Solving the Stochastic Time-Dependent Orienteering Problem with Time Windows

Cédric Verbeeck[1], El-Houssaine Aghezzaf[1], Pieter Vansteenwegen[2]

[1] Ghent University, Department of industrial management Technologiepark 903, 9052 Zwijnaarde - Belgium
cedric.verbeeck@ugent.be,elhoussaine.aghezzaf@ugent.be
[2] KU Leuven, Leuven Mobility Research Center, 3001 Leuven - Belgium
pieter.vansteenwegen@kuleuven.be

Keywords: *Metaheuristics, Orienteering problem, time-dependent stochastic travel times*

1 Problem definition

The orienteering problem (OP) is defined on a graph in which the vertices represent geographical locations where a score can be collected. An arc in the graph represents a connection between two vertices and is weighted by its travel time. The goal of the OP is to determine which subset of vertices to visit and in which order so that the total collected score is maximized and a given maximum total travel time is not exceeded. In addition to this, a feasible OP solution should start and end at a predetermined vertex. The OP integrates the knapsack problem (KP) and the travelling salesperson problem (TSP). In contrast to the TSP, not all vertices can be visited in an OP due to a limited travel time. However, determining the shortest path for visiting the selected vertices might decrease the total travel time and helps to visit extra vertices. The OP has many interesting applications in defence, tourism and logistics and a survey on this problem is presented in [1].

This paper presents the orienteering problem with time-dependent stochastic scores and time windows (TD-OPSWTW) where the travel time on each link is both dynamic and stochastic and is therefore modelled as a function of distributions. This specific problem formulation allows us to tackle congestion related issues in routing problems that deal with high uncertainty. Formally, the TD-OPSWTW can be described by defining a set $V_c = 1, ..., v$ of vertices. In this set vertex 1 represents the start depot and vertex v the end depot. We assume there is an arc $(i,\ j)$ between all i and j in V_c. Associated with each vertex $i \in V_c$ is a deterministic non-negative reward r_i and a deterministic non-negative penalty e_i. This reward is earned by visiting the vertex between its opening time o_i and closing time c_i. The penalty is incurred if the visit would occur after its closing time and may represent direct payment or a loss of goodwill. In reality there is a cost associated with a failure to service a customer and deterministic models generally don't consider this cost. Let $T_{i,j}(t_d)$ be a non-negative random variable representing the time required to traverse arc (i, j) departing at departure time t_d. We assume that the distribution on $T_{i,j}(t_d)$ is known for all i, j and t_d. Next, let s_i be the deterministic non-negative service time at vertex i.

Let the random variable A_i be the arrival time at vertex i. For a realization (actually observed value) of A_i, $\overline{A_i}$, we let $R(\overline{A_i})$ be a function representing the reward earned at vertex i when arriving at time $\overline{A_i}$. We assume that $R(\overline{A_i}) = r_i$ for $\overline{A_i} \leq c_i$ and e_i otherwise. Let τ be an order or tour of the vertices in the selected set $M \subseteq V_c$ which begins at the start vertex (1) at time t_0 and ends at end vertex (v). The opening time of the start and end vertices are set equal to t_0 and their scores are set to 0. Subsequently, the closing time of the end vertex is set equal to $t_0 + t_{max}$ and its penalty is set equal to e_v. This way a penalty is also obtained if the tour length exceeds the predetermined length, t_{max}. The expected reward of the a priori tour is equal to:

$$R(\tau) = \sum_{i \in \tau} [P(A_i \leq c_i) * r_i - (1 - P(A_i \leq c_i)) * e_i] \tag{1}$$

We seek an a priori tour τ^* such that $R(\tau^*) \geq R(\tau)$ for every τ.

Research on time-dependent orienteering problems and orienteering problems with stochastic scores is relatively scarce. Furthermore, the combination of both dynamic and stochastic travel times has to our knowledge not been researched yet.

2 SOLUTION METHOD

The proposed algorithm for the TD-OPSWTW consists of a pre-processing step and a metaheuristic framework that makes use of an estimation algorithm to calculate the time-dependent stochastic arrival time distributions.

2.1 Arrival time and objective function calculation

Given a sequence of vertices (the a priori route), an algorithm is required to calculate for every included vertex the arrival time distribution. This distribution allows to determine the probability a score or a penalty is collected. This algorithm will be explained in this section and outputs the expected profit of a vertex which equals the probability of the score minus the probability of the penalty.

In general, three actions can occur when travelling from one vertex to another vertex:

1. The traveller arrives within the opening hours of the vertex which enables him to collect the score after servicing the vertex.
2. The traveller arrives before the opening time of the vertex which forces him to wait first and collect the score after servicing the vertex.
3. The traveller arrives after the closing time of the vertex which forces him to collect the penalty without servicing the vertex. As there is also a penalty defined for the end depot and the closing time of the end vertex is set equal to T_{max}, this case also penalizes for the fact of having a total travel time larger than t_{max}.

The travel time distributions are assumed to be normally distributed and an arrival distribution is equal to the convolution of the departure time distribution and a corresponding travel time distribution. Therefore, most arrival time distributions are also normally distributed if the departure time is normally distributed. However, problems arise when the departure time distribution spans multiple time slots, when for a proportion of the arrival time distribution waiting is necessary at a vertex and when for a proportion of the arrival time distribution the vertex can not be served.

In the first case, when calculating the arrival time distribution, a time slot needs to be determined which corresponds with a departure time distribution. Based on this time slot, the correct travel time distribution can be obtained. However, the departure time distribution may span more than one time slot. If the mean departure time would be used to find the appropriate time slot, a proportion of the departure time distribution is convoluted with the wrong travel time distribution. Therefore, the different parts of the departure time distribution need to be convoluted with different travel time distributions which results in a non normal arrival time distribution. Secondly, waiting at a vertex converts the normally distributed arrival time distribution into a departure time which is a combination of a mass point and a truncated normal distribution. In such cases the arrival and/or departure time distribution and the resulting distribution can no longer be categorized as normal. Thirdly, when a vertex can not be served (due to a late arrival) for a certain proportion of the arrival time distribution, no service time is added to this proportion of the arrival time distribution which converts the departure time distribution into a piecewise function that deviates from a normal distribution. Sampling these distributions using a Monte Carlo simulation during the execution of a metaheuristic turned out to be too computationally expensive. Therefore, a mathematical expression is defined for the mean and standard deviation of these three distributions discussed above. Subsequently, the required arrival time or departure distribution is approximated using a normal distribution with the mean and standard deviation calculated by these mathematical expressions.

2.2 Stochastic ant colony system (SACS)

The ACS is an extension of the work discussed in [2] and is based on the behaviour of a foraging ant colony, using ethereal pheromones trails to mark travelled arcs. This heuristic starts by creating initial solutions that are improved afterwards using a stochastic version of an insert and replace local search move. The stochastic travel times are calculated using the estimation algorithm discussed above.

The stochastic insert and replace move are an extension of their deterministic counterpart discussed in [2]. For every vertex in the solution sequence, the latest possible starting time of the service at the vertex is calculated using the 99% percentile of the travel time distributions. The new arrival time distribution is (approximately) calculated for every non-included vertex for all positions x in the solution sequence. If the 99% percentile of the new arrival time distribution at position $x + 1$ is smaller than its latest possible starting time the vertex can be inserted in the solution. A similar logic is followed for the replace local search move.

As mentioned above, during the execution of the metaheuristic the arrival time distributions are calculated using the estimation algorithm. Therefore, the objective function of the final solution sequence provided by SACS is validated using a Monte Carlo simulation. The difference in objective function between the Monte Carlo simulation and the estimation algorithm is rather small, more specifically 0.01% on average.

3 Results

A set of realistic problem instances was developed based on the road network ($G = (V, A)$) of the Benelux (Belgium, The Netherlands and Luxembourg) containing 425,479 vertices (V) and 519,915 arcs (A). The historical travel time dataset consisting of accurate travel time observations every 15 minutes for a representative Tuesday is available for each arc. A set of instances was created by randomly selecting respectively 20, 50 and 100 vertices ($V_c \in V$) out of this road network and providing them with a score, penalty, opening and closing time and a service time. In a pre-processing step a virtual network is constructed. This network corresponds to a complete graph which consists of vertices representing locations to visit and virtual arcs with time-independent and time-dependent normally distributed travel time profiles which model the travel time behaviour on a concatenation of real arcs.

Since this is the first time the TD-OPSWTW is proposed, the algorithm can not be compared to other solution methods. To compare the performance, all TD-OPSWTW instances were solved as the deterministic and time-independent orienteering problem with time windows (OPTW) according to the mixed integer problem formulation presented by [1] using a commercial solver. Afterwards this optimal solution is evaluated in a stochastic context using a Monte Carlo simulation. Afterwards, a comparison can be made between this stochastically evaluated OPTW solution and the stochastic solution proposed by the SACS. Other performance metrics consist of the required CPU time together with the percentage difference between the solution sequence of the final SACS solution and the optimal OPTW solution sequence. The diversity between two solutions A and B is calculated as the sum of the number of vertices in A not present in B and the number of vertices in B not present in A, divided by the total amount of vertices present in A and B (start and end vertex not included).

On average the SACS is able to improve the stochastically evaluated OPTW solutions with 23.5% and requires 3.5 seconds of computation time. The average diversity between an OPTW solution and the solution provided by the SACS is 36.8%. These results are of course road network specific but nonetheless stress the severe impact of congestion on the quality and structure of the proposed solutions.

4 Conclusion

This research proposes a solution method for the orienteering problem with time windows and time-dependent stochastic travel times (TD-OPSWTW). This specific problem formulation allows tackling congestion related issues in realistic routing problems that deal with uncertainty. The solution method consists of a stochastic version of the ant colony optimization (SACS) specifically developed for the TD-OPSWTW. The SACS uses an insert and replace local search move equipped with a local evaluation metric which speeds up the insertion process.

The solution method is compared on solution quality and computational performance on realistic problem instances based on the realistic road network of Belgium, the Netherlands and Luxembourg.

Promising results were obtained in a reasonable amount of computation time. This should motivate vehicle trip planners to take into account the dynamic and stochastic nature of the travel times in the future.

References

[1] P. Vansteenwegen, W. Souffriau, and D. Van Oudheusden. The orienteering problem: a survey. *European Journal of Operational Research*, 209:1–10, 2011.

[2] C. Verbeeck, K. Sörensen, E.-H. Aghezzaf, and P. Vansteenwegen. A fast solution method for the time-dependent orienteering problem. *European Journal of Operational Research*, 236 (2):419–432, 2014.