps:

I. If q_1 and q_2 are symmetric, connect these two configurations with a cubic spiral directly.

II. Else, connect these two configurations with a specified symmetric mean

(i)*Non-parallel case:*

If $(q_1) \neq (q_2)$, we should define the center of the circle that go through the given configurations q_1 and q_2.

$$p_c = \left(\frac{x_1 + x_2 + c(y_1 - y_2)}{2}, \frac{y_1 + y_2 + c(x_2 - x_1)}{2} \right),$$
$$c = \cot\left(\frac{\theta_2 - \theta_1}{2} \right)$$

Thus the position of symmetric mean $[q_s]$ can be defined as:

$$[q_s] = (x_c + r \cdot \cos(\beta_1 + (\beta_2 - \beta_1) \cdot \gamma), y_c + r \cdot \sin(\beta_1 + (\beta_2 - \beta_1) \cdot \gamma))$$

where β_1 and β_2 are represent the orientations from p_c to q_1 and q_2 respectively. The orientation of the symmetric mean can be defined according to the position of symmetric mean [1].

(ii)*Parallel case:*

If $(q_1) = (q_2) = \theta$,

$$q_s = (\ x_1 + (x_2 - x)_1 \cdot \gamma\ ,\ y_1 + (y_2 - y)_1 \cdot \gamma\ ,\ \beta - (\theta - \beta)\ )$$

where $[q_1] = (x_1, y_1)$, $[q_2] = (x_2, y_2)$, $\beta = \tan^{-1}\left(\frac{y_2 - y_1}{x_2 - x_1} \right)$

For a given start configuration, a cubic spiral can be defined by the size d and deflected angle α, which has: its length via (5), its curvature function by (3) and its terminal configuration from equation (1). In addition, when the size of cubic spiral is negative, we can plan the backward motion of the robot according to equation (3) and (1) with l negative.

III. Cubic Spiral Path Planning via Evolutionary Multi-Objective Optimization

A. Individual representation: candidate path

A path segment is defined by a continuous mapping $\tau : [0,1] \to C$ where $q(s) = (x(s), y(s), \theta(s))$ denotes robot configuration with s arc length. A path is composed of a set of path segments connected via a pre-specified number of intermediate configurations. In this paper, we use cubic spiral as a path segment for a uni-cycle mobile robot, which is kinematically feasible. More specifically, a path is designed as a concatenation of three smooth subpaths, each is composed by several cubic spiral segments: subpaths ***S***, ***G***, ***M*** (Fig.2), which inherently encodes the curvature-continuity constraint into the path representation. The subpath ***S*** is composed of cubic spiral segments, planned forwardly from START through a prespecified number of intermediate configurations. Similarly, the subpath ***G*** is planned backwardly from GOAL.

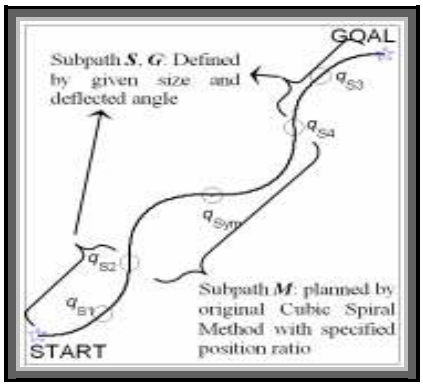


Fig. 2 A path is composed by three subpaths, each is composed by cubic spiral segments: subpaths ***S***, ***G***, ***M***

Finally, ***S*** and ***G*** are connected by two cubic spiral segments defined via a symmetric mean [1], i.e. subpath ***M***. If we define the subpaths ***S*** and ***G*** by N cubic spiral segments, i.e. N control points, the chromosome would consist of $2N+1$ genes (the size and deflected angle for each cubic spiral segment in ***S*** and ***G***, and the position ratio of symmetric mean for subpath ***M***). The composition of genes for a single chromosome p is, excluding the given START and GOAL configurations

$$p = [d_1\ \alpha_1\ ;\ d_2\ \alpha_2; \ldots ;\ d_N\ \alpha_N\ ;\ \gamma_{Sym}],\ N\text{: even number} \quad (6)$$

B. Island-based PGA with migration[11]

PGA based on island model is a parallelization scheme of genetic algorithms that can reduce the execution time. Since each island is run to follow a different solution trajectory, the island model may help to promote genetic diversity. The island model may have synchronous/asynchronous migration of individuals. This scheme divides the population into several communicating subpopulations each evolving via SGA in an island with a common pool serving as a migration center (Fig.3), created by cross-fertilization among the individuals of different islands. For every M generations (M is called migration frequency/interval), migration takes place. A fraction of subpopulations (called migration rate) of individuals of each island is selected based on their ranks to send to the common pool, and gathered. Then the common pool redistributes the individuals randomly onto the different islands. The size of the common pool equals the migration size (the number of individuals that migrate) of each island times the number of islands. Performance of island based PGA is affected by four factors: number of migrants, migration interval and the selection and replacement strategy of individuals. The following paragraphs describe the detail of the PGA.

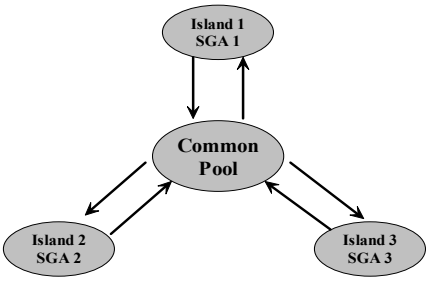


Fig.3 The migration policy of PGA based on island model

1). Fitness Definition: Rank-based Assignment & Pareto Ranking

Fast non-dominance sorting method proposed by [19] is used to rank the individuals in a population. Higher ranks will be given higher indices. The solutions of highest rank are termed as Pareto-frontier.

2). Selection

The roulette-wheel selection operator is employed. For faster convergence, elitism is used to retain some preferred individuals at each generation. One of the most popular criteria to select representative solutions from the Pareto-frontier is the min-max method [9]. The main idea of this method is to select a point within the two ends of Pareto-frontier that the maximum deviation of objectives is minimized. For a *m* objectives problem, if there are *k* number of solutions $p_1 \sim p_k$ within the Pareto-frontier, we can select a min-max picked solution according to

$$\min\left[\max\left(z_1,z_2,...,z_m\right)_1,\max\left(z_1,z_2,...,z_m\right)_2,...,\max\left(z_1,z_2,...,z_m\right)_k\right]$$

where the deviations $z_i=\left|f_i(p)-f_i^{\min}\right|/\left(f_i^{\max}-f_i^{\min}\right)$.

3). Genetic Operations: Crossover and Mutation

The crossover is implemented in this work as a linear interpolation between two chromosomes, also called arithmetic crossover [10]. The following equation shows the interpolation operation between two distinct chromosomes: $\theta_1'=\theta_1\cdot\gamma_1+\theta_2\cdot(1-\gamma_1),\theta_2'=\theta_2\cdot\gamma_1+\theta_1\cdot(1-\gamma_1)$,where γ_1 is a randomly generated real number between 0 and 1. Please note that γ_1 is different in different genes.Mutation is a mechanism introduced to explore new searching directions. Assuming $\theta_{\max}$ and $\theta_{\min}$ be the bounds of candidate solutions, the resultant descendant generated by the mutation operation will be $\theta'=\theta_{\min}+\gamma_2\cdot(\theta_{\max}-\theta_{\min})$,where θ stands for a chromosome in a population, γ_2 is a random number between 0 and 1.

4). Non-smoothness handling

The cubic spiral would be a spiral curve for certain values of deflected angle and size. As this situation occurs in the evolution of paths, a non-smooth path is generated when we connect the subpaths ***S*** and ***G*** by a subpath ***M,*** since there is no guarantee that the deflected angle between last nodes at subpaths ***S*** and ***G*** is smaller than the specific value, as shown by dot circle of Fig. 4 . It implies that a generated path is likely to be non-smooth during evolution . Thus the path problem we tackle with is a constraint-handling optimization problem. However, this type of constraint is very difficult to be reflected in the design space defined by p in (6), thus we only can make a boolean discrimination for every individuals (i.e., only indicating smooth or non-smooth). This results in two categories of population at every generation, i.e. smooth and non-smooth solutions. For the proposed PGA, we generate the offspring based on two rankings: The first is a ranking for total populations, the second is a ranking only for smooth solutions, to make the evolution to preserve the diversity of searching. In our implementation, these paths from these two rankings have the same probability to be selected, therefore smooth paths have more chances to be selected for further genetic operations.

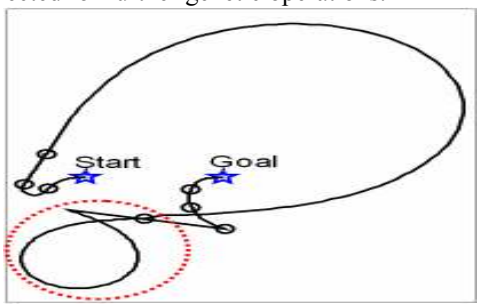


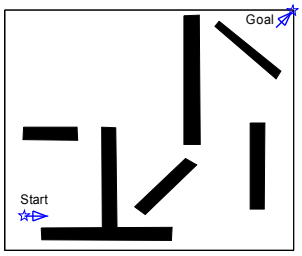
Fig. 4 Example of a non-smooth path (with 6 control points).

C. Sub-goals manipulation operator [11]

For the mobile robot path planning problem, it is important to consider how to elaborate an infeasible path into a more acceptable path. The subgoals manipulation operator operates on the infeasible paths segments that cross the obstacles in order to accelerate the evolution to find out the collision-free paths. The operator locally mutates those nodes in a predefined neighborhood, so that this local refinement of path shape occurs in the sun-regions.

IV. Simulation Results

For empirical comparison of path planning performance of various schemes to make clear which scheme performs better, four performance indices are considered: The success rate represents the number of specific runs that find out at least one feasible path for 20 independent runs. The best and worst path means the path with minimum and maximum length within those feasible paths in every single run of 20 runs and average length and standard deviation from all success runs. The initialization of candidate paths respects the following predefined range: size *d:* [20, diagonal distance of map/ *N*-2];deflected orientation α: [$-\pi$, π]; position ratio of symmetric mean γ: [0.2, 0.8], and are the same for all simulations



Static Environment-1 (map size: 600*500)

Fig. 5 Testing environment

A. Comparison of the preservation strategies: Table 1

Four different preservation strategies are implemented: two Min-Max picked solutions and two solutions with respective minimum objectives (Strategy-1); three Min-Max picked solution and a solution with minimum intrinsic cost (Strategy-2); four random selections from the Pareto-front (Strategy-3); two random selections from the Pareto-front and the solutions with respective minimum objectives (Strategy-4). Table 1 shows that

(a) new vs. old intrinsic cost: We could see that using new intrinsic cost as the cost of obstacle avoidance is more effective in achieving higher success rate than old intrinsic cost, since the colliding paths and collision-free paths are increasingly discriminatory. However, using old intrinsic cost often finds a shorter path in case of successful runs than using new intrinsic cost.

(b) comparison of preservation strategies: Strategy-I, which preserves the boundary solutions and has a good spread of non-dominated solutions, outperforms in the average and standard deviation of path length for all testing environments. Consequently, Strategy-I is adopted for PGA preservation scheme in later comparison.

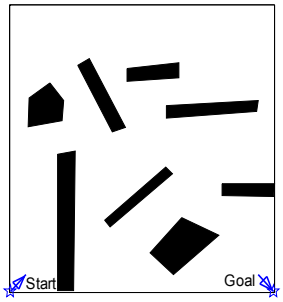
B. Comparison for island-based PGA with NSGA-II: Table 2, Table 3

In this subsection, PGA with different migration intervals, NSGA-II is implemented to this problem. The implementation of NSGA-II follows the description of [19], includes the parameters: crossover rate, mutation rate, and distribution indices for crossover and mutation operators. All schemes are incorporated with sub-goals manipulation operator. For the migration intervals, the PGA with 5 intervals (denoted as PGA(5) in the following) has better result than the others. PGA(5) performs well in the wall-like obstructed maps and NSGA-II performs better in environments containing only polygonal obstacles.

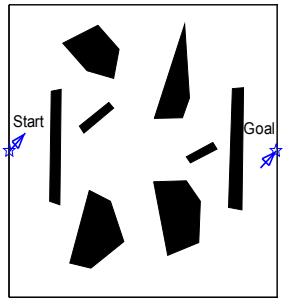
For further comparison of PGA(5) and NSGA-II, another obstructed environments mixed with wall-like and polygonal obstacles (Fig. 6) are tested. Table 3 demonstrates the results of this comparison. NSGA-II indeed has poor performance when there exist wall-like obstacles (e.g. Environment-3). This might be resulted from the crowded comparison operator of NSGA-II, since this operator might neglect many possible solutions. For the PGA, the success rates for each map are more stable than the NSGA-II. The NSGA-II performs dramatically well in the average length and STD than the PGA in Environment-2.

Table. 1 Comparison of different preservation strategies in Fig. 5- ***6 Control Points,*** new/old intrinsic cost(SGA, Pop size: 90, max generation:150, crossover/mutation rate: 0.85/0.1, number of manipulations of infeasible paths: 10% of population size/per generation)

	1	2	3	4
Success rate (/20 Runs)	20/5	20/12	19/12	20/13
Best path	827.43/818 0.14	26.92/795. 40	30.08/788. 53	24.70/798. 82
Worst path	878.13/879 1.69	61.56/991. 82	98.92/914. 64	1090.80/79 8.82
Average length ± SD of best paths	846.94 ±10.73/83 4.23 ±25.22	861.15 ±41.01/856. 06 ±54.86	857.69 ±39.08/838. 53 ±38.60	855.38 ±58.00/839. 91 ±43.73



Static Environment-2



Static Environment-3

Fig. 6 Another testing environments (map size: 600*600)

Table 2 Comparison of path planning performance in Fig.6

	PGA(1)	PGA(3)	PGA(5)	NSGA-II*
Environment-3				
Success rate(/20 Runs)	15	14	12	4
Best path	734.07	728.29	735.59	747.51
Worst path	1097.60	954.80	857.94	857.39
Average length	810.75	778.29	772.97	802.52
± SD of best paths	±103.98	±54.95	±34.82	±53.11
Environment-4				
Success rate(/20 Runs)	13	19	19	15
Best path	771.98	781.45	778.23	782.39
Worst path	849.17	1497.05	881.99	947.33
Average length	800.19	882.88	804.33	835.18
± SD of best paths	±19.97	±185.30	±22.74	±52.25

*The crossover and mutation probabilities are 0.9 and $1/N_d$ (where N_d is the number of genes) respectively. Distribution indices for crossover and mutation operator are defined as $\eta_c = 20$ and $\eta_m = 20$.

Table 3

Environment-2 with more multiple wall-like obstacles (8 Control Points)		
	PGA(5)	NSGA-II
Success rate(/20 Runs)	18	19
Best path	705.32	605.23
Worst path	1222.37	780.87
Average length	978.42	632.88
± SD of best paths	±94.51	±52.02
Environment-3 with equal number of wall-like and polygon obstacles (8 Control Points)		
Success rate(/20 Runs)	20	12
Best path	833.86	840.54
Worst path	991.71	1080.04
Average length	898.88	940.04
± SD of best paths	±53.69	±69.85

C. Summary of simulations

The evolutionary path planner based on PGA are more robust in successfully finding a safe smooth and shorter path, while NSGA-II achieves better distributed approximation of Pareto-front, but may degrade its searching performance greatly in some environments containing wall-like obstacles. In reality, the population size can be increased to yield better results. On the other hand, since the NSGA-II has higher computation complexity due to the sorting of combined populations, PGA would solves the problem more efficiently.

V. CONCLUSION

In this paper, two Pareto-based evolutionary multi-objective optimization schemes, PGA with migration intervals 5 and NSGA-II are employed in this comparative study to solve a bi-objective optimization problem for generating a smooth path to move a mobile robot safely from a start configuration to a goal configuration in completely known static environments containing polygonal and wall-like obstacles. The cubic spiral segments are used to compose a curvature-continuous directed planar curve. A modified intrinsic cost function incorporating the penetration depth into obstacle avoidance is effective for identifying which paths are closer to collision-free, thus raising the success rate in searching a feasible path via multi-objective evolution. Our comparative study based on simulations show that PGA is more robust over all testing environments, while NSGA-II has better-distributed approximation to Pareto-front, but may degrade greatly in some environments containing wall-like obstacles. Combining the merits of both schemes is currently under investigation.

ACKNOWLEDGMENT

This study was supported by National Science Council of R.O.C. under contract NSC 94- 2212 –E-001-001.

REFERENCES

[1] Y.J. Kanayama, B.I. Hartman, "Smooth local path planning for autonomous vehicles," *Proceedings Int. J. Robot. Res.*, Vol.16(3), 1997, pp 263-283.

[2] 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, 2005, pp. 312-335.

[3] K. Konolige, "A gradient method for real-time robot control," *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp.639-646, 2000.

[4] P. Wide and H. Schellwat, "Implementation of a genetic algorithm for routing an autonomous robot," *Robotica*, Vol. 15, 1997, pp 207-211.

[5] M. Gemeinder, and M. Gerke, "GA-based path planning for mobile robot systems employing an active search algorithm," *Applied Soft Computing*, Vol.3, 2003, pp149-158.

[6] 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, 1997, pp 18-28.

[7] A. C. Nearchou, "Path planning of a mobile robot using genetic heuristics," *Robotica*, Vol.16, 1998, pp 575-588.

[8] Y. Jin, T. Okabe, B. Sendhoff, "Dynamic weighted aggregation for evolutionary multi-objective optimization: why does it work and how", *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, pp.1042-1049, 2001.

[9] A.D. Belegundu and T.R. Chandrupatla, *Optimization Concepts and Applications in Engineering*, New Jersey, Prentice Hall, 1999.

[10] H.K. Lam, F.H. Leung, P.K.S. Tam, "Design and stability analysis of fuzzy model-based nonlinear controller for nonlinear systems using genetic algorithm" *IEEE Trans. Syst., Man and Cybernetics*, Part B, vol. 33, 2003, pp. 250-257.

[11] K.T. Hung, J.-S. Liu, and Y.Z. Chang, "Generation of multiple cubic spiral paths for obstacle avoidance of a car-like mobile robot using the evolutionary search," *International Conference on Autonomous Robots and Agents*, Dec. 12-14, 2006.

[12] N. Wang and Y.Z. Chang, "Application of the genetic algorithm to the multi-objective optimization of air bearings," *Tribology Letters*, Vol. 17, No. 2, 2004.

[13] O. Castillo, L. Trujillo and P. Melin, "Multiple objective genetic algorithms for path-planning optimization in autonomous mobile robots," *Soft Computing,* 2006.

[14] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*, West Sussex, England, John Wiley & Sons, 2001.

[15] K. Fujimura, "Path planning with multiple objectives," *IEEE Robotics and Automation Magzine*, pp.33-38, March 1996.

[16] C. M. Fonseca and P. J. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," *Evolutionary Computing*, Vol.3, No.1, 1995, pp. 1-16.

[17] G. Dozier, S. McCullough, A. Homeifar, E. Tunstel and L. Moore, "Multiobjective evolutionary path planning via fuzzy tournament selection," *IEEE World Congress on Computational Intelligence*, pp.684-689, 1998.

[18] A. Scheuer and Th. Fraichard, "Continuous-curvature path planning for car-like vehicles," *1997 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp.997-1003, 1997.

[19] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II," *IEEE Transaction on Evolutionary Computation*, Vol. 6, No. 2, 181-197, 2002.