

A algorithm for the Vehicle Problem

Aziz Ezzatneshan

Department of Computer Engineering, Amirkabir University of Technology, Tehran, Iran
 Corresponding author E-mail:

Abstract: *In this paper, we propose a hybrid ACO algorithm for solving vehicle routing problem (VRP) heuristically in combination with an exact In the basic VRP, geographically scattered customers of known demand are supplied from a single depot by a fleet of identically capacitated vehicles. The intuition of the proposed algorithm is that nodes which are near to each other will probably belong to the same branch of the minimum spanning tree of the problem graph and thus will probably belong to the same route in VRP. Given a clustering of client nodes, the solution is to find a route in these clusters by using ACO with a modified version of transition rule of the ants. At the end of each iteration, ACO tries to improve the quality of solutions by using a local search algorithm, and update the associated weights of the graph arcs.*

Keywords: *Hybrid algorithm; Vehicle Routing problem; Ant Colony Optimization.*

1. Introduction

The vehicle routing problem is one of the main combinatorial optimization problems that many heuristics are compositing to solve it. The basic vehicle routing problem (VRP) consist of a number of customers., each requiring a specified weight of goods to be delivered. Vehicles dispatched from a single depot must deliver the goods required, and then return to the depot. Each vehicle can carry a limited weight and may also be restricted in the total distance it can travel. Only one vehicle is allowed to visit each customer. The problem is to find a set of delivery routes satisfying these requirements and giving minimal total cost. In practice, this is often taken to be equivalent to minimizing the total distance traveled, or to minimizing the number of vehicles used and then minimizing total distance for this number of vehicles.

The VRP is categorized by Capacitated VRP (CVRP), Multi-depot VRP (MDVRP), VRP with Time Windows (VRPTW), Stochastic VRP (SVRP), Site-dependent VRP (SDVRP), Open VRP (OVRP) and so on with different constraints [1,2].

In the CVRP one has to deliver goods to a set of customers with known demands on minimum-cost vehicle routes originating and terminating at a depot. The vehicles are assumed to be homogeneous and having a certain capacity. In some versions of the CVRP one also has to obey a route duration constraint that limits the lengths of the feasible routes. The VRPTW extends the CVRP by associating time windows with the customers. The time window defines an interval during which the customer must be visited. The OVRP is closely related to the CVRP, but contrary to the CVRP a route ends as soon as the last customer has been served as the vehicles do not need to return to the depot. The MDVRP extends the CVRP by allowing multiple depots. The SDVRP is

another generalization of the CVRP in which one can specify that certain customers only can be served by a subset of the vehicles. Furthermore, vehicles do not need to have the same capacity in the SDVRP. In the CVRP, MDVRP and SDVRP one seeks to minimize the total traveled distance, whereas in the OVRP and VRPTW, the first priority is to minimize the number of vehicles and minimizing the traveled distance is the second priority. The choice of objective is not an intrinsic feature of the problems, but just the tradition in the metaheuristic literature. Most exact methods and some metaheuristics for the VRPTW minimize total traveled distance instead of minimizing number of vehicles used.

Due to the nature of the problem it is not viable to use exact approaches for large instances of the VRP (for instances with few nodes the branch and bound technique [3] is well suited and gives the best possible solution). Therefore, most approaches rely on heuristics that provide approximate solutions.

Basically, VRP heuristics can be divided into three categories: construction heuristics and improvement heuristics and metaheuristics. Most of the research efforts aimed at solving the VRP have focused on the development of various metaheuristic algorithms. It has been experimentally shown [4] that, in general, tour-improvement heuristics produce better quality results than tour-constructive heuristics. Still, a tour-constructive heuristic is necessary at least to build the initial solution for the tour-improvement heuristic.

Construction heuristics generate feasible solutions by performing sequences of simple operations that minimize or maximize a given criterion. This class of methods includes the well-known savings heuristic of Clarke and Wright [5] that proceeds by merging two routes into a single route according to the savings obtained by this merger. Other well-known examples of construction

heuristics are the sweep algorithm of Gillett and Miller [6], the insertion heuristic of Christofides et al. [7], the improved savings heuristic of Paessens [8] and the petal algorithm of Renaud et al. [9]. More details can be found in the survey papers of Laporte et al. [10], Laporte and Semet [11], and Van Breedam [12], which presents a parametric analysis of a number of construction heuristics.

Improvement heuristics seek to iteratively enhance a feasible solution (usually generated by a construction heuristic) through locally replacing a set of arcs or customer nodes. These local operations are performed as long as they produce improvements in the value of the objective function. These methods thus typically yield solutions that are local optima, with respect to the set of solution modifications considered. For more information, we refer the reader [13,14], and the surveys [10], [11].

Recently, most of the research efforts aimed at solving the VRP have focused on the development of various metaheuristic algorithms. A metaheuristic can be defined as a top-level general strategy which guides other heuristics to search for good solutions. Most of the VRP metaheuristics are based on some construction and improvement heuristics, i.e., they use the so-called local search principle [15].

The tabu search implementations of [16,17] have obtained the best known results to benchmark VRPs. Various authors have reported similar results, obtained using tabu search [18,22], or simulated annealing [23]. However, it has been observed by [19] that such heuristics require substantial computing times and several parameter settings.

Ant colony optimization is another recent approach to difficult combinatorial problems with architecture number of successful applications reported, including the VRP have been [20-21, 25] developed so far.

The second main metaheuristic principle is population search, which consists of maintaining a pool of solutions, and efficiently combining and replacing the solutions in the pool. A classical example of population search is the genetic algorithm. Applications of genetic algorithms have been reported for the CVRP [26-28], VRP with time windows [2,38], backhauls [40], multi-depot routing problem [35]. A hybrid approach to the vehicle routing problem using neural networks and genetic algorithms [40,41], another hybrid approach using GA an ACO [42] and a school bus routing problem [37] have also been reported.

Reference [29] adapt a variant of genetic algorithm, evolution strategies, for the VRP and combine it with guided local search. The ant algorithms of [21,30-31] can also be viewed as a population search metaheuristics. For more details on the metaheuristics for the VRP, we refer to extensive surveys of [33-34,19].

The paper is organized as follows. In Section II we give a formal definition of the CVRP problem and describe some heuristics for the VRP. Section 3 comprises a

description of ACO approach, while Section IV is devoted to our ACO approach to the VRP and method of constructing the minimum spanning tree for the graph of problem. Next, in Section V, we present the description of the proposed algorithm and in Section VI show the results of implementations. Finally in last Section we draw some overall conclusions and suggest directions for future work.

2. The vehicle routing problem

The basic vehicle routing problem (VRP) consists of a large number of customers, each with a known demand level, which must be supplied from a single depot. . Vehicles dispatched from the depot must deliver the goods required, then return to the depot. Each vehicle can carry a limited weight and may also be restricted in the total distance it can travel. Only one vehicle is allowed to visit each customer, so that all customer demands are satisfied and each customer is visited by just one vehicle. Possible objectives may be to find a set of routes which minimizes the total distance traveled, or minimizes the number of vehicles required and the total distance traveled with this number of vehicles. Various mathematical formulations of the VRP were given by, for example, [43,44].

2.1. Formal problem definition

We now present a mathematical formulation of the Capacitated Vehicle Routing Problem (CVRP) which is the most general version of the VRP. The CVRP is defined on a complete undirected network $G=(V, E)$ with a node set V and an arc set E . Node 0 is a depot with m identical vehicles of capacity Q , m can be fixed a priori or left as a decision variable. Each other node $i > 0$ represents a customer with a non-negative demand q_i and each arc (i, j) has a non-negative travel distance $d_{ij}=d_{ji}$.

The CVRP consists of determining a set of m vehicle trips of minimum total cost, such that each trip starts and ends at the depot, each client is visited exactly once, and the total demand handled by any vehicle does not exceed Q .

Let R_1, \dots, R_m be a partition of V representing the routes of the vehicles to service all the customers. The cost of a given route R_i (where $0 \in R_i$, denotes the depot), is given by:

$$Cost(R_i) = \sum_{(k,j) \in R_i} d_{kj} \quad , i = 1, 2, \dots, m \quad (1)$$

And the cost of the problem solution (s) is:

$$\min f(s) = \sum_{i=1}^m Cost(R_i) \quad (2)$$

With respect to:

$$\sum_{v_i \in R_i} q_i \leq Q \quad , j = 1, 2, \dots, m \quad (3)$$

3. Ant colony optimization

Ant Colony Optimization (ACO) [45,46] belongs to the class of metaheuristics [46]-[48], which are approximate algorithms used to obtain good enough solutions to hard CO problems in a reasonable amount of computation time. Other examples of metaheuristics are tabu search [45], simulated annealing [49], and evolutionary computation [50].

The inspiring source of ACO is the pheromone trail laying and following behavior of real ants which use pheromones as a communication medium. In analogy to the biological example, ACO is based on the indirect communication of a colony of simple agents, called (artificial) ants, mediated by (artificial) pheromone trails. The pheromone trails in ACO serve as a distributed, numerical information which the ants use to probabilistically construct solutions to the problem being solved and which the ants adapt during the algorithm's execution to reflect their search experience.

However, since the initial work of Dorigo, Maniezzo and Coloni on Ant System [51], ACO is now quickly becoming a mature research field: a large number of authors have developed more sophisticated models that were used to successfully solve a large number of complex combinatorial optimization problems and theoretical insights into the algorithm are now becoming available. For ACO convergence proofs, theories and open problems we refer the readers to [52].

Several successful applications of ACO to a wide range of different discrete optimization problems are now available. The large majority of these applications are to NP-hard problems, for example routing (such as vehicle routing problem), assignment, scheduling and subset problems. For a list of current applications of ACO algorithms we refer the readers to [53]. For many of these problems, ACO algorithms produce results that are very close to those of the best-performing algorithms, while on some problems they are the state-of-the-art. These latter problems include the sequential ordering problem, open shop scheduling problems, some variants of vehicle routing problems, classification problems, and protein-ligand docking.

ACO Applications to telecommunication networks are consist of routing problems in circuit switched networks [54] and then in packet-switched networks [55]. For a survey on routing algorithms in a variety of wired network scenarios; see [56]. More recently, an ACO algorithm designed for routing in mobile ad hoc networks was shown to be competitive with state-of-the-art routing algorithms [57], while at the same time offering better scalability.

Apart from the research on previous applications, new trends in ACO are multi-objective problems [61],[66], dynamic optimization problems [55], [57], Stochastic optimization problems [58],[60], Parallel implementations [62-63], Continuous optimization [64,65].

In the constructive phase of the ACO algorithm decisions are based on both heuristic information and the pheromone values. We add a new parameter to this decision in our approach which is called weight. At the end of each iteration, i.e. once all ants have gone through solution construction and local search, the pheromone update procedure is applied to these pheromone values.

Fig. 1. Title of the figure, left justified

4. Our approach

Due to its constructive nature of ACO and because at each construction step an ant chooses exactly one of possibly several ways of extending the current partial solution, ACO can be regarded as a tree search method [67]. This view of ACO as a tree search procedure allows us to put ACO into relation with classical minimum spanning tree.

Our strategy for solving VRP is based on the fact that the VRP is a generalization of the TSP. Thus, besides the routing aspect already existing in the TSP one has to find an assignment (or clustering) of customers to vehicles. Once this assignment is done, the problem reduces to independently solving a VRP. Each ant works only on its current cluster which is a branch of the minimum spanning tree. The intuition is that nodes that are near to each other will probably belong to the same branch of the minimum spanning tree of the graph and thus will probably belong to the same route in VRP.

The idea of this paper is to hybridize the solution construction mechanism of ACO with a modified version of MST, which results in a new approach to solve VRP. In this approach the extension of partial solutions is usually done by using a probabilistic greedy policy with respect to a weighting function that gives weights to the arcs of the problem graph and update the weights to reinforce the edges contained in the best.

4.1. Related work

Decomposing the problem to solve only the much smaller subproblems resulting from the decomposition was presented before in [16],[21],[68].

Reference [21] presents an algorithm that builds on the Savings based Ant System presented before by the authors and enhances its performance in terms of computational effort. Initially, an Ant System solves the master problem for a given number of iterations. Given the best found solution so far the algorithm determines for each route of this solution the center of gravity. Then it clusters these route centers and solves each of the resulting clusters independently by applying SbAS. After all subproblems have been solved it re-assembles the global solution and updates the global pheromone information.

The connection between ACO and minimum spanning tree techniques was established before in [16], [68]. Taillard [16] suggests to decompose the problem and proposes two distinct methods for partitioning a problem

instance. For uniform problems, a partition into sectors is suggested, while for non-uniform problems Taillard uses a partitioning method based on trees and associated shortest path. In this partitioning method, the decomposition of the arborescence consists in deleting arcs in such a way that the subproblems successively created involve a quantity of goods as close to the capacity of the specified number of vehicles and as far from the root as possible. Recently an ACO approach to solve capacitated minimum spanning tree was proposed in [68].

Problem instance	vehicles	Best	Worst	Average	deviation
C1 (50)	11	520	559	550	1.946863
C2 (75)	11	852	881	871	1.692397
C3 (100)	11	732	763	745	1.165359
C4 (150)	9	1082	1111	1101	1.676005
C5 (199)	12	1298	1395	1382	1.842710
C11 (120)	14	1027	1109	1078	6.648178
C12 (100)	12	804	869	840	3.613903

Table 1. Our algorithm’s results for Christofiede’s instances.

Problem instance	[74]	[24]	[16]	[28]	[73]	Our best	V1	V2
C1	524	524	524	524	524	520	5	11
C2	835	835	835	835	835	852	10	11
C3	826	829	826	829	829	732	8	11
C4	1031	1044	1028	1034	1038	1082	12	9
C5	1311	1334	1298	1327	1305	1298	17	12
C11	1042	1042	1042	1046	1074	1027	7	14
C12	819	819	819	819	819	804	10	12

Table 2. Comparison of our algorithm with published results.

Problem instance	Previous best result	Our best result	Worst	Average	deviation
Tai75a	1618	1614	1740	1700	8.881
Tai75b	1344	1335	1375	1371	2.128193
Tai75c	1291	1345	1380	1354	4.767333
Tai75d	1365	1303	1453	1402	17.810032
Tai100a	2041	2101	2162	2115	7.251285
Tai100b	1940	2022	2099	2065	7.072515
Tai100c	1406	1400	1441	1416	5.100808
Tai100d	1581	1692	1756	1727	5.371103
Tai150a	3055	2932	3368	3293	48.282687
Tai150b	2727	2746	3036	2854	12.070302
Tai150c	2341	2355	2673	2531	10.187444
Tai150d	2645	2625	2699	2657	11.233412

Table 3. Our algorithm's results for taillard's instances.

5. The proposed algorithm

The proposed algorithm exploits two important problem characteristics by hybridizing ACO with an exact algorithm for clustering the problem. More precisely, a solution of the problem asks for two decisions, namely the clustering of nodes to find heuristically the nearest nodes and the ‘design’ for each cluster.

The algorithm mainly consists of the iteration of these steps:

- decomposing the problem with minimum spanning tree,
- generation of solutions by ants according to pheromone information and arc weights,
- application of a local search to the ants’ solutions,
- update of the pheromone information,
- update of the arc weights.

The algorithm iteratively does these five steps until the stopping criterion is met. As for all other meta-heuristics, the stopping criterion can either be static, i.e. a fixed number of iterations or dynamic, i.e. a fixed number of non-improving iterations. The implementation of these five steps is described below.

Step 1. decomposing the problem with minimum spanning tree

At this step we build the MST of the problem graph. An MST can be computed in polynomial time by some well-known algorithms, e.g. the algorithms of Kruskal [69] or Prim [70].

Clearly, we want to find a decomposition that leads to sub-problems with geographically close tours in order to be able to then solve these regional sub-problems efficiently. This should help us to improve both the routing on these tours but also the clustering.

The problem of finding an MST is quite easy and several efficient exact algorithms have been proposed for its resolution. In this paper we will use the well known algorithm of Prim. Starting from an empty tree, i.e. no selected arcs, it repeatedly includes the shortest arc connecting a node that is already part of the growing tree with a node that is not yet part of this tree, if it doesn’t construct a tour for connected nodes. The algorithm ends once $n-1$ arcs have been included. The time complexity of an efficient implementation of this algorithm is $O(n^2)$, where n is the number of nodes.

We modify this method by assigning a weight w_{ij} to each arc (i,j) in the problem graph and using this weights instead of length of arcs for constructing the MST. In other words, in each step, the method selects the arc with maximum weight instead of minimum distance that doesn’t construct a tour. The initial weight values are the reverse of distances between nodes, that means.

Step 2. generation of solutions by ants according to pheromone information and arc weights

At the beginning of this step, all ants start their move from depot and select a random branch of the MST to follow. Each ant starts a vehicle which starts its route from the depot and ends it in the depot too. After a vehicle

completes its tour, the ant starts a new vehicle. In other words a tour is built node by node, so each ant iteratively adds new nodes until all nodes have been visited.

This method is implemented as follows. At each time step, each ant uses a transition rule, which is called modified pseudo-random proportional rule to select the next node from the current branch of the tree : Let k be an ant located at a node i , β be a parameter, and q a uniform random value in $[0, 1]$. The next node j to be visited is randomly chosen according to the following probability distribution:

- If $q < q_0$:

$$p_{ij}^k = \begin{cases} 1 & ,if \quad j = \arg \max_{u \in N_i^k} \{\tau_{iu}^\alpha \cdot \eta_{iu}^\beta \cdot w_{iu}^\gamma\} \\ 0 & ,otherwise \end{cases} \quad (4)$$

- else ($q > q_0$)

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta \cdot [w_{ij}]^\gamma}{\sum_{u \in N_i^k} [\tau_{iu}]^\alpha \cdot [\eta_{iu}]^\beta \cdot [w_{iu}]^\gamma} & ,if \quad j \in N_k(i) \\ 0 & ,otherwise \end{cases} \quad (5)$$

Where, τ and η correspond to the pheromone values, heuristic information and weight on edge (i,j) respectively.

In the above equations, $N_k(i)$ is the feasible neighborhood, i.e. the set of nodes which ant k can choose to move to them from node i . The neighborhood consists of all the nodes in the current branch of the MST which hasn't already been visited. If the selected node has a capacity less than or equal to the remaining capacity of the current vehicle, the ant moves to this node, adds it to the current route and subtracts the request of this node from the capacity of the current vehicle, otherwise moving to this node violates the capacity constraint of the vehicle, so in this situation, the ant returns to the depot (it starts a new vehicle with full capacity), and then moves to the selected node.

When all the nodes of the current branch, were selected before, the ant considers the nearest neighbor branch as the current branch. In other words, the nearest branch to node i is assigned to the current branch and neighborhood is updated.

All pheromone parameters have values in the interval $[0, 1]$. We set all initial pheromone values to 1 .

Step 3. application of a local search to the ants' solutions
It is very clear after all the existing literature on VRP that local search is almost mandatory to achieve results of high quality [26],[72]. So In order to find competitive solutions it is in general necessary to use some kind of local search to improve the individual ants' solutions. In the proposed approach, this could be done by the sweep and 3-opt algorithms.

After the ants have constructed their solutions but before the pheromone is updated each ants' solution is improved by applying a local search. We first apply a local search based on swap moves to an ant's solution. The use of swap moves was proposed in [24] for the VRP. A swap move aims at improving the clustering of the

solution by exchanging two customers from different tours, i.e. a customer i from tour k is exchanged with a customer j from tour l .

Following [25] we then apply the 3-opt algorithm [13]. This algorithm iteratively exchanges three arcs with three new arcs until no further improvements are possible. In the context of the VRP it is applied separately to all vehicle routes built by the ants, thus aiming to improve the routing of each tour, while leaving the clustering unchanged. Thus, we first improve the clustering and once no more improvements are possible, we improve the routing.

Step 4. update of the pheromone information

After all ants have constructed their solutions and all ants' solutions have been taken to a local optimum, the pheromone trails are updated on the basis of the local optimal solutions. According to the rank based scheme the pheromone update is as follows [71]:

$$\tau_{ij} = \rho \tau_{ij} + \sum_{r=1}^{n_e-1} \Delta \tau_{ij}^r + n_e \Delta \tau_{ij}^* \quad (6)$$

Where ρ is the trail persistence and n_e is the number of elitists. Using this scheme two kinds of trails are laid. First, the best solution found during the process is updated as if n_e ants had traversed it. The amount of pheromone laid by the elitists is $\Delta \tau_{ij}^*$, where L^* is the objective value of the best solution found so far. Second, the $n_e - 1$ best ants of the iteration are allowed to lay pheromone on the arcs they traversed. The quantity laid by these ants depends on their rank r as well as their solution quality L_r , such that the r th best ant lays $\Delta \tau_{ij}^r$. Arcs belonging to neither of those solutions just lose pheromone at the rate ρ , which constitutes the trail evaporation.

Step 5. update of the arc weights

At the end of each iteration, weight values of the arcs which belong to the best solution are being updated to increase the probability of selecting these arcs in future solutions. The weight update is in the following way

$$w_{ij} = w_{ij} + \lambda \cdot (\tau_{ij} + L^* / L^r)^\delta \quad (7)$$

Where λ is a suitable coefficient and L^* is the objective value of the best solution found in the current iteration. It is clear that arcs with more pheromone values should have more direct effect on new weight values. But since λ has a value less than 1, it couldn't help to increase the weight of selected arcs by itself. So in the above equation, we divided by L^r to use the benefit of pheromone values. We first update pheromones and then use these new pheromone values to modify the weight of arcs which contributed in the current solution.

6. Computational results

A computational experiment has been conducted to compare the performance of the proposed algorithm with some of the best techniques designed for VRP. We executed the algorithm on some of the well-known problem instances from one dataset. Then, additional

simulations were conducted over another dataset and results reported accordingly.

The algorithm were coded in C and, using a 2.0 GHz CPU. It was first applied to the 7 vehicle routing problems proposed by Christofides and can be downloaded from the OR-library (<http://mistic.heig-vd.ch/taillard/problemes.dir/vrp.dir/taillard.dir>), and which have been widely used as benchmarks. The first 5 problems (i.e. C1 to C5) have customers that are randomly distributed with the depot in an approximately central location. In the last two problems (i.e. C11 and C12), the customers are grouped into clusters. Then we applied the algorithm to the twelve instances from the second dataset which proposed by Taillard and can be downloaded from this url: <http://people.brunel.ac.uk/~mastjib/jeb/orlib/files/>. The results for those two datasets are reported for ten independent runs, and in each run the algorithm was executed for 200 iterations. Also we used 10 ants in each iteration for building the solutions. The common parameters for these instances include $\rho=0.8$, $\alpha=\gamma=\delta=1$, $\beta=2$, $q_0=0.9$ and $\lambda=0.2$. We set these parameters according to many experiments.

Table 1 shows the best, worst and average values over the ten runs for each problem. Also it shows the number of vehicles needed for best values and standard deviation from best-known solutions.

Table 2 shows the comparison of our algorithm with published results. The first column describes the various instances and their related size, whereas the second specifies well-known published best results obtained using simulated annealing and tabu search in five references. The following column refer to the best result of our method for these instances. New best results found with our approach are depicted in bold. The results show that our approach could find the new solutions for five instances.

Although minimization of the vehicles is not the aim of our algorithm, we presented the number of vehicles which was needed for finding the best result of previous algorithms in column V1 and for the best result of our algorithm in column V2.

Table 3 shows the results obtained for the second problem instances and presents the comparison of best results of previous methods and our algorithm. We depicted the best result in bold. It also includes the worst and average results of the new proposed algorithm. The results show that our approach could find the new solutions for six instances.

The results show our algorithm can find better solutions for most of the problem instances in comparison with the other metaheuristics proposed previously to solve the VRP.

7. Conclusions

According to recent researches, hybrid algorithms are very promising so we used this idea in our approach. In this paper we presented a new algorithm which hybridized the ACO with an exact algorithm means MST to improve both the performance of the algorithm and the quality of

solutions for the VRP. The intuition is that nodes which are near to each other will probably belong to the same branch of the minimum spanning tree of the graph and thus will probably belong to the same route in VRP. In the proposed algorithm, in each ACO iteration, we first apply a modified implementation of Prim's algorithm to the graph of the problem to obtain a feasible minimum spanning tree (MST). Given a clustering of client nodes, the solution is to find a route in these clusters by using ACO with a modified version of transition rule of the ants. At the end of each iteration, ACO tries to improve the quality of solutions by using a local search algorithm, and update the associated weights of the graph arcs.

Future work will be conducted to improve the proposed algorithm. Alternative formulas will be examined to enhance the method of weight update and pseudo-random decision of ants for selecting the next node in solution construction. Additional improvements might lie on the combination of our approach with the other metaheuristics like genetic algorithm.

8. References

- [1] L.D. Bodin, B. Golden and A. Bender, (1983). The state of the art in the routing and scheduling of vehicle and crews. *Computers and Operations Research*, 10, 79–116.
- [2] J.Y. Potvin, S. Bengio, (1996). The vehicle routing problem with time windows – Part II: Genetic search. *INFORMS. Journal on Computing*, 8, 165–172.
- [3] M.L. Fisher, Optimal solution of vehicle routing problems using minimum K-trees, *Operations Research*, Vol. 42, pp. 626–642, 1994.
- [4] G. Reinelt, *The Traveling Salesman: Computational Solutions for TSP Applications*. LNCS 840, Springer-Verlag, Berlin, Germany, 1994.
- [5] G. Clarke and J. W. Wright . Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research* 1964;12:568–81.
- [6] B.E. Gillett and L. R. Miller, A heuristic algorithm for the vehicle dispatch problem. *Operations Research* 1974;22:340–9.
- [7] N. Christofides, A. Mingozzi and P. Toth, The vehicle routing problem. In: Christofides N, Mingozzi A, Toth P, Sandi C, editors. *Combinatorial optimization*. Chichester:Wiley; 1979. p. 315–38.
- [8] H. Paessens, The savings algorithm for the vehicle routing problem. *European Journal of Operational Research* 1988;34:336–44.
- [9] J. Renaud, F. F. Boctor and G. Laporte, An improved petal heuristic for the vehicle routing problem. *Journal of the Operational Research Society* 1996; 47:329–336.
- [10] G. Laporte, M. Gendreau, J. Y. Potvin and F. Semet, Classical and modern heuristics for the vehicle routing problem. *International Transactions in Operations Research* 2000;7:285–300.
- [11] G. Laporte and F. Semet , Classical heuristics for the capacitated VRP. In: Toth P,Vigo D, editors. *The vehicle routing problem*. Philadelphia: SIAM; 2001. p. 109–28.
- [12] A. Van Breedam, A parametric analysis of heuristics for the vehicle routing problems with side constraints.

- European Journal of Operational Research 2002;137:348–70.
- [13] S. Lin, Computer solutions of the traveling salesman problem. *Bell System Technical Journal* 1965;44:2245–69.
- [14] I. Or, Traveling salesman-type combinatorial problems and their relation to the logistics of blood banking. PhD thesis, Northwestern University, Evanston, USA; 1976.
- [15] J. F. Cordeau, M. Gendreau, G. Laporte, J. Y. Potvin and F. A. Semet, A guide to vehicle routing heuristics. *Journal of the Operational Research Society* 2002;53:512–522.
- [16] R. E. Taillard, Parallel iterative search methods for vehicle routing problems. *Networks* 1993;23:661–73.
- [17] Y. Rochat and R. E. Taillard, Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics* 1995;1:147–67.
- [18] G. Barbarosoglu, D. Ozgur, A Tabu search algorithm for the vehicle routing problem. *Computers & Operations Research* 1999;26:255–70.
- [19] J. Renaud, F.F. Boctor, G. Laporte. An improved petal heuristic for the vehicle routing problem. *Journal of the Operational Research Society* 1996;47:329–336.
- [20] M. Reimann, K. Doerner, R. F. Hartl, Insertion based Ants for Vehicle Routing Problems with Backhauls and Time Windows, Springer LNCS 2463, Berlin/Heidelberg 2002, pp 135–147.
- [21] M. Reimann, K. Doerner, R.F. Hartl. D-ants: savings based ants divide and conquer the vehicle routing problem. *Computers & Operations Research* 2004; 31:563–91.
- [22] P. Toth, D. Vigo. The granular tabu search (and its application to the vehicle routing problem). *INFORMS Journal on Computing* 2003;15: 333–48.
- [23] F. Li, B. Golden, E. Wasil. Very large-scale vehicle routing: new test problems, algorithms, and results. *Computers & Operations Research* 2005;32:1165–79.
- [24] I. H. Osman, Metastrategy simulated annealing and Tabu search algorithms for the vehicle routing problem. *Operations Research* 1993;41:421–51.
- [25] B. Bullnheimer, R. F. Hartl, and C. Strauss. An improved ant system algorithm for the vehicle routing problem. *Annals of Operations Research*, 89:319–328, 1999.
- [26] J. Berger, M. Barkaoui, A Hybrid genetic algorithm for the capacitated vehicle routing problem. In: *Proceedings of the genetic and evolutionary computation conference*, Chicago; 2003. p. 646–56.
- [27] C. Prins, A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research* 2004;31: 1985–2002.
- [28] B.M. Baker, M.A. Ayechev, A genetic algorithm for the vehicle routing problem. *Computers & Operations Research* 2003;30:787–800.
- [29] D. Mester, O. Bräysy. Active guided evolution strategies for large scale vehicle routing problem. Working paper, University of Haifa, Israel; 2004.
- [30] B. Bullnheimer, R.F. Hartl, C. Strauss. Applying the ant system to the vehicle routing problem. In: Voss S, Martello S, Osman IH, Roucairol C, editors. *Metaheuristics: advances and trends in local search paradigms for optimization*. Boston: Kluwer; 1998. p. 109–20.
- [31] B. Bullnheimer, R.F. Hartl, C. Strauss. An improved ant system for the vehicle routing problem. *Annals of Operations Research* 1999;89:319–28.
- [32] M. Reimann, K. Doerner, R.F. Hartl. D-ants: savings based ants divide and conquer the vehicle routing problem. *Computers & Operations Research* 2004; 31:563–91.
- [33] M. Gendreau, G. Laporte, J.Y. Potvin. Metaheuristics for the capacitated VRP. In: Toth P, Vigo D, editors. *The vehicle routing problem*. Philadelphia: SIAM; 2001. p. 129–54.
- [34] J.F. Cordeau, M. Gendreau, A. Hertz, G. Laporte, J.S. Sormany, New heuristics for the vehicle routing problem. In: Langevin A, Riopel D, editors. *Logistics systems: design and optimization*. Boston: Kluwer.
- [35] S. Salhi, S.R. Thangiah and F. Rahman, (1998). A genetic clustering method for the multi-depot vehicle routing problem. In G. D. Smith, N. C. Steel, & R. F. Albrecht (Eds.), *ICANNGA '97 Vienna* (pp. 234–237). New York: Springer.
- [36] D. Pisinger, S. Ropke, A general heuristic for vehicle routing problems, *Computers and Operations Research* 34 (2007) 2403–2435.
- [37] S. R. Thangiah, and P.L. Nygard, School bus routing using genetic algorithms. In *Proceedings of the SPIE conference on applications of artificial intelligence X: Knowlarc based systems, 1992* (pp. 387–397), Orlando, FL.
- [38] S.R. Thangiah, P.L. Nygard and P.L. Juell, GIDEON. (1991). A genetic algorithm system for vehicle routing with time windows. In *Proceedings of the seventh IEEE conference on artificial intelligence applications* (pp. 322–328), Miami, FL.
- [39] S.R. Thangiah, R. Vinayagamoorthy and A. V. Gubbi, (1993). Vehicle routing with time deadlines using genetic and local algorithms. In *Proceedings of the Fifth International Conference on Genetic Algorithms* (pp. 506–515). Los Altos CA: Morgan Kaufmann.
- [40] J.Y. Potvin, D. Dube and C. Robillard, (1996). A hybrid approach to vehicle routing using neural networks and genetic algorithms. *Applied Intelligence*, 6, 241–252.
- [41] J.Y. Potvin, C. Duhamel, and F. Guertin, (1996). A genetic algorithm for vehicle routing with backhauling. *Applied Intelligence*, 6, 345–355.
- [42] M. Reimann, S. Shtovba and E. Nepomuceno, A hybrid ACO-GA approach to solve Vehicle Routing Problems, *Student Papers of the Complex Systems Summer School, Budapest, Santa Fe Institute*, (2001).
- [43] N. ChristoCdes, Vehicle routing. In: Lawler EL et al., editors. *The traveling salesman problem*. Chicester: Wiley, 1985.
- [44] G. Laporte, The vehicle routing problem: an overview of exact and approximate algorithms. *European Journal of Operational Research* 1992; 59:345–58.
- [45] M. Dorigo and G. Di Caro. The Ant Colony Optimization meta-heuristic. In D. Corne, M. Dorigo,

- and F. Glover, editors, *New Ideas in Optimization*, pages 11–32. McGraw Hill, London, UK, 1999.
- [46] M. Dorigo, G. Di Caro, and L. M. Gambardella. Ant algorithms for discrete optimization. *Artificial Life*, 5(2):137–172, 1999.
- [47] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, New York, NJ, 1999.
- [48] C. Blum, and A. Roli, *Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison*, ACM Computing Surveys, Vol. 35, No.3, September 2003, pp.268-308.
- [49] L. M. Gambardella and M. Dorigo. Ant Colony System hybridized with a new local search for the sequential ordering problem. *INFORMS Journal on Computing*, 12(3):237–255, 2000.
- [50] M. Dorigo, M. Middendorf, and T. Stützle, editors. Abstract proceedings of ANTS2000 – From Ant Colonies to Artificial Ants: A Series of International Workshops on Ant Algorithms. Université Libre de Bruxelles, 7–9 September 2000.
- [51] M. Dorigo, V. Maniezzo, and A. Colomi. The Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on systems, Man, and Cybernetics – Part B*, 26(1):29–41, 1996.
- [52] M. Dorigo, C. Blum Ant colony optimization theory: A survey, *Theoretical Computer Science* 344 (2005) 243 – 278.
- [53] D. Corne, M. Dorigo and F. Glover, editors, *New Ideas in Optimization*. McGraw-Hill, 1999.
- [54] R. Schoonderwoerd, O. Holland, J. Bruten, and L. Rothkrantz, “Ant-based load balancing in telecommunications networks,” *Adaptive Behavior*, vol. 5, no. 2, pp. 169–207, 1996.
- [55] G. Di Caro and M. Dorigo, “AntNet: Distributed stigmergetic control for communications networks,” *Journal of Artificial Intelligence Research*, vol. 9, pp. 317–365, 1998.
- [56] K. M. Sim and W. H. Sun, “Ant colony optimization for routing and load-balancing: Survey and new directions,” *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 2003.
- [57] G. Di Caro, F. Ducatelle, and L. M. Gambardella, “AntHocNet: An adaptive nature-inspired algorithm for routing in mobile ad hoc networks,” *European Transactions on Telecommunications*, vol. 16, no. 5, pp. 443–455, 2005.
- [58] W. J. Gutjahr, S-ACO: An ant-based approach to combinatorial optimization under uncertainty, *Proc. 4th Int. Workshop on Ant Algorithms and Swarm Intelligence*, Lecture Notes in Computer Science, Vol. 3172, Springer, Berlin, 2004, p. 238.
- [59] R. Montemanni, L. M. Gambardella, A. E. Rizzoli, and A. V. Donati, “Ant colony system for a dynamic vehicle routing problem,” *Journal of Combinatorial Optimization*, vol. 10, pp. 327–343, 2005.
- [60] M. Birattari, P. Balaprakash, and M. Dorigo, “ACO/F-Race: Ant colony optimization and racing techniques for combinatorial optimization under uncertainty,” in *MIC 2005: The 6th Metaheuristics International Conference*, K. F. Doerner et al., Eds. Vienna, Austria: University of Vienna, Department of Business Administration, 2005, pp. 107–112.
- [61] K. F. Doerner, W. J. Gutjahr, R. F. Hartl, C. Strauss, and C. Stummer, “Pareto ant colony optimization in multiobjective project portfolio selection with ILP preprocessing,” *European Journal of Operational Research*, vol. 171, no. 3, pp. 830–841, 2006.
- [62] M. Manfrin, M. Birattari, T. Stützle, and M. Dorigo, “Parallel ant colony optimization for the traveling salesman problem,” in *Proceedings of ANTS 2006*, ser. LNCS, M. Dorigo et al., Eds., vol. 4150. Springer Verlag, 2006, pp. 224–234.
- [63] S. Benker, K. F. Doerner, R. F. Hartl, G. Kiechle, and M. Lucka, “Communication strategies for parallel cooperative ant colony optimization on clusters and grids,” in *Complimentary Proceedings of PARA’04 Workshop on State-of-the-Art in Scientific Computing*, 2005, pp. 3–12.
- [64] K. Socha, “ACO for continuous and mixed-variable optimization,” in *Proc. Ant Colony Optimization and Swarm Intelligence*, 4th International Workshop, ANTS 2004, ser. LNCS, M. Dorigo et al., Eds., vol. 3172. Springer Verlag, 2004, pp. 25–36.
- [65] K. Socha and M. Dorigo, “Ant colony optimization for continuous domains,” *European Journal of Operational Research*, 2006, Unpublished.
- [66] C. García-Martínez, O. Cordon, F. Herrera, “A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP,” *European Journal of Operational Research* 180 (2007) 116–148.
- [67] C. Blum, Beam-ACO—hybridizing ant colony optimization with beam search: an application to open shop scheduling, *Christian Blum, Computers & Operations Research* 32 (2005) 1565–1591.
- [68] M. Reimann, M. Laumanns, Savings based ant colony optimization for the capacitated minimum spanning tree problem, *Computers & Operations Research* 33 (2006) 1794–1822.
- [69] J.B. Kruskal, On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society* 1956;7:48–50.
- [70] R.C. Prim. Shortest connection networks and some generalizations. *Bell Systems Technology Journal* 1957;36:1389–401.
- [71] B. Bullnheimer, R.F. Hartl, C.H. Strauss. A new rank based version of the ant system: a computational study. *Central European Journal of Operations Research* 1999;7(1):25–38.
- [72] Y. Rochat, E. Taillard, Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics* 1995;1:147–67.
- [73] J.-F. Cordeau, Gendreau Mand Laporte G (1997). “A tabu search heuristic for the periodic and multi-depot vehicle routing problems”. *Networks* 30: 105–119.
- [74] M. Gendreau, A. Hertz, and G. Laporte, “A tabu search heuristic for the vehicle routing problem”, *Management Science*. v.40 n.10, p.1276-1290, Oct. 1994.