A Subset-Selection Prize-Collecting TSP with Uncertain Speed

Dr Shane Dye, Dr John Giffin, Dr Nicola Petty
Department of Management
University of Canterbury
New Zealand
{shane.dye,john.giffin,nicola.petty }@canterbury.ac.nz

Abstract

Rogaine is a sport similar to orienteering but with the locations to visit and visit-order selected by each competitor during the event. Locations have an associated reward value and competitors have a time limit to collect the largest accumulated reward and return to the base location. Penalties apply if competitors return late. The competitor's problem can be modelled as a subset-selection prize-collecting TSP assuming deterministic travel times. In practice, travel times are estimated from the map provided at the start of the event and the route taken may be adjusted depending on conditions observed.

To better understand the effect of uncertain travel times, we model the problem as a two-stage stochastic program with uncertain speed. Speed is modelled as a finite number of constant-speed scenarios. In the first stage a single route is selected. In the second stage, the speed is revealed and the route may be shortened by heading directly back to base from any location on the route. The objective is to maximise expected accumulated reward.

We present an IP formulation and explore properties of optimal solutions. A branch and bound algorithm with knapsack based bounds is developed and initial computational results are provided.

Key words: Rogaine, TSP, prize-collecting, subset-selection, stochastic programming.

1 The Model and Rogaining

We examine a stochastic subset-selection prize-collecting TSP (travelling salesperson problem) model. For the deterministic version (DR), consider a set of locations $\{0, \ldots, n\}$ with given distances, d_{ij}, between locations, base location 0 and reward, $r_i > 0$, associated with each non-base location. Given time limit $W > 0$, linear time penalty per unit time, $c > 0$ and speed, $s > 0$, a solution is a route beginning at and returning to the base location, $A = \{a_0{=}0, a_1, \ldots, a_m, 0\}$, $m \le n$, visiting locations at most once. The objective is to maximise the net reward, that is, the accumulated reward from locations visited less any penalty accrued from returning after the time limit. More formally, the objective is:

$$\max \sum_{k=1}^{m} r_{a_k} - c \max\left\{0, \frac{1}{s}\left(\sum_{k=1}^{m} d_{a_{k-1},a_k} + d_{a_m,0}\right) - W\right\}$$

We assume the distances satisfy the triangle inequality: $d_{ij} \le d_{ik} + d_{kj}$ for all i, j, k, although this is non-restrictive since we may replace any distance matrix with the corresponding shortest path matrix. The data sign restrictions are used to eliminate uninteresting cases.

The stochastic version (SR) introduces a set of q speed scenarios, each described by a speed and corresponding probability, (s_t, p_t). A solution corresponds to a, first-stage, planned visit-order for all, or a subset, of the locations, $\{a_0=0, a_1, \ldots, a_m\}$, and, in the second stage, for each scenario an abandon-location, $a_{L(t)}$ where $0 \le L(t) \le m$, at which point in the visit-order the route heads directly back to the base location, $\{a_0=0, a_1, \ldots, a_{L(t)}, 0\}$, collecting reward only from those locations visited. The objective is to maximise expected net reward. The stochastic objective function can be written as:

$$\max \sum_{t=1}^{q} p_t \left(\sum_{k=1}^{L(t)} r_{a_k} - c \max \left\{ 0, \frac{1}{s_t} \left(\sum_{k=1}^{L(t)} d_{a_{k-1},a_k} + d_{a_{L(t)},0} \right) - W \right\} \right)$$

We study this model to gain insight into the effects of uncertain travel times in the sport of rogaining and to determine whether it is feasible to develop optimisation methods to study these solutions.

Rogaine is a sport similar to orienteering but with the locations to visit and visit-order selected by each team, of two or more, during the event. Teams are supplied with a map showing the area of the event and locations with reward values. They have a limited time to plan before being allowed to start-off. All members of a team must remain together throughout the event. Locations have an associated reward value and teams have a time limit to collect the largest accumulated reward and return to the base location. Linear time penalties apply if a team returns late.

During the planning phase, teams must make judgements about the travel times between locations. These judgements are used in planning the route. Computers and other electronic devices are not allowed in the planning phase or throughout the event, and our plan is to investigate and evaluate planning strategies. As a first step this model allows us to compare optimal strategies using deterministic and stochastic travel times for a very simple model of uncertainty. The speeds can represent average or relative speeds and the distances could be adjusted to account of terrain.

2 Literature Review and related problems

Gordon (2006) develops heuristics to use when developing routes during rogaining events. The heuristics are compared to the optimal route found through an enumeration algorithm. The model and data sets used are deterministic with predefined paths between control points and a constant speed throughout assumed. Testing with different speeds for up-hill and down-hill legs was reported.

Related problems include TSPs with profits, classified by Feillet *et al* (2005) into three groups based on how two metrics (distance and profit) are combined. The profitable tour problem includes both in the objective function, the orienteering problem maximises profit under a hard distance limit, and the prize-collecting TSP minimises distance under a hard minimum profit limit. These variations occur in the literature under various names.

Model (DR) is a relaxation of the orienteering problem since it penalises violations of the distance constraint. Laporte and Martello (1990) develop an enumeration algorithm for the orienteering problem using a knapsack-based relaxation to provide upper bounds. Ramesh et al (1992) use Lagrangain relaxation and an improvement procedure to develop a branch and bound algorithm for the same problem.

Stochastic versions have also been studied. Tang and Miller-Hooks (2005) study a two-stage profitable tour problem with stochastic travel times, service times and travel costs. A chance constraint is added to enforce a given minimum probability that the travel time is within a given limit. The full tour is determined *a priori*. Andreatta and Lulli (2008) look at a multi-stage TSP with the problem cast as a Markov decision process in which the objective function is to minimise long-term travel costs. Nodes with urgent demands need to be visited in the period the demand occurs with non-urgent demands able to be delayed by a period.

3 IP formulations

When the speed scenarios are ordered by decreasing speed, the abandon-locations, $a_{L(t)}$, are in a (non-strict) reverse order to the first-stage visit-order, *i.e.*, $L(t+1) \le L(t)$. This property can be used in IP formulations of the model.

Proposition 1: If the sequence of speeds $\{s_1, \ldots, s_q\}$ is decreasing, then there exist optimal abandon-location indices $\{L(1), \ldots, L(q)\}$ which are also decreasing.

Proof: It is sufficient to demonstrate that if $s_1 > s_2$ then $L(2) \le L(1)$ for a fixed visit-order. Define $Z(s,L)$ to be the net reward given speed s and abandon location a_L under the given visit-order. Assume the premise is false. This means that $L(2) > L(1)$, $Z(s_2,L(2)) > Z(s_2,L(1))$, and $Z(s_1,L(1)) > Z(s_1,L(2))$. For the last case, speed s_1, the accrued penalty using abandon-location $L(2)$ must be more than the accumulated reward from the additionally visited locations. In particular, it must be positive. From this it follows that $Z(s_1,L(2)) > Z(s_2,L(2))$ since $s_2 < s_1$ and more penalty will accrue under speed s_2.Then $Z(s_1,L(1)) > Z(s_1,L(2)) > Z(s_2,L(2)) > Z(s_2,L(1))$ and a penalty must accrue with speed s_2 and abandon-location $a_{L(1)}$ since otherwise $Z(s_1,L(1)) = Z(s_2,L(1))$. Put:

$$R_1 = \sum_{k=1}^{L(1)} r_{a_k}, \qquad R_2 = \sum_{k=L(1)+1}^{L(2)} r_{a_k},$$

$$D_1 = \sum_{k=1}^{L(1)} d_{a_{k-1},a_k} + d_{a_{L(1)},0}, \quad D_2 = \sum_{k=L(1)+1}^{L(2)} d_{a_{k-1},a_k} + d_{a_{L(2)},0} - d_{a_{L(1)},0}.$$

It follows that $R_2 > 0$, $D_2 > 0$ and:

$$\begin{aligned} Z(s_1,L(1)) - Z(s_1,L(2)) &= R_1 - c\max\left\{0, \frac{1}{s_1}D_1 - W\right\} - (R_1 + R_2) + c\left(\frac{1}{s_1}\left(D_1 + D_2\right) - W\right) \\ &= \frac{cD_2}{s_1} - R_2 - c\max\left\{W - \frac{1}{s_1}D_1, 0\right\} \\ &\le \frac{cD_2}{s_1} - R_2 \end{aligned}$$

As a consequence:

$$Z(s_2,L(1)) - Z(s_2,L(2)) = \frac{cD_2}{s_2} - R_2 > \frac{cD_2}{s_1} - R_2 \ge Z(s_1,L(1)) - Z(s_1,L(2)) \ge 0$$

This contradicts $Z(s_2,L(2)) > Z(s_2,L(1))$.□

This property is used extensively throughout this paper. For the remainder of the paper we assume $s_1 > s_2 > \cdots > s_q$.

Model SR can be formulated as an integer program with the first-stage representing the visit-order of locations, and the second-stage determining the order of locations visited and the arcs taken in each speed scenario. The abandon-location ordering property means the first-stage can be subsumed into Scenario 1, the fastest speed scenario.

Decision Variables

x_{ikt} – Indicates whether location $i \in \{0, \ldots, n\}$ is the kth location visited, $k = 0, \ldots, n + 1$, in scenario $t \in \{1, \ldots, q\}$,.

y_{ijt} – Indicates whether the route travels directly from location i to j in scenario $t \in \{1, \ldots, q\}$.

v_t – Time limit violation for scenario $t \in \{1, \ldots, q\}$.

Formulation (SR-1)

$$\max \sum_{t=1}^{q} p_t \left(\sum_{i=1}^{n} \sum_{k=1}^{n} r_i x_{ikt} - c v_t \right) \qquad (1)$$

$$\sum_{k=1}^{n} x_{ik1} \le 1 \qquad i = 1, \ldots, n. \qquad (2)$$

$$\sum_{i=1}^{n} x_{ik1} \le 1 \qquad k = 1, \ldots, n \qquad (3)$$

$$x_{00t} = 1,\ x_{01t} = 0,\ x_{i0t} = 0 \qquad i = 1, \ldots, n, t = 1, \ldots, q. \qquad (4)$$

$$\sum_{k=2}^{n+1} x_{0kt} = 1 \qquad t = 1, \ldots, q. \qquad (5)$$

$$\sum_{i=0}^{n} x_{ikt} \le \sum_{i=1}^{n} x_{i,k-1,t} \qquad k = 2, \ldots, n + 1, t = 1, \ldots, q. \qquad (6)$$

$$x_{ik,t+1} \le x_{ikt} \qquad i \in \{1, \ldots, n\}, k = 1, \ldots, n, t = 1, \ldots, q - 1. \qquad (7)$$

$$y_{ijt} \ge x_{i,k-1,t} + x_{jkt} - 1 \qquad i \ne j \in \{0, \ldots, n\}, k = 1, \ldots, n + 1, t = 1, \ldots, q. \qquad (8)$$

$$\sum_{\substack{j=0 \\ j \ne i}}^{n} (y_{ijt} + y_{jit}) \le 2 \qquad i \in \{0, \ldots, n\}, t = 1, \ldots, q. \qquad (9)$$

$$\sum_{i=0}^{n} \sum_{\substack{j=0 \\ j \ne i}}^{n} d_{ij} y_{ijt} - s_t v_t \le s_t W \qquad t = 1, \ldots, q. \qquad (10)$$

$$x_{ikt} \in \{0,1\}, y_{ijt} \in \{0,1\}, v_t \ge 0 \qquad i \ne j \in \{0, \ldots, n\}, k = 0, \ldots, n + 1, t = 1, \ldots, q.$$

The objective (1) calculates the expected net reward over all speed scenarios. Constraints (2)–(6) ensure loops are correctly defined, in sequential order starting from the base location, 0, with trivial loops excluded. Constraint (7) ensures speed ordering of scenarios according to Proposition 1. Constraint (8) forces arcs to be correctly indicated between consecutive locations and Constraint (10) determines the time limit penalty. The formulation allows irrelevant arcs to be indicated if they do not violate the time limit when this is slack. Constraint (9) avoids these irrelevant arcs being connected to the optimal loop—strictly it is not necessary.

Formulation SR-1 can be strengthened by adding additional scenario ordering constraints based on the abandon-location ordering property.

$$y_{ij,t+1} \le y_{ijt} \qquad i \ne j \in \{1, \ldots, n\}, t = 1, \ldots, q - 1.$$

An alternative model can be formulated using only the arc and node indicator variables and subtour elimination constraints. Due to the implicit ordering of routes by speed, the subtour elimination constraints are only needed for the Scenario 1.

Decision Variables

u_{0j} – Indicates whether all routes travel directly from location 0 to j as the first arc, $j \in \{1, \ldots, n\}$.

y_{ijt} – Indicates whether the route travels directly from location i to j, $0 \le i < j \le n$, in scenario $t \in \{1, \ldots, q\}$.

z_{it} – Indicates whether location $i \in \{1, \ldots, n\}$ is visited in scenario $t \in \{1, \ldots, q\}$.

v_t – Time limit violation for scenario $t \in \{1, \ldots, q\}$.

Formulation (SR-2)

$$\max \sum_{t=1}^{q} p_t \left(\sum_{i=1}^{n} r_i z_{it} - c v_t \right) \tag{11}$$

$$\sum_{j=1}^{n} u_{0j} = 1 \tag{12}$$

$$\sum_{i=1}^{n} y_{0it} = 1 \qquad t = 1, \ldots, q. \tag{13}$$

$$u_{0k} + \sum_{i=0}^{k-1} y_{ikt} + \sum_{i=k+1}^{n} y_{ikt} = 2 z_{kt} \qquad k = 1, \ldots, n,\ t = 1, \ldots, q. \tag{14}$$

$$\sum_{j=1}^{n} d_{0j} u_{0j} + \sum_{i=0}^{n-1} \sum_{j=i+1}^{n} d_{ij} y_{ijt} - s_t v_t \le s_t W \qquad t = 1, \ldots, q. \tag{15}$$

$$y_{ij,t+1} \le y_{ijt} \qquad 1 \le i < j \le n,\ t = 1, \ldots, q-1. \tag{16}$$

$$z_{i,t+1} \le z_{it} \qquad i = 1, \ldots, n,\ t = 1, \ldots, q-1. \tag{17}$$

$$2\sum_{k \in S} z_{k1} \le |S| \left(\sum_{\substack{i \in S,\, j \notin S \\ i<j}} y_{ij1} + \sum_{\substack{i \notin S,\, j \in S \\ i<j}} y_{ij1} + \sum_{j \in S} u_{0j} \right) \qquad S \subset \{1, \ldots, n\},\ |S| \ge 3. \tag{18}$$

$$u_{0j} \in \{0,1\}, y_{ijt} \in \{0,1\}, z_{jt} \in \{0,1\}, v_t \ge 0 \qquad 0 \le i < j \le n,\ t = 1, \ldots, q.$$

The objective (11) calculates the expected net reward over all speed scenarios. Constraints (12) and (13) force exactly one leaving and returning arc from base location 0. Constraint (14) ensures the degree of location nodes is two on the loop and zero otherwise. Constraint (15) determines the time limit penalty. Constraints (16) and (17) ensure speed ordering of scenarios according to Proposition 1, for both locations and arcs not connecting to the base location, 0. Constraint (18) eliminates subtours in the fastest speed scenario (scenario 1). The scenario ordering constraints ensure no subtours for the other scenarios.

Optimising SR-2 could be done by relaxing the subtour elimination constraints, and then adding back violated constraints during a branch-and-cut type algorithm.

4 Enumerative algorithm

Notice that if the first-stage visit order is set, the second-stage optimal solutions can be found by a linear-time search for each scenario. This suggests a branch-and-bound or enumerative scheme branching on the visit-order rather than the arcs to use in each scenario. We follow a scheme based on that proposed by Laporte and Martello (1990) for a deterministic version of the problem. The algorithm framework consists of incrementally extending a simple path from location 0, representing a partial visit-order, using branch-and-bound. Each branch-and-bound node, h, records a partial visit-order $A(h) = \{a_0{=}0, a_1, \ldots, a_{m(h)}\}$. From $A(h)$ local upper and lower bounds are determined using solutions; global bounds are also kept. These bounds are used, as usual, to fathom nodes, maintain an incumbent solution and indicate termination of the algorithm. Descendent nodes are generated by branching over all possible choices for $a_{m(h)+1}$.

4.1 Properties and Bounds

Various upper and lower bounds are examined, many based on a partial or full visit-order A. In the following Z^{SR} refers to the optimal solution value of an instance SR. The lower bound fixes the visit-order in SR. It is tight if the visit-order is optimal.

Proposition 2: Given any instance of SR, and a visit-order $A = \{a_0{=}0, a_1, \ldots, a_m\}$, define

$$F(t) = \max_{0 \le \ell \le m} \sum_{k=1}^{\ell} r_{a_k} - \frac{c}{s_t} \max\left\{0, \sum_{k=1}^{\ell} d_{a_{k-1}a_k} + d_{a_\ell 0} - s_t W\right\} \tag{19}$$

Then $Z^{\mathrm{F}} = \sum_{t=1}^{q} p_t F(t) \le Z^{\mathrm{SR}}$. Furthermore, the bound is tight if A is the optimal visit-order for SR.

The following upper bounds follow knapsack bounds derived in Laporte and Martello (1990) for a deterministic version of the problem. They are based upon generalised knapsack problems with vertex weights

$$w_i = \alpha \min_{i \ne j}\left\{d_{ji}\right\} + (1-\alpha) \min_{i \ne j}\left\{d_{ij}\right\} \qquad i = 0, \ldots, n. \tag{20}$$

for some real value α ($0 \le \alpha \le 1$).

Proposition 3: Given any instance of SR, real value α ($0 \le \alpha \le 1$), and vertex weights given by (20), let Z^{KL} be the optimal solution value of the following problem, KL, consisting of q linked 0-1 knapsack problems with linear over-capacity penalties:

$$\max \sum_{t=1}^{q} p_t \left(\sum_{i=1}^{n} r_i z_{it} - c v_t \right) \tag{21}$$

$$\sum_{i=1}^{n} w_i z_{it} - s_t v_t \le s_t W - w_0 \qquad t = 1, \ldots, q \tag{22}$$

$$z_{i,t+1} \le z_{it} \qquad i = 1, \ldots, n,\ t = 1, \ldots, q-1. \tag{23}$$

$$z_{it} \in \{0,1\}, v_t \ge 0 \qquad i = 1, \ldots, n,\ t = 1, \ldots, q.$$

Then, $Z^{\mathrm{KL}} \ge Z^{\mathrm{SR}}$. Furthermore, let $Z^{\mathrm{K}}(t)$ be the optimal solution value of individual 0-1 knapsack problem (with linear over-capacity penalty):

$$\max \sum_{i=1}^{n} r_i z_{it} - c v_t \tag{24}$$

$$\sum_{i=1}^{n} w_i z_{it} - s_t v_t \le s_t W - w_0 \tag{25}$$

$$z_{it} \in \{0,1\}, v_t \ge 0 \qquad i = 1, \ldots, n.$$

Then:

$$\sum_{t=1}^{q} p_t Z^{\mathrm{K}}(t) \geq Z^{\mathrm{SR}}$$

Proof: It is sufficient to prove the first part of the theorem as the second part is a consequence of relaxing constraints (23). Let $A^* = \{a_0=0, a_1, \ldots, a_m\}$ be the optimal visit-order, $L(t)$, $t = 1, \ldots, q$, the optimal abandon-location indices, and v_t^* the optimal time-limit violations for SR. We show there is a feasible solution to KL with the same objective function value. Define $z_{a_i t} = 1$ for $i = 1, \ldots, L(t)$, $t = 1, \ldots, q$, and zero otherwise. From (10), for $t = 1, \ldots, q$, we must have

$$\sum_{k=1}^{L(t)} d_{a_{k-1}a_k} - s_t v_t^* \leq s_t W,$$

so,

$$(1-\alpha)d_{0a_1} + \sum_{k=1}^{L(t)-1} \left(\alpha d_{a_{k-1}a_k} + (1-\alpha)d_{a_k a_{k+1}} \right) + \alpha d_{a_{L(t)}0} - s_t v_t^* \leq s_t W.$$

From (22) $\alpha d_{a_{k-1}a_k} + (1-\alpha)d_{a_k a_{k+1}} \geq w_{a_k}$ and the conclusion follows.□

Proposition 4: Given any instance of SR with an initial fixed partial vertex-order $A(h) = \{a_0=0, a_1, \ldots, a_{m(h)}\}$, real value α ($0 \leq \alpha \leq 1$), and vertex weights given by (20), let $F(t)$ be given by (19) using $A(h)$ as the visit-order and $Z^{\mathrm{KF}}(A(h), t)$ be the optimal solution value of problem (24)–(25) with items in $A(h)$ fixed in the knapsack. Then:

$$\sum_{t=1}^{q} p_t \max\{F(t), Z^{\mathrm{KF}}(A(h),t)\} \geq Z^{\mathrm{SR}}.$$

Proof: Let $A^* = \{a_0=0, \ldots, a_{m(h)}, \ldots, a_m\}$ be the optimal visit-order, $L(t)$, $t = 1, \ldots, q$, the optimal abandon-location indices, and v_t^* the optimal time-limit violations for SR. If $L(t) \leq m(h)$ then $F(t)$ equals the optimal solution value contribution from scenario t, otherwise, following the proof of Proposition 3, $Z^{\mathrm{KF}}(t)$provides an upper bound on that value. The conclusion follows.□

5 Computational results

Formulation SR-1 was solved in CPLEX 10, but was only able to optimise problems with fewer than 5 locations in a reasonable time. Solution using formulation SR-2 was not tried as experience from Laporte and Martello (1990) suggests the enumeration algorithm is more likely to perform better.

The enumeration algorithm was coded in Visual Basic and tested on randomly generated instances. The programming environment use allowed for a fast development, but not necessarily the most efficient code. The instances used 8, 12, 16, 24 or 30 reward locations, 1, 2, 4, or 8 speed scenarios and three different time limits which were dependent on the instance. The time limits were set proportional to an estimate of the optimal TSP time at the average speed over all scenarios. The TSP time was estimated by summing, for each location, the average time to the 5 nearest locations. The time limits were set to be close to 0.1, 0.2 or 0.4 of this value. One would expect tour to reach 10%, 20% and 40%, respectively, of the locations in optimal Rogaine routes, for each of these time limits.

Locations were generated randomly over the unit square, with the base location constrained to be in a 0.4 by 0.4 square centred in the middle. Instances with locations closer than 0.03 were not used. Only one location-layout was generated for each number of rewards.

The speed probability distributions were symmetric with mean speed 1, and a higher probability of being closer to the mean.

Instances were run for a maximum of 10 minutes. Timing results are shown in Table 1.

Time limit proportion	Scena -rios	Locations 8	12	16	24	30
0.1	**1**	0	0	0.02	0.39	0.70
0.1	**2**	0	0	0.02	0.24	0.48
0.1	**4**	0	0	0.02	0.39	0.80
0.1	**8**	0	0	0.02	0.59	1.3
0.2	**1**	0	0.05	0.63	28	>600
0.2	**2**	0	0.11	0.47	31	>600
0.2	**4**	0.02	0.05	0.55	65	>600
0.2	**8**	0	0.05	0.89	>600	>600
0.4	**1**	0.03	1.8	>600	>600	>600
0.4	**2**	0.02	2.2	>600	>600	>600
0.4	**4**	0.02	2.3	>600	>600	>600
0.4	**8**	0.03	3.2	>600	>600	>600

Table 1: Solution times (in seconds) for 80 instances.

In Table 1 times are shown in seconds. Those showing 0 are less than 0.01 seconds, while those showing >600 were stopped after 10 minutes. Interestingly, for many of the instances, two scenario instances solved faster than the corresponding deterministic instances. The times only appear to increase linearly with the number of scenarios, although the number of test instances is too small to see this clearly. As expected solution times increased exponentially as the number of locations and the time limit increase.

Different branch and bound node selection polices were tried. Depth-first and breadth-first policies out performed policies which searched through bundles of nodes to find the largest or smallest upper bound. Depth-first search was marginally faster.

Examining the solution shows that often similar solutions appear for the deterministic and stochastic but the order of locations becomes very important. Further analysis is needed to determine how well the various solutions perform under out-of-sample scenarios.

6 Conclusions and Future Work

A stochastic version of the prize-collecting, subset-selection TSP was studied to give some insights into solutions and the feasibility of an optimisation algorithm. It does appear that the algorithm is viable, although one could expect the exponential nature of the algorithm to provide a limit on the size or relative time limits which can be feasibly solved by this method. The algorithm could be improved by the use of stronger lower and upper bounds. The stochastic nature of the problem did not appear to have as much effect on solution times as the number of locations or the relative time limit.

Future work will involve refining the algorithm, strengthening bounds and improving the time to calculate these. Alternative bounds also need to be tested. The test problems used do not allow control to find for which features of the problem the algorithm performs best and worst, and some form of instance-generation approach is needed to more fully test the algorithm.

7 References

Andreatta, G., and G. Lulli. 2008. "A multi-period TSP with stochastic regular and urgent demands." *European Journal of Operational Research* **185**: 122–132.

Feillet, Dominique, Pierre Dejax and Michel Gendreau. 2005. "Travelling salesman problems with profits." *Transportation Science* **39** 2 (May): 188–205.

Gordon, Samuel P. 2006 "Rogaining: a prize-collecting orienteering problem." *Proceedings of the 41st Annual Conference of the Operational Research Society of New Zealand: ORSNZ'06*: 173–182.

Laporte, Gilbert, and Silvano Martello. 1990. "The selective travelling salesman problem." *Discrete Applied Mathematics* **26**: 193–207.

Ramesh, R., Y.S. Yoon and M.H. Karwen. 1992. "An optimal algorithm for the orienteering tour problem." *INFORMS Journal on Computing* **4** (2): 155–165.

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