A New Swarm Intelligence Approach for Routing Algorithms

Jose L. Aguilar

Universidad de Los Andes, Av. Tulio Febres Cordero, Facultad de Ingeniería, CEMISID, Dpto. De Computación, Escuela de Ingenieria Sistemas,
5101, Mérida, Estado Mérida, Venezuela,
aguilar@ula.ve

Abstract. In this paper is presented a distributed algorithm based on Ant System concepts, called Combinatorial Ant System, to solve dynamic combinatorial optimization problems. Our approach consists of mapping the solution space of the dynamic combinatorial optimization problem in the space where the ants will walk, and defining the transition probability and the pheromone update formula of the Ant System according to the objective function of the optimization problem. We test our approach on a telecommunication problem.

1 Introduction

Real Ants are capable of finding the shortest path from a food source to their nest without using visual cues by exploiting pheromone information [1]. While walking, ants deposit pheromone trails on the ground and follow pheromone previously deposited by other ants. The above behavior of real ants has inspired the Ants System (AS), an algorithm in which a set of artificial ants cooperate to the solution of a problem by exchanging information via pheromone deposited on a graph. Dorigo [2] proposed the first AS in his Ph.D. thesis. AS has been applied to the traveling salesman problem and quadratic assignment problem, among others combinatorial optimization problems [1-9]. On the other hand, different groups have been working on various extended versions of the AS paradigm (Ant-Q, etc.) [1, 5, 6].

In the AS applied to the Traveling Salesman Problem (TSP), a set of cooperating agents, called ants, cooperate to find good solutions to TSP's using an indirect form of communication through pheromone trails that they deposit on the edges of the TSP graph while building solutions. Informally, each ant constructs a TSP solution in an constructive way: it adds new cities to a partial solution by exploiting information gained from both past experience and a greedy heuristic. Memory takes the form of pheromone trails deposited by ants on TSP edges, while heuristic information is simply given by the edge's weights. There are two reasons to use the AS on the TSP: a) The TSP graph represents the solution space of this problem; b) The AS transition function has goals similar to the TSP objective function.

That is not the case for other combinatorial optimization problems. We have proposed a distributed algorithm based on AS concepts, called the Combinatorial Ant System (CAS), to solve static discrete-state combinatorial optimization problems [8, 9]. The main novel idea introduced by our model is the definition of a general procedure to solve Combinatorial Optimization Problems using AS. In our approach, the graph that describes the solution space of the Combinatorial Optimization Problem is mapped on the AS graph, and the transition function and the pheromone update formula of the AS are built according to the objective function of the Combinatorial Optimization Problem. In this paper we test the CAS on dynamic combinatorial optimization problems, that is, problems changing over time. Particularly, we study a telecommunication problem. This paper is organized as follows: Section 2 presents the AS and the CAS. Section 3 summarizes the experiments. Finally, conclusions of this work are presented in Section 4.

2 Theoretical Aspects

2.1 The Routing Problem like a Dynamic Combinatorial Optimization Problem

A dynamic combinatorial optimization problem is a problem changing over time. That is, it is a distributed time-varying problem which is a current challenger in the combinatorial optimization domain. The dynamic problem that we are going to study is the routing in telecommunication networks. Routing is a mechanism that allows information transmitted over a network to be routed from a source to a destination through a sequence of intermediate switching/buffering stations or nodes. Routing is necessary because in real system not all nodes are directly connected. The problem to be solved by any routing system is to direct traffic from sources to destinations maximizing network performances (e.g., rate of call rejection, throughput, etc.). In real networks traffic, the conditions and the structure of the network are constantly changing, for this reason are necessary dynamic routing algorithms.

2.2 Ant Systems

In general, the behavior of Ant Colonies is impressing to perform their objective of survival. It is derived from a process of *Collective Behavior*. This process is based on the ant communication capacities, which define the inter-relations between them. These inter-relations permit the transmission of information that each ant is processing. The communication among agents (ants) is made through a trace, called *pheromone*. Thus, an ant leaves a certain quantity of pheromone trail when it moves. In addition, the probability that an ant follows a path depends on the number of ants having taken the path (a large quantity of pheromone in a path means a large probability that it will be visited).

AS is the progenitor of all research efforts with ant algorithms and it was first applied to the TSP problem [2, 4]. Algorithms inspired by AS have manifested as heuristic methods that permit resolving combinatorial optimization problems. These algorithms mainly rely on their versatility, robustness and operations based on populations. The procedure is based on the search of agents called "ants", that is, agents with very simple capabilities that try to simulate the behavior of the ants.

AS utilizes a graph representation (*AS graph*) where each edge (r, s) has a desirability measure γ_{rs}, called *pheromone*, which is updated at run time by artificial ants. Informally, the AS works as follows. Each ant generates a complete tour by choosing the nodes according to a probabilistic state transition rule; ants prefer to move to nodes that are connected by short edges, which have a high pheromone presence. Once all ants have completed their tours, a global pheromone updating rule is applied: a fraction of the pheromone evaporates on all edges, and then each ant deposits an amount of pheromone on edges which belong to its tour in proportion to how short this tour was. Then, we continue with a new iteration of the process.

The state transition rule used by ant system is given by the equation (1), which gives the probability with which ant *k* in city *r* chooses to move to the city *s* while building its t^{th} tour (transition probability from node *r* to node *s* for the k^{th} ant) [1-5]:

$$P_{rs}^{k}(t)=\begin{cases}[\gamma_{rs}(t)]^{\alpha}[\eta_{rs}]^{\beta}\Big/\sum_{u\in J_{r}^{k}}[\gamma_{ru}(t)]^{\alpha}[\eta_{ru}]^{\beta} & \text{If } s\in J_{r}^{k}\\ 0 & \text{Otherwise}\end{cases} \qquad (1)$$

Where $\gamma_{rs}(t)$ is the pheromone at iteration *t*, η_{rs} is the inverse of the distance between city *r* and city *s* (d(r,s)), $J_k(r)$ is the set of nodes that remain to be visited by ant *k* positioned on node *r* and, β and α are two adjustable parameters which determine the relative importance of trail intensity (γ_{rs}) versus visibility (η_{rs}). In AS, the global updating rule is implemented as follows. Once all ants have built their tours, pheromone (that is, the trail intensity) is updated on all edges according to the equation [1-5]:

$$\gamma_{rs}(t)=(1-\rho)\gamma_{rs}(t-1)+\sum_{k=1}^{m}\Delta\gamma_{rs}^{k}(t) \qquad (2)$$

Where ρ is a coefficient such that $(1-\rho)$ represents the trail evaporation in one iteration (tour), *m* is the number of ants, and $\Delta\gamma_{rs}^{k}(t)$ is the quantity per unit of length of trail substance laid on edge (r, s) by the k^{th} ant in that iteration:

$$\Delta\gamma_{rs}^{k}(t)=\begin{cases}1\big/L_{k}(t) & \text{If edge}(r,s)\in\text{tour completed by ant k}\\ 0 & \text{Otherwise}\end{cases}$$

Where $L_k(t)$ is the length of the tour performed by ant *k* at iteration *t*. Pheromone updating is intended to allocate a greater amount of pheromone to shorter tours. The general algorithm is summarized as follows:

Place the *m* ants randomly on the nodes of the AS graph

```
Repeat until system convergence
 2.1. For i=1, n
    2.1.1. For j= 1, m
        2.1.1.1. Choose the node $s$ to move to, according
           to the transition probability (equation 1)
        2.1.1.2. Move the ant $m$ to the node $s$
 2.2 Update the pheromone using the pheromone update
 formula (equation 2)
```

2.3 The Combinatorial Ant System

There are two reasons for using AS on the TSP. First, the TSP graph can be directly mapped on the AS graph. Secondly, the transition function has similar goals to the TSP. This is not the case for other combinatorial optimization problems. In [8, 9], we have proposed a distributed algorithm based on AS concepts, called the CAS, to solve Combinatorial Optimization Problems. In our approach, we need to define:

- *The graph that describes the solution space of the Combinatorial Optimization Problem (COP graph).* The solution space is defined by a graph where the nodes represent partial possible solutions to the problem, and the edges the relationship between the partial solutions. This graph will be used to define the *AS graph* (this is the graph where the ants will walk).
- *The transition function and the pheromone update formula of the CAS,* which are built according to the objective function of the Combinatorial Optimization Problem.

In this way, we can solve any Combinatorial Optimization Problem. Each ant builds a solution walking through the AS graph using a transition rule and a pheromone update formula defined according to the objective function of the Combinatorial Optimization Problem. The main steps of CAS are:

- Build the AS graph.
- Define the transition function and pheromone update formula of the CAS.
- Execute the classical AS procedure (or one of the improved versions).

Building the AS graph

The first step is to build the COP graph, then we define the AS graph with the same structure of the COP graph. The AS graph has two weight matrices: the first one is defined according to the COP graph and registers the relationship between the elements of the solution space (COP matrix). The second one registers the pheromone trail accumulated on each edge (pheromone matrix). This weight matrix is calculated/updated according to the pheromone update formula. When the incoming edge weights of the pheromone matrix for a given node become high, this node has a high probability to be visited. On the other hand, if an edge between two nodes of the COP

matrix is low then it means that ideally if one of these nodes belongs to the final solution then the other one must belong too. If the edge is equal to infinite then it means that the nodes are incompatible (they can't be at the same time in a final solution).
We define a data structure to store the solution that every ant *k* is building. This data structure is a vector (A^k) with a length equal to the length of the solution (number of nodes that an ant must visit). For a given ant, the vector keeps each node of the AS graph that it visits.

Defining the transition function and the pheromone update formula
The state transition rule and the pheromone update formula are built using the objective function of the combinatorial optimization problem. The transition function between nodes is given by:

$$Tf\left(\gamma_{rs}(t), Cf^{k}_{r->s}(z)\right) = \gamma_{rs}(t)^{\alpha} \Big/ Cf^{k}_{r->s}(z)^{\beta}$$

Where $Cf^{k}_{r->s}(z)$ is the cost of the partial solution that is being built by the ant *k* when it crosses the edge (r, s) if it is in the position *r*, *z-1* is the current length of the partial solution (current length of A^k), and, α and β are two adjustable parameters that control the relative weight of trail intensity ($\gamma_{rs}(t)$) and the cost function. In the CAS, the transition probability is as follows: an ant positioned on node *r* choose the node *s* to move according to a probability $P^{k}_{rs}(t)$, which is calculated according to the equation given by:

$$P^{k}_{rs}(t) = \begin{cases} Tf(\gamma_{rs}(t), Cf^{k}_{r->s}(z)) \Big/ \sum_{u \subset J^{k}_{r}} Tf\left(\gamma_{ru}(t), Cf^{k}_{r->u}(z)\right) & \text{If } s \in J^{k}_{r} \\ 0 & \text{Otherwise} \end{cases}$$

When β=0 we exploit previous solutions (only trail intensity is used) and when α=0 we explore the solution space (a stochastic greedy algorithm is obtained). A tradeoff between quality of partial solutions and trail intensity is necessary. The pheromone updating rule is defined by the equation (2), where the quantity per unit of length of trail substance laid on edge (r, s) by the k^th ant in that iteration ($\Delta\gamma_{rs}^{k}(t)$) is calculated according to the following formula:

$$\Delta\gamma^{k}_{rs}(t) = \begin{cases} 1 \Big/ C^{k}_{f}(t) & \text{If edge } (r,s) \text{ has been crossed by ant } k \\ 0 & \text{Otherwise} \end{cases}$$

Where $C_j^k(t)$ is the value of the cost function (objective function) of the solution proposed by ant k at iteration t. The general procedure of our approach is summarized as follows:

```
1. Generation of the AS graph.
2. Definition of the state transition rule and the pheromone update formula, according to the Combinatorial Optimization Problem.
3. Repeat until system convergence
    3.1. Place the m ants on different nodes of the AS graph.
    3.2. For i=1, n
        3.2.1 For j= 1, m
            3.2.1.1. Choose the node s to move to, according to the transition probability (equation 3).
            3.2.1.2. Move the ant m to the node s.
    3.3 Update the pheromone using the pheromone update formula (equations 2 and 4).
```

3 Experiments

3.1 Routing Problem Resolution using the CAS

We can use our approach for point to point or point to multipoint requests. In the case of N nodes, N ants are launched to look for the best path to the destination. For a multipoint request with m destinations, N.m ants are launched. The source node keeps in memory all paths that have been found by ants. Then, it chooses the best one. Finally, the path is reserved and a connection is eventually set up (in the case of a multipoint request, it is spanning trees found by ants to the multiple destination nodes which are compared).

Building the AS graph

For this case we use the pheromone matrix of our AS graph like the routing table of each node of the network. Remember that this matrix is where the pheromone trail is deposited. Particularly, each node i has k_i neighbors, is characterized by a capacity C_i, a spare S_i, and by a routing table $R_i=[r^i_{n,d}(t)]_{ki,N-1}$. Each row of the routing table corresponds to a neighbor node and each column to a destination node. The information of each row of the node i is stored in the respective place of the pheromone matrix (p.e., in the position i, j if k_i neighbor = j). The value $r^i_{n,d}(t)$ is used as a probability. That is, the probability that a given ant, where the destinatior is node d, be routed from node i to neighbor node n. We use the COP matrix of our AS graph to describe the network structure. If there are link or node failures, then the COP graph is modified to show that. In addition, in each arc of the COP graph is stored the estimation of the trip times from the current node i to its neighbor node j, denoted $\Gamma_i=\{\mu_{i\text{-}>j}, \sigma^2_{i\text{-}>j}\}$, where $\mu_{i\text{-}>j}$ is the average estimated trip times from node i to node j, and $\sigma^2_{i\text{-}>j}$ is its associated variance. Γ_i allows maintenance a local idea of the global network's status at node i. Finally, we define a cost function for every node, called $C_{ij}(t)$, that is the cost associated with this link. It is a dynamic variable that depends on the link's load, and is calculated at time *t* using Γ_i.

Defining the transition function and the pheromone update formula

In our model (equation 3), $C_j^k(t)$ is the cost of kth ant's route, $\Delta\gamma_{is}^k(t)$ is the amount of pheromone deposited by ant k if edge (i, s) belongs to the kth ant's route (it is used to update the routing table R_i in each node), and $P_{ij}^k(t)$ is the probability that ant k chooses to hop from node i to node j. Ant k updates its route cost each time it traverses a link $C_j^k(t)=C_j^k(t)+C_{ij}(t)$. An ant collects the experience queues and traffic load, which allows it to define information about the state of the network. Once it has reached its destination node d, ant k goes all the way back to its source node through all the nodes visited during the forward path, and updates the routing tables (pheromone concentration) and the set of estimations of trip times of the nodes that belong to its path (COP graph) as follows:

- The times elapsed of the path i->d ($T_{i\text{-}>d}$) in the current kth ant's route is used to update the means and variance values of Γ_i. $T_{i\text{-}>d}$ gives an idea about the goodness of the followed route because it is proportional to its length from a point of view and from a traffic congestion point of view.
- The routing table R_i is changed by incrementing the probability $r^j_{i-1,d}(t)$ associated with the neighbor node i-1 that belongs to the kth ant's route and the destination node d, and decreasing the probabilities $r^j_{n,d}(t)$ associated with other neighbor nodes n, where n ≠ i-1, for the same destination (like a pheromone trail). The values stored in Γ_i are used to score the trip times so that they can be transformed in a reinforcement signal $r= f_i(\Gamma_i)$, $r\in[0,1]$. r is used by the current node i as a positive reinforcement for the node i-1:

$$r^j_{i-1,d}(t+1) = r^j_{i-1,d}(t)\,(1-r)+r$$

And the probabilities $r^j_{n,d}(t)$ for destination d of other neighboring nodes n receive a negative reinforcement

$$r^j_{n,d}(t+1) = r^j_{n,d}(t)\,(1-r) \qquad \text{for } n \neq i-1$$

Finally, $C_{ij}(t)$ is updated using Γ_i too

$$C_{ij}(t+1)= \mu_{i\to j}/\sigma^2_{i\to j}$$

3.2 Result Analysis

We have tested our algorithm on a set of model networks among which is US NSFNET-T1 (composed by 14 nodes and 21 bidirectional links, with a bandwidth of 1.5 Mbits and propagation delay with range from 4 to 20 ms). A number of different traffic patterns, both in term of spatial and temporal characteristics, have been considered. The network performance is expressed in throughput (delivered bits/s) and delivered time from source to destination. We compare our algorithm with the AntNET approach and the Shortest Path First algorithm (SPF) [1, 7]. Due to the space, we present part of the result, see [9] for the rest of experiments. Figures 1 and 2 show some results regarding throughput and packet delay for a Poisson temporal and random spatial distribution of traffic (this is the traffic pattern used) on NSFNET. These results are exemplar of the behavior of our algorithms, results obtained on other traffic pattern ant network topology combinations are qualitatively equivalent (see [9] for more details).

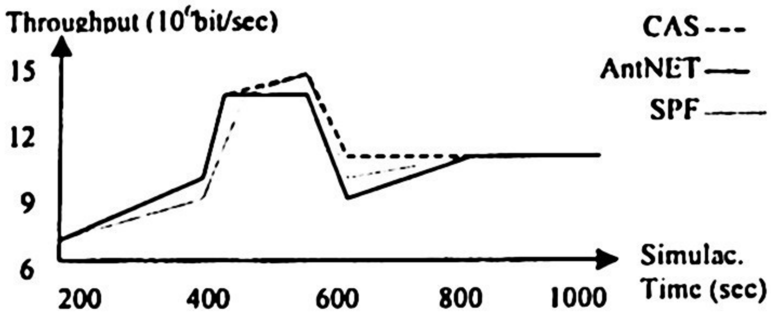

Fig 1. Throughput comparison between the algorithms

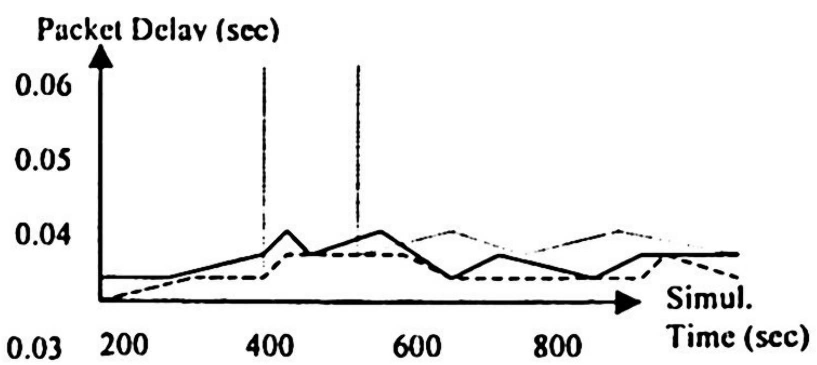

Fig 2. Packet delay comparison between the algorithms

The throughput of our approach is at least as good as that AntNET and the packet delays are much better than that of the others. Particularly, at the beginning our approach has not the best performance because it has learnt the current network situation, etc. After, it can optimize the route to be chosen in an impressing way.

4 Conclusions

In this work we have presented the versatile of the CAS to solve dynamic combinatorial optimization problems. Our system is suited for both static discrete-state and dynamic combinatorial optimization problems. This versatility has been exemplified by the possibility of using the same model to solve different combinatorial optimization problems (static and dynamic) of various sizes. Our approach can be applied to any combinatorial optimization problems by defining an appropriate graph representation of the solution space of the problem considered, the dynamic procedure to update that representation, and an objective function that guides our heuristic to build feasible solutions. In our approach, the dynamic environment of the combinatorial optimization problem is defined through the COP matrix (it form part of the space where the ants will walk (AS graph)). Ants walk through this space according to a set of probabilities updated by a state transition and a pheromone update rule defined according to the objective function of the combinatorial optimization problem considered.

We have tested our approach on a dynamic optimization problem (the routing problem). The results show that our approach obtains good performances, but we must improve the execution time of a given iteration and reduce the number of iterations. In general, CAS allows making an exhaustive searched, in this way it can obtain better performances than previous heuristic routing algorithms. Furthermore, we will develop a parallel version of our approach, we will test our approach over other dynamic combinatorial optimization problems. In addition, for the routing problem, we will test with a general packet-switching network avoiding the "symmetric path costs", and we will develop a network failure management system based on this approach.

References

1. E. Bonabeau, M. Dorigo and G. Theraulaz, Swarm Intelligence: from Natural to Artificial Swarm Systems, Oxford University Press: USA (1999).
2. Dorigo, M. Optimization, Learning and Natural Algorithms, Ph.D Thesis, Politecnico de Milano, Italy (1992).
3. Corne, D., Dorigo, M., Glove, F. New ideas in Optimization, McGraw Hill: Holland, 1999.
4. Dorigo, M., Maniezzo V., Coloni, A. The ant system: Optimization by a colony of cooperating agents, IEEE Trans. Syst. Man, Cybern., Vol. 26, (1996) 29-41.
5. Dorigo M., Gambardella, L. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem, IEEE Trans. on Evolutionary Computation, Vol. 1, (1997) 53-66.
6. Hidrobo F., Aguilar, J. Toward a Parallel Genetic Algorithm Approach Based on Collective Intelligence for Combinatorial Optimization Problems, Proc. IEEE International Conference on Evolutionary Computation. (1998). 715-720.
7. Schoonderwoerd, R., Holland, O., Bruten, P., Rothkrantz, L. Ant-based Load Balancing in Telecommunications Networks, Adaptive Behavior, Vol. 5, (1997) 169-207.

8. Aguilar, J., Velásquez, L., Pool, M. The Combinatorial Ant System, Applied Artificial Intelligence Journal, Vol. 18, (2004) 427-446.

9. Aguilar, J. La Inteligencia Colectiva en la Resolución de Problemas de Optimización Combinatoria Dinámicos, Proceeding of the XIII Simposio Internacional de Métodos Matemáticos Aplicados a las Ciencias, (2004) 21-22.