Construction and Improvement Heuristics applied to the Capacitated Vehicle Routing Problem

Leonardo G. Tavares, Heitor S. Lopes, Carlos R. Erig Lima
Bioinformatics Laboratory
Federal University of Technology Parana (UTFPR)
Av. 7 de setembro, 3165 80230-901, Curitiba (PR), Brazil
leonardo.tavares@up.edu.br, hslopes@pesquisador.cnpq.br, erig@utfpr.edu.br

***Abstract*—The capacitated vehicle routing is a problem of combinatorial optimization that has aroused major interest because it is present in various areas (logistics, transport and other) and, it is a problem of considerable difficulty. There are currently various techniques that have been developed to try to solve this problem efficiently. In this article, we present a solution to this problem based on the strategy of combining different single techniques to obtain the best results. Computing experiments have been conducted on six instances of well-known data sets available in literature.**

***Keywords*-Heuristics, Capacitated Vehicle Routing Problem, Combinatorial optimization.**

I. Introduction

A class of problems has aroused major interest, on the part of reseedgehers, which are the problems known as non-deterministic polynomial time (NP). These are problems that, as the order of instances increases, do not present feasible solutions through exact algorithms, due to them being too time-consuming in their resolution.

One of these problems is the capacitated vehicle routing problem [13]. It is a problem of major interest to the scientific community because, besides dealing with a problem that is present in various situations, it is a combinatorial problem of considerable difficulty.

The problem consists in finding a set of lower cost paths for a certain fleet of vehicles, so that these meet the demand of all clients respecting the capacity of their vehicles.

The exact methods, in general, guarantee the optimal solution to the problem, however they resolve only smaller problems, which normally do not reflect the reality. As a result of this fact, little attention has been given to the seedgeh for optimal solutions. An alternative to the exact methods are the heuristic methods.

The objective of the heuristic methods is to find an approximate solution, according to some criterion of acceptance, so that the time spent in seedgehing for a solution is acceptable. Considerable attention has been given to this class of methods in the past few years.

II. Capacitated Vehicle Routing Problem

The Vehicle Routing Problem was introduced into literature by Dantzig and Ramser [4], to solve a problem of gasoline distribution to gas stations. The Vehicle Routing Problem (or, simply VRP) is, in fact, the generalization of a class of problems in which a number of routes for a fleet of vehicles should be determined to serve a number of scattered customers. Basically, we should find a number of paths with a minimal cost that meet the demand of all the clients [13].

The Capacitated Vehicle Routing Problem (or simply CVRP) is the simplest form of the VRP. In this form, a fleet of vehicles, located at a single depot, should meet the different demands of a group of consumers for the products to be distributed. Figure 5 shows a group of CVRP routes. The middle rectangle represents the depot and the smaller rectangles represent the consumers.

The basic restrictions of the CVRP are:

- all the routes start and end at the depot;
- each consumer is visited at least once;
- the total demand of any route should not exceed capacity Q of the respective vehicle involved.

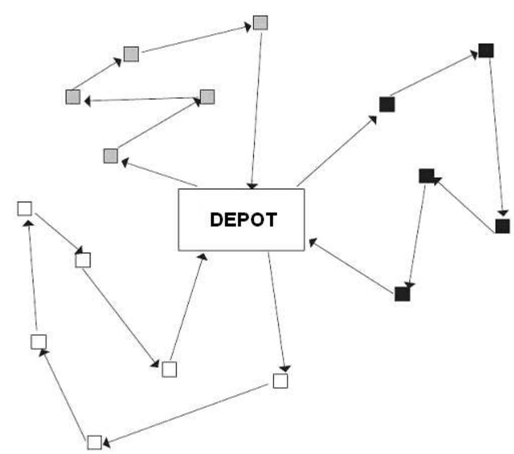

Figure 1. Example of a group of CVRP routes.

Formally, the CVRP can be described in the following way: let $G = (V, E)$ be a direct graph, with $V =$

$v_0, v_1, ..., v_n$ being a group of vertexes, that represent cities or consumers, and E is the set of edges $E = e_{ij}$, where $e_{ij} = edge(v_i, v_j) \forall v_i, v_j, i \neq j$ that link the vertexes (vi, vj). Vertex v_0 represents the depot, but, in some formulations, the depot can be represented by vertex $(n+1)$. A non-negative value c_{ij} is associated to each edge and represents the cost of travelling of consumer i to consumer j. Normally, the use of closed edges is not permitted.

Many approaches have been proposed to give approximate solutions to the CVRP in the last years. Clark and Wright [3] proposed one of the first heuristic methods to this problem. Fisher [5] used a technique known as minimum K-trees. Taillard [11] proposed parallel iterative methods. Toth and Vigo [12] and Xu and Kelly [14] used tabu seedgeh. Gambardella et al. [6] and Lopes et al. [9] proposed solutions using the ant colony optimization.

III. HEURISTICS

The heuristics implemented in this study can be divided into two types: Construction Heuristics and Improvement Heuristics (or of Refinement).

Construction Heuristics generate a solution incrementally, so that, in every iteration, a new element is selected to integrate the model in seedgeh of the solution to the problem.

In Improvement Heuristics, in each iteration, improvements are applied to a complete initial solution until a stopping criterion is met. In this case, the initial solution can be generated randomly or even through a more efficient construction heuristic. The stopping criterion can be based on the time of seedgeh, on the number of iterations or on the stagnation of the algorithm through a certain number of iterations.

The construction heuristic that was used in this study is known as the Nearest Neighbor Heuristic (NNH). To refine the solutions generated by the NNH, three improvement heuristics were used: relocation using the Cheapest Insertion, 2-opt Inter-Route and 2-opt Intra-Route.

A. Nearest Neighbor Heuristic

The Nearest Neighbor Heuristic is one of the most intuitive and best-known methods of seedgeh for the solution of the traveling salesman problem. The method consists of constructing a route from a randomly chosen point and, extending the route interactively by inserting a vertex that has not been visited, which is nearest to the current vertex, i.e. the nearest neighbor.

For the traveling salesman problem, where the lowest cost route is being sought, the heuristic finalizes the construction when there are no more vertexes to be included in the path. At this point, the last connected vertex is linked to the initial vertex closing the path, which is known as the nearest neighbor tour. It can be clearly observed that the quality of the solution reached depends on the choice of the initial vertex. However, for instances of the TSPLIB, this heuristic normally generates solutions of around 20-35% worst than the optimal solution [7].

For the vehicle routing problem, where you try to find the set of routes with the lowest cost, the heuristic is executed many times for the construction of *n* paths. In the specific case of the routing of limited capacity vehicles, the heuristic ends the construction of a path when the insertion of a new vertex in this path exceeds the capacity of the vehicle. The last vertex tested, and which was not inserted in the previous path due to the limited capacity of the vehicle, becomes the first vertex in the following path.

Figure 2 shows some examples of solutions generated through the Nearest Neighbor Heuristic.

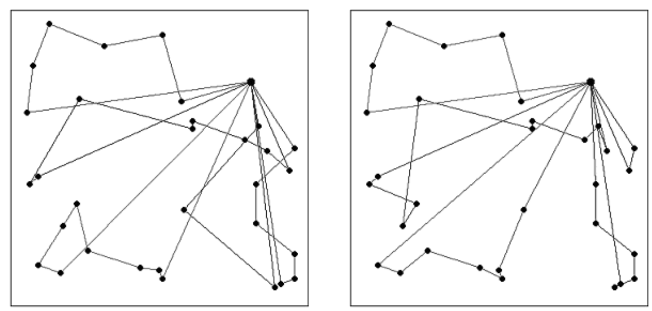

Figure 2. Examples of solutions generated through the NNH.

B. Relocation moves using the Cheapest Insertion Heuristic

A relocation movement consists in removing the consumer from any route and inserting him in another position, which can be either in the same route or in any other distinct route.

The criteria for the selection of the consumer to be removed and from the point in which he is inserted are diverse. In this study, the choice of the consumer is made randomly and the option for the point of his insertion is made using the heuristic of the Cheapest Insertion [10]. This heuristic seeks the point where the new insertion of the consumer has the least increment in the cost of the route. In our case, the relocation of a consumer is tested in all of the remaining routes according to the Cheapest Insertion and, then the movement that promotes the best increment in the quality of the solution is selected.

Figure 3 shows a relocation movement of a consumer using this heuristic. The consumer that was relocated is indicated by a circle in the figure.

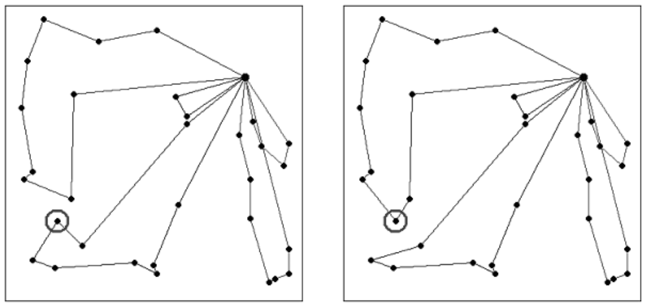

Figure 3. Relocation of a consumer using the Cheapest Insertion Heuristic.

C. 2-opt Intra-Route Heuristic

The 2-opt intra-route method for refinement was initially proposed for the traveling salesman problem by Lin and Kernighan [8]. The heuristic consists initially in the random choice of two non-consecutive edges, which belong to the same route, as shown in figure 4. The two edges selected are removed from the original path and their points are reconnected so that they create a new path.

If the cost of the new path is less than the cost of the original path, the new path is selected as the current solution. This concept is known as the first improvement, i.e. in the first movement in which the quality of the solution has improved, the new route is adopted as the current solution to the problem. The process ends when a certain stopping criterion is reached, for example, a number of cycles with no improvement in the current solution.

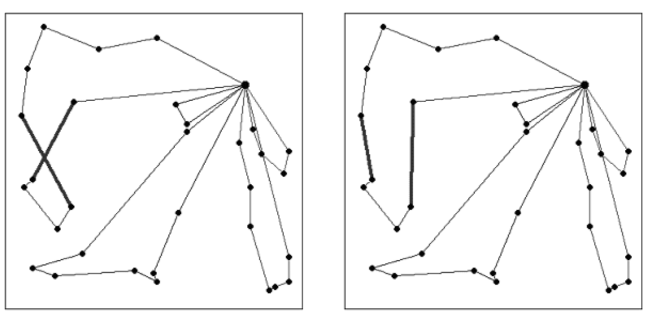

Figure 4. Movement of the 2-opt Intra-Route Heuristic.

D. 2-opt Inter-Route Heuristic

The 2-opt inter-route route method of refinement is, in fact, an expansion of the 2-opt intra-route heuristic. While in the 2-opt intra-route heuristic two edges belonging to the same route are selected randomly, in the 2-opt inter-route heuristic two edges that necessarily belong to distinct routes are selected randomly (figure 5).

The result of this exchange can lead to a route whose vehicle capacity is disrespected. In this case, the new solution is not considered.

The same first improvement concept can be adopted for the 2-opt inter-route heuristic.

A stopping criterion should be adopted to end the algorithm.

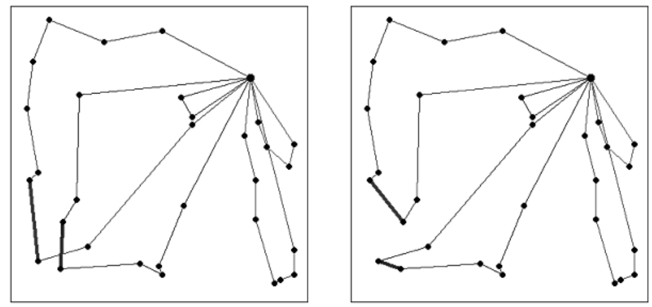

Figure 5. Movement of the 2-opt Inter-Route Heuristic.

IV. METHODOLOGY

In this study, a model was implemented with two distinct steps. In the first step, a constructive heuristic was created to generate an initial solution, and in the second step, heuristics were used to improve the initial solution. A certain predetermined criteria can make the algorithm return to the initial step, thus configuring, a restart of the seedgeh.

To generate an initial solution, the nearest neighbor heuristic is used, indicated in figure 6 by *NN* (Nearest Neighbor). Beginning with this initial solution, the control mechanism passes through the step of intensification, where the initial solution needs to be refined. This is done through the combination of three different heuristics: the relocation of consumers using the cheapest insertion heuristic, the 2-opt inter-route heuristic and the 2-opt intra-route heuristic. In figure 6, these heuristics are indicated by H1, H2 and H3.

The process may execute in cycles the three heuristics proposed until a certain stopping criterion is met, for example, a certain number of cycles with no improvement in the quality of the solution. If the step of intensification is concluded and the refined solution does not meet a predetermined evaluation criterion (indicated in the figure by *CA*), the control mechanism returns to the initial state and the process is restarted.

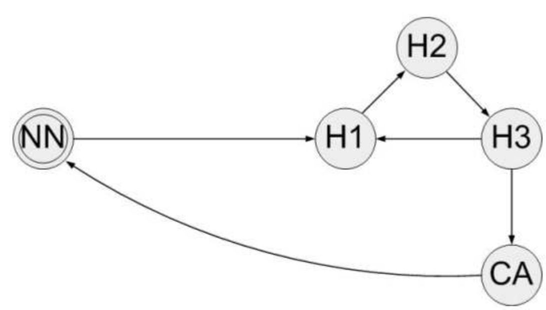

Figure 6. Control mechanism of the heuristics.

The application was created so that it accepts text files in the *.vrp format as input, containing the information of the instance like dimension, maximum number of vehicles, vehicle capacity, etc. The graphic interface (figure 7) shows a map of the consumers and depot (numbered according to the information of the vrp file) and the paths obtained as the solution to the problem. The following control parameters are available to the user:

- Number of vehicles: indicates the initial number of vehicles necessary to solve the problem.
- Number of executions: indicates the number of cycles that an algorithm should be executed.
- Criterion for restart: indicates the number of cycles with no improvement in the refinement step so that the algorithm can be restarted.

There are also checkboxes to select the heuristics that should be part of the refinement step of the solution (relocation with cheapest insertion, 2-opt inter-route and 2-opt intra-route heuristics). With this resource, it is possible to switch off some refinement heuristics in order to test the best configurations.

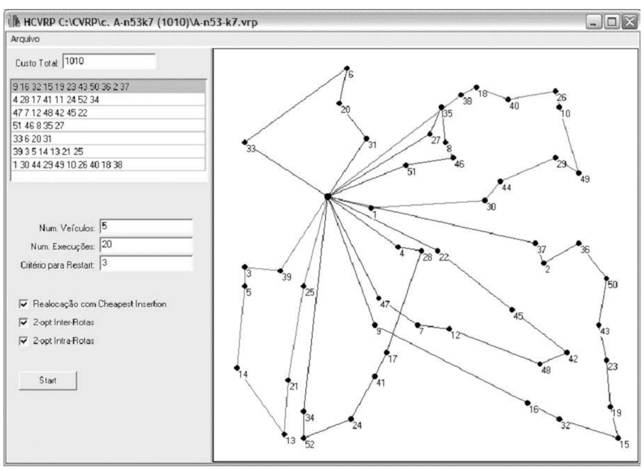

Figure 7. Graphic interface of the application.

V. Experiments and Results

To evaluate the concept proposed, various tests were conducted using benchmark data available on the Internet. Five instances of Augerat et al [1] were used, (A-n32k5, A-n37k6, A-n53k7, A-n60k9 and A-n80k10) and one of Christofides and Eilon [2] (E-n135k7). For each instance, the numbers after *n* and *k* mean the number of cities and number of vehicles, respectively.

For each one of the instances tested, various experiments were carried out, in order to test different configurations and values of the control parameters.

For the data reported in this study, the number of executions of the algorithm selected was 20, and the number of cycles with no improvement for the algorithm to restart the process was three. All configurations were tested for the experiments. In all the configurations, the initial solution was generated from the nearest neighbor heuristic. For the step of intensification, the following was adopted:

- Cfg 1 - only relocation with cheapest insertion.
- Cfg 2 - relocation with cheapest insertion + 2-opt inter-route heuristic.
- Cfg 3 - relocation with cheapest insertion + 2-opt inter-route heuristic + 2-opt intra-route heuristic.

The algorithm was executed 50 times for each configuration proposed. Table I shows the average results obtained in each configuration. The first column shows the name of the instance, the second column, the optimal known until now, and the remaining columns, the average results obtained.

For each configuration, the average value obtained and the relative value (as a percentage) are shown, between the value found using this method and the optimal known solution. Table I shows the averages of the relative values (as a percentage) in relation to the optimal known solution for each one of the instances tested.

Instance	Optimal	Cfg 1		Cfg 2		Cfg 3	
		Average	Dif.(%)	Average	Dif.(%)	Average	Dif.(%)
A-n32k5	784	848.34	8.21	831.86	6.10	795.84	1.51
A-n37k6	949	1022.84	7.78	1008.46	6.27	976.90	2.94
A-n53k7	1010	1151.72	14.03	1130.58	11.94	1079.20	6.85
A-n60k9	1408	1489.84	5.81	1475.18	4.77	1420.42	0.88
A-n80k10	1764	1964.40	11.36	1954.82	10.82	1871.88	6.12
E-n101k8	825	960.88	16.47	942.90	14.29	883.04	7.03

Table I
Average of the results in all 3 configurations.

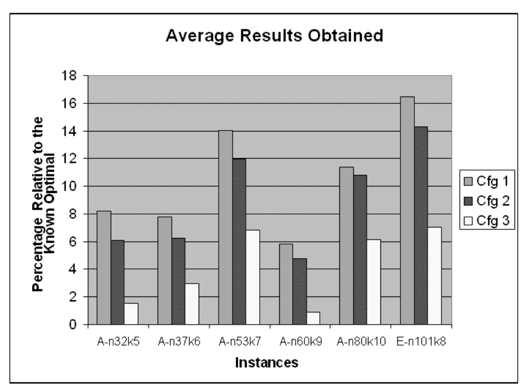

Figure 8. Average of the results in all 3 configurations.

Table II shows the best results obtained with each configuration. Similar to table I, the results are shown in absolute values and as percentages in relation to the optimal known solution. Figure 9 shows a graph with the best results obtained in relation to the optimal known solution for each of the instances tested.

Instance	Optimal	Cfg 1		Cfg 2		Cfg 3	
		Average	Dif.(%)	Average	Dif.(%)	Average	Dif.(%)
A-n32k5	784	797	1.66	793	1.15	784	0.00
A-n37k6	949	970	2.21	967	1.90	953	0.42
A-n53k7	1010	1093	8.22	1039	2.87	1024	1.39
A-n60k9	1408	1426	1.28	1426	1.28	1386	-1.56
A-n80k10	1764	1893	7.32	1907	8.11	1807	2.44
E-n101k8	825	895	8.48	897	8.73	835	1.21

Table II
BEST RESULTS FOUND IN ALL 3 CONFIGURATIONS.

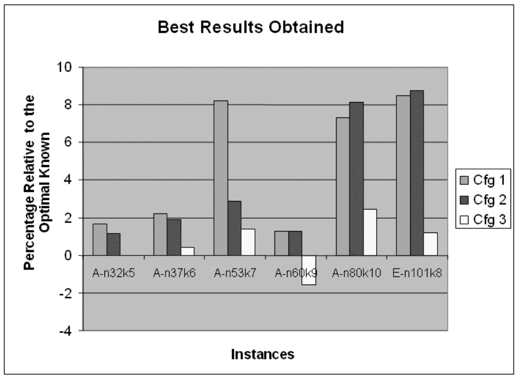

Figure 9. Best results found in all 3 configurations.

Table III shows the average time (in seconds) to reach the terminal criterion with each configuration. Similarly, figure 10 presents a graph with the same information.

Instance	Cfg 1 time (sec.)	Cfg 2 time (sec.)	Cfg 3 time (sec.)
A-n32k5	0.17	0.31	0.37
A-n37k6	0.25	0.52	0.56
A-n53k7	0.60	0.99	1.09
A-n60k9	0.86	1.31	1.39
A-n80k10	1.74	2.33	2.51
E-n101k8	2.88	3.54	4.04

Table III
AVERAGE TIME TO REACH THE TERMINAL CRITERION.

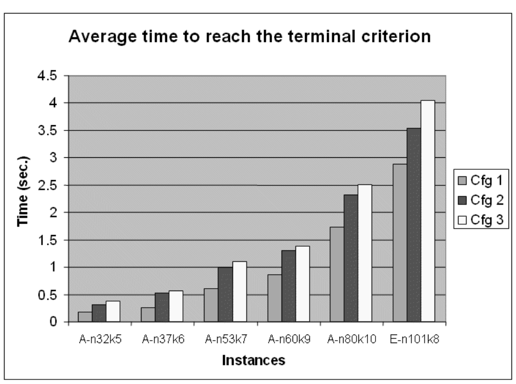

Figure 10. Average time to reach the terminal criterion.

Figures 11 and 12 present the best results for instances A-n80k10 and E-n101k8 graphically, for illustration purposes.

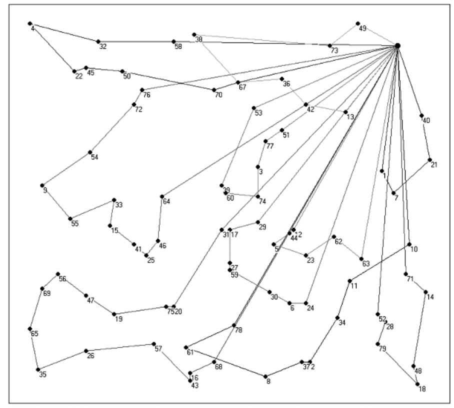

Figure 11. Best solution obtained for instance A-n80k10.

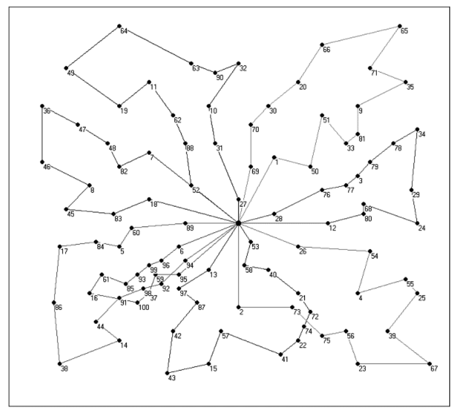

Figure 12. Best solution obtained for instance E-n101k8.

VI. CONCLUSIONS

The combination of different heuristic techniques has shown to be an interesting alternative in the solution of problems of great complexity. However, the correct choice of heuristics and mechanism control is a decisive factor to obtain good results.

As more heuristics were added in the step of intensification, the time of processing increased, but the average solution obtained increased in a much larger proportion. From the first to the last configuration, the time of execution increased 80% on average, while the quality of the solution improved, i.e., the proportion of the results obtained in relation to the optimal known solution, decreased on average 250%.

As a general result, configuration, Cfg 3 (combination of the four heuristics) managed to obtain the best results in relation to the other implementations. The strategy used has a greater time of execution in relation to the others, however, the results obtained were much superior.

The processing time to reach the terminal criterion increases as the order of instances increases. However, it is not possible to affirm if there is a direct relation between the difficulty of the problem and the order of the instance. It is possible to verify this fact by observing the average results obtained for instances A-n53k7 and A-n80k10. Despite the second presenting a larger number of cities and vehicles, the algorithm obtained, on average, better results than the first.

The method adopted in the present study seems quite promising even because a new optimal solution was found for one of the instances tested (A-n60k9). The optimal known solution until now was 1408, while in the present study the optimal value obtained was 1386.

An improvement that can be obtained in future studies is the implementation of other constructive heuristics in order to compare the efficiency of the nearest neighbor, adopted in this study. Another improvement that could be incorporated is the implementation of a more sophisticated metaheuristics (like the Simulated Annealing, Tabu Search, or any other) in hybrid form with the other improvement heuristics implemented in the step of intensification.

REFERENCES

[1] P. Augerat, J. Belenger, E. Benavent, "Separating capacity constraints in the CVRP using tabu seedgeh," *European Journal of Operations Reseedgeh*, vol. 106, pp. 546–557, 1998.

[2] N. Christofides, S. Eilon, "An algorithm for the vehicle-dispatching problem," *Operations Reseedgeh Quarterly*, vol. 20, pp. 309–318, 1969.

[3] G. Clark, J.W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," *Operations Reseedgeh*, vol. 12, pp. 568–581, 1964.

[4] G.B. Dantzig, J.H. Ramser, "The truck-dispatching problem," *Management Science*, vol. 6, pp. 80–91, 1959.

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

[6] L.M. Gambardella, E. Taillard, G. Agazzi, "MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows," In D. Corne, M. Dorigo and F. Glover, editors, *New Ideas in Optimization*, McGraw-Hill, London, UK, pp. 63–76, 1999.

[7] H.H. Hoos, T. Sttzle, "Stochastic Local Seedgeh: Foundations and Applications," *Morgan Kaufmann*, San Francisco, 2004.

[8] S. Lin, B.W. Kernighan, "An Effective Heuristic Algorithm for the Traveling Salesman Problem," *Operations Reseedgeh*, vol. 21, pp. 498–516, 1973.

[9] H.S. Lopes, V.L. Dalle Molle, C.R.E. Lima, "An ant colony optimization system for the capacitated vehicle routing problem," Proceedings of the XXVI Iberian Latin-American Congress on Computational Methods in Engineering - CILAMCE, Guarapari, 2005.

[10] D.H. Rosenkrantz, R.E. Stearns, P.M. Lewis, "An analysis of several heuristics for the traveling salesman problem," *SIAM Journal on Computing*, vol. 6, pp. 563–581, 1977.

[11] E.D. Taillard, "Parallel Iterative Seedgeh Methods for Vehicle Routing Problems," *Networks*, vol. 23, pp. 661–673, 1993.

[12] P. Toth, D. Vigo, "The Granular Tabu Seedgeh (and its Application to the Vehicle Routing Problem)," Technical Report, Dipartimento di Elettronica, Informatica e Sistemistica, Universit di Bologna, Italy, 1998.

[13] P. Toth, D. Vigo, "The Vehicle Routing Problem," SIAM Monographs on Discrete Mathematics and Applications, *SIAM Press*, Philadelphia, 2001.

[14] J. Xu, J. Kelly, "A Network Flow-Based Tabu Seedgeh Heuristic for the Vehicle Routing Problem," *Transportation Science*, vol. 30, pp. 379–393, 1996.