Analyzing a Unified Ant System for the VRP and Some of Its Variants

Marc Reimann, Karl Doerner, and Richard F. Hartl

Institute of Management Science, University of Vienna, Brünnerstrasse 72,
A-1210 Vienna, Austria
{marc.reimann, karl.doerner, richard.hartl}@univie.ac.at
http://www.bwl.univie.ac.at/bwl/prod/index.html

Abstract. In this paper we analyze the application of an Ant System to different vehicle routing problems. More specifically, we study the robustness of our Unified Ant System by evaluating its performance on four different problem classes within the domain of vehicle routing.

1 Introduction

Vehicle routing and Scheduling has been of major interest to the scientific community for the last 50 years due to the inherent complexity of the associated problems. A large number of different approaches, in recent years mostly meta-heuristics have been proposed for different variants of vehicle routing and scheduling problems, see further Toth and Vigo [1].

On the other hand, goods distribution and more specifically vehicle routing represents an important area of an economy. Basically, in every supply chain transportation occurs between member companies of the chain or from the chain to final customers. Thus, firms have recognized the need to automatize this process and use software to support their distribution process. Part of this software is an optimization tool for routing and scheduling.

However, while it seems that scientific interest and company interest overlap in this regard, this first impression is not completely true. Rather, the level of development of tools in the scientific world is not matched in industry. Most of the tools used in industry are based on very simple optimization techniques developed many years ago.

The reasons for this are manyfold. First, it can not be expected that cutting edge research will instantaneously be transferred to industry. Rather a certain lag has to be accepted. Second, in the quest for ever more sophisticated optimization techniques the scientific community has reached a level of specialization that prevents the tools developed from being applicable to a wide range of problems. Thus, while these tools solve particular problem scenarios much better than the 'old' simple techniques, they lack the flexibility of the latter approaches. However, for practical purposes solution quality is not the only criterion for the choice of an optimization tool.

This gap has been recognized by some researchers. For example Cordeau et al.[2] present four measures of algorithm performance, namely solution quality,

S. Cagnoni et al. (Eds.): EvoWorkshops 2003, LNCS 2611, pp. 300–310, 2003.
© Springer-Verlag Berlin Heidelberg 2003

speed, flexibility and simplicity. According to their analysis most of the metaheuristics, while outperforming simple heuristics with respect to solution quality can not compete with simple heuristics when it comes to speed or simplicity. Simplicity is customarily measured in terms of the number of parameters of an algorithm, while flexibility is concerned with the applicability of an algorithm to different variants of a basic problem. This applicabilty covers also issues of robustness, i.e. whether parameter changes are necessary if the problem instance to be solved changes. On the other hand, a research group in Oslo works on a formal representation and algorithms for rich vehicle routing problems, which cover a wide range of problems with different constraints and objectives (c.f. [3]).

In light of these observations, the aim of our paper is to present an Ant System algorithm that is capable of solving different variants of vehicle routing problems reasonably well, with little or no adjustments to the implementation. More specifically, we aim to solve the vehicle routing problem (VRP), the vehicle routing problem with time windows (VRPTW), the vehicle routing problem with backhauls (VRPB) and the vehicle routing problem with backhauls and time windows (VRPBTW). The algorithm is based on the Insertion based AS we presented in Reimann et al. [4].

The remainder of this paper is organized as follows. In the next section we describe the characteristics of the different problems we aim to solve. After that we briefly describe our Unified AS approach. In section 4 we present our rich computational study, before we conclude in Section 5.

2 The Vehicle Routing Problem and Some of Its Variants

The classic VRP (c.f. e.g [1]) aims to find a set of minimum cost routes, each starting and ending at a central depot. These routes have to satisfy the known demands of a number of customers, where each customer must be served by exactly one vehicle, i.e. order splitting is not an option. Of course vehicle capacities and (possibly) maximum tour length restrictions have to be respected. It is assumed that the available fleet is homogeneous, i.e. the vehicles are identical.

This basic setup can be extended in several directions to deal with more realistic scenarios. The most important extensions are those dealing with more complicated restrictions at the customers. More specifically, two characteristics are dominant:

- time windows: Certain or all customers may require that service is performed within a certain time span. In this case, vehicles arriving at a customer before the beginning of the time window have to wait until the customer is willing to accept the service. A vehicle that arrives at a customer location after the end of the time window set by the customer may not serve the customer anymore. However as service is obligatory, this means that any algorithm has to make sure that all vehicles arrive at each customer before the end of the respective time window. The VRPTW has been studied extensively in the last decade; for a recent overview of metaheuristic approaches see (e.g. [5]).

– backhauls: Intermediate companies of a supply chain often deal with suppliers and customers. It has been recognized that a combination of the two distribution areas leads to significant improvements. Thus, a vehicle that delivers goods to customers may also pick up goods from the company's suppliers. This leads to a mixed problem where some locations need delivery while others require pick up of goods. In the backhauling case an additional restriction is that all the pickups be after the deliveries. The intuition for this is that picking up goods before all deliveries are done may lead to the necessity of rearranging goods in the truck. Doing these rearrangements en route is expensive and thus to be avoided. The VRPB has also received a lot of attention recently (see e.g. [6]).

Given these additional problem characteristics, we end up with 4 problem classes. The simplest case is the vehicle routing problem (VRP) without time windows or backhauls. Adding one of the two customer characteristics leads to either the vehicle routing problem with time windows (VRPTW) if time windows are considered, or the vehicle routing problem with backhauls (VRPB) if backhauls are treated. Finally, both types of constraints are added in the vehicle routing problem with backhauls and time windows (VRPBTW).

Compared to the first three problems, the VRPBTW has received only very little attention. The only meta-heuristic approach is due to Duhamel et al. [8] who proposed a Tabu Search algorithm to tackle the problem.

In the next section we will present an Ant System that can be applied to each of the four problems. We will discuss the implementation with respect to the characteristics of the different problems. In closing this section let us note that other possible extensions are the consideration of vehicle heterogeneity, multiple depots or multiple trips per vehicle. These extensions have been studied by different authors (c.f. Gendreau et al. [9], Cordeau et al. [10]). Testing whether our algorithm applies to these problems is left for further research.

3 Ant System Algorithms for the VRP and Some of Its Variants

In this section we briefly describe our Ant System algorithm. In particular, we focus on the constructive heuristics used.

3.1 Ant Systems

The Ant System approach, originally proposed by Colorni et al. (see e.g. [11]) is based on the behavior of real ants searching for food. Real ants communicate with each other using an aromatic essence called pheromone, which they leave on the paths they traverse. In the absence of pheromone trails ants more or less perform a random walk. However, as soon as they sense a pheromone trail on a path in their vicinity, they are likely to follow that path, thus reinforcing this trail. More specifically, if ants at some point sense more than one pheromone trail, they will choose one of these trails with a probability related to the strenghts of the existing trails. This idea has first been applied to the TSP, where an ant located in a city chooses the next city according to the strength of the artificial trails.

Improved versions of the basic algorithm have been applied to a large number of different combinatorial optimization problems (for an overview see [12]). Recently, a convergence proof for a generalized Ant System has been developed by Gutjahr [13]. Generally, the Ant System algorithm consists of the iteration of three steps:

- Generation of solutions by ants according to private and pheromone information
- Application of a local search to the ants' solutions
- Update of the pheromone information

Below we will discuss the first step in more detail. The second and third step are reviewed only briefly (c.f. [4] for a more detailed description of these steps).

3.2 Generation of Solutions

As proposed in Reimann et al. [4], we use an Insertion algorithm derived from the I1 insertion algorithm proposed by Solomon [14] for the VRPTW. This algorithm works as follows: Routes are constructed one by one. First, the unrouted customer farthest from the depot is selected as a seed customer for the current route, that is, only this customer is served by the route. Sequentially other customers are inserted into this route according to a cost criterion based on their distance to the depot as well as the detour and delay caused by inserting them. Once no more insertions are feasible with respect to time window, capacity or tour length constraints, another route is initialized with a seed customer and the insertion procedure is repeated with the remaining unrouted customers. The algorithm stops when all customers are assigned to routes.

In order to use the algorithm described above within the framework of our Ant System we need to adapt it to allow for a probabilistic choice in each decision step. This is done in the following way. To initialize a tour, seed customers are not chosen deterministically but probabilistically according to their distance from the depot.

Inserting further customers on the current tour is done using a roulette wheel selection over all unrouted customers with positive evaluation function κ_i. The decision rule used can be written as

$$\mathcal{P}_i = \begin{cases} \frac{\kappa_i}{\sum_{h|\kappa_h>0} \kappa_h} & \text{if } \kappa_i > 0 \\ \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

The chosen customer i is then inserted into the current route at its best feasible insertion position.

To compute the evaluation function κ_i for inserting an unrouted customer i at its best insertion position on the current tour we first determine for each unrouted customer i the attractiveness of insertion at any feasible insertion position on the current tour. Formally the attractiveness of inserting customer i immediately after customer j can be written as

$$\begin{aligned}&\eta_{ij} = \max\ \{0,\\ &\alpha \cdot d_{0i} - \beta \cdot (d_{ji} + d_{ik} - d_{jk}) - (1-\beta) \cdot (b_k^i - b_k)\\ &+\delta \cdot (\gamma \cdot type_i + (1-\gamma) \cdot (1 - type_i))\} \qquad \forall i \in N_u, \forall j \in R_i,\end{aligned}$$

where d_{0i} denotes the distance between the depot and customer i and N_u denotes the set of unrouted customers. Further, k is the customer visited immediately after customer j in the current solution, b_k^i is the actual arrival time at customer k, if i is inserted between customers j and k, while b_k is the arrival time at customer k before the insertion of customer i and R_i denotes the set of customers assigned to the current tour after which customer i could feasibly be inserted. Finally, $type_i$ is a binary indicator variable denoting whether customer i is a linehaul ($type_i = 0$) or a backhaul customer ($type_i = 1$). The intuition is that we want to be able to discriminate between linehaul and backhaul customers. Note, that given 'inappropriate' values for the parameters α, β, γ and δ the attractiveness η_{ij} can become zero. Note further, that our attractiveness function is an extension of the function proposed by Solomon [14] for the VRPTW. More precisely, our function reduces to Solomon's function if δ is set to $\delta = 0$.

Given the attractiveness we then compute the evaluation function of the best insertion position for each customer i on the current tour by

$$\kappa_i = \max\ \{0, \max_{j \in R_i}[\eta_{ij} \cdot \frac{\tau_{ji} + \tau_{ik}}{2 \cdot \tau_{jk}}]\} \qquad \forall i \in N_u,$$

where τ_{ji} denotes the pheromone concentration on the arc connecting locations (customers or depot) j and i. The pheromone concentration τ_{ji} contains information about how good visiting two customers i and j immediately after each other was in previous iterations. Note, that the same pheromone utilization is done for route initialization, thus augmenting the attractiveness of initializing a route with an unrouted customer i by the search-historic information.

Computing the attractiveness η_{ij} reflects the tradeoff between detour and delay costs associated with inserting a customer. This tradeoff is weighted by the parameter β. A higher value for β puts higher emphasis on the detour and lower emphasis on the delay. The tradeoff is also influenced by customer characteristics such as distance to the depot or type and these characteristics are weighted by the parameters α and γ, δ respectively. More precisely, higher α will favor customers far from the depot, while higher δ will put more emphasis on the customer type. Finally, γ influences the discrimination between linehaul and backhaul customers.

3.3 Local Search

After an ant has constructed its solution, we apply a local search algorithm to improve the solution quality. In particular, we sequentially apply *Move* and *Swap* operators to the solution. Generalized versions of both operators have been proposed by Osman [15] for the VRP.

The *Move* operator tries to eject one or two adjacent customers from their current position and insert it at another position. It is a special case of the 3-opt operator. The elementary *Swap* operator, aims at improving the solution by exchanging a customer i with a customer j. This operator is a special case of the 4-opt operator. Both operators are used to move and exchange customers of the same tour and between tours.

3.4 Pheromone Update

After all ants have constructed their solutions, the pheromone trails are updated on the basis of the solutions found by the ants. According to the rank based scheme proposed in [16] the pheromone level on all arcs is first decreased according to the evaporation factor $(1-\rho)$. Second, only arcs belonging to either the best solution found so far or to one of the $E-1$ best solutions found in the current iteration are receiving positive reinforcement. The amount of positive reinforcement depends on the rank of the ant and is proportional to the inverse solution quality found by the ant. That way the search is driven to promising areas of the search space. For more details on this approach we refer to [16].

4 Numerical Analysis

As testing our algorithm on all instances in the four classes would have been computationally too expensive we performed our numerical analysis on selected problem instances. First, to eliminate size effects we only considered problems with approximately 100 customers. In fact, only the VRPB instances have 90 and 94 customers respectively as for this class 100 customer instances are not available. Also, in order not to bias the results by taking 'easy' or 'hard' instances we randomly chose 19 instances. These are:

- VRP (from [17]): vrp3, vrp8, vrp12, vrp14
- VRPTW (from [14]): r101n, r206n, c105n, c207n, rc104n, rc203n
- VRPB (from [18]): lhbh-i1, lhbh-i3, lhbh-j1, lhbh-j3
- VRPBTW (from [7]): bhr101b, bhr102a, bhr103b, bhr104c, bhr105c

4.1 Parameter Settings

Let n be the problem size, i.e. the number of customers to be served, then the Ant System parameters were: $m = \lceil n/2 \rceil$ ants, $\rho = 0.95$ and $E = 4$ elitists. These parameters are standard settings and were not tested systematically as our experience suggests that the rank based Ant System is quite robust.

Generally, the objective for time constrained routing problems is to first minimize the fleet size required to serve all customers and given a minimal fleet size to minimize the total distance travelled. This objective was established by minimization of the following objective function:

$$L = M \cdot FS + TT, \tag{2}$$

where L denotes the total costs of a solution, FS denotes the fleet size found, and TT corresponds to the total travel time (or distance). The parameter M has to be chosen in a way to ensure that a solution that saves a vehicle always outperforms a solution with a higher fleet size. More precisely we set $M = 10000$ in our experiments.

Finally, as we are interested in the interplay of the parameters of our insertion algorithm with the characteristics of the different problems we analyzed the four parameters of the insertion algorithm in the ranges $\alpha \in \{0.5, 1, 1.5, 2\}$, $\beta \in \{0, 0.33, 0.66, 1\}$, $\gamma \in \{0, 0.33, 0.66, 1\}$ and $\delta \in \{0, 0.33, 0.66, 1\}$ on the instances described above.

This means that we tested 256 parameter constellations for each instance. For each constellation we performed 1 run of 10 minutes. All our computations were performed on a Pentium 3 with 900MHz. The code was implemented in C.

4.2 Influence of the Problem Characteristics on the Parameter Values

As stated in the last section, we have tested a large number of constellations for each of the 19 problem instances. As our main interest lies in the robustness of the final approach we have compared the different settings based on averages over all instances. First, we select for each instance the best solution found by our Unified Ant System (over all parameter settings) as a reference value. Each individual setting was then evaluated by relating its performance to the reference values.

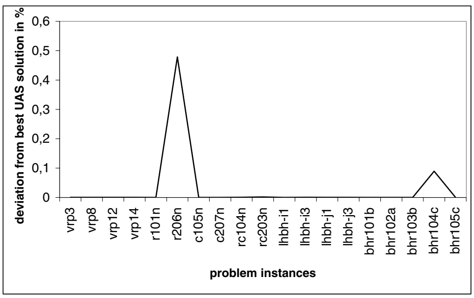

Fig. 1. Robustness of the 'best' setting over the 19 test instances

Using this approach, we found the following 'best' setting:

$$\alpha = 2, \beta = 1, \gamma = 0.33, \delta = 1$$

The average deviation of this setting over the best solution found for each instance is equal to 0.03%. Let us look more precisely at the behavior of this setting over all instances. Figure 1 shows the percentage deviation from the best solution found by our Unified AS.

Clearly, for 17 out of the 19 instances this setting shows excellent behavior with virtually 0% deviation. For the remaining two instances the deviations of 0.1% and 0.5% respectively still are more than reasonable.

Given these encouraging results, let us now perform some sensitivity tests. More precisely, we are interested how changing one parameter from its 'best' setting influences the performance of the Unified AS given that all other parameters remain unchanged. Table 1 shows the deviation in % from the best solution found by our Unified AS, for different settings of the 4 parameters.

Table 1. Sensitivity analysis of the parameter settings

Parameter setting	0.5	1	1.5	2
α	0.14%	0.07%	0.05%	0.03%

Parameter setting	0	0.33	0.66	1
β	0.03%	0.04%	0.04%	0.03%
γ	0.03%	0.03%	0.03%	0.03%
δ	0.03%	0.03%	0.03%	0.03%

Clearly, these results show that the parameter α has the strongest influence on solution quality. The other three parameters show basically no sensitivity to their actual setting. To understand this result let us consider the effects of the parameters. Both the parameters α and δ are free parameters, i.e. they can take any values. Moreover, higher levels for these parameters may implicitly increase the number of positive evaluations κ_i. Thus, the number of customers on each truck may increase which in turn leads to a possible reduction in fleet sizes. As this is the main goal, higher values for α and δ should be expected to improve the solution quality. However, both of the free parameters α and δ are at the highest level we tested. This suggests that considering even higher values for these parameters might further improve the performance of the algorithm. This question is not considered in the remainder of this paper, but rather left for future research.

Finally, the parameters β and γ have more indirect effects on the fleet size such that their actual values will influence mainly the second objective namely the total distance travelled.

4.3 Comparison of Our Unified AS with Existing Results

Now that we have analyzed the influence of the actual parameter settings on the solution quality let us evaluate the performance of the 'best' setting we found against state of the art results. More precisely, we consider for each instance the best known solution.[1] Apart from the VRP instances - for those the standard objective is to minimize total distance travelled only - all the other best known results are based on the same lexicographic ordering of objectives we use.

In our comparison, we split the solutions into the fleet size and the total distance travelled. Table 2 shows the results for all of the 19 instances.

Table 2. Comparison of our Unified AS with best known results

Problem	Unified AS		Best known results	
instance	fleet size	total distance	fleet size	total distance
$vrp3$	8	859.68	8	826.14
$vrp8$	9	886.47	9	865.94
$vrp12$	10	848.64	10	819.56
$vrp14$	10	971.82	11	866.37
$r101n$	19	1655.41	19	1650.80
$r206n$	3	979.45	3	912.00
$c105n$	10	828.94	10	828.94
$c207n$	3	589.20	3	588.29
$rc104n$	10	1196.05	10	1135.48
$rc203n$	3	1175.88	3	1060.45
$lhbh-i1$	9	361.08	10	360.00
$lhbh-i3$	5	317.07	5	306.00
$lhbh-j1$	10	352.93	10	352.00
$lhbh-j3$	7	301.67	7	302.00
$bhr101b$	23	1975.07	23	1999.20
$bhr102a$	19	1681.34	19	1677.60
$bhr103b$	15	1405.02	16	1395.88
$bhr104c$	12	1216.32	12	1208.50
$bhr105c$	17	1642.70	17	1633.01
Total	202	19244.75	205	18788.16

Our results show that considering the first goal, namely the fleet size, our algorithm improves the best known results in three cases, leading to a reduction in fleet sizes equal to 1.46%. With respect to the second objective, our algorithm deviates by 2.43% from the best known solution. Thus, the reduction in fleet size is achieved through giving up some solution quality associated with the total distance travelled. Overall, our algorithm being the first unified approach for the four classes studied shows very promising behavior.

[1] Note, that these best known results are taken from different sources as to date there is no unified approach for the studied problem classes. Additional information about the corresponding sources can be found at http://www.bwl.univie.ac.at/bwl/prod/reimann/vrpresults.html.

4.4 Flexibility and Simplicity of Our Unified Ant System

The approach presented and evaluated above was developed to deal with the issues of flexibility and simplicity, which become more and more important as additional criteria to compare heuristic algorithms as pointed out in the introduction. The best known results presented above come from a number of researchers using different techniques and (presumably) none of these approaches was designed to be flexible in terms of applicability for different variants of the VRP. Thus, a comparison of flexibility and simplicity is impossible.

However, we believe that our results enable us to make some statements for our approach. First, our algorithm can be applied to (at least) the four types of problems studied without modification. In fact, once the format of the input data is fixed, our algorithm does not make a distinction between the types of problems. Further, while the Ant System features a significant number of parameters, our results suggest that the performance of the algorithm is quite robust over the different types of problems and instances. Finally, the basic principle of the Ant System meta-heuristic is quite simple.

Overall, our algorithm seems to be rather flexible, simple and robust. However, these careful statements show that a much more thorough analysis of these issues is necessary.

5 Conclusions and Future Research

The results presented in this paper suggest that our Unified AS is capable of finding high quality solutions for four important problem classes within the domain of vehicle routing. While some solution quality has to be sacrificed to gain this flexibility, the overall performance of the algorithm is still very reasonable.

However, we clearly have to gather more experimental evidence to support our thesis about the robust performance of our approach. First, we need to extend the parameter ranges for α and δ as we found that these parameters should take the maximum possible values in the domains we currently specified. Second, we have to evaluate our Unified AS on an extended problem set that also includes larger instances. Finally, we aim to extend our model further by giving up the linehaul backhaul precedence relationship and by considering, among others, soft time windows and multiple depots.

Apart from that the tradeoff between the two objectives suggests, that a true multi-objective approach should be applied to further enhance the applicability of the approach to real world problems. An existing approach by Jozefowiez et al.[19] studies this possibility for the basic VRP. Their objectives are the total distance travelled and a load balancing criterion that favors solutions with tours that differ little in their total distance.

Acknowledgments. The authors wish to thank three anonymous referees for valuable comments on the paper and Harald Andree for performing the numerical tests. This work was supported by the Oesterreichische Nationalbank (OeNB) under grant #8630.

References

1. Toth, P. and Vigo, D. (Eds.): The Vehicle Routing Problem. Siam Monographs on Discrete Mathematics and Applications, Philadelphia (2002)
2. Cordeau, J. F., Gendreau, M., Laporte, G., Potvin, J. Y. and Semet, F.: A guide to vehicle routing heuristics. Journal of the Operational Research Society **53** (5) (2002) 512–522
3. http://www.top.sintef.no/
4. Reimann, M., Doerner, K., Hartl, R. F.: Insertion based Ants for Vehicle Routing Problems with Backhauls and Time Windows. In: Dorigo, M. et al. (Eds.): Ant Algorithms, Springer LNCS 2463, Berlin/Heidelberg (2002) 135–147
5. Bräysy, O. and Gendreau, M.: Metaheuristics for the Vehicle Routing Problem with Time Windows. Sintef Technical Report STF42 A01025 (2001)
6. Toth, P. and Vigo, D.: VRP with Backhauls. In Toth, P. and Vigo, D. (Eds.): The Vehicle Routing Problem. Siam Monographs on Discrete Mathematics and Applications, Philadelphia (2002) 195–224
7. Gelinas, S., Desrochers, M., Desrosiers, J. and Solomon, M. M.: A new branching strategy for time constrained routing problems with application to backhauling. Annals of Operations Research. **61** (1995) 91–109
8. Duhamel, C., Potvin, J. Y. and Rousseau, J. M.: A Tabu Search Heuristic for the Vehicle Routing Problem with Backhauls and Time Windows. Transportation Science. **31** (1997) 49–59
9. Gendreau, M., Laporte, G., Musaraganyi, C. Taillard, E. D.: A tabu search heuristic for the heterogeneous fleet vehicle routing problem. Computers and Operation Research **26** (1999) 1153–1173
10. Cordeau, J. F., Gendreau, M., Laporte, G.: A Tabu Search Heuristic for Periodic and Multi-Depot Vehicle Routing Problems. Networks **30** (1997) 105–119
11. Colorni, A., Dorigo, M. and Maniezzo, V.: Distributed Optimization by Ant Colonies. In: Varela, F. and Bourgine, P. (Eds.): Proc. Europ. Conf. Artificial Life. Elsevier, Amsterdam (1991) 134–142
12. Bonabeau, E., Dorigo, M. and Theraulaz, G.: Swarm Intelligence. Oxford University Press, New York (1999)
13. Gutjahr, W. J.: ACO algorithms with guaranteed convergence to the optimal solution. Information Processing Letters. **82** (2002) 145–153
14. Solomon, M. M.: Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. Operations Research. **35** (1987) 254–265
15. Osman, I. H.: Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. Annals of Operations Research. **41** (1993) 421–451
16. Bullnheimer, B., Hartl, R. F. and Strauss, Ch.: A new rank based version of the ant system: a computational study. Central European Journal of Operations Research **7**(1) (1999) 25–38
17. Christofides, N., Mingozzi, A. and Toth, P.: The vehicle routing problem. In: Christofides, N. et al. (Eds.): Combinatorial Optimization. Wiley, Chicester (1979)
18. Jacobs-Blecha, C., Goetschalckx, M.: The vehicle routing problem with backhauls: properties and solution algorithms. Technical Report, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia (1992)
19. Jozefowiez, N., Semet, F. and Talbi, E.: Parallel and Hybrid Models for Multi-objective Optimization: Application to the Vehicle Routing Problem. In Guervos, J. M. et al. (Eds.): Parallel Problem Solving from Nature – PPSN VII, Springer LNCS 2439, Berlin/Heidelberg (2002) 271–280