

Solving Path Planning Problem by an ACO-PSO Hybrid Algorithm

Chunxue Shi¹ Yingyong Bu¹ Ziguang Li² Jun Tan¹

¹School of Mechanical and Electrical Engineering, Center South University, Changsha 410083, China

²School of Mechanical and Electrical Engineering, Changsha Science & Technology University, Changsha 410076, China

Abstract

An ACO-PSO Hybrid Algorithm Solving Path Planning Problem based on Swarm Intelligence (SI) is proposed. The problem first is described and some corresponding definitions are presented. Ant colony optimization (ACO) is used to establish the corresponding solution, and some material algorithm steps are set out. Particle swarm optimization (PSO) is applied to optimize the parameters in ACO, and parameters can be selected self- adaptively. Through the steps above, algorithm performance can be greatly enhanced and the reliability of parameters selecting can be guaranteed. At last simulation experiments are carried out, and the results prove the veracity of the algorithm.

Keywords: Ant colony optimization (ACO), Particle swarm optimization (PSO), Hybrid algorithm, Robot path planning problem

1. Introduction

The concept of Swarm Intelligence (SI) [1] was first proposed by the Beni, Hackwood and Wang in Self-organization of Sensors. Here, Swarm can be portrayed as a set of some interactive neighboring individuals, in which a single individual capacity is limited, but multi- individuals as Swarm have the extremely strong survivability. That may regard as Swarm Intelligence. Ant colony optimization (ACO) [2]-[4] is a intelligent optimization algorithm which is subject to the inspiration of ants' behavior of seeking for food, and is proposed by Italian scholar Dorigo. Moving ants can release a kind of pheromone in the passing path, and between the two locations which ants have passed by, more shorter path concentrate more stronger pheromone. Because the shorter path has higher probability to be selected, the pheromone concentration has been constantly strengthened. Other ants can sense pheromone concentration, and thus guide their selection of path: The path with high

pheromone concentration has greater probability to be chosen. ACO is widely applied in Traveling Salesman Problem (TSP) [5], Network Routing Problem(NRP) [6], Job-shop Scheduling Problem(JSP) [7], Quadratic Assignment Problem(QAP) [8], Graph Coloring Problem(GCP) [9], Vehicle Routing Problem(VRP) [10], and gains satisfying effect. However, the algorithm is a typical probability algorithm, the algorithm parameters are usually given by experiments, that makes the performance of the algorithm relate to individual experience, difficult achieve optimization. Particle Swarm Optimization (PSO) [11] is proposed most early by psychology researcher Dr. Kennedy and the computation intelligence Researcher Dr. Eberhart in 1995 and is originated from the simulation flock movement. The system is initialize to be a series of stochastic solutions, and search optimal value through iterations. particles in the solution space carry out search by following the optimal particle. PSO is quite fast speed to approach optimal solutions, so that it can optimize the parameter of ACO [12]. Therefore ACO and PSO can combine to a Hybrid Algorithm to solve the robot path planning problem.

2. Problem descriptions

The robot path planning problem is that in two-dimensional limited space with obstacles, finding a integrity path which can make the robot move from the start point S to the goal point G avoiding collisions.

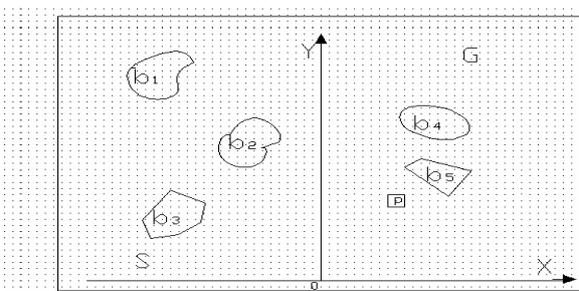


Fig.1: Sketch map of Environment model.

The figure above is the sketch map of environment model. The method of environment modeling is Bitmap method and a number of pixels are in uniform distribution. In order to standardize the description of the problem, now the following definitions are listed:

Definition 1 It is supposed that PS is two-dimensional limited space for robots, and p_i is one of the pixels in the space, and P is the set of all pixels, thus $\forall p$, has corresponding coordinates in Σ_0 , such as: $p_i(x_i, y_i)$; $S = \{s_1, s_2, \dots, s_k, \dots, s_M\}$ is the set of all of the obstacles, and $s_k \in S$ is one of the obstacles in S , then M is amount of the obstacles; $O = \{o_1, o_2, \dots, o_k, \dots, o_N\}$ is the set of pixels which are covered by obstacles, and $o_k \in O$ is one of the pixels in O , then N is amount of the covered pixels.

Definition 2 The line segment between two pixels ($p_i, p_j \in P, i \neq j$) become one edge in PS , which is called L_{ij} , and $d(p_i, p_j)$, as the length of L_{ij} , is

$$d(p_i, p_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (1)$$

Definition 3 $\tau_{ij}(t)$ is the remained pheromone amount on L_{ij} at t time; $agroup = \{1, 2, \dots, k, \dots, m\}$ is the set of all ants in ant group $agroup$, and $k \in agroup$ is one of the ant in $agroup$, and m is the amount of ants.

Definition 4 If $\forall p \in P, p \notin O$, p is called "transit pixel", and the set of p is called "transit space" W ; The pixels in O is covered by obstacles, where ants can't pass, so that O is called "no transit space"; $Bp_i = \{p | p \in P, d(p, p_i) \leq \sqrt{2}R_h\}$ is called neighbor space of ant k at p_i (as is shown in Fig.1); $tabu_k = \{p(0), p(1), \dots, p(t)\}$ is the set of pixels where ant k has passed from o time to t time, and $tabu_k$ is called taboo list as it can't be passed twice; $Wp_i = \{p | p \in Bp_i, p \notin tabu_k, p \notin O\}$, is called "transit space" for ant k at p_i , in the other words, when ant k comes to p_i , the pixels in this set can be chosen next step.

Algorithm process is that m ants in $agroup$ was set on start point S , then each ant chose next pixel which could be passed by the strategy of nearest neighbor choice, and avoided obstacles, at last, moved close to aim point G .

3. The solution process by ACO

There are three core steps of ACO: walk rules, local updating and overall updating. Focus on these three steps, material algorithm is described as follows:

Step 1: Initialization: The start point S was added to $tabu_k$ ($k = 1, 2, \dots, m$), and m ants were placed on S ; With setting $\tau_{ij}(0) = \tau_0$ (τ_0 is a constant), 0

was set to the iteration counter (NC) for seeking food, and the iteration maximum was $Ncmax$.

Step 2: Walk rules: Each ant seek next pixel by the following rules, then walked between two station points:

$$p_j = \begin{cases} \arg \max[\tau_{ij}^\alpha(t) \eta_j^\beta(t)] & \text{if } (q \leq q_0) \\ X & \text{else} \end{cases} \quad (2)$$

$$p_{ij}^k(t) = \frac{\tau_{ij}^\alpha(t) \eta_j^\beta(t)}{\sum_{p_r \in Wp_i} \tau_{ij}^\alpha(t) \eta_j^\beta(t)} \quad (3)$$

As to the two formulas above, both have the same condition - $p_j \in Wp_i$.

$$\text{In the two formulas: } \eta_j(t) = \frac{E}{d(p_j, G)} \quad (4)$$

X is a stochastic variable, which correspond with a certain pixel; q is a stochastic numeral from 0 to 1; q_0 is a critical value set through initialization; $\eta_j(t)$ is a elicitation function rest with formula (6); E 、 α 、 β respectively are right constant; $p_{ij}^k(t)$ represent the probability for ant k passing from p_i to p_j at t time.

$\forall k$ at current pixel p_i , moved to the next pixel by principle of nearest neighbor choice.

Step 3: Local updating: After each ant had passed by an edge, pheromone updated according to formula (5).

$$\tau_{ij}(t+1) = (1 - \rho_1) \tau_{ij}(t) + \rho_1 \Delta \tau_{ij}^k \quad (5)$$

$$\Delta \tau_{ij}^k = \begin{cases} \frac{Q_1}{\sum_e^u d_e} & \text{if } : L_{ij} \text{ has been passed by ant } k \\ 0 & \text{else} \end{cases}$$

In the formula above, Q_1 is a constant; ρ_1 is local pheromone volatilization coefficient; d_e is the length of the edge ant k has passed by; u is the amount of edges which ant k has passed by in same iteration.

When $\tau_{ij}(t+1) < \tau_{\min}$, set $\tau_{ij}(t+1) = \tau_{\min}$ (τ_{\min} is a constant).

Step 4: Iteration process: Ant group proceeded to search next pixel. when ant k found aim point ($g_j = G$), a integrity path L_k was gained. The length of the integrity path ($d_k = \sum_e d_e^k$) was calculated and. When each ant in $agroup$ reached G , each integrity path of each ant was compared, and the length of the shortest path (L_{\min}) was noted down. which was named $d_{1\min}$.

Step 5: Overall updating: When a iteration finished, the pheromone on $L_{1\min}$ needed to update in speeding up the convergence:

$$\tau_{ij}^{Nc+1} = (1 - \rho_2)\tau_{ij}^{Nc} + \rho_2\Delta\tau_{ij}^{Nc} \quad (6)$$

$$\Delta\tau_{ij}^{Nc} = \begin{cases} \frac{Q_2}{d_{Nc\min}} & \text{if } L_{ij} \text{ is on } L_{Nc\min} \\ 0 & \text{else} \end{cases}$$

In the formula above, Q_2 is a constant; ρ_2 is overall pheromone volatilization coefficient; τ_{ij}^{Nc} is the pheromone amount on L_{ij} after NO. Nc iteration finished; $L_{Nc\min}$ is the best path during NO. Nc iteration; $d_{Nc\min}$ is the length of $L_{Nc\min}$.

Step 6: Iteration complete: After a iteration finished, 1 was added to Nc till the max iteration number. The best path length of each iteration should be compared each others, then noted down and exported the shortest path (L_{\min}) length among them.

4. Parameters training by PSO

In solving Path Planning Problem by ACO, the parameter α and β is determined through experiments and have great impact to the performance of the algorithm. α indicates the importance of the remained pheromone on each pixel, and β indicates the importance of heuristic information. The effect of α and β given to the performance of the algorithm is mutual supported and closely related. Choosing parameters through experiments only depends on experience or repeated debugging, in this paper choosing parameters is by the guidance of PSO. Information sharing mechanism is used in PSO, which has the excellence of its simpleness to achieve and profound intelligent background. In training the ACO parameters α and β by PSO, the particles are expressed as a two-dimensional real-coded strings. The training steps is as following: It is supposed that n particles compose of a community, in which the $NO.i$ particles is expressed as a two-dimensional vector $X_i=(x_{i1},x_{i2})$, $i=1,2,\dots,n$, in the other words, the $NO.i$ particle in the search space is located at x_i . Therefore the position of each particle is a potential solution. The adaptive value can be calculated by adding x_i to ACO through objective function and the solutions are measured by the magnitude of objective value. The flying velocity of the $NO.i$ particle is also a two-dimensional vector, recorded as $V_i = (v_{i1}, v_{i2})$. The optimal location for the $NO.i$ particle by that time is $P_i = (p_{i1}, p_{i2})$, and the optimal location for particles

swarm is $P_g = (p_{g1}, p_{g2})$. Operating the particles by the following formulas:

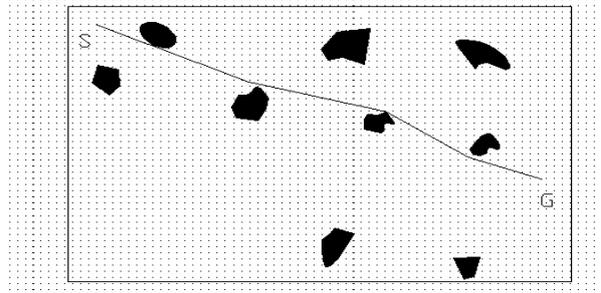
$$V_i(k+1) = wV_i(k) + c_1r_1(P_i - x_i(k)) + c_2r_2(P_g - x_i(k)) \quad (7)$$

$$x_i(k+1) = x_i(k) + V_i(k+1) \quad (8)$$

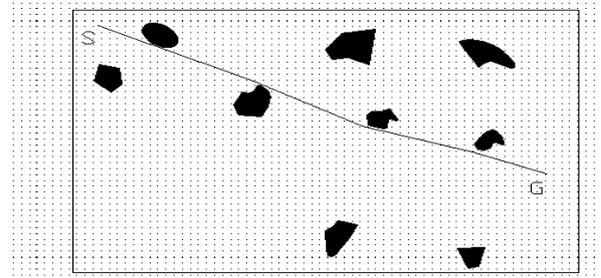
In the formulas above, w is the inertia weight, and is non-negative, and linearly varies from 0.9 to 0.4, that can slow down the movement of the particles, and prevent the particles move closer to g_{Best} during oscillation. c_1 and c_2 separately is weight p_{id} of individual optimal location p_{best} and weight p_{gd} of swarm optimal location g_{best} , commonly $c_1=c_2=2$. r_1 and r_2 is random number between 0 to 1. The flying velocity vector weight of iterative particle i $v_{id} \in [-v_{max}, v_{max}]$, and v_{max} is a constant. Iteration termination condition is ten largest iteration number 10 or the optimal location can meet predetermined minimum adaptation threshold value.

5. Simulation research

Based on the above analysis, mathematical simulation software MATLAB was used to carry on experiment simulation study: setting $m=20$, $Ncmax=50$, $\rho_1 = 0.75$, $\rho_2 = 0.55$, $c_1=1.8$, $c_2=0.9$.



(a) The results of 10 Iterations later



(b) The optimum result

Fig.2: Results of first experiment

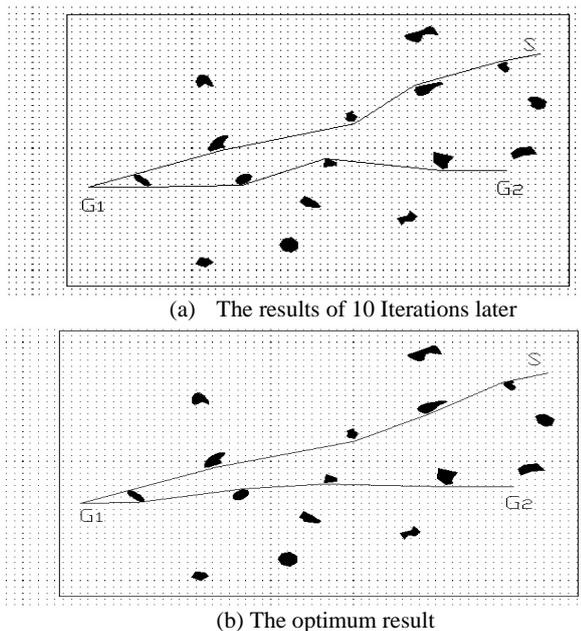


Fig.3: Results of second experiment.

Fig.2 and Fig.3 respectively represent the results of two items of experiments. In Fig.3, G_1 and G_2 are two planning aim point, at the same time, G_1 is also the planning start point for G_2 .

Simulation results indicate that the method used in this study can reach satisfying precision under the price of a certain period of time.

6. Conclusions

In this paper, an ACO-PSO Hybrid Algorithm Solving Path Planning Problem based on Swarm Intelligence(SI) is proposed. ACO as its positive feedback, parallelism and easy combining with other algorithms, is widely applied on lots of filed; Using PSO is because that ACO greatly need experience or repeated experiments to choose corresponding parameters. experience always occupies in a leading role. Thus the parameters in ACO can be optimized through PSO and parameters can be chosen self-adaptively, so that the optimal performance of ACO can be effectively enhanced. By simulating experiments show that the algorithm is correct.

Acknowledgement

This work is partially supported by National Nature Science Foundation of China (Grant No. 50474052).

References

- [1] S. Hackwood and G. Beni, Self-organization of sensors for swarm intelligence. *IEEE International Conference on Robotics and Automation*, pp.819-829, 1992.
- [2] M. Dorigo and L.M. Gambardella, A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, pp.53-66, 1997.
- [3] A. Colomi, M. Dirigo and V. Maniezzo, Distributed optimization by ant colonies. *Proc of the ECAL-91*, pp.134-142, 1991.
- [4] M. Dorigo, V. Maniezzo and A. Colomi, The ant system: optimization by a colony of cooperating agents. *IEEE Trans on Systems, Man & Cybernetics-B*, pp.29-41, 1996.
- [5] B. Wu and Z.Z. Shi, An ant colony algorithm based partition algorithm for TSP. *Chinese J Computers*, pp. 1328-1333, 2001.
- [6] D. Subramanian, P. Druschel and J. Chen, Ants and reinforcement learning: a case study in routing in dynamic networks. *International Joint Conference on Artificial Intelligence*, pp.832-838, 1997.
- [7] A. Colomi, M. Dirigo, V. Maniezzo and M. Trubian, Ant system for job-shop scheduling. *Belgian Journal of Operations Research, Statistics and Computer Science(JORBEL)*, pp.39-53, 1994.
- [8] L.M. Gambardella, E.d. Taillard and M.Dorigo, Ant colonies for the QAP. *Journal of the Operational Research Society(JORS)*, pp.53-66, 1997.
- [9] D. Costa and A. Hertz, Ants can color graphs. *Journal of the operational Research Society*, pp.295-305, 1997.
- [10] B. Bullnheiner, R.F. Hartl and C. Strauss, Applying the ant system to the vehicle routing problem. *Advances and Trends in Local Search Paradigms for Optimization*, pp.109-120, 1998.
- [11] J. Kennedy and R. Eberhart, Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks*, pp.1942-1948, 1995.
- [12] B.B. Thompson, I.I. Marks and J. Robert, Inversion of neural network underwater acoustic model for estimation of bottom parameters using modified particle swarm optimizers. *Proceedings of the International Joint Conference on Neural Networks*, pp.1301-1306, 2003.