Proceedings of the XXVI Iberian Latin-American Congress on Computational Methods in Engineering CILAMCE 2005
Brazilian Assoc. for Comp. Mechanics & Latin American Assoc. of Comp. Methods in Engineering
Guarapari, Espírito Santo, Brazil, 19^{th}–21^{st} October 2005

Paper CL28-0880

AN ANT COLONY OPTIMIZATION SYSTEM FOR THE CAPACITATED VEHICLE ROUTING PROBLEM

Heitor S. Lopes
Vilson L. Dalle Molle
Carlos R. Erig Lima
hslopes@cpgei.cefetpr.br
vilson@md.cefetpr.br
erig@cefetpr.br
Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial,
Centro Federal de Educação Tecnológica do Paraná
Av. 7 de setembro, 3165, 80230-901, Curitiba (PR) – Brazil

Abstract. *Amongst the many problems in logistics, there is the Capacitated Vehicle Routing Problem – CVRP. This problem is present in many daily activities such as garbage collection, mail delivery, school bus transportation. CVRP includes not only the optimization of a path, but many of them simultaneously, since a fleet of evenly-capacitated vehicles have to deliver goods to geographically-distributed customers with variable demand, travelling the least distance as possible. This paper presents the first results of an ongoing project, related to the implementation of an Ant Colony Algorithm for the CVRP. This heuristic method is inspired in the behavior of real ants in the search for food. We devised a two-level optimization scheme so as to accomplish both global and local optimization. We applied the system to several benchmark instances of CVRP and the results obtained so far were very promising. Further work will include more tests to find optimized running parameters as well as experiments with other instances.*

Keywords: *Ant Colony Optimization, Vehicle Routing, Travelling Salesman Problem, Evolutionary Computation, Logistics.*

1. INTRODUCTION

In both Engineering and Computer Science, there are many problems for which there is no deterministic algorithm to find a solution or, when it is available, it is computationally inviable. Hence, assuming that an approximated solution is acceptable (according to a given satisfiability criteria), a wide range of possibilities is opened, based on new paradigms. The quest for computational methods that are, at the same time, simple (easy implementation), robust (good performance for many different instances of a problem), flexible (useful for many different problems) and efficient (capable of finding satisfactory solutions) have gained focus in the research community related to engineering, operations research and computer science. Consequently, many heuristic methods of optimization have appeared and gained widespread usage. Amongst them, there are those based on Evolutionary Computation (Bäck et al., 2000) and Swarm Intelligence (Bonabeau et al., 1999).

Around a decade ago, a new heuristic method for search and optimization has emerged. This method was created by Dorigo and coleagues (Dorigo and Gambardella, 1997;

Dorigo and Stützle, 2004) and is based on an analogy with the way real ants establish a trail between the nest and a food source. This method, known as Ant Colony Optimization (ACO), belongs to a group of heuristic techniques collectively known as Swarm Intelligence. ACO has been successfully applied to a number of complex real-world problems, such as: data mining (Parpinelli et al., 2002; Tsai et al., 2004), bioinformatics (Perretto and Lopes, 2005), combinatorial optimization (Bu et al., 2004) and several problems in logistics (Silva et al., 2003; Dorigo and Stützle, 2004).

Amongst the many problems in logistics, there is the multiple vehicle routing problem. A particular case of such problem is when vehicles have a limited capacity (Toth and Vigo, 2001). This problem is present in real-world organizations, such as hauliers and goods distributors. There are many daily tasks that can be modelled as this kind of problem, for instance: trash collection, school transportation and mail home delivery. This problem includes not only the optimization of one path, but many of them at the same time. In general, the task begins with a set of stop point that must be visited (consumers). By using some methodology, such set of points is divided into subsets, each one corresponding to a complete path. Next, the set of points for each path have to be ordered in such a way that the total travelled distance is minimized. The final objective is to minimize the sum of all travelled distances over all paths.

2. ANT COLONY OPTIMIZATION

The ACO (Dorigo and Stützle, 2004) heuristics is a distributed and cooperative search method that imitates the behavior of real ants in its the search for food. The observation of such behavior inspired the development of this optimization algorithm.

Basically, the ACO replicates the way ants promptly establish the shortest path between the nest and a food source (Bonabeau et al., 1999). Ants begin the search for food randomly around the nest. As they move, a chemical compound, named pheromone, is dropped to the ground over the path. The deposit of pheromone creates a trail and serve as an indirect way of communication between ants. Such mechanism is known as stigmergy. When a food source is found, the ant takes the food and returns back to the nest, leaving pheromone in the return path. Other ants that are randomly walking are attracted by this pheromone trail in the proportion of the amount deposited (and not evaporated yet). By following the trail, ants deposit even more pheromone. The larger the amount of pheromone, the larger the probability of the trail to be found (and followed) by other ants. However, there is also a small probability of an ant doesn't follow an attractive pheromone trail along all its extension. A small detour may cause a new trail that can (or cannot) be shorter than the former. Another ants can detect this new (shorter) trail and follow it. Pheromone continuously evaporates. Once the pheromone level falls down under a threshold, ants are not influenced to follow the trail. Therefore, if a trail is not used for a time, it tends to disappear. This autocatalytic phenomenon of positive feedback is the key point for establishing the shortest path from nest to food source (Beckers et al., 1992).

Based on this behavioral mechanism, the ACO (Dorigo and Stützle, 2004) was developed to solve combinatorial problems, making artificial ants search the shortest path (smaller cost) in the search space of problem's solutions. An adaptive memory simulates the pheromone trails by means of graphs, and a fitness function measures the quality of the solutions found as an analogy to the distance between food source and nest.

3. THE TRAVELLING SALESMAN PROBLEM AND VEHICLE ROUTING

The Travelling Salesman Problem (TSP) is a classical problem in Computer Science/Operations Research, and can be stated as follows. Given a set of fully interconnected cities, the traveller must visit all of them and return to the starting point, visiting any city only once and making the tour the smallest as possible. Figure 1 presents a simple instance of TSP with five cities (ABCDE), represented by a graph $G(V, E)$, in which each vertex (or node) $v_i \in V$ represents a city, and each edge (or arc) $e_{ij} \in E$ represents a path between a pair of cities. In this figure, a possible tour would be $ABCDEA$, at a cost of 12 length unities. Such cost is the sum of the weights of the edges of the corresponding graph of the tour. Notice that it is possible to find another tour of smaller cost, for instance, $ABCEDA$ with cost 11.

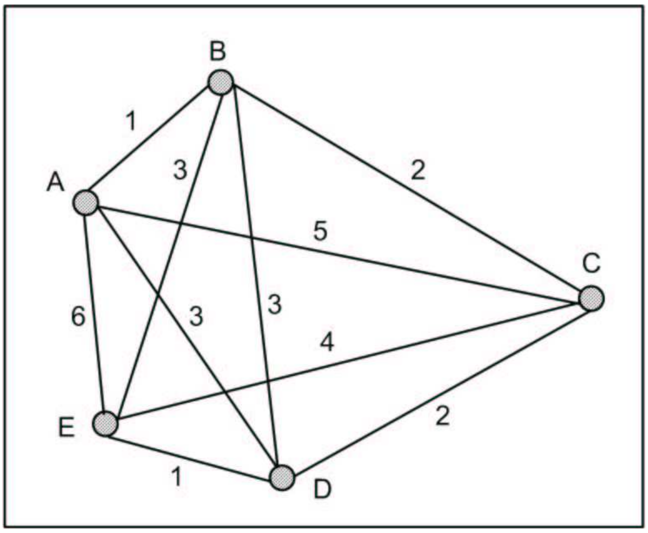


Figure 1: An example of symmetric TSP with five interconnected cities.

The problem of vehicle routing is a central issue in distribution logistics. In particular, the Capacitated Vehicle Routing Problem – CVRP can be defined as follows. Departing from a central depot (for some instances there are several depots), a number of consumers must be served with different demands of a single product, by means of a fleet of vehicles of finite and equal capacity. It is aimed to find the set of tours of minimal cost (travelled distance) that satisfies demands of all customers (Toth and Vigo, 2001). The basic constraints in this problem are:

- each customer is served only once by only one vehicle;
- the total demand covered by a vehicle cannot exceed its capacity;
- all individual tours begin and ends at the central point (depot);
- the total distance travelled in a tour cannot exceed the limited vehicle's autonomy.

Figure 2 shows an example of CVRP with a central depot and 10 customers. In this example, to supply demand, three tours are enough (dashed lines). Tours depart from depot, visit some cities and return to the depot, with the smallest possible distance travelled.

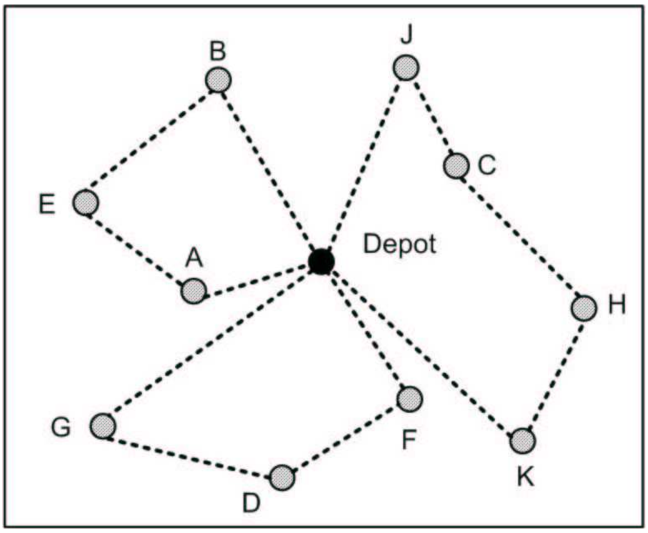


Figure 2: An example of CVRP with a central depot, 10 consumers and three tours.

The CVRP is NP-hard (Lenstra and Rinnooy Kan, 1981), since it contains one or more TSP as subproblems. Obviously, a CVRP is more difficult to solve than a TSP, because it requires a two-level solution. First, it is necessary to find which consumers will be aggregated in tours (without knowing *a priori* the minimum number of tours). Second, for each tour, it is necessary to find the permutation of customers that will represent the shortest path. Generally speaking, the first level can be understood as a bin-packing problem, while the second, as a TSP.

Formally, the CVRP can be defined as follows. Given a directed graph $G(V, E)$, where V is the set of vertices $V = \{v_0, v_1, \ldots, v_n\}$ and E is the set of edges $E = \{e_{ij}\}$, where $e_{ij} = edge(v_i, v_j) \forall v_i, v_j, i \neq j$. Considering the depot located at v_0, we have $V^{'} = V - \{v_0\}$ as the set of destinations (customers). To each vertex $v_i \in V^{'}$, an associated value $d_i \in \Re$ represents its demand. For each edge $e_{ij} \in E$ there is an associated cost c_{ij} corresponding to the distance between vertices v_i and v_j. For a given vehicle X_i with load capacity C^x, vector R^{X_i} contains the set of segments compounding the tour of this vehicle – that is, the set of customers served by vehicle X_i. The solution for the CVRP is the set of vectors R^{X_i} for which the sum of costs of the tours is minimal.

4. METHODOLOGY

The proposed algorithm divides the CVRP in two levels. At the first level, ants search for a set of tours. Each ant works as if it was the whole fleet. That is, each ant establishes several independent tours, and the set of tours cover all consumers. At the second level, each tour is submitted to a new population of ants which, in turn, runs the original ACO algorithm for the TSP (Dorigo and Gambardella, 1997). At this level, the algorithm is run for a fixed number of cycles (in our experiments, 20 cycles), aiming at a local optimization of the tour. The two-level process is repeated until a stopping criterion is met, in our case, a fixed number of optimization cycles. In the final cycle, the best ever solution found is submitted to 100 iterations of the ACO/TSP, for a final fine-tuning of the solution.

Since the second level is reduced to a TSP for each isolated tour, from the conceptual point of view, this methodology balances global exploration (distribution of consumers in tours, satisfaction of constraints) and local exploitation (minimizing tours). This two-level strategy is consistent with the way the CVRP is described. In this sense, our work is similar to the approach of (Bullnheimer et al., 1997), but they differ in the way trails are created by ants. In the same way, our work is different from (**?**)bell2004) in the way the ACO is structured to deal with the problem. Also, (Gambardella et al., 1999) have developed a multiple ant colony system for a more complex version of the CVRP. Their approach used two ant colonies to optimize a multiple objective function: the first colony minimizes the number of vehicles while the second colony minimizes the travelled distances.

4.1 Trail and tour construction

Following the basic idea of the ACO paradigm (see section 2.), ants construct pheromone trails iteratively, first constructing valid tours and, then, complete solutions. As soon as an ant obtains a solution (set of tours), a given amount of pheromone is granted to this solution. This amount of pheromone (τ) is inversely proportional to the cost of the complete solution (S), according to equation 1:

$$\tau(S) = \frac{u}{c(S)} \tag{1}$$

where: u is the unit amount of pheromone to be deposited on a trail, and $c(S)$ is the cost of the solution S, given by the sum of the Euclidean distances between points of the r permutations (P) that defines the set of tours of the solution.

For the i-th tour of the solution ($i = 1..r$), the amount of pheromone deposited is defined as a product of two terms: the proportion of pheromone by the cost of the tour, and the proportion of demand serviced by this specific tour, according to equation 2:

$$\tau(T_i) = \frac{\tau(S).D}{c(T_i)} \tag{2}$$

where: $\tau(T_i)$ is the amount of pheromone deposited throughout the tour T_i; $c(T_i)$ is the cost of tour T_i; and D is the demand serviced, proportional to the total demand. Therefore, the pheromone deposited is inversely proportional to its distance and directly proportional to the demand serviced. Consequently, for each pair of nodes (i, j) of the graph, there will be an associated amount of pheromone τ_{ij} contributed by all ants of the colony.

Time is discrete and corresponds to the steps of ants. Ants move collectively on the tour paths and, after each iteration, the network of tours is updated accordingly. Pheromone evaporates as a function of time, according to an user-defined parameter. Therefore, at the end of a cycle, pheromone trails are updated according to the previous equations and decremented according to the pheromone evaporation rate. Another user-defined parameter, the residual pheromone, defines minimum value of pheromone to be perceived by an ant. Once the pheromone level falls down under such threshold, the trail does not attract ants anymore.

In the first cycle, when there is no pheromone trail, ants choose the next step (customer) at random. In the remaining cycles, once an ant is at a given node of the graph, it chooses another unvisited node taking into account both the accumulated experience (given by the pheromone trail, τ) and an heuristic term depending on the cost of the tour. This approach follows the method proposed by (Perretto and Lopes, 2005). When ant k is at node i, the probability that node j is chosen to be added to the current partial tour is given by equation 3:

$$p_{ij}^{k} = \frac{(\tau_{ij})^{alpha}.(1/c_{ij})^{beta}}{\sum_{i,j \in T_i} \{(\tau_{ij})^{alpha}.(1/c_{ij})^{beta}\}} \tag{3}$$

where: τ_{ij} is the amount of pheromone in the edge between nodes i and j, T_i is the i-th tour, c_{ij} is the cost of edge ij, $alpha$ and $beta$ are user-defined constants.

Only unvisited nodes are considered for the next step of an ant, thus avoiding the construction of invalid tours. Parameters $alpha$ and $beta$ of equation 3 follows the terminology proposed by (Dorigo and Stützle, 2004), and control the amount of global and local exploration, respectively. In words, these parameters weights the probability of an ant to choose the next customer, when at a given point of the tour, considering the amount of pheromone already in the tour ($alpha$) or the inverse of the distance to be travelled to reach the customer ($beta$).

The capacity of the vehicle is not checked when the tour is being constructed. This may lead to unfeasible solutions. However, due to the nature of the ACO algorithm, a given unfeasible solution can evolve to a feasible solution in few cycles, as the pheromone trails are updated.

4.2 Implementation

The system was developed using Java and, therefore, it is compatible with many different platforms. The system accepts as input, text files in the format usually available in benchmarks. Such files have information about the instance (e.g., dimension, maximum number of vehicles, vehicle's capacity, etc), a section with the Euclidean coordinates of consumers, a section with the consumers' demands and a section with the depot location (the current version of the system works with single-depot instances).

An user-friendly interface shows a dynamic map with customers, depot and current tours created by ants, as shown in figure 3. This interface allows user to follow the evolution of the solution, as well as to interrupt the run and change control parameters. User can set the following control parameters of the ACO: pheromone evaporation rate, maximum number of unity pheromone deposited in a given tour, percent of residual pheromone, $alpha$ and $beta$, number of ants, and drawing scale. User can also access the name of the file under execution and the total cost of the best solution found up to the current iteration.

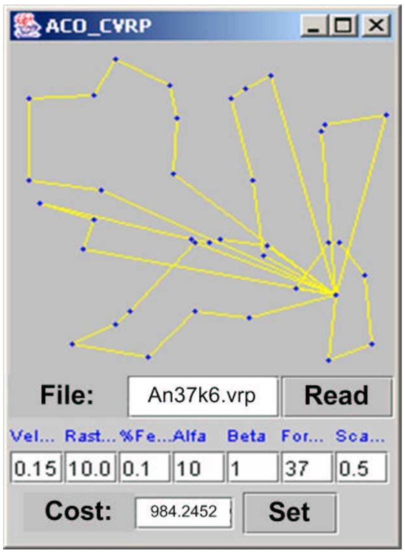


Figure 3: User interface of the developed ACO-CVRP system.

5. COMPUTATIONAL EXPERIMENTS AND RESULTS

To evaluate the proposed system, several instances of CVRP were used. Such instances were previously used as benchmark for other published works and are available in the Internet[1]. Table 1 shows details of the instances used. For each instance, the number after "n" represents the problem dimension (the total number of customers to be served), and the number after "k" represents the maximum number of vehicles to be used for serving customers. The initial letter relates to the authors who first introduced these benchmarks, respectively, "A", "C" and "F", for (Augerat et al., 1998), (Christofides and Eilon, 1969) and (Fisher, 1994).

For each instance, many experiments were run, so as to evaluate the performance of the system under different sets of control parameters. For the results reported here, the following parameters were used for all instances: number of cycles: 500, $alpha$=10, $beta$=1, number of pheromone unities dropped: 10. Other parameters were changed in the following ranges: pheromone evaporation rate: 0.15-2.0, number of ants: 40-250.

It is worth to note that parameters $alpha$ and $beta$ were adjusted empirically in such a way to favour global exploration relative to local exploitation, in a 10:1 relation. When the algorithm was run with small values for $alpha$ (relative to $beta$), we observed that convergence was too fast and towards low quality solutions. This phenomenon occurred even if we compensate with a decrement in the unity amount of pheromone deposited or even with an increment in the evaporation rate. Possibly, the reason for this behavior is in the way the system was conceived, in two levels of optimization.

In table 1, we show the best results found by our system, as well as the best known solution. In this table, column "Difference" represents the relation (in percent) between the best solution found and the best known solution.

[1] *http://elib.zib.de/pub/Packages/mp-testdata/tsp/tsplib/vrp* and *http://neo.lcc.uma.es/radi-aeb/WebVRP/data/instances/Augerat*

Table 1: Test instances used in the experiments and results obtained by the proposed ACO.

Instance	Optimum	Best result	Difference (%)
A-n32k5	784	798.29	1.82
A-n37k6	949	984.24	3.71
A-n53k7	1010	1043.13	3.28
A-n60k9	1408	1420.43	0.88
A-n80k10	1764	1886.13	6.92
E-n101k8	825	899.13	8.98
F-n135k7	1165	1234.42	5.96

The best solutions found for two instances are shown in figures 4 and 5. These instances are those for which our system obtained the best (0.88% above the optimum) and the worst (8.98% above the optimum) performance. The remaining solutions found, as well as the visual comparison between solutions found and the optima, are not shown here due to space limitations.

Figure 6 shows the evolution curve of the algorithm for two instances, A-n60k9 and E-n101k8. It can be observed that the convergence is fast in the beginning of the run and, later, it slows-down, reflecting some stagnation. Even so, some improvement can be obtained by means of the local search performed at the second level. The reason for the fast convergence is that, in the beginning of the process, there is a high probability of finding a shorter tour (than the one currently found) by an ant, since there are many alternative tours to be explored. Therefore, even considering the stochastic behavior of the algorithm, convergence will takes place fast in the first cycles. As the iterations increase, and the current “best” solution is improved (towards the global optimum), and the number of changes in the tour that can lead to some improvement decreases. Consequently, the probability that a random choice of an ant takes to some improvement, reduces very fast.

The use of ACO to optimize the TSP level, that is, the individual tours of each vehicle, promoted a small, although important, improvement. Although during evolution it is not possible to distinguish clearly the effect of such improvement, we believe that this procedure was essential to allow the algorithm to escape from most local optima. Also, at the end of the run, one can see some small improvement – see, for instance the plot of figure 6. This suggests, at least, that local search can be useful for fine-tuning the best solution found during run.

6. CONCLUSIONS AND FUTURE WORK

This work presented the application of the Ant Colony Optimization heuristic to a relevant problem in logistics, the capacitated vehicle routing problem. The proposed approach divides the problem in two levels of optimization. First, ants search for interesting tours, by dividing the set of customers into groups. Next, each individual tour is optimized as a travelling salesman problem.

The system was applied to a benchmark of 7 CVRP instances, with 32 to 135 customers (n) and 5 to 10 vehicles (k). Results obtained were very promising for a heuristic

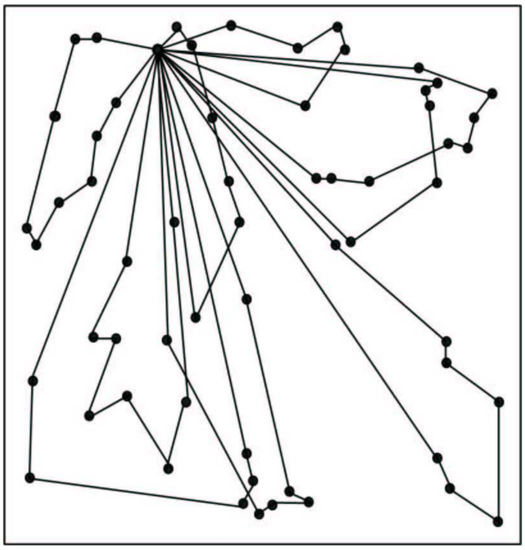


Figure 4: Best solution found for instance A-n60k9.

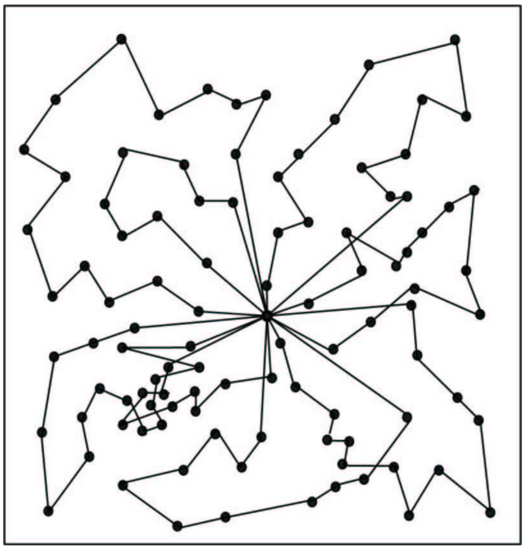
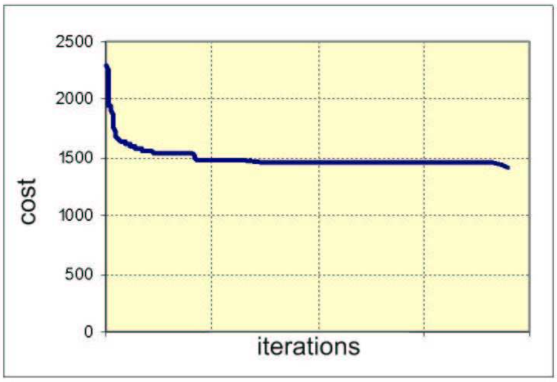


Figure 5: Best solution found for instance E-n101k8.



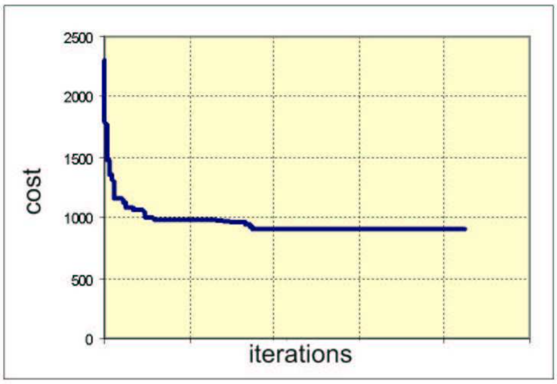


Figure 6: Evolution curve of the ACO for two CVRP instances: A-n60k9 (left) and E-n101k8 (right).

method, considering that no serious effort was done to optimize control parameters of the ACO. We observed that the control parameters have a very significant influence in the performance of the algorithm. Depending on how they are set, the algorithm can be taken to stagnation in few iterations. Future work will include a sensitivity analysis of the running parameters using more instances, so as to determine an optimized set of control parameters. Also, it would be interesting to investigate a self-adaptive strategy, similar to that of (Maruo et al., 2005), in order to release user of setting parameters in advance and adjusting them during the evolutionary search.

Based on the experiments reported here, it was not possible to identify a clear relationship between the size of an instance (that is, n and k) and the hardness for finding satisfactory solutions. However, as expected, the quality of the solutions found is inversely proportional to the size of the instance.

In general, results suggest that the proposed methodology is viable and efficient for tackling with the CVRP. This project shall be continued towards the improvement of the algorithm. Future research will include modifying the algorithm in such a way that each ant simulates a single vehicle. Hence, a population of k ants would represent the fleet of k vehicles. We suppose that with this approach the way cities are selected for the tours will be more efficient, and thus leading to better solutions. Another direction to be investigated could be to emphasize the first level of optimization with ACO and leaving the second level for an exhaustive search. This proposal is based on the fact that the individual tours are often small, thus justifying an enumerative method (at expense of a large computational cost) that guarantees the optimal tour. For the second level, methods such branch-and-bound could be also considered.

Acknowledgements

This work was partially supported by a research grant to H.S. Lopes from the Brazilian National Council of Research (CNPq) (process 305720/04-0).

REFERENCES

Augerat, P., Belenguer, J.M., Benavent, E. et al., 1998. Separating capacity constraints in the CVRP using tabu search., *European Journal of Operations Research*, vol. 106, n. 2–3, 1998, pp. 546–557.

Bäck, T., Fogel, D.B. & Michalewicz, Z., 2000. *Evolutionary Computation*, vols. 1 and 2. Institute of Physics Publishing, Bristol, UK.

Beckers, R., Deneubourg, J.L. & Goss, S., 1992. Trails and u-turns in the selection of the shortest path by the ant Lasius niger. *Journal of Theorethical Biology*, vol. 159, pp. 397–415.

Bell, J.E. & McMullen, P.R., 2004. Ant colony optimization techniques for the vehicle routing problem. *Advanced Engineering Informatics*, vol. 18, pp. 41-48.

Bodin, L., Golden, B., Assad, A. & Ball, M., 1983. Routing and scheduling of vehicles and crews – the state of art. *Computers & Operations Research*, vol. 10, n. 2, pp. 62–212.

Bonabeau, E., Dorigo, M. & Theraulaz, G., 1999 *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, Oxford, UK.

Bu, T.M., Yu, S.N. & Guan, H.W., 2004. Binary-coding-based ant colony optimization and its convergence. *Journal of Computer Science and Technology*, vol. 19, n. 4, pp. 472–478.

Bullnheimer, B., Hartl, R.F. & Strauss, C., 1997. Applying the ant system to the vehicle routing problem. In *Proc.* 2^{nd} *International Conference on Metaheuristics*. Sophia-Antipolis, France.

Christofides, N. & Eilon, S., 1969. An algorithm for the vehicle dispatching problem. *Operations Research Quartely*, vol. 20, pp. 309–318.

Dorigo, M. & Gambardella, L.C., 1997. Ant colonies for the travelling salesman problem. *Biosystems*, vol. 43, pp. 73–81.

Dorigo, M. & Stützle, T., 2004. *Ant Colony Optimization*. The MIT Press, Cambridge, MA.

Fisher, M.L., 1994. Optimal solution of vehicle routing problems using minimum K-trees, *Operations Research*, vol. 42, pp. 626–642.

Gambardella, L.M., Taillard E. & Agazzi,G., 1999. MACS-VRPTW: a multiple ant colony system for vehicle routing problems with time windows. In Corne, D., Dorigo, M. & Glover, F., eds., *New Ideas in Optimization*. pp. 63–76, McGraw-Hill, New-York, NJ.

Lenstr, J.K. & Rinnooy Kan, A.H.G., 1981. Complexity of vehicle routing and scheduling problems. *Networks*, vol. 11, pp. 221–227.

Maruo. M.H., Lopes, H.S. & Delgado, M.R.B.S., 2005. Self-adapting evolutionary parameters: encoding aspects for combinatorial optimization problems. In Raidl, G.R. & Gottlieb, J., eds., *Evolutionary Computation for Combinatorial Problems – LNCS,* vol. 3448, pp. 154-165, Springer-Verlag, Heidelberg.

Parpinelli, R.S., Lopes, H.S. & Freitas, A.A., 2002. Data mining with an ant colony algorithm. *IEEE Transactions on Evolutionary Computation*, vol. 6, n. 4, pp. 321–332.

Perretto M. & Lopes, H.S., 2005. Reconstruction of phylogenetic trees using the ant colony optimization paradigm. To appear in: *Genetics and Molecular Research*, 2005.

Silva, C.A., Runkler, T.A., Sousa, J.M. & Costa, J.M.S., 2003. Optimization of logistic processes in supply-chains using meta-heuristics. In *Progress in Artificial Intelligence – LNAI*, vol. 2902, pp. 9–23, Springer-Verlag, Heidelberg.

Toth, P. & Vigo, D., 2001. The vehicle routing problem. *SIAM Monographs on Discrete Mathematics and Applications*. SIAM Press, Philadelphia, PA.

Tsai, C.F., Tsai, C.W., Wu, H.C. & Yang, T., 2004. ACODF: a novel data clustering approach for data mining in large databases. *Journal of Systems Software*, vol. 73, n. 1, pp. 133–145.