Journal of Mathematical Modelling and Algorithms (2006) **5:** 91–110 © Springer 2005
DOI: 10.1007/s10852-005-9033-y

Hybrid Metaheuristics for the Vehicle Routing Problem with Stochastic Demands

LEONORA BIANCHI[1,★], MAURO BIRATTARI[2], MARCO CHIARANDINI[3], MAX MANFRIN[2], MONALDO MASTROLILLI[1], LUIS PAQUETE[3], OLIVIA ROSSI-DORIA[4] and TOMMASO SCHIAVINOTTO[3]
[1]*IDSIA, Lugano, Switzerland. e-mail:leonora@idsia.ch*
[2]*IRIDIA, Université Libre de Bruxelles, Brussels, Belgium.*
[3]*Intellectics Group, TU Darmstadt, Germany.*
[4]*School of Computing, Napier University, Scottland, UK.*

Abstract. This article analyzes the performance of metaheuristics on the vehicle routing problem with stochastic demands (VRPSD). The problem is known to have a computationally demanding objective function, which could turn to be infeasible when large instances are considered. Fast approximations of the objective function are therefore appealing because they would allow for an extended exploration of the search space. We explore the *hybridization* of the metaheuristic by means of two objective functions which are surrogate measures of the exact solution quality. Particularly helpful for some metaheuristics is the objective function derived from the traveling salesman problem (TSP), a closely related problem. In the light of this observation, we analyze possible extensions of the metaheuristics which take the hybridized solution approach VRPSD-TSP even further and report about experimental results on different types of instances. We show that, for the instances tested, two hybridized versions of iterated local search and evolutionary algorithm attain better solutions than state-of-the-art algorithms.

Mathematics Subject Classifications (2000): 68T20, 90C27, 90C59, 90B06, 90C15.

Key words:: objective function approximation, local search, a priori tour.

1. Introduction

Vehicle routing problems (VRPs) concern the transport of items between depots and customers by means of a fleet of vehicles. Solving a VRP means to find the best set of routes servicing all customers and respecting the operational constraints, such as vehicles capacity, time windows, driver's maximum working time. VRPs are a key issue in supply-chain and distribution systems today, and they are becoming increasingly complex. For this reason there is an ever increasing interest in routing models that are dynamic, stochastic, rich of constraints, and thus have more and more complex objective functions.

★ Corresponding author.

Models that focus particularly on the *stochasticity* of information are mainly known in the literature as Stochastic VRPs (SVRPs), and the problem we are addressing in this paper belongs to this class. In SVRPs elements of the problem such as the set of customers visited, the customers demands, or the travel times, are modeled as stochastic variables with known probability distributions, and the objective function is usually the expected cost of the planned routes. Note, however, that in SVRPs one needs to specify not only the concept of 'planned routes,' but also the way planned routes are to be modified in response to the realization of the stochastic information.

One common feature of SVRPs is that they all have at least one deterministic counterpart, which is the VRP that one obtains by considering zero-variance probability distributions for the stochastic elements of the problem. SVRPs are thus NP-hard problems, like most VRPs. An important point that increases the difficulty of SVRPs is that they have an objective function (expected cost) which is much more computationally expensive than their deterministic counterparts. For this reason, a key issue in solving SVRPs by heuristics and metaheuristics is the use of fast and effective objective function approximations that may accelerate the search process. Due to the analogy between stochastic and deterministic VRPs, a reasonable choice for the objective function approximation of a given SVRP is the objective function of corresponding, or similar, deterministic problems.

In this paper, we investigate the use of objective function approximations derived from deterministic problems in the context of the vehicle routing problem with stochastic demands (VRPSD). This is an NP-hard problem, and despite the fact that it has a quite simple formulation, it arises in practice in many real world situations. One example is garbage collection, where it is indeed impossible to know *a priori* how much garbage has to be collected at each place. Another example where the demand is uncertain is the delivery of petrol to petrol stations. In fact, when a customer issues the order it is still unknown how much he will sell in the time between the order and the delivery.

In the VRPSD problem one vehicle of finite capacity leaves from a depot with full load, and has to serve a set of customers whose exact demand is only known on arrival at the each customer location. A planned route in this context is very simple: a tour starting from the depot and visiting all customers exactly once; this is also called *a priori* tour, and it will be addressed as such in the remainder of the paper. The *a priori* tour is a sort of skeleton that fixes the order in which customers will be served, but the actual route the vehicle would travel would include return trips to the depot for replenishments when needed. The points at which return trips are performed are, in general, stochastic. The objective function to be minimized is the expected cost of the *a priori* tour.

Due to the nature of the *a priori* tour, a feasible solution for the VRPSD may also be seen as a feasible solution for a traveling salesman problem (TSP) on the set of customers (depot included). Moreover, if the vehicle has infinite capacity, the consequent VRPSD is, in fact, a TSP. Due to these analogies, a natural approximation of the VRPSD objective function is the length of the *a priori* tour.

In this paper we consider basic implementations of five metaheuristics: simulated annealing [23], tabu search [14], iterated local search [25], ant colony optimization [10] and evolutionary algorithms [1]. Our main goal is to test the impact on metaheuristics of interleaving the exact VRPSD objective function with the *a priori* tour length as an approximation of it. This mixture changes the *search landscape* during the search for good quality solutions and can be seen as innovative type of hybridization of metaheuristic's search process that has not been yet explored in the literature. In particular, we investigate two types of hybridization: first, we consider a local search algorithm (OrOpt) for which a quite good approximation for the exact VRPSD objective function is available, and we compare metaheuristics using this underlined local search by applying both VRPSD approximation and TSP approximation. Second, we further exploit the TSP analogy, by choosing the 3-opt local search operator, which is very good for the TSP, but for which there is no immediate VRPSD approximation.

The remainder of the paper is organized as follows. In Section 2 we give the formal description of the VRPSD, we describe in detail the objective function, the state of the art about the VRPSD, and the relevant aspects of generating a benchmark of instances for this problem, taking into account the existing literature. Section 3 describes at high level the metaheuristics and the other algorithms analyzed. Section 4 reports details about tested instances, parameters used for the metaheuristics, computation times allowed. Sections 5 and 6 describe the computational experiments on the first type of hybridization (the use of TSP objective function in OrOpt) and on the second type of hybridization (the use of the 3-opt local search with the TSP objective function), respectively. Section 7 summarizes the main conclusions that can be drawn from the experimental results.

2. The Vehicle Routing Problem with Stochastic Demand

The VRPSD is defined on a complete graph $G = (V, A, D)$, where $V = \{0, 1, \ldots, n\}$ is a set of nodes (customers) with node 0 denoting the depot, $A = \{(i, j) : i, j \in V,\ i \neq j\}$ is the set of arcs joining the nodes, and $D = \{d_{ij} : i, j \in V, i \neq j\}$ are the travel costs (distances) between nodes. The cost matrix D is symmetric and satisfies the triangular inequality. One vehicle with capacity Q has to deliver goods to the customers according to their demands, minimizing the total expected distance traveled, and given that the following assumptions are made. Customers' demands are stochastic variables ξ_i, $i = 1, \ldots, n$ independently distributed with known distributions. The actual demand of each customer is only known when the vehicle arrives at the customer location. It is also assumed that ξ_i does not exceed the vehicle's capacity Q, and follows a discrete probability distribution $p_{ik} = \text{Prob}(\xi_i = k)$, $k = 0, 1, 2, \ldots, K \leq Q$. A feasible solution to the VRPSD is a permutation of the customers $s = (s(0), s(1), \ldots, s(n))$

starting at the depot (that is, $s(0) = 0$), and it is called *a priori* tour. The vehicle visits the customers in the order given by the *a priori* tour, and it has to choose, according to the actual customer's demand, whether to proceed to the next customer or to go to depot for restocking. Sometimes the choice of restocking is the best one, even if the vehicle is not empty, or if its capacity is bigger than the expected demand of the next scheduled customer; this action is called 'preventive restocking.' The goal of preventive restocking is to avoid the risk of having a vehicle without enough load to serve a customer and thus having to perform a back-and-forth trip to the depot for completing the delivery at the customer.

The expected distance traveled by the vehicle (that is, the objective function), is computed as follows. Let $s = (0, 1, \dots, n)$ be an *a priori* tour. After the service completion at customer j, suppose the vehicle has a remaining load q, and let $f_j(q)$ denote the total expected cost from node j onward. With this notation, the expected cost of the *a priori* tour is $f_0(Q)$. If L_j represents the set of all possible loads that a vehicle can have after service completion at customer j, then, $f_j(q)$ for $q \in L_j$ satisfies

$$f_j(q) = \text{Minimum}\{f_j^p(q), f_j^r(q)\}, \tag{1}$$

where

$$\begin{aligned} f_j^p(q) = d_{j,j+1} &+ \sum_{k:k\le q} f_{j+1}(q-k)p_{j+1,k} \\ &+ \sum_{k:k>q} [2d_{j+1,0} + f_{j+1}(q+Q-k)]p_{j+1,k}, \end{aligned} \tag{2}$$

$$f_j^r(q) = d_{j,0} + d_{0,j+1} + \sum_{k=1}^{K} f_{j+1}(Q-k)p_{j+1,k}, \tag{3}$$

with the boundary condition $f_n(q) = d_{n,0},\ q \in L_n$. In (2–3), $f_j^p(q)$ is expected cost corresponding to the choice of proceeding directly to the next customer, while $f_j^r(q)$ is the expected cost in case preventive restocking is chosen. As shown by Yang et al. in [33], the optimal choice is of threshold type: given the *a priori* tour, for each customer j there is a load threshold h_j such that, if the residual load after serving j is greater than or equal to h_j, then it is better to proceed to the next planned customer, otherwise it is better to go back to the depot for preventive restocking. The computation of $f_0(Q)$ runs in $O(nKQ)$ time; the memory required is $O(nQ)$, if one is interested in memorizing all intermediate values $f_j(q)$, for $j = 1, 2, \dots, n$ and $q = 0, 1, \dots, Q$, and $O(Q)$ otherwise. Procedure 1 is an implementation of the recursion (2–3) for the computation of $f_0(Q)$ and of the thresholds. According to the above definition, the VRPSD is a single-vehicle routing problem. Note that there would be no advantage in considering a multiple-vehicle problem by allowing multiple

Procedure 1 Computation of the VRPSD objective function $f_0(Q)$

```
for $(q = Q, Q-1, \ldots, 0)$ do
  $f_n(q) = d_{n,0}$
  for $(j = n-1, n-2, \ldots, 1)$ do
    compute $f_j^r$ (by means of equation (3))
    for $(q' = Q, Q-1, \ldots, 0)$ do
      compute $f_j^p(q')$ (by means of equation (2))
      compare $f_j^r$ and $f_j^p(q')$ for finding the threshold $h_j$
      compute $f_j(q')$ (by means of equation (1))
    end for
  end for
end for
compute $f_0(Q)$
return $f_0(Q)$
```

a priori tours, since, as proved by Yang et al. [33], the optimal solution is always a single tour.

The literature about the VRPSD and SVRPs in general is quite rich. Formulations of SVRPs include the Traveling Salesman Problem with Stochastic Customers (TSPSC), the Traveling Salesman Problem with Stochastic Travel Times (TSPST), the Vehicle Routing Problem with Stochastic Customers (VRPSC), the Vehicle Routing Problem with Stochastic Customers and Demands (VRPSCD). For a survey on the early approaches to these problems, see [12] and [4]; for a more recent survey, especially on mathematical programming approaches used for SVRPs, see [22]. In the following we summarize the main contributions to solve the VRPSD and similar problems, relevant to this paper.

- Jaillet [17, 18] and Jaillet-Odoni [19] derive analytic expressions for the computation of the expected length of a solution for the Probabilistic Traveling Salesman Problem and variations of it;
- Bertsimas [2] proposes the **cyclic** heuristic for the VRPSD, by adapting to a stochastic framework one of the heuristics presented by Haimovitch and Rinnooy Kan [16] in a deterministic context; later, Bertsimas et al. [3] improve this heuristic by applying dynamic programming, to supplement the *a priori* tour with rules for selecting returns trips to the depot, similarly to the preventive restocking strategy; their computational experience suggests that the two versions of the cyclic heuristic provide good quality solutions when customers are randomly distributed on a square region;
- Gendreau, Laporte and Séguin [11] present an exact stochastic integer programming method for the VRPSCD (the same method can be applied to the VRPSD as well); by means of the integer L-shaped method [24] they solve instances with up to 46 and 70 customers and 2 vehicles, for the VRPSCD and VRPSD, respectively; in [13], they also develop a tabu search algorithm

called TABUSTOCH for the same problem; this algorithm is to be employed when instances become too large to be solved exactly by the L-shaped method;

- Teodorović and Pavković [31] propose a Simulated Annealing algorithm to solve the multi-vehicle VRPSD, with the assumption that no more than one route failure is allowed during the service of each vehicle.
- Gutjahr [15] applies S-ACO to the Traveling Salesman Problem with Time Windows, in case of stochastic service times. S-ACO is a simulation-based Ant Colony Optimization algorithm, that computes the expected cost of a solution (the objective function), by Monte Carlo sampling.
- Secomandi [27, 28] applies Neuro Dynamic Programming techniques to the VRPSD; he addresses the VRPSD with a re-optimization approach, where after each new exact information about customers demand is updated, the *a priori* tour planned on the not yet served customers is completely re-planned; this approach may find solutions with a lower expected value with respect to the preventive restocking strategy, but it is much more computationally expensive; moreover, the *a priori* tour may be completely different from the actual tour followed by the vehicle, and this situation is often seen as a disadvantage by companies;
- Yang et al. [33] investigate the single- and multi-vehicle VRPSD; the latter is obtained by imposing that the expected distance traveled by each vehicle does not exceed a given value; the authors test two heuristic algorithms, the route-first-cluster-next and the cluster-first-route-next, which separately solve the problem of clustering customers which must be served by different vehicles and the problem of finding the best route for each cluster; both algorithms seem to be efficient and robust for small size instances, as shown by comparisons with branch-and-bound solutions to instances with up to 15 customers; the authors also adapt to the stochastic case (both single- and multi-vehicle VRPSD) the OrOpt local search due to Or [26], by proposing a fast approximation computation for the change of the objective function value of a solution modified with a local search move.

The OrOpt local search and objective function approximation are used in the present paper as building blocks of our metaheuristics, therefore we will describe them in detail in the following subsection.

2.1. THE OrOpt LOCAL SEARCH

A basic move of the OrOpt local search, as suggested by Yang et al. in [33], works as follows. Given a starting *a priori* tour, sets S_k of k consecutive customers with $k \in \{1, 2, 3\}$ are moved from one position to another in the tour, like in Figure 1. In the following we describe the two types of approximation schemes used for the computation of the move cost. The first one, that we call

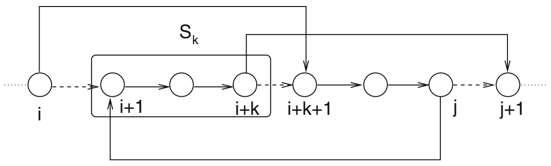

Figure 1. How an *a priori* tour is modified after performing an OrOpt move, where the set of consecutive customers S_k (here, $k = 3$) is moved forward in the tour.

VRPSD, is the one proposed in [33], and the second one, that we call *TSP*, only computes the length change of the *a priori* tour.

VRPSD approximation scheme. The move cost is computed in two stages: i) compute the saving from extracting the set of customers from the tour; ii) compute the cost of inserting it back somewhere else in the tour. Let i and $i + k + 1$ be the nodes immediately preceding, respectively following, S_k in the tour, and let j be the node immediately after which S_k is to be inserted, as shown in Figure 1. Here, we assume that j is *after* i in the *a priori* tour. Let $f_i(q)$ and $f_{i+k+1}(q)$ be the expected cost-to-go from nodes i, respectively $i + k + 1$ onward before the extraction of S_k. Apply one dynamic programming recursion step starting with cost vector $f_{i+k+1}(\cdot)$ at node $i + k + 1$ back to node i, without considering the sequence S_k. Let $f_i'(\cdot)$ be the resulting cost vector at node i, that is, after extracting S_k from the tour. Then, define the approximate extraction saving as a simple average over q of $f_i(q) - f_i'(q)$. The computation of the approximate insertion cost of S_k between nodes j and $j + 1$ in the tour, is done analogously, if we assume that the insertion point (node j) is after the extraction point (node i). Let $f_j(q)$ be the cost-to-go at node j before inserting S_k, and $f_j''(q)$ be the cost-to-go at node j after inserting the S_k. The total approximate cost of an OrOpt move is computed by subtracting the approximate extraction saving from the approximate insertion cost, as follows

$$\Delta_{\text{VRPSD}} = \frac{\sum_{q=0}^{Q}[(f_j''(q) - f_j(q)) - (f_i(q) - f_i'(q))]}{Q+1}. \tag{4}$$

Note that the cost vectors are assumed to be already available from the computation of the expected cost for the starting tour, thus, they do not need to be computed when evaluating equation (4). The only computations that must be done here are the evaluation of cost vectors $f_{i+1}'(\cdot)$ and $f_j''(\cdot)$, requiring $O(KQ)$ time, and the average of equation (4), requiring $O(Q)$ time. Therefore, with the proposed *VRPSD* approximation, the cost of an OrOpt move can be computed in $O(KQ)$ time. Although it is possible that tours which are worsening with respect to the evaluation function are accepted because recognized as improving by the approximate evaluation, in practice this approximation scheme behave quite well. For a deeper discussion on the issues related with this scheme we refer the reader to the original paper [33].

TSP approximation scheme. In the *TSP* approximation scheme the cost of an OrOpt move coincides with the difference between the length of the tour before the move and after the move:

$$\Delta_{\mathrm{TSP}} = d_{i,i+k+1} + d_{j,i+1} + d_{i+k,j+1} - d_{i,i+1} - d_{i+k,i+k+1} - d_{j,j+1}, \tag{5}$$

where, as before, i and j are the extraction, respectively insertion point of a string of k consecutive customers (see Figure 1). Clearly, Δ_{TSP} is computable in constant time.

The OrOpt neighborhood examination follows the same scheme proposed in [33]. Briefly, all possible sequences of length $k \in \{1, 2, 3\}$ are considered for insertion in a random position of the tour after the extraction point. Then, only the ‘best’ move among those of length k is chosen. The ‘best’ move is the move corresponding to the most negative move cost, which is computed by equation (4) in the *VRPSD* approach and by equation (5) in the *TSP* approach.

2.2. BENCHMARK

In the literature there is no commonly used benchmark for the VRPSD, therefore we have generated our own testbed. We have tried to consider instances which are ‘interesting’ from different points of view, by controlling four factors in the generation of instances: customer position, capacity over demand ratio, variance of the stochastic demand, and number of customers.

Instances may be divided into two groups, uniform and clustered, according to the position of customers. In uniform instances, the position of customers is chosen uniformly at random on a square of fixed size. In clustered instances, coordinates are chosen randomly with normal distributions around a given number of centers. This results in clusters of nodes, a typical situation for companies serving customers positioned in different cities.

The ratio between the total (average) demand of customers and the vehicle’s capacity is an important factor that influences the ‘difficulty’ of a VRPSD instance [11]. The bigger the ratio, the more ‘difficult’ the instance. Here, the vehicle capacity Q is chosen as $Q = \lceil \frac{\text{total average demand} \cdot r}{n} \rceil$, where the parameter r may be approximately interpreted as the average number of served customers before restocking.

Each customer’s demand is an integer stochastic variable uniformly distributed on an interval. The demand interval for each customer i is generated using two parameters: the average demand D_i, and the spread S_i, so that the possible demand values for customer i are the $2S_i + 1$ integers in the interval $[D_i - S_i, D_i + S_i]$. The spread is a measure of the variance of the demand of each customer.

The particular parameters used to generate the test instances for our experiments are reported in Section 4.

3. The Metaheuristics

We aim at an *unbiased* comparison of the performance of five well-known different metaheuristics for this problem. In order to obtain a fair and meaningful analysis of the results, we have restricted the metaheuristic approaches to the use of the common OrOpt local search. In the following we briefly and schematically describe the main principles of each metaheuristic and give the details of their implementations for the VRPSD.

- **Simulated Annealing (SA)**

```
Determine initial candidate solution $s$
Set initial temperature $T = T_i$ according to annealing schedule
While termination condition not satisfied:
    Probabilistically choose a neighbor $s'$ of $s$
    If $s'$ satisfies probabilistic acceptance criterion (depending on $T$):
        $s := s'$
    Update $T$ according to annealing schedule
```

The initial temperature T_i is given by the average cost of a sample of 100 solutions of the initial tour multiplied by μ; every $\psi \cdot n$ iterations the temperature is updated by $T \leftarrow \alpha \cdot T$ (standard geometric cooling); after $\rho \cdot \psi \cdot n$ iterations without improvement, the temperature is increased by adding T_i to the current value; the solution considered for checking improvements is the best since the last re-heating.

- **Tabu Search (TS)**

```
Determine initial candidate solution $s$
While termination criterion is not satisfied:

    Choose the best neighbor $s'$ of $s$ that is either non-tabu or
        satisfies the aspiration criterion, and set $s := s'$
    Update tabu attributes based on $s'$
```

The neighborhood of the current solution s is explored. The non-tabu neighbors are considered for the selection of the next current solution. If the value of a tabu neighbor is better than the best found solution, then this neighbor is also considered for selection (aspiration criterion). The best considered neighbor, i.e., the one with the lowest value, is selected. In order to avoid cycling around the same set of visited solutions, we found convenient to have a *variable neighborhood*: for any current solution s, instead of considering the whole neighborhood, we choose a subset of it according to a probability distribution. We experimentally verified that the used variable neighborhood is able to avoid cycles and to explore a larger part of the search space.

– **Iterated Local Search (ILS)**

```
Determine initial candidate solution $s$
Perform local search on $s$
While termination criterion is not satisfied:
    $r := s$
    Perform perturbation on $s$
    Perform local search on $s$
    Based on acceptance criterion, keep $s$ or revert to $s := r$
```

The perturbation consists in a sampling of n neighbors according to the 2-opt exchange neighborhood [20]; each new solution is evaluated by exact cost function (Procedure 1) and if a solution is found that has cost smaller than the best solution found so far plus ε, the sampling ends; otherwise, the best solution obtained during the sampling is returned; the acceptance criterion keeps s if it is the best solution found so far.

– **Ant Colony Optimisation (ACO)**

```
Initialise weights (pheromone trails)
While termination criterion is not satisfied:
    Generate a population $sp$ of solutions by a randomised
        constructive heuristic
    Perform local search on $sp$
    Adapt weights based on $sp$
```

Pheromone trails are initialized to τ_0; sp solutions are generated by a constructive heuristic and refined by local search; a *global update rule* is applied r times; sp solutions are then constructed by using information stored in the pheromone matrix; after each construction step a *local update rule* is applied to the element $\tau_{i,j}$ corresponding to the chosen customer pair: $\tau_{i,j} = (1 - \psi) \cdot \tau_{i,j} + \psi \cdot \tau_0$, with $\psi \in [0,1]$; after local search, weights are again updated by the global update rule $\tau_{i,j} = (1 - \rho) \cdot \tau_{i,j} + \rho \cdot \frac{q}{C^{\text{bs}}}$, with $\rho \in [0,1]$ and C^{bs} the cost of the *best-so-far* solution (note that heuristic information is only used in the initialization phase).

– **Evolutionary Algorithm (EA)**

```
Determine initial population $sp$
While termination criterion is not satisfied:
    Generate a set $spr$ of solutions by recombination
    Generate a set $spm$ of solutions from $spr$ by mutation
```

```
        Perform local search on $spm$
        Select population $sp$ from solutions in $sp$, $spr$, and $spm$
```

At each iteration two solutions are chosen among the best ones to generate a new solution *spr* through Edge Recombination [32] (a tour is generated using edges present in both two other tours, whenever possible); the mutation swaps adjacent customers (without considering the depot) with probability p_m; finally, the solution improved by local search replaces the worst solution in the population.

The initial solution(s) for all metaheuristics are obtained by the Farthest Insertion constructive heuristic [21]; it builds a tour by choosing as next customer the *not yet visited* customer which is farthest from the current one. Here, we consider a randomized version of this heuristic (RFI) which picks the first customer at random, and after the tour has been completed, shifts the starting customer to the depot.

In order to use a reference algorithm for comparison among metaheuristics (see Section 4), we also implemented a simple random restart algorithm which uses the RFI heuristics *plus* local search and restarts every time a local optimum is found, until the termination condition is reached; the best solution found among all the restarts is picked as the final solution; we call such algorithm RR.

4. Experimental Setup

In this article we report two computational experiments with two different goals: i) we analyze metaheuristics performance to test the hypothesis of the relation between the TSP and VRPSD; ii) we study the performance of enhanced versions of the best algorithms found in the first set of experiments. Moreover, we relate these results with parameters of the instance.

We consider instances with 50, 100 and 200 customers and with customers uniformly distributed or grouped in clusters. In uniform instances, the position of customers is chosen uniformly at random in the square $[0,99]^2$. Clustered instances are generated with two clusters, with centers randomly chosen in the square $[0,99]^2$. The variance of the normal distribution used to generate customers coordinates around the centers is equal to $(0.8/\sqrt{n}) \cdot (max.coordinate)$. This corresponds to variance values of about 11, 8, and 6, respectively for instances with 50, 100, and 200 customers. The position of the depot is fixed at (1,1). The average number of customers served before restocking is maintained fixed to 4, thus yielding ratios for total demand over vehicle capacity in the range from 12 to 50. Typical values for this ratio in the VRPSD literature are below 3, however higher values are closer to the needs of real contexts [5]. In all instances, average demands D_i at each customer i are taken with equal probability from $[1,49]$ or $[50,100]$. With regard to demand spread, instead, instances may be divided into two groups: low spread instances, in which each customer i has S_i chosen at random in $[1, 5]$, and high spread instances, in which each customer i has S_i chosen at random in $[10, 20]$. For each combination of size, distribution of customers, and demand spread, 75 instances were generated, making a total of 900 instances.

The metaheuristic parameters were chosen in order to guarantee robust performances over all the different classes of instances; preliminary experiments suggested the following settings:

SA: $\mu = 0.05$, $\alpha = 0.98$, $\psi = 1$, and $\rho = 20$;
TS: $p_{nt} = 0.8$ and $p_t = 0.3$;
ILS: $\varepsilon = \frac{n}{10}$;
ACO: $sp = 5$, $\tau_0 = 0.5$, $\psi = 0.3$, $\rho = 0.1$, $q = 10^7$, and $r = 100$;
EA: $sp = 10$, $p_m = 0.5$.

Given the results reported in [6, 7], we decided to only perform one run for each metaheuristic on each instance.[★] The termination criterion for each algorithm was set to a time equal to 30, 120 or 470 sec. for instances respectively of 50, 100 or 200 customers. Experiments were performed on a cluster of 8 PCs with AMD Athlon(tm) XP 2800+ CPU running GNU/Linux Debian 3.0 OS, and all algorithms were coded in C++ under the same development framework.

In order to compare results among different instances, we normalized results with respect to the performance of RR. For a given instance, we denote as c_{MH} the cost of the final solution of a metaheuristic MH, c_{RFI} the cost of the solution provided by the RFI heuristic, and C_{RR} the cost of the final solution provided by RR; the normalized value is then defined as

$$\text{Normalized Value for MH} = \frac{c_{MH} - c_{RR}}{c_{RFI} - c_{RR}}. \tag{6}$$

Besides providing a measure of performance independent from different instance hardness, this normalization method gives an immediate evaluation of the minimal requirement for a metaheuristic; it is reasonable to request that a metaheuristic performs at least better than RR within the computation time under consideration.

5. First Hybridization: Using Approximate Move Costs in Local Search

The main goal of this first experiment is to see whether approximating the exact but computationally demanding objective with the fast computing length of the *a*

[★] In [6] it is formally proved that if a total of N runs of a metaheuristic can be performed for estimating its expected performance, the *best* unbiased estimator, that is, the one with the least variance, is the one based on *one single run* on N randomly sampled (and therefore typically distinct) instances.

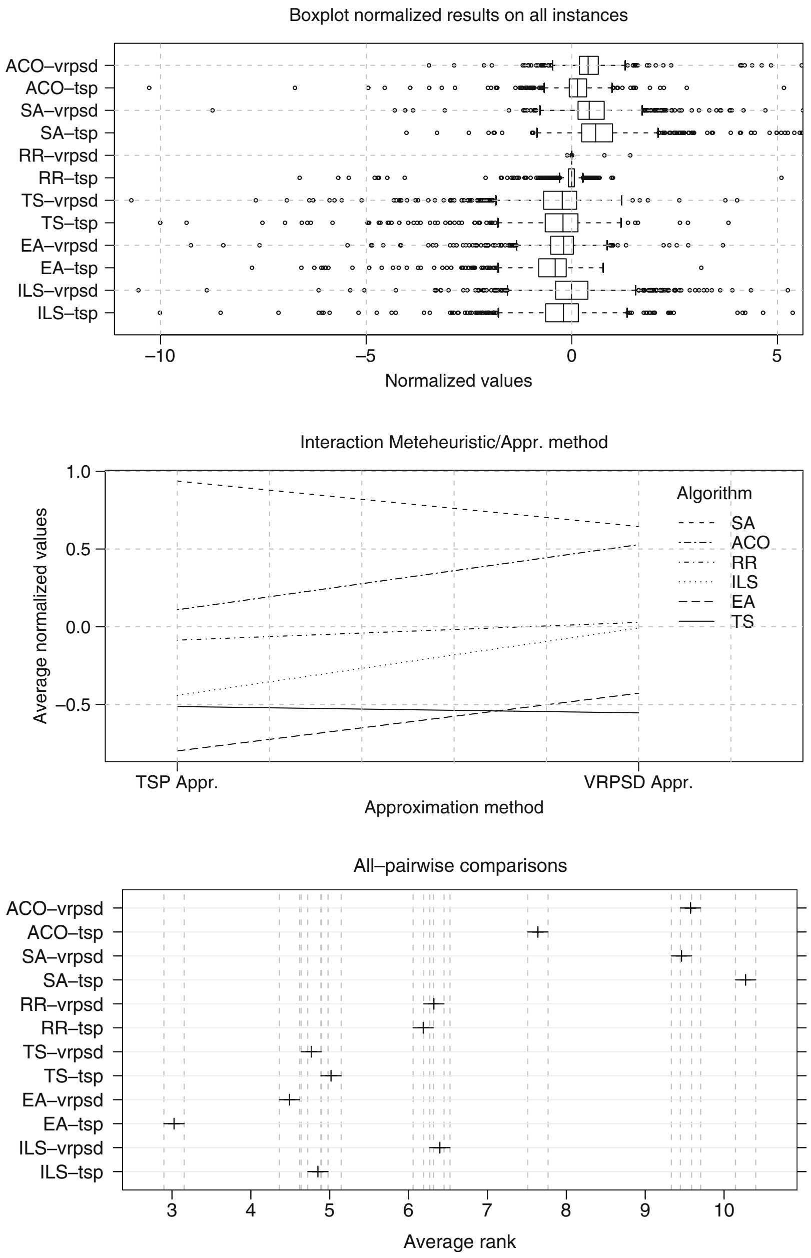

Figure 2. Aggregate results over all 900 instances. From top to bottom: boxplot of normalized results; interaction plot for the two factors: metaheuristic and objective function approximation scheme; error bars plot for simultaneous confidence intervals in the all-pairwise comparison. The boxplot is restricted to values in $[-10.5, 5]$, few outliers fell outside this interval. In the error bar plot, ranks range from 1 to 12, the number of algorithms considered in the experiment.

priori tour is convenient or not. Our hypothesis is that the speedup due to the use of a fast approximation of the objective is an advantage especially during the phase of local search, when many potential moves must be evaluated before one is chosen.

In order to test the advantage of a speedup across the metaheuristics, we apply to each of them the OrOpt local search described in Section 2.1, and we test two versions for each metaheuristic according to the type of approximation scheme used in the local search, *VRPSD-approximation* or *TSP-approximation*. This set up allows to use statistical techniques for a systematic experimental analysis. In particular, we consider a two-way factor analysis, where metaheuristics and type of objective function are factors and instances are considered as blocks [9].

The first plot of Figure 2 is the boxplot of the results over all instances after the normalization. Negative values indicate an improvement over random restart and the larger is the absolute value the larger is the improvement. It emerges that, in average, ILS, EA and TS are able to do better than RR, while SA and ACO perform worse. We check whether these results are also statistically significant. The assumptions for a parametric analysis are not met, hence we rely on non-parametric methods.

The central plot of Figure 2 shows the interaction between the two factors, metaheuristic and approximation scheme. Interaction plots give an idea of how different combinations of metaheuristic and approximation scheme affect the average normalized result. The lines join the average value for each metaheuristic. If lines are not parallel it means that there is an *interaction effect*, that is, metaheuristics perform differently with different approximation scheme. From the plot, we see that a certain interaction effect is present between the two factors, hence, it would be appropriate to report the effects of one factor separately for each level of the other. In the third plot of Figure 2 we report, however, together both VRPSD and TSP approximation schemes, but we distinguish the effects of this factor on each metaheuristic. The plot presents the simultaneous confidence intervals for the all-pairwise comparison of algorithms. Interval widths are obtained by the Friedman two-way analysis of variance by ranks [8], after rejecting the hypothesis that all algorithms perform the same. The difference between two algorithms is statistically significant at a level of confidence of 5% if their intervals do not overlap.

From the interaction and the all-pairwise comparison plots of Figure 2 we can conclude that:

– The improvements of EA, ILS, and TS over RR are statistically significant.
– The presence of an interaction effect between metaheuristic and approximation scheme shows that EA, ILS and ACO perform better with *TSP-approximation* while the opposite result holds for TS and SA. While EA, ILS and ACO use the local search as a black-box, TS and SA employ their own strategy for examining the neighborhood in the local search. This result

indicates that, for becoming competitive, these two methods require a good approximation of the objective function.
- The metaheuristics which perform better are EA, ILS and TS. Furthermore, EA and ILS take significant advantage from *TSP-approximation* scheme.

6. Second Hybridization: Further Exploiting the TSP Analogy

Given the results in the previous section, it is reasonable to investigate what happens if we exploit even more the hybridization based on the TSP objective function. For this purpose, we consider one of the best performing TSP state-of-the-art metaheuristics, and we observe that it is based on iterated local search with the 3-opt local search operator [30]. We, therefore, hybridize the best algorithms determined in the previous section (ILS, EA) with the 3-opt local search for TSP. We do not hybridize TS, instead, since we observed that it exhibits better results with the *VRPSD-approximation* rather than with the *TSP-approximation*. The new algorithms that we consider are the following.

TSP-soa. If the solution found solving the TSP was comparable in quality to those found by our metaheuristics, there would be no point in investigating algorithms specific for the VRPSD. We consider therefore a transformation of the problem into TSP by focusing only on the coordinates of the customers (depot included). We then solve the TSP with a TSP state-of-the-art algorithm [30] and shift the TSP solution found to start with the depot thus yielding a VRPSD solution. This solution is finally evaluated by the VRPSD objective function (Procedure 1). The algorithm for the TSP is an iterated local search algorithm that uses a 3-opt exchange neighborhood, and a double bridge move as perturbation, i.e., it removes randomly four edges from the current tour and adds four other edges, also chosen at random, that close the tour.

ILS-hybrid. It is obtained by modifying the acceptance criterion in *TSP-soa*, that is, the best solution according to Procedure 1 is chosen rather that the best TSP solution.

EA-hybrid. It is derived from *EA-tsp* by replacing the local search with a 3-opt local search based on the TSP objective function. The recombination and mutation operators are maintained unchanged, since these are deemed to be the components that determined the success of EA in Figure 2. The selection operator remains also unchanged, and is based on the VRPSD objective function computed by Procedure 1.

In order to have a comparison with previous methods from the VRPSD literature, we also test the effective CYCLIC heuristic [2], that we implemented as follows:

- solve a TSP over the n customers, plus the depot, by using the *TSP-soa* algorithm;

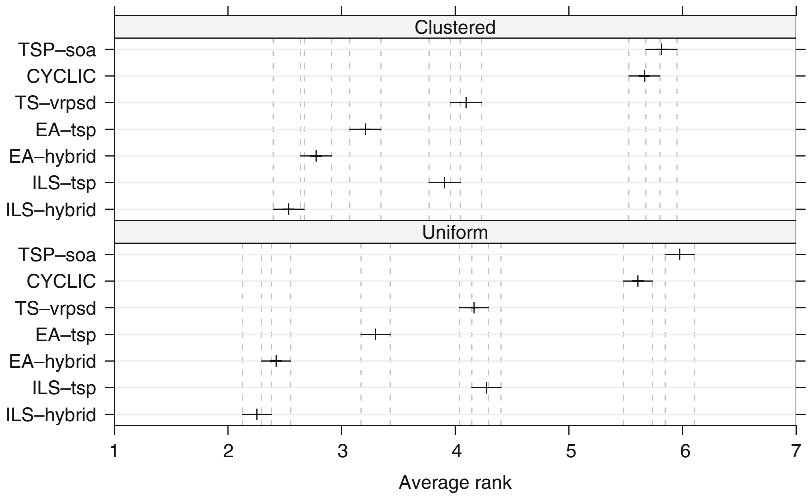

Figure 3. All-pairwise comparisons by means of simultaneous confidence intervals on uniform and clustered instances.

- consider the *n a priori* tours obtained by the cyclic permutations $s_i = (0, i, i+1, \ldots, n, 1, \ldots, i-1), i = 1, 2, \ldots, n$;
- evaluate the *n a priori* tours by the VRPSD objective function (Procedure 1) and choose the best one.

For the comparison of these algorithms we designed an experimental analysis based on a single factor layout with blocks [9]. Since the assumption for parametric tests are again violated, we use the Friedman two-way analysis of variance by ranks.

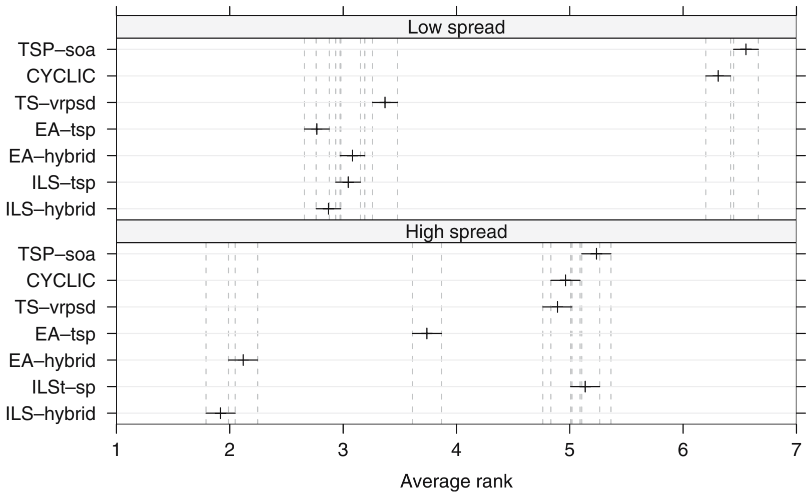

Figure 4. All-pairwise comparisons by means of simultaneous confidence intervals on instances with different demand spread.

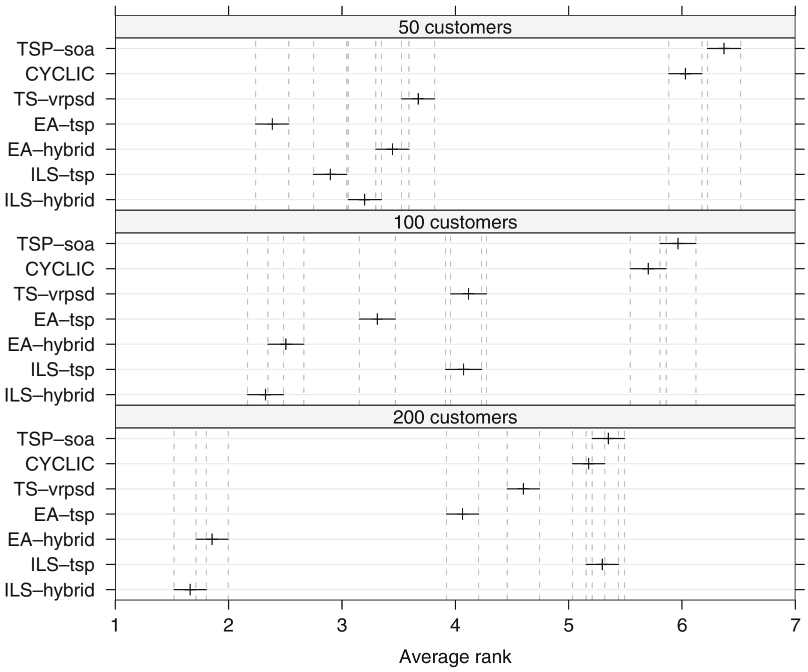

Figure 5. All-pairwise comparisons by means of simultaneous confidence intervals on instances with different number of customers.

In Figures 3–5 we present the simultaneous confidence intervals after rejecting the hypothesis that all algorithms perform the same. We distinguish results by presenting the instances grouped according to the three main features defined in Section 4. Since these the analysis is repeated on the same data, we adjust the 'family-wise' level of confidence for each test dividing it by a factor of three [29].

The main indication arising from the results is that a mere TSP algorithm is not sufficient to produce high quality solutions for the VRPSD instances under any of the circumstances tested. VRPSD problem-specific algorithms, which take into account the stochasticity of the problem, yield always better solutions. Nevertheless, the best performances are achieved by hybridization of TSP and VRPSD approaches. *EA-hybrid* and *ILS-hybrid* in many cases perform statistically better than all other algorithms and, when significant, differences are also practically relevant. On the contrary, the difference between these two approaches is statistically significant only in one case. The comparison with the CYCLIC heuristic indicates that the algorithms we presented improve considerably the solution for VRPSD with respect to previous known approaches.

Considering the characteristics of the instances, the position of customers seems not to have an impact on the rank of algorithms. On the contrary, the different position occupied by the algorithms in the ranks with respect to instance size indicates that the larger the instances the higher is the importance of speed over precision in the evaluation of neighbors in local search. This expected result is exemplified by the worsening of the average rank of *EA-tsp* as size increases.

Finally, spread of demand has also a considerable influence. With small spread it is convenient to consider the costs of 'preventive restockings' and thus the use of Procedure 1 is appropriate. With large spread, on the contrary, the higher uncertainty of the right point in the tour for 'preventive restocking' makes the evaluation of solution by means of Procedure 1 not far from that provided by a TSP evaluation, with the negative impact of a higher computational cost.

7. Conclusions

The main contribution of this paper is the study of the hybridization on five well known metaheuristics with different objective function approximations to solve the VRPSD. In particular, we introduced a *TSP-approximation*, which uses the length of the *a priori* tour as surrogate for the exact but computationally demanding VRPSD evaluation, and we showed its superiority with respect to a known *VRPSD-approximation*.

More precisely, first we considered a local search algorithm (OrOpt) for which a good approximation for the exact VRPSD objective function is available, and we compared metaheuristics using this local search by applying both the *VRPSD-approximation* and the *TSP-approximation*. Secondly, we exploited further the TSP analogy, by choosing the 3-opt local search operator, which is very good for the TSP, but for which there is no immediate VRPSD approximation.

With the first type of hybridization, we have shown that metaheuristics using the local search as a black-box (EA, ILS and ACO) perform better when using the *TSP-approximation*, while metaheuristics that use their own strategy for examining the neighborhood in the local search, perform better with the more precise but more computationally expensive *VRPSD-approximation*. With the second type of hybridization based on the use of the 3-opt local search, we considerably improved the performance of the best metaheuristics for the VRPSD, and we were unable to discover significant differences between the two best algorithms, EA and ILS.

All the metaheuristics implemented found better solutions with respect to the CYCLIC heuristic (which is known from the literature to perform well on different types of instances) and with respect to solving the problem as a TSP. This latter point is important because it demonstrates that the stochasticity of the problem and the finite demand over capacity ratio is not neglectable and that the development of VRPSD-specific algorithms is needed. We proposed a TSP-VRPSD hybrid approach that clearly outperforms the current state-of-the-art.

Acknowledgments

This work was supported by the *Metaheuristics Network*, a Research Training Network funded by the IHP programme of the European Commission (EC), grant HPRN-CT-1999-00106. M. Manfrin also acknowledges the support of ‘COMP2SYS,’ a Marie Curie Early Stage Training Site, funded by the EC through the HRM activity. The information provided is the sole responsibility of the authors and does not reflect the EC’s opinion. The EC is not responsible for any use that might be made of data appearing in this publication. M. Birattari also acknowledges the support of the ‘ANTS’ project, an “Action de Recherche Concertée” funded by the Scientific Research Directorate of the French Community of Belgium.

References

1. Baeck, T., Fogel, D. and Michalewicz Z. (eds.): *Evolutionary Computation 1: Basic Algorithms and Operators*, Institute of Physics, Bristol, UK, 2000.
2. Bertsimas, D. J.: A vehicle routing problem with stochastic demand, *Oper. Res.* **40**(3) (1992), 574–585.
3. Bertsimas, D. J., Chervi, P. and Peterson, M.: Computational approaches to stochastic vehicle routing problems, *Trans. Sci.* **29**(4) (1995), 342–352.
4. Bertsimas, D. J. and Simchi-Levi, D.: A new generation of vehicle routing research: Robust algorithms, addressing uncertainty, *Oper. Res.* **44**(2) (1996), 216–304.
5. Bianchi, L., Birattari, M., Manfrin, M., Mastrolilli, M., Paquete, L., Rossi-Doria, O. and Schiavinotto, T.: Research on the vehicle routing problem with stochastic demand. Technical Report IDSIA-07-04, IDSIA, March 2004.
6. Birattari, M.: On the estimation of the expected performance of a metaheuristic on a class of instances. How many instances, how many runs? Technical Report TR/IRIDIA/2004-01.2, IRIDIA, Université Libre de Bruxelles, Brussels, Belgium, 2004.
7. Birattari, M.: *The Problem of Tuning Metaheuristics, as seen from a machine learning perspective*, PhD thesis, Université Libre de Bruxelles, Brussels, Belgium, 2004.
8. Conover, W. J.: *Practical Nonparametric Statistics*, John Wiley & Sons, New York, New York, 1999.
9. Dean, A. and Voss, D.: *Design and Analysis of Experiments*, Springer, Berlin Heidelberg New York, 1999.
10. Dorigo, M. and Stützle, T.: *Ant Colony Optimization*, MIT, 2004.
11. Gendreau, M., Laporte, G. and Séguin, R.: An exact algorithm for the vehicle routing problem with stochastic demands and customers, *Trans. Sci.* **29**(2) (1995), 143–155.
12. Gendreau, M., Laporte, G. and Séguin, R.: Stochastic vehicle routing, *Eur. J. Oper. Res.* **88** (1996), 3–12.
13. Gendreau, M., Laporte, G. and Séguin, R.: A tabu search heuristic for the vehicle routing problem with stochastic demands and customers, *Oper. Res.* **44**(3) (1996).
14. Glover, F.: Tabu search – Part I, *ORSA J. Comput.* **1**(3) (1989), 190–206.
15. Gutjahr, W.: S-ACO: An ant-based approach to combinatorial optimization under uncertainty, in *Proceedings of ANTS 2004 – Ant Colony Optimization and Swarm Intelligence*, volume 3172 of *Lecture Notes in Computer Science*, Springer, Berlin Heidelberg New York, 2004, pp. 238–249.
16. Haimovitch, M. and Rinnooy Kan, A.: Bounds and heuristics for capacitated routing problems, *Math. Oper. Res.* **10** (1985), 527–542.
17. Jaillet, P.: *Probabilistic Traveling Salesman Problems*, PhD thesis, MIT, Cambridge, Massachusetts, 1985.

18. Jaillet, P.: *A priori* solution of a travelling salesman problem in which a random subset of the customers are visited, *Oper. Res.* **36**(6) (1988), 929–936.
19. Jaillet, P. and Odoni, A.: in B. L. Golden and A. A. Assad (eds.), *Vehicle Routing: Methods and Studies*, chapter The probabilistic vehicle routing problems, Elsevier, Amsterdam, The Netherlands, 1988.
20. Johnson, D. S. and McGeoch, L. A.: The travelling salesman problem: A case study in local optimization, in E. H. L. Aarts and J. K. Lenstra (eds.), *Local Search in Combinatorial Optimization*, Wiley, New York, USA, 1997, pp. 215–310.
21. Johnson, D. S. and McGeoch, L. A.: Experimental analysis of heuristics for the STSP, in G. Gutin and A. Punnen (eds.), *The Traveling Salesman Problem and its Variations*, Kluwer, Dordrecht, The Netherlands, 2002, pp. 369–443.
22. Kenyon, A. and Morton, D. P.: A survey on stochastic location and routing problems, *Cent. Eur. J. Oper. Res.* **9** (2002), 277–328.
23. Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P.: Optimization by simulated annealing, *Science* (4598) (1983), 671–680.
24. Laporte, G. and Louveaux, F.: The integer l-shaped method for stochastic integer programs with complete recourse, *Oper. Res. Lett.* **33** (1993), 133–142.
25. Lourenço, H. R., Martin, O. and Stützle, T.: in F. Glover and G. Kochenberger (eds.), *Handbook of Metaheuristics*, volume 57 of *International Series in Operations Research & Management*, chapter Iterated Local Search, Kluwer, Boston, USA, 2002, pp. 321–353.
26. Or, I.: *Traveling Salesman-Type Combinatorial Problems and their Relation to the Logistics of Blood Banking*, PhD thesis, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, 1976.
27. Secomandi, N.: Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands, *Comput. Oper. Res.* **27**(5) (2000), 1171–1200.
28. Secomandi, N.: A rollout policy for the vehicle routing problem with stochastic demands, *Oper. Res.* **49**(5) (2001), 796–802.
29. Sheskin, D. J.: *Handbook of Parametric and Nonparametric Statistical Procedures*, 2nd edn., Chapman & Hall, 2000.
30. Stützle, T. and Hoos, H.: in P. Hansen and C. Ribeiro (eds.), *Essays and Surveys on Metaheuristics*, chapter Analyzing the Run-time Behaviour of Iterated Local Search for the TSP, Kluwer Academic, Boston, USA, 2002, pp. 589–612.
31. Teodorović, D. and Pavković, G.: A simulated annealing technique approach to the vehicle routing problem in the case of stochastic demand, *Trans. Plan. Tech.* **16** (1992), 261–273.
32. Whitley, D., Starkweather, T. and Shaner, D.: The travelling salesman and sequence scheduling: Quality solutions using genetic edge recombination, in L. Davis (ed.), *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, USA, 1991, pp. 350–372.
33. Yang, W., Mathur, K. and Ballou, R. H.: Stochastic vehicle routing problem with restocking, *Trans. Sci.* **34**(1) (2000), 99–112.