ISCIT 2004

International Symposium on Communications and Information Technologies 2004 (ISCIT 2004)
Sapporo, Japan, October 26- 29, 2004

Ant Colony Optimization for Best Path Planning

Ying-Tung Hsiao, Cheng-Long Chuang, and Cheng-Chih Chien

Department of Electrical Engineering
Tamkang University, Taipei, Taiwan, 251, R.O.C.
Tel: +886-2-26215656 Ext. 2786, Fax: +886-2-26209814
E-mail: clchuang@ee.tku.edu.tw

Abstract—**This paper presents an optima approach to search the best path of a map considering the traffic loading conditions. The main objective of this work is to minimizing the path length to get the best path planning for a given map. This study proposes a solution algorithm based on the ant colony optimization technique to search the shortest path from a desired original to a desired destination of the map. The proposed algorithm is implemented in C++. Furthermore, the simulation program can randomly generate maps for evaluating its flexibility and performance. Simulation results demonstrate that the proposed algorithm can obtain the shortest path of a map with fast speed.**

I. Introduction

Best path planning based on the standard search algorithms, such as Dijkstra's algorithm [1], is to find the shortest path by expending, comparing, and selecting promising partial paths that merge the origin toward the destination of the desired trip [2-5]. We determine weather a partial path is promising based on the objective function for specific applications, such as shortest or fastest path for the trips. This paper discusses applications of map guidance to the path planning problems when the moving absolutely freely in a given map is not permitted.

The best path searching algorithms have been applied in many application fields, such as network routing strategy, or driving guidance systems [5] – [7]. Various paths may exist some constraints such as the maximum bandwidth capacity of a line, or the maximum transmission speed of a node, etc. These constraints may influence the result of the best path routing. For a package transmission problem, in network routing, if we route a package through a heavily loading node, the arrival time may be delayed because the overall transmission rate is bear down on the maximum transmission speed. In order to solve this problem, we have to avoid routing the package to the heavily loading nodes instead of routing the package through the network without considering the loading balances. This work has also considered the loading balance conditions for the routing strategy to preserve the smooth package transmission.

This paper formulates the problem of best path searching as an optimization problem to find the shortest path of a map. Moreover, this work proposes a solution algorithm based on the ant colony optimization (ACO) technique. ACO is a general-purpose optimization technique that has been recently developed and applied in many combinatorial optimization problems [14], such as traveling salesman problem [15], quadratic assignment problem [6], graph coloring problems [7], and hydroelectric generation scheduling problems [8]. The feature of the presenting technique different from other methods is that it can be implemented easily, it is flexible for many different problems' formulation, and the most of all, it can escape the local optima of the given problem. In order to adapt some applications regarding management such as network traffic management, this work presents the path max loading parameter to control the package flow of the considering network.

This paper is organized as follows. Section II formalizes the problem of the best path planning. Section III introduces the ACO technique. Appling ACO for planning the best path is proposed in Section IV. Section V discusses experimental results as illustrations. Finally, the conclusions are drawn in the last section.

II. Problem Formulation

We address the best path planning problem in which the passengers in public transportation systems or packages in computer network that cannot change their path absolutely freely. The object of the best path planning is to search the shortest and most efficient path for passengers or packages to pass through the system from the desired original to the desired destination.

For a map, there exists a set of nodes in a virtual 2-dimensional space, and each node N_S has its own unique number S, and coordinate (x_S, y_S). The relationship of each pair of nodes can be defined as unidirectional route or bidirectional route. The bidirectional route was modeled by two separate unidirectional routes. Each route has a relatively Euclidean distance which can determine as follow:

$$D_{Si,Sj} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (1)$$

where $D_{Si,Sj}$ is the distance from route original terminal node S_i to route destination terminal node S_j. And $\{x_i,y_i\}$, $\{x_j,y_j\}$ are the coordinates of the node S_i and node S_j, respectively.

According to the definition of the best path planning problem, the package passing through the system cannot change its path absolutely freely. The package has to transport between two non-connected nodes S_i, S_k by transporting through other nodes as indirect transportation. Hence, S_i -> S_j -> S_k implies that the package transports through node S_i, S_j, S_k. The proposed algorithm employs some of these phrase definitions. All information of nodes and routes are saved on the main memory of the computer to speed up the simulation performance of the best path planning algorithm.

III. Ant Colony Optimization Algorithm

Ant Colony Optimization (ACO) is a graph representation based evolutionary meta-heuristic algorithm. ACO has been successfully employed to solve many different combinatorial optimization problems. The main idea of ACO is to model the problem as a minimum cost path searching problem in a graph. ACO consists of many artificial ants, and these artificial ants walk through the graph to find the shortest path of the graph. An artificial ant has a very simple behavior, and typically, it only is able to find a very poor-quality path by itself. Better paths are found by the global cooperation among all artificial ants in the colony.

The behavior of artificial ants is modeled from biological ants. In real world, biological ants act in a very simple behavior as well. When they walk through a place, they lay pheromone trails on the path. The moving direction of each ant is decided by the consistency of the pheromone that lies on the path. Ants prefer to go through a path with a relatively high among of pheromone on it. In addition, artificial ants have some extra feature that do not find in biological ants. In particular, they live in a discrete world. Their moves consist of transitions from nodes to nodes. Their moves usually associated with their pervious action that stored in the memory with a specific data structure.

The pheromone consistencies of all paths are updated only after the ant finished its trip from the original node to destination node and not during the ant is still walking. Every artificial ant has a constant amount of pheromone stored in it when the ant proceeds from the original node. When the ant has finished the trip to the destination node, the pheromone that stored in it will be distributed averagely on the path of its journey. By this pheromone distribution strategy, if an ant finished its journey with a good path, the average distribution amount of pheromone will be high. In the other hand, if an ant finished its journey with a poor path, then the average distribution amount of pheromone will be low. The amount of pheromone deposited is usually corresponding to the quality of the path. The pheromone of the routes progressively decreases by evaporation in order to avoid artificial ants stick in local optima solution.

procedure ACO
  Set parameters and Initialize pheromone trails
repeat
  for k=1 to m do construct an assignment X_k
  update pheromone trails using $\{X_1,\ldots X_m\}$
until best solution achieved
  or maximum cycle reached

Fig. 1. ACO algorithm

Fig. 1 depicts the ACO algorithm. At each generation, each ant generates a complete journey by choosing the nodes according to a probabilistic state transition rule. It is necessary to spread a little amount of pheromone on all paths at the initial of the algorithm. Every ant selects the nodes in the order in which they will appear in the permutation. For the selection of a node, ant uses heuristic factor as well as pheromone factor. The heuristic factor denoted by η_{ij} and the pheromone factor denoted by τ_{ij} are indicators of how good it seems to have node S_j at node S_i of the permutation. The heuristic value is generated by some problem dependent heuristic whereas the pheromone factor stems from former ants that have found good solutions.

The rule for an ant to choose its next node is called Pseudo-Random-Proportional Action Choice Rule. With probability q_0, where $0 \le q_0 < 1$ is a parameter of the algorithm, the ant chooses a node S_j from the set S of nodes that have not been selected so far which maximizes

$$[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta} \tag{2}$$

where $\alpha \ge 0$ and $\beta \ge 0$ are constants that determine the relative influence of the pheromone values and the heuristic values on the decision of the ant. With probability $(1-q_0)$ the next node is chosen from the set S according to the probability distribution that is determined by

$$p_{ij} = \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{h \in S}[\tau_{ih}]^{\alpha}[\eta_{ih}]^{\beta}} \tag{3}$$

Therefore the transition probability is a trade-off between the heuristic and pheromone factor. For the heuristic factor, the close nodes, that mean the paths with low cost, should be chosen with high probability, thus implementing greedy constructive heuristic. As to the pheromone factor, if on edge (S_i, S_j) there has been a lot of traffic then it is highly desirable, thus implementing the *autocatalytic process*. If the selection results a P_{ij}, carry out the rule (3). In (3), the heuristic factor η_{ij} is computed according to the following rule

$$\eta_{ij} = \frac{1}{f(X_j)} \tag{4}$$

where $f(X)$ represents the cost function of X. In (3), it is favor

that the choice of edges which are shorter, which mean the path with low cost, and which have a greater amount of pheromone. Along the path from S_i to S_j the ant lays a trail substance so defined

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{Q}{L_k} & \text{if } k\text{th ant uses edge } (S_i,S_j) \text{ in its tour} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where Q is a constant related to the quality of pheromone trail laid by ants and L_k is the cost of the journey performed by the kth ant. In other word, pheromone updating is intended to allocate a greater amount of pheromone with low cost, which means shorter journey. This value is evaluated when the ant has completed a journey and consisting of a cycle of n iterations (generations). Then, it is used to update the amount of substance previously laid on the trail, on the following rules

$$\tau_{ij}(t+n) = \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (6)$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \tau_{ij}(t) \quad (7)$$

where m denotes the number of ants, ρ, $\rho \in (0,1)$, is a coefficient of persistence of trail during a cycle such that $(1-\rho)$ represents the evaporation of trail between generation n_g and n_g+1. The pheromone-updating rule was meant to the addition of new pheromone deposited by ants on the visited edges and to pheromone deposited by ants on the stops iterating either when an ant found a solution or when a maximum number of generations have been performed.

IV. Apply ACO to Best Path Planning

The objective of the best path planning is to find the shortest path from the desired original node to desired destination node. The object of this work is to obtain the shortest path for the best path planning problem considering preventing packages passing through heavily loading path and avoiding transportation delay.

First, it is need to setup the map environment. The nodes and routes of a map for testing can set manually, or generate by computer randomly. In order to evaluate the flexibility and performance of the proposed algorithm, we use a random generating map to test it. Fig. 2 shows a random generating map. Every node has its own absolute coordinate, and the length of each route can be calculated by the coordinate of its original terminal node and destination terminal node. The loading parameters LD of each route have to set manually or are set as 0 by default. In the first step of the simulation, it is need to select the desired original node and the destination node.

The solution algorithm of the best path planning is presented as the followings:

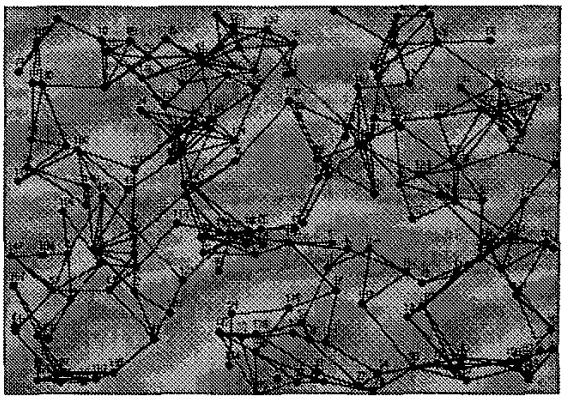

Fig. 2. Random generated map

Step 1. Initialization

An initial population of ant colony individuals X_i, i=1, 2, ... ,m, is initialized in this step. The maximum node transport action number of each ant is E. Format the *journey nodes lists* of each ant to provide the following step to record the ant journey history. Randomly generate a pheromone consistency value and set the value as initial condition for each route. Set time counter t=0 and generation counter n_g=0.

Step 2. Starting tour.

Place all ants on the desired original node $S_{original}$ that is already defined manually.

For ant counter k=0 to m **do**

Let ants search for its path.

Step 3. Searching for neighborhood

Repeat until the *journey nodes list* is full

For k=0 to m **do**

IF ant X_k is already reached the desired destination node $S_{destination}$

Then Break and sim next ant X_{k+1}.

Choose the node j to move to, with probability P_{ij} given in (3).

If node j already exists in the *journey nodes list*

Then Repeat last procedure.

If the ant X_k already stuck at a dead end or reach the maximum transport action number E.

Then suspend ant X_k from simulation.

Move the kth ant to the node S_j.

Insert node j into the *journey nodes list* of ant X_k.

Step 4. Calculate the path length (cost) for each ant's journey.

For k=0 to m **do**

Compute the cost (L_k) of the journey node list of ant X_k.

For every route (S_i, S_j)

For k=1 to m **do**

Calculate the pheromone trail of each edge

according to (5).

Step 5. Update the pheromone factor.

For every edge (S_i,S_j) update the pheromone value according to the update rule (6) and (7).

For every edge (S_i,S_j) updates the pheromone value by multiply it by (LD/LD_{MAX}), where LD_{MAX} is the max traffic volume of the edge.

Find the best path by employ a single ant into the map.

Update the best path found.

Let $t=t+1$; $n_g=n_g+1$

For every edge (S_i,S_j), reset $\Delta\tau_{ij}=0$

Step 6. Check the stop criterion.

If $(n_g<n_{gmax})$ or (not user stop)

Then

Record and draw the best *journey nodes list*.

Clear the *journey nodes list*, and **Goto** Step 2.

Else

Draw the best journey nodes list and **Stop**.

V. Simulation Results

This section presents the simulation results of the proposed algorithm. The proposed solution algorithm is implemented with Borland C++ Builder on a Intel Pentium4 2.66GHz computer with 1GB RAM. The values of the parameters in ACO are n=100, m=100, α=1, β=1, maximum generation n_{gmax}=1000, the original node is $S_{original}$=35 (at the left bottom side of the map), and the destination node is $S_{destination}$=19 (at the right top side of the map).

This work utilizes a random generating map shown as fig. 2 for evaluating the performance of the proposed algorithm. This map consists 100 nodes and 417 edges. The length of the total best path is 824.0537232 (Pixels). In this example, the ACO algorithm takes 0.69 seconds with 94 generations to found the best path shown in Fig. 3.

The next test case is shown in Fig. 4. All parameters of the proposed algorithm are the same as the first simulation The map of the second test case consists of 500 nodes (n=500), and 1473 edges. The desired original node is $S_{original}$=295 (at the left top side of the map), and the desired destination node is $S_{destination}$=299 (at the right bottom side of the map). Fig. 5 displays the simulation result of the second case. In this example, the ACO algorithm takes 6.94 seconds with 514 generations to found the best path.

The 3rd case is the traffic constrains case shown in Fig. 6. All parameters are all the same as former simulations. The map of the 3rd test case consists of 300 nodes (n=300), and 646 edges. The desired original node is $S_{original}$=106 (at the left top side of the map), and the desired destination node is $S_{destination}$=99 (at the right bottom side of the map). There exit heavy traffic loading on the routes of the 3rd test case circled at the bottom of the map. The shortest path of this case includes these edges with heavy traffic loading. The simulation results shown in Fig.7 show that the optimal route searched by the ACO can avoid the package passing through these edges with heavy traffic loading to prevent transportation delays.

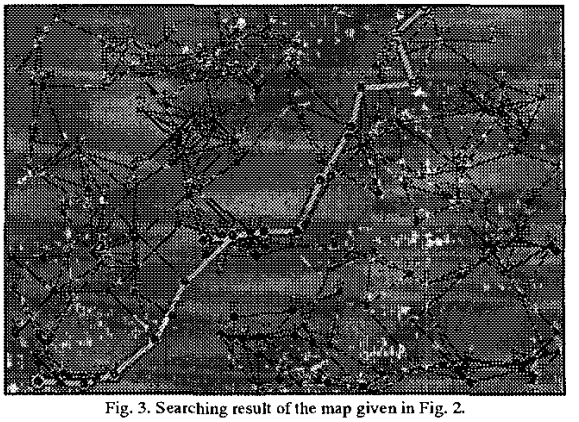

Fig. 3. Searching result of the map given in Fig. 2.

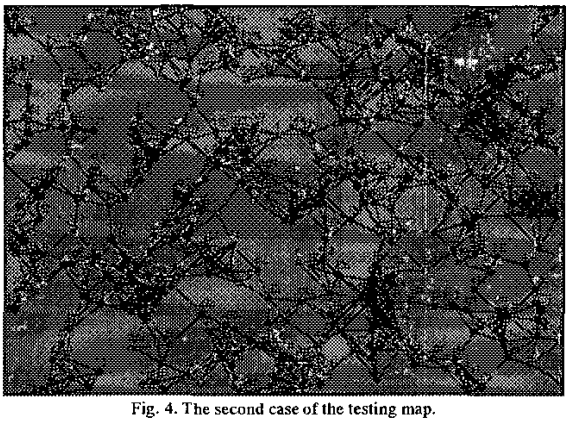

Fig. 4. The second case of the testing map.

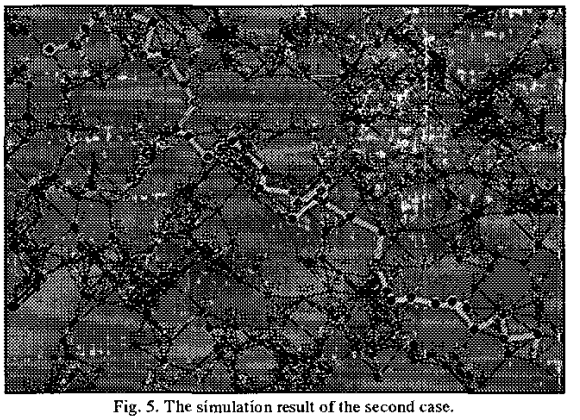

Fig. 5. The simulation result of the second case.

Table 1 summarizes the ACO simulation results compared with of other cases. It is shown that the time requirement of the proposed ACO method is less than of other method for a given map with more than 200 node numbers.

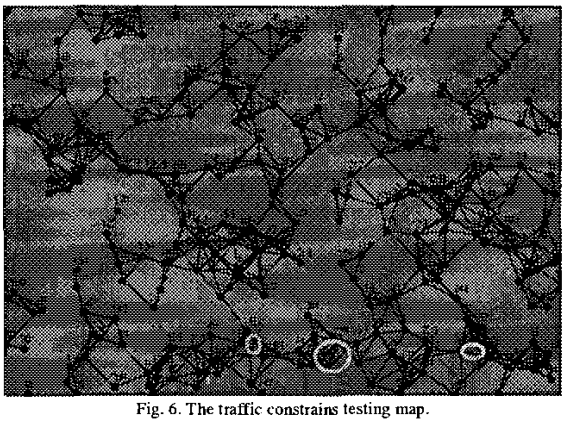

Fig. 6. The traffic constrains testing map.

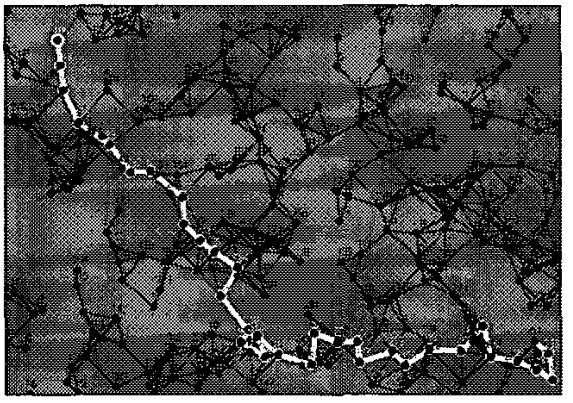

Fig. 7. Simulation result of the 3-rd case.

Table 1
Summary of simulation results

Total Node Number	ACO Length (Pixel)	ACO Time (Sec)	Brute Length (Pixel)	Brute Time (Sec)
100	824.053	0.89	824.053	0.66
200	874.141	2.07	874.141	2.91
300	947.935	2.98	940.417	6.11
400	1046.886	4.13	1041.912	12.19
500	1181.610	6.94	1178.059	19.84

Note: A short distance exists in arbitrarily two nodes does not mean that the two nodes should be connected. In other words, the connection generation mechanism does not consider the distance between two nodes to decide the connectivity of them.

VI. Conclusion

This study presented a novel approach to search the best path of a given map or network. The problem of best path planning is formulated as an optimal problem to get the shortest path length of the map and consider loading constraints. Furthermore, a solution algorithm based on the ACO was developed to derive the global optimal solution. The proposed technique has many advanced features that different from other methods: i) it is easy to deploy and implement to solve many optimal problems, ii) it is very flexible for many problem formulations, and iii) it has the ability to escape the local optima solution and achieve the global optima solution. Finally, the proposed method was implemented and tested on several network with promising results.

References

[1] E. W. Dilkstra, a note on two problems in connextion with graphs, *Numerische Mathematic*, 1, 1959, pp. 269-271.

[2] R. K. Ahuja, T. L. Magnanti, and J. B. Ohlin, Network Flows: Theory, Algorithms, and Applications. Prentice Hall, 1993.

[3] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, 1995.

[4] M. P. Wellman, M. Ford, and K. Larson, "Path planning under time- dependent uncertainty," In Proceedings of the Eleventh Conference on Uncertainty of Artificial Intelligence, 1995, pp. 532-539.

[5] C. L. Liu, "Best path planning for public transportation systems," in Proceedings of the IEEE 5th International Conference on Intelligence Transportation Systems, 2002, pp. 834-839.

[6] K. M. Sim, W. H. Sun, "Ant colony optimization for routing and load-balancing: survey and new directions," IEEE Trans. Systems, Man and Cybernetics, Part A, vol. 33, no. 5, Sept. 2003, pp. 560-572.

[7] K. M. Sim, W. H. Sun, "Multiple ant-colony optimization for network routing," in Proceedings of First International Symposium on Cyber Worlds, 2002, Nov. 2002, pp. 277-281.

[8] M. Dorigo, G. Di Caro, The ant colony optimization meta-heuristic. In D. Corne, M. Dorigo, F. Glover (Eds.), New Ideals in Optimization, McGraw-Hill, 1999.

[9] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents", IEEE trans. on System, Man, and Cybernetics-PartB: Cybernetics, vol. 26, no. 1, Feb. 1993, pp. 29-41.

[10] D. Costa and A. Hertz, "Ants can color graph," J. Oper. Res. Soc., vol. 48, no. 3, Mar. 1997, pp. 295-305.

[11] M. Dorigo, T. S. Zel, Ant colony optimization. Bradford, 2004.

[12] R. S. Parpinelli, H. S. Lopes, A. A. Freitas, "Data mining with an ant colony optimization algorithm," IEEE Trans. Evolutionary Computation, vol. 6, no. 4, Aug. 2002, pp. 321 - 332.

[13] L. Bo, H. A. Abbas, B. McKay, "Classification rule discovery with ant colony optimization," in Proceedings of IEEE/WIC International Conference on Intelligent Agent Technology, 2003, pp. 83-88.

[14] L. Schoofs, B. Naudts, "Ant colonies are good at solving constraint satisfaction problems," in Proceedings of the 2000 Congress on Evolutionary Computation, vol. 2, 2000, pp. 1190-1195.

[15] G. Lu, Z. Liu, "Multicast routing based on ant-algorithm with delay and delay variation constraints," in Proceedings of the 2000 IEEE Asia-Pacific Conference on Circuits and Systems, 2000, pp. 243-246.

[16] S. J. Huang, "Enhancement of hydroelectric generation scheduling using ant colony system based optimization approaches," IEEE Trans. Energy Conversion, vol. 16, no. 3, Sept. 2001, pp. 296-301.