Ashdin Publishing
International Journal of Swarm Intelligence and Evolutionary Computation
Vol. 1 (2012), Article ID Z110403, 5 pages
doi:10.4303/ijsiec/Z110403

Research Article

Intelligent System to Emergencies Based on Ant Colony Optimization

Alberto Ochoa,[1] Julio Ponce,[2] Omar Cano,[3] Paula Hernández,[4] Jöns Sánchez,[5] and Rubén Jaramillo[3]

[1] *Center of Social Researcher, Autonomous University of Ciudad Juárez, 32310 Ciudad Juárez, Chihuahua, Mexico*
[2] *Department of Computer Science, Autonomous University of Aguascalientes, 20100 Aguascalientes, Mexico*
[3] *Postgraduate Program at PICYT, CIATEC, 37545 León, Guanajuato, Mexico*
[4] *Doctoral Program at PICYT, Technological Institute of Ciudad Madero, 89440 Ciudad Madero, Tamaulipas, Mexico*
[5] *Department of Industrial Engineering, University of Guanajuato - Silao Campus, 36130 Silao, Guanajuato, Mexico*
Address correspondence to Alberto Ochoa, alberto.ochoa@uacj.mx

Received 27 April 2011; Revised 26 June 2011; Accepted 27 June 2011

Abstract The shortest path problem is a typical problem of optimization. This paper presents the ant colony optimization (ACO) algorithm to solve a problem of routing vehicles in a Fire Department of Leon, Mexico. In this work, diverse components are described to characterize this problem through the use of a bioinspired algorithm. The algorithm was developed in Java, thus obtaining a tool which determines the best tracks to the vehicles. An experiment was realized to probe the validations, the results were used to compare it with Dijkstra algorithm and determine the quality of results. The future work of this research is determine an innovative perspective related to pheromone evaporation and as this topic is determinative to found and remember the best solutions quickly, additionally we compare it with a code from other postgraduate students trying to implement an algorithm similar to Logistics but using a PSO and another with cultural algorithms.

Keywords ant colony; shortest path; vehicle routing problem

1 Introduction

Today's digital maps are increasingly common. With the progress that has been made in technology, these maps are becoming more sophisticated, able to come to find specific locations, draw routes and so forth. Also noteworthy is that they show how the information has improved dramatically, as they changed from traditional maps to maps with totally real images taken from the air, satellite, or even a hybrid version of these two. The motivation of this project is specifically focused on the use of this increased interaction today, in order to achieve an improvement in the logistics of the heroic Fire Department of Central Apollo in Leon city. The objective of this work is to develop a system to help create routes on the basis of the emergencies given in Leon city, through the use of a system of colonies of ants that allows them to create routes to take care of fires or other types of emergencies in a fast form. This is important because the life of the people is in risk. In Mexico, to minimize the time to arrive to the place of the accident, they realize a comparison according to three possible emergencies at the same time and require other vehicles to respond to them. To provide assistance to citizens, firefighters need a route to arrive as quickly as possible to the place of the incident. If there are many emergencies, they are classified in order of importance: Firefighting, Rescue, and Prevention Action on public hazard. In all these activities, the time is vital because with a timely arrival the effect of the damage in a fire can be decreased to prevent an explosion in the leak case, and find alive persons among others. The bio-inspired algorithms are a technique of the artificial intelligence focused in the solution of different problems, especially optimization problems. One of these algorithms is the swarm intelligence algorithm, where we can find the algorithm of ant colony (ACO), particle swarm optimization (PSO), bees and so on. The proposed algorithm to solve the routing problem in Leon city is an Ant Colony System. This paper is organized as follows: Section 1 is an introduction; Section 2 contains descriptions of the model components; Section 3 displays the Ant Colony algorithms to solve route problems; Section 4 shows the proposed model; Section 5 displays the experimental results. Then we dedicate the last section to the conclusions and future work.

2 Descriptions of the model components

In this section, we offer details of each component related to the application domain that is involved in the problem, in our case we solve a Logistics problem related to the Fire Departments by the use of a bioinspired algorithm to create route of vehicles to attend emergencies.

2.1 The shortest route

The problem known as the shortest path or shortest route, as its name suggests, tries to find the minimum or shortest route between two points. This minimum may be the distance between origin and destination points or the time to travel from one point to another. Mathematically, this system is described as a weighted graph $G = (V, A, d)$ where vertices are represented by $V = \{V_0, V_1, \ldots, V_n\}$, and arcs are represented by $A = \{(v_i, v_j) \mid i \neq j\}$. The distances associated with each arc are represented by the variable C_{ij} measured by the Euclidean distance.

The objective functions of the problem [7] are

$$\min Z = \sum_{\substack{\text{Todos los arcos}\\ \text{definidos}}} C_{ij} X_{ij}. \tag{1}$$

Decision variables are as follows:

X_{ij}: action to move from node i to node j
0 indicates that there is no displacement and 1 indicates that yes there is movement.
C_{ij}: cost or time to get from node i to node j.

Restrictions

Total input flow = total output flow

$$\begin{aligned}&(\text{external input into node } j) + \sum_{\substack{i\\ \text{Todos los arcos}\\ \text{definidos } (i,j)}} X_{ij}\\ &= (\text{external output from node } j) + \sum_{\substack{k\\ \text{Todos los arcos}\\ \text{definidos } (j,k)}} X_{jk}.\end{aligned} \tag{2}$$

This type of optimization problems cannot be solved using exact methods. We cannot find its optimal solution with acceptable computational efforts. Since the early 50s, many algorithms have been developed to find a solution to this problem by finding good solutions but not necessarily optimal solutions. In the 80s, the solution techniques focused on the implementation of general-purpose metaheuristics including among others the ant colony, genetic algorithms and taboo search.

2.2 The shortest path algorithm

The shortest path algorithm, also called the Dijkstra algorithm, is an algorithm for determining the shortest path given in a source vertex to other vertices in a directed graph with weights on each edge. The shortest path algorithm belonging to the greedy algorithm [9] is an efficient algorithm of complexity $O(n^2)$, where n is the number of vertices, used to find the least cost path from a source node to all other nodes in the graph. It was designed by the Dutchman Edsger Wybe Dijkstra in 1959 [13]. The foundation on which sits this algorithm is the principle of optimality; the solution is built with the election of local optima in the hope of obtaining a global optimum.

2.3 Ant colony

Ant colony optimization (ACO) algorithm is a new simulative evolutionary algorithm named ant colony system and was proposed in the 1990s by the Italian scholar Marco Dorigo. It has been applied to TSP, allocation problem, JSP, and got excellent results. Hence, more attention has arisen to the ant colony system, and the model has been applied to many practical problems [5, 12]. The ants are social insects that live in colonies and, because of their mutual cooperation, are capable of displaying complex behaviors and difficult tasks from the point of view of an individual ant. An interesting aspect of the behavior of many species of ants is their ability to find the shortest path between their nest and food sources. This is especially interesting when you consider that many species of ants are almost blind, which avoids the use of visual cues [10]. While they make their way between the nest and food source, some species of ants deposit a chemical called pheromone. If there is no trace of pheromone, the ants move essentially in a random manner, but when there is pheromone deposited, it is more likely to be traced [10]. The choice between different ways takes place when several paths cross. Then, the ants choose the way forward with a probabilistic decision biased by the amount of pheromone: the stronger related to the pheromone trail, the more likely selection. Because ants deposit pheromone on the path to follow, this behavior leads to a self-reinforcing process that concludes with the formation of traces marked by a high concentration of pheromone. This behavior also allows the ants to find the shortest path between their nest and food source [1, 2]. As time passes and while the ants are most promising on the roads, they will receive a higher amount of pheromone. This occurs when was selected the shortest path, the ants that are able to find food more quickly; they begin their return journey before. Then, in the shortest path a trail of pheromone is slightly higher and, therefore, decisions of the following ants will be directed more to the way [10]. In addition, this road will receive a greater proportion of pheromone by ants returning it, the roads longer receive least pheromone amount. This process ends by making that an ant chooses the shortest path with more probability and increases progressively the pheromone amount in the path to the colony.

3 Ant colony algorithms to solve route problems

Actually, some related works exist in the literature. Ghoseiri and Nadjari proposed in 2010 an ant algorithm to the multi-objective shortest path problem (MOSP). MOSP is one of the most important problems in network optimization with wide applications in telecommunication industries, transportation and project management, which is a route problem that has the same characteristics of the problem that is simulated [6]. Another research that can be analyzed are the Yu's works. Yu et al. proposed a temporal ant colony optimization (TACO) algorithm, which is used to search for the shortest paths in the network [15]. Barcos et al. worked with a type of problem that is known as a many-to-many problem (i.e. several origins to several destinations in which each terminal acts simultaneously as origin and destination), which is unlike other classic problems such as the vehicle routing problem, and they implemented an ant colony system [1].

3.1 Relation research with this topic

Many authors described different applications to resolve this kind of problem. Lianxi Hong proposed an algorithm which permits to determine the time to reach a specific point in a city according to different possible scenarios [8]. On the other hand, Hongtao lei et al. proposed a concept similar to ours, with relation to "emergencies," which may occur in any time and place [11], and organize the demand to the vehicles in our case is to rescue units, and try to minimize the effort to attend the demands in a day with the estimation of arrive, and solve another emergence for example bee swarm which can be damaged to children.

4 The proposed model

An intelligent tool was developed using the ant colony algorithm and the programming language Java (J2SE) and as first step we begin with the creation of the graph for the central area covering the Apollo fire station, a total of 2451 streets, avenues and boulevards (edges) and 1710 nodes. Subsequently an entity was designed called "object" to store information about each node, as the impact to neighboring nodes and their respective distance. These objects were related to a data structure called a multidimensional array which saves computer resources, because this structure does not cause the overflow of memory relate with the cells which compound the grid and generates a square incidence matrix, it stores only the necessary track which is visualized to their analysis. The Ant Colony algorithm has proved effectiveness to solve NP-Hard problems when they use multidimensional arrays [2]. The structure of the generic algorithm is as follows [5]:

Algorithm: Optimization based on Ant Colony

```
Initialize_parameters ( )
while not stop_condition ( )
    for ant = 1 to n
        construct_solution ( )
        evaluate_solution ( )
        update_pheromones ( )
    end for
end while
```

The optimization quantity is the distance of the route. Thus, the truck movement cost between loading spots i and j is a function of all separate costs for each factor which affects the track route:

$$d_{ij} = \alpha da_{ij} + \beta db_{ij} + \gamma dc_{ij} + \cdots . \tag{3}$$

Let $t_{ij}(t)$ be the intensity of trail on edge (i, j) at time t.

Each ant at time t chooses the next node, where it will be at time $t + 1$. Therefore, if we call an iteration of the ACO algorithm the n moves carried out by the n ants in the interval $(t, t+1)$, then for every n iterations of the algorithm, each ant has completed a tour. At this point the trail intensity is updated according to the following formula:

$$\tau_{ij}(t+n)\rho \cdot \tau_{ij}(t) + \Delta\tau_{ij}, \tag{4}$$

where ρ is a coefficient that represents the evaporation of trail between

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}. \tag{5}$$

The coefficient ρ must be set to a value < 1 to avoid unlimited accumulation of trail (see note 1). In our experiments, we set the intensity of trail at time 0, $t_{ij}(0)$, to a small positive constant c.

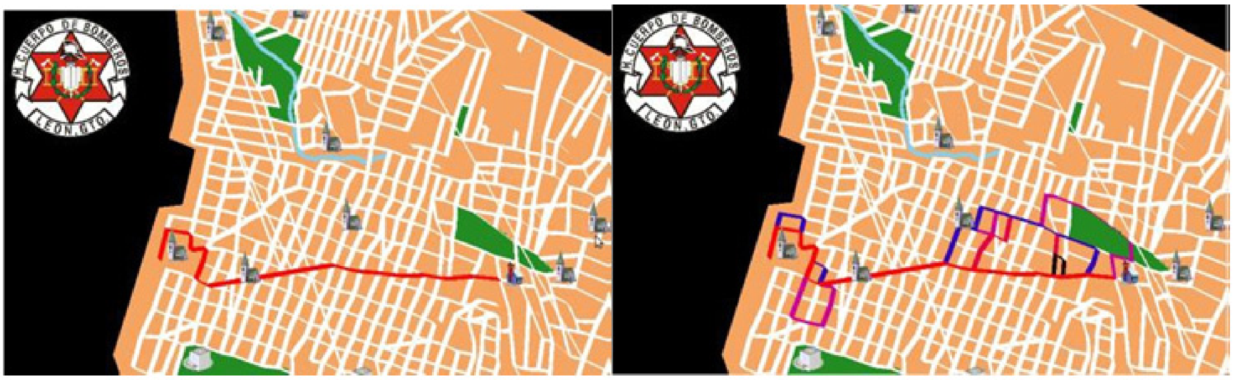
In order to satisfy the constraint that an ant visits all the n different loading spots, we associate with each ant a data structure called the hlist, that saves loading spots already visited up to time t and forbids the ant to visit them again before n iterations (a tour) have been completed. When a tour is completed, the hlist is used to compute the ant's current solution (i.e. the movement cost of the path followed by the ant). The hlist is then emptied and the ant is free to choose again,

$$\eta_{ij} = 1/d_{ij}. \tag{6}$$

We call visibility h_{ij} the quantity $1/d_{ij}$. This quantity is not modified during the run of the AS, as opposed to the trail which instead changes according to the previous formula (4). We define the transition probability from loading spot i to loading spot j for the kth ant as

$$p_{ij}^{k} = \frac{\left[\tau_{ij}(t)\right]^{\alpha} \cdot \left[\eta_{ij}\right]^{\beta}}{\sum_{k \in \text{allowed}_k} \left[\tau_{ik}(t)\right]^{\alpha} \cdot \left[\eta_{ik}\right]^{\beta}}. \tag{7}$$

The software implements the ability to block and alter the meaning of the streets, a fact that occurs in the central city of Leon because of events, accidents, public works and so on. The method Initialize_parameters enters the source node, the destination node, blocked streets and the number of ants involved in the search for the solution similar to the proposal in [14]. construct_solution takes place when ants move randomly with both probabilities using the Monte



(a) (b)

Figure 1: (a) Drawing of a single route. (b) Drawing of five routes.

#	Origin	Destination	Dijkstra	ACOs
1		945	222	212
2		614	464	507
3		903	755	841
4		941	732	698
5		1044	693	709
6		1202	984	1093
7		1094	953	927
8		1057	538	538
9	759	1418	231	231
10		170	338	328
11		526	347	324
12		462	718	718
13		846	859	1030
14		524	359	333
15		809	365	406
16		1107	886	1011
17		698	302	302
18		1062	517	499
19		1342	519	564
20		1199	885	984

Table 1: Results.

Carlo method if there is already a trail of pheromone. Once an ant has found evaluate_solution, the destination node determines if the journey is of good quality, discarding those paths that do not decrease the distance obtained by other ants, and Updating pheromone if you have found a shorter route. The user interface displays the found routes to the destination, with the option of display all of them or some-one in particular in a map (Figure 1), which has the options of adding landmarks (churches, schools, hospitals, parks, rivers), zoom, viewing the different layers, storing in the route file, exporting the map as an image and sending it via Bluetooth to a mobile device.

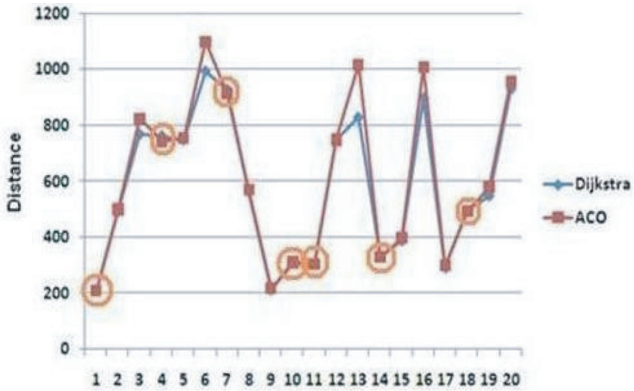


Figure 2: ACO and Dijkstra comparison.

5 Experimental results

The proposed algorithm was compared to the algorithm of Operations Research: The shortest path (Dijkstra). The comparison was carried out with the generation of 20 runs starting from the central fire (node 759) to different nodes (Table 1).

The results were obtained with $\mu = 25.15$ seg and $\sigma = 15.65$ seg, in 35% of cases. While the Ant Colony gives better results than the shortest path algorithm (1, 4, 7, 10, 11, 14 and 18), in 20% the results were similar (8, 9, 12 and 17) and 45% was surpassed by the shortest paths as in Figure 2.

Another comparison was using the same instances and information obtained from Table 1 of three different vehicles from the Fire Department, with intention of built a robust design of experiments try to understand the accumulative number of optimal solutions to reach the best track using the search space to three different codes of PSO, ACO and Cultural Algorithm. The results will be observed in Figure 3.

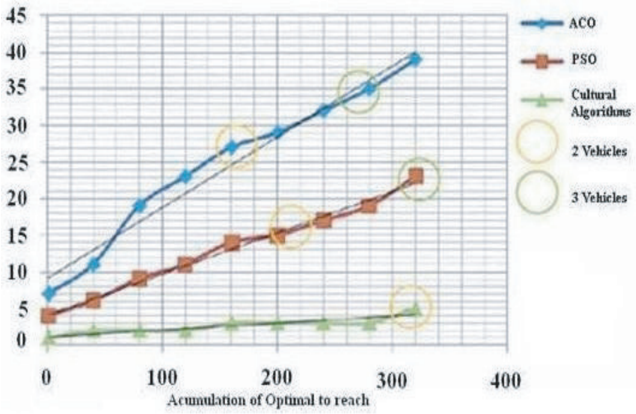


Figure 3: Comparative analysis of an ACO, a PSO and a Cultural Algorithm for an instance of three emergencies at the same time, when we consider its performance on the basis of Table 1.

6 Conclusions and future work

The algorithm currently implemented gives good quality solutions to an NP-hard problem, improving by 35% of cases the routes provided by the shortest path algorithm. The 45% where the shortest path algorithm exceeds the ACO which is attributed to not yet been implemented evaporation of the pheromone, the pheromone amount in nature may to remain a few hours to several months depending on different aspects, such as ant species, soil type [7], causing a minor influence on the effect of evaporation in the process of finding the shortest path. Due to the long persistence of pheromone, it is difficult for the ants to "forget" a path that has a high level of pheromone but have found a way even shorter. Keep in mind that if this behavior is transferred to the computer to design a search algorithm sometimes it can converge quickly to the local optimum. In this section, the results of the trial are presented. First of all, the data collection and the measurement of variables are described. Based on the results obtained, we recommend the implementation of heuristic algorithms such as ant colony, which have supplied to do well on a variety of problems [3,4].

As future work, it would be important to implement the evaporation of the pheromone, find benchmarks that are being used at international level and prove to those instances of the problem, replicate the project using Java (J2ME) for the system to operate on mobile devices which provide advantages to the system in units of H. Fire Department.

We decide to realize a comparison of our algorithm with relation to a PSO Algorithm and a Cultural Algorithm. We discovered the proposal of ACO Algorithm to obtain three different paths to a successful number of emergencies occuring at the same time, and that its performance improves by 22% the performance of PSO Algorithm and by 37% the performance of Cultural Algorithms.

In the future research, it is important to describe the different times in other quadrants (the city is divided in four regions named Quadrants) of the city in the border of zone which covers the Station of Fire Department, which covers only the 17% of territorial space of the city.

References

[1] L. Barcos, V. M. Rodríguez, M. J. Álvarez, and F. Robusté, *Routing design for less-than-truckload motor carriers using ant colony techniques*. Business Economics Working Papers, Series 14. Universidad Carlos III de Madrid. Departamento de Economía de la Empresa, paper number wb043814, 2004.

[2] J. E. Bell and P. R. McMullen, *Ant colony optimization techniques for the vehicle routing problem*, Advanced Engineering Informatics, 18 (2004), 41–48.

[3] M. Birattari, P. Pellegrini, and M. Dorigo, *On the invariance of ant system*, in Ant Colony Optimization and Swarm Intelligence, 5th International Workshop, ANTS 2006, LNCS 4150, M. Dorigo, L. M. Gambardella, M. Birattari, A. Martinoli, R. Poli, and T. Stützle, eds., Springer-Verlag, Berlin, 2006, 215–223.

[4] M. Dorigo, M. Birattari, and T. Stützle, *Ant colony optimization – artificial ants as a computational intelligence technique*, IEEE Computational Intelligence Magazine, 1 (2006), 28–39.

[5] M. Dorigo and G. Di Caro, *The ant colony optimization meta-heuristic*, in New Ideas in Optimization, D. Corne, M. Dorigo, and F. Glover, eds., McGraw Hill, London, UK, 1999, 11–32.

[6] K. Ghoseiri and B. Nadjari, *An ant colony optimization algorithm for the bi-objective shortest path problem*, Applied Soft Computing, 10 (2010), 1237–1246.

[7] R. P. Grimaldi, *Matemáticas Discretas y Combinatoria*, Editorial Addison Wesley, México, 5th ed., 1995.

[8] L. Hong, *An improved LNS algorithm for real-time vehicle routing problem with time windows*, Comput Oper Res, 39 (2012), 151–163.

[9] R. Johnsonbaugh, *Matemáticas Discretas*, Grupo Editorial Pearson Education, México, 6 ed., 2005.

[10] S. R. Kolavali and S. Bhatnagar, *Ant colony optimization algorithms for shortest path problems*, in Network Control and Optimization, vol. 5425 of Lectures Notes in Computer Science, Springer-Verlag, Berlin, 2009.

[11] H. Lei, G. Laporte, and B. Guo, *The capacitated vehicle routing problem with stochastic demands and time windows*, Comput Oper Res, 38 (2011), 1775–1783.

[12] J. C. Ponce, *Diseño de un Modelo de Optimización de Colonias de Hormigas Aplicado al Problema del Clique Máximo*. Tesis Doctoral. Universidad Autónoma de Aguascalientes, Doctorado en Ciencias de la Computación, Aguascalientes, México, 2010.

[13] H. A. Taha, *Investigación de Operaciones*, Editorial Alfaomega, México, 5th ed., 2008. No. de páginas 960.

[14] W. Xiong, L. Wang, and C. Yan, *Binary ant colony evolutionary algorithm*, Int J Inform Technol, 12 (2006), 10–20.

[15] F. Yu, Y. Li, and T. Wu, *A temporal ant colony optimization approach to the shortest path problem in dynamic scale-free networks*, Physica A: Statistical Mechanics and its Applications, 389 (2010), 629–636.