

## Ant Colony Solving Multiple Constraints Problem: Vehicle Route Allocation

Sorin C. Negulescu, Claudiu V. Kifor, Constantin Oprean

**Abstract:** Ant colonies are successfully used nowadays as multi-agent systems (MAS) to solve difficult optimization problems such as travelling salesman (TSP), quadratic assignment (QAP), vehicle routing (VRP), graph coloring and satisfiability problem. The objective of the research presented in this paper is to adapt an improved version of Ant Colony Optimisation (ACO) algorithm, mainly: the Elitist Ant System (EAS) algorithm in order to solve the Vehicle Route Allocation Problem (VRAP). After a brief introduction in the first section about MAS and their characteristics, the paper presents the rationale within the second section where ACO algorithm and its common extensions are described. In the approach (the third section) are explained the steps that must be followed in order to adapt EAS for solving the VRAP. The resulted algorithm is illustrated in the fourth section. Section five closes the paper presenting the conclusions and intentions.

**Keywords:** Ant Colony Optimisation, Vehicle Route Allocation Problem, Multi-Agent Systems.

### 1 Introduction

Ant colonies are used nowadays as multi-agent systems to solve different optimization problems from the NP-hard class. Examples of such problems are traveling salesman (TSP), quadratic assignment (QAP), vehicle routing (VRP), graph colouring and satisfiability problem.

The multi-agent systems are copying some or all the characteristics of their biological counterparts (depending on how much these characteristics are helpful to the MAS solving a particular type of problem) such as [6]:

- distributed society of autonomous individuals/agents;
- fully distributed control among the agents;
- localized communications among the individual;
- stochastic agent decisions;
- system-level behaviours transcending the behavioural repertoire of the single (minimalist) agent;
- simple interaction rules.

Consequently, the overall very important features of the system are: robustness, adaptability and scalability.

The paper presents the technique of solving yet another difficult problem (VRAP) which belongs to the NP-hard class of problems. Related work, regarding the ACO and EAS algorithms, is described recently (2006-2007) in [1], [2], [3], [9]; to weaken redundancy, here details are skipped over. As a result, the rest of the paper is structured as follows: Section 2 expounds the rationale where Ant Colony Optimisation (ACO) and its common extensions are explained. Section 3 describes the steps that must be followed in order to adapt EAS for solving the VRAP. The next section concentrates on presenting the resulted algorithm. Conclusions and directions of future work (section 5) are closing the paper.

## 2 Rationale and Approach

The ant colony optimization algorithm (ACO), introduced by Marco Dorigo in 1992 in his PhD thesis, is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs and is inspired by the behaviour of ants in finding paths from the colony to food [8].

In all early Ant System (AS) algorithms, ants are searching for (candidate) solutions based on two main components: pheromone trails and problem-dependent heuristic information. These algorithms have suffered frequent modifications in order to improve their efficiency. Thus, the AS [5] developed into the Elitist Ant System (EAS) [4], because each ant that finds a better solution has the chance to deposit more pheromone.

Other systems that emerged from the AS are the Max-Min Ant System (MMAS) [10] and the Ant-Q [7] where the deposited pheromone amount is directly proportional to the quality of the found solution. In the Rank-Based Ant System (ASrank) all solutions are ranked accordingly to their fitness. The amount of pheromone deposited is then weighted for each solution, such that the more optimal solutions deposit more pheromone than the less optimal solutions.

The basic VRAP consists in allocating vehicles from a fleet to different routes in such a way that the average distance travelled in a month is (optimally) equal for all vehicles.

Before designing an ACO algorithm some questions regarding important aspects of the problem must be answered first [6]:

- The representation of the problem (pheromone model): *What are the most effective state features to consider for learning "the good decisions"?*
- Heuristic variables: *What are they and what is their relative weight regarding the pheromone in the decisions?*
- Ant-routing table: *How pheromone and heuristic variables are combined together to define the goodness of a decision?*
- Stochastic decision policy: *How to make good exploration without sacrificing exploitation of promising/good actions?*
- Regarding pheromones: *Is it useful to put limits in pheromone ranges? How should the pheromone initialization and evaporation be dealt with? What is the best policy for pheromone updating: online or offline?*
- Scheduling of the ants: *How and/or how many per iteration?*
- Restarting the algorithm after stagnation: *Would it be better to restart the algorithm when stagnation is detected?*
- Daemon components: *What is the effect of local search?*

The standard form of the EAS algorithm for solving the TSP is presented in figure 1.

```

// Initialize pheromone trails
  for (every edge i, j) {
     $\tau = \tau_0$ 
  }
// Choose the starting town for every ant
for (k = 1; k ≤ m; k++) {
  Place ant k on a randomly chosen city
}
// Initialize the best tour and length
T+ = the shortest tour found from the beginning
L+ = the length of the best tour
// Main loop
for (t = 1; t ≤ Tmax; t++) {
  // Compute a tour for every ant
  for (k = 1; k ≤ m; k++) {
    Build tour Tk(t) by applying n – 1 times the following step:
    Choose the next node (city) j with the probability
    
$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{l \in J_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}(t)]^\beta}, \text{ if } j \in J$$


$$p_{ij}^k(t) = 0, \text{ if } j \notin J$$

    where i is the current city.
  }
  // Compute the tour lengths for all ants
  for (k = 1; k ≤ m; k++) {
    Compute the length Lk(t) of the tour Tk(t) produced by ant k
  }
  // Update the best tour if an improved tour is found
  if (an improved tour is found) {
    Update T+ and L+
    print T+ and L+
  }
  // Global update for the pheromone trails
  for (every edge i, j) {
    Update the pheromone trails by applying the rule:
    
$$\tau_{ij}(t) = (1 - \rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t) + e \cdot \tau_{ij}^e(t), \text{ where}$$


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


$$\Delta\tau_{ij}^k(t) = \begin{cases} Q/L^k(t), & \text{if } (i,j) \in T^k(t) \\ 0, & \text{otherwise} \end{cases}$$

    and
    
$$\tau_{ij}^e(t) = \begin{cases} Q/L^+, & \text{if } (i,j) \in T^+ \\ 0, & \text{otherwise} \end{cases}$$

  }
  // Calculate the intensity of the pheromone for next iteration
  for (every edge i, j) {
     $\tau_{ij}(t+1) = \tau_{ij}(t)$ 
  }
}

```

Figure 1: Elitist Ant System pseudocode for solving the Travelling Salesman Problem.

In this paper is presented a problem that has more constraints than TSP in its basic form:

- The number of routes (destinations);
- The number of vehicles in the fleet;
- The route allocation history (the routes must be allocated in an equally distributed manner for each month - TURNUS);
- The average distance travelled by vehicles in a month (this distance must be (optimally) equal for all the vehicles - AVERAGE DISTANCE);
- The driver's rest (of two days) between trips (REST).

Because of these constraints, new variables are necessary to be included in the algorithm for solving VRAP. Moreover a different approach is necessary regarding the development of the graph used to solve the problem. In figure 2 are depicted the graphs for solving TSP and VRAP accordingly.

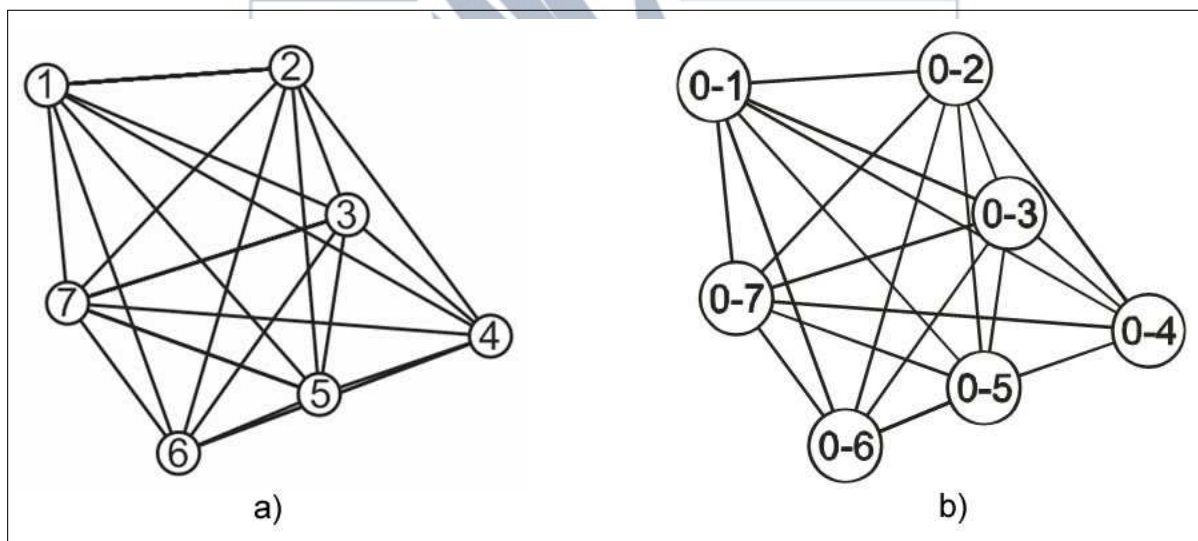


Figure 2: Comparison between the graphs for solving TSP (a) and VRAP (b).

If in the case of TSP the nodes in the graph are representing the cities that the salesman must reach, and the arches are standing for the roads (their cost being directly proportional to the distance between cities), in the case of VRAP the nodes (seen as pairs node0 and noden) are equivalent to a certain route (e.g. Bucharest - Paris) and the arches are representing the cost of allocating a vehicle to that route. Another approach is that the node0 is a unique virtual node regardless of the name of the city that corresponds to the departure station (e.g. node0 can be Bucharest, Paris, etc.).

The cost function must include the following factors:

- The vehicle history (the cost is greater if the vehicle departed from node0 and arrived at noden more recently and smaller otherwise);
- The average distance ran by a vehicle in a month (the cost is directly proportional with the difference between the average distance and the actual distance).
- The rest (of two days) of the drivers (the cost is directly proportional with the difference between the required resting days and the actual resting days).

### 3 The resulted algorithm

The algorithm for solving VRAP by using the modified Elitist Ant System is presented in figure 3.

```

// Variables used
//  $T_{max}$  - maximum numbers of iterations
//  $R$  - number of routes
//  $V$  - number of vehicles
//  $H$  - routes history matrix
//  $D_{ij}$  - distance between cities  $i$  and  $j$ 
//  $P_{ij}$  - pheromone intensity between nodes  $i$  and  $j$ 
//  $a$  - number of ants used to solve the problem ( $a$  is also equal with the number of routes)
//  $T_{kr}(t)$  - the matrix tour of ant  $k$  at the iteration  $t$ 
//  $C$  - the vector containing ant tours costs
//  $T^+$  - vector containing the best tour
//  $C^+$  - the cost of the best tour
//  $\alpha \simeq 0.7$  - how much the ants takes in consideration the history
//  $\beta \simeq 0.3$  - how much the ants takes in consideration the pheromone intensity
//  $\gamma \simeq 0.8$  - how much the ants takes in consideration the routes distances
//  $\rho \simeq 0.5$  - percent of the pheromone that evaporates
//  $\varepsilon \simeq 0.5$  - the multiplication value for the pheromone intensity of the elitist ant
//  $p_0$  - minimum pheromone intensity on arches =  $10^{-6}$ 

// Initialize pheromone trails
for (every edge  $i, j$ ) {
     $P_{ij} = p_0$ 
}
// Initialize the best tour and best tour cost
 $T^+$  = the best tour found from the beginning
 $C^+$  = the length of the best tour

// Main loop
for ( $t = 1; t \leq T_{max}; t++$ ) {
    // For every vehicle (every ant colony)
    for ( $v = 1; v \leq V; v++$ ) {
        // Choose the starting town for every ant
        for ( $k = 1; k \leq a; k++$ ) {
            Place ant  $k$  on node zero (start node)
        }
        // Compute a tour for every ant
        for ( $k = 1; k \leq a; k++$ ) {
            Build tour  $T_k^r(t)$  by applying  $R - 1$  times the following step:
            Choose the next node  $i$  in witch the ant  $k$  will go
            with the probability
            
$$p_{vi}^k(t) = \frac{[H_{vi}(t)]^\alpha \cdot [P_{vi}(t)]^\beta \cdot [V_{vi}]^\gamma}{\sum_{v \in T_k^r(t)} [H_{vi}(t)]^\alpha \cdot [P_{vi}(t)]^\beta \cdot [V_{vi}]^\gamma}, \text{ if } i \notin T_k^r(t)$$

            
$$p_{vi}^k(t) = 0, \text{ if } i \in T_k^r(t),$$

            where  $v$  is the current vehicle
        }
    }
}

```

```

// Compute the tour costs for all ants
for ( $k = 1; k \leq a; k++$ ) {
    Compute the cost  $C_k(t)$  of the tour  $T_k^r(t)$  produced by ant  $k$ 
}
// Update the best tour if an improved tour is found
if (an improved tour is found) {
    Update  $T^+$  and  $C^+$ 
    Print  $T^+$  and  $C^+$ 
}
// Global update for the pheromone trails
for (every edge  $i, j$ ) {
    Update the pheromone trails by applying the rule:
     $P_{ij}(t) = (1 - \rho) \cdot P_{ij}(t) + \Delta P_{ij}(t) + \varepsilon \cdot P_{ij}^e(t)$ 
}
// Calculate the intensity of the pheromone for next iteration
for (every edge  $i, j$ ) {
     $P_{ij}(t+1) = P_{ij}(t)$ 
}
}

```

Figure 3: Elitist Ant System pseudocode for solving the Route Allocation Problem.

The  $\alpha$ ,  $\beta$  and  $\gamma$  parameters are controlling the relative weight of the *history*, *pheromone intensity* and the *distance* in the process of choosing the next node in the itinerary. The formula for computing the pheromone intensity between nodes  $i$  and  $j$  at the  $t$  moment is:

$$P_{ij}(t) = (1 - \rho) \cdot P_{ij}(t) + \Delta P_{ij}(t) + \varepsilon \cdot P_{ij}^e(t), \text{ where}$$

- $(1 - \rho) \cdot P_{ij}(t)$  - the pheromone quantity that evaporates on the arch between nodes  $i$  and  $j$  at the moment  $t$ ;
- $\Delta P_{ij}(t) = \sum_{k=1}^a \Delta P_{ij}^k(t)$  - the total pheromone quantity that is deposited on the arch between nodes  $i$  and  $j$  at the moment  $t$  as sum of pheromone quantity deposited by all the ants that have used this arch;
- $\Delta P_{ij}^k(t) = \begin{cases} \frac{1-H_i^k}{0, otherwise} \cdot \mathbb{1}_{(i,j) \in T_k^i(t)} \end{cases}$  - the pheromone quantity deposited by ant  $k$  on the arch between the nodes  $i$  and  $j$  if that arch belongs to its itinerary ( $T_{ki}(t)$ ) at the moment  $t$ . This quantity is inversely proportional with the history of the vehicle on the route associated to the node that points to that arch;
- $\Delta P_{ij}^e(t) = \begin{cases} \frac{1-H_i^k}{0, otherwise} \cdot \mathbb{1}_{(i,j) \in T^+(t)} \end{cases}$  - the pheromone quantity deposited by the *elitist* ant  $k$  on the arch between nodes  $i$  and  $j$  if that arch belongs to its itinerary ( $T_{ki}(t) = T^+$ ) at the moment  $t$ . This quantity is inversely proportional with the history of the vehicle on the route associated to the node that points to that arch.

This process repeats until a certain condition is met (number of iterations, time, the finding of an acceptable solution, etc.).

## 4 Conclusions and intentions

The self-organisation (as in the case of ant colonies) indirectly supports the quality of the allocation service through the lack of the distortions or the deficiencies that may appear in a centralized coordination. The resulting implemented algorithm is robust (both reliable and error tolerant). The optimisation algorithm (Elitist Ant System) proved to be perfectly adaptable to the *Vehicle Route Allocation Problem*. The ant colony algorithms are showing an obvious potential as problem-solving tool, at least in the context of simple MAS, and relevant results can be achieved on affordable hardware configurations.

The *short-range* intentions are:

- Improving the cost formula;
- Finishing the research regarding the control of pheromones and search paths;
- Speeding up the search by fine-tuning the algorithm.

The *middle-range* target is to apply the optimisation in a real-time manner by modifying the algorithm.

## 5 Acknowledgements

This research is supported by the Ceex 116 / 2006 project: eTransMobility - Agent Oriented System for Modelling and Optimizing the Transport of Persons; coordinated by "Lucian Blaga" University of Sibiu.

## Bibliography

- [1] Bărbat B.E, Moiceanu A., Pleșca S., Negulescu S.C. (2007). Affordability and Paradigms in Agent-Based Systems. *Computer Sc. J. of Moldova*, 15, 2(44), pp.178-201.
- [2] Bărbat B.E., Negulescu S.C. (2006). From Algorithms to (Sub-)Symbolic Inferences in Multi-Agent Systems. *International Journal of Computers, Communications and Control*, 1, 3, pp.5-12.
- [3] Bărbat B.E., Negulescu S.C., Zamfirescu C.B. (2005). Human-Driven Stigmergic Control. Moving the Threshold. In N. Simonov (Ed.), *Proc. of the 17th IMACS World Congress (Scientific Computation, Applied Mathematics and Simulation)*, pp.86-92. Paris: ISBN 2- 915913-02-01.
- [4] Bonabeau E., Dorigo M., Theraulaz G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. New York: Oxford University Press.
- [5] Dorigo M., Maniezzo V., Colorni A. (1996). The Ant System: Optimisation by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 26 (1), pp.29-42.
- [6] Gambardella L.M., Di Caro G. (2005). The Ant Colony Optimization (ACO) Metaheuristic: a Swarm Intelligence Framework for Complex Optimization Tasks. Retrived 2008, from University of Bologna: First Summer School on Aspects of Complexity. Web site: [http://www.cs.unibo.it/~fioretti/AC/AC2005/docs/slides\\_dicaro.pdf](http://www.cs.unibo.it/~fioretti/AC/AC2005/docs/slides_dicaro.pdf)
- [7] Gambardella L.M., Dorigo M. (1995). Ant-Q: A Reinforcement Learning Approach to the Traveling Salesman Problem. In A. Prieditis and S. Russell (Ed.), *Proceedings of the Eleventh International Conference on Machine Learning* (pp.252-260). San Francisco, CA: Morgan Kaufmann.

- [8] Negulescu S.C., Bărbat B.E. (2004). Enhancing the effectiveness of simple multi-agent systems through stigmergic coordination. In ICSC-NAISO (Ed.), *Fourth International ICSC Symposium on ENGINEERING OF INTELLIGENT SYSTEMS (EIS 2004)*, pp.149-156. Canada: ICSC-NAISO Academic Press.
- [9] Negulescu S.C., Zamfirescu C.B., Bărbat B.E. (2006). User-Driven Heuristics for nondeterministic problems. *Studies in Informatics and Control (Special issue dedicated to the 2nd Romanian-Hungarian Joint Symp. on Applied Computational Intelligence)*, 15, 3, pp.289-296.
- [10] Stützle T., Hoos H.H. (2000). MAX-MIN Ant System. *Future Generation Computer Systems*, 16 (8), pp.889-914.



Sorin C. Negulescu  
"Lucian Blaga" University of Sibiu  
Faculty of Engineering  
4, Emil CIORAN, IM 502  
550025 Sibiu, Romania  
E-mail: sorin\_negulescu@yahoo.com

Claudiu V. Kifor  
"Lucian Blaga" University of Sibiu  
Faculty of Engineering  
4, Emil CIORAN, IM 502  
550025 Sibiu, Romania  
E-mail: claudiu.kifor@ulbsibiu.ro

Constantin Oprean  
"Lucian Blaga" University of Sibiu  
Faculty of Engineering  
4, Emil CIORAN, IM 502  
550025 Sibiu, Romania  
E-mail: constantin.oprean@ulbsibiu.ro