Ann Oper Res (2011) 186:61–81
DOI 10.1007/s10479-011-0895-2

# The orienteering problem with stochastic travel and service times

**Ann M. Campbell · Michel Gendreau · Barrett W. Thomas**

Published online: 26 May 2011


**Abstract** In this paper, we introduce a variant of the orienteering problem in which travel and service times are stochastic. If a delivery commitment is made to a customer and is completed by the end of the day, a reward is received, but if a commitment is made and not completed, a penalty is incurred. This problem reflects the challenges of a company who, on a given day, may have more customers than it can serve. In this paper, we discuss special cases of the problem that we can solve exactly and heuristics for general problem instances. We present computational results for a variety of parameter settings and discuss characteristics of the solution structure.

**Keywords** Orienteering · Stochastic travel times · Variable neighborhood search · Dynamic programming

## 1 Introduction

This paper considers a new variant of the orienteering problem where travel and service times are stochastic. For this problem, from a given set of customers, we want to select a subset, and for this subset, determine an ordering of the subset that maximizes expected profit given a known deadline $D$. For all customers in the ordering that are reached before the deadline, a customer specific reward is received, and for those not reached before the

This work was partially supported by the National Science Foundation through grant number 0237726 (Campbell).

A.M. Campbell · B.W. Thomas (✉)
Dept. of Management Sciences, Tippie College of Business, The University of Iowa, Iowa City, USA
e-mail: barrett-thomas@uiowa.edu

A.M. Campbell
e-mail: ann-campbell@uiowa.edu

M. Gendreau
CIRRELT and MAGI, École Polytechnique de Montréal, Montreal, Canada
e-mail: michel.gendreau@cirrelt.ca

deadline, a customer specific penalty is incurred. We call this the orienteering problem with stochastic travel and service times (OPSTS).

There are many applications where the OPSTS is relevant. Consider any business that involves deliveries or service at its customers' locations. The payments that the customers make for these deliveries or services can be considered a reward. If the number of deliveries or service requests is more than can be accomplished within available business hours, only a subset of customers can be served, at least on that day. The service provider must determine which ones to schedule, and we propose that they should do so in a way that maximizes profit. This requires carefully accounting for how long it takes to travel to and service each customer. However, in most urban environments, the travel and service times between customers can vary greatly and can reasonably be modeled stochastically. Thus, it is generally impossible to know with certainty which of even the scheduled customers can be visited before the deadline. For those customers who are scheduled but who cannot be served, the provider pays a penalty. This penalty may represent a direct payment to the customer, similar to payments made by cable companies when they miss their service appointments, or a loss of goodwill. It is important to recognize there is a cost associated with a failure to make a delivery, and deterministic models cannot consider such a cost. For many technicians, there is no need to return to a depot, or whatever the point of origin is, at the end of the day, so an orienteering, rather than a traveling salesman, model is appropriate.

Formally, let $N = \{1, \dots, n\}$ be a set of customers. Node 0 represents the depot. We assume there is an arc $(i, j)$ between all $i$ and $j$ in $N$. Associated with each $i \in N$ is a reward $r_i$ and penalty $e_i$. This reward $r_i$ is earned by visiting customer $i$ at or before a known deadline $D$ if $i$ is selected to be served. The penalty $e_i$ is incurred if the visit to customer $i$ would occur after a known deadline $D$ if $i$ is selected to be served.

Let $X_{ij}$ be a non-negative random variable representing the time required to traverse arc $(i, j)$. We assume that the distribution on $X_{ij}$ is known for all $i$ and $j$. Let $S_i$ be a non-negative random variable representing the service time at customer $i$. We assume that the distribution of $S_i$ is known for all $i$. Let the random variable $A_i$ be the arrival time at customer $i$. For a realization of $A_i$, $\bar{A}_i$, we let $R(\bar{A}_i)$ be a function representing the reward earned at customer $i$ when arriving to $i$ at time $\bar{A}_i$. We assume that $R(\bar{A}_i) = r_i$ for $\bar{A}_i \le D$ and $e_i$ otherwise.

Let $\tau$ be an order or tour of the customers in the selected set $M \subseteq N$ which begins at the depot 0. Then, the expected profit of the tour is $v(\tau) = \sum_{i\in\tau}[P(A_i \le D)r_i - (1 - P(A_i \le D))e_i]$. We seek a tour $\tau^\star$ such that $v(\tau^\star) \ge v(\tau)$ for every $\tau$.

In this paper, we study the OPSTS in detail. We will begin by reviewing the related literature in Sect. 2. In Sect. 3, we identify versions of the problem that can be solved exactly, and we show how existing variable neighborhood search heuristics can be used to solve general versions of the problem in Sect. 4. Section 5 discusses the datasets we generated for the OPSTS. In Sect. 6, we examine how different problem parameters affect the solutions and develop a good understanding of how deterministic and stochastic solution approaches differ in terms of the solutions they yield. Finally, Sect. 7 suggests future work.

## 2 Literature review

The orienteering problem (OP) has a long history in the literature. However, the research focuses on deterministic variants of the problem. Feillet et al. (2005) present a classification of orienteering literature. A broad overview of the orienteering problem, it variants, and associated solution methods can be found in Vansteenwegen et al. (2011).

Literature on stochastic variants of the OP is limited. The most closely related problem to the OPSTS is the time-constrained traveling salesman problem with stochastic travel and service times (TCTSP). Teng et al. (2004) introduce and solve the TCTSP. Unlike the problem discussed in this paper, their model is limited to discrete travel and service time distributions. The problem only uses a single penalty parameter as a mechanism for maintaining the feasibility of the solution. Teng et al. use the L-shaped algorithm to solve problems of up to 35 customers. Their experiments focus on demonstrating algorithmic performance only and do not examine solution structure.

Related to the TCTSP is the stochastic selective travelling salesperson problem (SSTSP) introduced by Tang and Miller-Hooks (2005). With the SSTSP, the travel times and service times are stochastic. The key difference between the OPSTS and the SSTSP is that the SSTSP incorporates deadlines via a chance constraint rather than modeling the economic cost of their violation as we do in this paper. The authors propose both exact and heuristic methods for solving SSTSP. Their experiments focus on demonstrating algorithmic performance and consider only tight delivery deadlines.

To the best of our knowledge, this paper, Tang and Miller-Hooks (2005), and Teng et al. (2004) are the only papers that address the selective traveling salesman or orienteering problem with stochasticity in service or travel times. Ilhan et al. (2008) examine a stochastic version of the orienteering problem, but the stochastic component is the profits associated with individual customers. The objective is to maximize the probability of collecting a target profit level within the time constraint or deadline. The authors propose an exact method for solving small instances and genetic algorithms for solving larger problem instances.

While only a few papers address stochastic variants of the OP, many papers address the vehicle routing problem (VRP) with random travel times and time constraints. Because of the key differences in the objectives of the problems, solution techniques for the VRP with stochastic travel times cannot be used to solve the OPSTS. Recent work includes Kenyon and Morton (2003) who consider a capacitated vehicle routing problem with stochastic travel times in which the objective is to maximize the probability that all vehicle tours are completed by a given time. The authors propose a Monte Carlo procedure that allows for the creation of a population of solutions from which a best solution can be chosen. The authors test their algorithm on a single 28 customer problem with two vehicles. Wong et al. (2003) introduce a 2-stage stochastic integer program with recourse for a problem where customers have time windows and travel times are discrete random variables. In the problem, the vehicle incurs a penalty if a customer is visited outside of its time window. Ando and Taniguchi (2006) present a case study which illustrates the value of accounting for travel-time variability in the construction of capacitated vehicle routes when customers visits are constrained by time windows. Jula et al. (2006) discuss a vehicle routing problem with stochastic travel times and time windows. To overcome difficulties in computing arrival time distributions, the authors demonstrate how to compute the first and second moments of the arrival time distribution. They propose a dynamic-programming-based solution approach that is capable of solving problems of up to 80 customers with tight time windows. Russell and Urban (2007) consider a problem with stochastic travel times in which violation of the time windows is penalized. The authors focus on the shifted Gamma distribution and use the Taguchi loss function to compute the penalties for violating the time windows. With considerable computational effort, problems of 100 customers are solved using a tabu search algorithm.

In addition to stochastic travel times, stochastic presence also affects the likelihood that time constraints can be satisfied in vehicle routing problems. Campbell and Thomas (2008a) provide a review of the related literature.

## 3 Exact solution methods for three problem variants

In this section, we present three special cases of the general model that can be solved exactly. For the first case, we can characterize the structure of the optimal tour. In the second case, we use a partial characterization of the optimal policy to develop a simple dynamic programming approach. The third case also involves a dynamic programming approach, but it applies to a broader class of instances than the second case.

### 3.1 Straight line distance and negligible service times

In this case, we assume that we can order the customers such that, for $i < j < k$, $P(X_{ik} \leq x) = P(X_{ij} + X_{jk} \leq x)$ for every $x$ and assume $P(X_{ik} \leq x) \geq P(X_{ij} + X_{jk} \leq x)$ for every $x$ when $i < j < k$ does not hold. That is, when $i < j < k$ does not hold, $X_{ij} + X_{jk}$ stochastically dominates $X_{ik}$. We assume that the service times are 0 for all customers. For this special case, we provide results analogous to those in Gendreau et al. (1995) and Campbell and Thomas (2008b), but modify them to account for rewards and penalties. This result holds for general penalty and reward structures. With these assumptions, for any subset of customers $C$, the optimal order of the customers in $C$ is the identity order, $c_1, \ldots, c_{|C|}$. The result follows from considering any other order of the customers in $C$. Let this alternative order be $\tau$. Let $i$ be the first customer in $C$ not in topological order in $\tau$. Create $\tau'$ from $\tau$ by removing $i$ from $\tau$ and inserting $i$ back into the tour in topological order. Let $m$ be the last customer in $\tau$ before $i$, and $n$ be the first customer in $\tau$ after $i$. Let $h$ be the last customer in $\tau'$ before $i$ and $j$ be the first customer in $\tau'$ after $i$. Thus, in creating $\tau'$ from $\tau$, we no longer travel from $h$ to $j$, from $m$ to $i$, or from $i$ to $n$ and instead travel from $h$ to $i$, from $i$ to $j$, and from $m$ to $n$. Because of our choice of $i$, we know $h < i < j$ and hence, by assumption, $P(X_{hj} \leq x) = P(X_{hi} + X_{ij} \leq x)$. Thus, by inserting $i$ between $h$ and $j$, the arrival-time distribution at $j$ and consequently all succeeding customers through $m$ is unchanged in $\tau'$ from $\tau$. Further, the likelihood of reaching $i$ in $\tau'$ before the deadline $D$ is greater than in $\tau$, increasing the expected value of the reward earned at $i$. From our assumptions, we also know that, by removing $i$ from between $m$ and $n$, $P(X_{mn} \leq x) \geq P(X_{mi} + X_{in} \leq x)$ for every $x$. Thus, the expected value of the reward at $n$ increases in $\tau'$ from $\tau$.

Thus, we are left to determine an optimal subset of customers to visit. Because we know that we will visit the customers in topological order, we can characterize the arrival distribution for each customer $1, \ldots, n$ a priori, regardless of which customers are included on the tour. The optimal subset is then those customers who have a positive expected profit.

### 3.2 Independent homogeneous travel and service time distributions

For this solvable case, we assume that $P(X_{ij} \leq x) = P(X_{kl} \leq x)$ for $i, j, k, l$ and for every $x$. We also assume that $P(S_i \leq s) = P(S_j \leq s)$ for all $i$ and $j$ and for all $s$. Thus, the arrival time distribution at the $q$th customer is the same regardless of who the $(q-1)$st customers are and of their order. This result holds for general penalty and reward structures. For simplicity, we write $P(A_q \leq D)$ as $p_q$. Then, customer $i$ in the $q$th position has a positive profit if $p_q r_i - (1 - p_q)e_i = p_q(r_i + e_i) - e_i > 0$.

Consider an optimal tour $\tau^\star$ and an alternate tour $\tau$ in which the $q$th and $(q+1)$st positions in $\tau^\star$ have been interchanged. In $\tau^\star$, let the customer in the $q$th position be $i$ and the customer in the $(q+1)$st be $j$. Then,

$$v(\tau^\star) - v(\tau) = p_q(r_i + e_i) - e_i + p_{q+1}(r_j + e_j) - e_j - (p_q(r_j + e_j) - e_j + p_{q+1}(r_i + e_i) - e_i).$$

It follows from the optimality of $\tau^\star$ that

$$p_q(r_i + e_i) - e_i + p_{q+1}(r_j + e_j) - e_j \geq p_q(r_j + e_j) - e_j + p_{q+1}(r_i + e_i) - e_i,$$

implying

$$r_i + e_i \geq r_j + e_j.$$

Unfortunately, the optimal solution is not found by simply ordering the customers on the tour in nonincreasing order of $r_i + e_i$ and then dropping any customers whose expected contribution is negative. Consider the following two customer example. Let $r_1 = 1$ and $e_1 = 10$ while $r_2 = 5$ and $e_2 = 3$. Assume that the probability of reaching the first customer in the tour before the deadline is 1 and the probability of reaching the second is 0.5. The expected profit from ordering the customers in nonincreasing order of $r_i + e_i$ is $1 + 0.5(5) - 0.5(3) = 2$. Yet, including only customer 2 in the tour yields an expected profit of 5.

Our result, though, does suggest a simple dynamic programming solution approach. Since we know that $r_i + e_i \geq r_j + e_j$ if $i$ is visited before $j$ on the tour, we begin by ordering the customers in nonincreasing order of $r_i + e_i$. Define $(i, k)$ as the state where $i$ is the current customer being considered and $k$ is the number of customers already included on the tour including $i$. The correctness of the state follows from the fact that our assumptions result in arrival time distributions depending on only the number of customers in the tour up to customer $i$ and not on their order. Then, letting $f(i, k)$ be the expected profit up to customer $i$ when $k$ customers are on the tour, we initialize $f(i, k) = 0$ for every $i$ and $k$. Next, letting $f(0, 0) = 0$, we can use the recursion $f(i, k) = \max\{f(i-1, k), f(i-1, k-1) + p_k r_i - (1 - p_k)e_i\}$. The optimal value is then obtained by $\max_k\{f(|N|, k)\}$. Overall, the complexity is $O(|N|^2)$, which follows from noting that there are $O(|N|^2)$ states. A dynamic programming approach analogous to that presented in the next subsection would result in a complexity of $O(|N|^3)$.

### 3.3 A restricted set of travel and service time distributions

Next, we discuss an exact solution approach for the case in which the travel and service time distributions differ only in a single characterizing parameter, the sum of which characterizes the convolution of the arrival time distributions. Our approach is motivated by the discussion in Kao (1978) that develops a dynamic programming solution approach for the traveling salesman problem with stochastic travel times. In particular, for Poisson, gamma (with common scale parameter $\lambda$), binomial and negative binomial (with common parameter $p$), and the normal (when the variance is a constant multiple of the mean) distributions, we can completely characterize the arrival-time distribution with a single distinguishing parameter. For each arc $(i, j)$, we denote this distinguishing parameter as $m_{ij}$. For each customer, we denote the service time parameter as $m_i$. For the named distributions, the convolution of the travel time and service time distributions is then characterized by the sum of the distinguishing parameters.

Assuming travel and service time distributions are independent but follow identical distributions differing in their parameters as noted above, we formulate the dynamic program for this model as follows. Let $i \in N$ represent the last node visited. We let the set $K$ represent the nodes visited prior to and including $i$. We also let $m$ represent the value of the distinguishing parameter for the arrival time distribution at the last node visited. The triple $(i, K, m)$ then characterizes the state of the dynamic program.

For a given state $(i, K, m)$, the available actions are to travel to a node in the set $N \setminus K$ or to end the tour. The action to travel from $i$ to $j$ results in a transition to state $(j, K \cup j, m + m_{ij} + m_i)$. We note that the transition of the distinguishing parameter follows from the restriction to the particular set of distributions mentioned previously. Further, because $m$ characterizes the arrival time distribution, the computation of the expected profit earned for traveling to a customer $j$ is straightforward. Choosing to visit $j$ from state $(i, K, m)$ results in the expected profit $\mathcal{R}(i, K, m, j) = F_{m'}(D)r_j + (1 - F_{m'}(D))e_j$, where $F_{m'}$ is the cumulative distribution function with parameter $m' = m + m_{ij} + m_i$. Choosing to end the tour yields no profit.

Another feature of our set of available actions for each state $(i, K, m)$ is that it clearly demonstrates that there exists an acyclic order of the states. Given this acyclic order and our objective, we have the following functional equation:

$$f(j, K', m') = \max_{(i,K,m):K=K'\setminus j, m'=m+m_{ij}+m_i} \{f(i, K, m) + \mathcal{R}(i, K, m, j)\}. \tag{1}$$

As noted in Fox and Denardo (1979), (1) can be solved recursively. To take advantage of pruning opportunities (described subsequently), we look to solve this described problem using the well known reaching algorithm for dynamic programming (see Denardo 2003). To initialize the algorithm, we set the functional value for each state $f(i, K, m) = -\infty$. For each successor of a state $(i, K, m)$, we “reach” out from $(i, K, m)$ to each of its successors $(j, K \cup j, m + m_{ij} + m_i)$ such that $j \notin K$. For each successor $(j, K \cup j, m + m_{ij} + m_i)$, we update $f(j, K \cup j, m + m_{ij} + m_i) = \max\{f(j, K \cup j, m + m_{ij} + m_i), f(i, K, m) + \mathcal{R}(i, K, m, j)\}$. We begin the algorithm by reaching from the state $(0, \emptyset, 0)$.

We note that, for two states $(i, K, m)$ and $(i, K, m')$ such that $m \leq m'$ and for any set and sequence of nodes chosen to complete the tour, say $\tau$, we have $P(A_j \leq D) \geq P(A'_j \leq D)$ for every $j \in \tau$ and $A_j$ and $A'_j$ (the random variables representing the arrival time to node $j$ starting from states $(i, K, m)$ and $(i, K, m')$, respectively). Thus, $E[R(A_j)] \geq E[R(A'_j)]$ for all $j \in \tau$ (see Puterman 1994, p. 106). Thus, if, for states $(i, K, m)$ and $(i, K, m')$ such that $m \leq m'$, $f(i, K, m) \geq f(i, K, m')$, then $(i, K, m)$ dominates $(i, K, m')$ and we can prune $(i, K, m')$ from the states which need to be considered for expansion in our dynamic program. To make the comparison of states more efficient, as in Feillet et al. (2004), we augment our state space with $l$ which is the cardinality of the set of visited nodes $K$. Further, we note that, when reaching from a state $(i, K, m)$ to a state $(j, K \cup j, m + m_{ij} + m_i)$, if $R(i, k, m, j) \leq 0$, then we never travel to node $j$ from state $(i, K, m)$.

## 4 Variable neighborhood search heuristic

While the proposed dominance relations in Sect. 3.3 improve the computational efficiency of the proposed dynamic program, the state space still grows exponentially in the size of the node set $N$, limiting the size of the problem that can be solved. To be able to explore problem characteristics of the OPSTS on realistically sized problems for distributions with a single characterizing parameter as well as to be able to consider other distributions, we need a heuristic capable of generating high quality solutions in a reasonable time. Inspired by the success in Sevkli and Sevilgen (2006) who use a variable neighborhood search (VNS) metaheuristic for the deterministic orienteering problem, we develop a variant of this well known metaheuristic that works well for the OPSTS.

### 4.1 Our implementation of VNS

The general concept of the VNS was first introduced by Mladenović and Hansen (1997), and an extensive review of the literature is available in Hansen et al. (2010). VNS operates by changing search neighborhoods to escape local minima. VNS has two components: a shaking phase and a local-search phase. In the shaking phase, a given solution is perturbed. The local-search phase improves the perturbed solution returned by the shaking phase. A general outline of the procedure can be found in Algorithm 1.

In our implementation, through experimentation, we determined a $t_{max}$ value of 100 was robust enough to return good solutions across the various datasets and parameter settings. The shaking phase simply chooses a random solution from the neighborhood $k \in N_s$, where $N_s$ is an ordered set of neighborhoods used in the shake phase. We use five neighborhoods in the set $N_s$. The first four neighborhoods are the neighborhoods that Feillet et al. (2005) identifies as being most commonly used in local-search heuristics for orienteering-type problems. The operations are: resequencing the route, replacing a customer on the route with one not on the route, adding a customer to the route, and deleting a customer from the route. To resequence the route, we use the well known 1-shift neighborhood. The other three neighborhoods are implemented exactly as their names imply. To create even larger changes in the current solution, we include a fifth neighborhood inspired by the ruin and recreate heuristic introduced in Schrimpf et al. (2000). Our particular implementation is adapted from Goodson (2008). Each time the ruin and recreate neighborhood is called, we remove from the tour $\lfloor \frac{nk}{10} \rfloor$ customers, where $n$ is the number of customers served by the current tour and $k$ is the current iteration of the VNS algorithm. We then add a random selection of customers who were not previously on the tour.

---

**Algorithm 1** Variable Neighborhood Search

---

**Input:**
 Data for an OPSTS Instance including a function $v$ for determining the cost of an OPSTS solution
 Ordered set of OPSTS neighborhoods $N_s$
**Output:** OPSTS solution, $\tau^\star$
**Initialization:**
 Determine initial feasible OPSTS solution $\tau$
 $t = 1, k = 1$
**repeat**
  $\tau' \leftarrow Shake(\tau, k)$.
  $\tau'' \leftarrow LocalSearch(\tau')$
  **if** $v(\tau) > v(\tau'')$ **then**
   **if** $k < |N_s|$ **then**
    $k \leftarrow k + 1$
   **else**
    $k \leftarrow 1$
   **end if**
  **else**
   $\tau \leftarrow \tau''$
  **end if**
  $t \leftarrow t + 1$
**until** $t > t_{max}$

---

**Algorithm 2** Variable Neighborhood Descent

**Input:**
Data for an OPSTS Instance including a function $v$ for determining the cost of an OPSTS solution
Ordered set of OPSTS neighborhoods $N_d$
**Output:** OPSTS solution, $\tau^\star$
**Initialization:**
Feasible OPSTS solution $\tau$
**repeat**
  $\tau' \leftarrow \tau$
  $k \leftarrow 1$
  **repeat**
    $\tau'' \leftarrow BestImproving(\tau, k)$
    **if** $v(\tau) > v(\tau'')$ **then**
      $k \leftarrow k+1$
    **else**
      $\tau \leftarrow \tau''$
    **end if**
  **until** $k > |N_d|$
**until** $\tau' = \tau$

For our local-search phase, we implement a form of the method known as variable neighborhood descent (VND). For our VND, we search a particular neighborhood of a current solution until no improving solution can be found and then the process is repeated with a different neighborhood. Typically, and in our implementation, the search is done using a steepest descent criteria. A description of our VND is given in Algorithm 2. The function *BestImproving* returns the best solution in the neighborhood $k$ of a given solution $\tau$. Of note, we increment the search neighborhood only when the search of the neighborhood does not return an improving solution. For our neighborhood set, we use the four neighborhoods suggested by Feillet et al. (2005).

### 4.2 Comparing an exact solution approach with VNS

To demonstrate that the VNS can return high quality solutions for the OPSTS, we compare its performance to that of the dynamic program introduced in Sect. 3.3. Details of the datasets and implementation can be found in the next section.

The results of our tests are found in Table 1. Because of the time-consuming nature of the dynamic programming approach for large problem sizes, only a subset of problems were tested using both approaches. In our 18 tests, both approaches yield solutions with the exact same objective values, indicating our VNS heuristic is capable of finding good, if not optimal, solutions for these types of problems. Because both approaches yield identical objective values, we do not report these in Table 1, but instead provide details about the runtimes of each approach and details about the solution process of the dynamic program. We see in Table 1 that the dynamic program is quite fast with low deadlines, but grows exponentially in runtime with increasing deadlines. This increased runtime is a reflection of the number of nodes being considered, even though our approach clearly prunes a significant number of them. Based on these results, the value of a VNS approach becomes quite clear for deadlines as low as 20.

**Table 1** Comparison of dynamic programming and heuristic (VNS) approaches

| Dataset | Deadline | Scale | Penalty-reward ratio | Nodes | Nodes pruned | DP runtime | VNS runtime |
|---|---|---|---|---|---|---|---|
| 221 | 15 | 1 | 0.1 | 1699710 | 1459782 | 62 | 13 |
| 221 | 20 | 1 | 0.1 | 9053007 | 7669717 | 4192 | 16 |
| 221 | 23 | 1 | 0.1 | 21473490 | 17701719 | 46890 | 17 |
| 221 | 25 | 1 | 0.1 | 33962690 | 27082892 | 129566 | 19 |
| 333 | 15 | 1 | 0.1 | 7402721 | 7010963 | 545 | 48 |
| 333 | 20 | 1 | 0.1 | 79018109 | 74064133 | 76012 | 57 |
| 432 | 10 | 1 | 0.1 | 177737 | 170127 | 0 | 28 |
| 432 | 15 | 1 | 0.1 | 3834790 | 3643130 | 124 | 38 |
| 432 | 20 | 1 | 0.1 | 54497662 | 51280727 | 37070 | 66 |
| 221 | 15 | 1 | 0.5 | 556584 | 487106 | 0 | 13 |
| 221 | 20 | 1 | 0.5 | 2455483 | 2098135 | 299 | 14 |
| 221 | 23 | 1 | 0.5 | 6054943 | 5132692 | 2097 | 15 |
| 221 | 25 | 1 | 0.5 | 10632274 | 8917989 | 6599 | 17 |
| 333 | 15 | 1 | 0.5 | 1264812 | 1201753 | 4 | 42 |
| 333 | 20 | 1 | 0.5 | 13333799 | 12588779 | 1359 | 57 |
| 432 | 10 | 1 | 0.5 | 29079 | 27917 | 0 | 28 |
| 432 | 15 | 1 | 0.5 | 556088 | 530573 | 1 | 36 |
| 432 | 20 | 1 | 0.5 | 7460384 | 7062242 | 416 | 54 |

## 5 Dataset generation and implementation details

As the problem explored in this paper is new, no datasets exist. We generate datasets based on five datasets from the deterministic orienteering literature. Two of the sets first appear in Tsiligirides (1984). These sets contain 21 and 33 customers, respectively, and 11 and 20 different deadlines, respectively. Using the convention of Sevkli and Sevilgen (2006), we label these sets 221 and 333.

The other three sets are introduced in Chao et al. (1996). The sets contain 32, 66, and 64 customers, respectively, and 18, 26, and 14 different deadlines, respectively. Again using the convention of Sevkli and Sevilgen (2006), we label these sets 432, 566, and 664.

For all of the datasets, we assume that the customers are fully connected and that the travel times on the arcs are gamma distributed. As noted in Russell and Urban (2007), distributions of this form are good approximations of travel time distributions. For these empirical tests, we ignore the service times as they can be accounted for by setting the travel time distributions for outgoing arcs to the convolution of the service time and arc distributions. For each arc, we set the mean travel time to the Euclidean distance between the two customers constituting the end nodes of the arc. For each instance, we fix the scale parameter of the gamma distribution to the same value for every arc. The effect of fixing the scale parameter to the same value for every arc is that the arrival time distributions can be characterized by the sum of the shape parameters for the arcs traversed. With the mean and scale determined, it follows that the shape parameters are found by dividing the mean by the scale parameter. To see the impact of increasing travel time variance, we consider scale parameter values of 1, 2, 3, and 4. Larger scale parameters correspond with larger travel time variances.

While the original datasets contain rewards, they do not have penalty data. Traditionally, orienteering problems are solved assuming deterministic travel and service times. In the solution of such problems, the size of the reward associated with each customer is important because only a limited number of customers can be served before the deadline. There is no need to consider the economic cost of violating a time constraint as any such violation renders the deterministic problem infeasible. Thus, we explore how a stochastic approach that incorporates both penalties and rewards improves solution quality and changes the structure of the solutions. To generate penalty values, we set the penalty to a fraction of the reward. We consider fractions of 0.1, 0.2, 0.5, and 1.0.

Because of the stochastic nature of the VNS algorithm, we run the VNS 10 times on each instance. For the presented results, we seed the VNS with deterministic solutions (produced by modifying the code used in Sevkli and Sevilgen (2006) to account for the fact that a return to the depot is not required in the OPSTS). The deterministic solutions use the Euclidean distance between any two customers as the deterministic travel time. In general, random seeds produce comparable solution values, but require greater computation time. For brevity, we omit the results of the random seeded runs.

Tables 2, 3, 4, 5, 6 in the Appendix present important experimental data that is not reflected in Sect. 6. Each table presents the runtimes required by the VNS to create the solutions discussed below. The runtime value reflects the total for the 10 runs. Each table also provides the standard deviation of the objective values from the 10 solutions. Tables 2–6 show that, across all datasets, runtimes initially increase with deadlines but stabilize once deadlines reach 35 for all scale parameters and penalty-reward ratios. The increase in runtimes occurs because more solutions offer a positive expected profit as deadlines increase. For a similar reason, with low deadlines and increasing scale parameters, lower penalty-reward ratios, or both, runtimes increase. Across all instances, the tables indicate that the standard deviations in solutions values are usually low. This low value is a reflection that the heuristic tends to converge to the same value for each run of an instance. Such behavior suggests that computation time might be reduced by reducing the number of runs and without affecting solution quality.

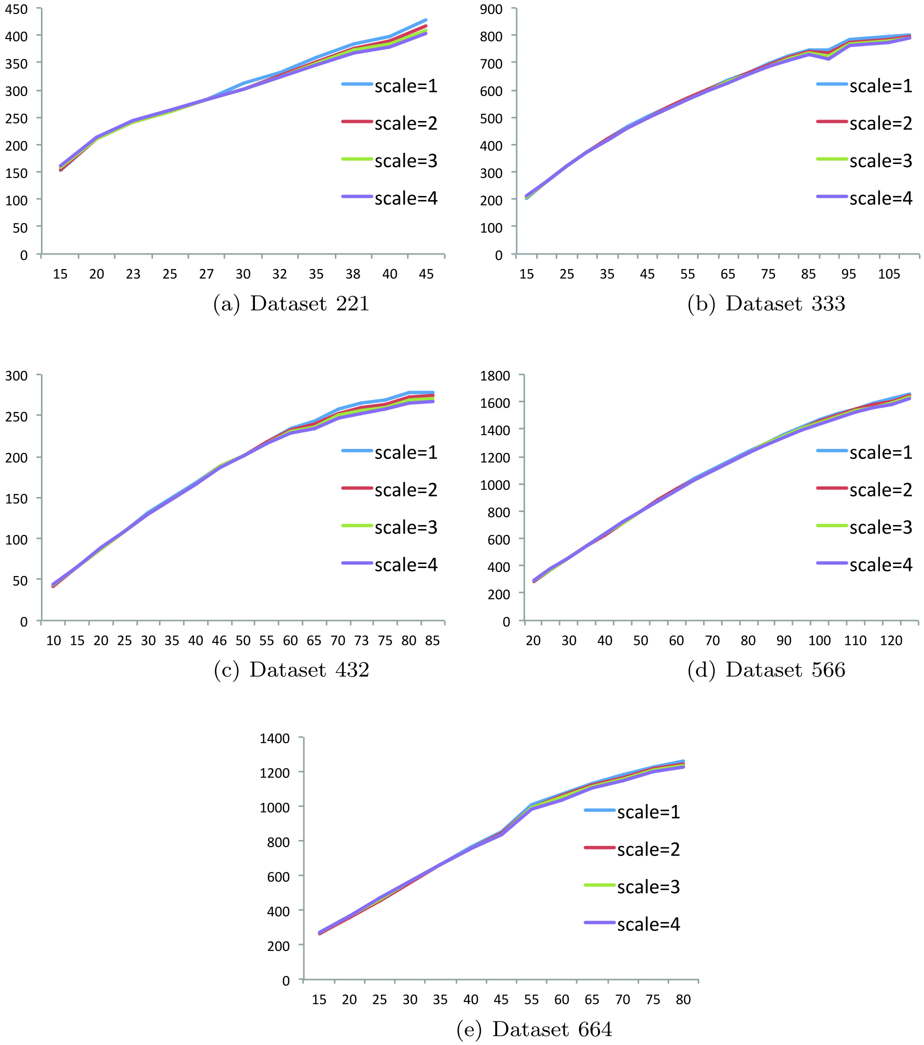
We implement our solution methods in C++ and run the instances on 2.40 GHz Intel Core 2 Quad processors using SuSE Linux 10.3. While memory consumption was never a bottleneck, four processors shared 3077 MB of RAM.

## 6 Computational experiments

This section presents the results of computational experiments. We designed our computational experiments with two purposes in mind. First, we want to determine the effect that problem parameters have on the objective value of the problem. The second goal of our experiments is to understand the changes in the routes that result from modeling the problem stochastically.

### 6.1 Examining the impact of scale parameters on objective values

Figure 1 examines how changing the scale parameter impacts the objective value in each of the different datasets using our VNS solution approach. In Fig. 1, we see that all scale parameters yield very similar expected profits for short deadlines across all datasets. As deadlines increase, though, scale values tend to make more difference, with lower scale values yielding larger rewards and higher scale values yielding lower rewards. The outcome is expected as higher scale values correspond with a higher variance in arc travel times, which we would expect to create lower objective values. We note that the unexpected dip




(a) Dataset 221

(b) Dataset 333

(c) Dataset 432

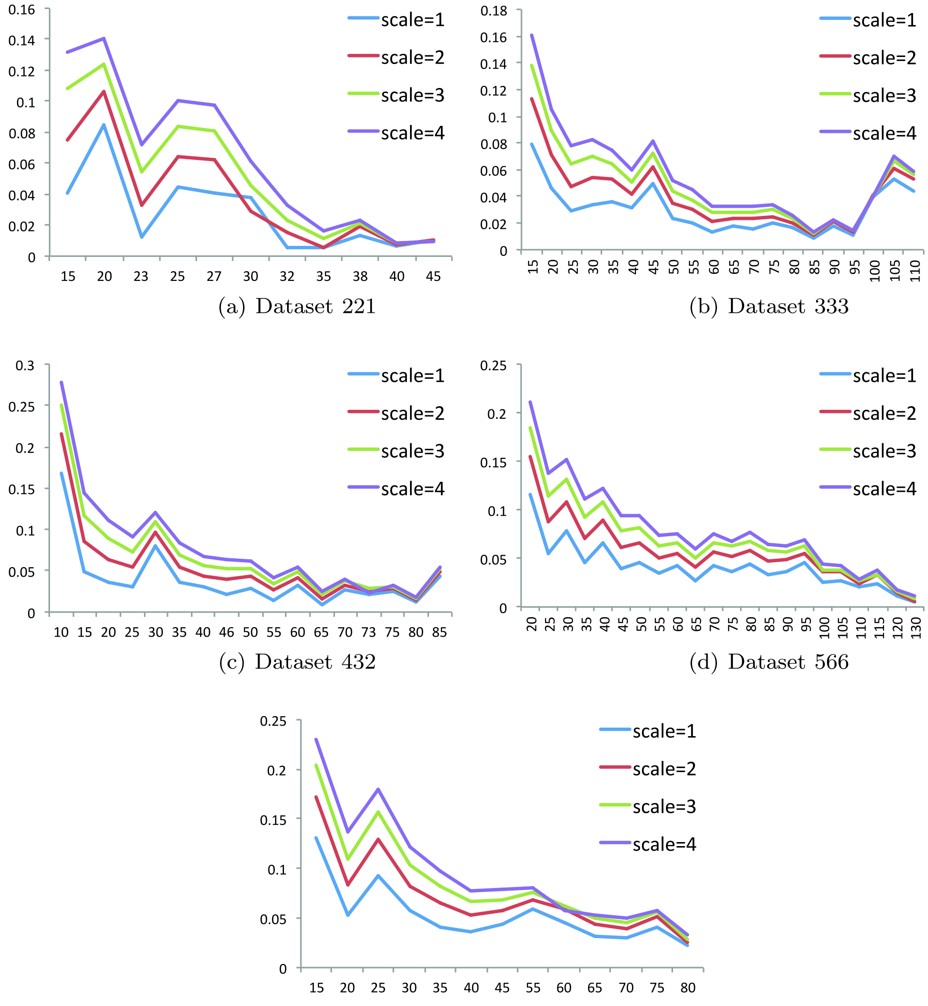
(d) Dataset 566

(e) Dataset 664

**Fig. 1** Deadline vs. objective value for different scale parameters and datasets

in Fig. 1(b) which reflects a lower objective at a deadline of 90 than at 85. This case is an example of one of the rare occurrences where starting from the solution to the deterministic problem and using our VNS approach clearly led us to a suboptimal solution. When we solved this problem with a random seed, we found a problem with a better objective value but a significantly longer runtime.

Figure 2 compares the expected profit of the seed solution (the solution generated using deterministic data evaluated with the stochastic objective function) with the expected profit of the best solution found by VNS. We express the difference relative to the stochastic solution value by a percentage and graph these percentages for the different scale parameters. These graphs reflect that including stochastic information in the model can dramatically improve the objective value. We observe that the higher the scale, the higher the percentage difference in objective values. As noted earlier, higher scale values are associated with




(a) Dataset 221

(b) Dataset 333

(c) Dataset 432

(d) Dataset 566

(e) Dataset 664

**Fig. 2** Percentage difference in expected reward from solving stochastic vs. deterministic problem versions for different scale parameters and datasets

higher variances in the travel time distributions. Thus, the increase in the percentage differences is a reflection of the value of including stochastic information in the model as problem variance increases. We also see that these percentages decrease for all scale parameters as the deadline increases, with a convergence across all scale parameters for the highest deadlines. The latter results because, as the deadlines increase, it is more likely that all customers will be on the tours, and thus the penalties will have less impact on the objectives.

## 6.2 Examining the impact of penalty-reward ratios on objective values

Next, we examine how changing the penalty-reward ratio impacts the objective value in each of the different datasets. Figure 3 shows the results of the experiments. As expected, the figure shows that the lowest penalty-reward ratio yields the highest expected profits, and

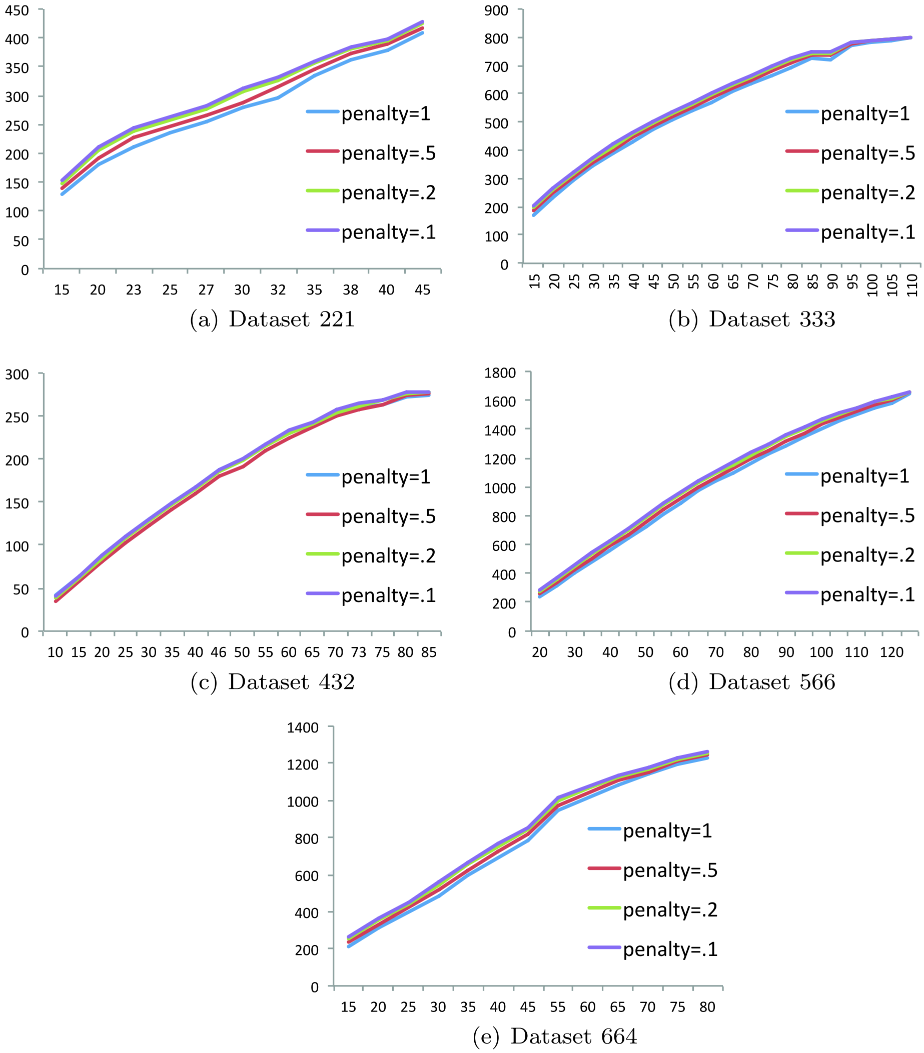
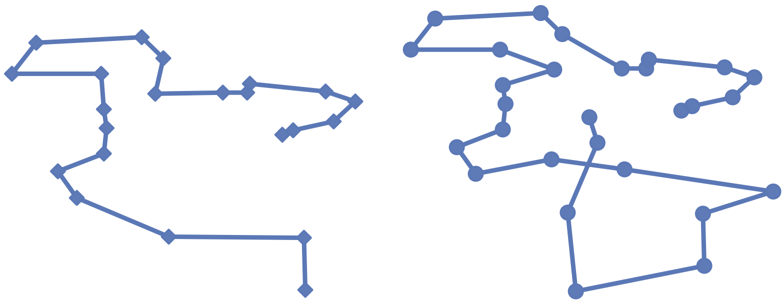



**Fig. 3** Deadline vs. objective value for different penalty-reward ratios and datasets

expected profit decreases as the penalty increases. We can also observe that the amount of impact that the penalty ratio has on the objective seems to be somewhat dataset specific. Across datasets, the objective values increase with deadlines, but, unlike with scale parameters, the values tend to converge rather than diverge with increasing deadlines. This trend reflects the fact that higher deadlines create smaller penalties since customers are more likely to be served. We again see the unexpected dip in Fig. 3(b) which is a reflection of the starting solution.

### 6.3 Examining the impact on solution structure

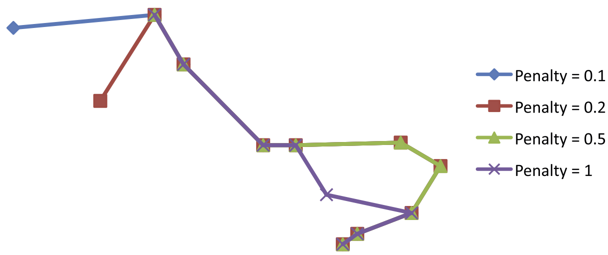
To gain further insight into the effect of modeling the problem with stochastic information, we examine the final tours created by deterministic and stochastic approaches to understand



(a) Deterministic Tour (b) Stochastic Tour

**Fig. 4** Comparing stochastic and deterministic routes

**Fig. 5** Routes for different penalty-reward ratios




how they differ in structure. Because the tours for the deterministic version of the problem must be feasible, these deterministic tours are generally smaller in terms of the number of customers. The stochastic tour includes customers that may not necessarily be reached, but the probability is high enough that the expected profit from the additional customers is positive. For example, for Dataset 333, scale of 4, and penalty-reward ratio of 1, we find with a deadline of 15 that the deterministic tour has 8 customers where the stochastic one has 12. With a deadline of 60, the difference is 22 versus 28 customers. With a deadline of 110, the routes have the same number of customers, all of them in fact, because the large deadline is not constraining. In terms of which customers appear on the shorter tours, across most examples, the deterministic tour contains customers that are a subset of the customers on the stochastic ones, and the customers that are excluded are ones from near the end of the stochastic tour. For example, see Figs. 4(a) and 4(b) which represent Dataset 333, scale of 4, ratio of 1, and deadline of 60. The routes begin the same, but the deterministic route skips 6 of the last 9 customers on the tour. We also see that on the last part of the stochastic tour, the tour crosses itself which would never happen on a deterministic tour. The customers near the end of the stochastic tours are typically ones with low rewards, and thus not a priority, so they are placed at the end of the tour where there is a low chance of them being reached. This type of ordering would not occur with the deterministic counterpart.

We also explore the impact of the penalty-reward ratios on the actual structure of the routes. As the penalty gets larger, it is not surprising that the number of customers on the stochastic tours tends to get smaller. This is because each additional customer on the longer routes has a very low probability of being reached by the deadline and thus a high probability of causing a penalty. Here, we found that the customers on the shorter routes (with higher penalties) are usually a subset of the larger routes (with lower penalties) except for a few exchanges in order to make the routes with the higher penalties more likely to be feasible. This type of result is best portrayed through an illustrative example. In Fig. 5, we will examine Dataset 333 with a deadline of 15, scale of 10, and the different penalty-reward ratios. The results for this dataset are very similar to the results found for the other datasets. Here, the ratios of 0.1 and 0.2 yield the longest tours, and the ratio of 0.2 yields a tour that is identical except for the last customer. The last customer on the tour with ratio of 0.2 can be reached with a slightly higher probability than the last customer on the tour with ratio of 0.1, creating a lower expected penalty. The reduction in penalty is sufficient to cause a customer to be selected with a lower potential reward to end the tour than the tour with ratio of 0.1. The tour of ratio of 0.2 has one more customer at the end than the tour with ratio of 0.5. With a higher penalty, it is not worth the added risk to include this last customer. The tour with ratio of 1 has one fewer customer than the tour with ratio of 0.5, but the customers that are different are early on the tour. The fourth and fifth customers on the other tours are replaced by a different single customer on the tour with ratio of 1. This one customer can be reached in much shorter time than the two customers, thus decreasing the expected penalty for this customer and for all subsequent customers.

## 7 Future work

A number of directions exist for future work. For one, this paper is limited to a single tour of the customers. A multiple tour version is relevant in many real-world situations where there exists a fleet of vehicles to serve customers and/or make deliveries. Also, many real-world implementations are likely to include individual time windows for the customers. It also may be interesting to incorporate stochastic rewards, because for service companies, the actual service that must be completed and its associated "reward" are often not known until the service is completed. Finally, in this paper, we have ignored the fact that customers not served today must be served in the near future. Including this consideration would make for an interesting extension and could make a big impact on which customers are selected on a given day.

From a computational standpoint, there are opportunities to improve the runtime of the VNS heuristic for instances of this problem with discrete travel and service time distributions. Applying approximation ideas such as those discussed in Campbell and Thomas (2009) offers a direction for future research. In addition, our solution approach is an adaptation of a well known heuristic. A more tailored approach may yield improved solutions. To promote such work, our datasets and corresponding solution values are available at http://myweb.uiowa.edu/bthoa/Research.htm.

**Acknowledgements** The authors are grateful to Z. Sevkli and F.E. Sevilgen for the use of their code. The authors would also like to thank the referees for the helpful suggestions to improve the paper.

## Appendix

**Table 2** Comparison of runtimes and standard deviation of solution values across different scale and penalty-reward ratios for dataset 221

| | Scale | | | | | | | | Penalty-reward ratio | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 0.50 | 0.2 | 0.1 | 1 | 0.50 | 0.2 | 0.1 |
| Deadline | Runtime | | | | Std dev | | | | Runtime | | | | Std dev | | | |
| 15 | 13.19 | 14.55 | 15.86 | 19.02 | 0.13 | 0.00 | 0.00 | 0.00 | 13.36 | 13.17 | 13.74 | 13.19 | 0.00 | 0.31 | 0.20 | 0.13 |
| 20 | 15.89 | 16.17 | 18.74 | 23.21 | 0.00 | 0.00 | 0.00 | 0.00 | 14.06 | 13.65 | 13.77 | 15.89 | 0.00 | 0.00 | 0.00 | 0.00 |
| 23 | 17.46 | 18.82 | 21.04 | 26.43 | 0.00 | 0.00 | 0.00 | 0.00 | 14.53 | 15.40 | 16.00 | 17.46 | 0.00 | 0.00 | 0.00 | 0.00 |
| 25 | 19.27 | 18.94 | 22.43 | 28.50 | 0.33 | 0.02 | 0.01 | 0.00 | 14.87 | 16.76 | 17.89 | 19.27 | 0.00 | 0.00 | 0.00 | 0.33 |
| 27 | 20.01 | 22.10 | 27.63 | 25.74 | 0.00 | 0.00 | 0.00 | 0.00 | 16.17 | 18.25 | 19.28 | 20.01 | 0.00 | 0.00 | 0.00 | 0.00 |
| 30 | 23.64 | 28.01 | 21.02 | 23.54 | 2.64 | 0.00 | 0.00 | 0.00 | 18.44 | 20.92 | 22.62 | 23.64 | 1.92 | 1.30 | 3.66 | 2.64 |
| 32 | 19.59 | 20.10 | 26.53 | 17.99 | 0.00 | 0.53 | 0.47 | 0.09 | 17.40 | 18.19 | 18.60 | 19.59 | 0.00 | 0.00 | 0.00 | 0.00 |
| 35 | 24.48 | 28.23 | 19.73 | 19.99 | 0.58 | 0.00 | 0.00 | 0.00 | 21.27 | 22.38 | 23.23 | 24.48 | 0.00 | 0.00 | 1.38 | 0.58 |
| 38 | 29.73 | 21.08 | 22.07 | 22.47 | 1.48 | 1.18 | 0.68 | 0.70 | 23.04 | 24.50 | 25.06 | 29.73 | 0.00 | 0.00 | 0.92 | 1.48 |
| 40 | 20.97 | 19.79 | 18.94 | 19.43 | 0.00 | 0.00 | 0.00 | 0.00 | 24.77 | 25.36 | 25.91 | 20.97 | 0.00 | 0.00 | 1.43 | 0.00 |
| 45 | 23.49 | 22.59 | 20.12 | 21.58 | 0.00 | 0.00 | 0.00 | 0.00 | 23.62 | 23.59 | 23.67 | 23.49 | 0.00 | 0.00 | 0.00 | 0.00 |

**Table 3** Comparison of runtimes and standard deviation of solution values across different scale and penalty-reward ratios for dataset 333

| | Scale | | | | | | | | Penalty-reward ratio | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 0.50 | 0.2 | 0.1 | 1 | 0.50 | 0.2 | 0.1 |
| Deadline | Runtime | | | | Std dev | | | | Runtime | | | | Std dev | | | |
| 15 | 48.19 | 55.88 | 63.20 | 66.26 | 3.39 | 0.00 | 0.00 | 0.00 | 39.79 | 42.17 | 41.05 | 48.19 | 0.00 | 0.00 | 3.00 | 3.39 |
| 20 | 56.51 | 66.71 | 83.08 | 87.43 | 0.00 | 0.00 | 0.00 | 0.00 | 61.27 | 57.45 | 54.92 | 56.51 | 0.00 | 0.00 | 0.00 | 0.00 |
| 25 | 81.94 | 81.08 | 98.70 | 109.14 | 0.00 | 0.00 | 0.00 | 0.00 | 63.22 | 71.49 | 75.53 | 81.94 | 0.00 | 0.00 | 0.00 | 0.00 |
| 30 | 107.58 | 95.49 | 108.65 | 113.65 | 0.00 | 0.00 | 0.00 | 0.00 | 84.67 | 91.56 | 95.81 | 107.58 | 0.00 | 0.00 | 0.00 | 0.00 |
| 35 | 118.72 | 106.41 | 121.32 | 119.91 | 0.00 | 0.00 | 0.00 | 0.00 | 107.10 | 115.59 | 112.56 | 118.72 | 0.52 | 0.00 | 0.00 | 0.00 |
| 40 | 123.31 | 118.21 | 110.35 | 110.52 | 0.00 | 0.00 | 0.00 | 0.00 | 115.67 | 125.05 | 133.34 | 123.31 | 0.00 | 0.00 | 0.00 | 0.00 |
| 45 | 128.70 | 122.99 | 112.26 | 105.71 | 0.00 | 0.00 | 0.00 | 0.00 | 137.06 | 137.41 | 135.96 | 128.70 | 0.00 | 0.00 | 0.00 | 0.00 |
| 50 | 133.92 | 121.96 | 99.87 | 103.53 | 0.00 | 0.00 | 0.00 | 0.00 | 146.51 | 127.11 | 129.48 | 133.92 | 0.00 | 0.00 | 0.00 | 0.00 |
| 55 | 133.11 | 112.89 | 103.42 | 106.07 | 0.00 | 0.00 | 0.00 | 0.00 | 129.36 | 138.48 | 129.62 | 133.11 | 0.00 | 0.00 | 0.00 | 0.00 |
| 60 | 131.53 | 120.63 | 118.18 | 112.85 | 0.00 | 0.00 | 0.00 | 0.00 | 128.67 | 132.60 | 123.98 | 131.53 | 0.00 | 0.00 | 0.00 | 0.00 |
| 65 | 132.04 | 122.65 | 121.75 | 118.53 | 0.00 | 0.00 | 0.20 | 0.00 | 122.25 | 123.16 | 127.60 | 132.04 | 0.00 | 0.00 | 0.00 | 0.00 |
| 70 | 142.57 | 136.93 | 131.29 | 120.82 | 0.00 | 0.00 | 0.00 | 0.00 | 124.14 | 128.49 | 136.30 | 142.57 | 0.00 | 0.00 | 0.00 | 0.00 |
| 75 | 147.49 | 140.05 | 132.07 | 96.38 | 0.00 | 0.00 | 0.00 | 0.00 | 123.07 | 137.19 | 136.02 | 147.49 | 0.00 | 0.00 | 0.00 | 0.00 |
| 80 | 156.31 | 151.05 | 108.07 | 98.99 | 0.00 | 0.00 | 0.00 | 0.00 | 139.04 | 138.57 | 143.47 | 156.31 | 0.00 | 0.00 | 0.00 | 0.00 |
| 85 | 151.58 | 114.63 | 109.38 | 100.58 | 0.00 | 0.30 | 0.31 | 0.00 | 141.22 | 136.89 | 169.19 | 151.58 | 0.50 | 0.43 | 0.23 | 0.00 |
| 90 | 160.87 | 101.51 | 91.28 | 88.83 | 0.00 | 0.00 | 0.00 | 0.00 | 129.44 | 128.48 | 139.23 | 160.87 | 0.00 | 0.00 | 0.00 | 0.00 |
| 95 | 111.76 | 109.10 | 107.64 | 100.30 | 0.00 | 0.00 | 0.00 | 0.00 | 144.00 | 109.08 | 113.67 | 111.76 | 0.00 | 0.00 | 0.00 | 0.00 |
| 100 | 104.06 | 94.71 | 88.92 | 92.30 | 0.00 | 0.00 | 0.00 | 0.00 | 96.89 | 99.05 | 101.39 | 104.06 | 0.00 | 0.00 | 0.00 | 0.00 |
| 105 | 114.75 | 114.86 | 111.99 | 109.56 | 1.21 | 0.67 | 1.28 | 2.62 | 109.32 | 111.34 | 116.83 | 114.75 | 1.84 | 1.51 | 1.32 | 1.21 |
| 110 | 134.87 | 115.22 | 104.05 | 99.62 | 0.53 | 0.45 | 0.50 | 0.22 | 138.99 | 134.69 | 132.05 | 134.87 | 0.90 | 0.58 | 0.47 | 0.53 |

**Table 4** Comparison of runtimes and standard deviation of solution values across different scale and penalty-reward ratios for dataset 432

| | Scale | | | | | | | | Penalty-reward ratio | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 0.50 | 0.2 | 0.1 | 1 | 0.50 | 0.2 | 0.1 |
| Deadline | Runtime | | | | Std dev | | | | Runtime | | | | Std dev | | | |
| 10 | 28.46 | 39.38 | 44.30 | 43.94 | 0.00 | 0.00 | 0.00 | 0.00 | 26.36 | 28.49 | 26.72 | 28.46 | 0.00 | 0.00 | 0.00 | 0.00 |
| 15 | 37.97 | 52.92 | 60.42 | 65.79 | 0.00 | 0.00 | 0.05 | 0.00 | 31.73 | 35.77 | 38.12 | 37.97 | 0.00 | 0.00 | 0.00 | 0.00 |
| 20 | 66.40 | 62.81 | 78.72 | 80.29 | 0.05 | 0.00 | 0.18 | 0.24 | 56.46 | 53.68 | 54.26 | 66.40 | 0.00 | 0.00 | 0.00 | 0.05 |
| 25 | 79.39 | 82.82 | 81.15 | 84.85 | 0.00 | 0.00 | 0.00 | 0.00 | 67.33 | 72.37 | 77.83 | 79.39 | 0.00 | 0.00 | 0.00 | 0.00 |
| 30 | 98.20 | 84.02 | 87.64 | 99.38 | 0.00 | 0.00 | 0.00 | 0.00 | 81.04 | 89.43 | 89.83 | 98.20 | 0.00 | 0.00 | 0.00 | 0.00 |
| 35 | 116.51 | 97.57 | 91.55 | 88.09 | 0.00 | 0.00 | 0.00 | 0.00 | 104.82 | 101.65 | 105.00 | 116.51 | 0.00 | 0.00 | 0.00 | 0.00 |
| 40 | 117.51 | 104.82 | 92.97 | 101.31 | 0.00 | 0.00 | 0.00 | 0.00 | 103.17 | 118.27 | 111.88 | 117.51 | 0.00 | 0.00 | 0.00 | 0.00 |
| 46 | 116.76 | 106.23 | 88.53 | 88.38 | 0.00 | 0.00 | 0.00 | 0.00 | 114.32 | 119.78 | 121.81 | 116.76 | 0.00 | 0.00 | 0.00 | 0.00 |
| 50 | 119.30 | 101.36 | 88.28 | 92.50 | 0.00 | 0.00 | 0.00 | 0.00 | 127.81 | 116.72 | 115.89 | 119.30 | 0.00 | 0.00 | 0.00 | 0.00 |
| 55 | 121.76 | 106.37 | 99.26 | 101.47 | 0.00 | 0.00 | 0.00 | 0.00 | 112.39 | 126.35 | 113.48 | 121.76 | 0.00 | 0.00 | 0.00 | 0.00 |
| 60 | 119.41 | 113.70 | 127.32 | 112.96 | 0.00 | 0.00 | 0.00 | 0.00 | 115.92 | 114.47 | 115.88 | 119.41 | 0.00 | 0.00 | 0.00 | 0.00 |
| 65 | 161.74 | 155.46 | 155.60 | 110.62 | 0.00 | 0.33 | 0.54 | 0.75 | 130.31 | 132.16 | 142.39 | 161.74 | 0.00 | 0.00 | 0.00 | 0.00 |
| 70 | 176.40 | 173.65 | 106.29 | 88.92 | 0.44 | 0.76 | 0.78 | 0.76 | 129.42 | 169.46 | 150.06 | 176.40 | 0.00 | 0.00 | 0.00 | 0.44 |
| 73 | 123.00 | 107.58 | 100.38 | 86.48 | 0.47 | 0.68 | 0.89 | 0.00 | 143.61 | 179.21 | 167.42 | 123.00 | 0.00 | 0.00 | 0.00 | 0.47 |
| 75 | 123.81 | 122.36 | 106.05 | 99.13 | 0.00 | 0.00 | 0.00 | 0.00 | 151.18 | 157.45 | 117.33 | 123.81 | 0.00 | 0.00 | 0.00 | 0.00 |
| 80 | 114.48 | 102.51 | 112.58 | 95.20 | 0.00 | 0.00 | 0.00 | 0.00 | 116.30 | 114.40 | 113.08 | 114.48 | 0.00 | 0.00 | 0.00 | 0.00 |
| 85 | 108.67 | 99.40 | 91.63 | 89.87 | 0.00 | 0.02 | 0.00 | 0.00 | 121.16 | 108.71 | 110.76 | 108.67 | 0.00 | 0.40 | 0.21 | 0.00 |

**Table 5** Comparison of runtimes and standard deviation of solution values across different scale and penalty-reward ratios for dataset 566

| | Scale | | | | | | | | Penalty-reward ratio | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 0.50 | 0.2 | 0.1 | 1 | 0.50 | 0.2 | 0.1 |
| Deadline | Runtime | | | | Std dev | | | | Runtime | | | | Std dev | | | |
| 20 | 586.95 | 647.23 | 836.08 | 976.47 | 0.71 | 1.31 | 1.01 | 0.98 | 558.29 | 544.61 | 579.91 | 586.95 | 0.79 | 0.89 | 0.00 | 0.71 |
| 25 | 722.61 | 828.91 | 1034.55 | 940.67 | 0.00 | 0.00 | 0.00 | 0.00 | 654.28 | 753.23 | 688.93 | 722.61 | 0.00 | 0.00 | 0.00 | 0.00 |
| 30 | 877.15 | 1046.67 | 1062.00 | 1167.89 | 5.89 | 1.21 | 5.66 | 4.19 | 846.46 | 908.84 | 952.03 | 877.15 | 0.63 | 5.05 | 1.06 | 5.89 |
| 35 | 978.08 | 1062.63 | 1183.67 | 1197.62 | 0.00 | 0.00 | 0.00 | 0.00 | 947.78 | 1110.32 | 1045.92 | 978.08 | 0.00 | 0.00 | 0.00 | 0.00 |
| 40 | 1308.59 | 1329.52 | 1307.86 | 1297.94 | 1.08 | 1.07 | 1.01 | 3.86 | 1134.11 | 1182.13 | 1338.60 | 1308.59 | 0.99 | 0.90 | 1.14 | 1.08 |
| 45 | 1447.26 | 1468.19 | 1431.40 | 1316.71 | 0.00 | 0.00 | 0.00 | 0.00 | 1250.32 | 1390.44 | 1441.35 | 1447.26 | 0.00 | 0.00 | 0.00 | 0.00 |
| 50 | 1637.29 | 1632.88 | 1427.14 | 1391.51 | 0.00 | 0.00 | 0.00 | 0.00 | 1512.76 | 1426.32 | 1574.79 | 1637.29 | 0.00 | 0.00 | 0.00 | 0.00 |
| 55 | 1944.38 | 1709.53 | 1427.91 | 1517.45 | 0.00 | 0.00 | 0.00 | 0.00 | 1678.41 | 1572.20 | 1751.70 | 1944.38 | 0.00 | 0.00 | 0.00 | 0.00 |
| 60 | 2049.40 | 1794.36 | 1731.67 | 1529.20 | 0.00 | 0.00 | 0.00 | 0.00 | 1779.48 | 1959.58 | 1894.24 | 2049.40 | 0.00 | 0.00 | 0.00 | 0.00 |
| 65 | 2141.53 | 1833.05 | 1943.08 | 1605.07 | 0.00 | 0.00 | 0.00 | 0.00 | 1850.13 | 1980.79 | 2012.99 | 2141.53 | 0.00 | 1.76 | 0.00 | 0.00 |
| 70 | 2243.63 | 1995.39 | 2077.42 | 1689.81 | 0.00 | 0.00 | 0.26 | 0.00 | 2034.54 | 1930.55 | 2150.52 | 2243.63 | 0.00 | 0.00 | 0.72 | 0.00 |
| 75 | 2352.97 | 2138.39 | 1856.54 | 1732.79 | 1.34 | 1.42 | 1.39 | 0.99 | 2272.77 | 2287.51 | 2427.48 | 2352.97 | 0.98 | 0.00 | 0.00 | 1.34 |
| 80 | 2401.52 | 2184.83 | 1851.48 | 1621.06 | 0.22 | 1.11 | 1.89 | 8.29 | 2298.87 | 2257.01 | 2310.55 | 2401.52 | 0.00 | 0.00 | 0.00 | 0.22 |
| 85 | 2467.78 | 2005.66 | 1699.47 | 1677.75 | 0.71 | 0.63 | 0.00 | 0.83 | 2279.23 | 2398.26 | 2368.71 | 2467.78 | 0.13 | 0.29 | 0.25 | 0.71 |
| 90 | 2118.19 | 1914.62 | 1672.56 | 1505.49 | 0.64 | 1.74 | 1.79 | 8.36 | 2168.86 | 2321.81 | 2297.83 | 2118.19 | 0.21 | 0.52 | 1.03 | 0.64 |
| 95 | 2124.78 | 1837.54 | 1584.94 | 1374.21 | 5.83 | 8.14 | 9.30 | 11.39 | 2175.57 | 2164.39 | 2111.86 | 2124.78 | 0.92 | 2.15 | 5.96 | 5.83 |
| 100 | 1842.78 | 1569.32 | 1398.37 | 1290.39 | 0.58 | 2.90 | 0.00 | 1.88 | 2057.22 | 2021.02 | 1891.97 | 1842.78 | 1.40 | 1.13 | 2.49 | 0.58 |
| 105 | 1770.80 | 1563.14 | 1266.52 | 1115.00 | 2.77 | 1.89 | 1.25 | 1.46 | 1882.28 | 1855.65 | 1706.43 | 1770.80 | 2.44 | 3.81 | 0.87 | 2.77 |
| 110 | 1679.57 | 1227.73 | 1198.13 | 1120.28 | 1.82 | 0.85 | 0.48 | 1.87 | 1775.36 | 1781.16 | 1777.19 | 1679.57 | 1.26 | 1.56 | 1.52 | 1.82 |
| 115 | 1190.02 | 1194.57 | 1139.74 | 1070.42 | 1.49 | 5.24 | 3.00 | 4.95 | 1574.08 | 1587.20 | 1514.95 | 1190.02 | 5.91 | 4.61 | 3.67 | 1.49 |
| 120 | 1359.29 | 1288.16 | 1330.15 | 1234.89 | 1.47 | 2.08 | 2.04 | 2.77 | 1698.79 | 1840.46 | 1326.48 | 1359.29 | 2.18 | 1.28 | 0.73 | 1.47 |
| 130 | 1271.90 | 1247.09 | 1151.98 | 1148.53 | 1.28 | 1.36 | 2.76 | 1.99 | 1249.92 | 1253.31 | 1275.29 | 1271.90 | 1.10 | 1.44 | 1.70 | 1.28 |

**Table 6** Comparison of runtimes and standard deviation of solution values across different scale and penalty-reward ratios for dataset 664

| | Scale | | | | | | | | Penalty-reward ratio | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 0.50 | 0.2 | 0.1 | 1 | 0.50 | 0.2 | 0.1 |
| Deadline | Runtime | | | | Std dev | | | | Runtime | | | | Std dev | | | |
| 15 | 617.53 | 759.40 | 940.50 | 1024.80 | 0.45 | 0.66 | 0.60 | 0.83 | 490.77 | 519.03 | 533.92 | 617.53 | 0.00 | 0.00 | 0.00 | 0.45 |
| 20 | 819.19 | 869.23 | 993.03 | 1105.43 | 0.46 | 0.84 | 0.00 | 0.25 | 692.70 | 747.96 | 784.00 | 819.19 | 0.50 | 0.46 | 0.44 | 0.46 |
| 25 | 1094.23 | 1175.40 | 1333.80 | 1191.32 | 0.97 | 0.44 | 0.56 | 0.82 | 844.72 | 926.44 | 1072.20 | 1094.23 | 0.42 | 1.17 | 1.36 | 0.97 |
| 30 | 1253.84 | 1224.47 | 1287.20 | 1306.71 | 0.00 | 0.48 | 1.88 | 1.32 | 1059.95 | 1076.72 | 1210.56 | 1253.84 | 0.00 | 0.00 | 0.00 | 0.00 |
| 35 | 1455.61 | 1415.35 | 1434.31 | 1393.85 | 0.00 | 1.20 | 2.12 | 2.01 | 1248.25 | 1398.52 | 1434.58 | 1455.61 | 0.00 | 0.00 | 0.00 | 0.00 |
| 40 | 1760.53 | 1539.88 | 1462.00 | 1434.74 | 0.00 | 0.69 | 2.53 | 3.08 | 1521.03 | 1563.98 | 1551.67 | 1760.53 | 0.34 | 0.76 | 0.81 | 0.00 |
| 45 | 1990.97 | 1819.51 | 1480.71 | 1557.04 | 1.10 | 1.33 | 2.02 | 2.81 | 1759.58 | 1807.03 | 1850.93 | 1990.97 | 1.18 | 1.52 | 2.12 | 1.10 |
| 55 | 1980.63 | 1552.12 | 1327.61 | 1356.76 | 1.49 | 1.30 | 1.85 | 0.97 | 1914.88 | 2045.67 | 2025.01 | 1980.63 | 1.28 | 2.15 | 1.79 | 1.49 |
| 60 | 2044.91 | 1651.23 | 1520.47 | 1161.47 | 2.90 | 4.09 | 2.98 | 1.10 | 1878.56 | 2208.02 | 2060.57 | 2044.91 | 5.79 | 4.67 | 4.64 | 2.90 |
| 65 | 1712.56 | 1553.24 | 1454.56 | 1258.52 | 0.92 | 2.05 | 2.43 | 2.59 | 1981.85 | 1937.16 | 1747.06 | 1712.56 | 1.59 | 2.02 | 2.18 | 0.92 |
| 70 | 1731.47 | 1567.93 | 1413.91 | 1349.26 | 3.91 | 1.07 | 2.16 | 1.75 | 2066.43 | 1716.05 | 1788.60 | 1731.47 | 6.47 | 2.23 | 3.30 | 3.91 |
| 75 | 1600.62 | 1517.89 | 1457.45 | 1058.45 | 2.06 | 1.60 | 2.04 | 1.09 | 1691.32 | 1769.10 | 1594.04 | 1600.62 | 3.43 | 3.65 | 2.86 | 2.06 |
| 80 | 1569.70 | 1634.05 | 1077.38 | 993.43 | 1.08 | 0.72 | 1.05 | 2.67 | 1554.32 | 1524.71 | 1581.45 | 1569.70 | 2.99 | 1.74 | 0.98 | 1.08 |

## References


Ando, N., & Taniguchi, E. (2006). Travel time reliability in vehicle routing and scheduling with time windows. *Networks and Spatial Economics*, *6*(3–4), 293–311.

Campbell, A. M., & Thomas, B. W. (2008a). Advances and challenges in a priori routing. In B. Golden, R. Raghavan, & E. Wasil (Eds.), *The vehicle routing problem: latest advances and challenges, operations research/computer science interfaces* (Vol. 43, pp. 123–142). Berlin: Springer.

Campbell, A. M., & Thomas, B. W. (2008b). The probabilistic traveling salesman problem with deadlines. *Transportation Science*, *42*(1), 1–21.

Campbell, A. M., & Thomas, B. W. (2009). Runtime reduction techniques for the probabilistic traveling salesman problem with deadlines. *Computers & Operations Research*, *36*, 1231–1248.

Chao, I. M., Golden, B. L., & Wasil, E. A. (1996). A fast and effective heuristic for the orienteering problem. *European Journal of Operational Research*, *88*(3), 475–489.

Denardo, E. V. (2003). *Dynamic programming: models and applications*. Mineola: Dover.

Feillet, D., Dejax, P., Gendreau, M., & Gueguen, C. (2004). An exact algorithm for the elementary shortest path problem with resource constraints: application to some vehicle routing problems. *Networks*, *44*(3), 216–229.

Feillet, D., Dejax, P., & Gendreau, M. (2005). Traveling salesman problem with profits. *Transportation Science*, *39*(2), 188–205.

Fox, B. L., & Denardo, E. V. (1979). Shortest route methods: 1. reaching, pruning, and buckets. *Operations Research*, *27*, 161–186.

Gendreau, M., Laporte, G., & Solomon, M. M. (1995). Single-vehicle routing and scheduling to minimize the number of delays. *Transportation Science*, *29*, 56–62.

Goodson, J. (2008). *Fixed, dynamic, and hybrid routing strategies for the vehicle routing problem with stochastic demand*. Ph.D. Dissertation Proposal, Department of Management Sciences, University of Iowa.

Hansen, P., Mladenović, N., Brimberg, J., & Moreno Perez, J. A. (2010). Variable neighborhood search. In M. Gendreau & J. Y. Potvin (Eds.), *Handbook of metaheuristics, international series in operations research & management science* (Vol. 146, Chap. 3, pp. 61–86). New York: Springer.

Ilhan, S., Iravani, S. M. R., & Daskin, M. (2008). The orienteering problem with stochastic profits. *IIE Transactions*, *40*(4), 406–421.

Jula, H., Dessouky, M., & Ioannou, A. (2006). Truck route planning in nonstationary stochastic networks with time windows at customer locations. *IEEE Transactions on Intelligent Transportation Systems*, *7*(1), 51–62.

Kao, E. P. C. (1978). A preference order dynamic program for a stochastic traveling salesman problem. *Operations Research*, *26*(6), 1033–1045.

Kenyon, A. S., & Morton, D. P. (2003). Stochastic vehicle routing with random travel times. *Transportation Science*, *37*(1), 69–82.

Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research*, *24*(11), 1097–1100.

Puterman, M. L. (1994). *Markov decision processes: discrete stochastic dynamic programming*. New York: Wiley.

Russell, R. A., & Urban, T. L. (2007). Vehicle routing with soft time windows and Erlang travel times. *Journal of the Operational Research Society*, 1–9.

Schrimpf, G., Schneider, J., Stamm-Wilbrandt, H., & Dueck, G. (2000). Record breaking optimization results using the ruin and recreate principle. *Journal of Computational Physics*, *159*(2), 139–171.

Sevkli, Z., & Sevilgen, F. E. (2006). Variable neighborhood search for the orienteering problem. In *Lecture notes in computer science* (Vol. 4263, pp. 134—143). Berlin: Springer.

Tang, H., & Miller-Hooks, E. (2005). Algorithms for a stochastic selective travelling salesperson problem. *The Journal of the Operational Research Society*, *56*, 439–452.

Teng, S. Y., Ong, H. L., & Huang, H. C. (2004). An integer L-shaped algorithm for the time-constrained traveling salesman problem with stochastic travel times and service times. *Asia-Pacific Journal of Operational Research*, *21*, 241–257.

Tsiligirides, T. (1984). Heuristic methods applied to orienteering. *Journal of Operations Research Society*, *35*, 797–809.

Vansteenwegen, P., Souffriau, W., & Van Oudheusden, D. (2011). The orienteering problem: a survey. *European Journal of Operational Research*. doi:10.1016/j.ejor.2010.03.045.

Wong, J. C. F., Leung, J. M. Y., & Cheng, C. H. (2003). *On a vehicle routing problem with time windows and stochastic travel times: models, algorithms, and heuristics* (Tech. Rep.). SEEM2003-03, Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong.