

# Collision-free Composite $\eta^3$ -Splines Generation for Nonholonomic Mobile Robots by Parallel Variable-Length Genetic Algorithm

Jiun-Hau Wei and Jing-Sin Liu

Institute of Information Science, Academia Sinica, Nangang, Taipei, Taiwan 115, ROC  
bloodiron@gmail.com, liu@iis.sinica.edu.tw

## Abstract

*A  $G^3$ -continuous path planner for wheeled nonholonomic mobile robots is presented, where the path is a composite curve composed of a number of  $\eta^3$ -splines segments. Based on the variable-length genetic algorithm implemented in the architecture of island parallel genetic algorithm (IPGA), the path planner automatically selects the number and locations of a sequence of intermediate nodes aiming for collision-free shorter composite  $\eta^3$ -spline path generation. The experimental result demonstrates the capability of the proposed evolutionary path planner.*

## 1. Introduction

Recently various researches of mobile-robots path planning using different path primitives and planning schemes in a given static environments has been presented in the literature, genetic algorithm in particular [1]-[6], [11]-[18]. In general, path planning in robotics can be formulated as nonlinear constrained optimization problems in configuration or Cartesian space. Numerical techniques are devised to solve the problems in different context. Among them, genetic algorithms (GA) has attracted a lot of attentions due to their parallel problem solving capability and adaptability for large scale hard combinatorial optimization problems, and no need to compute the gradients of the objective functions. In addition, GA-based path planner has a merit of producing several alternative paths in a single run. Practically, the mechanical operation of robots gives severe demands of planned paths. For fuel efficiency, the shortest path is favored; however, under some circumstances, the robots can encounter a sudden change of direction, causing its speed drops dramatically, consuming more fuel. To prevent this, the path with continuous and limited curvature 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. Introducing proper primitive curve as path segments is useful to smooth path generation, and numerous attempts of employing novel parametric curves (e.g. [1], [8], [12]) and splines (e.g. [11]) to mobile robot path planning are shown. Among these,  $\eta^3$ -splines published by Piazzzi et al [8] reveal its ability to superiorly approximate a rich class of practically useful, commonly used path primitives in wheeled mobile robot path planning such as straight line segment, circular arc, clothoid (Conru spiral) [2]. More importantly, it can also be shown that the  $\eta^3$ -splines can represent any seventh-order polynomial curve with third order geometric continuity, denoted as  $G^3$  continuity [8] (continuous in position, curvature, and derivative of the curvature). The superior features of  $\eta^3$ -splines motivate us to apply it on path planning of wheeled mobile robots.

To improve the efficiency of evolution toward feasible paths, parallel genetic algorithm [14] was proposed as a viable means. Island parallel genetic algorithm (IPGA) [10] provides a path computing framework which allows for more flexibility in searching for useful solutions, since it includes several design parameters that can be tuned to improve actual and statistical performance. As demonstrated in [7], IPGA may be well adapted for solving smooth path planning problems, since each island can exploit separate and distinct paths. In [7], IPGA is designed for discovery of paths resolving a bi-objective path-planning problem of generating a shortest collision-free paths in stationary obstructed environments, where a path is composed of a pre-specified number of cubic spiral segments whose size and deflection angle (i.e. change of heading) are unknown parameters to be optimized. For generating a composite path, the complexity of path planning increases dramatically with the number of path segments (i.e. the number of waypoints, or the degrees of freedom). A larger number, though provides more flexibility, requires more computation time and memory [17], and could be harmful [4]. Deciding a priori adequate or

optimal number of nodes satisfying the shortest length criterion is hard even in free environment. We have performed an extensive comparative simulation study of the number of waypoints on the success rate, path length, and speed of finding a feasible path for IPGA based composite  $\eta^3$ -spline path planner in simple environments. As an attempt to overcome the problem of decision of number of waypoints, a variable-length genetic algorithm, which adopts variable chromosome size for initial population, for IPGA-based path planner is proposed with the Pareto-based evolutionary multi-objective optimization using IPGA, which can be shown to be robust to chromosome size within a specified range. Once the initial parameters, including the ranges of number of nodes, are set, the evolutionary path planner will self adjust via evolution, so that the number of nodes and the location of each node satisfying the required boundary constraints on the end points are automatically selected according to the environment map size, the locations of START and GOAL, the configuration, i.e. density (number and distribution) and geometric complexity of the obstacles.

This rest of this paper is organized as follows. Section II gives a brief introduction of the  $\eta^3$ -splines, the primitive curve segments used in path planning. Section III shows the implementation of proposed variable-length genetic algorithm. Section IV describes the implementation of IPGA. Section V presents our simulations revealing the variations in the number of intermediate nodes. Section VI gives the conclusion.

## 2. The $\eta^3$ -splines

All Following Piazzzi et al [8], given two arbitrary configurations

$$\Omega_A = [x_A, y_A, \theta_A, \kappa_A, \dot{\kappa}_A]^T$$

and

$$\Omega_B = [x_B, y_B, \theta_B, \kappa_B, \dot{\kappa}_B]^T,$$

where  $x(s), y(s)$ , and  $\theta(s)$  denote the coordinate and orientation, and  $\kappa$  and  $\dot{\kappa}$  are the curvature and curvature derivative of the path at that point. By these two configurations, a seventh-order polynomial of a  $G^3$ -interpolating curve can be generated through:

$$P(s) = [p_x(s), p_y(s)]^T,$$

$$p_x(s) \equiv \alpha_0 + \alpha_1 s + \alpha_2 s^2 + \alpha_3 s^3 + \alpha_4 s^4 + \alpha_5 s^5 + \alpha_6 s^6 + \alpha_7 s^7;$$

$$p_y(s) \equiv \beta_0 + \beta_1 s + \beta_2 s^2 + \beta_3 s^3 + \beta_4 s^4 + \beta_5 s^5 + \beta_6 s^6 + \beta_7 s^7.$$

where  $s \in [0, 1]$  is a parameterization. Besides

the given terminal configurations, this seventh-order polynomial curve still has additional six degrees of freedom, which is stated through the  $\eta$  vector:

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

One way to reduce the degrees of freedom is 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

$$\begin{cases} \Omega_A = [x_A, y_A, \theta_A, \kappa_A = 0, \dot{\kappa}_A = 0]^T \\ \Omega_B = [x_B, y_B, \theta_B, \kappa_B = 0, \dot{\kappa}_B = 0]^T, \end{cases}$$

resulting in one family of the  $\eta^3$ -splines with both terminal points of zero curvature and zero derivative of curvature. The family has been numerically demonstrated its capability in its effect of minimizing the maximal curvature, which is a heuristic measure of the maximal centrifugal force experienced along the path.

Following the standard formulae given by Piazzzi, this convention of choice in the  $\eta$  vector completely determines the primitive curve segment for further path planning in obstacles free environment. For the  $x$ -coordinate coefficients, we have:

$$\alpha_0 = x_A$$

$$\alpha_1 = \eta_1 \cos \theta_A; \quad \gamma_1 = \eta_2 \cos \theta_B;$$

$$\alpha_2 = 0; \quad \gamma_2 = 0$$

$$\alpha_3 = 0; \quad \gamma_3 = 0$$

$$\alpha_4 = 35(x_B - x_A) - 20\alpha_1 - 10\alpha_2 - 4\alpha_3 - 15\gamma_1 + 5\gamma_2 - \gamma_3$$

$$\alpha_5 = -84(x_B - x_A) + 45\alpha_1 + 20\alpha_2 + 6\alpha_3 + 39\gamma_1 - 14\gamma_2 + 3\gamma_3$$

$$\alpha_6 = 70(x_B - x_A) - 36\alpha_1 - 15\alpha_2 - 4\alpha_3 - 34\gamma_1 + 13\gamma_2 - 3\gamma_3$$

$$\alpha_7 = -20(x_B - x_A) + 10\alpha_1 + 4\alpha_2 + \alpha_3 + 10\gamma_1 - 4\gamma_2 + \gamma_3$$

Similar formulae for the  $y$ -coordinate coefficients can be easily obtained by replacing  $\cos\theta$  to  $\sin\theta$ .

## 3. The variable-length genetic algorithm for path planning

### 3.1. Individual representation: variable-length chromosome

Using  $\eta^3$ -splines as path primitive, a composite path is composed of connected splines. A  $N+1$  connected segments of  $\eta^3$ -splines can be fully defined by  $N$  control points, including start and end terminal. One choice of chromosomes is the collection of  $N$  ordered control points represented as

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

where  $\vec{r}_i$  is the position vector of waypoint  $i$ , and  $\theta_i$  is the tangent direction of the path at the waypoint  $i$ ;  $i=1$  corresponds to start location,  $i=N$  corresponds to goal location. This results in a composite  $\eta^3$ -spline curve which has continuity of position, curvature and curvature derivative at the  $N$  intermediate nodes. Notably,  $N$

is not fixed a priori, but only given a range with upper and lower limits.

### 3.2. Mutation, Crossover, and Selection

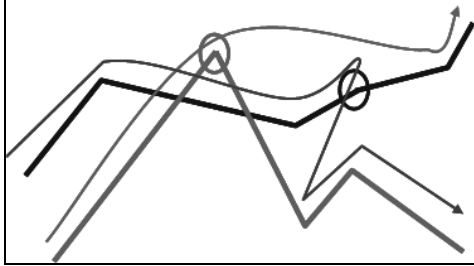


Fig. 1 Variable-length crossover operation

The mutation operator is defined as the random perturbation of a randomly selected node within its allowable range of a randomly selected path [7]. The crossover operator is defined as the random combination of intermediate nodes between two individual as shown in Fig. 1. In the beginning, two parent paths are randomly selected. Each parent breaks about a randomly selected crossover node into two path segments. Then two child paths are directly obtained by recombining broken segments with different permutation. Suppose the numbers of nodes of two parents are  $\alpha$ ,  $\beta$  (including start and goal). Then depending on the crossover nodes selected, crossover produces two child paths with possible number of nodes between  $\alpha+\beta-2$  and 2. Note individual can do crossover more than once, and if individuals with more nodes do crossover more frequently, the total node number, also the average node number, grows, or vice versa, it decades.

The quality of solution is, of course, proportional to total length of collision-free path segments, or it should be penalized otherwise. One approach to quantize the extent of penalty is the intrinsic cost function [4], simply summing up the lengths of all forbidden path segments; despite this, it only counts the amount of violation, lacking the ability for giving the directions of improvement. In order to accelerate the convergent process, the modified intrinsic cost proposed in [7] incorporated the penetration depth for its forbidden segment helps. The new intrinsic cost of an evaluated path gives a good measure of how difficult is to make this path collision-free, which is empirically shown to be a better fitness function [7]. For the bi-objective optimization, we use the fast non-dominance sorting method proposed by Deb [9] to rank both the total path length cost and the total modified intrinsic cost, gene-

rating the Pareto fronts to be selected by the rank-based selection operator.

### 4. Parallelism of the genetic algorithm

The trade-off between diversity and convergent speed is the most critical issue in genetic algorithms. In general, diversity can be enhanced by using high mutation rate or low selection pressure but either degrades the convergent speed and solution quality. Conversely, high crossover rate and high selection pressure improves the speed and quality, paid in low diversity, possibly low success rate. The island parallel genetic algorithm (IPGA) [10] is chosen as a means of allowing a suitable balance of diversity and speed of convergence. The parallelization scheme can also reduce the execution time [14].

IPGA starts with dividing the population into several isolated subpopulations, called islands; each evolves independently with different user-setting crossover and mutation rates, except exchanging a small portion of best individuals periodically via a common pool serving as a migration center. The parameters setting for each island are usually quite different in order to have high-diversity and high-convergent-speed evolution within different islands. While these islands exchange their individuals via migration, these ultimate evolution parameters is injecting 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 number and too small interval will break the isolation among islands, implicating poor diversity.

### 5. Experimental result

The path planner is implemented in the programming language C++, and the results are dumped into text-based file for further processing. Our current implementation of IPGA is composed of three equal-sized islands, which differ in settings of mutation rate and crossover rate, and migrates ten-percent of the total population size every five generations. Consequently, with sub-population size of 100 individuals in each island while a total of 30 individuals, 10 for each island, migrates in the given migration period. Moreover, in order to preserve diversity, three islands are configured as high mutation rate, high crossover rate, and balanced one respectively. Maximum number of generations is set as 50.

To demonstrate the capability of the path planner based on island parallel variable-length GA, six maps

with different size, which are tested in [4] by fixed – length GA, are examined again to characterize the behavior. 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. Here we present some selected simulation results shown in Fig. 2 to Fig. 7, where the shown paths may be of different number of intermediate nodes. The simulation results of proposed evolutionary multi-objective path planning scheme are summarized below:

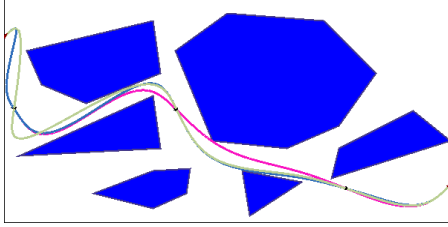


Fig. 2 Paths discovered in Map I, 600x300

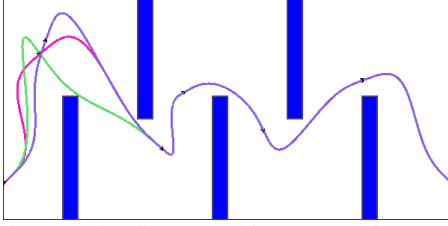


Fig. 3 Paths discovered in Map II, 600x300

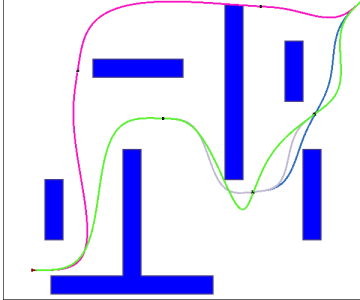


Fig. 4 Paths discovered in Map III, 600x500

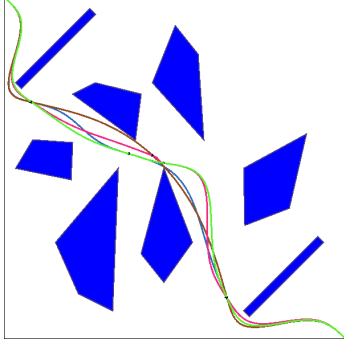


Fig. 5 Paths discovered in Map IV, 600x600

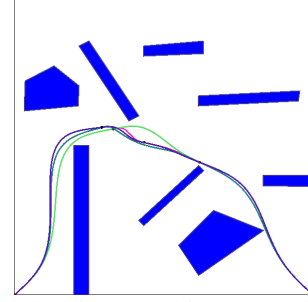


Fig. 6 Paths discovered in Map V, 600x600

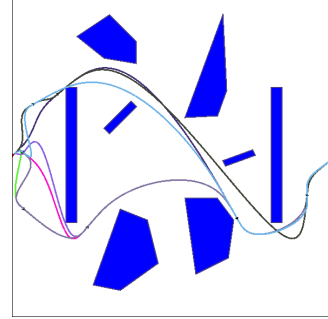


Fig. 7 Paths discovered in Map VI, 600x600

## 5.1. Evolution Characteristics

To avoid excessive computation and longer convergence, the setting of range of gene length for initial population is between 4 and 10 (including start and end terminals). The initial population consisting of variable length chromosome is generated randomly, and an example is shown in Fig. 8. With  $avg(L)$ , the average length of all paths,  $min(L)$ , the minimum length of “feasible” paths, and  $avg(M)$ , the average modified intrinsic cost of all paths in a simulation run, simulation results of six maps are listed from Fig. 9 to Fig. 14. As summarized in [4], map II reveals the highest difficulty which optimal successful rate and path length are met with larger gene length. Fig. 8 compares the average of converged gene length among these maps; for complex maps, e.g. map II, the converged value is higher than the others. Vice versa, the simpler one, e.g. map III, shows lower value.

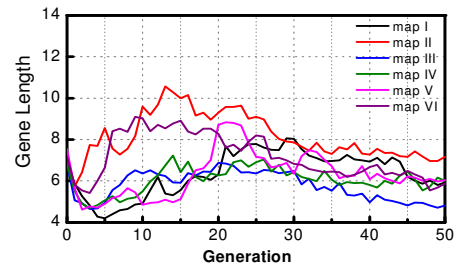


Fig. 8 The average gene length for six maps

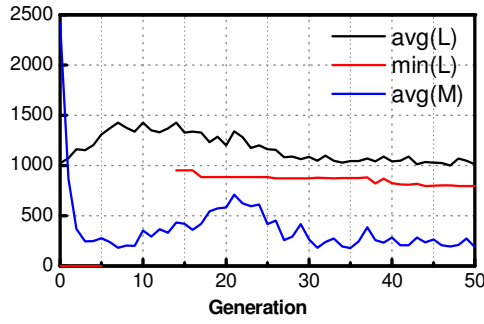


Fig. 9 Evolution of costs for Map I

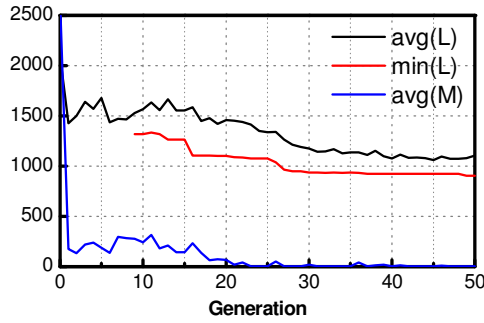


Fig. 10 Evolution of costs for Map II

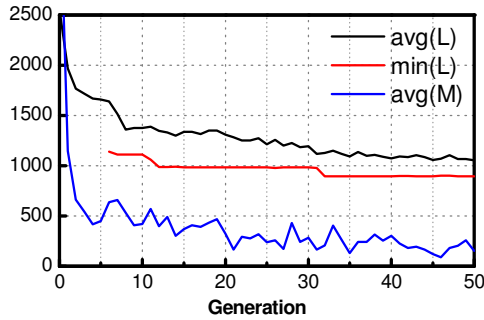


Fig. 11 Evolution of costs for Map III

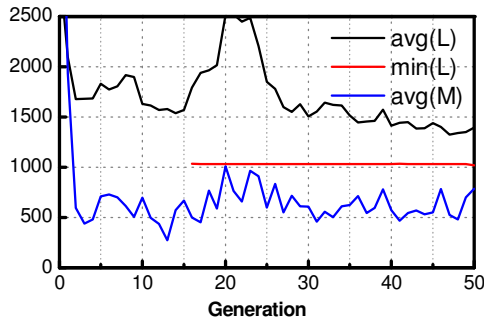


Fig. 12 Evolution of costs for Map IV

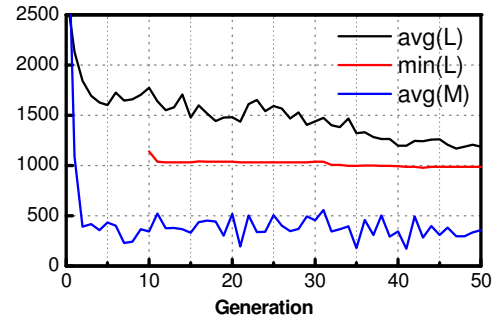


Fig. 13 Evolution of costs for Map V

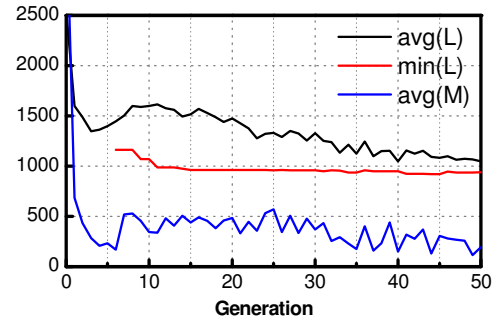


Fig. 14 Evolution of costs for Map VI

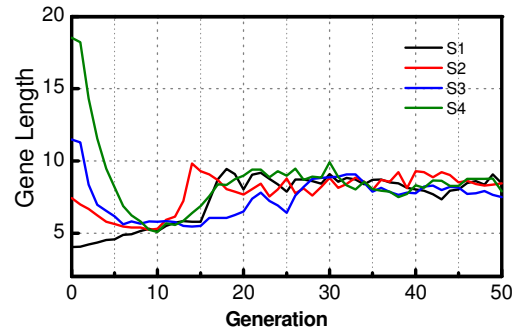


Fig. 15 Convergence of gene length for map II with different initial setting of range of gene length

## 5.2. Robustness

The variable-length GA shows superior robustness on the initial setting of gene length. This robustness makes the proposed evolutionary path planner based on island parallel variable-length GA preferable to other non-evolutionary path planning algorithms which may require a manual setting of the number of waypoints. In order to examine this property, map II is chosen to be demonstrated with four different initial gene length sets: S1: 3~5, S2: 6~9, S3: 10~13, S4: 17~20.

The convergence of gene length during the evolution is shown in Fig. 15. It can be observed that the range of initial genetic length is not a critical parameter but affects diversity and convergent speed of the evolution process. However, too high or too low initial gene length setting in these maps will cause either a longer converging process or worse, failing to find feasible paths in given generations. But if evolving enough generations, the convergent gene length depends only on the map feature, thus revealing the robustness to variation in initial gene length setting.

## 6. Conclusion

To plan a collision-free shorter composite  $\eta^3$ -splines path in static environment, the selection of waypoints is important. This paper presented an IPGA based path planner that automatically selects the number and locations of intermediate nodes. The chromosome size (or the number of intermediate nodes of composite curve) is not fixed a priori, but is self-adjusting according to the complexity of environment through the variable-length evolution rules, resulting in a more robust path-planner. Compared to our work [4] using fixed-length IPGA on the same problem, variable-length scheme offers a more robust and efficient approach to discover one or multiple alternative reasonable good paths in a variety of environments

## 7. Reference

- [1] O. Castillo, L. Trujillo and P. Melin, "Multiple objective genetic algorithms for path-planning optimization in autonomous mobile robots," *Soft Computing*, Vol.11, No. 3, pp.269-279, 2007.
- [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] HC Chang and JS Liu, "High quality path planning for autonomous mobile robots with  $\eta^3$ -spline and parallel genetic algorithm," *2008 IEEE International Conference on Robotics and Biomimetics*, Bangkok, Thailand, Dec. 2008.
- [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, "Smooth path planning for a mobile robot by evolutionary multiobjective optimization," *IEEE Int. Symposium on Computational Intelligence in Robotics and Automation*, Jacksonville, Florida, June 2007.
- [8] A. Piazzzi, Corrado Guarino Lo Bianco, and M. Romano, " $\eta^3$ -Splines for the Smooth Path Generation of Wheeled Mobile Robots", *IEEE Trans. Robotics*, Vol. 23, No. 5, pp. 1089-1095, 2007.
- [9] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-2," *IEEE Transaction on Evolutionary Computation*, Vol. 6, No. 2, 181-197, 2002.
- [10] K.A. De Jong, *Evolutionary Computation: a unified approach*, MIT Press, Cambridge, Massachusetts, 2006.
- [11] V. Munoz, A. Ollero, M. Prado, A. Simon, "Mobile robot trajectory planning with dynamic and kinematic constraints," *IEEE Int. Conf. Robotics and Automation*, pp.2802-2807, 1994.
- [12] V.A. Spais and L.P. Petrou, "Multiobjective motion planning for a nonholonomic vehicle," *2003 IEEE Congress on Evolutionary Computation*, pp.2058-2065, 2003.
- [13] P. Gallina and A. Gasparetto, "A technique to analytically formulate and to solve the 2-dimensional constrained trajectory planning problem for a mobile robot," *Journal of Intelligent and Robotic Systems*, vol.27, pp.237-262, 2000.
- [14] J.M. Ahuactzin, E.-G. Talbi, P. Bessiere and E. Mazer, "Using genetic algorithms for robot motion planning," *ECAI 92*, Vienna, Austria.
- [15] 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.
- [16] J. Tu and S.X. Yang, "Genetic algorithm based path planning for a mobile robot," *2003 IEEE International Conference on Robotics and Automation*, pp.1221-1226, 2003.
- [17] M. Seyr and S. Jakubek, "Dynamic trajectory generation via numerical multi-objective optimization," *2007 American Control Conference*, pp.3336-3341, 2007.
- [18] D. Gallardo and O. Colomina, "A genetic algorithm for robust motion planning," *11<sup>th</sup> Int. Conf. on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, pp.115-121, 1998.