IIE Transactions

ISSN: 0740-817X (Print) 1545-8830 (Online) Journal homepage: https://www.tandfonline.com/loi/uiie20

The orienteering problem with stochastic profits

Taylan İlhan , Seyed M. R. Iravani & Mark S. Daskin

To cite this article: Taylan İlhan , Seyed M. R. Iravani & Mark S. Daskin (2008) The orienteering problem with stochastic profits, IIE Transactions, 40:4, 406-421, DOI: 10.1080/07408170701592481

To link to this article: https://doi.org/10.1080/07408170701592481

IIE Transactions (2008) **40**, 406–421

ISSN: 0740-817X print / 1545-8830 online
DOI: 10.1080/07408170701592481

The orienteering problem with stochastic profits

TAYLAN İLHAN, SEYED M. R. IRAVANI and MARK S. DASKIN*

Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL 60208, USA
E-mail: m-daskin@northwestern.edu

Received October 2005 and accepted May 2007

Given a graph $G = (N, E)$, N representing the set of nodes associated with Normally distributed random profits and E representing the set of edges with associated travel times, we define the Orienteering Problem with Stochastic Profits (OPSP) as the problem of finding a tour that visits a subset of N within a prespecified time limit and maximizes the probability of collecting more than a prespecified target profit level. We develop an exact solution scheme based on a parametric formulation of the problem and present a bi-objective genetic algorithm. We also analyze the characteristics of the problem and test the algorithms computationally. The experimental results identify conditions under which the OPSP results in significant improvements in reaching the target profit when compared with the solution from the deterministic variant of the problem.

Keywords: Stochastic routing, bi-objective optimization, genetic algorithm

1. Introduction

When an Original Equipment Manufacturer (OEM), e.g., an auto manufacturer, makes a significant design change that eliminates a product line and leads to inventory at suppliers becoming obsolete, the OEM must often reimburse the supplier for the inventory on hand. However, it is clearly in the OEM's best interest to audit the inventory claims of the suppliers so that the OEM only pays for actual inventory. Suppliers may overestimate the quantity of unusable inventory they have accumulated and/or they may be able to sell some of the inventory to other firms (e.g., as repair parts) before the OEM must pay for the inventory. Clearly, the difference between the claimed inventory and the actual audited inventory is a random variable that will vary from supplier to supplier and from part to part. Judiciously chosen audits can significantly reduce the OEM's reimbursement liabilities. This study was motivated by this problem at a major US auto company.

With a limited number of auditors, the OEM must determine which suppliers to visit to maximize the recovered claims. A recovered claim is the difference between the value of the inventory claimed by the supplier and the audited inventory value. If the firm has only one auditor, the problem can be viewed as an instance of the Orienteering Problem with Stochastic Profits (OPSP). The OPSP is defined on a graph $G = (N, E)$, where N is the set of nodes and E is the set of edges that connect nodes. Each edge requires a known deterministic time to traverse. An associated profit (the recovered claim value) is received when a node is visited. However, the profits are stochastic with a known distribution and their values are not revealed before the nodes are visited. The objective is to find a subset of nodes (suppliers) to visit and a tour through those nodes that maximizes the probability with which the collected profit exceeds a target profit level, while not violating the time limit constraint.

Although the audit times may be random, in most cases the prearrangement of travel and audit schedules makes the time spent at each location effectively deterministic. For example, if the travel to the supplier location is by plane, the auditor takes the late night flight to the city, audits inventories the next day and leaves the location with the night flight on that day. Whether the audit time takes 1 hour or the whole day does not affect the total time spent at that location. In addition, audit times can be added to the travel times for modeling purposes. If s_a and s_b are the audit times for suppliers a and b and t_{ab} is the travel time between them, then one can assume zero time spent in each location and use the modified travel time $t'_{ab} = t_{ab} + (s_a + s_b)/2$.

We believe that our definition of the stochastic orienteering problem is new to the literature. However, many applications of the deterministic Orienteering Problem (OP), which was first proposed by Tsiligirides (1984), may be better formulated as examples of the OPSP, since most real-life problems inherently contain stochasticity. One application of the *deterministic* OP, given by Golden *et al.* (1987), is an inventory/routing problem in which a fleet of vehicles must periodically deliver fuel to a number of customers. In this problem, a customer's fuel level must be kept above a

*Corresponding author

minimum level at all points in time. The problem is to select a subset of customers depending on the urgency level of each, and to find the most efficient route to visit them. For the problem, the authors determine the urgency levels from the "forecasted" fuel levels of customers. This problem can be seen as an application of the OPSP since the demand for fuel at each location is actually stochastic. Vehicle capacities are an added complication of most inventory/routing problems that are not reflected in our model.

In this paper, we propose an exact parametric solution technique for the OPSP and a Pareto-based bi-objective genetic algorithm that is based on the conflict between high mean profit and low variance in a solution. Since this is the first study of the OPSP, our purpose is to identify possible solution strategies for the OPSP and to describe properties of the problem by analyzing different instances with different characteristics, rather than developing the fastest algorithm. Nevertheless, the computational results indicate that the parametric solution technique and the bi-objective genetic algorithm are promising tools to solve the OPSP.

In the following section, we present a brief review of the literature related to the OPSP. In Section 3, we define the problem and present the mathematical formulation. In Section 4, we introduce the proposed parametric solution technique. In Section 5, we outline a bi-objective genetic algorithm to solve the OPSP. In Section 6, we evaluate the performance of the proposed algorithms through a computational study. In the last section, we conclude the paper by giving final remarks.

2. Literature review

Since the problem studied in this paper is related to the OP, the stochastic knapsack problem and stochastic routing problems, we briefly cite the related literature on those areas and compare and contrast our model with the models in the literature in the above areas. The standard OP was first defined by Tsiligirides (1984). In the OP, the objective is to find a path from a starting node to an ending node so that the total profit collected from the nodes on the path is maximized while the length of the journey stays within the time constraint. If the starting and ending nodes are the same, as in the OPSP, then it is called the "rooted" OP. In the OPSP the profits are *stochastic* whereas in the OP, the profits are *deterministic*. We note that a problem with stochastic profits can be modeled as a standard OP if the decision maker wants to maximize the expected total profit. In this case, the objective can be stated as maximizing the sum of the expected profits of the visited nodes.

There are numerous studies of the *deterministic* OP in the literature including: heuristic methods (e.g., Tsiligirides (1984), Golden *et al.* (1987), Golden *et al.* (1988) and Chao *et al.* (1996)); branch-and-cut-based exact solution techniques (e.g., Fischetti *et al.* (1998) and Gendreau *et al.* (1998)); lower bounding schemes based on linear programming and minimum 1-subtree relaxations (e.g., Leifer and Rosenwein (1994) and Kataoka *et al.* (1998)) and upper bounding schemes based on the knapsack problem formulations (e.g., Laporte and Martello (1990) and Millar and Kiragu (1997)). Also, the OP was shown to be NP-hard by Golden *et al.* (1987). From the perspective of this paper, many of the techniques developed to solve the OP can be utilized in solving the OPSP since, as we show in Section 4, the OPSP can be solved by solving a series of OPs, under certain assumptions.

One of the specific applications of optimization problems with probabilistic objective functions that is related to the OPSP is the Stochastic Knapsack Problem (SKP). The objective in the SKP is to maximize the probability that the total profit will be greater than a predefined target value subject to a knapsack constraint. Variants of the SKP with Normally distributed profits are studied by Steinberg and Parks (1979), Sniedovich (1980), Henig (1990), Carraway *et al.* (1993) and Morton and Wood (1998). In terms of the objective function and its treatment, the OPSP is similar to the SKP with stochastic profits. However, the main difference between the OPSP and the SKP is that the space or time consumed by an item in the SKP is independent of other items and in this sense it is a selection problem, whereas in the OPSP, the time to reach a particular node is dependent on the node visited before it (i.e., dependent on the tour). From this perspective, the OPSP is much harder than the SKP, since it has some features of the stochastic routing problems.

Stochastic routing problems have been studied since the early 1980s. One of the most basic stochastic routing problems is the probabilistic Traveling Salesman Problem (TSP) (Jaillet, 1988), in which there are $|N|$ nodes on a graph and for any given instance of the problem only a random subset S of those nodes have to be visited. The probability of having to visit set S is $p(S)$. The objective is to find an *a priori* tour with the minimum expected length where the expectation is taken over all subsets of nodes. In this problem, all stochastic elements are known at the time of execution of the *a priori* solution. However, in the OPSP the profits are not known with certainty until we reach each node and we cannot change the route based on the observed profits. For other basic probabilistic routing problems, see Bertsimas *et al.* (1990).

Another related problem is the Stochastic Vehicle Routing Problem (SVRP). The classical vehicle routing problem is defined on a graph, where one of the nodes represents the depot at which m identical vehicles are based while the remaining vertices correspond to cities or customers with demands that need to be satisfied by the vehicles. However, all vehicles are capacitated and a vehicle must return to the depot when no capacity is left for any other city. Also, there is a travel cost between any two nodes. The objective is to find m routes that minimize the total cost while visiting all nodes just once. In the stochastic case, any of the parameters—travel costs between cities and demand

levels—can be random. The survey papers by Bertsimas and Simchi-Levi (1996) and Gendreau *et al.* (1996) summarize research on the SVRP. There are two main approaches to the SVRP: formulating it as a two-level stochastic model with recourse (as in the probabilistic TSP) and using chance constraints. Our model is closer to the latter one. However, in our study, the stochasticity is contained in the objective function of the model, not in its constraints.

The first study that includes stochastic elements in the OP is very recent. Tang and Miller-Hooks (2005) focus on an OP with stochastic service times. In their study, the travel times between customers and the profits obtained from customers are deterministic, whereas in the OPSP the profits are random. Tang and Miller-Hooks (2005) model the time spent at each customer site as a random variable with a discrete distribution. A chance constraint in their model states that the probability that the total tour time, which is the sum of the time spent on the road and at the customer sites, exceeds a threshold should be less than a prespecified service level. The authors propose a branch-and-cut algorithm that solves the problem exactly and an approximation algorithm which is called the "Construct-and-Adjust" heuristic.

3. Problem formulation

We are given a graph $G = (N, E)$, where N is the set of nodes and E is the set of edges. Node 1 represents the depot. Each node $i \in N$ other than the depot has a Normally distributed random profit W_i with mean μ_i and variance σ_i^2 , independent of the other profits. Each edge that connects node i to j has an associated non-random time, t_{ij}. The problem is to find a tour that visits a subset of the nodes and the depot and maximizes the probability that the profit collected within a given time limit, T, exceeds a given target profit level, K. We define binary decision variables x_{ij} (which equals one if the edge (i, j) is in a tour S, and zero otherwise) and y_i (which equals one if the node i is in a tour S, and zero otherwise). The formulation of the OP with stochastic profits is given as follows:

$$\text{(OPSP)}\max \quad P\left(\sum_{i\in N} W_i y_i \geq K\right), \tag{1}$$

subject to

$$\sum_{i\in N:(i,j)\in E} x_{ij} = y_j, \qquad \forall j \in N \tag{2}$$

$$\sum_{j\in N:(i,j)\in E} x_{ij} = y_i, \qquad \forall i \in N \tag{3}$$

$$\sum_{i,j\in V:(i,j)\in E} x_{ij} \leq |V| - 1, \quad \forall V \subseteq N - \{1\}, \quad V \neq \emptyset, \tag{4}$$

$$\sum_{(i,j)\in E} t_{ij} x_{ij} \leq T, \tag{5}$$

$$y_1 = 1, \tag{6}$$

$$x_{ij}, y_i \in \{0, 1\}, \qquad \forall (i, j) \in E, i \in N. \tag{7}$$

The objective function (1) maximizes the probability that the sum of the profits associated with the selected nodes is greater than or equal to the predefined target or quota K. Constraints (2) and (3) are the assignment and linkage constraints. Constraint (4) is the subtour elimination constraint and constraint (5) is the time constraint. Constraint (6) ensures that the depot is on the tour and constraints (7) are integrality constraints.

Target levels are often used by managers as a metric to evaluate the performance of a system (i.e., sales or profit targets). A target is generally set so that it is challenging but also not exceptionally difficult to attain. We do not think that a manager can justify a target level K to his/her supervisors if there is a high possibility of reaching $10K$. Conversely, setting obviously high target levels would not be reasonable since unrealistic objectives can be discouraging and difficult to justify when they are consistently missed by large margins. Thus, achieving a reasonable and well designed target is a desirable objective function in many situations.

Having a time limit, T, on a tour is common in transportation systems. The limit may have natural, legal and financial reasons. For example, in planning daily routes, the decision maker must complete the day's deliveries by a particular hour. There are also legal limits on the drivers' total time on the road; i.e., time laws to prevent accidents related to *driver fatigue*. Finally, some firms have policies that restrict the amount of time an employee can spend in the field so that they can avoid overtime charges.

The OPSP is NP-hard, since its special case, in which all profits are deterministic, is NP-hard. When there is no variability in the profits, the objective function can take only two values, either zero or one. In other words, this special case seeks an answer to the following question: can we find a solution whose value is greater than or equal to K or not? Clearly this is the decision problem version of the OP and solving it is as hard as solving the OP. Since the OP is NP-hard, its decision problem version, which is a special case of the OPSP, is also NP-hard. This proves that the OPSP is an NP-hard problem.

4. An exact algorithm

Since the profits are Normally distributed, we can restate the problem by replacing the probabilistic objective function with its deterministic equivalent as follows (Birge and Louveaux, 1997):

$$\text{(Q)} \quad \max\left\{\frac{m_y - K}{\sqrt{v_y}} : y \in Y\right\}, \tag{8}$$

where Y represents the set of $y = (y_1, \ldots, y_{|N|})$ values for which there exists an $x = (x_1, \ldots, x_{|N|^2})$ satisfying the constraints (2)–(7) and $m_y = \sum_{i\in N} \mu_i y_i$ and $v_y = \sum_{i\in N} \sigma_i^2 y_i$.

To solve (Q) as defined in Equation (8), we follow the concept developed by Geoffrion (1967) for bi-criteria mathematical problems and its application to the SKP by Henig (1990). Geoffrion (1967) suggests a parametric solution approach to solve $Z = \max\{h(x) = g(f_1(x), f_2(x)) : x \in X\}$ assuming h is a quasi-concave function; i.e., $h(\alpha x_1 + (1-\alpha)x_2) \geq \min\{h(x_1), h(x_2)\}$ for $0 \leq \alpha \leq 1$. Geoffrion proves that there is an $\alpha^* \in [0, 1]$ such that $Z = Z(\alpha^*) = \max\{\alpha^* f_1(x) + (1-\alpha^*) f_2(x) : x \in X\}$. Henig (1990) applies a similar idea to the SKP, whose objective function is the same as the objective function of (Q) and proves that the objective function, $(m_y - K)/\sqrt{v_y}$, is quasi-convex under the condition $m_y \geq K$. The idea is based on finding the convex hull of the feasible (m, v) values, named H and searching its extreme points for the optimal solution to (Q). The following proposition is due to Henig (1990):

Proposition 1. *Let* (Q) *be defined as* $\max\{(m_y - K)/\sqrt{v_y} : y \in Y\}$ *and* H *be the convex hull of* $\{(m_y, v_y) : y \in Y\}$ *where* Y *is a set of feasible solutions. If* $\max\{m_y : y \in Y\} \geq K$, *then there exists an extreme point of* H *that is optimal for* (Q).

In the rest of the paper, we assume that $\overline{m} = \max\{m_y : y \in Y\} \geq K$. This assumption is reasonable since if $m_y < K$, the probability of collecting a profit more than K cannot be more than 0.5 with Normally distributed profits. If our assumption does not hold (i.e., there is no solution whose total mean profit is greater than K), then (Q) favors solutions whose total mean profit and total variance are *both* as high as possible. Moreover, a solution which has a lower mean profit and a higher variance than another solution may be optimal and this optimal solution may not be an extreme point of H. However, as indicated above, such target levels are rare in most business environments since high target levels lead employees and managers to take more risk and cause stress and discouragement among the employees (see Stroh *et al.* (2002, p. 347)). Under this assumption, (Q) favors solutions whose total mean is as large as possible and whose total variance is as small as possible.

Thus, in light of Proposition 1, it is enough to generate the mean–variance pairs, (m_y, v_y), that form the convex hull, and then to pick the solution that maximizes the objective function of (Q). Moreover, we are interested only in the part of the convex hull where $K < m_y \leq \overline{m}$ and $0 < v_y \leq \overline{v}$ (where $\overline{v}$ is the total variance of the solution that has the total mean $\overline{m}$).

To generate the extreme points of H, we use *the weighted sum method*, which is a well-known bi-objective optimization method (Cohon, 1978). To find the Pareto-optimal points for the bi-objective problem $\max\{(m_y, -v_y) : y \in Y\}$, we define the following parametric problem with $0 \leq \alpha \leq 1$:

$$(\mathrm{QA}(\alpha))\ \max\{\alpha m_y - (1-\alpha)v_y : y \in Y\}, \tag{9}$$

or equivalently

$$\max\left\{\sum_{i \in N}\left(\alpha\mu_i - (1-\alpha)\sigma_i^2\right)y_i : y \in Y\right\}. \tag{10}$$

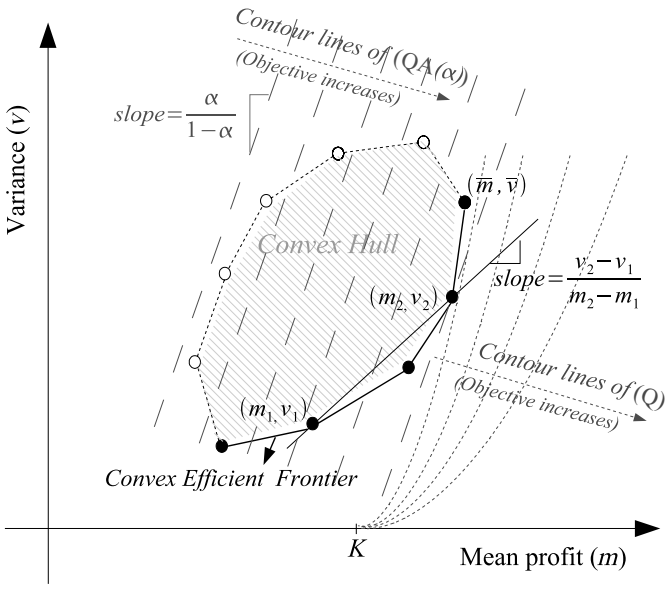

Fig. 1. Illustration of the convex hull formed by the feasible mean-variance, (m, v), solutions of (Q), the contour lines of the objective functions of (Q) and (QA(α)) over the (m, v) space and their relations.

Note that the objective function of (QA(α)) for $\alpha = 1$ is $\max_{y \in Y} m_y$ and the problem is equivalent to the deterministic OP, which uses the expected profits to find a solution tour. The solution of this problem provides us a $(\overline{m}, \overline{v})$ pair. Also, this problem may have multiple optimal solutions, which all have the same $\overline{m}$ value but different $\overline{v}$ values. However, this does not pose any problem for our exact algorithm since it eventually finds the solution with the minimum $\overline{v}$ value.

An illustrative example of the convex hull, efficient frontier and their relations with the problems (Q) and (QA(α)) is given in Fig. 1. Proposition 1 indicates that one of the extreme points of H is the optimal solution of (Q). Also, since its objective function value increases as m and $-v$ increase, the optimal solution lies on the convex efficient frontier. As shown in Fig. 1, the slope of the contour lines of the objective function of (QA(α)), which is $\alpha/(1-\alpha)$ (see Deb (2001)), increases as α increases.

The exact algorithm is based on selecting different α values and solving the subproblem (QA(α)). However, selecting appropriate α values is critical. The naive approach is to generate many different α values without any knowledge of the possible solutions and then to solve the subproblem for each of them. This approach may result in solving the problem for many values of α for which the solution is not needed. At the same time, the naive approach may fail to find all solutions on the efficient frontier, as critical α values may be skipped. The alternative approach is to generate new α values recursively by evaluating the solutions obtained from the subproblem for the previously generated α values. We adopt the latter approach since it is more efficient and guarantees that we generate all extreme

points. Before presenting this recursive approach, we first show that if (QA(α)) has the same optimal solution for two different α values, say α_1 and α_2, then (QA(α)) has the same optimal solution for any $\alpha \in [\alpha_1, \alpha_2]$. This result immediately follows from the monotonicity results given below.

Proposition 2. *The total mean profit, $m_{y(\alpha)}$ and total variance, $v_{y(\alpha)}$, obtained by solving* (QA(α))*, are monotonically nondecreasing functions of* α $(0 < \alpha < 1)$.

Proof. Assume $0 < \alpha_1 < \alpha_2 < 1$ and let $y(\alpha_1)$ and $y(\alpha_2)$ be the corresponding solutions of (QA(α)). For the sake of simplicity, let (m_i, v_i) be equivalent to $(m_{y(\alpha_i)}, v_{y(\alpha_i)})$, $i = 1, 2$. Since (m_i, v_i) is optimal for (QA(α_i)),

$$\alpha_1 m_1 - (1-\alpha_1)v_1 \geq \alpha_1 m_2 - (1-\alpha_1)v_2, \tag{11}$$

$$\alpha_2 m_2 - (1-\alpha_2)v_2 \geq \alpha_2 m_1 - (1-\alpha_2)v_1. \tag{12}$$

Since $\alpha_i > 0$, $i = 1, 2$, we can rewrite inequalities (11) and (12) as follows:

$$m_1 - \frac{1-\alpha_1}{\alpha_1}v_1 \geq m_2 - \frac{1-\alpha_1}{\alpha_1}v_2, \tag{13}$$

$$m_2 - \frac{1-\alpha_2}{\alpha_2}v_2 \geq m_1 - \frac{1-\alpha_2}{\alpha_2}v_1. \tag{14}$$

Subtracting the right-hand side of Equation (14) from the left-hand side of Equation (13) and subtracting the left-hand side of Equation (14) from the right-hand-side of Equation (13), we obtain:

$$\left(\frac{1-\alpha_2}{\alpha_2} - \frac{1-\alpha_1}{\alpha_1}\right)v_1 \geq \left(\frac{1-\alpha_2}{\alpha_2} - \frac{1-\alpha_1}{\alpha_1}\right)v_2. \tag{15}$$

Since

$$\left(\frac{1-\alpha_2}{\alpha_2} - \frac{1-\alpha_1}{\alpha_1}\right) < 0,$$

it follows that $v_1 \leq v_2$. This result and inequality (12) imply that $m_1 \leq m_2$. ■

Proposition 3. *Let (m_i, v_i) be the optimal solution of* (QA(α_i)), $i = 1, 2, 3$. *Also, assume* $\alpha_3 \geq \alpha_2 \geq \alpha_1 \geq 0$. *If* $(m_1, v_1) = (m_3, v_3) = (m, v)$, *then* $(m_2, v_2) = (m, v)$.

Proof. The monotonicity result (Proposition 2) implies $m_1 \leq m_2 \leq m_3$ and $v_1 \leq v_2 \leq v_3$. Then, if $m_3 = m_1 = m$, we have $m_2 = m$. Similarly, $v_3 = v_1 = v$ gives $v_2 = v$. ■

Now, we can construct the exact algorithm described in Fig. 2. The algorithm takes the solution $(K, 0)$ and the solution of the deterministic OP as starting points, generates the Convex Efficient Frontier (CEF) and then selects the optimal solution from among the solutions on the frontier. Clearly, the point $(K, 0)$ may not be on the CEF. However, given the shape of the contour line and the increasing direction of the objective function (see Fig. 1), the points that are on the left side and just on a line passing through $(\overline{m}, \overline{v})$ and $(K, 0)$ cannot be optimal for (Q). Thus, we can find all necessary solutions on the relevant range of the convex hull H by starting the algorithm from $(\overline{m}, \overline{v})$ and $(K, 0)$.

The value of α in Step 1 of the function *FindEfficientPoint* in Fig. 2 is found by using the slope of the line that passes through two adjacent solutions, (m_1, v_1) and (m_2, v_2), as follows:

$$\alpha m_1 - (1-\alpha)v_1 = \alpha m_2 - (1-\alpha)v_2 \Longrightarrow$$
$$\alpha = \frac{v_2 - v_1}{m_2 - m_1 + v_2 - v_1}. \tag{16}$$

The proposed algorithm terminates in a finite number of steps. If there are k extreme points, (m, v), in H, the recursive function in Fig. 2 is called at most $2k - 1$ times. The obvious drawback of this algorithm is that, each time the recursive function is called, it requires the solution of a deterministic OP, which is NP-hard. One alternative to overcome this problem is to use lower and upper bounds instead of solving the OP exactly each time. In our implementation, we solved the Mixed Integer Programming (MIP) formulation by using the MIP solver in the CPLEX 8.1 callable libraries. To deal with the exponential number of subtour elimination constraints (4), we employed a simple iterative procedure common in solving routing problems. We first solve the problem without those constraints and then, if there are subtours, we add the constraints to eliminate them and resolve the problem; otherwise we

Step 1. Initialization: Set $(m_1, v_1) = (K, 0)$ and $(m_2, v_2) = (\overline{m}, \overline{v})$; where $(\overline{m}, \overline{v})$ is obtained by solving (QA(1)). Then; set $CEF = \{(m_1, v_1), (m_2, v_2)\}$.

Step 2. Recursion: Call the function *FindEfficientPoint*$\{(m_1, v_1), (m_2, v_2), CEF\}$

Step 3. Selection of the optimum: Set $(m^*, v^*) = argmax\{(m - K)/\sqrt{v} \mid (m, v) \in CEF\}$

The function *FindEfficientPoint*$\{(m_1, v_1), (m_2, v_2), CEF\}$:

Step 1. Set $\alpha = (v_2 - v_1)/[(m_2 - m_1) + (v_2 - v_1)]$ and solve (QA(α))

Step 2. If the optimal solution $(m_{y(\alpha)}, v_{y(\alpha)}) = (m_1, v_1)$ *or* (m_2, v_2) then exit; otherwise:

2.1. Let $CEF = CEF \cup (m_{y(\alpha)}, v_{y(\alpha)})$

2.2. Call *FindEfficientPoint*$\{(m_{y(\alpha)}, v_{y(\alpha)}), (m_2, v_2), CEF\}$

2.3. Call *FindEfficientPoint*$\{(m_1, v_1), (m_{y(\alpha)}, v_{y(\alpha)}), CEF\}$

Fig. 2. Outline of the proposed exact algorithm.

stop. Clearly, this approach performs worse than the specific branch-and-cut algorithms developed for the OP (Fischetti *et al.*, 1998; Gendreau *et al.*, 1998) in terms of computation time. Therefore, using branch-and-cut algorithms within the proposed algorithm would improve its performance significantly. However, reimplementing and testing these specific algorithms is not the focus of this paper.

In the parametric (or weighted-sum) approach given here, we solve one problem at a time with different weights and generate a CEF. Our hope is that the number of extreme points on the CEF will not be too large. However, even using branch-and-cut algorithms to solve the subproblems will not eliminate the problem that the worst-case running time of the algorithm increases exponentially with respect to the number of nodes since the deterministic OP is NP-hard. As a result, it is clear that we also need an algorithm to deal with large instances of the OPSP. Another way to solve the OPSP is to use bi-objective evolutionary algorithms. Although these algorithms do not guarantee the global optimum, they generally terminate with a good local optimum. The advantage of using evolutionary algorithms is that they maintain a population of solutions that is used to carry the traits of good solutions to the next generation. As such, they do not necessitate solving a single-objective problem several times to create a Pareto-optimal frontier. In the next section, we present such an algorithm.

5. The bi-objective genetic algorithm

A Genetic Algorithm (GA) is an evolutionary algorithm that imitates the natural selection process of living organisms. A standard GA starts with a set (*population*) of encoded solutions (or *chromosomes*), which are either structured or randomly generated initially. These solutions do not have to be good or even feasible at the beginning. After construction of the initial population, the GA generates *offspring* (a new set of solutions) from the chromosomes in the *parent* population. This reproduction process is carried out with a *crossover* operator, which recombines parts of two parent chromosomes to generate new chromosomes. To increase the diversity of the solutions and to avoid local optima, a GA may employ a *mutation* operator which modifies a child chromosome with a small probability during the reproduction process. A GA measures the quality of a chromosome by using a *fitness function*, which is based on the objective function value and the feasibility of the corresponding solution. After the generation of the offspring, some chromosomes in both the parent population and the offspring are eliminated according to their relative fitness function values. The remaining chromosomes form a new population, which will be the next parent population. As the algorithm evolves from generation to generation, solutions converge to good feasible ones since the solutions with the best objective function values are more likely to be selected as parent solutions in forming new solutions for the next generation. Thus, the characteristics of the better solutions are more likely to be carried forward than are the characteristics of the inferior solutions. The GA continues until a stopping criterion is reached.

Although our problem has a single-objective function, which is to maximize the probability of attaining a target profit level, it can also be solved as a bi-objective optimization problem, in which the objectives are to maximize the mean profit and to minimize the variance of the profit. Thus, we propose a Multi-Objective Genetic Algorithm (MOGA) to solve the OPSP heuristically. A MOGA has a structure similar to that of a single-objective GA. However, since we have more than one objective function, we are interested in finding a set of efficient solutions rather than a single solution. We want this set to be as close to the optimal Efficient Frontier (EF) as possible while maintaining diversity in terms of the spread of solutions across the entire EF. For a discussion of the techniques for these purposes and a general discussion of MOGAs, we refer the reader to Coello (2000), Veldhuizen and Lamont (2000), Deb (2001) and Knowles and Corne (2005). Since the proposed MOGA does not ensure optimality, in the rest of the paper, when we refer the EF (or CEF) generated by the MOGA, we mean the "approximate" EF (or CEF).

To generate solutions close to the optimal EF, we use a rank-based method, which is very similar to the method developed by Fonseca and Fleming (1993). Under our assumptions, solution i is dominated by solution j, which we represent by $i \succ j$, if $m_i < m_j$ and $v_i > v_j$. The rank of solution j, D_j, counts the number of solutions that dominate solution j, i.e., $D_j = \sum_{i \in P} \Pi(i \succ j)$, where P is the solution population and $\Pi(i \succ j) = 1$ if $i \succ j$, and $\Pi(i \succ j) = 0$, otherwise. Since we are more interested in solutions that correspond to the points of the CEF (i.e., the extreme points of the EF), we define a *modified* rank for solution j, r_j, and set it to $D_j + I_j$, where $I_j = 0$ if the (m_j, v_j) of solution j is a point on the CEF and $I_j = 1$, otherwise. This modified ranking system ensures that the extreme points have higher priority and survive to the next generations.

Clearly, if $D_j \geq 1$ then $I_j = 1$, since if a point is not on the EF then it is also not on the CEF. The rule to determine if a point on the EF is also a point on the CEF is as follows. Consider three efficient points i, j and k such that $v_i < v_j < v_k$. If the slope of the line passing through points i and j is greater than the slope of the line passing through points i and k, then point j is not an extreme point on the convex hull. Thus, we can define $I_j, j \in EF$, as follows (disregarding the trivial cases in which $m_k = m_j$ and/or $m_k = m_i$):

$$I_j = \begin{cases} 1 \text{ if } \exists\, i, k \in EF \text{ such that } v_i < v_j < v_k \text{ and} \\ \qquad (v_k - v_i)/(m_k - m_i) < (v_j - v_i)/(m_j - m_i), \\ 0 \text{ otherwise.} \end{cases} \quad (17)$$

Also, to ensure diversity of the solutions we employed the *niche count* measure (see Deb (2001) for details). The niche count of a solution is a measure of crowding near that

solution. As the number of solutions whose (m, v) values are close to a particular solution's (m, v) value and the level of their closeness increase, the niche count of that solution also increases. We use this measure as we select the parents for crossover operations and to construct the next generation. We do not give the details of the derivation of the niche count here; rather, we limit ourselves to giving the required equations. The niche count of solution i, nc_i, is the sum of the *sharing function* values, $Sh(d_{ij})$, of the solutions, which are within the *sharing radius*, θ_{share}, of solution i and which have the same modified rank of solution i. Formally, if we define $m_{\min}$ ($v_{\min}$) and $m_{\max}$ ($v_{\max}$) as the minimum and the maximum expected profits (variances) corresponding to solutions in a population, then $nc_i = \sum_{j:r_j=r_i, i\neq j} Sh(d_{ij})$ where:

$$d_{ij} = \sqrt{\left(\frac{m_i - m_j}{m_{\max} - m_{\min}}\right)^2 + \left(\frac{v_i - v_j}{v_{\max} - v_{\min}}\right)^2}$$

$$Sh(d_{ij}) = \begin{cases} 1 - \left(\dfrac{d_{ij}}{\theta_{\text{share}}}\right) & \text{if } d_{ij} \leq \theta_{\text{share}}, \\ 0 & \text{otherwise.} \end{cases} \tag{18}$$

In the subsections below, we present the GA operators in the following order: the encoding and decoding schemes, the initial population construction method, parent selection, crossover, mutation and population preserving algorithms. Then, in Fig. 3, we summarize the proposed GA and how it utilizes these operators.

5.1. *Encoding and decoding schemes*

The encoding and decoding schemes define how we store a solution in a chromosome and how we obtain a solution from a chromosome, respectively. We employed the random-key encoding scheme, as suggested by Bean (1994), in our GA. In this scheme, each chromosome is composed of $|N|$ genes as a list of real numbers (or *keys*), which are randomly picked from the interval [0,1]. The indices of the genes correspond to the depot and the customer nodes in the OPSP (i.e., node number = gene index). The relative values of the keys contained in a chromosome give the relative positions of the nodes in a tour. For example, if the second and sixth genes in a chromosome contain the keys 0.31 and 0.15, respectively, then in a tour represented by that chromosome, node 6 will precede node 2. We use the following algorithm to decode a chromosome to form a solution.

Step 1. Sort the genes by their random keys in ascending order.

Step 2. Starting from the beginning of the sorted chromosome, construct a tour by inserting nodes into the tour, while keeping the length of the tour less than $1.5 \times T$. We inflate the time limit because tours generated from the sorted chromosome are usually very inefficient and their lengths can be substantially decreased in the next step.

Step 3. Apply the 2-Opt improvement heuristic to the tour obtained from Step 2. This generates a tour which does not cross itself by switching the order of appropriate nodes in the tour.

Step 4. After the 2-Opt algorithm, the one-to-one relationship between the relative order of nodes in the solution tour and the relative values of corresponding keys in the chromosome breaks down. To restore this relationship, rearrange the keys in the chromosome according to the tour generated by the 2-Opt algorithm so that the relative values of the keys still

Initialization	*Step 0.*	Define limits on the population size (p_{size}), the number of offspring generated (q_{size}), runtime ($rt_{\max}$), the maximum number of consecutive generations with no improvement ($ni_{\max}$), mutation probability (p_{mutate}) and mutation strength (θ_{mutate}).
	Step 1.	Generate the initial population P which contains p_{size} chromosomes. Set the current time $rt = 0$ and, the number of steps with no improvement in the CEF $ni = 0$.
	Step 2.	Set the offspring $Q = \emptyset$.
Offspring generation	*Step 3.*	Select two parents from P using our *binary tournament selection* approach and generate two offspring, c_1 and c_2, by *two-point crossover*.
	Step 4.	Mutate c_1 and c_2 with probability p_{mutate} by using the *Gaussian mutation* operator with strength θ_{mutate} and apply the *2-Opt improvement heuristic* on c_1 and c_2.
	Step 5.	If the tour represented by c_i, $i = 1, 2$ is unique, then set $Q = Q \cup \{c_i\}$.
	Step 6.	If $\|Q\| < q_{\text{size}}$ (i.e., the set of offspring is not filled yet), then go to Step 3; else go to Step 7.
Population preservation	*Step 7.*	Set $R = P \cup Q$ and apply a *rank-and-niche-based sorting procedure* on R and set its first p_{size} elements as P.
Termination criteria	*Step 8.*	If the CEF contained in P is improved, set $ni = 0$; else $ni = ni + 1$.
	Step 9.	If $ni > ni_{\max}$ or $rt > rt_{\max}$, then Stop; else go to Step 2.

Fig. 3. Outline of the proposed bi-criteria GA.

represent the order of the nodes in the tour. (See the example below.)

Step 5. If the solution tour is not feasible, drop nodes from the end of the tour one by one until feasibility is attained.

Before giving an example of the decoding scheme, let the ith gene of a chromosome with a key value a be represented as i_a. Now, consider the following chromosome containing ten genes, (1_0, $2_{0.11}$, $3_{0.82}$, $4_{0.54}$, $5_{0.29}$, $6_{0.45}$, $7_{0.71}$, $8_{0.65}$, $9_{0.84}$, $10_{0.43}$). After sorting the chromosome, it becomes (1_0, $2_{0.11}$, $5_{0.29}$, $10_{0.43}$, $6_{0.45}$, $4_{0.54}$, $8_{0.65}$, $7_{0.71}$, $3_{0.82}$, $9_{0.84}$). To illustrate the rest of the algorithm, assume that we can visit at most six cities without violating the inflated travel time limit ($1.5 \times T$) and the solution tour is (1, 2, 5, 10, 6, 4, 8, 1). Assume further that after the 2-Opt improvement heuristic the tour becomes (1, 2, 5, 8, 4, 6, 10, 1). Then, we update the chromosome by changing the order of the keys stored in genes 4, 6, 8 and 10. The revised (and sorted) chromosome becomes (1_0, $2_{0.11}$, $5_{0.29}$, $8_{0.43}$, $4_{0.45}$, $6_{0.54}$, $10_{0.65}$, $7_{0.71}$, $3_{0.82}$, $9_{0.84}$). If the travel time does not fall below T after the 2-Opt algorithm, then we drop node 10 and the tour becomes (1, 2, 5, 8, 4, 6, 1). If it still violates the travel limit, we continue dropping nodes until feasibility is achieved. We note that, to not interfere with the order of nodes induced by the keys any further, we did not employ a greedy algorithm in Step 5. Too much manipulation of a solution encoded in the keys may severely limit the ability of the GA to converge to good solutions.

The encoding and decoding schemes presented here inherently prevent a possible duplication of nodes in a tour during the reproduction process. The reason is that, when we generate new child chromosomes, the random keys contained in the genes change but the indices of the genes, from one to $|N|$, stay the same. As a result, the same node cannot be represented in a chromosome more than once. Finally, although we called our evolutionary algorithm a GA, since we use the 2-Opt improvement heuristic in Step 3, it is actually called a "Hybrid" GA or a "Memetic" algorithm as discussed in the GA literature.

5.2. *Initial population construction*

The first step of the proposed GA is to construct an initial population of solutions from which the subsequent generations are to be produced. We generate 75% of the chromosomes in the initial population randomly by picking the value of each gene from the standard Uniform distribution. The remaining quarter of the initial population is filled with chromosomes that are generated by the following *randomized* cheapest insertion algorithm.

Step 1. Set the depot point as the initial tour, $S = \{1\}$, and $B = N - \{1\}$, where S represents the ordered set of nodes in the tour and B is the set of nodes not in the tour.

Step 2. $\forall w \in B$, calculate the cheapest insertion cost to the current tour, which is $c(w) = \min\{t_{iw} + t_{wj} - t_{ij} : (i, j) \in E\}$.

Step 3. Choose the node with perturbed minimum insertion cost $w^* = \text{argmin}\{c'(w) : c'(w) = c(w) + c(w) \times U(-0.3, 0.3), w \in B\}$, where $U(-0.3, 0.3)$ is a Uniform random varible with parameters -0.3 and 0.3.

Step 4. If ($c(w^*)$ + Length of tour S) $> T$, then stop, insertion is not possible, go to Step 6; else go to Step 5.

Step 5. Set $S = S \cup \{w^*\}$, and $B = B - \{w^*\}$, and go to Step 2.

Step 6. Randomly generate $|N|$ numbers from the interval [0, 1] for each solution and sort them in increasing order. Assign the first $|S|$ of them to the genes represented in tour S, following the order of the nodes in the tour. Assign the remaining numbers to the rest of genes in the order of their indices.

5.3. *Parent selection*

Two parent solutions are selected from the population by using a binary tournament method, which is similar to the crowded binary tournament method developed by Deb (2001). The formal algorithm is as follows:

Step 1. Select two chromosomes i and j from P randomly.

Step 2. If $r_i \neq r_j$ then set the chromosome with the lowest modified rank as the first parent, else

2.1. If $nc_i < nc_j$ then set i as the first parent, else set j as the first parent.

Step 3. Repeat Steps 1 and 2 until the second parent differs from the first parent.

5.4. *Crossover*

After testing different types of crossover operators such as single-point, two-point, uniform and fusion crossover operators (see Deb (2001) for the details of these operators), we found that the two-point crossover operator worked better than the others. Thus, in our GA, we use two-point crossover and preserve both child chromosomes in the set of offspring. For example, if the two parent chromosomes are (1_0, $2_{0.01}$, $3_{0.31}$, $4_{0.11}$, $5_{0.21}$) and (1_0, $2_{0.02}$, $3_{0.52}$, $4_{0.32}$, $5_{0.92}$), and crossover points are just after the second and fourth genes, then the child chromosomes are (1_0, $2_{0.01}$, $3_{0.52}$, $4_{0.32}$, $5_{0.21}$) and (1_0, $2_{0.02}$, $3_{0.31}$, $4_{0.11}$, $5_{0.92}$). After each crossover, if the tour obtained from a child chromosome is already in the population or the offspring generated up to that point, then we discard that child chromosome.

If the OPSP is defined as a symmetric problem (which is the case in our numerical study), then a sequence of nodes can be visited from the first to the last or from the last to the first within that sequence. Thus, to prevent having the same tour with both the forward and reverse directions in the population, we can follow a simple rule: the node

preceding the depot must have a higher index than the node that follows the depot in the tour. For example, the tour (1, 10, 4, 5, 2, 1) violates the rule and has to be reversed. After reversing, it becomes (1, 2, 5, 4, 10, 1). The random keys are also reversed in the manner explained in Section 5.1.

5.5. *Mutation*

To increase the diversity in the population, we use the Gaussian mutation operator on each child chromosome: each gene of a chromosome is mutated with probability p_{mutate} by adding an amount picked from the Normal distribution $N(0, \theta^2_{\text{mutate}})$ to its key, where θ_{mutate} represents the mutation strength. If the new key value in the gene is less than zero then we assign it a new value just above zero (i.e., 10^{-6}) to ensure that the depot always has the minimum key value.

5.6. *Population preservation*

After generating an offspring population of size q_{size} from the parent population of size p_{size}, we combine them into one population of size $p_{\text{size}} + q_{\text{size}}$. Then, we calculate the modified ranks of chromosomes based on the (m, v) values of the tours they represent and sort the solutions in the combined population by their modified ranks. In addition to this, within each group of solutions having the same modified rank, we sort chromosomes by their niche count in ascending order. Then, we select the top p_{size} elements of the sorted population as our next parent generation.

6. Computational results

In this section, we summarize the results of a numerical study designed to analyze the structural properties of the OPSP and the effectiveness of the proposed solution algorithms. The proposed Exact Solution Algorithm (ESA) and the bi-objective GA were coded in C and run on a Sun Ultra *5* workstation. The values of the parameters used in the bi-objective GA are presented in Table 1.

We analyzed the OPSP by using four problem sets (named Set 1 through Set 4). The characteristics of each problem set will be discussed as we present the results for each problem set. In Section 6.1, we investigate whether solving the OPSP is beneficial compared to solving the OP with average profits; i.e., the Value of the Stochastic Solution (VSS) which is defined as the difference between the value of the objective function for the optimal stochastic solution and that for the solution of the deterministic OP with expected profits (Birge and Louveaux, 1997). We use the ESA on small problem instances (Set 1) to analyze the effect of the target profit level, K, the time limit, T, and the spatial distribution of mean profits and standard deviations among the nodes on the value of the objective function. In Section 6.2, we show how the runtime of the ESA increases as the number of nodes in the OPSP increases by using 15-node to 32-node problem instances (Set 2). Then, we run the GA on Set 1 and compare its results with the results obtained from the ESA. Finally, in Section 6.3 we investigate some characteristics of approximate CEFs obtained by the GA for large problem instances (Set 3). We also study how quickly the number of points on the approximate CEF increases as the number of the nodes in the OPSP increases by using 25-node to 150-node problem instances (Set 4).

Table 1. The values of parameters in the proposed bi-objective GA

θ_{share}	= 0.1	p_{size}	= $4 \times \lvert N \rvert$
p_{mutate}	= 0.01	q_{size}	= $4 \times \lvert N \rvert$
θ_{mutate}	= 0.05	$ni_{\max}$	= 200
		$rt_{\max}$	= 2000 seconds

6.1. *VSS*

We ran the ESA on Set 1 to evaluate the advantages of solving the OPSP instead of solving the OP formulation, which maximizes the expected total profit. The coordinates of the nodes were the same in all instances of Set 1 as depicted in Fig. 4 for two cases in Set 1. To incorporate the effect of the spatial distribution of mean profits and standard deviations among the nodes, we built six different profit and standard deviation instances. In the first three cases (named 1A, 1B and 1C), all nodes had the same mean profit values but different standard deviations. In the last three cases (1D, 1E and 1F) the mean profit values also varied among the nodes. In cases 1A, 1B and 1C, the mean profits of the nodes were equal to ten, and in cases 1D, 1E and 1F, the nodes had mean profits drawn from a discrete Uniform distribution on the integers from ten to 20, inclusively. In all six cases, the variances for about half of the nodes were selected from the interval [16, 25] and the variances for the other half were selected from the interval [225, 400] by using a discrete Uniform distribution. Numerical values for Set 1 are given in Ilhan *et al.* (2006). Note that for some cases, due to a high coefficient of variation, it was possible for some nodes to have negative realized profits. In the context of auditing, this means the actual inventory levels are higher than the levels reported by the supplier, and as a consequence, the OEM has to pay more than the anticipated amounts. In cases 1A, 1B, 1D and 1F, the location of a node determined whether the node was a low or high mean (low or high variance) node; in the other two cases, low and high mean (variance) nodes were selected randomly. We solved each of these cases for five different travel time limits; i.e., T =150, 200, 250, 300 and 350. We set the target profit level, K, to zero so that the algorithm finds all of the extreme points on the CEF between (0, 0) and $(\overline{m}, \overline{v})$. We could then analyze which one is optimal for different values of K.

The optimal (m, v) values on the CEF found by the ESA are given in the upper portion of Table 2. The multiple solutions given for each case (1A to 1F) and each

Table 2. The optimal (m, v) values on the CEF found by the ESA (the upper part of the table) and the best (m, v) values found by the proposed GA (the lower part of the table) for problems in Set 1. The corresponding intervals of the target level, K, are excluded. See Ilhan *et al.* (2006) for tables including both optimal points and the intervals

		1A		*1B*		*1C*		*1D*		*1E*		*1F*	
Algorithm	*T*	*m*	*v*	*m*	*v*	*m*	*v*	*m*	*v*	*m*	*v*	*m*	*v*
The ESA	150	80	163	70	144	60	112	108	144	123	146	96	119
		80	498#	80	723	70	381	112	163	133	733	99	130
				80	728#	80	821	133	1422			114	382
						80	1526#					129	821
												133	1008
	200	100	202	100	207	80	157	147	204	164	209	132	162
		110	493	110	483	110	1124	164	493	179	483	175	1124
		110	1396#	110	1626#	110	1929#	184	2363	184	794	184	1680
	250	140	296	120	250	100	198	193	271	185	250	160	200
		140	2387#	130	584	140	1480	201	297	222	1152	213	1148
				140	1092	140	2349#	209	603	235	1651	235	1856
				140	2000#			235	3478			235	2090#
	300	160	337	150	319	120	241	228	337	235	320	190	241
		170	666	160	600	160	1491	248	666	249	601	274	2152
		170	3208#	170	1174	170	2116	262	1612	261	1209		
				170	2271#	170	3014#	268	2484	274	2279		
								273	3539				
								274	4106				
	350	160	337	160	342	140	282	228	337	245	342	215	282
		170	562	200	1534	200	2234	288	1307	262	567	283	1591
		200	1631	200	2649#	200	3468#	310	2559	295	1205	302	2161
		200	3277#					313	3241	309	1902	313	2716
								314	3604	313	2133	314	3099
										314	3178		
The bi-objective GA	150	80	163	70	144	60	112	108	144	123	146	96	119
				80	724	70	381	112	163	133	733	99	130
						80	821	133	1422			114	382
												129	821
												133	1008
	200	100	202	100	207	80	157	147	204	164	209	132	162
		110	493	110	483	100	733	164	493	179	483	163	866
								172	1757			168	1045
												169	1100
												171	1351
	250	130	269	120	250	100	198	193	271	185	250	160	200
		140	2853	140	1092	140	1480	201	297	210	830	210	1161
								209	603	235	1907	219	1455
								222	3149				
	300	120	241	150	320	120	241	228	337	235	320	190	241
		160	1750	160	605	160	1750	248	666	249	601	274	2152
		170	2414	170	1233	170	2414	270	3794	270	1967		
	350	160	342	160	342	140	282	228	337	245	342	215	282
		200	1534	200	1534	190	2078	288	1307	265	661	291	1830
						200	2716	291	1557	279	952	297	2084
								305	4185	292	1271	308	2737
										303	1602		
										304	1851		

#These points correspond to the solutions of the (QA(α)) for $\alpha = 1$; i.e., they are the (m, v) values of the deterministic OP that uses mean profit levels as input data. Although, they are not on the CEF since they are dominated by lower variance points, we include them in this table to emphasize how the deterministic OP fails to find the solutions with lower variance without reducing the mean profit level.

time limit (150 to 350) correspond to solutions found for different target value intervals, which are all provided in Ilhan *et al.* (2006). For any specific target value K, the optimal solution and the optimal objective function value can be obtained from Table 2. For example, for 1D and T = 150, there are only three possible optimal solutions for $K \geq 0$, corresponding to $(m, v) = (108, 144)$, (112, 163) and (132, 1422). If $K = 100$, then we can calculate $z = (m - K)/\sqrt{v}$ value for each (m, v) point, i.e., $z_1 = (108 - 100)/\sqrt{144} = 0.67$, $z_2 = (112 - 100)/\sqrt{163} = 0.94$, and $z_3 = (132 - 100)/\sqrt{1422} = 0.85$. Since $z_2 > z_1, z_3$, then the optimal solution corresponds to $(m, v) = (112, 163)$ with probability $1 - \Phi(-0.94) = 0.83$ of attaining the target.

In all instances we created, the stochastic solution differed from the solution obtained by solving the deterministic OP. Note that there were alternative solutions that are optimal under different target profit levels. As stated before, the solution to the OP was found at the first iteration of the algorithm when the subproblem for $\alpha = 1$ was solved. Thus, the (m, v) point for the last target profit interval for each instance was also the optimal solution to the corresponding OP. Also, there may be multiple optimal solutions with the same expected profit level. By solving the OPSP, we obtained other solutions that are optimal for lower levels of the target profit. For instances 1A, 1B and 1C, the solution obtained by solving the OP was optimal for a single value of the target profit level, $K = \overline{m}$ since there were multiple solutions with $m = \overline{m}$ and the deterministic OP failed to find the one with the minimum variance.

The solutions for instances of 1A with $T = 200$ and instances of 1F with $T = 200$ are illustrated in Fig. 4. The solutions for cases 1B, 1C, 1D and 1E can be found in Ilhan *et al.* (2006). The dashed-line and solid-line tours are the optimal solutions under different target levels for the OPSP. The solid-line tour is also the optimal solution of the OP. We note that the OP gave the same tour for cases 1B and 1C as in 1A (and similarly for 1D and 1E as in 1F) since the mean profit values were the same in these three cases. In Fig. 5, we illustrate the objective values of the OPSP for instances of 1A and 1F with target values of $K > 40$ and $K > 80$, respectively, for which the probabilities obtained are visibly less than one.

Our first observation regarding the VSS is that when the mean profits of the nodes are close to each other, the VSS may be relatively high as in 1A, 1B and 1C with $T = 200$, where the maximum value of the VSS was 0.18 and the average VSS over $40 < K < 110$ was 0.05. In cases 1D, 1E and 1F, the maximum VSS was 0.06 and the average VSS over $80 < K < 180$ was 0.01. This results from the fact that in cases 1A, 1B and 1C, there may be many feasible solutions with similar mean profit values but significantly different variances. Second, as shown in Fig. 4, the tours of the stochastic solutions may significantly differ from the tours of the deterministic solutions. However, if the solution of the deterministic OP includes the nodes with low variance by chance, most nodes may be shared by the deterministic and stochastic solutions and the VSS may not be large.

The difference between the alternative tours is also reflected in the corresponding objective function values, except in problem 1F with $T = 200$. In this instance, although the alternative solutions differ from the tour obtained by the OP, the decrease in the total variance is not enough to provide a substantial advantage over the solution of the OP. Thus, even if the stochastic solution is very different from the tour obtained by the OP, the VSS may not be very large.

We also observed that the VSS converges to zero as the target profit, K, decreases or increases. First, as K approaches $\overline{m}$, the domain of the (m, v) values that have better objective function values than the OP solution becomes

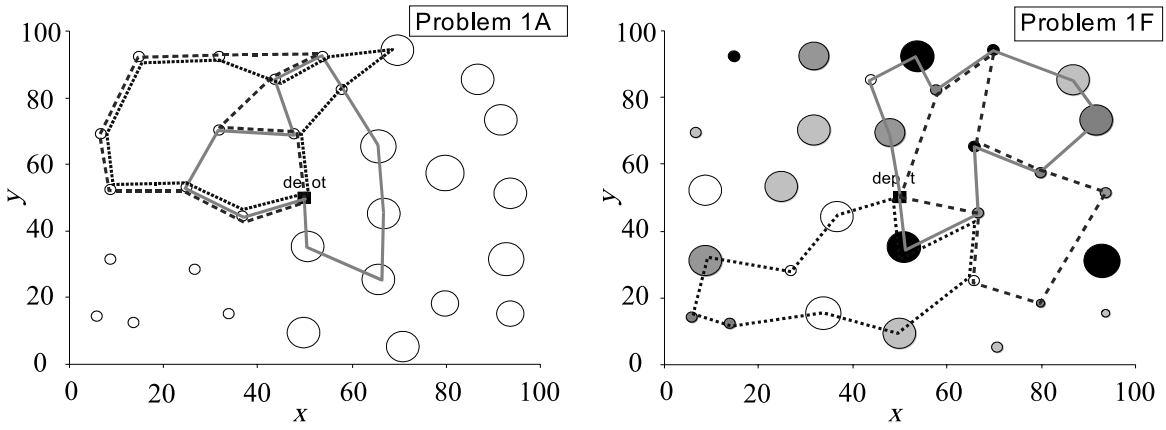

Fig. 4. Locations of the nodes and the solutions of the instances with $T = 200$ in cases 1A and 1F in Set 1. Within each graph, the darker the node, the higher the mean profit; the larger the circle, the larger the standard deviation relative to the other nodes. The solid-line tour is the optimum solution of both of the OPSP for a particular range of K and the OP for all K, the dashed-line tours are the optimal solutions of the OPSP for the other K values.

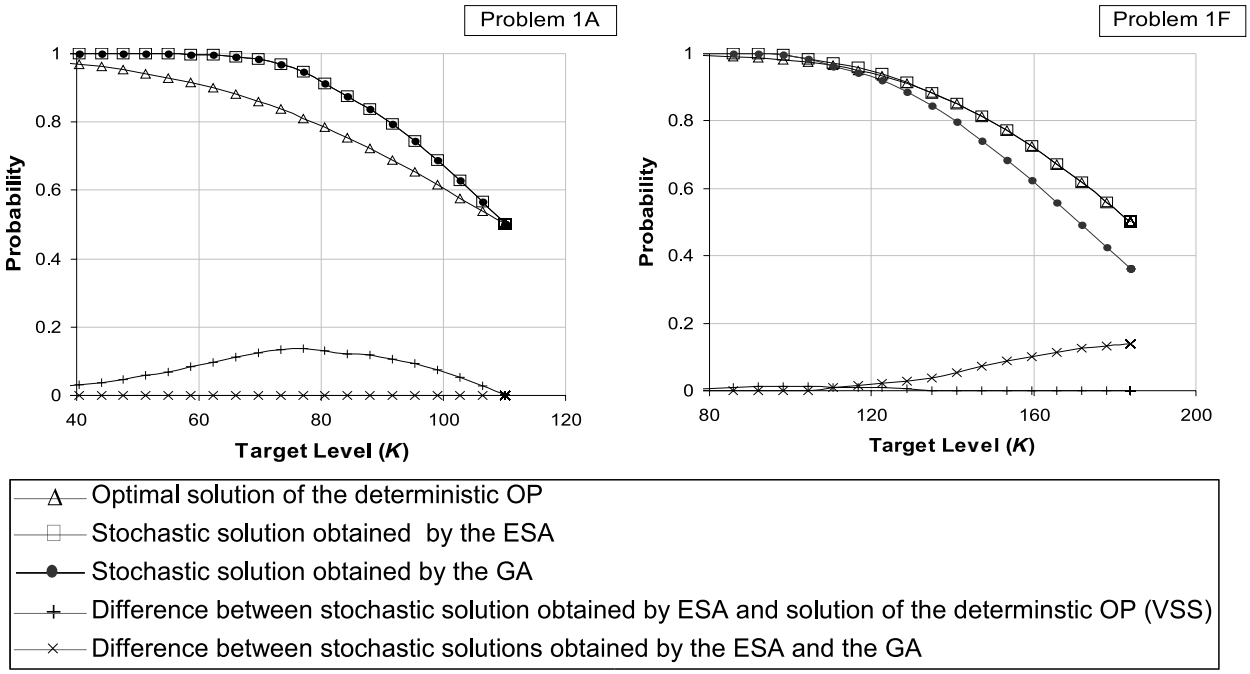

Fig. 5. Comparison of the probabilities of reaching the target, which are obtained by solving the instances with $T = 200$ in cases 1A and 1F in Set 1 by using the OPSP and the deterministic OP.

smaller and when K reaches $\overline{m}$, that domain is the line between $(\overline{m}, 0)$ and $(\overline{m}, \overline{v})$. As a result, the number of feasible solutions that may have a better objective function value decreases accordingly. Second, as K approaches zero, the objective function values for all solutions get closer to one and the difference between any two solutions starts to disappear.

6.2. *Comparing the performance of the ESA and the GA*

We compared the runtime of the ESA and the GA and the quality of solutions for Set 1. Among the instances in Set 1, the maximum number of points on CEFs was 22 for the case 1D with $T = 350$. Thus, the ESA had to solve at most 41 subproblems. Both the number of points on the CEF and the difficulty of the subproblems solved to obtain those points are important factors in the runtime. The average runtime of the ESA for all problem instances was 375 seconds with a minimum 12 seconds and a maximum 2890 seconds. On the other hand, the GA's minimum, average and maximum runtimes were 5, 16 and 40 seconds, respectively (see Table 3 for the runtime of each instance in Set 1).

To further analyze the effect of the number of nodes on the runtime of the ESA, we ran the algorithm on Set 2, which includes 15-node to 32-node problem sizes. For each problem size, we generated five different instances. In total, Set 2 consisted of 90 instances. The x and y coordinates of each node were between zero and 100. The mean profits and variances of the nodes were selected using a discrete Uniform distribution. The range for the mean profits was [10, 20]. The range for the variances of half of the nodes was [16, 25] and the range for the variances of the other half was [225, 400]. The locations of the high and low-variance nodes were decided randomly. For all instances, we set the travel time limit to 25% of the travel time required to visit all of

Table 3. CPU times of the ESA and the proposed GA in seconds to have the entire convex efficient frontiers (i.e., solutions for all K values) for problems in Set 1

	1A		*1B*		1C		*1D*		*1E*		*1F*	
T	*ESA*	*GA*	*ESA*	*GA*	ESA	GA	*ESA*	*GA*	*ESA*	*GA*	*ESA*	*GA*
150	12	5	96	7	51	5	94	9	15	9	85	7
200	212	10	63	8	273	11	153	27	55	6	156	16
250	38	22	1994	13	267	16	102	20	2890	9	264	9
300	36	26	362	22	571	11	153	23	913	40	306	18
350	52	14	42	19	410	28	329	17	464	21	754	25

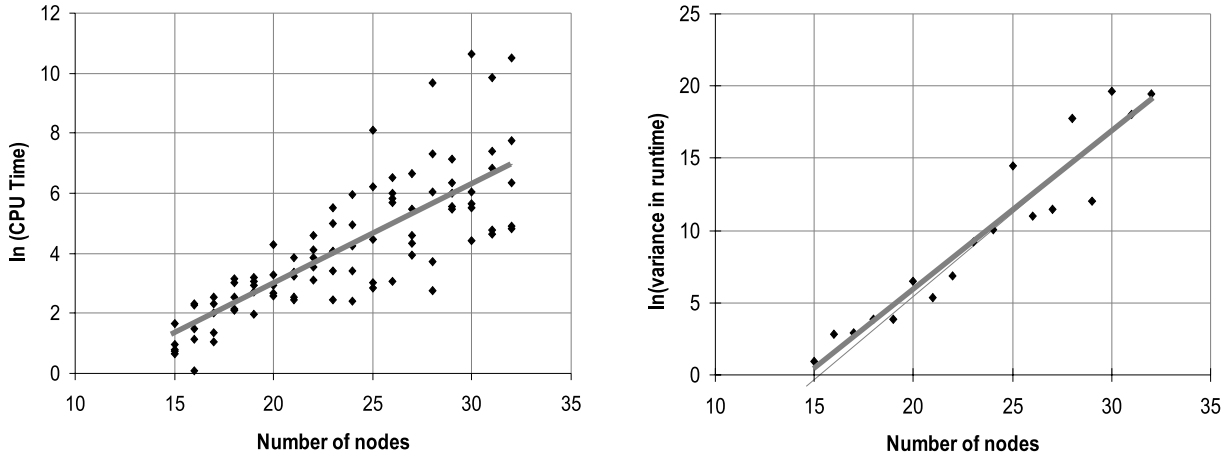

Fig. 6. The runtime of the ESA and its variance on logarithmic scale versus the number of nodes. The gray lines are the fitted lines obtained by the regression model $\ln X = a + b \times |N|$. For the line on the left side $a = -3.65$ and $b = 0.33$ with $R^2 = 0.64$; for the line on the right side $a = -16.09$ and $b = 1.10$ with $R^2 = 0.91$. All coefficient estimates have $p < 0.0005$.

the nodes. The runtime of the algorithm is plotted on a logarithmic scale in Fig. 6. As expected, the runtime increased exponentially as the number of nodes increased. Moreover, the variance of the runtime also increased exponentially. For example, the runtimes of the 30-node instances varied between 80 and 40 000 seconds, while the range of the runtime for problem instances with fewer nodes was much smaller.

In many instances in Set 1, the GA successfully found the optimal solutions obtained by the ESA. This can be seen if we compare the (m, v) values obtained by the ESA and the GA in Table 2. However, for some cases the GA failed to find the efficient point with the highest mean profit value (e.g., the case 1F with $T = 200$) and this resulted in deviations from the optimum as K moved closer to the maximum expected profit (see 1F in Fig. 5). The average difference between the objective function values of the ESA and the GA was 0.03, which was calculated by averaging the difference for cases 1A, 1B and 1C over $40 < K < 110$ and for cases 1D, 1E and 1F over $80 < K < 180$ (for these ranges of K, the objective function values of the ESA and the GA were less than 0.99). We also observed that in 75% of the cases for which we calculated averages, the difference between the objective function values of the ESA and the GA was less than 0.02. The maximum difference was 0.14, which occurred when the target profit level, K, was very close to the maximum expected profit, $\overline{m}$, in case 1C. Thus, we can conclude that the GA performs well compared to the ESA as long as K is not very close to $\overline{m}$. When K is very close to $\overline{m}$, using any bi-objective method is not sensible since the domain for the mean profit values of the possible optimal points, $(K, \overline{m})$, becomes too small for such an algorithm (see Fig. 1). Moreover, note that if $K = \overline{m}$, there is no need to develop a probabilistic model at all since the solution obtained by solving the deterministic OP, which has the mean profit value $\overline{m}$, gives a probability of 0.5 of reaching K, which is the highest probability value one can obtain in this case.

6.3. *Experiments with the GA for large instances*

We also ran the GA on Set 3, which consisted of instances with 101 nodes including the depot. Again, all the instances had the same coordinates on the plane. The mean and variances of the nodes were generated in the same way as was done for Set 1. The nodes in the first three cases of Set 3 (3A, 3B and 3C) had the same mean profits but different standard deviations, and the nodes in the other three cases had different mean profits and standard deviations (see the illustration in Fig. 7). The travel time limits were $T = 200$, 300, 400, 500 and 600. In total, Set 3 contained 30 problem instances.

All CEFs found by the GA follow similar patterns to the CEFs illustrated in Fig. 8 for cases 3A and 3F. The point with the highest total mean profit on each CEF can be considered as the approximate solution to the deterministic OP whose objective is to maximize the total mean profit. If we analyze the solutions of the deterministic OP on all CEFs, they have much higher variances than the other solutions on each CEF. They mostly fall into the trap of collecting profits without considering variances. However, careful selection of the nodes decreases the variance of the total profit while sacrificing little from the mean profit level, which results in a higher chance of collecting more than the target profit level K. As a result, in most of the CEFs generated in Fig. 8, a steep decrease in the variance is followed by a very slow decrease thereafter. One of the reasons is that the data set was constructed to have two variance levels, and there were some tours with very low variances that dominate the tours with higher variances.

Another interesting observation is that the travel time constraint is not binding for solutions on the lower portion

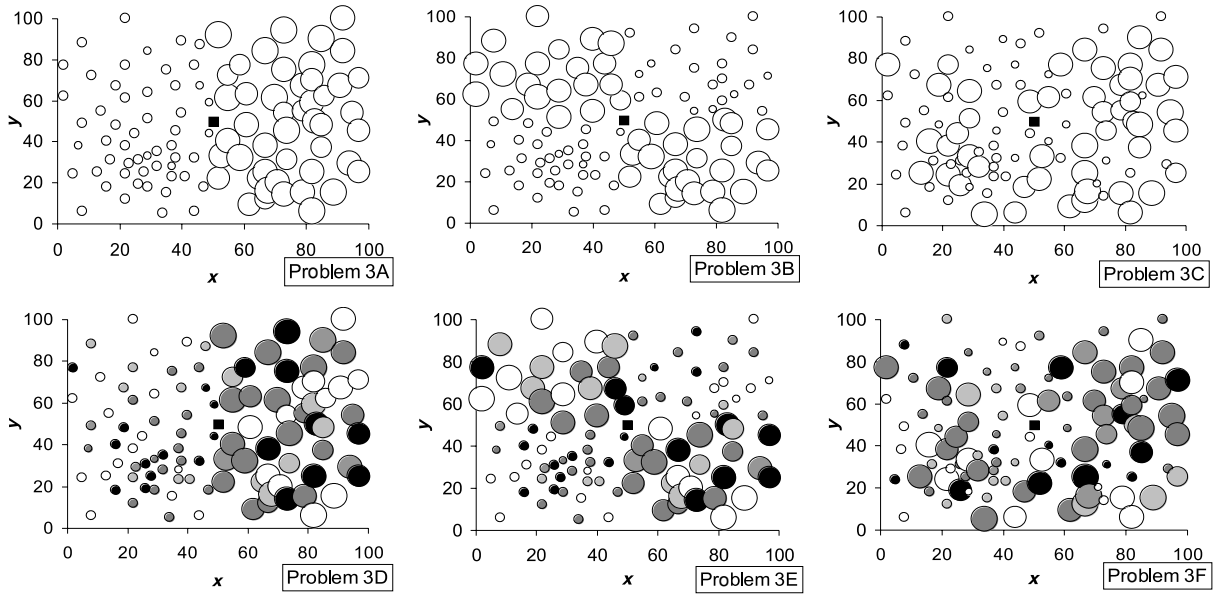

Fig. 7. Locations of the nodes in Set 3. Within each graph, the darker color means the higher mean profit and the larger circle means the larger standard deviation in relative to the others.

of the CEFs. Although it is still possible to visit more nodes without violating the travel time constraint, those solutions choose not to visit any more nodes since doing so would decrease the probability of attaining the target profit value. This fact can be deduced from the CEFs graphed in Fig. 8; in each problem the CEF with a higher time limit converges to the CEF with a lower time limit. Moreover, some of these tours are subtours of other solutions on the CEF.

Although the value of K has no effect on the shape of the CEF, as K increases, the number of solutions on the CEF that may be optimal to the OPSP decreases. The reason is that the mean profit of an optimal solution should lie in the interval $[K, \overline{m}]$ and this interval gets smaller as K increases. Moreover, the tours that are optimal under different target values are usually the ones whose mean profits are closer to the mean profit obtained by the deterministic OP. The reason is that as the gap between the mean profits of two solutions increases, the one with the lower mean profit must have a much lower variance to be optimal.

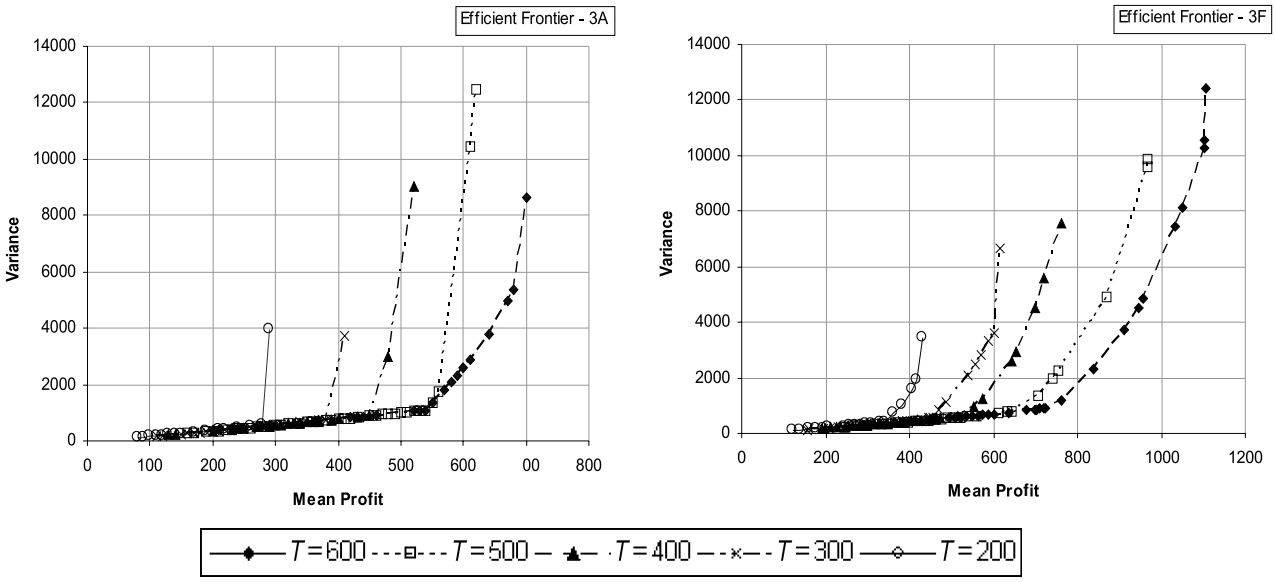

Fig. 8. The approximate EFs for different travel time limits, which are obtained by running the GA on cases 3A and 3F in Set 3.

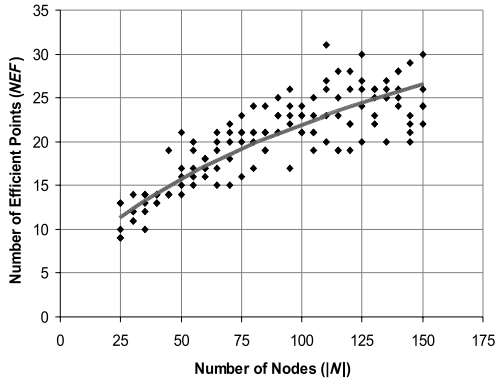

Fig. 9. The number of efficient points obtained by the GA versus the number of nodes in Set 4. The gray line is the fitted line obtain by the regression model $\ln(NEF) = a + b\ln(|N|)$. $a = 0.87$ and $b = 0.48$ with significances, $p_a < 0.0005$ and $p_b < 0.0005$, and $R^2 = 0.79$.

Finally, we applied the GA to Set 4 to analyze how the number of points on the CEF changes as the number of nodes in the OPSP increases. We generated 25-node to 150-node problems in five-node increments without any nested relationship between different problem sizes. The other problem parameters were set in a manner similar to that used for Set 2. Again, we generated five different instances for each problem size, resulting in 130 instances. The number of points on the CEF, named NEF, versus the number of the nodes in the problem, $|N|$, is plotted in Fig. 9. The result of the regression analysis shows a relation defined by equation $NEF = 2.387|N|^{0.48}$. This indicates that there may be a square-root relationship between the number of nodes in the problem and the number of subproblems to be solved to get the CEF. Unfortunately, we cannot be sure about the exact relationship without further theoretical analysis, which we leave to future research. However, it is highly possible that the main bottleneck in the ESA is solving the subproblems. Using more effective algorithms for the subproblems may substantially reduce the computation time needed by the ESA.

7. Conclusions

In this paper, we introduced the OPSP. We developed an exact solution scheme and a bi-objective GA. We showed that under some conditions there may be substantial value in solving the OPSP and in other cases solving its deterministic counterpart may be sufficient. Moreover, we revealed some structural properties of the OPSP solutions. When the VSS is high, the stochastic and deterministic solutions are quite different from each other; however, when the solutions are different from each other, the VSS is not necessarily high. Also, the number of points on the mean–variance EF seems to increase with the square root of the number of nodes in the problem.

Because of its generality, the parametric solution approach used here can be a promising technique to solve other similar routing problems in addition to the problem presented here. Moreover, as demonstrated in this study, bi-objective GAs can be successfully used as an alternative solution method for certain routing problems with non-linear objective functions.

An important future research direction is the OPSP with recourse. The problem investigated in this paper assumes that no recourse is allowed. However, after an auditor visits a city and collects the profits, he or she may want to modify the tour. For example, if the realized profit in the middle of the tour is too low compared to the target profit level, in a probabilistic sense, it may be more appropriate to take a risk and to visit nodes with high profit variance. Other future research directions are the multiple-period OPSP, in which new demand nodes appear periodically, and the multiple-salesman OPSP. In the multiple-period case, the number of new locations in each period may be deterministic or stochastic. Under these circumstances, completely different tour construction strategies may be required.

References

Bean, J.C. (1994) Genetic algorithms and random keys for sequencing and optimization. *ORSA Journal on Computing*, **6**(2), 154–160.

Bertsimas, D., Jaillet, P. and Odoni, A. (1990) A priori optimization. *Operations Research*, **38**(6), 1019–1033.

Bertsimas, D. and Simchi-Levi, D. (1996) A new generation of vehicle routing research: robust algorithms, addressing uncertainty. *Operations Research*, **44**(2), 286–304.

Birge, J.R. and Louveaux, F. (1997) *Introduction to Stochastic Programming*, Springer, New York, NY.

Carraway, R., Schmidt, R. and Weatherford, L. (1993) An algorithm for maximizing target achievement in the stochastic knapsack problem with normal returns. *Naval Research Logistics*, **40**, 161–173.

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

Coello, C.A.C. (2000) An updated survey of GA-based multiobjective optimization. *ACM Computing Surveys*, **32**, 109–143.

Cohon, J.L. (1978) *Multiobjective Programming and Planning*, Academic Press, New York, NY.

Deb, K. (2001) *Multi-Objective Optimization Using Evolutionary Algorithms*, Wiley, Chichester, UK.

Fischetti, M., Gonzales, J. and Toth, P. (1998) Solving the orienteering problem through branch-and-cut. *INFORMS Journal on Computing*, **10**(2), 133–148.

Fonseca, C.M. and Fleming, P.J. (1993) Genetic algorithms for multiobjective optimization: Formulation, discussion, and generalization, in *Proceeding of the Fifth International Conference on Genetic Algorithms*, 416–423.

Gendreau, M., Laporte, G. and Séguin, R. (1996) Stochastic vehicle routing. *European Journal of Operational Research*, **88**, 3–12.

Gendreau, M., Laporte, G. and Semet, F. (1998) A branch-and-cut algorithm for the undirected traveling salesman problem. *Networks*, **32**, 263–273.

Geoffrion, A.M. (1967) Solving bicriterion mathematical programs. *Operations Research*, **15**, 39–54.

Golden, B.L., Levy, L. and Vohra, R. (1987) The orienteering problem. *Naval Research Logistics*, **34**, 307–318.

Golden, B.L., Wang, Q. and Liu, L. (1988) A multifaceted heuristic for the orienteering problem. *Naval Research Logistics*, **35**, 359–366.

Henig, M. (1990) Risk criteria in a stochastic knapsack problem. *Operations Research*, **38**(5), 820–825.

Ilhan, T., Iravani, S.M.R. and Daskin, M. (2006) The orienteering problem with stochastic profits. Working paper, Department of Industrial Engineering and Management Sciences, Northwestern University. Evanston, IL.

Jaillet, P. (1988) A priori solution of a traveling salesman problem in which a random subset of the customers are visited. *Operations Research*, **36**(6), 929–936.

Kataoka, S., Yamada, T. and Morito, S. (1998) Minimum 1–subtree relaxation for the score orienteering problem. *European Journal of Operational Research*, **104**, 139–153.

Knowles, J. and Corne, D. (2005) Memetic algorithms for multiobjective optimization: issues, methods and prospects, in *Recent Advances in Memetic Algorithms*, Springer, Berlin, Volume 166 of Studies in Fuziness and Soft Computing, pp. 313–352.

Laporte, G. and Marello, S. (1990) The selective traveling salesman problem. *Discrete Applied Mathematics*, **26**, 193–207.

Leifer, A.C. and Rosenwein, M.B. (1994) Strong linear programming relaxations for the orienteering problem. *European Journal of Operational Research*, **73**, 517–523.

Millar, H.H. and Kiragu, M. (1997) A time based formulation and upper bounding scheme for the selective traveling salesperson problem. *Journal of the Operational Research Society*, **48**(5), 511–518.

Morton, D.P. and Wood, R. (1998) On a stochastic knapsack problem and generalizations, in *Advances in Computational and Stochastic Optimization, Logic Programming and Heuristic Search*, Woodruff, D. (ed.), Kluwer, Boston, MA, pp. 149–168.

Sniedovich, M. (1980) Preference order stochastic knapsack problems: methodological issues. *Journal of the Operational Research Society*, **31**, 1025–1032.

Steinberg, E. and Parks, M.S. (1979) A preference order dynamic program for a knapsack problem with stochastic rewards. *Journal of the Operational Research Society*, **30**, 141–147.

Stroh, L.K., Northcraft, G.B. and Neale M.A. (2002) *Organizational Behavior: A Management Challenge*. Lawrence Erlbaum Associates, Mahwah, NJ.

Tang, H. and Miller-Hooks, E. (2005) Algorithms for a stochastic selective travelling salesperson problem. *Journal of the Operational Research Society*, **56**(4), 439–452.

Tsiligirides, T. (1984) Heuristic methods applied to orienteering. *Journal of the Operational Research Society*, **35**(9), 797–809.

Veldhuizen, D.A.V. and Lamont, G.B. (2000) Multiobjective evolutionary algorithms: analyzing the state-of-the-art. *Evolutionary Computation*, **8**(2), 125–147.

Biographies

Taylan Ilhan received his Ph.D. from the Department of Industrial Engineering and Management Sciences of Northwestern University in 2007. He earned his B.S. and M.S. degrees in Industrial Engineering from Bilkent University, Ankara, Turkey and was awarded the Walter P. Murphy Fellowship by Northwestern University for graduate study in 2002. His research interests are stochastic modeling and optimization of production and logistics problems. He is also a member of INFORMS.

Seyed Iravani is an Associate Professor in the Department of Industrial Engineering and Management Sciences at Northwestern University. He received his Ph.D. from the University of Toronto, and worked as Postdoctoral Fellow in the Industrial and Operations Engineering Department at the University of Michigan. His research interests are in the applications of stochastic processes and queueing theory in production and service operations systems as well as in supply chains. He is associate editor for *IIE Transactions*, *Naval Research Logistics*, *Operations Research* and *Management Science*.

Mark S. Daskin is in the Department of Industrial Engineering and Management Sciences at Northwestern University where he holds a Walter P. Murphy Professorship. He is the author of over 50 refereed publications on various aspects of location modeling and supply chain management. He is also the author of a text entitled, *Network and Discrete Location: Models, Algorithms and Applications*. He is a fellow of both IIE and INFORMS and has served as the editor-in chief of both *Transportation Science* and *IIE Transactions*. He served as president of INFORMS in 2006.