An ACO-based Off-line Path Planner for Nonholonomic Mobile Robots

Nuwan Ganganath, Chi-Tsun Cheng, and Chi K. Tse
Department of Electronic and Information Engineering,
The Hong Kong Polytechnic University,
Hung Hom, Kowloon, Hong Kong
Email: nuwan.marasinghearachchige@connect.polyu.hk

***Abstract*—The path planning is an important issue as it allows a robot to get from a point to another. Such a path between two points has to be optimized based on user defined requirements and environmental conditions. Most of the existing solutions to path planning problem assume robots to be holonomic. However, ordinary mobile robots are kinematically constrained in practice. A novel solution has been proposed for the path planning problem based on existing ant colony optimization methods which can be realized with practical mobile robots with the aforementioned constraints. Simulation results show the applicability of the proposed path planner to the non-holonomic mobile robots. Performance of the proposed algorithm has been compared with its preceding version. The performance of the proposed path planner may be further improved by fine-tuning its parameters.**

***Index Terms*—ACO, path planning, field-of-view, non-holonomic, mobile robots**

I. Introduction

Path planning is one of the most discussed problems in mobile robotics. It can be defined as an optimization process in which the path between two points needs to be decided such that predefined requirements are satisfied [1]. In most of the previous solutions to path planning problem, these requirements are limited to path length only [2], [3]. However, in most of the real world scenarios, the shortest path between two points may not be the most desirable path. As an example, consider a mobile robot operating in a military battle field. Even though it is supposed to reach the soldiers with minimum delay when support is needed, it also has to avoid being captured by the enemies. On the other hand, the shortest path always does not guarantee the minimum delay. It also depends on the some other characteristics of the trajectory, such as climbing and descending ratios and turning angles [4].

In [5], Ganganath and Cheng proposed an off-line path planner for mobile robots based on an ant colony optimization (ACO) algorithm (ACO-2-Gauss). Unlike its preceding version (ACO-Gauss) [4], where artificial ants only make routing decisions at limited number of parallel lines which are evenly distributed on a given terrain, the ACO-2-Gauss path planner allows ants to select their control points anywhere in the terrain. Comparing to the ACO-Gauss path planner, ACO-2-Gauss provides artificial ants with more flexibility to make their routing decisions. Two-dimensional (2-D) Gaussian functions used in pheromone distribution considerably improve the path quality due to the extra degree of freedom. However, ACO-2-Gauss assumes mobile robots to be holonomic. In practice, most of the robots are kinematically constrained [6], [7]. This paper is an extension to the ACO-2-Gauss path planner proposed in [5]. The aim of this work is to generalize the ACO-2-Gauss by introducing a field-of-view (FoV) concept for artificial ants so that the proposed algorithm can be realized with most real-world robots, which are kinematically constrained.

II. Problem Formulation

A. Scenario Under Study

The scenario under study is a meshed 3-D model mimicking a hilly landscape. The base of the 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,

$$z(x, y) = \sigma[a\sin(y) + b\cos(x) + c\sin(d\sqrt{x^2 + y^2})]^2, \quad (1)$$

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

B. Desirability of a Path

Similar to its preceding versions, the purpose of the ACO-2-Gauss path planner is to find the optimum trajectory from an arbitrary starting point P_s to an arbitrary target point P_t within the given terrain such that it satisfy the predefined requirements. In this work, desirability of a path is calculated based on three such requirements (desirability factors), namely, path length, turning angle, and climbing and descending ratio.

Since the length of a path is proportional to the wear and tear and the energy consumption of a robot, it is always desirable to have shorter paths. Therefore, the first desirability factor (ρ_1) is defined as,

$$\rho_1(\eta) = d_{\text{proj}}(P_{\text{s}}, P_{\text{t}}), \tag{2}$$

where $d_{\text{proj}}(P_{\text{s}}, P_{\text{t}})$ is the length of the projection of the path η on the x–y plane which is connecting P_{s} and P_{t}. In this paper, a path is represented using a B–Spline curve, which is first introduced by Schoemberg [8]. An illustration of a B–Spline curve with $n = 6$ control points $(c_1, \cdots, c_6)$ is shown in Fig. 1.

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. A path with sharp turning angles is undesirable, especially because they are difficult to achieve with nonholonomic robots. Therefore, given a path η with n control points $(c_1, \cdots, c_n)$, its second desirability factor (ρ_2) is defined as,

$$\rho_2(\eta) = 180^\circ - \min_{i=2,\cdots n-1} \angle c_{i-1}c_i c_{i+1}. \tag{3}$$

Paths with frequent climbing and descending are undesirable to mobile robots as they can result in more energy consumption. Also it is difficult to maneuver robots on steep hills/slops. Therefore, the robot trajectory should be planned in a way that the climbing and descending ratio is low. After the control points of a B–Spline curve are decided, a fixed number of evaluation points are used to represent the piecewise polynomial curves between adjacent control points. Given a path η with k evaluation points $(e_1, \cdots, e_k)$, its third desirability factor (ρ_3) is defined as,

$$\rho_3(\eta) = \max_{i=1\cdots n-1} \left| \frac{\Delta z_i}{\Delta d_i} \right|. \tag{4}$$

Δz_i and Δd_i are illustrated in Fig. 1.

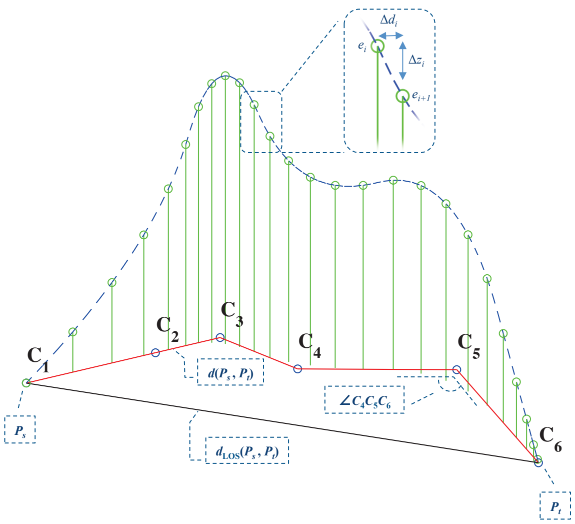

Fig. 1. An illustration of a trajectory between P_{s} and P_{t}. The dashed line in blue is representing the path. The solid line in red is showing the interpolation of the control points.

In order to obtain a desirable trajectory, we need to minimize all of the above mentioned factors. Therefore, the cost function δ used in this paper is defined as,

$$\delta(\eta) = \frac{\sum_{i=1}^{3} w_i \rho_i(\eta)}{\sum_{i=1}^{3} w_i}, \tag{5}$$

where w_1, w_2, and w_3 are the weights. Here, the optimization problem is to find an optimum path η_{opt} which satisfy

$$\eta_{\text{opt}} = \arg\min_{\eta} \delta(\eta). \tag{6}$$

III. The Proposed Path Planner for Nonholonomic Mobile Robots

A. Background of ACO Algorithms

ACO algorithms are inspired by the pheromone communication between ants regarding a desired path between their colony and a food source [9]. Initially ants travel randomly in their environment. Once an ant find a food source, it starts to release pheromone on their path between the food source and the colony. If several trips are performed on the same route, the pheromone concentration on that route increases accordingly. Other ants may follow the same path by sensing previous pheromone and they release more pheromone onto the path. Since pheromone decays with time, older paths are likely to disappear from the terrain [10].

Ordinary ACO algorithms performs well in network based routing problems which has finite number of routing options at each intersection of a given network [4]. However, these routing networks limits the routing options of the ants. In ACO-Gauss [4], these constrained are released to a certain extent by introducing Gaussian functions which lies along so called control point lines (CPLs). CPLs are distributed evenly along the x–axis and parallel to the y–axis of 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, onto the CPLs.

In ACO-2-Gauss [5], the flexibility of the routing decisions are further enhanced by introducing 2-D Gaussian functions. Here, ants are allowed to select any point on the circumference of a circle on the x–y plane with a radius of r centered at the current location as its next

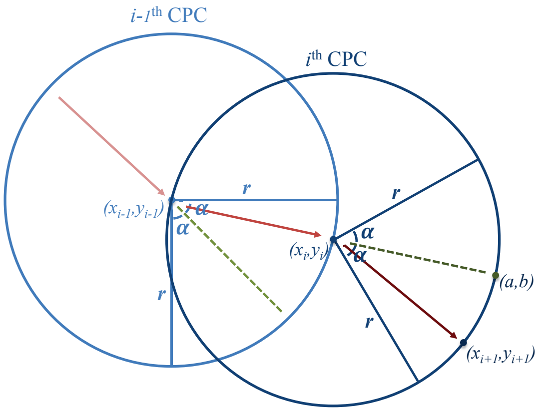

Fig. 2. An illustration of CPCs in two consecrative time steps. The FoV of an ants is considered as 2α. r is the radius of CPCs.

hop. Such a circle is defined as the control point circle (CPC). 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-D Gaussian functions. Ants in the following iterations will decide their moving direction based on the pheromone residue in the terrain. After several iterations, it releases a path construction ant (PCA) to generate the final path.

B. Generalized ACO-2-Gauss Path Planner

In ACO-2-Gauss, ants are allowed to select any point on the circumference of current CPC based on the pheromone distribution. However, the paths generated using such an algorithm might not be successfully realized with nonholonomic robots. Therefore, we introduce an FoV concept to all the ants, including the path construction ants so that we can obtain realizable paths using ACO-2-Gauss algorithm.

Let (x_i, y_i) is the current location of any arbitrary ant and (x_{i-1}, y_{i-1}) is its location in previous time step. Our aim is to calculate the next location, (x_{i+1}, y_{i+1}), so that it lies in the FoV (2α) of the given ant (see Fig. 2). Since (x_i, y_i) is at the center point of the line which connects (x_{i-1}, y_{i-1}) and (a, b),

$$x_{i-1} + a = 2x_i \text{ and } y_{i-1} + b = 2y_i. \tag{7}$$

Since the next hop should be located within the given FoV,

$$\sqrt{(x_{i+1} - a)^2 + (y_{i+1} - b)^2} \leq 2r\sin(\frac{\alpha}{2}). \tag{8}$$

By using (7) and (8), we can obtain the final condition, which is needed to be satisfied by the coordinates of the next hop.

Initially, (x_0, y_0) are set to the coordinates of P_s. As we always position P_s somewhere closer to y-axis in this study, we assume $x_{-1} = x_0 - r$ and $y_{-1} = y_0$ in order to calculate (x_1, y_1) such that calculated location using (7) and (8) always lies inside of the given terrain. As an ant further moves in the terrain, it might reach closer to a border of the terrain and it might not be able to find its next location inside its FoV. In such situations, we reset its FoV to 360^o for a single time step, which can result in greater turning angles than expected. Detailed procedures of the generalized ACO-2-Gauss are elaborated as follows:

- Step 1 Given P_s and P_t, initialize the optimization process by distributing pheromone [5] at P_t. Set iteration number $T = 1$.
- Step 2 Release q ants at P_s. Set counter $i = 0$. Set (x_0, y_0) to the coordinates of P_s. Take $x_{-1} = x_0 - r$ and $y_{-1} = y_0$.
- Step 3 For each ant, check weather the circumference of the current CPC within their FoV lies within the boundaries of the terrain. If it is, jump to Step 5. Otherwise, continue to Step 4.
- Step 4 Reset its FoV to 360^o for a single time step. Repeat to Step 3.
- Step 5 Calculate the pheromone concentration at the circumference of the current CPC within their FoV. Convert the pheromone concentration into a probability distribution function (PDF). Select the next moving direction based on the PDF and move forward by r. Set $i \leftarrow i + 1$.
- Step 6 If $i \geq n-2$, proceed to Step 7. Otherwise, repeat Step 3.
- Step 7 Select P_t as the next hop. Evaluate the path using (5). Distribute pheromone [5] at all its CPCs except P_s and P_t.
- Step 8 Release a PCA at P_s. Set counter $i = 0$
- Step 9 Unlike other ants, PCAs decide their routing directions based on the global peak of the PDFs and move forward by r. Set $i \leftarrow i + 1$.
- Step 10 If $i \geq n - 2$, proceed to Step 11. Otherwise, repeat Step 9.
- Step 11 Select P_t as the next hop. Evaluate the path using (5). If the result is satisfactory or if the maximum iteration number $T_{\max}$ has been reached, terminate. Otherwise, update pheromone concentration of all Gaussian functions in the terrain [5]. Set $T \rightarrow T + 1$ repeat Step 2.

TABLE II
SIMULATIONS RESULTS OF THE PROPOSED PATH PLANNER FOR DIFFERENT FOV VALUES

FoV	30^0	60^0	90^0	120^0	150^0	180^0	210^0	240^0	270^0	300^0	330^0	360^0
$\rho_1(\eta_{opt})$	1.2942	1.2668	1.2491	1.2435	1.2414	1.2784	1.3696	1.4378	1.4338	1.3726	1.3190	1.2948
$\rho_2(\eta_{opt})$	64.07^0	51.63^0	54.96^0	59.51^0	78.59^0	87.47^0	101.89^0	115.18^0	140.20^0	154.31^0	167.89^0	179.98^0
$\rho_3(\eta_{opt})$	1.6717	1.7371	1.6724	1.5653	1.4705	1.5177	1.6473	2.0178	2.2245	2.2850	2.3833	2.4193
$\delta(\eta_{opt})$	17.08	13.98	14.78	15.89	20.64	22.89	26.57	30.02	36.32	39.84	43.23	46.25

TABLE I
SIMULATION PARAMETERS

Parameters	Values
terrain constants (a, b, c, d)	$0.5, 1.5, -0.5, 2.5$
weighting (w_1)	8
weighting (w_2)	4
weighting (w_3)	4
maximum iteration number $(T_{\max})$	5
no. of ants per iteration (q)	5
no. of CPCs per path $(n-2)$	8
pheromone evaporating rate	1.75
order of B-spline curves	4

IV. SIMULATION RESULTS

The proposed path planner has been implemented and simulated using Matlab. It is provided with the terrain introduced in Section II-A. The coordinates of P_s and P_t are set to $(0.01, 0.45)$ and $(0.99, 0.95)$, respectively. The desirability of a path is evaluated based on the factors discussed in Section II-B. The weight of the each desirability factor is kept equal during all the experiments for a fair comparison. Variables and tuning parameters used in the simulations are shown in Table I. The proposed algorithm is tested for different FoV values ranging from 30^o to 360^o. Simulation results are shown in Table II. When, FoV = 360^o, the path planner under test is equivalent to its preceding version, ACO-2-Gauss.

According to the simulation results, the proposed path planner shows improved performance over ACO-2-Gauss. Having FoV set close to 360^o gives unnecessary freedom to the ants and allow them to search in loops, which results in longer path lengths and larger turning angles. Therefore, it can be observed that the value of $\delta(\eta_{opt})$ increases with the value of FoV $\geq 60^0$. In contrast, having FoV set too narrow (FoV $< 60^0$) limits the search space of the ants. Especially, when ants are close to the boundary of the terrain, they might fail to find the next hop in the first scan and leads to greater turning angles. Interestingly, according to the simulation results, we can conclude that the proposed path planner is able to produce better performances even when FoV $< 180^o$, which matches closely with the properties of ordinary nonholonomic mobile robot robots.

V. CONCLUSIONS

In this paper, a path planner for nonholonomic mobile robots based on an ant colony optimization method is proposed. In the proposed path planner, an artificial ant in the optimization process is only provided with a limited FoV. Such extra constraint on FoV can effectively avoid ants from going into loops during the search process. Simulation results show that path planners with extreme FoV values may, however, yield poor results. With an appropriate selection of FoV value, the proposed path planner shows significant improvements in the computer simulations over the other path planner under test. However, the performance of the proposed path planner needs to be further verified with the nonholonomic mobile robots in real world scenarios.

REFERENCES

[1] C.-T. Cheng, K. Fallahi, H. Leung, and C. K. Tse, "An AUVs Path Planner using Genetic Algorithms with a Deterministic Crossover Operator," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2010, pp. 2995–3000.

[2] E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[3] P. Salaris, D. Fontanelli, L. Pallottino, and A. Bicchi, "Shortest Paths for a Robot with Nonholonomic and Field-of-view Constraints," *IEEE Trans. on Robotics*, vol. 26, no. 2, pp. 269–281, 2010.

[4] C.-T. Cheng, K. Fallahi, H. Leung, and C. K. Tse, "Cooperative Path Planner for UAVs using ACO Algorithm with Gaussian Distribution Functions," in *Proc. IEEE Int. Symp. on Circuits and Systems (ISCAS)*. IEEE, 2009, pp. 173–176.

[5] N. Ganganath and C.-T. Cheng, "A 2-Dimensional ACO-based Path Planner for Off-line Robot Path Planning," in *Proc. IEEE Int. Workshop on Cyber-enabled Distributed Computing and Knowledge Discovery (CyberC)*. IEEE, Oct. 2013, pp. 302–307.

[6] N. Ganganath and H. Leung, "Mobile Robot Localization Using Odometry and Kinect Sensor," in *Proc. IEEE Int. Conf. on Emerging Signal Processing Applications (ESPA)*. IEEE, Jan. 2012, pp. 91–94.

[7] N. Ganganath, "Goal-driven Robot Navigation in Unknown Indoor Environments," Master's thesis, The University of Calgary, Calgary, Canada, Jan. 2013.

[8] H. Späth, *One Dimensional Spline Interpolation Algorithms*, ser. Ak Peters Series. A K Peters, 1995.

[9] M. Dorigo, "Optimization, Learning and Natural Algorithms," Ph.D. dissertation, Politecnico di Milano, Italy, 1992.

[10] J. Brownlee, *Clever Algorithms: Nature-inspired Programming Recipes*. Jason Brownlee, 2011.