Generation of multiple cubic spiral paths for obstacle avoidance of a car-like mobile robot using the evolutionary search

Yau-Zen Chang
Chang Gung University

Appeared in The 3rd International Conference on Autonomous Robots and Agents (ICARA 2006)
12-14 December 2006, Palmerston North, New Zealand

Generation of multiple cubic spiral paths for obstacle avoidance of a car-like mobile robot using evolutionary search

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

Abstract

Cubic spiral is a curve whose direction is a cubic polynomial of arc length. It has been used as a path primitive for nonholonomic car-like mobile robot with restricted turning capability in free environment [1], [2]. In this paper, we investigate the use of cubic spiral segments for car-like mobile robots moving from a given start configuration to a given goal configuration in an environment cluttered with known static obstacles. We propose a parallel genetic algorithm (PGA) based on the island model to generate multiple collision-free paths in parallel via sets of a predefine number of intermediate configurations connected by collision-free cubic spirals. The architecture of PGA is simple: it consists of multiple islands, where subpopulation of each island runs a simple genetic algorithm independently with the same fixed-length real-coded chromosome representation of cubic spiral path segments, basic genetic-operators, one subgoals manipulation operator, and maintains diversity of subpopulations by synchronous migration every certain number of generations among the islands and a Big-bang activity. The generation of multiple collision-free paths by PGA is via the following strategy: (i) the generated collision-free paths are similar to an already found collision-free path, then they are eliminated and replaced by randomly generated paths for later evolution, (ii) otherwise, they are extracted as new alternative collision-free paths. Simulation results of mobile robot for a set of different start orientations show the effectiveness of this evolutionary approach to evolve a more rich set of collision-free paths.

Keywords: obstacle avoidance, smooth path planning, car-like mobile robot, parallel genetic-algorithm (PGA)

1. Introduction

Path planning of mobile robots with different designs has been extensively and intensively studied in recent years. A lot of methods, for example: artificial potential field method, probabilistic roadmap method, either global or local, offline or online path planning in complex environments based on a variety of path primitives have been proposed [11]. In general, a path is made up of a set of local curve pieces. Path planning problem can be formulated as an optimization problem subject to motion constraints, which constrain the differential geometry of the set of feasible paths [17]. In particular, genetic algorithm (GA) based path planning for a mobile robot [3]-[7], [14], [15] has attracted more attentions, mainly for shortest piecewise-linear path in grid environments with obstacles modelled as fully occupied connected grids. The advantage of GA based path planner lies in the parallel problem solving capability and the minimum requirements regarding the problem formulation. However, implicitly assumed in this kind of problem formulation is the mobile robot can instantaneously turn from any orientation to arbitrary orientation, without any constraint of turning radius (i.e. no consideration of nonholonomy of robot motion). In addition, piece-wise linear paths does not allow smooth tracking by mobile robots, and thus for practical use, need smoothing or refinement [17]. Besides, most existing evolutionary path planners contain many special design (problem-specific knowledge based) operators to “repair” a longer or colliding line segment. The solution techniques are often engineered for a specific context. These complicated the overall design of GA-based path planner and reduce the flexibility of GA.

For smooth path, not only the smooth motion along the local curve pieces but also the smooth transition between the path segments (i.e. orientation of current end configuration is the orientation are guaranteed for mobile robot. B-spline is a good candidate for smooth path due to its local and global control property and ease of computation of points and derivative on it [16]. In this paper, cubic spiral is chosen as the primitive for generating smooth path of better curvature control (than the more well-known B-spline). Cubic spiral has been employed to mobile robot path generation for the first time by [1]. Its two drawbacks: (1)overly long path for close locations with large changes in orientation, or very distant goals,(2)lengthy paths with no use of the reverse motion capability are remedied by [2]. However, the obstacle avoidance capability of the cubic spiral path planning methods of [1], [2] is very limited. In this paper, we aim to enhance the obstacle avoidance capability of car-like mobile robot in static environment using cubic spiral as path primitive. To this aim, the parallel genetic algorithm (PGA) based on the island model, embedded with specially designed problem-specific operations, is designed to generate multiple paths composed of collision-free cubic spiral path segments linking from a given start configuration to a given goal configuration through a predefined number (depending on the complexity of the environment and task) of intermediate configurations found by evolutionary search without cross the static obstacles or walls. Multiple paths, which allow more path options for improving the success of robot path planning, are generated. Our evolutionary method is simple and general to be potentially applied to dynamic environment.

The remainder of this paper is organized as follows. Cubic spiral path planning method [1] is briefly reviewed. Then our evolutionary smooth path generation method based on multi-island PGA would be detailed in Section 3. Some comparisons to show the effectiveness of specially designed operators and experiments to demonstrate the performance of multiple paths generation are conducted in static rectangular environments cluttered with different polygonal obstacles configuration in Section 4. Finally, we make conclusions and describe the future works in Section 5.

2. Preliminaries

This section contains two parts. One part summarizes from [1] some materials related to cubic spiral and its planning method. The other part briefly introduces the genetic algorithms related to this work.

2.1 Review of cubic spiral and its planning method [1]

By definition, a cubic spiral is a set of curves that the direction function θ is a cubic polynomial of arc length. For an arbitrary configuration q, $[q]$ denotes its position (x, y), and (q) its direction θ. A configuration pair $[q_1, q_2]$ is said to be *symmetric* if

$$\tan\left(\frac{\theta_1+\theta_2}{2}\right)=\frac{y_2-y_1}{x_2-x_1},\ \text{if } x_1 \neq x_2$$

$$\Phi\left(\frac{\theta_1+\theta_2}{2}\right)=\pm\frac{\pi}{2},\ \text{if } x_1 = x_2, y_1 \neq y_2$$

where the angle-normalizing function Φ is defined as $\Phi(\theta) \equiv \theta - 2\pi\left\lfloor \frac{\theta+\pi}{2\pi} \right\rfloor$.

A *symmetric mean* q of any configuration pair (q_1, q_2) is a configuration that (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 $(q_1) \neq (q_2)$ or a line segment connecting q_1 and q_2 if $(q_1) = (q_2)$ (i.e. parallel configurations) (Proposition 3, [1]). The symmetric mean of two given configurations q_1 and q_2 can be defined via the position ratio γ (refer to Fig.1):

Non-parallel case: $(q_1) \neq (q_2)$

Define the center of the circle that go through the given configurations q_1 and q_2.

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

where $q_1 = (x_1,y_1,\theta_1)$, $q_2 = (x_2,y_2,\theta_2)$, and $c = \cot\left((\theta_2-\theta_1)/2\right)$. Thus the position of symmetric mean $[q_s]$ can be defined as:

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

where β_1 and β_2 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].

Parallel case: $(q_1) = (q_2) = \theta$

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

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

It is noted that a cubic spiral can connect two symmetric configurations. The local path planning method using cubic spiral can connect two given configuration q_1 and q_2: If q_1 and q_2 are symmetric, connect these two configurations with a cubic spiral directly. Else, connect these two configurations through a selected symmetric mean using two cubic spirals.

For a configuration pair (q_1, q_2), the size d is the distance between the two points $[q_1]$ and $[q_2]$, and the deflection angle α the angle between the two orientations (q_1) and (q_2). Let $q \equiv (x, y, \theta)$ represent the configuration of a car-like mobile robot (modelled as a unicycle) where (x, y) is the Cartesian position and θ is the orientation. The $q(s) = (x(s), y(s), \theta(s))$ at arc length s of a car-like mobile robot is defined by the integral equations describing the path starting from the initial configuration (x_0, y_0, θ_0)

$$\begin{cases} \theta(s) = \theta_0 + \int_0^s \kappa(t)dt \\ x(s) = x_0 + \int_0^s \cos(\theta(t))dt \\ y(s) = y_0 + \int_0^s \sin(\theta(t))dt \end{cases} \qquad (2)$$

where s is defined as 0 at the initial point (x_0, y_0).

2.2 Multi-island PGA

For simple genetic algorithm (SGA), the conflict between speedup of convergence rate and avoidance of local minimum is embodied in the selection pressure and population diversity. PGA based on island model is a parallelization scheme of genetic algorithms that can potentially reduce the execution time. 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.2), created by random sampling from each island. 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 randomly selected 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. Migration introduces diversity into a population, and the selection pressure increases monotonically with higher migration rates. It is possible to raise selection pressure in subpopulations of some islands, and concurrently to augment the population diversity in subpopulations of other islands by migration scheme. The number of generations until convergence is reduced by the additional selection pressure caused by the migration policy. [18] provided a guidance of migration sizes and intervals based on experimental observations for multi-island PGA applications.

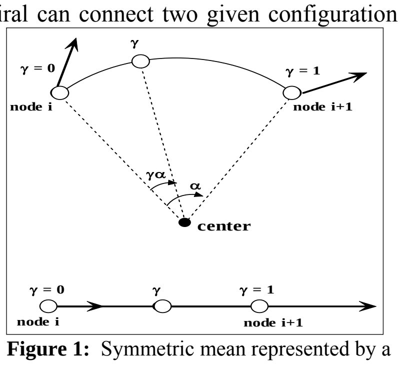


Figure 1: Symmetric mean represented by a position ratio γ. (a) non-parallel case: the symmetric mean is a circle; (b) parallel case: the symmetric mean is a line segment

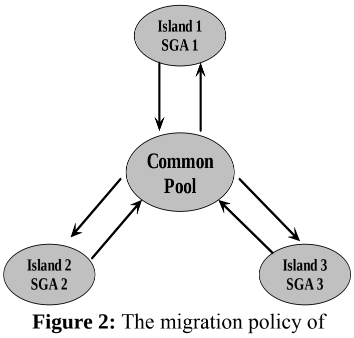


Figure 2: The migration policy of PGA based on island model

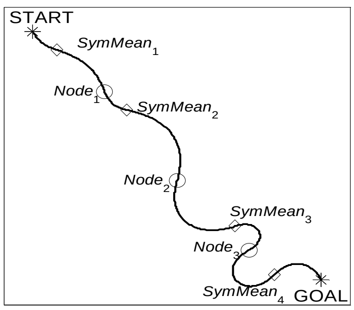


Figure 3: Path representation

3. Evolutionary Path Planning Method

For path planning of mobile robot using evolutionary search, in this paper two kinds of operators are applied to the path planning problem during evolution: one is the standard genetic operators, the other is a local path refinement operators which is highly specific to infeasible path segments (subgoals manipulation operator). Furthermore, to extract a more rich set of feasible paths from generation to generation, a filtering mechanism and partial reinitialization (called "Big-bang") is also devised.

3.1 Attributes of evolutionary process

A. Individual/path Representation

For path planning, it is most natural to use real-coded chromosome, which directly represents the path as a vector of real parameters to avoid decoding operation [8]. In this paper, a path is composed of cubic spiral segments connecting the given initial and goal configurations through a prespecified number of ordered intermediate points. Let the number of intermediate points of a path be *N*, then *N+1* pairs of configurations need to be connected by cubic spirals, or in total *N+1* symmetric means are to be determined for these segments. As the symmetric mean is located on a circle or a line connected by two given configurations, it can be represented by a position ratio, as depicted in Fig. 1. Additionally, the chromosome consists of an additional gene representing the position ratio (i.e. symmetric mean) between the last intermediate subgoal and Goal configuration. Therefore, a path is encoded as a chromosome of *4N+1* genes by a sequence of ordered cubic spirals. Concatinating the representation of each subpath, a path is represented by:

$$(\gamma_1 x_1 y_1 \alpha_1, \gamma_2 x_2 y_2 \alpha_2, \cdots, \gamma_N x_N y_N \alpha_N, \gamma_{N+1}) \cdot$$

Fig.3 illustrates the cubic spiral path representation. Each path is composed of a set of adjacent subpaths, which are two cubic spirals connecting two configurations through their respective symmetric mean.

B. Fitness

In this paper, we consider 2D, static and known rectangular environments with (convex or concave) polygonal and wall-like obstacles. The intrinsic cost of a candidate path is the total number of intersected sampled points of a path with all existing obstacles, or how many sampling nodes of path outside the boundary of map. Since the path of the car-like mobile robot is governed by the equations of motion (2), we could compute the intrinsic cost of each path according to [12]. The cost of the optimum path would be zero since all path segments need to be collision-free. Actually, we arbitrate every sampling node of a path that is located inside or outside of a specific obstacle. For checking the intersection of a cubic spiral segment with a wall-like obstacle, we check if its end points are on the same side of the wall. The path planning problem is to search for solutions that result in a minimum intrinsic cost. The fitness of an individual (candidate solution) is inversely proportional to its rank value based on its intrinsic cost. A rank-based fitness assignment strategy is adopted in this research. The path planning problem is then formulated as searching for chromosomes/paths to maximize the fitness.

C. Genetic operators

The genetic operators (includes crossover, mutation, and selection) modifies the population of candidate paths to create a new (and hopefully more feasible) population of paths. Real coded GA requires a special design of crossover and mutation operators to work with real parameters to ensure that the parameters stay within the specified upper and lower limits.

1). Crossover

The crossover operator is the main search operator of GA by sharing the information between chromosomes. It is implemented in this work as a gene level crossover: a linear interpolation between two chromosomes which can be executed component-by-component rather than to the whole chromosome, also called arithmetic crossover [10]. The crossover of two chromosomes θ_1, θ_2 in the gene level produces two offspring $\overline{\theta}_1, \overline{\theta}_2$ in accordance with the gene-wise equation:

$$\begin{cases} \overline{\theta_1} = \gamma\theta_1 + (1-\gamma)\theta_2 \\ \overline{\theta_2} = \gamma\theta_2 + (1-\gamma)\theta_1 \end{cases}$$

where γ is a random number in (0,1) and γ is different for different genes. The population mean is the same before and after the crossover. The crossover rate can be defined as the ratio of the expected number of offspring produced by crossover in the population.

2). Mutation

Mutation is a mechanism introduced to restore lost or unexpected genetic material into a population. The operation prevents the premature convergence of GA to sub-optimal solutions, thus increasing the probability of finding the global optimum solution. In this work, the operation is executed to replace selected genes/variables of chromosomes by random numbers that are within the limits of variables. Thus, random modification of paths is enforced. The rate is defined as the ratio of the expected number of offspring introduced by mutation in the population. Let *p* denote one of the four shape variables (x, y, deflected angle and position ratio) of chromosome in a population. Assuming p_{max} and p_{min} be its upper and lower bounds. The mutation operation applied to a gene *p* with a probability mutation rate whether it should be mated or not. It is defined as:

$$p' = p_{\min} + c \cdot (p_{\max} - p_{\min}),$$

where c is a random number between 0 and 1.

3). Selection

In general, there are mainly two selection operators: roulette wheel and tournament selection, both have similar effects on convergence. The roulette-wheel selection operator is employed in this work to allow the chromosomes with a higher fitness value to have a higher chance to be selected. For faster convergence, elitism [9] is adopted for preserving the best two individuals from generation to generation, which may not be transferred through the genetic operators of selection, crossover and mutation.

D. Subgoals manipulation operator

For the mobile robot path planning problem, it is important to consider how to elaborate an infeasible path into a more acceptable path. To this aim, we further design a local path refinement operator, subgoals manipulation operator, operating 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 manipulates those nodes in a predefined neighborhood, so that this local refinement of path shape occurs in the sun-regions. Here the manipulation is primarily based on the mutation operator. Fig.4 shows the diagram of this operator. The second path segment of original path (solid path) crosses an obstacle, and the operator randomly shifts the configuration of *Node 2* to a new neighboring location to obtain new path (dashed path), very possibly becoming collision-free.

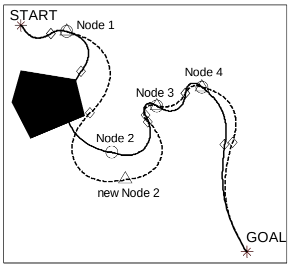


Figure 4: The subgoals manipulation operator

E. Filtering and Big-bang Activity

Despite of the introduction of the subgoals manipulation operator, another difference with the traditional PGA is filtering and Big-bang activity. Since GA needs to retain better solutions to reproduce the new population, it might lead GA to search the same region repetitively. To avoid the situation that the population contains mostly similar paths, our PGA incorporates a filtering mechanism for a continual extraction of distinct feasible paths from the evolutionary search process: For each generation, the feasible paths are chosen from the population and they are compared with the contents of the external file (the file is initially empty). Those that are not similar to the paths of the external file are picked up to the external file without further evolution. Furthermore to retain the already found feasible paths and create new feasible paths in the current and later evolutions, those that are similar to the paths within the external file are eliminated, and an equal number of randomly generated chromosomes are put into the population to form a new population for further evolution. This strategy of restart search, called Big-bang activity, avoids a regeneration of similar feasible paths that have already been generated, thus accelerating the search process and increasing the diversity of paths generated.

Given two generated paths with the same start and goal, a similarity measure of two paths/chromosomes is introduced as follows to recognize similar paths. Given two paths linking the same start and goal configurations, and N intermediate configurations, let $({}^m x_1, {}^m y_1),\ldots,({}^m x_N, {}^m y_N)$ denote the positions of all subgoals, ${}^m\alpha_1,\ldots,{}^m\alpha_N$ the variation of orientations, and ${}^m\gamma_1,\ldots,{}^m\gamma_N$ the position ratios in the *m-th* path (or chromosome). comparison of the two paths is made according to their position, deflection angle, and position ratio of all control nodes as follows:

$$\begin{cases} NormP_i = \left\| \left({}^m x_i, {}^m y_i\right) - \left({}^n x_i, {}^n y_i\right) \right\| / MapDist & \\ NormA_i = \left| {}^m\alpha_i - {}^n\alpha_i \right| / \pi & ,i = 1 \sim N \\ NormR_i = \left| {}^m\gamma_i - {}^n\gamma_i \right| / 0.6 & \end{cases}$$

where *MapDist* equals the diagonal distance of the map, m and n are the indices of two paths/chromosomes in a specific (sub)population. The four shape variables of a cubic spiral path segment are normalized by their respective ranges: *MapDist*, π, and 0.6 are allowable ranges of the path variables during evolution. If the maximum of these normalized terms is larger than a predefined threshold (0.1 in all the simulations of the paper), we say that the two paths are distinct (dissimilar).

3.2 Evolutionary path planner based on multi-island PGA

Pseudo code of the proposed multi-island PGA for path planning is described as follows:

τ = 0. //τ : the generation number
Initialize $P_i(\tau)$. // $P_i(\tau)$: population of chromosomes
// in the i-th island at generation τ
While (termination condition is not satisfied) do
For i = 1 to the number of islands
- Check the Big-bang activities.
- Evaluate the fitness $f(P_i(\tau))$
- Extract all chromosomes $P_i'(\tau)$ that fulfil the curvature constraint and the natural constraint of cubic spiral.
- Evaluate the fitness $f(P_i'(\tau))$.

While size of $P_i(\tau+1)$ < size of $P_i(\tau)$ -2-N_m do
- If $P_i'(\tau)$ exists,
  Put the best two individuals in $P_i'(\tau)$ into $P_i(\tau+1)$.
  Select N_m sets of individuals to be manipulated based on $f(P_i'(\tau))$. Select two parents θ_{i1} and θ_{i2} from $P'_i(\tau)$ based on $f(P'_i(\tau))$; otherwise, select two parents θ_{i1} and θ_{i2} from $P_i(\tau)$ based on $f(P_i(\tau))$. Perform genetic operations on θ_{i1} and θ_{i2} to generate θ_{i1}' and θ_{i2}' according to crossover rate $\varpi_c(i)$ and mutation rate $\varpi_m(i)$. Put θ_{i1}' and θ_{i2}' into $P_i(\tau+1)$.

End While
- Randomly select N_i migration members from the i-th island and copy them into randomly selected positions in the common pool. Randomly select N_i migration members from the common pool and copy them into randomly selected positions in the i-th island.

End For
$\tau = \tau + 1$.
End While

The initial subpopulations for the islands are created by randomly generated x-monotone positions, deflection angles within a range of –π/2 to π/2, and position ratios of symmetric means in the range of 0.2 to 0.8, of all intermediate points within the predefined ranges. The subpopulations of each island have constant population size (100 individuals per generation in this paper).

4. Experimental Results

In this section, simulations are conducted for investigating the effectiveness of the specially designed operations for 50 independent runs. Performance of multiple paths generation of our evolutionary planner is shown for rectangular environment that consists of six polygonal obstacles with different initial configurations.

Table 1: Parameters setting for comparison of effectiveness of subgoals manipulation operator

	SGA-1	*SGA-2*
No. Intermediate Points	5	
Population Size	100	
Evolutionary Generations	200	
Crossover rate , Mutation rate	0.71,0.01	
No. manipulations	none	20/***generation***

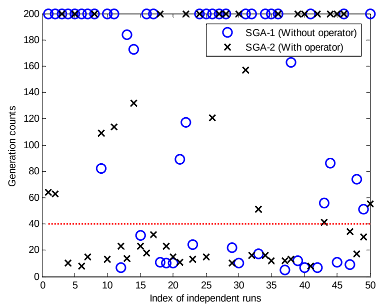


Figure 5: The generation that first feasible path is found, recorded for 50 runs of SGA with or without the subgoals manipulation operator.

Table 2: Parameters setting for 3-island PGA

Crossover rate, Mutation rate	(0.71,0.01) (0.1,0.9)(0.4,0.3)
No. migration	6/ ***island***
No. manipulations	20/***generation***

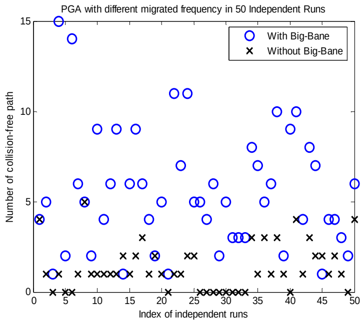


Figure 6: Number of distinct feasible paths of 3-island PGA with or without the Big-bang activity in 50 independent runs.

Table 3: Number of accumulated distinct feasible paths with varied migration intervals

Migration intervals	1	5	10	15	20
Number of path	110	102	78	102	67

4.1 Effectiveness of subgoals manipulation operator and Big-bang

A. Effectiveness of Subgoals manipulation operator for locally repair the infeasible segments

For validating the necessity of subgoals manipulation operator, we perform a series of repeated runs of two SGAs, one without (SGA-1) and the other with (SGA-2) the subgoals manipulation operator for validating its effectiveness. The parameters setting for this comparison can be found in Table 1. Fig.5 shows the following: (i)In 50 runs, SGA-1 has a success rate of 50% and SGA-2 has a success rate of 68% that can find a collision-free path within 200 generations. It is clear that the subgoals manipulation operator helps raising the probability of finding a feasible path more efficiently. (ii) Among 50 runs, SGA-1 and SGA-2 have 30% and 48%, respectively, that can find a collision-free path within the first 40 generations (below the horizontal dotted line in Fig.5). In both cases, SGA is more likely to premature in early generations, i.e. so-called genetic-drift, where there are largely unexplored or under-explored regions of search space, indicating that SGA rapidly converged in an area of search space.

B. Effectiveness of Big-bang

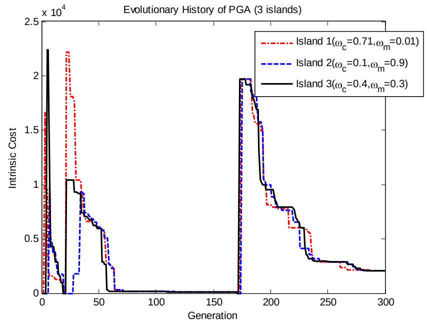
Fig.6 shows that the number of distinct collision-free paths found by the 3-island PGA with Big-bang activity are most more than or at least equal to that without Big-bang activity in each of 50 independent runs with the same initialization in both schemes.

4.2 Effect of migration intervals

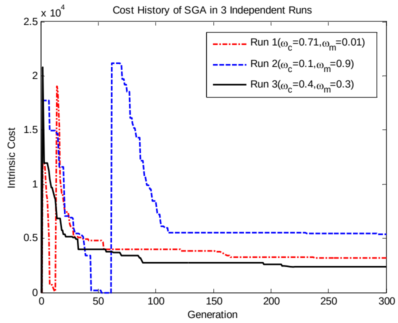
Table 3 show the number of distinct collision-free paths with varied migration intervals. As the 3-island PGA migrates every generation, the performance of PGA for generating multiple paths in this testing environment can be improved. This indicates that for the purpose of efficiently generating as many as possible feasible paths, any means that increase diversity of evolution or prohibit similarity/duplication of individuals can have a positive effect on searching performance of evolutionary path planner.

Table 4: Parameter setting for comparison of PGA and SGAs

	3- island PGA	***3independent SGAs***
Population Size	100/ island	300/run
No. Generations	300	
No. migration	6/ island	none
No. manipulations	20***s/generations***	60
Big-bang	Yes	



(a)



(b)

Figure 8: Evolution of intrinsic cost. (a) PGA (b) 3 separate SGAs.

4.3 Comparison of the PGA and SGA

Both the subgoals manipulation operator and the Big-bang activity are implemented in this comparison. Table 4 shows the parameters setting for three-island PGA and three separate SGAs. Each island/SGA has a different crossover and mutation pair of rates in this comparison. Fig.8 illustrates the evolution of the best cost of three subpopulations of PGA and 3 independent runs of SGA in each island, whose performance strongly depends on the crossover and mutation rates. For PGA, solutions are obtained for each simulation, while no collision-free path is found for SGA within the maximum number of generations in the worst case. PGA based on island model shows significant improvements to finding more feasible paths.

4.4 Multiple paths generation using PGA

Fig.9 shows the searching results with varied start orientations that would cover the relative different situations between the start and goal. In all three cases performed, PGA shows its adaptation to successfully generate at least 5 collision-free paths. It is likely that different runs of PGA may generate different number of possibly different feasible paths. This is due to the experience acquired during the exploration phase of evolution can affect the later convergence/exploitation outcome.

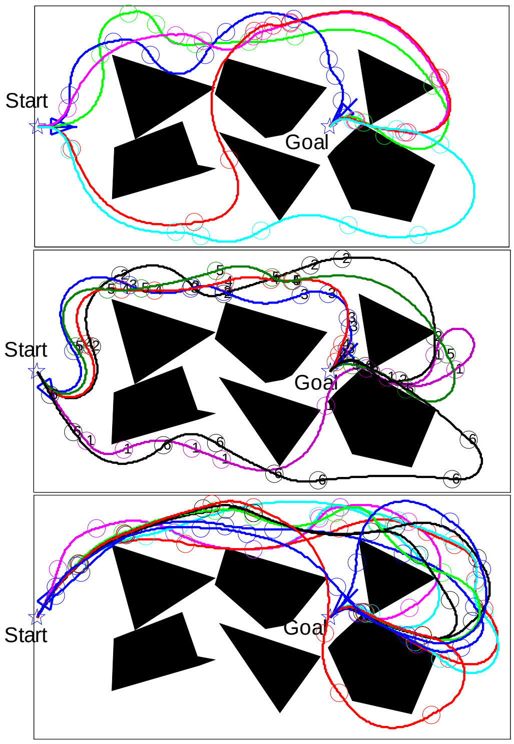


Figure 9: Multiple-path planning (4 intermediate nodes) for environment-I as start orientation is varied (3 cases)

5. Conclusion and Future Works

In this paper, a PGA based on island model is applied to explore potential feasible paths for a car-like mobile robot. Our approach is effective to generate multiple alternative ordered sequences of a pre-specified number of intermediate collision-free configurations that smoothly moves a car-like mobile robot from an initial configuration to a goal configurations to make the robot move along a configuration in an obstacles obstructed environment. Cubic spiral paths are used to linking these smoothly directed planar curves. Actually, subgoals manipulated operator, filtering and big-bang activity, as justified by simulations, effectively improve the evolutionary search capability, in that more distinct collision-free paths are discovered during the evolution. This study shows that for generating as many as possible feasible paths, any means of increasing the diversity of evolution, such as we have employed frequent migration and big-bang, are effective for this purpose. As for the incorporation of more criteria of paths, like curvature constraint and length minimization [2], in obstructed environments can be tackled by a path selection process from those collision-free paths generated, but no guarantee. This multi-objective path optimization problem is our work in progress. Furthermore, cubic spiral path planning in the dynamic environments is a challenging problem in the future.

6. Acknowledgements

This research was supported by National Science Council, R.O.C., under contract no. 94-2212-E-001-001.

7. References

[1] Y.J. Kanayama and B.I. Hartman, “Smooth local path planning for autonomous vehicles,” *Int. J. Robot. Res.*, Vol.16, no.3, pp. 263-283, 1997.

[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, pp. 312-335, 2005.

[3] C. Hocaoglu and A.C. Sanderson, “Planning multiple paths with evolutionary speciation,” *IEEE Transactions on Evolutionary Computation*, Vol.5, No.3, pp. 169-191, 2001.

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

[5] M. Gemeinder, and M. Gerke, “GA-based path planning for mobile robot systems employing an active search algorithm,” *Applied Soft Computing*, Vol.3, pp. 149-158, 2003.

[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, pp 18-28, 1997.

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

[8] A.H. Wright, “Genetic Algorithms for real parameter optimization.” In G. Rawlins, ed., *Fundamentals of Genetic Algorithms*, Morgan Kaufmann, 1991, pp. 205-218.

[9] K.A. De Jong, *An Analysis of the Behaviour of a Class of Genetic Adaptive Systems*, Ph.D. Thesis, University of Michigan, Ann Arbor, 1975.

[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. System, Man and Cybernetics*, Part B, vol. 33, 2003, pp. 250-257.

[11] J.C. Latombe, *Robot motion planning*, Kluwer Academic Publisher, New York, 1991.

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

[13] Z. Skolicki and K. De Jong, “The influence of migration sizes and intervals on island models,” *GECCO’05*, pp. 1295-1302, 2005.

[14] Y. Hu and S.X. Yang, “knowledge based genetic algorithm for path planning of a mobile robot,” *2004 IEEE Int. Conf. Robotics and Automation,* pp. 4350-4355, 2004.

[15] K. Sugihara and J. Smith, “Genetic algorithms for adaptive motion planning of an autonomous mobile robot,” *IEEE Int. Conf. Computational Intelligence in Robotics and Automation*, pp. 138-141, 1997.

[16] P.N. Azariadis and N. KA. Aspragathos, Obstacle representation by Bump-surfaces for optimal motion-planning, *Robotics and Autonomous Systems,* vol.51, pp. 129-150, 2005.

[17] G. Dudek and S. Simhon, Path planning using learned constraints and preferences, *2003 IEEE Int. Conf. Robotics and Automation*, pp. 2907-2913, 2003.

[18] Z. Skolicki and K. De Jong, “The influence of migration sizes and intervals on island models,” *GECCO’05*, pp. 1295-1302, 2005.