A Hybrid Algorithm for the Vehicle Routing Problem with Time Window

Guilherme Bastos Alvarenga, Ricardo Martins de Abreu Silva, Rudini Menezes Sampaio
Dept. of Computer Science, Federal University of Lavras, Brazil
{guilherme, rmas, rudini} @ dcc.ufla.br

Abstract. The Vehicle Routing Problem with Time Windows (VRPTW) is a well-known and complex combinatorial problem, which has received considerable attention in recent years. The VRPTW benchmark problems of Solomon (1987) have been most commonly chosen to evaluate and compare all exact and heuristic algorithms. A genetic algorithm and a set partitioning two phases approach has obtained competitive results in terms of total travel distance minimization. However, a great number of heuristics has used the number of vehicles as the first objective and travel distance as the second, subject to the first. This paper proposes a three phases approach considering both objectives. Initially, a hierarchical tournament selection genetic algorithm is applied. It can reach all best results in number of vehicles of the 56 Solomon's problems explored in the literature. After then, the two phase approach, the genetic and the set partitioning, is applied to minimize the travel distance as the second objective.

Keywords: vehicle routing, hybrid genetic algorithm, hierarchical tournament selection

(Received April, 07, 2005 / Accepted July 15, 2005)

1. Introduction

Vehicle routing problems (VRP) have received considerable attention in recent years. The usual static version, namely Vehicle Routing Problem with Time Windows (VRPTW) includes capacity and time window constraints. In the VRPTW, a fleet of identical vehicles supplies goods to N customers. All vehicles have the same capacity Q. For each customer i, $i = 1,\ldots,N$, the demand of goods, q_i, the service time s_i, and the time window $[a_i,b_i]$ to meet the demand in i are known. The component s_i represents the loading or unloading service time at the customer i, and a_i describes the earliest time when it is possible to start the service. If any vehicle arrives at customer i before a_i it must wait. The vehicle must start the customer service before b_i. All vehicle routes start and finish at the central depot. Each customer must be visited once. The locations of the central depot and all customers, the minimal distance d_{ij} and the travel time t_{ij} between all locations are given. Different objectives have been proposed in the literature, including the minimal total travel distance, the minimal number of vehicles used, the total wait time, the total time to complete the whole service and combination of these.

Alvarenga [1] and [11] has proposed a genetic and a set partitioning two phases approach for the VRPTW using travel distance as the unique objective. The tests were produced using both, real numbers and truncated data type, and the results were compared with previous published heuristic and exact methods. These results show that the heuristic CGH (Column Generation Heuristic) proposed is very competitive in terms of travel distance (TD) minimization. In present paper, an extension of that approach is proposed to minimize the number of vehicles (NV) first and the total travel distance is minimized as a second objective.

Berger [2] has improved some of the results of Solomon's benchmark using parallel two-population co-evolution genetic algorithms, Pop1 and Pop2. The first population, Pop1, has the objective to minimize TD to a fixed number of vehicles. On the other hand, Pop2 works to minimize the violated time window, in order to find at least one feasible individual. In Pop2, NV is fixed as the number obtained by Pop1 minus one. The global objective is to minimize NV, and after that to minimize TD, in a second priority. Pop1 works to minimize TD over the population received from Pop2, where at least one feasible solution is known. Each time a feasible individual is found the population Pop1 is substituted by Pop2 and the fixed number of vehicles considered in both populations is decreased by one.

Berger has also tested the algorithm in the 56 Solomon's instances. Berger found 6 new results (for instances R108, R110, RC105, RC106, R210 and R211). Currently, three of them continue to be the best solutions known (R108, RC105 and RC106), considering NV as the first objective and TD as the second. One of the most important advantages of his work is the total NV for all 56 instances of Solomon, with 405 vehicles, one of the best results in the literature.

Homberger [7] has also presented good results for many Solomon's benchmark problems using two evolutionary metaheuristic methods in a similar two-stage strategy. Two different heuristic methods were proposed, ES1 and ES2. Homberger emphasizes the importance of the evaluation criterion. The travel distance selection does not drive the search in the number of vehicles global minimum. Consequently, it is necessary to add new evaluation criteria. The first new criterion explored is the number of customers in the shortest route. The second criterion is namely the *minimal delay* D_R. That is the sum of the minimal time violations caused by the forced elimination of the customers from the shortest route R, equation (1).

$$D_R = \sum_{k \in R} D_k \tag{1}$$

Where

- $D_k = 0$ if the customer k can be eliminated and inserted in any other route $R' \neq R$;
- $D_k = \infty$ if the insertion of k always results in a capacity violation to any route $R' \neq R$;
- $D_k =$ *the minimal violated time*, if the insertion of the customer k results in any time window violation. In this case, the minimal sum of time violations for the entirely route which received the customer k is assigned to D_k.

Both, ES1 and ES2 have considered two phase approach in the search. In the first step, the total travel distance is suppressed. In the second and last step, after the number of vehicle has been minimized, the search is redirected to minimize the total travel distance. The difference between ES1 and ES2 stay on the existence or not of the crossover to produce a new generation. In the ES1 the new generations are produced directly by mutations. These heuristic methods reduced the number of vehicles in two instances from the class R1 (R104 and R112). In the instance R109, the strategy ES1 still produced a new result, maintaining NV and reducing TD. In the same way, ES2 improved the results of TD in R105 and R107. In the class R2, five new results were produced by ES1 and three by ES2. The results in the classes, C1 and C2, were equivalent to the best known for all problems. ES2 still produced two new results for RC1 and two others for RC2, while ES1 produced two other new results for RC2. In summary, 20 new results were produced, where only 2 still continue undefeated.

This paper is organized as follows. Section 2 describes the first phase of the search (GA_NV), in order to minimize the number of vehicles. Section 3 describes how CGH is applied to minimize the second objective, travel distance, and the overall algorithm. The sections 4 and 5 describe the results and conclusions, respectively.

2. Hierarchical Tournament Selection Genetic Algorithm

In the first phase, an independently genetic algorithm is proposed to reduce the number of vehicles. Strategies exploring the number of vehicles (NV) and travel distance (TD) objectives in distinguished phases have reached the best results at the moment in the literature. It's possible to see the best results in [13] and their respective references.

Although the number of vehicles is considered as the first objective, the total travel distance minimization continues to be very important, because it is the differentiate criterion as a second objective, once many algorithms have reached the same number of vehicles for many instances. By one side, the genetic algorithm and set partitioning two phase approach, composing the column generation heuristic CGH proposed by [1] showed to be competitive to minimize TD. Consequently, the main question is if the proposed CGH continues to be efficient when the NV is fixed in a minimized solution. However, an additional phase to minimize the NV is necessary. This is a second objective of this paper, because it's very difficult for a robust algorithm to present good NV results to different instances. In this way, the fitness approach of Homberger evolution strategy, summarized in the previous section, was extended adding new criteria to evaluate how easy could be to eliminate one route from the individual solutions. The basic idea of the genetic algorithm GA_NV to minimize NV is the same utilized in [1]. However, new operators are necessary to change the search. The main characteristics of the algorithm are described bellow.

Chromosome and Individual Representation

The individual representation is the same utilized in [1]. Each customer has a unique integer identifier i, i = 1,..,N, where N is the number of customers. The chromosome is defined as a string of integers, representing the customers to be served by only one vehicle. An individual, that represents a complete solution, and consequently many routes, is a set of chromosomes. The central depot is not considered in this representation, because all routes necessarily start and end on it.

Initial Population

To start the first generation in the GA, the *stochastic PFIH* (Push Forward Insertion Heuristic) also proposed in [1] is utilized. It can produce quickly diversified individuals. In the original PFIH, see [14], the first customer in each new route is deterministically defined. Customers then are chosen one by one minimizing the travel distance. The original PFIH is deterministic, but differently, in the stochastic PFIH a random choice is used to define the first customer for each new route. That is necessary to produce distinguished individuals in the first GA generation.

Fitness

There are basically two main alternatives to evaluate individuals to the VRPTW in order to minimize the number of vehicles. In the first one the search occurs by the unfeasible region, and the evaluation is based on how much the time windows are violated. This option was adopted, for example in the genetic algorithm proposed by [2]. On the other hand, the second one treats only feasible individual solutions. This option needs the identification of characteristics in the individual solutions. They have hints of easiness to reduce routes, once the main objective NV is not sufficient to differentiate individuals and ensure the population evolution.

In this paper, a k-way tournament selection method is used in the GA. In a k-way tournament, k individuals are selected randomly. Then, the individual with the highest fitness is the winner. This process is repeated until the necessary number of selected individuals to the crossover phase has been reached. Two individuals are selected for each crossover, which produces only one new offspring. As showed before, [7] propose some criteria of evaluation in a lexicography order. The first criterion is the number of vehicle by itself, followed by the number of customers in the shortest route and the minimal delay time D_R, previously described. It's possible to see that the minimal delay time resembles the idea of the search by the unfeasible time windows region.

The strategy utilized in this paper to expand the fitness idea proposed by [7] evaluates how hard is to eliminate customers from the shortest route. However, it is possible to show that the use of all three criteria of evaluations from Homberger is not enough to differentiate the individuals. In other words, it's possible to find different individuals with the same value in all three criteria. Consequently, the idea is to use additional parameters to identify the easiest individual from the current tournament to eliminate a shortest route. Surprisingly, no less than nine additional criteria of evaluation can be useful in order to permit the identification and to eliminate one more route or a customer in the shortest route. These fitness criteria are presented bellow in their respective hierarchical order in the selection algorithm:

1st - Number of Routes (Fitness_NV): The main objective of the problem (NV) is the first criterion

2nd - The Number of Customers in the Shortest Route (Fitness_NCSR): Naturally, the number of customers in the shortest route is the second criterion to identify the easiest individual solution to eliminate one more route.

3rd - Difficulty to Eliminate One Customer from the Shortest Route (Fitness_1CSR): This criterion is very similar with the minimal delay time D_k proposed by [7], but only one customer from the shortest route is considered (the minimal D_k). The other difference is that no delay time is added in D_k for subsequent not violated time windows. In contrast, Homberger considers all delay in any customer caused by a previous violated time window.

4th - Difficulty to Eliminate All Customers from the Shortest Route (Fitness_AllCSR): This criterion (D_R) is the sum of D_k to all customers in the shortest route (equation (1)). This is the main additional criterion utilized by [7]. The difference mentioned above to calculate the delay persists.

5th - Difficulty to Eliminate One Customer from the *Taker Route* (Fitness_1CTR): The *taker route* presents the minimal time window violation if receives a customer from the shortest route. Sometimes, it's necessary to eliminate customers from this route, because inserting some customers in another route can make possible the reception of others from the shortest route. Obviously the shortest route cannot be used as destination of these customers.

6th - Difficulty to Eliminate All Customers from the *Taker Route* (Fitness_AllCTR): Again, the sum of the violated time for all customers from the *taker route* is considered to calculate the Fitness_AllCTR. The idea is the same applied to Fitness_AllCSR, but now the focus is the *taker route* and not the shortest route.

7th - Total Travel Distance (Fitness_TD): The total travel distance (TD) minimization was mentioned as a concurrent objective to the NV minimization. In fact, this can guide the search to a minimal TD with a bigger NV. However, when applied following all priority criteria above, it is verified to be useful to minimize NV.

8th - The number of customers in the *taker route* (Fitness_NCTR): This criterion is motivated by the fact that a short route has more probability to receive other customers without violated constraints.

9th - Sum of Squares of the Number of Customers (Fitness_SSNC): The sum of squares of the number of customers is as large as the concentration of customers in few routes. This criterion is interesting because has an effect over all routes together.

10th - Total load in the Taker Route (Fitness_TLTR): Any route with few loads inside has more capacity to receive any other customer from the shortest route. This criterion is useful for instances where the capacity is more critical than the time window constraints.

Selection

In the k-way tournament selection method k individuals are selected randomly. Then, the individual who presents the highest fitness is the winner and will participate of the crossover. In the hierarchical tournament there is not only one fitness value but many. Each fitness criterion is used to maintain in the tournament process only those individuals which present the highest values. One or more individuals continue in the tournament selection considering the other hierarchical fitness criteria. The hierarchical criteria utilized, as showed in the previous section, can differentiate solution individuals with different number of vehicles, the main objective, but also identify those individuals, with the same number of vehicles, that are easier to eliminate customers in the last route and consequently to eliminate that route.

Crossover

The crossover algorithm is based on the crossover operator proposed by [1] and [11]. In the first step, the algorithm select a route from each parent individual in turns, in order to inherit routes with maximum number of customers. After all feasible routes have been inserted in the offspring, the insertion of the remainder customers is tested in the existing routes (second step). If some customers continue to be without any route there is no other option than to insert them in empty vehicles (new routes). In this case, the stochastic PFIH is again applied.

Mutation

Many mutation operators in the GA proposed by [1] are not appropriated to minimize NV. Those operators give directions to minimize the total travel distance which can represent a local minimum in terms of NV. The new set of operators is proposed, as explained below:

Random Customer Migration (M_RCM): This operator chooses a vehicle randomly and a random customer associated to it; a migration of this customer to other non-empty vehicle is tried. If the insertion results a feasible route, then it is accepted independently of the new function cost.

Simple Customer Exchange (M_SCE): This operator tries a random customer exchange between two different routes. The unique requirement to do the exchange is the feasibility of the resulted individual solution.

Two Customers Swap (M_TCS): This operator is very similar with M_SCE, but the customers come from the same route.

Customer Exchange with Gain in Fitness_1CSR, Fitness_AllCSR or Fitness_1CTR (M_CEGF1): Two routes are randomly selected. Every possible pair of customers, one from each route is candidate to exchange. The customer position in the target route is not necessarily the empty position. The exchange is performed only if there is gain in at least one of these fitness criteria: Fitness_1CSR, Fitness_AllCSR or Fitness_1CTR.

Customer Exchange with Gain in Fitness_1CSR, Fitness_AllCSR (M_CEGF2): Two routes are randomly selected. Every possible pair of customers, one from each route is candidate to exchange. The customer position in the target route have to be the empty position. The exchange is performed only if there is a gain in at least one of these fitness values: Fitness_1CSR or Fitness_AllCSR.

Simple Customer Exchange with Travel Distance Gain (M_SCETDG): This operator tries a random customer exchange between two different routes. Differently of the M_SCE this exchange is performed only if the resulted solution presents a TD reduction.

Customer Exchange with Travel Distance Gain (M_CETDG): This operator tries a random customer exchange between two different routes. Differently of the M_SCETDG the customer previous position does not necessarily need to be used. Again the exchange only is performed if the resulted solution presents a TD reduction.

Taker Route Customer Migration (M_TRCM): This operator chooses a random customer from the *taker route;* a migration of this customer to another non-empty, different of the shortest route, and a random vehicle is tried. If the insertion results a feasible route, then it is accepted independently of the new function cost. If it is impossible to insert this customer in that vehicle, another one is randomly selected until all available vehicles have been tested.

Customer Reinsertion with TD Gain (M_CRTDG): This mutation randomly choses customer in a route and try all position available. If there is another feasible position with travel distance reduction the position is changed.

All Customers Reinsertion (M_AllCR): This mutation is very similar with M_CRTDG but this operation is repeated to every customer in a randomly selected route. Every new position with travel distance reduction is performed.

Taker Route Elimination (M_TRE): This mutation tries to find a new feasible position for every customer of the taker route in another one. Obviously, the shortest route is not considered as destination once the objective is to improve the capacity to receive customers from that one.

Last Customer Elimination from the Shortest Route including Customer Removal in the Destination (M_LCE1): This operator gets this last customer and tries to insert into other non-empty route. If necessary, a second customer in the destination route is removal to permit the insertion. In the last case, a new destination to the removed customer is tried, initially, in all nonempty route in the individual solution. If another feasible position to the removed customer is found, the shortest route has been eliminated. Otherwise the removed customer is inserted in an empty route and stays as the new shortest route.

Last Customer Elimination from the Shortest Route including 2 Customers Removal in the Destination (M_LCE2): This operator is very similar to M_LCE1, but includes the possibility to remove 2 customers in order to permit the insertion of the last customer from the shortest route. If other feasible positions to the removed customers are found, the shortest route has been eliminated. Otherwise the remainder customers are inserted in an empty route and a different new shortest route has been established.

Shortest Route Elimination (M_SRE): This mutation tries to find a new feasible position for every customer of the shortest route in another one.

3. Second Phase and the Complete Algorithm

In the previous sections, the new genetic algorithm GA_NV is proposed to minimize the number of vehicles. The algorithm is applied for a fixed interval (15 min.) in order to find a solution to the VRPTW with the minimal number of vehicles as possible. After that, the entire final population produced by GA_NV is used as the first population in the Column Generation Heuristic (CGH) proposed by [1] to minimize the second optimization criterion , the total travel distance.

The result of GA_NV, block 1 in the 0 are 50 solution individuals containing the best solution in terms of NV. The CGH proposed by [1] gets this population many times to produce many local minima, while the TIME_LIMIT (block 3) permits. But now, the objective is to minimize TD after NV. Because there are many individuals with the same NV and the operators utilized in CGH unlikely can reduce NV, actually TD is the practical objective to be reduced.

The complete algorithm CGH_NV, described in Figure 1, is composed by the basic CGH algorithm (for details see [1] and [11]) joined with GA_NV. It’s important to emphasize the main aspects of that approach. The loop from the block 4 to 8 produces many different routes or local minima using the GA_TD (genetic algorithm wirh the criterion of travel distance minimization). The number of solutions, Max_Evol, generated is 10. The set partitioning problem SSP then is solved to obtain the optimal solution over this limited set of routes *R*, block 9. The solution then is applied to divide the whole problem into many different sub-problems, solved again using the GA_TD, blocks 11 to 16. These 2 cycles are repeated until the TIME_LIMIT (60 min.) is reached. Finally, the SPP is solved over all previous routes generated and included in the set R_{GLOBAL}.

4. Computational Results

The GA_NV parameters were empirically adjusted. A fixed period of 15 minutes has been used to run GA_NV. After that, as show the 0, the final population is passed to CGH in order to minimize TD. This phase has an additional period of 60 minutes to perform the search. In order to define the best hierarchical way to evaluate the individuals the GA_NV algorithm was tested with different options. The Fitness_NV is always the first, because this is the main objective. Again the number of customers in the shortest route (Fitness_NCSR) demonstrates clearly as the second fitness to be applied in the selection. Homberger [7] has used the difficulty to eliminate all customer from the shortest route as third criterion (Fitness_AllCSR) or *delay time*, but there was not information if the author has tried another criterion or comparisons. Also the GA_NV in this work has produced the best results considering the difficult to eliminate all customers in the shortest route (Fitness_AllCSR) in the third priority. The use of Fitness_1CSR has been used in forth priority. The use of Fitness_1CSR as the third criterion has produced fast customer elimination in the first phase but has increased the possibility to entrap in the local minima. The Fitness_TD produces better results only in seventh priority, but with substantial relevance in the search.

Table 1 shows the main results from the literature and the results from CGH_NV proposed. They are positioned in crescent order of NV, the main objective. The results of the CGH_NV are very expressive in NV, reaching the best known value for 100% of the problems. However, in the second objective TD, the results stay a bit above of the best known, produced by Berger.

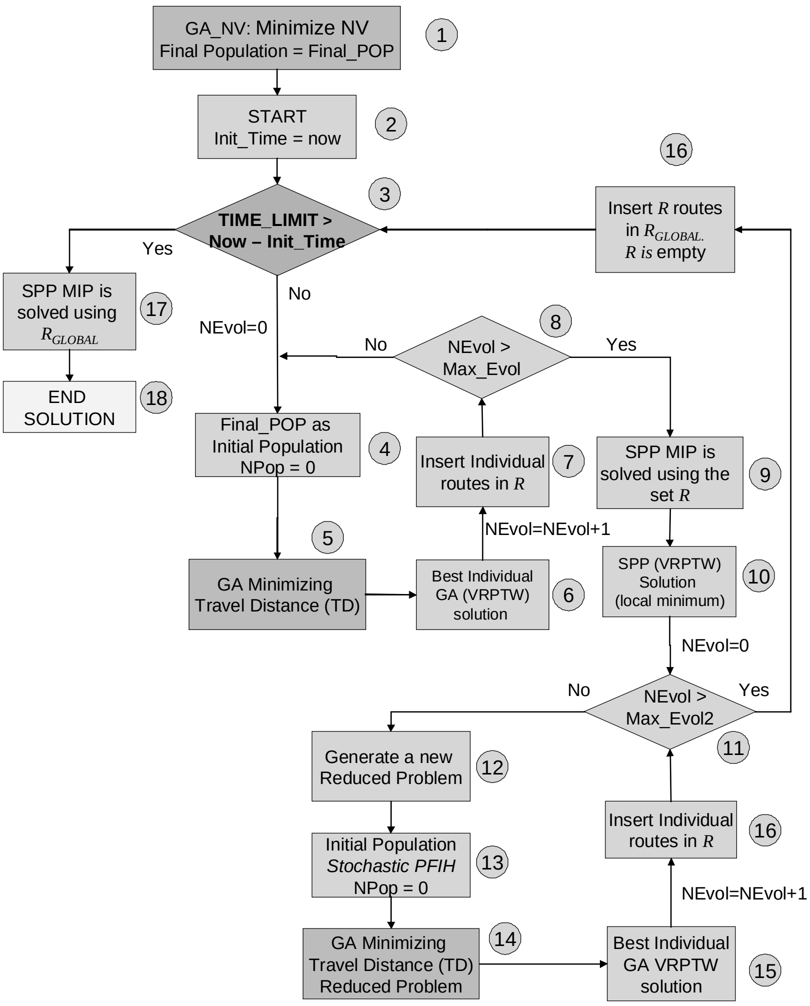

Figure 1:CGH_NV heuristic proposed

Reference	R1	R2	C1	C2	RC1	RC2	NV DT
Berger et al. (2003)	**11.92** 1221	**2.73** 975.4	**10.00** 828.5	**3.00** 590	**11.50** 1390	**3.25** 1159	**405** 5795
CGH_NV (this paper)	**11.92** 1224.	**2.73** 1012.	**10.00** 828.4	**3.00** 590.9	**11.50** 1417.	**3.25** 1195.	**405** 5891
Braysy et al. (2004)	**12.00** 1220.	**2.73** 970.4	**10.00** 828.4	**3.00** 589.9	**11.50** 1398.	**3.25** 1139.	**406** 5779
Homberg et al. (1999)	**11.92** 1228.	**2.73** 970.0	**10.00** 828.4	**3.00** 589.9	**11.63** 1392.	**3.25** 1144.	**406** 5787
Homberg et al (in press)	**12.08** 1211.	**2.82** 950.7	**10.00** 828.5	**3.00** 590.0	**11.50** 1395.	**3.25** 1135.	**408** 5742
Liu et al. (1999)	**12.08** 1215.	**2.91** 953.4	**10.00** 828.4	**3.00** 589.9	**11.75** 1385.	**3.25** 1142.	**411** 5746
Taillard et al. (1997)	**12.25** 1216.	**3.00** 995.4	**10.00** 828.5	**3.00** 590.3	**11.88** 1367.	**3.38** 1165.	**416** 5799
Rochat et al. (1995)	**12.58** 1197.	**3.09** 954.4	**10.00** 828.5	**3.00** 590.3	**12.38** 1369.	**3.62** 1139.	**427** 5712

Table 1: Best known results (july/04) GCH_NV

5. Conclusions

In the literature, the main results minimizing NV for Solomon's test set have been compared with the results produced by CGH_NV. The proposed approach has produced the best NV results and a reasonable TD. The GA_NV proposed to minimize NV has produced the best results in NV using only 15 minutes of execution using a Pentium IV 2.4 GHz.

The new hierarchical tournament selection proposed has been the main reason to the excellent performance of GA_NV in the NV minimization. However the CGH proposed by [1] changed to minimize first NV and then TD as the second criterion seems to have difficulty to perform the search in narrow routes. Once the TD results from CGH without any NV restriction were very significantly [1], it's interesting to extend this work producing new mutation operators in the GA_TD to work with narrow routes found in the GA_NV.

6. References

[1] Alvarenga, G.B., Mateus, G.R. and Tomi, G. (2004), "A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows". *Proceedings of Fourth Intenational Conference on Hybrid Intelligent Systems*, Kitakyushu, Japan, p. 410-415.

[2] Berger, J., Barkaoui, M. and Bräysy, O., (2003), "A route-directed hybrid GA approach for the VRPTW", *Information Systems and Operational Research* 41, 179-194.

[3] Bräysy, O., Hasle G. and Dullaert W., (2004), "A multi-start local search algorithm for the vehicle routing problem with time windows". *European Journal of Operational Research, 159, 586-605.*

[4] Corne, D., Dorigo, M. and Glover, F., (1999), "New Ideas in Optimization".

[5] Gambardella, L.M., Taillard, E. and Agazzi, G. (1999), "MACS-VRPTW: A multipe Ant Colony System for Vehicle Routing Problems with Time Windows", in Chap. 5 of [4].

[6] Homberger, J. and Gehring, H., (1999), "Two evolutionay metaheuristics for the vehicle routing problem with time windows. *Information Systems and Operational Research,* 37, 297-318.

[7] Homberger, J. and Gehring, H., (2002) "Parallelization of a two-phase metaheuristic for routing problems with time windows". *Journal of Heuristics*, 8, 251-276.

[8] Homberger, J. and Gehring, H., (in press) "A two-phase hybrid metaheuristic for the vehicle routing problems with time windows". *European Journal of Operational Research*.

[9] Kohl N., Desrosiers J., Madsen O. B. G., Solomon M. M. and Soumis F. (1997), "K-path Cuts for the Vehicle Routing Problem with Time Windows", Manuscript

[10] Liu, F.-H. and Shen, S.-Y., (1999), "A route-reighborhood-based metaheuristic for vehicle routing problem with time windows". *European Journal of Operational Research,* 118, 485-504.

[11] Oliveira, H.C.B.; Souza, M.M.; Alvarenga, G.B.; Silva, R.M.A. "Adaptação do Algoritmo Genético no tratamento do Problema de Roteamento de Veículos com Janela de Tempo". *INFOCOMP Journal of Computer Science*, V3, N.2, p.51-58.

[12] Rochat, Y., Taillard, E., (1995), "Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics*, 1, 147-167.

[13] Sintef (2004), "Best Known Solutions Identified by Heuristics for Solomon's Benchmark Problems". http://www.sintef.no/static/am/opti/projects/top/vrp/bknown.html.

[14] Solomon, M. M. (1987), "Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints", *Operations Research* 35 (2), 254-265.

[15] Taillard, E., Badeau, P. Gendreau, M., Guertin, F. and Potvin, J.-Y., (1997), "A tabu search heuris tic for the vehi cle routing problem with soft time windows. *Transportation Science,* 31, 170-186.