A 2–Dimensional ACO-based Path Planner for Off-line Robot Path Planning

Nuwan Ganganath
Department of Computing and Information Systems,
Wayamba University of Sri Lanka,
Kuliyapitiya, Sri Lanka.
Email: manganganath@ieee.org

Chi-Tsun Cheng
Department of Electronic and Information Engineering,
The Hong Kong Polytechnic University,
Hunghom, Kowloon, Hong Kong
Email: chi-tsun.cheng@polyu.edu.hk

***Abstract*—Wireless sensor networks are usually deployed in scenarios that are too hostile for human personnel to perform maintenance tasks. Wireless sensor nodes usually exchange information in a multi-hop manner. Connectivity is crucial to the performance of a wireless sensor network. In case a network is partitioned due to node failures, it is possible to re-connect the fragments by setting up bridges using mobile platforms. Given the landscape of a terrain, the mobile platforms should be able reach the target position using a desirable path. In this paper, an off-line robot path planner is proposed to find desirable paths between arbitrary points in a given terrain. The proposed path planner is based on ACO algorithms. Unlike ordinary ACO algorithms, the proposed path planner provides its artificial ants with extra flexibility in making routing decisions. Simulation results show that such enhancement can greatly improve the qualities of the paths obtained. Performances of the proposed path planner can be further optimized by fine-tuning its parameters.**

***Index Terms*—ACO, Path Planning, B-Spline, Resource Management, Wireless Sensor Networks**

I. Introduction

Mobile wireless platforms are widely employed in wireless sensor networks for deploying wireless sensor nodes [1], [2], maintaining network connectivity [3], [4], and collecting sensing data [5], [6]. Wireless sensor networks usually consist of massive volumes of wireless sensor nodes, which are deployed in hostile environment. Mobile wireless platforms provide network operators a low cost solution to conduct network-wide maintenance tasks. Two requirements for a mobile wireless platform are to navigate to the event location and perform specific tasks as instructed. The first requirement is also known as path planning, which can further be classified as on-line path planning [7], [8] and off-line path planning [9], [10].

In [11], Cheng *et al.* proposed an off-line path planner for co-operative UAVs based on an Ant-Colony Optimization (ACO) algorithm. Unlike ordinary ACO-based path planner, where artificial ants are only making route decisions at intersection points, the ACO algorithm proposed in [11] allows ants to make decisions at several lines, namely control point lines. Comparing with a limited number of outgoing paths at each intersection, the introduction of the control point lines allows ants to have virtually an infinite number of routes to choose from, which provide the path planner with greater flexibility. This paper is an extension to the ACO-based path planner proposed in [11]. The aim of this work is to further enhance the flexibility of an ACO-based path planner by allowing artificial ants to make routing decisions at any place within a terrain under consideration.

The rest of the paper is organized as follows. Section II elaborates the formulations of the terrain landscapes, path evaluation factors, and the objective function of the optimization problem. The proposed 2-D ACO-based path planner is explained in Section III. The proposed path planner was evaluated using computer simulations. Simulation settings and results are presented in Section IV. Performance of the proposed algorithm is analyzed and compared with the original ACO-based algorithm [11] in Section V. Concluding remarks are given in Section VI.

II. Problem Formulation

In this section, the formulations of the terrain landscapes, the path evaluation factors, and the objective function will be elaborated.

A. Terrain Landscapes

The terrain considered in this paper is a three–dimensional (3-D) hilly landscape. The base of the 3-

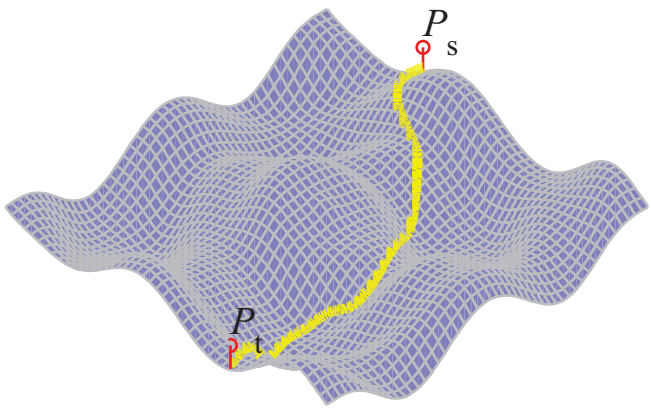

Fig. 1. An illustration of the terrain model considered for this paper. The yellow curve is representing the trajectory of a path between an arbitrary starting point P_s to an arbitrary target point P_t.

D terrain is defined as a 1-by-1 unit^2 square. The z–dimension of a position (x, y) inside a terrain is controlled by,

$$z(x,y) = \begin{array}{l}\zeta[a\sin(b\sqrt{x^2+y^2}) \\ +c\cos(x)+\sin(y+d)+e\sin(y)]^2,\end{array} \quad (1)$$

where a, b, c, d, e are arbitrary constants. Parameter ζ is the normalizing factor such that $z(x, y)$ will lie within the range $[0, 1]$.

B. Path Evaluation Factors

The aim of the proposed path planning algorithm is to find a desirable path from an arbitrary starting point P_s to an arbitrary target point P_t within the given terrain (see Fig. 1). The desirability of a path is evaluated based on the following factors.

1) Path Length: The length of a path is often proportional to the fuel consumption of a robot, and thus it is desirable to have shorter paths. The first path desirability factor (D_1) is expressed as follows

$$D_1(U) = d_\text{proj}(P_\text{s}, P_\text{t}). \quad (2)$$

Here, $d_\text{proj}(P_\text{s}, P_\text{t})$ is the length of the projection of the path U on the x–y plane that is connecting P_s and P_t. An illustration of $d_\text{proj}(P_\text{s}, P_\text{t})$ is shown in Fig. 2.

2) Turning Angle: A path with sharp turning points is undesirable as turnings often associated with decelerations, which are both time and energy consuming. In this paper, a path is represented using a B–Spline curve, which was invented by Schoemberg [12]. A B–Spline curve consists of multiple piecewise polynomial segments. B–Spline curves are often used to represent robots' trajectories due to their continuity nature. The shape of a B–Spline curve can be adjusted by moving a set of control points [13]. As the control points are populated along a B–Spline curve, an angle formed by three consecutive control points can be used as an estimate of a turning angle. Given a path U with n control points $(c_1, \cdots, c_n)$, its second desirability factor (D_2) is expressed as follows

$$D_2(U) = 180^\circ - \min_{i=2,\cdots n-1} \angle c_{i-1}c_ic_{i+1}. \quad (3)$$

An illustration of a B–Spline curve with $n = 6$ control points is shown in Fig. 2.

3) Climbing and Descending Ratio: Steep climbing and descending should be avoided whenever possible as it is fuel inefficient to a robot. It can also be dangerous as a robot may flip or slide while maneuvering on slopes. Therefore, the resultant path should avoid having rapid changes on z–dimension. After the control points of a B–Spline curve are defined, a fixed number of evaluation points (e_i) are used to represent the piecewise polynomial curves between adjacent control points. Let $z(e_i)$ be the z–dimension of the terrain at the position of evaluation point e_i. Given a path U with k evaluation points $(e_1, \cdots, c_k)$, its third desirability factor (D_3) is expressed as follows

$$D_3(U) = \max_{i=1\cdots n-1} \left|\frac{\Delta z_i}{\Delta d_i}\right| = \max_{i=1\cdots n-1} \left|\frac{z(e_{i+1}) - z(e_i)}{d(e_i, e_{i+1})}\right|. \quad (4)$$

Here, $d(e_i, e_{i+1})$ is the separation between consecutive evaluation points e_i and e_{i+1} on the x–y plane. An illustration on Δz_i and Δd_i are shown in Fig. 2.

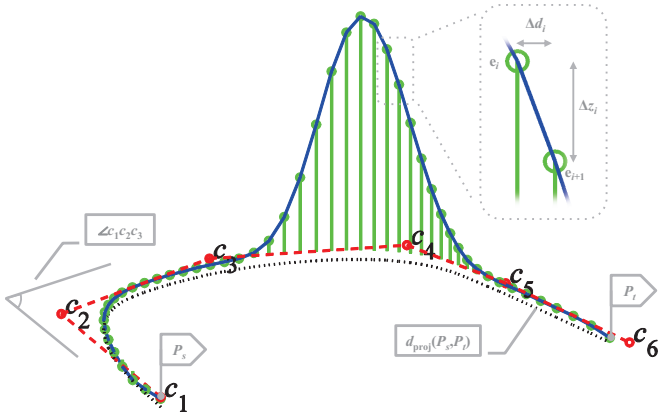

Fig. 2. An illustration of a trajectory between $P_\text{s}$and P_t. The solid line in blue is representing the path. The dotted line in black is representing the projection of the path on the x–y plane. The dashed line in red is showing the interpolation of the control points.

C. An Objective Function

A desirable path should have all the aforementioned desirability factors being minimizing. Therefore, the objective function f used in this paper is a weighted-sum function of D_1, D_2, and D_3, which is expressed as follows

$$f_{\text{obj}}(U) = \frac{w_1 D_1(U) + w_2 D_2(U) + w_3 D_3(U)}{w_1 + w_2 + w_3}. \quad (5)$$

Here, w_1, w_2, and w_3 are the weights. Given (5), the optimization problem is to find an optimum path U_{opt} that has a minimum f_{obj}, such that

$$U_{\text{opt}} = \arg\min_{U} f_{\text{obj}}(U). \quad (6)$$

III. The Proposed 2-D ACO-based Path Planner

In this section, the basic mechanisms of ordinary ACO algorithms are revisited. Afterward, the proposed ACO-based path planner will be introduced.

A. Background of ACO Algorithms

The first ACO algorithm was proposed by Dorigo in [14], who was inspired by the foraging behaviors in ant colonies. Suppose an ACO algorithm is used to find the shortest path between two arbitrary points in a network. In the first iteration, artificial ants will be released at the starting point and make random move until they reach the target point. Artificial pheromone will be injected to their paths, the concentration will be inversely proportional to the corresponding path lengths. Ants in the following iterations will make routing decisions based on the pheromone concentration on each path left by their precedents. Ordinary ACO algorithms work well for network-based routing problems that a network is given and the number of routing options (i.e. connection degree) at each intersection is finite. An illustration of an ACO algorithm used to find the shortest path between vertices in a network is shown in Fig. 3.

However, the routing network may impose extra constraint to a path planning problem as it limits the possible directions an ant can visit. The ACO-based path planner with Gaussian functions (ACO-Gauss) in [11] tries to relax such limitation with the introduction of the control point lines (CPLs). CPLs are distributed evenly along the x–axis and parallel to the y–axis in the terrain. Instead of moving from one vertex to another in a routing network, an ant is moving from one CPL to another. An ant is allowed to select any point on the next CPL as its next hop. At the end of an iteration, the ants will leave pheromone, which is in form of Gaussian functions,

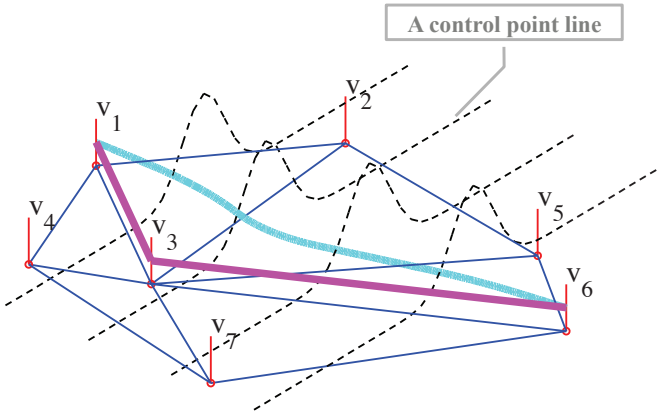

Fig. 3. An illustration of paths between vertices v_1 and v_6. The path obtained using an ordinary ACO-based path planner is shown in magenta color while the path obtained using the ACO-Gauss in [11] is shown in cyan color. The network underneath represents the routing network used in the ordinary version. The dotted lines represent the pheromone distributions on the CPLs in ACO-Gauss.

onto the CPLs. An illustration of the CPLs is shown in Fig. 3. The CPLs in ACO-Gauss provide the ants with more routing options. However, the flexibility of the path planner is still limited by the number and the orientations of the CPLs in the terrain.

B. ACO-based Path Planner with 2-D Gaussian Functions

In the proposed ACO-based Path Planner with 2–Dimensional Gaussian Functions (ACO-2-Gauss), an ant is allowed to select any point on the circumference of a circle on the x–y plane with a radius of R m centered at the current location as its next hop. Such circle is defined as the control point circle (CPC). An illustration of a CPC is shown in Fig. 4. An ant will decide its moving direction based on the distribution of the pheromone concentration on the circumference of the current CPC. An ant will select $n-2$ CPCs from the terrain and move toward P_{t}. The last CPC will hop to P_{t} directly. Including P_{s} and P_{t}, a path will consists of n control points. Once an ant has reach P_{t}, its control points are used to construct a trajectory using B–Spline. The path will be evaluated based on (5). Artificial pheromone will then be distributed at the centers of the CPCs as 2–dimensional (2-D) Gaussian functions. Ants in the following iterations will decide their moving direction based on the pheromone residue in the terrain. Detailed procedures of ACO-2-Gauss are elaborated as follows

Step 1 Given P_{s} and P_{t}, initialize the optimization process by distributing pheromone at P_{t}. Set itera-

tion number $T = 1$.

Step 2 Release q ants at P_{s}. Set counter $\tau_1 = 0$.

Step 3 For each ant, calculate the pheromone concentration at the circumference of the current CPC. Convert the pheromone concentration into a probability distribution function (PDF). Select the next moving direction based on the PDF and move forward by R m. Set $\tau_1 \leftarrow \tau_1 + 1$.

Step 4 If $\tau_1 \geq n - 2$, proceed to Step 5. Otherwise, repeat Step 3.

Step 5 Select P_{t} as the next hop. Evaluate the path using (5). Distribute pheromone at all its CPCs except P_{s} and P_{t}.

Step 6 Release a path construction ant (PCA) at P_{s}. Set counter $\tau_2 = 0$

Step 7 PCAs behave the same as other ants that decide their routing directions based on the PDFs and move forward by R m. Set $\tau_2 \leftarrow \tau_2 + 1$.

Step 8 If $\tau_2 \geq n - 2$, proceed to Step 9. Otherwise, repeat Step 7.

Step 9 Select P_{t} as the next hop. Evaluate the path using (5). If the result is satisfy or if the maximum iteration number $T_{\max}$ has reached, terminate. Otherwise, update pheromone concentration of all Gaussian functions in the terrain. Set $T \rightarrow T + 1$ repeat Step 2.

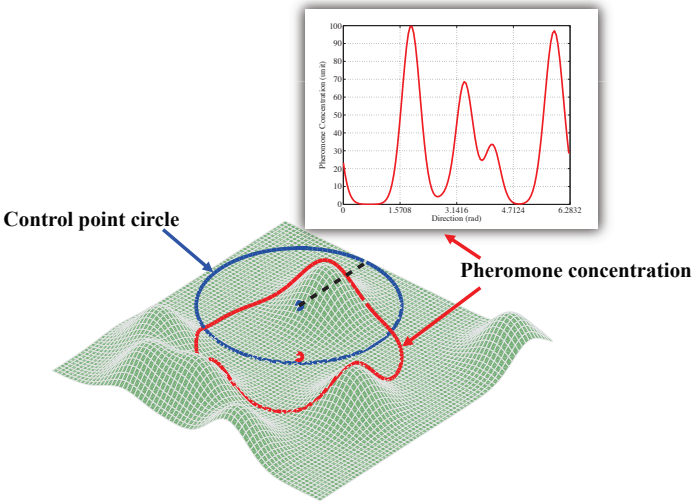

Fig. 4. An illustration of a CPC used in the proposed ACO-2-Gauss. The green surface is representing the pheromone concentration. The blue circle is representing the CPC, while the red curve is represent the corresponding pheromone distribution on the circumference of the CPC. The graph is showing the z–dimension of the red curve from 0 to 2π.

In Step 1, the pheromone distributed at P_{t} is to develop a virtual potential field that attract ants toward the target [15]. Such practice is essential at the initial phase as the terrain has no pheromone residue. In later iterations, the pheromone distributed at P_{t} acts as a biasing factor to favor directions toward P_{t}. The pheromone concentration due to $P_{\text{t}} = (x_{\text{t}}, y_{\text{t}})$ measured at an arbitrary location (x, y) is expressed as the following 2-D Gaussian function

$$g_{\text{t}}(x, y) = \phi_{\text{t}} \exp - \left(\frac{(x - x_{\text{t}})^2}{2\sigma_{\text{t\_}x}^2} + \frac{(y - y_{\text{t}})^2}{2\sigma_{\text{t\_}y}^2} \right). \quad (7)$$

Here, ϕ_{t} is representing the magnitude of the pheromone concentration at P_{t}, which is a tuning parameter. Variables $\sigma_{\text{t\_}x}$ and $\sigma_{\text{t\_}y}$ are the variances of the blob along x-axis and y-axis, respectively. In the proposed path planner, only symmetric 2-D Gaussian functions are considered and thus $\sigma_x = \sigma_y$.

In Step 5, for a path U_i, the initial magnitude of its pheromone is inversely proportional to its $f_{\text{obj}}(U_i)$, such that

$$\phi(U_i) = \frac{\phi_{\text{c}}}{f_{\text{obj}}(U_i)}. \quad (8)$$

Here, ϕ_{c} is a tuning parameter. Suppose U_i consists of m CPCs. The pheromone concentration due to the j^{th} CPC measured at an arbitrary location (x, y) is expressed as the following 2-D Gaussian function

$$g_{i,j}(x, y) = \phi(U_i) \exp - \left(\frac{(x - x_{i,j})^2}{2\sigma_x^2} + \frac{(y - y_{i,j})^2}{2\sigma_y^2} \right). \quad (9)$$

Here, $x_{i,j}$ and $y_{i,j}$ are the coordinates of the j^{th} CPC on the path U_i. Variables σ_x and σ_y are the variances of the blob along x-axis and y-axis, respectively. By the end of an iteration, the magnitudes of the pheromone are updated in Step 9 as follows

$$\phi(U_i) = \frac{\phi(U_i)}{\epsilon}. \quad (10)$$

Here, $\epsilon > 1$ represents the evaporation rate of the pheromone. Steps 3 and 7, the pheromone concentration at the circumference of a CPC is the sum of all the Gaussian functions within the terrain.

IV. Simulations

The performance of the proposed ACO-2-Gauss path planner is evaluated against the ACO-Gauss path planner in [11] using computer simulations. In this section, simulation settings and simulation results are presented.

TABLE I
PARAMETERS USED IN SIMULATIONS

Parameters	ACO-Gauss	ACO-2-Gauss
terrain constants (a, b, c, d, e)	$-0.3, 3, 1.3, 0.5, -2$	$-0.3, 3, 1.3, 0.5, -2$
terrain width	1	1
terrain height	1	1
variances used in (7) ($\sigma^2_{t\_x}$, $\sigma^2_{t\_y}$)	N/A, 0.0005	$0.0125, 0.0025$
variances used in (9) (σ^2_x, σ^2_y)	N/A, 0.0005	$0.125, 0.025$
weighting (w_1)	8	8
weighting (w_2)	4	4
weighting (w_3)	4	4
maximum iteration number ($T_{\max}$)	5	5
no. of ants per iteration (q)	5	5
no. of CPLs in the terrain	5	N/A
no. of CPCs per path ($n-2$)	N/A	5
pheromone evaporating rate (ϵ)	1.75	1.75
order of B-spline curves	4	4

TABLE II
SIMULATIONS RESULTS OF ACO-GAUSS AND ACO-2-GAUSS

Evaluations	ACO-Gauss	ACO-2-Gauss						
		$R = 0.09$	$R = 0.10$	$R = 0.11$	$R = 0.12$	$R = 0.13$	$R = 0.14$	$R = 0.15$
Averaged path length D_1 (unit)	1.4020	1.1220	1.1320	1.1313	1.1411	1.1350	1.1437	1.1519
Averaged turning angle D_2 (degree)	119.84	102.1743	97.8110	89.8771	95.7861	84.7613	92.7532	90.1543
Averaged climbing and descending ratio D_3	1.6297	1.8630	1.7122	1.6281	1.5481	1.5671	1.4928	1.4443

A. Simulation Settings

Simulations are conducted in Matlab. In the simulations, the path planners under test are provided with the terrain introduced in Section II-A. The desirability of a path is evaluated based on the factors discussed in Section II-B. The weight of the each factor is kept equal for both the path planners for a fair comparison. Optimum values for the variances are decided based on the experimental results. Variables and tuning parameters used in the simulations are shown in Table I.

B. Simulation Results

Based upon the parameters given in Table I, both ACO-based path planners are tested. Coordinates of the starting point P_{s} and the target point P_{t} are located at $(0.0208, 0.3000)$ and $(0.9792, 0.8000)$, respectively. In each simulation, the path planners are allowed to operate for 5 iterations. In each iteration, both ACO-based path planners release 5 artificial ants into the terrain. An ant picks 5 locations inside the terrain and uses them as control points to construct their B–Spline curves. In ACO-Gauss, artificial ants can only select locations on the CPCs as their control points. Simulations for ACO-2-Gauss were carried out with 7 different R values of the CPCs. In this paper, all the results are the averaged values obtained from 50 individual simulations. Simulation results are shown in Table II.

V. PERFORMANCE ANALYSIS

According to the simulation results, the proposed ACO-2-Gauss path planner shows greatly improved performance over the original ACO-Gauss path planner, in terms of path length, turning angle, and climbing and descending ratio for all different R values under test. According to the simulation results for ACO-2-Gauss, the average path length has increased with R. A good balance among all three evaluation factors can be found when $R = 0.12$. When $R = 0.12$, the proposed ACO-2-Gauss outperforms the original ACO-Gauss by reducing the path length by more than 18%, the turning angle by more than 20%, and the climbing and descending ratio by nearly 5%. However, due to the extra degree of freedom, the computational complexity of the ACO-2-Gauss is slightly higher than that of the ACO-Gauss. Nevertheless, the computational complexity

can be neglected as both the path planners are utilized in off-line path planning.

VI. Conclusions

Mobile wireless platforms can be used for reconnecting several fragments of a wireless sensor network by setting up bridges between them. In order to bridge fragments and shorten the down-time of a network, mobile robots needed to reach the target location quickly. The proposed ACO-2-Gauss can be used for off-line path planning of mobile wireless platforms. Comparing with its preceding version ACO-Gauss, the current version can considerably improve path quality due to the extra freedom provided by the 2-D Gaussian functions. In the proposed path planner, artificial ants have higher flexibility in making routing decisions. The simulation results verify the improved performance of the proposed ACO-2-Gauss path planner over the original ACO-Gauss path planner.

References

[1] T. Suzuki, K. Kawabata, Y. Hada, and Y. Tobe, "Deployment of wireless sensor network using mobile robots to construct an intelligent environment in a multi-robot sensor network," in *Advances in Service Robotics*, H. S. Ahn, Ed. InTech, 2008, ch. 17, pp. 315–328.

[2] G. Tuna, T. Mumcu, K. Gulez, V. Gungor, and H. Erturk, "Unmanned aerial vehicle-aided wireless sensor network deployment system for post-disaster monitoring," in *Emerging Intelligent Computing Technology and Applications*, ser. Communications in Computer and Information Science, D.-S. Huang, P. Gupta, X. Zhang, and P. Premaratne, Eds. Springer Berlin Heidelberg, 2012, vol. 304, pp. 298–305.

[3] N. Atay and B. Bayazit, "Mobile wireless sensor network connectivity repair with k-redundancy," in *Algorithmic Foundation of Robotics VIII*, ser. Springer Tracts in Advanced Robotics, G. Chirikjian, H. Choset, M. Morales, and T. Murphey, Eds. Springer Berlin Heidelberg, 2009, vol. 57, pp. 35–49.

[4] X. Li, R. Falcon, A. Nayak, and I. Stojmenovic, "Servicing wireless sensor networks by mobile robots," *IEEE Communications Magazine*, vol. 50, no. 7, pp. 147–154, 2012.

[5] A. Erman, L. Hoesel, P. Havinga, and J. Wu, "Enabling mobility in heterogeneous wireless sensor networks cooperating with uavs for mission-critical management," *IEEE Wireless Communications*, vol. 15, no. 6, pp. 38–46, 2008.

[6] D. J. Klein, S. Venkateswaran, J. T. Isaacs, J. Burman, T. Pham, J. a. Hespanha, and U. Madhow, "Localization with sparse acoustic sensor network using uavs as information-seeking data mules," *ACM Transactions on Sensor Networks*, vol. 9, no. 3, pp. 30:1–30:29, Jun. 2013. [Online]. Available: http://doi.acm.org/http://dx.doi.org/10.1145/2480730.2480733

[7] W.-G. Han, S.-M. Baek, and T.-Y. Kuc, "Ga based online path planning of mobile robots playing soccer games," in *40th Midwest Symposium on Circuits and Systems, 1997*, vol. 1, Sacramento, California, USA, August 1997, pp. 522–525.

[8] D. Henrich, C. Wurll, and H. Worn, "Online path planning with optimal c-space discretization," in *1998 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, Victoria, British Columbia. Canada, October 1998, pp. 1479–1484.

[9] S. Mittal and K. Deb, "Three-dimensional offline path planning for uavs using multiobjective evolutionary algorithms," in *IEEE Congress on Evolutionary Computation, 2007, (CEC 2007)*, Singapore, September 2007, pp. 3195–3202.

[10] C.-T. Cheng, K. Fallahi, H. Leung, and C. Tse, "A genetic algorithm-inspired uuv path planner based on dynamic programming," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 42, no. 6, pp. 1128–1134, 2012.

[11] ——, "Cooperative path planner for uavs using aco algorithm with gaussian distribution functions," in *IEEE International Symposium on Circuits and Systems 2009, (ISCAS 2009)*, Taipei, Taiwan, May 2009, pp. 173–176.

[12] H. Späth, *One dimensional spline interpolation algorithms*, ser. Ak Peters Series. A K Peters, 1995.

[13] L. Piegl and W. Tiller, *The Nurbs Book*, ser. Monographs in Visual Communications. Springer-Verlag GmbH, 1997.

[14] M. Dorigo, "Optimization, learning and natural algorithms," Ph.D. dissertation, 1992.

[15] F. Ding, P. Jiao, X. Bian, and H. Wang, "Auv local path planning based on virtual potential field," in *2005 IEEE International Conference on Mechatronics and Automation, (ICMA 2005)*, vol. 4, Niagara Falls, Ontario, Canada, July 2005, pp. 1711–1716.