ERASMUS UNIVERSITEIT ROTTERDAM

The Orienteering Problem with Stochastic Weights

A Two- Stage Approach

Gökçe Işlak

1 July 2013

Bachelor thesis Econometrics and Operations Research

Thesis supervisor: T. Dollevoet

The Orienteering Problem with Stochastic Weights

ABSTRACT

In the Orienteering Problem (OP), a set of nodes is given, each with a profit. The goal is to determine a tour, limited in length, that visits a subset of nodes and maximizes the sum of the collected profits. In this paper, the Orienteering Problem with Stochastic Weights (OPSW) is used to reflect uncertainty in real-life applications. This problem is approached by the Two-Stage Orienteering Problem (TSOP). Hereby, the problem is formulated as a two-stage stochastic model with recourse for the OPSW where the capacity constraint is hard. An existing heuristic for this TSOP is described and tested on 29 problem instances. The computational results are presented in detail.

Keywords: {orienteering problem}, {uncertainty}, {stochastic weights}, {TSOP heuristic}

1.INTRODUCTION

The Orienteering Problem is a routing problem which possesses characteristics of both the Traveling Salesman Problem (TSP) and the Knapsack Problem (KP). The TSP asks the following question: Given a list of cities and the distances between each pair of cities, what is the shortest possible tour that visits each city exactly once and returns to the origin city? The KP is a problem in combinatorial optimization and it asks the following question: Given a set of items, with a weight and a value, determine the number of each item to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible.

The OP considers a profit associated to each node and a weight associated to each arc. The profit of a node is collected only if that node is visited, what means that not every node has to be visited (in contrast to the TSP). The aim is to find a feasible tour that maximizes the profit, starting and ending at the depot. A tour is only feasible if it satisfies a capacity constraint on the total weight of the arcs selected in that tour. The fact that we select nodes in such a way that the weight associated to this selection does not exceed a predefined capacity, relates to the KP. Contrary to the KP, the ordering of the nodes in the selection influences the associates total weight.

There are many applications where the OP is relevant. One of these applications is the tourist tour planning problem (W. Souffriau (2008)). In this problem, the tourist wants to visit several different sightseeing locations. But for each of these locations the tourist has a different preference level and the length of the tourist tour is of course restricted by the total time the tourist can spend on sightseeing. Another application is a military application. This application of the OP considers Unmanned Aerial Vehicle (UAV) mission planning to collect intelligence information about different locations in the area of operations. The aim of these missions is to acquire as much information as possible during the flight, while the length of the flight is limited by the available fuel capacity of the UAV (Evers et al. (2012)).

In most urban environments, the travel times can vary greatly. Thus, it is in general impossible to know with certainty which of the used nodes can be visited without exceeding the predefined capacity. This uncertainty can be caused by several factors like weather circumstances, heavy-traffic and other unforeseen events. Many choices could be made to solve this problem. The average weight could be chosen and used to construct a tour and this tour will be followed as long as possible. Or the maximum weight could be chosen as starting-point to calculate the best tour. But the disadvantage of taking the maximum weight is that it would give you a worst case scenario tour, which is not very likely and thus suboptimal since all other scenarios would give you a better result. However, this paper considers a new variant of the OP where the weights are stochastic, only the distribution of the uncertain weights is known beforehand. This problem can be classified as the Orienteering Problem with Stochastic Weights (OPSW). I assume that the distribution of the weights follows a gamma distribution with fixed scale parameters and that the capacity constraint on the total weight is hard. That means that the depot has to be reached before the total realized weight exceeds the capacity and that no profit can be obtained for the unvisited nodes in the planned tour. The weights in this paper represent the travel times and the profits model the importance of the location.

Evers et al. (2013) have introduced a new model to tackle the OPSW where the capacity constraint is hard, which they call the Two-Stage Orienteering Problem (TSOP). The TSOP considers uncertainty in the weights of the arcs of the OP and the effect thereof on the profit value to be obtained. The TSOP is a two-stage recourse model in which the first-stage decision is to construct a tour, which may have to be aborted in the second stage before reaching the final target because of the weight realization. They have used two approaches for solving the TSOP: Sample Average Approximation (SAA), a well-known technique from stochastic programming, and a heuristic approach. Whereupon they have shown that the SAA is time-consuming and that the TSOP heuristic provides good quality solutions with much less computational effort. So in this paper the TSOP heuristic is used to construct toures.

This paper is structured as follows. In Section 2, a mathematical formulation of the TSOP is given. Section 3 describes the heuristic that is used to solve the TSOP. The information about the data and the implementation details are given in Section 4 and the computational results of the heuristic are present in detail in Section 5. Finally, concluding remarks are given in Section 6.

2.MATHEMATICAL FORMULATION

In order to give a better picture of the TSOP, in this section I will first give a mathematical formulation of the OP and then the mathematical formulation of the TSOP which is introduced in the paper of Evers et al. (2013).

2.1 The deterministic orienteering problem

Consider a complete graph $G = (N^+, A)$ with $|N| + 1$ nodes, where $N = \{1, \dots, n\}$ the set of all targets. Denote the depot location by node $0 \notin N$, thus we get the set $N^+ = N \cup \{0\}$. To each node $i \in N$ we associate a profit value p_i. To each arc $(i, j) \in A$ we associate a value f_{ij} representing the weight of arc (i, j). Further we know that the total capacity of the arcs that can be selected in any tour is C.

We have two different decision variables. One of these decision variables is the binary variable x_{ij}:

$$x_{ij} = \begin{cases} 1 & if\ arc\ (i,j)\ is\ used\ for\ the\ tour \\ 0 & otherwise \end{cases}$$

And the other decision variable is the auxiliary variable $u_i \in \{1, \dots, n\}$, this variable denotes the position of node i in the tour.

The aim is to find a tour that maximizes the total profit, that is feasible with respect to the capacity constraint and that starts and ends at the depot.

The mathematical formulation of the deterministic OP is as follows:

$$\max \sum_{i \in N} p_i \sum_{j \in N^+ \setminus \{i\}} x_{ij} \tag{1}$$

subject to

$$\sum_{i \in N} x_{0i} = \sum_{i \in N} x_{i0} = 1 \tag{2}$$

$$\sum_{i \in N^+ \setminus \{k\}} x_{ik} = \sum_{i \in N^+ \setminus \{k\}} x_{ki} \leq 1 \qquad \forall k \in N \tag{3}$$

$$\sum_{(i,j) \in A} f_{ij} x_{ij} \leq C \tag{4}$$

$$u_i - u_j + 1 \leq \left(1 - x_{ij}\right)|N| \qquad \forall i, j \in N \tag{5}$$

$$1 \leq u_i \leq |N| \qquad \forall i \in N \tag{6}$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \tag{7}$$

Constraint 1: The objective function to maximize the total collected profit.

Constraint 2: The constraint guarantees that the tour starts at the depot and ends at the depot. By setting the summation over all nodes to one, you guarantee that there is exactly one node that comes before and one that comes after the depot.

Constraint 3: The constraint ensures the connectivity of the tour and guarantees that every node is visited no more than once. The summation ensures that a node is either visited once or not visited at all. After visiting a node you have to 'leave' the node and if you have not visited the node you cannot 'leave' it, thus the equality.

Constraint 4: The capacity constraint that guarantees that the total weight of the selected arcs does not exceed the capacity.

Constraint 5 and 6: These constraints are necessary to prevent the construction of subtours. These are the well-known Miller-Tucker-Zemlin (1960) constraints. These constraints indeed exludes subtours:

1. The arcs constraint for (i, j) forces $u_j \geq u_i + 1$, when $x_{ij} = 1$.
2. If there are more than one subtour constructed, then at least one of these subtours does not contain the depot node. Along this subtour the u_i values would have to increase to infinity and constriant (6) will not permit this: Suppose that we have the subtour $1 \rightarrow 2 \rightarrow 3$, with $x_{12} = x_{23} = 1$. There is no way to return to node 1, because $u_1 < u_2 < u_3 < u_1$ is not a valid inequality. Thus there is only one tour that can be valid and that is the one with the depot as the starting and ending location. Here we have $x_{0a} = \ldots = x_{b0} = 1$, this is valid because the depot does not have to met the MTZ constraint $1 \leq u_i \leq |N|$.

2.2 The two-stage orienteering problem

Consider now a tour with an associated ordering of nodes. Because of the uncertainty in the weights, the probability that a node cannot be reached is higher if that node is scheduled further ahead in the tour. This means that, while following the tour, it can be necessary to return to the depot at some point based on the actual realization of the weights. As a result the profit of all nodes that is not visited due to the actual realization of the weights, will not be obtained and the total realized profit will be smaller than the total profit of the given tour. Using a two-stage recourse model, it is possible to model all possible moments of returning to the depot and the associated loss in profit. To obtain the best solution you must return to the depot at the moment that the remaining capacity equals the expected weight required to return from the current location to the depot.

The associated recourse cost is defined as the profit shortage. The profit shortage is the sum of the profit of the nodes selected in the first-stage tour that cannot be visited. One assumption is made for the TSOP: a certain amount of extra capacity is available to cover the maximum deviation from the expected weight on any of the arcs to the depot. This safety stock is not part of the capacity C, which is used in the model.

The aim now is to find a tour that maximizes the profit that is obtained from the first stage of the TSOP corrected by the expected second-stage profit shortage.

2.3 From orienteering problem to two-stage orienteering problem

For the TSOP we need the weight f_{ij} of arc (i, j), which is a random variable that follows a predefined probability distribution, in my case the gamma distribution. For notational convenience, I will also use

f_{ij} to denote the realization of these random variables. The vector that contains the weight realization of each arc is denoted by f. The distance associated to each arc is denote by d_{ij}.

Now we can define the expected value of the weight associated to each arc (i, j) as follows:

$$\overline{f_{ij}} = d_{ij} + gamma(shape_{ij}, scale).$$

Note that the travel time (weight) can be broken down into its deterministic (minimum time that is needed) and stochastic (represented by a two-parameter gamma distribution) components.

Based on these definitions, the mathematical formulation of the TSOP is the following:

$$\max \sum_{i \in N} p_i \sum_{j \in N^+ \setminus \{i\}} x_{ij} + \mathbb{E}_f\big(v(x, f)\big) \tag{8}$$

subject to

$$\sum_{i \in N} x_{0i} = \sum_{i \in N} x_{i0} = 1 \tag{2}$$

$$\sum_{i \in N^+ \setminus \{k\}} x_{ik} = \sum_{i \in N^+ \setminus \{k\}} x_{ki} \leq 1 \qquad \forall k \in N \tag{3}$$

$$u_i - u_j + 1 \leq \big(1 - x_{ij}\big)|N| \qquad \forall i, j \in N \tag{5}$$

$$1 \leq u_i \leq |N| \qquad \forall i \in N \tag{6}$$

$$x_{ij} \in \{0,1\} \qquad \forall (i, j) \in A \tag{7}$$

Constraint 8: The objective function to maximize the total expected profit.

$v(x, f)$ is a function that models the *profit shortage* for a tour x and weight realization f.

$\mathbb{E}_f\big(v(x, f)\big)$ is the expected *profit shortage* for a tour x with respect to the distribution of f.

Constraint 4 (OP formulation): The constraint is not necessary for the TSOP, since capacity limitations are already incorporated in the expected profit shortage in the objective function (8).

3.TWO-STAGE ORIENTEERING PROBLEM HEURISTIC

In this section I will describe the heuristic that I have used to solve the TSOP (Evers et al. (2013)). This heuristic makes use of a randomization concept and a score measure. This score measure incorporates the profit of a node as well as the profits of the nodes in its proximity.

The OP has in most of the cases a larger set of optimal solutions than the TSOP because of the uncertainty in the weights. Suppose that there are, with respect to the deterministic capacity constraint, two feasible tours that contain the same nodes but in a different order. In the OP these two tours will have the same objective function, in contrast with the TSOP: different orderings of nodes in a solution are likely to result in different expected profit values. A heuristic approach for the TSOP is required to adequately take into consideration the uncertainty and the associated different expected profits between different solutions.

In the TSOP heuristic, multiple iterations are performed. First, an initial solution is constructed which is then improved by local search moves. The best solution found within these iterations will be the final solution of the TSOP heuristic. To avoid that the final solution is a local minimum that is not a global minimum, a diversity of initial solutions is created and explored. The initial solutions are obtained using a constructive heuristic based on the problem structure, combined with a randomization concept. The pseudo code of the TSOP heuristic is given in *figure 1*.

```
bestSolution ← null
for node i ← 1 to numberOfNodes
    do  compute score measure s_i
for iteration ← 1 to numberOfIterations
    do  Construction phase:
        initialSolution ← null
        while TSOP(initialSolution) can be extended
            do  for each node i ∉ initialSolution
                    do compute ratio r_i(initialSolution, s_i)
                randomization:
                draw random number, select new node i with prob des_i(r_i)
                initialSolution ← initialSolution + new node
        Improvement phase:
        currentSolution ← initialSolution
        while TSOP(currentSolution) can be improved
            do  interchange(currentSolution)
                insert(currentSolution)
                remove-insert(currentSolution)
        Store best tour found so far:
        If TSOP(currentSolution) > TSOP(bestSolution)
            then bestSolution ← currentSolution
return (bestSolution)
```

Figure 1 Pseudo code of the TSOP heuristic

The first thing to do in de TSOP heuristic is calculating the score measure s_i for all nodes $i \in N$. For this I have used the formula that Evers et al. (2013) used in their paper, which is developed by Golden et al. (1988). The formula looks like:

$$s_i = \sum_{j \in N} p_j \cdot e^{-\mu f_{ij}}$$

Where $\mu > 0$ is a parameter setting of the heuristic. This score measure takes into account the profit of node i itself and the profit of the nodes in the neighborhood of i. This means that the closer node j is located to node i, the higher the contribution of the profit of node j will be to the score measure of node i. As a result, the nodes that are positioned in clusters of nodes get higher values for the score measure, rather than the individual nodes with high profit, which are more isolated. This score measure will be used in the construction phase.

3.1 The construction phase

In constructing an initial solution, nodes will be added iteratively at the end of the current solution. To also take weight uncertainty into account in deciding which node to add to the current solution, I made use of the probability that the weight realization allow that node i will be reached (and thus the profit of node i is obtained). More specifically, the profit of node i can be obtained if the remaining capacity after visiting node i as the final node in the current tour, is greater than or equal to the expected weight required to return form node i to the depot. Because the distribution of the weights is known, this probability $g_i(x)$ can be calculated as follows:

$$g_i(x) = F(C; shape, scale)$$

Where F is the cumulative distribution function of the gamma distribution and C the total capacity. The shape parameter is equal to the sum of the shape parameters of the current tour x plus the shape parameter of the arc that is between the last node of the current tour and the node i. The expected score measure is then defined by multiplying this probability $g_i(x)$ by the score measure s_i.

With these information the ratio $r_i(x)$ can be calculated for each node i not yet in the solution, to indicate the attractiveness of adding node i to the current solution:

$$r_i(x) = \frac{g_i(x) s_i}{f_{ki} + f_{i0} - f_{k0}}$$

where k is the last node of the current solution. This ratio expresses the expected score measure relative to the expected weight required to add node i to the current solution.

To decide which node will be added next to the current initial solution, a selection must be made by choosing one out of the four nodes with the highest value for the ratio $r_i(x)$ (Tsiligirides (1984)). For each of these nodes, a desirability probability des_i is computed by normalizing the ratios as follows:

$$des_i = \frac{r_i(x)}{\sum_{i'=1}^{4} r_{i'}(x)}$$

where the index i' denotes the node with the $i'th$ largest ratio. Node i will be selected as the next node in the current solution with probability des_i.

Nodes will be added until there is not enough remaining capacity after visiting one of the nodes that are not yet selected in the current solution, to return to the depot.

3.2 Improvement phase

In the improvement phase, three different local search moves are used to improve the initial solution. The local search move that is evaluated are ‘interchange’, ‘insert’ and ‘remove-insert’.

Interchange: Switching the position of two nodes in the tour and also reversing the order of the nodes in between. I have chosen to switch a node with the node that is 3 steps further in the tour. For example, suppose that the current tour consist of the ordered sequence of nodes (0, 1, 2, 3, 4, 5,0). By applying an interchange move at the first selected node (1) we get the solution (0, 4, 2, 3, 1, 5,0). If reversing the order of the nodes in between, (0, 4, 3, 2, 1, 5 0), provides a higher estimated TSOP objective we also apply the reversing move, otherwise only the interchange move. If more than one of such moves results in an increased estimated TSOP objective, the move which results in the highest TSOP objective value is applied.

After applying interchange, the total expected weight of the new tour might be different from the previous tour. Therefore, it might be possible to insert additional nodes in the current tour.

Insert: For all nodes that are not yet selected in the current tour, we evaluate the estimated TSOP value after insertion at all possible positions in the current tour. The node which results in the highest possible increase in the estimated TSOP objective will be inserted at its individual best insert position. Insertion will be performed as long as an increase in the estimated TSOP objective is possible.

Remove-insert: Removing one node from the current tour and inserting one or more new nodes according to the insertion procedure just described. For this remove-insert procedure I accept the first improvement found.

This process of interchange, insert and remove-insert is repeated until none of these moves can improve upon the estimated TSOP objective value.

4.DATASET AND IMPLEMENTATION DETAILS

There are 6 datasets used for this paper. The sets contain 5, 10, 15, 20, 25 and 30 nodes, respectively, and 2, 6, 2, 7, 2 and 10 different capacity parameters. This gives a total of 29 problem instances.

For all of the datasets, I assume that the nodes are fully connected and that the weights on the arcs are gamma distributed. The gamma distribution is a very rich function. This distribution is especially relevant for the OPSW as it is restricted to nonnegative values: travel times cannot be negative. It has also a small probability of large values: an accident in the roadway or a flat tire, for example, can disrupt the delivery. Furthermore, for each instance, the scale parameters of the gamma distribution are fixed to the same value for every arc. The effect of fixing the scale parameter to the same value for every arc is that the sum of the weights distribution can be characterized by the sum of the shape parameters for the arcs traversed.

The datasets contain information about the location of the nodes and the profit obtained if the node is visited. We also know the distance between the different nodes and the shape parameters for the gamma distribution. With the fixed scale parameter value of 0.5, we can calculate the expected weights of each arc.

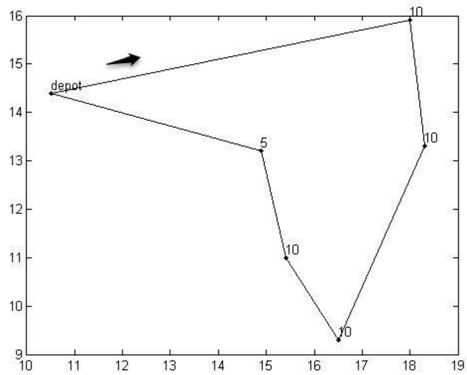
For the score measures s_i, a value of $\mu = 10$ is used, because this value was used by Evers et al. (2013) and it provided also good results in the TSOP heuristic. Since a randomization concept is applied in the construction of the initial solutions, different iterations of the heuristics might result in different solutions. Based on experimental testing I have found that good results are obtained when using 30 iterations.

5.COMPUTATIONAL RESULTS

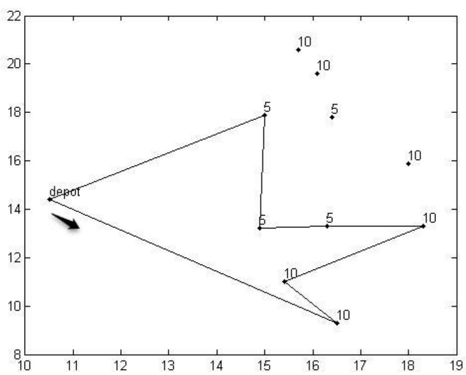
This section presents the results of the TSOP heuristic. I have looked at the effect that capacity parameters have on the objective value of the problem and on the number of selected nodes. I have also checked what the difference is in computation time within the different instances.

5.1 Tours

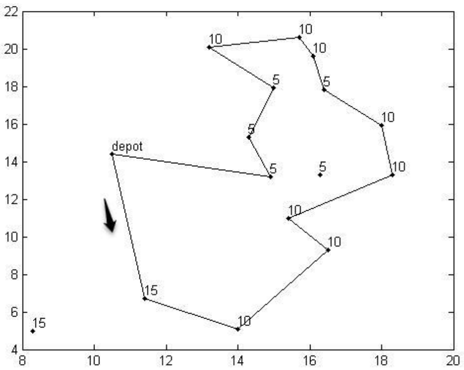
In *figure 2* tours are shown that are obtained from different datasets with different capacity parameters. As can be seen, the nodes near the end of the stochastic tours are typically ones with low profits, and thus not a priority. So they are placed at the end of the tours where there is low probability of them being reached.



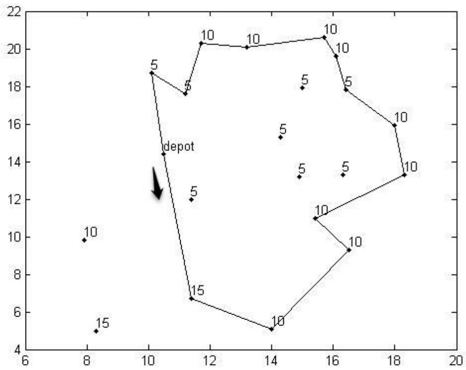
(a) Dataset 1, Capacity 30



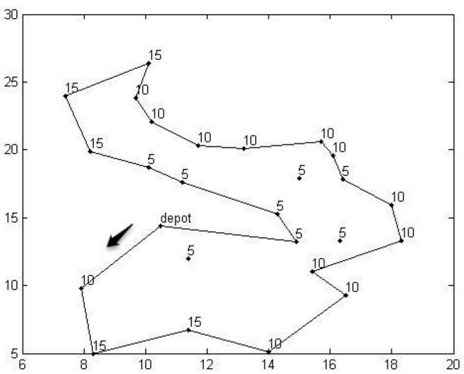
(b) Dataset 2, Capacity 30



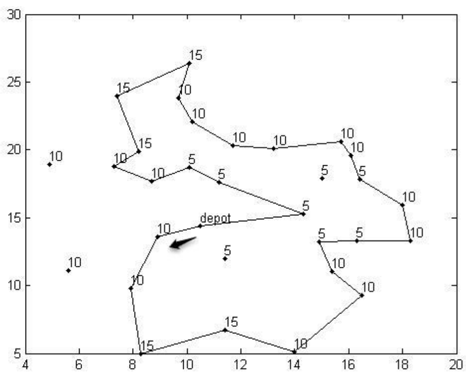
(c) Dataset 3, capacity 55



(d) Dataset 4, capacity 55



(e) Dataset 5, capacity 90

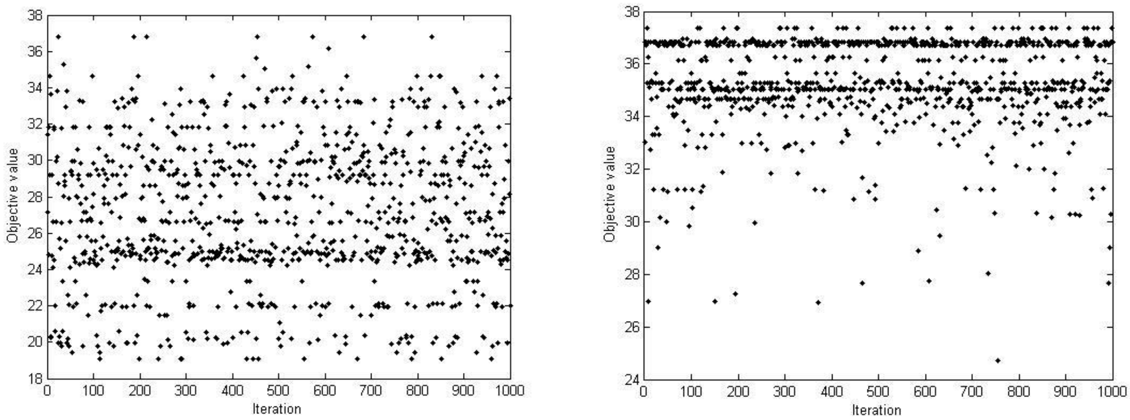


(f) Dataset 6, Capacity 100

Figure 2 Tour

5.2 Stability

First, to illustrate the stability of the TSOP heuristic that is used in this paper, I will focus on one specific problem instance. For this purpose, I use the problem instance from Dataset 2 with capacity 30, 1000 iterations are performed. In *figure 3 (a)* the objective values of the initial solutions that are constructed in the construction phase are shown. Note that the points are very scattered. This is caused by the fact that randomization concepts are used in the construction phase. But in *figure 3 (b)* it can be seen that this is resolved in the improvement phase of the heuristic. The best solution obtained within these 1000 iterations is equal to 37.37. *Figure 3 (b)* shows that a lot of solutions are near to the best solution, what means that the TSOP heuristic is reasonably stable.



(a) Initial solutions (Construction phase) (b) Improved Solution (Improvement phase)

Figure 3 Iteration vs. Objective value for Dataset 2, Capacity 30.

In *table 1*, the mean and the standard deviations of both the initial solutions and improved solutions are shown, to get a better view of the situation. The difference between the mean of the improved solutions and the best solution is circa 6%. Thus, using 30 iterations is also good enough to get a good quality solution. Which is also clear if we look at the values in *table 2.*

	Initial solution	**Improved solution**
Mean	26.75	34.97
Standard deviation	12.95	1.85

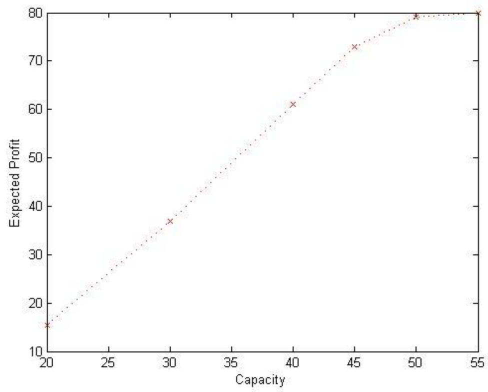
Table 1 Statistics of the initial solutions and improved solutions

	Initial Solution		Improvement Solution	
	Amount	**Percentage**	**Amount**	**Percentage**
= 37.37	0	0.0%	53	5.3%
>36	7	0.7%	302	30.2%
>35	10	1.0%	607	60.7%
>34	28	2.8%	850	85.0%

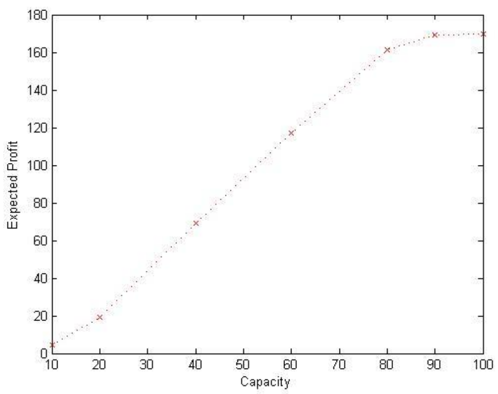
Table 2 The amount and percentage

5.3 Capacity constraint

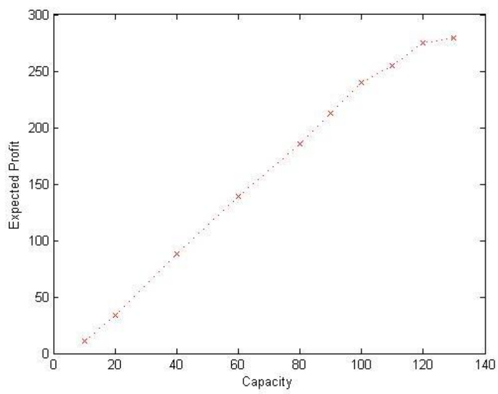
Figure 4 shows the relation of the capacity parameter and the objective value. All graphs have the same form. It is obvious that the profit will increase if the capacity increase. It could be that in de optimal solution all nodes are visited, but that not all of the profit is obtained due to the fact that we work with stochastic weights. Thus when the capacity constrains less, the profit will increase less too (see *figure 3*). This means that the profit will reach the maximum what could be obtain.



(a) Dataset 2



(b) Dataset 4



(c) Dataset 6

Figure 4 Capacity vs. objective value for dataset 2, 4 and 6

Table 3 shows also the number of selected nodes. It can be seen that the profits can increase, while the number of selected nodes remains the same. This will continue until the profit has reached the maximum what could be obtained, because the capacity does not constrain anymore.

Capacity	**Number of selected nodes**	**Profit**	**Capacity**	**Number of selected nodes**	**Profit**	**Capacity**	**Number of selected nodes**	**Profit**
10	4	15.48	**10**	2	4.37	**10**	3	10.53
30	6	37.37	**20**	5	19.02	**20**	5	33,80
40	9	61.07	**40**	11	69.13	**40**	10	88,12
45	10	72,93	**60**	15	117.17	**60**	17	138,76
50	10	79,15	**80**	20	161.42	**80**	20	185,72
55	10	79,91	**90**	20	169,48	**90**	24	213,15
			100	20	169,92	**100**	26	239,72
						110	26	254.92
						120	30	274,68
						130	30	279,60

Table 3 Number of selected nodes and the associated profits for Dataset 2, 4 and 6

5.4 Running time

As earlier mentioned the TSOP heuristic provides good quality solutions with less computational effort, also for larger instances (Evers et al. (2013)). To illustrate the computation time for the heuristic used in this paper, I will make use of all datasets, all with a capacity of 40. The results are shown in *figure 5*. As can be seen in this figure the computation time increases if the dataset gets larger. This makes sense because the heuristic needs more time in larger instances to go along all nodes, in the construction phase as in the improvement phase. Note that the computation time increases reasonably: if there are twice as many nodes, the computation time gets three times as large.

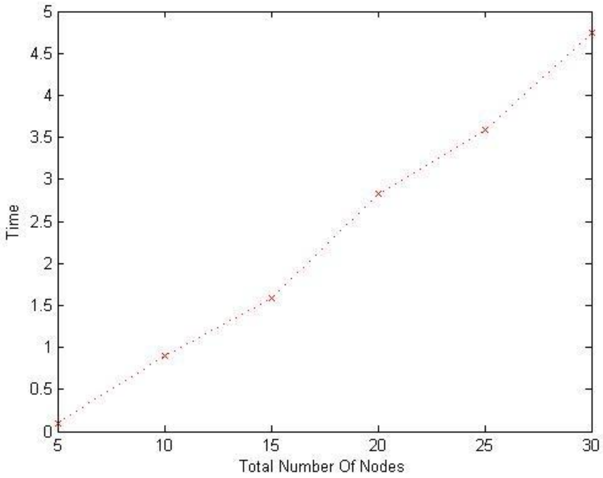


Figure 5 Total number of nodes vs. Running time (in sec). The capacity is equal to 40.

The running times of the instances of dataset 3 are calculated and shown in *figure 6*. As can be seen in this figure, the running time increases if the predefined capacity increases. This is caused by the fact that there are more opportunities to construct a tour and also more opportunities to improve the tour. The computation time increases here also reasonably: if the predefined capacity is twice as large, the computation time gets circa four times as large.

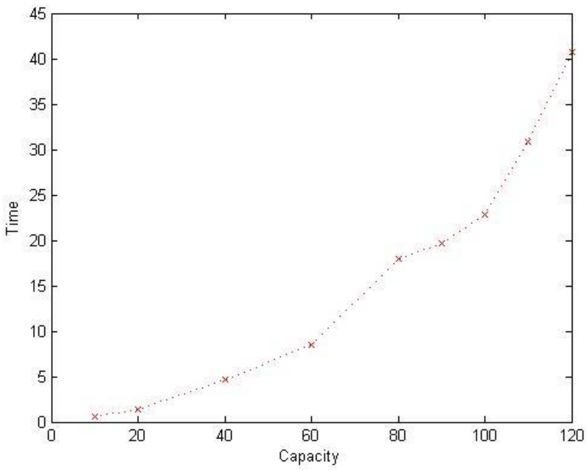


Figure 6 Capacity vs. Running time (in sec). Dataset 3.

6.CONCLUSION

This paper covers the Orienteering Problem with Stochastic Weights (OPSW), where the weights represent travel times and the profits model the importance of the locations. To solve the OPSW, the Two-Stage Orienteering Problem (TSOP) approach, in which we assume that the weights of the arcs follow a given probability distribution, is used. The TSOP is a two-stage recourse model in which the first-stage decision is to construct a tour, which may have to be aborted in the second stage before reaching the final target because of the weight realization. A heuristic approach to solve the TSOP is used and tested. The computational results show that the TSOP heuristic provides good quality results within a reasonable computation time. Tours are obtained, where the nodes near the end of these stochastic tours are typically the ones with low profit, and thus not a priority. The stability of the heuristic is also tested and we conclude that the heuristic is stable, thus the results that are obtained are reliable.

REFERENCES

L. Evers, T. Dollevoet, A.I. Barros, H. Monsuur, Robust UAV mission planning, Annals of Operations Research (2012). doi: 10.1007/s10479-012-1261-8.

L. Evers, K. Glorie, S. van der Ster, A.I. Barros, H. Monsuur, A Two-Stage Approach to the Orienteering Problem with Stochastic Weights. (2013) 1-29.

B. Golden, Q. Wang, L. Lui, A multifaceted heuristic for the orienteering problem. Naval Research Logistics 35 (1988) 359-366.

C. Miller, A. Tucker, R. Zemlin, Integer programming formulations and traveling salesman problems. Journal of the ACM 7 (1960), 326-329.

W. Souffriau, P. Vansteenwegen, J. Vertommen, G. Vanden Berghe, D. Van Oudheusden, A personalised tourist trip design algorithm for mobile tourist guides. Applied Artificial Intelligence 22 (2008) 964-985.

T. Tsiligirides, “Heuristic methods applied to orientering”. *Journal of Operational Research Society 35*, (1984) 797-809.