The Hazardous Orienteering Problem

Author 1: Alberto Santini

Author 2: Claudia Archetti

Journal: Networks (2022).

The final version is available at:
https://onlinelibrary.wiley.com/doi/10.1002/net.22129

The Hazardous Orienteering Problem

Alberto Santini[1] and Claudia Archetti[2]

[1]Department of Economics & Business, Universitat Pompeu Fabra, Spain; Data Science Centre, Barcelona School of Economics, Spain; ESSEC Business School, France; Institute for Advanced Studies, Paris Cergy Université, France
[2]ESSEC Business School, France

29th December 2022

Abstract

This paper studies the Hazardous Orienteering Problem (HOP), a variant of the more famous Orienteering Problem (OP). In the OP, a vehicle earns a profit for each customer it visits (e.g., to pick up a parcel) subject to an upper bound on the tour time. In the HOP, the parcels picked up at some customers have a probability of triggering a catastrophic event. The probability depends on how long the parcels travel on the vehicle. If any catastrophic event triggers, the entire collected profit is lost. The goal is to determine the tour that maximises the expected profit. The problem has interesting applications in routing of hazardous material, cash-in-transit and law enforcement. We propose a mixed-integer non-linear formulation and techniques both to obtain dual bounds and to produce primal solutions. Computational tests investigate the efficacy of the methods proposed and allow to gain insights into solution features.

Keywords: Orienteering Problem, Primal and Dual Bounds, Hazardous Material Transportation, Cash-in-Transit Logistics

1 Introduction

Travelling Salesman Problems (TSP) with profits are well-studied combinatorial optimisation problems. The TSP asks to find the cheapest Hamiltonian cycle on a graph with costs on the edges (or arcs). Interesting variants occur when not all vertices need to be visited and profits are associated with each of them. In this case, there are three main problems which collectively fall under the umbrella of TSP with profits:

- The Orienteering Problem (OP) [19, 24], which asks to collect the highest possible profit, while respecting an upper bound on travel costs (or time).
- The Prize-collecting TSP [5], which asks to minimise the tour cost (or time), while respecting a lower bound on the amount of profit collected.
- The Profitable Tour Problem [11], which asks to maximise the difference between collected profits and tour cost.

In this paper we consider a novel stochastic generalisation of the OP, which arises in the context of routing of hazardous material, law enforcement with drones, and transportation of high-value objects. Consider the following scenarios.

Routing of hazardous material. A logistic company collecting parcels at client sites must decide which pickups to accept and how to route a vehicle to satisfy the corresponding requests. Each accepted pickup earns a profit, so the company would like to accept as many as possible. However, the number of accepted pickups is limited by the total time the vehicle can spend en route, e.g., to comply with driver regulations or because collected parcels must be available at a warehouse by a given deadline. This setting can be modelled as an Orienteering Problem. In our case, however, some requests involve

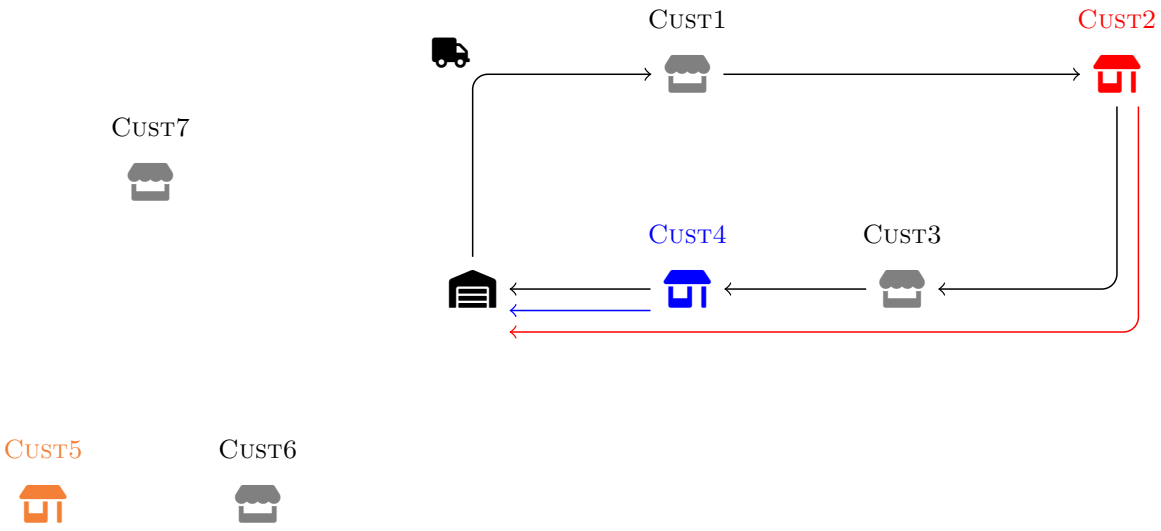


Figure 1: Example of a cash-in-transit scenario with seven customers. The customers drawn in colour increase the risk that the truck is attacked after visiting them. The black arrows indicate the truck's route in a feasible solution.

hazardous material. Such material has a small (but non-zero) probability to catch fire and cause the loss of all transported parcels. The probability of this event is different for each hazardous shipment and is directly proportional to the time the shipment spends in the vehicle, following an exponential distribution. For example, lithium-ion batteries have a small probability to catch fire under mechanical stress suffered on a road vehicle [15, 23].

Law enforcement with drones. A law enforcement agency must decide which targets to visit with an unmanned aerial vehicle, subject to a maximum travel distance dictated by the range of the aircraft. Each target has an associated profit, which reflects how important it is to perform a reconnaissance action at the corresponding location. This scenario can also be modelled as an Orienteering Problem. However, visiting some of the targets allows criminals to detect the aircraft and try to interfere with its operation. The more targets are visited and the longer the aircraft keeps flying afterwards, the higher is the chance of being intercepted or shot down—a catastrophic event which we can model with the loss of the entire collected profit.

Cash-in-transit. A security transport company receives requests for cash pickup from several shops, to deposit their daily income to a bank. The company charges each customer that it can include in its tour. The scenario can be modelled as an Orienteering Problem, given a bound on the travel time due to, e.g., the security guards' work shift duration. The more customers are visited and the higher the amount of cash collected, the more the vehicle becomes an interesting target for assailants: each additional minute spent on the road gives potential robbers a chance to attack the vehicle and cause the total loss of its content. Figure 1 depicts an example scenario. The black node represents the bank. Customers which do not cause the vehicle to be endangered after visiting them (i.e., they are not attractive for potential robbers) are drawn in grey. On the other hand, the red, blue, and orange customers increase the chance that the vehicle is attacked after visiting them. The longer the vehicle is on the road after visiting these customers, the higher the chances of attack. In the solution depicted in Figure 1, the vehicle visits two such customers: the red one and the blue one. Black arrows indicate the truck's route, while we draw coloured arrows to represent the portions of the truck's route which occur after visiting each dangerous customer.

To model the above scenarios, we introduce the stochastic **Hazardous Orienteering Problem** (HOP).

The difference with the classical OP is that visiting some "hazardous" (or "time-bomb") customers causes that, after the visit, the vehicle is at risk of losing its entire collected profit. This probability increases with the number of hazardous customers visited and with the time that the vehicle travels after visiting each hazardous customer. Although, as seen above, the HOP models different real-life applications, in the rest of the paper we will use logistic-based terminology and we will say that the vehicle visits "customers" to collect "parcels", which can "explode" en route (an event which we denote as a "catastrophic event"). We propose a problem formulation which involves the maximisation of a non-concave objective function, together with a set of valid inequalities. Then, we develop different upper bounds based on either a relaxation of the objective function (leading to functions which are easier to handle than the original one) or to a relaxation of a subset of the constraints. We design both exact solution algorithms and a heuristic approach. Extensive computational results show that it is hard to devise tight dual bounds but possible to obtain good primal solutions. In particular, using (i) a black-box non-linear optimisation solver after a log-transformation of the objective function and (ii) a neighbourhood-based heuristic, we were able to generate high-quality solutions in short runtimes.

The main contributions of the paper can be summarised as follows:

1. We introduce the HOP and provide a formal definition of the problem.
2. We present a mathematical formulation of the problem which maximises a non-concave (but log-concave) objective function.
3. We propose multiple techniques to find effective upper and lower bounds, focusing on the diversity of the techniques we propose. In this introductory paper we do not further fine-tune the most powerful approaches, but instead we present promising ideas and compare both exact and heuristic methods.
4. We perform a computational study to investigate the efficacy of the bounds proposed and to gain insights into solution features.

The paper is organised as follows. Section 2 reviews the literature on closely related problems. Section 3 provides the formal description of the HOP and its mathematical formulation, while dual bounds are described in Section 4. Section 5 introduces algorithms to obtain primal solutions. Finally, Section 6 presents computational results and an analysis of solution characteristics, while we report our conclusions in Section 7.

2 Literature Review

In this section we revise the literature related to the HOP. Although the problem is new, it has aspects in common with other problems arising in contiguous areas: stochastic orienteering problems and stochastic problems with catastrophic consequences. It also shares characteristics or application areas with other problems, such as routing of hazardous materials and cash-in-transit.

Stochastic Orienteering Problems. Similar to other routing problems, real-life applications of the OP are often affected by uncertainty. As a consequence, researchers have developed stochastic variants of the OP that incorporate such uncertainty. The main stochastic OP variants are the following:

- The *OP with Stochastic Profits*, in which uncertainty affects the profit collected at each customer. This problem was introduced by lhan, Iravani and Daskin [22] who associated a normal random distribution with each customer and aimed at maximising the probability that the total collected profit is not smaller than a given threshold. Different from the HOP, the visit order does not affect the stochastic profits of the customers.
- The *Probabilistic OP* [2, 3], in which each customer has an associated Bernoulli random variable that models whether the customer will be available. The decision-maker first builds an a priori OP tour visiting a subset of customers and respecting the time bound. In the second stage, when customer availability is revealed, the decision-maker removes unavailable customers from

the tour and visits the remaining customers. The objective is to maximise the expected profit, considering that only available customers yield their corresponding profit. Different from the HOP, if a customer is not available, it is only its profit which is lost, rather than the entire collected profit.

- Uncertainty can also affect time, as in the case of the *OP with Stochastic Travel and Service Times* [7, 14, 30], in which travel and service times are uncertain, and the *OP with Stochastic Time Windows* [40], in which customers are available during stochastic time windows.

Stochastic problems with catastrophic consequences. In the HOP, if a catastrophic event happens then the whole collected profit is lost. Therefore, we can place the HOP in the category of stochastic problems in which some low-probability event has a large negative impact. Sherali et al. [36] were among the first to consider this type of problems in the context of routing of hazardous materials. In their work, different from the HOP, the authors are more concerned with assessing the external impact of the catastrophic event: for example, it is worse to spill toxic liquids in a city centre than in an unpopulated area. Thus, rather than minimising the expected probability that a catastrophic event happens, they minimise the expected impact conditional on the event happening. Other examples of problems with catastrophic consequences are the *Distributionally Robust Stochastic Knapsack Problem* [8] and the *0–1 Time-Bomb Knapsack Problem* [26]. Both are stochastic variants of the classical 0–1 Knapsack Problem. In the first problem, items arrive one by one and their weight is given by a random variable. At any point, the user packing the knapsack can either ask for a new object or stop and collect the profit currently accumulated. If the user asks for a new object and its weight is larger than the residual capacity, the knapsack breaks and all the profit accumulated is lost. In the second problem, a subset of items to pack are time-bombs which can explode with a Bernoulli-distributed probability. If any such item is packed and explodes, then the entire profit packed in the knapsack is lost. Similar to the HOP, the objective is to maximise the expected profit.

Problems with similar characteristics or applications. Finally, we describe existing problems which share some of the peculiarities of the HOP and which are used to model real-life applications in similar areas. One characteristic of the HOP is that the (expected) profit collected at a hazardous customer depends indirectly on the visit time. More precisely, the expected profit depends on the travel time after visiting the customer and therefore a hazardous customer yields a higher expected profit if the vehicle visits it towards the end of the tour (later time) compared to the beginning of the tour (earlier time). The *Team OP with Decreasing Profits* (TOPDP), introduced by Afsar and Labadie [1], also has time-dependent profits. The differences are that in the TOPDP (i) the time dependence is explicit (the profit is a linear function of the visit time), (ii) the dependence is inverse compared to the HOP (later visits yield lower profit), (iii) there is no probability of losing the entire profit in case a negative event happens, (iv) the authors consider a multi-vehicle version of the problem. Typical applications of the TOPDP are in maintenance routing, in which a customer might be willing to pay more to be visited earlier, and in humanitarian logistics or firefighting, in which visiting a target earlier increases the probability of saving lives or property. Other problems which share applications with the HOP are *cash-in-transit* problems [9]. They involve routing one or more vehicles containing cash, which are typically used to either replenish ATM machines or collect cash at stores at the end of the day. These problems have been modelled as extensions of the OP (see, e.g., [28]), although the main focus has been put on coverage and service level considerations [29] rather than on the probability that the vehicle is attacked. An exception is the case of multiperiod cash-in-transit problems, in which this risk is often explicitly considered and mitigated by generating different routes for each period [18, 20, 27]. A similar problem is the *Risk-Constrained Cash-in-Transit Vehicle Routing Problem* (RCTVRP) introduced in Talarico, Sörensen and Springael [37] where the goal is to design routes for a fleet of vehicles carrying cash considering that the probability that a vehicle is robbed is a function of both the amount of cash it carries and the distance it travels. More precisely, the authors define the risk of being attacked over an edge of length t_{ij} while carrying an amount D_i of cash as $D_i \cdot t_{ij}$. The objective function of the RCTVRP minimises the total travel cost, under a constraint that the maximum risk over any used arc is not larger than a given threshold. Under these assumptions, the authors formulate the RCTVRP

as a Mixed-Integer Program, which they show to work in reasonable computing times only on small instances. The authors then develop a metaheuristic algorithm to deal with medium- and large-size instances.

3 Problem definition and mathematical formulation

Let $G = (V', A)$ be a complete directed graph with vertex $0 \in V'$ representing the depot, where the route starts and ends, while set $V = \{1, \ldots, n\} \subset V'$ represents the customer locations; set A is the arc set. Each vertex $i \in V$ has an associated profit $p_i \in \mathbb{R}^+$ and each arc $(i, j) \in A$ has a travel time $t_{ij} \in \mathbb{R}^+$. We assume that travel times satisfy the triangle inequality.

We model the probability of catastrophic events through the exponential distribution. This distribution is commonly used in survival analysis and reliability theory, to model the occurrence of low-probability/high-impact events [13, 31]. Its popularity derives both from its theoretical properties (e.g., it is memoryless and it assumes constant risk over time) and because it fits well empirical quantitative risk assessment curves [39]. We consider a subset $H \subseteq V$ of hazardous vertices, and associate with each vertex $i \in H$ a parameter $\lambda_i \in \mathbb{R}^+$. A vertex $i \in H$ has an associated exponential random variable with rate λ_i modelling the possibility that the parcel collected at i explodes and causes the loss of the entire vehicle content. The cumulative distribution function of the exponential random variable associated with i is

$$F_i(t) = 1 - e^{-\lambda_i t} \quad (t \geq 0), \tag{1}$$

and it represents the probability that the catastrophic event will happen within a time $t \geq 0$ counted from the moment the vehicle visits vertex i. Finally, we denote with $T \in \mathbb{R}^+$ the upper bound on the tour travel time. The goal is to determine an elementary tour that maximises the expected collected profit and does not exceed the bound T on the duration.

3.1 A non-linear mixed-integer formulation

Consider the following decision variables:

- $x_{ij} \in \{0, 1\}$ for $(i, j) \in A$, taking value 1 iff the vehicle traverses arc (i, j).
- $y_i \in \{0, 1\}$ for $i \in V'$, taking value 1 iff the vehicle visits vertex i.
- $w_i \in \mathbb{R}^+$ for $i \in V$, representing the travel time along the path that leads from i back to the depot, if i is visited, or 0 otherwise.

Consider the example presented in Figure 1. Figure 2 refers to the same example: in the top-most part of the figure we depict the vehicle's tour and the travel times of the arcs used. We report the corresponding value of the w variables in the middle part of the figure. Each variable w_i measures the amount of time that the vehicle spends travelling, after it leaves customer i. In the bottom-most part of the figure, we report the probability that no catastrophic event occurs, as a function of time. Because the first hazardous customer is Cust2, the probability starts decreasing at time 3, following the exponential curve $e^{-\lambda_2 w_2}$. When the vehicle visits the next hazardous customer, Cust4, the decrease is even steeper because it follows function $e^{-\lambda_2 w_2} \cdot e^{-\lambda_4 w_4}$.

A non-linear mixed-integer formulation for the HOP is:

$$\max \quad \Big(\sum_{i \in V} p_i y_i\Big) \cdot \Big(\prod_{i \in H} e^{-\lambda_i w_i}\Big) \tag{2}$$

$$\text{s.t.} \quad y_0 = 1 \tag{3}$$

$$\sum_{(i,j) \in \delta^+(i)} x_{ij} = y_i \qquad \forall i \in V' \tag{4}$$

$$\sum_{(j,i) \in \delta^-(i)} x_{ji} = y_i \qquad \forall i \in V' \tag{5}$$

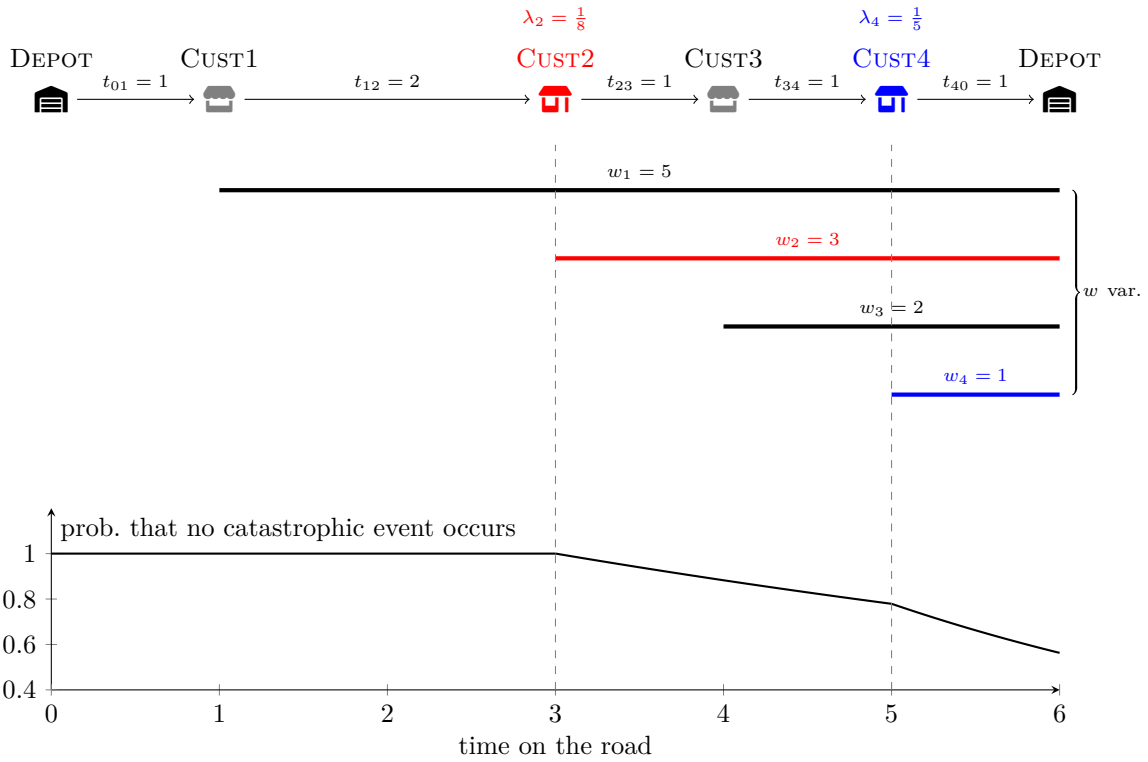


Figure 2: Value of the w variables and probability that no catastrophic event occurs, in the example depicted in Figure 1. The probability that no catastrophic event occurs corresponds to $\prod_{i \in H} e^{-\lambda_i w_i}$.

$$\sum_{(i,j) \in A} t_{ij} x_{ij} \leq T \tag{6}$$

$$w_i \leq M y_i \qquad \forall i \in V \tag{7}$$

$$w_i \geq w_j + t_{ij} - M(1 - x_{ij}) \qquad \forall (i,j) \in A \ : \ i,j \neq 0 \tag{8}$$

$$w_i \geq t_{i0} x_{i0} \qquad \forall i \in V \tag{9}$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \tag{10}$$

$$y_i \in \{0,1\} \qquad \forall i \in V' \tag{11}$$

$$w_i \in [0, T - t_{0i}] \qquad \forall i \in V, \tag{12}$$

where $\delta^+(i)$ denotes the set of arcs with origin i, $\delta^-(i)$ denotes the set of arcs with destination i, and M is a sufficiently large number (for example, $M = T - t_{0i}$ in (7) and $M = T - t_{j0} + t_{ij}$ in (8)). Objective function (2) maximises the expected profit under the assumption that the entire profit is lost if a parcel explodes. Constraint (3) ensures that the depot is visited, while (4) and (5) are the classical flow balance constraints. Constraint (6) enforces that the total travel time of the tour does not exceed the bound T. Constraint (7) forces w_i to be zero for vertices not visited in the tour. Constraints (8) set a bound on the value of w_i in case i is visited immediately after j. Note that these constraints exclude the case in which j is the depot and, thus, i is the last visited vertex. This latter case is considered in constraint (9). Also note that, if $x_{ij} = 1$, constraint (8) forces $w_i \geq w_j + t_{ij}$, but because higher values of w_i are penalised in the objective function, the above inequality will be enforced with strict equality in any optimal solution. Analogously, (9) is enforced with equality when $x_{i0} = 1$. Inequalities (8)–(9) are adapted from the Miller-Tucker-Zemlin (MTZ) inequalities for the TSP [25] and prevent subtours. Constraints (10)–(12) define variables domain. Note that the upper bound on variable w_i in constraint (12) is valid because the latest time a customer can be visited corresponds to the time bound T minus the shortest time to reach the depot from the customer location, i.e., t_{0i}.

3.2 Lifting of inequalities

We lift constraints (7) and (8) as proposed by Desrochers and Laporte [12]. In the following, we provide the formulation of the lifted inequalities and we refer the reader to [12] for the proof that the new inequalities are valid. Inequality (7) can be lifted as follows:

$$w_i \leq (T - t_{0i})y_i + \sum_{(i,j)\in A} (T - t_{0j} - t_{ji} + t_{0i})x_{ji} \quad \forall i \in V. \tag{13}$$

Inequality (8) can be strengthened as follows:

$$w_i \geq w_j + t_{ij} - M_{ij}(1 - x_{ij}) + (M_{ij} - t_{ij} - t_{ji})x_{ji} \quad \forall (i,j) \in A \ : \ i,j \neq 0. \tag{14}$$

where $M_{ij} = \max\{t_{i0} - t_{ij}, T - t_{0j} + t_{ij}, T - t_{0i} - t_{j0} + t_{ij}\}$. Note that Desrochers and Laporte [12] propose additional lifted inequalities, but they cannot be applied to the HOP because they assume that the vehicle visits all customers.

3.3 Valid inequalities

We propose four valid inequalities. The first three are specific to the HOP, while the last comprises the classical subtour elimination constraints.

The first inequality is the following:

$$w_i \geq \sum_{(i,j)\in\delta^+(i)} t_{ij}x_{ij} \quad \forall i \in V, \tag{15}$$

which ensures that the travel time after visiting i is at least as large as the travel time of the arc used to leave i, if i is visited. This inequality is particularly useful when solving the continuous relaxation of the HOP (see Section 4.4) because "big-M" constraints (8) and (9) allow fractional solutions in which variables w_i associated with visited customers take value 0.

The second inequality reads as follows:

$$w_i \leq \sum_{(j,k)\in A} t_{jk}x_{jk} - t_{0i} \quad \forall i \in V. \tag{16}$$

It assumes that the triangle inequality holds and ensures that the travel time after visiting i cannot exceed the total tour travel time, minus the minimum time required to get from the depot to i (this minimum is t_{0i} and is achieved when i is the first visited vertex). This latter inequality can be extended considering the case when i is at least the *second* vertex visited after leaving the depot. Let $t'_{0i} = \min_{j\neq i}\{t_{0j} + t_{ji}\}$ be the shortest path from 0 to i when visiting at least one more vertex in between. Then the following inequality is valid:

$$w_i \leq \sum_{(j,k)\in A} t_{jk}x_{jk} - t'_{0i} + Mx_{0i} \quad \forall i \in V, \tag{17}$$

where M is a sufficiently large number (e.g., $M = t'_{0i} - t_{0i}$). Note that constraint (17) becomes moot when i is the first vertex visited after leaving the depot. We also remark that the last two valid inequalities, which bound w_i from above, can be omitted for customers $i \in H$. For these customers, in fact, the objective function already penalises the corresponding w_i.

We finally include the generalised subtour elimination constraints (GSECs)

$$\sum_{i\in S}\sum_{j\notin S} x_{ij} \geq y_k \quad \forall S \subseteq V, \ \forall k \in S \tag{18}$$

as valid inequalities. Although subtours are already prevented by inequalities (8) and (9), these MTZ-like constraints give notoriously poor continuous relaxation bounds (see, e.g., [6]). GSECs, on the other

hand, give better bounds but are impractical to enumerate because their number grows exponentially in the instance size. Therefore, we initially do not include cuts (18) in the model and we dynamically separate them, as explained below.

To detect the GSECs violated by a fractional solution $\left(x^*_{ij}, y^*_i\right)$, we solve several max-flow/min-cut problems extending the separation procedure of Fischetti, Gonzalez and Toth [16]. We build an auxiliary graph $G^* = (V^*, A^*)$, with $V^* = \left\{i \in V' \,:\, y^*_i > 0\right\}$ and $A^* = \left\{(i,j) \mid i,j \in V^*,\ i \neq j\right\}$. We associate with each arc $(i,j) \in A^*$ a capacity equal to x^*_{ij}. The separation procedure considers vertices $i \in V^* \setminus \{0\}$ in decreasing order of value of y^*_i. For each vertex, it finds a minimum-capacity $(0,i)$-cut. Let $S_i, V^* \setminus S_i$ be such cut, where $i \in S_i$ and $0 \notin S_i$. If the corresponding maximum flow between 0 and i is smaller than y^*_i, then the solution violates a GSEC and we add to the model the constraint associated with S_i. Similar to what Fischetti, Gonzalez and Toth [16] proposed for the OP, a simple way to avoid generating the same set S_i twice is to increase the capacity of arc $(0,i)$ by $1 - \sum_{j \notin S_i} \sum_{k \in S_i} x_{jk}$ before moving to the next vertex.

4 Upper bounds for the HOP

In this section, we propose upper bounds for the HOP based on either the relaxation of the objective function or of a subset of the constraints. We also present an upper bound obtained from dynamic programming.

4.1 Local hazard bound

The first bound stems from the following observation. If, instead of destroying the entire vehicle content, the explosion of a parcel only caused the loss of the parcel itself, the corresponding expected profit would overestimate the expected profit of the HOP. Therefore, the first upper bound is based on assuming that, when a catastrophic event caused by a visit to $i \in V$ occurs, the planner only loses the profit associated with i. This corresponds to changing objective function (2) as follows:

$$\sum_{i \in V} p_i y_i e^{-\lambda_i w_i}, \tag{19}$$

with the convention that $\lambda_i = 0$ if $i \notin H$. This bound, *per se*, does not dramatically simplify the objective function; for example, (19) is still non-linear. However, we will use (19) in the rest of this section to devise linear upper bounds.

4.2 Linear approximation

The product of exponential functions in objective (2) can be bounded from above by a product of linear functions. Let $g_i(w_i) = e^{-\lambda_i w_i}$. When considering g_i as a function $[0, T - t_{0i}] \to \mathbb{R}$, it is easy to show that g_i is bounded from above by the linear function

$$h_i(w_i) = \frac{\pi_i - 1}{T_i} w_i + 1, \tag{20}$$

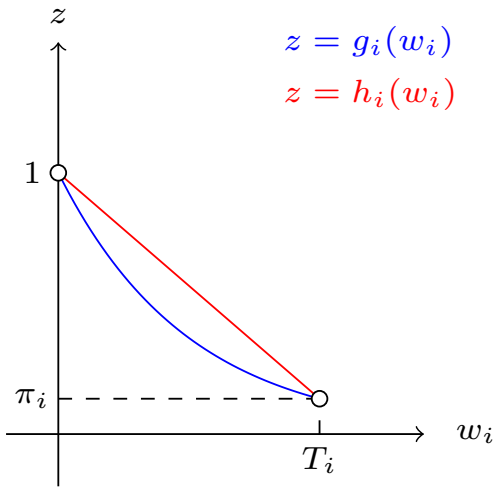
where $T_i = T - t_{0i}$ and $\pi_i = e^{-\lambda_i T_i}$. Figure 3a depicts the relation between g_i and h_i. Each function h_i overestimates the corresponding g_i and, thus, objective function

$$\Big(\sum_{i \in V} p_i y_i\Big) \cdot \Big(\prod_{i \in H} \big(\frac{\pi_i - 1}{T_i} w_i + 1\big)\Big) \tag{21}$$

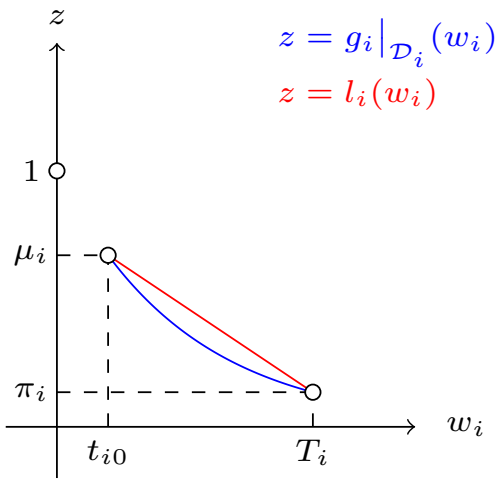
is an upper bound for (2).

Combining (20) with (19) we obtain a quadratic relaxation of the original problem with objective function

$$\sum_{i \in V} p_i y_i \Big(\frac{\pi_i - 1}{T_i} w_i + 1\Big). \tag{22}$$



(a) Relation between g_i and h_i.



(b) Relation between $g_i\big|_{\mathcal{D}_i}$ and l_i.

Figure 3: Linear and piecewise-linear approximations overestimating of the original objective function.

Because $y_i w_i = w_i$ and $\pi_i = 1$ for $i \notin H$, we can write (22) as a linear objective function:

$$\sum_{i \in H} p_i \frac{\pi_i - 1}{T_i} w_i + \sum_{i \in V} p_i y_i. \tag{23}$$

Using (23) we can solve a mixed-integer linear programme to obtain an upper bound on the original non-linear problem.

4.3 Piecewise linear approximation

We can tighten the linear approximation presented in Section 4.2 if we note that the domain of variables w_i is not the entire interval $[0, T_i]$, but rather the disconnected set $\mathcal{D}_i = \{0\} \cup [t_{i0}, T_i]$. Thus, functions g_i restricted to domain $\mathcal{D}_i$ are bounded from above by functions l_i defined as

$$l_i(w_i) = \begin{cases} 1 & \text{if } w_i = 0 \\ \frac{\mu_i - \pi_i}{t_{i0} - T_i} w_i + \pi_i - \frac{T_i(\mu_i - \pi_i)}{t_{i0} - T_i} \coloneqq \alpha_i w_i + \beta_i & \text{if } w_i \in [t_{i0}, T_i] \end{cases}, \tag{24}$$

where $\mu_i = e^{-\lambda_i t_{i0}}$. Figure 3b depicts the relation between $g_i\big|_{\mathcal{D}_i}$ and l_i.

Each function l_i overestimates the corresponding function $g_i\big|_{\mathcal{D}_i}$ and thus, by the fact that $w_i = 0 \iff y_i = 0$, function

$$\Big(\sum_{i \in V} p_i y_i\Big) \cdot \left(\prod_{i \in H} \big(y_i(\alpha_i w_i + \beta_i - 1) + 1\big)\right) \tag{25}$$

is an upper bound for (2). Combining (25) with (19) we obtain a polynomial relaxation of the original problem with objective function

$$\sum_{i \in V} p_i \beta_i y_i \big(y_i(\alpha_i w_i + \beta_i - 1) + 1\big) \tag{26}$$

and, using the fact that $y_i^2 = y_i$, $y_i w_i = w_i$, and $\alpha_i = 0$ for $i \notin H$, we obtain the linear objective function

$$\sum_{i \in V} p_i \beta_i y_i + \sum_{i \in H} p_i \alpha_i w_i. \tag{27}$$

Using (27), we can solve a mixed-integer linear program to obtain an upper bound on the original problem and, because $l_i(w_i) \leq h_i(w_i)$ (for all $i \in H$), such bound is tighter than the one derived using objective function (23).

4.4 Continuous relaxation

Denote with $X = (\vec{x}, \vec{y}, \vec{w})$ the vector of all variables, and with $b(X)$ objective function (2). The continuous relaxation of model (2)–(12) asks to maximise b subject to constraints (3)–(9), but with the variable domain definitions replaced by:

$$0 \leq x_{ij} \leq 1 \quad \forall (i,j) \in A \tag{28}$$
$$0 \leq y_i \leq 1 \quad \forall i \in V' \tag{29}$$
$$0 \leq w_i \leq T_i \quad \forall i \in V, \tag{30}$$

where T_i is as defined in Section 4.2.

Unfortunately, b is neither convex nor concave (as it follows immediately from $B(u,v) = u \cdot e^v$ being neither convex nor concave). However, b is log-concave and by maximising the logarithm of (2) one gets the same maximiser as for the original problem. We then propose to solve the continuous optimisation problem of minimising $f(X) := -\ln b(X)$ subject to (3)–(9). After preliminary experiments (see Section 6.2), in the final version of the algorithm we use the lifted (13) and (14) instead of, respectively, (7) and (8), and we add additional valid inequalities (15) and (18). An explicit formula for f is

$$f(X) = -\ln\left(\sum_{i \in V} p_i y_i\right) + \sum_{i \in H} \lambda_i w_i.$$

Because both the feasible region of this problem (which we denote as P) and objective function f are convex, we can use the Frank-Wolfe algorithm [17] to find a minimum of f. In the following, we detail the procedure used to minimise f.

The Frank-Wolfe algorithm starts with a feasible solution $\bar{X}$ and gets an improving solution $\bar{X}'$ solving the auxiliary problem

$$\bar{X}' = \arg\min\left\{\nabla f(\bar{X})^\top X \mid X \in P\right\}. \tag{31}$$

Solution $\bar{X}$ is a local (and, by convexity of f, global) optimum if $\nabla f(\bar{X})^\top (\bar{X}' - \bar{X}) \leq 0$. Otherwise, the next solution is obtained by minimising f over the segment joining $\bar{X}$ and $\bar{X}'$:

$$\bar{X} \leftarrow \arg\min\left\{f(X) \mid X \in \left[\bar{X}, \bar{X}'\right]\right\}. \tag{32}$$

We can solve (32) with a line search, minimising the one-dimensional function which maps $\gamma \mapsto f\big(\gamma \bar{X} + (1-\gamma)\bar{X}'\big)$. Such problem has closed-form solution

$$\gamma^* = \frac{\frac{\sum_{i \in V} p_i(\bar{y}_i - \bar{y}'_i)}{\sum_{i \in H} \lambda_i (\bar{w}_i - \bar{w}'_i)} - \sum_{i \in V} p_i \bar{y}'_i}{\sum_{i \in V} p_i (\bar{y}_i - \bar{y}'_i)}, \tag{33}$$

where $\bar{y}_j$ and $\bar{w}_j$ are, respectively, the values of variables y_j and w_j in solution $\bar{X}$ (and analogously for $\bar{y}'_j$ and $\bar{w}'_j$ in solution $\bar{X}'$).

As for the auxiliary problem (31), it is a linear problem (LP) with feasible region P and objective function

$$\min \sum_{i \in V} \frac{\partial f}{\partial y_i}(\bar{X}) \cdot y_i + \sum_{i \in H} \frac{\partial f}{\partial w_i}(\bar{X}) \cdot w_i =$$
$$\min \sum_{i \in V} \frac{-p_i}{\sum_{j \in V} p_j \bar{y}_j} y_i + \sum_{i \in H} \lambda_i w_i. \tag{34}$$

4.5 Non-elementary tour

We obtain an additional upper bound by dropping the elementarity requirement and allowing the same customer to be visited multiple times. To obtain the optimal solution to this relaxed problem, we use the state-space-relaxed Dynamic Programming algorithm described in Section 5.1.

5 Dynamic Programming

In this section we present an exact method for the HOP, based on dynamic programming. In practice, the runtime of this algorithm is too high to use it as an exact solution method. However, one can use it as a heuristic stopping its execution and using the best solution found.

In the labelling algorithm we propose, we associate a label with each partial path from a vertex to the depot. Extending such partial paths "backwards" until the first visited vertex is the depot itself (while ensuring that no constraint is violated) produces a valid HOP tour. The number of such tours to consider is limited by pruning unpromising labels via dominance rules as explained below.

Label definition. We denote a label as a tuple $L = (v, W, p, \eta, \pi, t)$, where: $v \in V$ is the current starting vertex of the partial path; $W \in \{0,1\}^{|V'|}$ is a binary vector whose i-th component is 1 iff the partial path visits vertex $i \in V'$; $p \in \mathbb{R}_0^+$ denotes the total profit accumulated at the customers visited by the partial path; $\eta \in (0, 1]$ is the probability that no loaded parcel explodes while travelling along the partial path; $\pi = p \cdot \eta$ is the expected profit accumulated along the partial path; $t \in \mathbb{R}_0^+$ denotes the travel time of the partial path. The set of labels is initialised with the single label $L_0 = (0, \vec{0}, 0, 1, 0, 0)$.

Label extension. A label $L = (v, W, p, \eta, \pi, t)$ is extended to a vertex $i \in V'$ if $W_i = 0$, i.e., the path has not visited customer i, and $t_{0i} + t_{iv} + t \leq T$, i.e., visiting i would not violate the travel time bound. If the extension is feasible, label L is extended to label $L' = (v', W', p', \eta', \pi', t')$, where:

$$v' = i, \quad W'_j = \begin{cases} W_j & \text{if } j \neq i \\ 1 & \text{if } j = i \end{cases}, \quad p' = p + p_i, \quad \eta' = \eta e^{-\lambda_i (t_{iv} + t)}, \quad \pi' = p'\eta', \quad t' = t_{iv} + t.$$

Label dominance. We say that label L_1 dominates label L_2 if any feasible completion of the partial path associated with L_1 is feasible for L_2 and this latter has a worse objective value. In this case, we shall not extend L_2, as the optimal solution cannot contain the path associated with L_2 as a sub-path. In our problem, L_1 dominates L_2 if: (i) $v_1 = v_2$, i.e., both labels refer to a partial path starting at a same vertex; (ii) $W_1 \leq W_2$ component-wise, i.e., the partial path associated with L_1 visits a subset of the vertices visited by the partial path associated with L_2; (iii) $p_1 \geq p_2$, i.e., the profit collected by L_1 is not lower than the profit collected by L_2; (iv) $\eta_1 \geq \eta_2$, i.e., L_1 has a probability of *not* exploding which is not lower than that of L_2; (v) $t_1 \leq t_2$, i.e., the travel time of L_1 does not exceed that of L_2; (vi) at least one of the inequalities at points (ii)–(v) is strict. Note that $W_1 \leq W_2$ and $p_1 \geq p_2$ imply that $W_1 = W_2$ and $p_1 = p_2$. Therefore, $\eta_1 \geq \eta_2$ also implies $\pi_1 \geq \pi_2$.

Optimal path. We recover the optimal path choosing the label with the highest objective value π, among all undominated labels which are extended up to the depot 0. Keeping, for each label L, a pointer to the label from which L was extended, allows us to recover the entire path.

Heuristic. We note that any partial path can be converted into a complete solution by connecting the depot to the head of the path. As such, if we take care to always perform such a connection as the first one tried when extending a label, we can build a pool of primal feasible solutions very early during the run of the algorithm. Therefore, we are able to return a feasible solution at any moment (for example, when we reach a maximum runtime), by choosing the best label which was extended up to the depot.

5.1 State-space relaxation

We have explained how to use the proposed labelling algorithm to produce primal solutions either as an exact algorithm or as a heuristic. We conclude this section showing that the same algorithm can also be used to compute upper (dual) bounds, using the relaxation technique described below.

State-space relaxation (SSR) is a technique introduced by Christofides, Mingozzi and Toth [10] to reduce the number of labels generated by a labelling algorithm. SSR projects the space of all possible labels

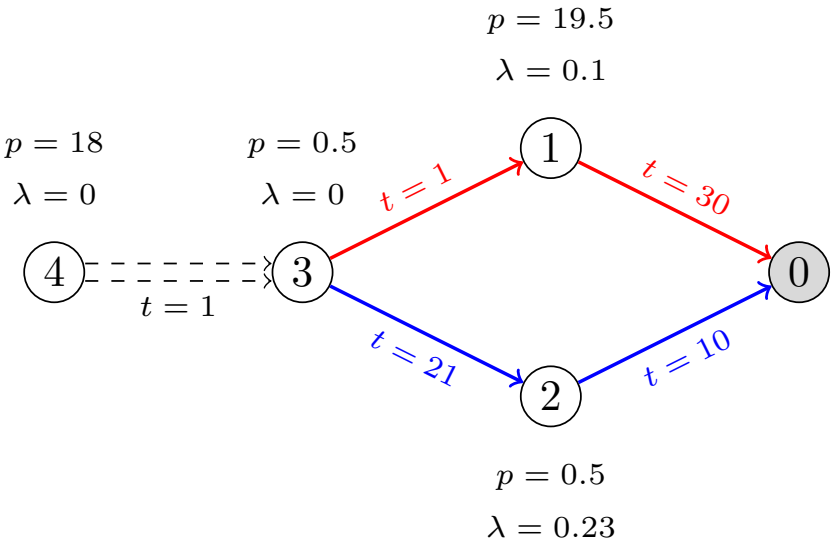


Figure 4: Sample instance showing that it is necessary to use both p and η in the dominance criterion of state-space relaxed labels, because comparing π can lead to wrongly excluding valid labels.

(say, $\mathcal{S}$) into a lower-dimensional space (say, $\mathcal{S}'$), and runs the algorithm in this reduced space. However, during the reduction, the original labels corresponding to both feasible and infeasible partial paths are projected to $\mathcal{S}'$. Therefore, the optimal label in the reduced space might correspond to an infeasible tour. In this case, the algorithm would provide an upper bound on the value of the optimal solution.

In our case, the projection maps vector W to the sum of its elements $\Sigma = \sum_{j \in V'} W_j$. In this way, labels cannot keep track of which customers have been visited and their partial paths can include multiple visits to the same customer.

In the label extension checks, condition $W_i = 0$ is replaced by two checks: $j \neq v$, i.e., we are not "staying" at the same customer, and $\Sigma < n + 1$, i.e., we are not visiting more vertices than there are in the graph. The new component is updated as $\Sigma' = \Sigma + 1$ and dominance condition (ii) is replaced by $\Sigma_1 \leq \Sigma_2$.

Remark. *It might be tempting, when using SSR, to replace dominance conditions (iii) $p_1 \geq p_2$ and (iv) $\eta_1 \geq \eta_2$ with the single condition $p_1\eta_1 = \pi_1 \geq \pi_2 = p_2\eta_2$. This choice would allow to dominate more labels, because π_1 can be larger than π_2 without both p_1 and η_1 being, respectively, larger than p_2 and η_2. However, this check would lead to an incorrect dominance rule. In fact, it is possible that $\pi_1 > \pi_2$ but, when extending both L_1 and L_2 to the same vertex j, $\pi'_1 = \pi'_2$. In case $\pi_1 > \pi_2$ was the only inequality satisfied strictly, we would have that L_1 dominates L_2 but extension L'_1 does not dominate L'_2.*

This situation, although rare, can happen; Figure 4 shows an example with a graph and the partial path associated with two labels (in blue and red, respectively). Consider label $L_1 = (v_1 = 3, \Sigma_1 = 2, p_1 = 20, \eta_1 = 0.05, \pi_1 = 1, t_1 = 31)$ and $L_2 = (v_2 = 3, \Sigma_2 = 2, p_2 = 1, \eta_2 = 0.1, \pi_2 = 0.1, t_2 = 31)$. If we do not consider p and η in the dominance, but only consider π, L_1 would dominate L_2 and the only strict inequality would be $\pi_1 > \pi_2$ (Σ and t being equal). If we now extend both labels to a new non-hazardous vertex 4 with $p_4 = 18$, $\lambda_4 = 0$ and $t_{43} = 1$, we would obtain: $L'_1 = (v'_1 = 4, \Sigma'_1 = 3, p'_1 = 38, \eta'_1 = 0.05, \pi'_1 = 1.9, t'_1 = 32)$ and $L'_2 = (v'_2 = 4, \Sigma'_2 = 3, p'_2 = 19, \eta'_2 = 0.1, \pi'_2 = 1.9, t'_2 = 32)$. Because all tested components are equal, L'_1 does not dominate L'_2, invalidating the validity of the "simplified" dominance criterion.

We strengthen the SSR version of the algorithm by explicitly forbidding 2-cycles (see, e.g., [21]). To this end, we add to our label a component $u \in V'$ indicating the vertex which, in the partial path, is

visited immediately after the current vertex v. We forbid extending a label L to a vertex j if $u = j$. When L is extended to a new label L', we set $u' = v$. Finally, when checking dominance between labels L_1 and L_2, we add the additional condition that $u_1 = u_2$. This version of the algorithm allows fewer labels to dominate each other, but produces a considerably tighter upper bound on the objective value of the optimal solution.

6 Computational results

In this section we present the results of our computational tests using a machine equipped with 4 GB RAM and an Intel Xeon CPU running at 2.4 GHz. All algorithms were coded using the Python programming language, version 3.8. We used Gurobi 9.0.0 as the black-box MIP solver for the linear and piecewise-linear bounds and as the LP solver for auxiliary problem (31). We used Baron 22.1 as the black-box solver for the non-linear model.

The instance generator, instance files, solvers, results files and all the scripts used to generate tables and figures in this paper are available on-line [35].

6.1 Instances

We generate HOP instances starting from the popular Tsiligirides OP instances [38], a set of 49 instances divided into three groups. Instances of each of the three groups contain, respectively, 20, 31 and 32 vertices. To transform OP instances into HOP instances, we use the following generation method. For each base instance, we first choose the number of customers that will be hazardous as $n \cdot \alpha^{\text{Tsi}}$, where n is the number of customers in the instance and $\alpha^{\text{Tsi}} \in \{0.1, 0.2, 0.3, 0.4\}$ is a parameter. The hazardous customers are randomly chosen among all customers. Next, we assign a value for the exponential distribution parameter λ_i to each hazardous customer. In our generation process, λ_i is chosen uniformly at random between 0.05 and 0.1. Finally, to make hazardous customers more attractive, we multiply their original profit by a parameter $\beta^{\text{Tsi}} \in \{2, 3, 4, 5\}$.

Following the above procedure, we generate 784 instances (49 Tsiligirides OP instances $\times$ 4 values of α^{Tsi} $\times$ 4 values of β^{Tsi}).

6.2 Lifting and valid inequalities

In this section we measure the impact of lifting constraints (7) and (8) into, respectively, (13) and (14). We also assess the impact of adding valid inequalities (15)–(18). For our benchmark, we use the continuous relaxation of the non-linear model as the baseline, and incrementally add lifting and valid inequalities. As discussed in Section 4.4, we calculate the solution of the continuous relaxation using the Franke-Wolfe algorithm on the convexified problem in which we replace the objective function with its negative logarithm. Once the optimal solution is returned, in terms of variables y and w, we compute the original objective function and use the corresponding value.

The six models we tested are:

- Base Model, with constraints (3)–(9).
- Lift, in which inequalities (13) and (14) replace, respectively, (7) and (8).
- The four models Lift + (15), Lift + (15)–(16), Lift + (15)–(17), and Lift + (15)–(18), in which we incrementally add valid inequalities.

Table 1 shows the results of our experiment. For a given instance and a given model out of the six presented above, we compute the gap (columns "Gap%") as

$$\text{Gap\%} = 100 \cdot \frac{\text{UB} - \text{BKS}}{\text{UB}}.$$

In this formula, UB is the upper bound obtained solving the model; note that we solve the continuous relaxation of the HOP and, hence, obtain upper bounds. BKS is the best known solution, obtained

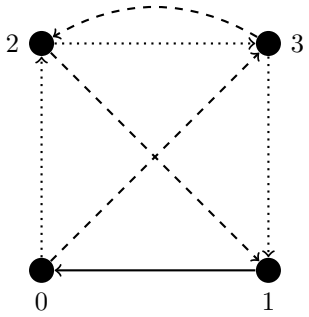


Figure 5: An example instance in which variables w_i take value 0 for customers visited in the optimal solution of the continuos relaxation of the HOP.

by taking the best integer feasible solution found in all experiments of Section 6 for the given instance. Columns "T (s)" report the runtime needed to solve the model, in seconds. The values in each row are averaged over all instances generated using the corresponding values of α^{Tsi} and β^{Tsi} (first two columns).

		Base Model		Lift		Lift + (15)		Lift + (15)–(16)		Lift + (15)–(17)		Lift + (15)–(18)	
α^{Tsi}	β^{Tsi}	Gap%	T (s)	Gap%	T (s)	Gap%	T (s)	Gap%	T (s)	Gap%	T (s)	Gap%	T (s)
0.1	2	28.42	0.1	25.13	0.1	14.99	0.7	14.99	0.7	14.99	0.8	5.59	6.0
0.1	3	34.61	0.0	31.61	0.0	16.70	0.7	16.70	0.7	16.70	0.7	8.23	5.7
0.1	4	40.72	0.1	38.27	0.0	20.62	0.7	20.62	0.7	20.62	0.7	12.95	6.1
0.1	5	41.38	0.1	39.06	0.1	20.86	0.7	20.86	0.7	20.86	0.7	12.96	6.4
0.2	2	41.08	0.2	38.21	0.2	16.42	0.8	16.42	0.8	16.42	0.8	8.00	6.4
0.2	3	48.19	0.2	46.12	0.3	19.58	0.8	19.58	0.8	19.58	0.9	12.62	6.5
0.2	4	52.80	0.2	51.10	0.2	22.43	0.8	22.43	0.8	22.43	0.9	16.40	6.8
0.2	5	56.55	0.2	55.15	0.2	25.91	0.8	25.91	0.8	25.91	0.9	20.49	6.7
0.3	2	48.81	0.2	46.36	0.2	16.55	0.8	16.55	0.8	16.55	0.9	9.29	5.8
0.3	3	57.89	0.3	56.10	0.3	21.52	0.9	21.52	0.9	21.52	1.0	15.30	8.0
0.3	4	63.00	0.2	61.68	0.2	27.17	0.9	27.17	0.9	27.17	1.0	21.91	6.7
0.3	5	67.12	0.2	66.12	0.2	31.21	0.9	31.21	1.0	31.21	1.0	26.90	8.3
0.4	2	57.11	0.2	55.09	0.2	18.45	0.8	18.45	0.9	18.45	0.9	11.78	5.8
0.4	3	65.30	0.3	63.92	0.3	25.70	0.8	25.70	0.9	25.70	1.0	19.86	6.8
0.4	4	68.54	0.3	67.43	0.2	28.33	0.9	28.33	0.9	28.33	1.0	23.25	6.6
0.4	5	72.46	0.3	71.61	0.2	33.51	0.8	33.51	0.9	33.51	1.0	29.09	7.0
Overall		52.75	0.2	50.81	0.2	22.50	0.8	22.50	0.8	22.50	0.9	15.91	6.6

Table 1: Impact of lifting and valid inequalities on the continuous relaxation of the problem, solved via the Frank-Wolfe algorithm. Columns "Gap%" report the optimality gap with respect to the best known primal solution. Columns "T (s)" report the time in seconds.

The results show that lifting the MTZ-like constraints has a modest but positive impact on the bound, and no large effect on the solution time which, in any case, remains below 1 second. Valid inequality (15) has a larger impact, driving the average gap down from 50.81 % to 22.50 %. Inequalities (16) and (17) do not significantly change the gaps, and they slightly increase the computation time. Finally, separating subtour elimination constraints (18) further reduced the average gap to 15.91 %. Because the separation procedure is relatively expensive, average solution times raise from 0.8 to 6.6 seconds. We judge, however, that the better solution quality obtained separating (18) justifies the corresponding increase in compute time.

The large impact of SECs (18) is not surprising, because it is well-known that they provide a tighter

continuous relaxation compared to the MTZ-like constraints in many other routing problems [4]. The role of valid inequality (15), on the other hand, deserves some explanation. Its main role is to prevent fractional solutions in which a customer i is visited, i.e., $y_i > 0$, $\sum_{(i,j)\in A} x_{ij} > 0$, and $\sum_{(j,i)\in A} x_{ji} > 0$, but the value of corresponding variable w_i is zero. Such solutions, in our experience, occur quite often. Figure 5 shows an example instance in which two of the three visited customers have $w_i = 0$ in the optimal fractional solution. The four vertices (depot 0 and three customers) are placed at the corners of the $[0, 1] \times [0, 1]$ square and distances are Euclidean. The profits are $p_1 = p_3 = 10$, and $p_2 = 20$. Values of λ are $\lambda_1 = 0$, $\lambda_2 = 0.05$, and $\lambda_3 = 0.1$. The time bound is $T = 10$.

The optimal fractional solution is $y_1 = y_2 = y_3 = 1$, $w_1 = 1$, $w_2 = w_3 = 0$. The value of the x variables is given in the figure: dotted arrows indicate a value of 0.23, dashed arrows of 0.77, and the solid arrow denotes $x_{10} = 1$. Inequality (9) forces $w_1 = 1$. However, neither (8) nor (9) are binding for w_2 and w_3. Take, for example, $i = 2$: constraints (8) are

$$w_2 \geq w_1 + \sqrt{2} - M(1 - x_{21}) = w_1 + \sqrt{2} - 10.41 = w_1 - 9$$
$$w_2 \geq w_3 + 1 - M(1 - x_{23}) = w_3 + 1 - 9.59 = w_3 - 8.59,$$

and constraint (9) simply states $w_2 \geq 0$. Therefore, solutions with $w_2 = 0$ are feasible for the continuous relaxation of the HOP and, given the large penalty carried in the objective function when $w_2 > 0$, they are rewarded.

Valid inequality (15), despite its simplicity, cuts away such solutions. For $i = 2$, for example, the inequality states that

$$w_2 \geq 1 \cdot x_{20} + \sqrt{2} \cdot x_{21} + 1 \cdot x_{23}.$$

Because $\sum_{(2,j)\in A} x_{2j} = 1$, this implies that w_2 is at least 1.

In light of the results presented in Table 1, in the rest of the experiments we will use the lifted version of the MTZ constraints, together with valid inequalities (15) and (18). We remark, however, that we can use constraints (18) only with LP or MIP solvers, because the non-linear solver Baron does not implement a callback mechanism to separate cuts.

6.3 Upper bounds

We next analyse the quality of the upper bounds proposed in Section 4. We present the results of our analysis in Table 2. Columns "Gap%" and "T (s)" have the same meaning as in Table 1; the meaning of column "Val%" is discussed below.

We consider the following upper bounds:

1. UB-Lin: bound obtained through the linear approximation of the objective function (see Section 4.2).
2. UB-PwLin. Similar to bound UB-Lin, but we use the piecewise-linear approximation of the objective function (see Section 4.3).
3. UB-Cont. The upper bound from the optimal solution of the continuous relaxation (see Section 4.4).
4. UB-SSR. The bound obtained removing the elementarity requirement, and using the corresponding state-space-relaxed (SSR) dynamic programming algorithm (see Section 5.1). In these experiments, we did not use the strengthening which forbids 2-cycles.
5. UB-SSR-2CE. Similar to bound UB-SSR, but using the strengthening which forbids 2-cycles.

Bounds UB-Lin, UB-PwLin, and UB-Cont all use the lifted version (13)–(14) of the MTZ-like inequalities, as well as valid inequalities (15) and (18). Constraints (18), which are the subtour elimination constraints, are separated on integer and fractional solutions alike.

Regarding the time limits used, we note the following:

		UB-Lin		UB-PwLin		UB-Cont		UB-SSR			UB-SSR-2CE		
α^{Tsi}	β^{Tsi}	Gap%	T (s)	Gap%	T (s)	Gap%	T (s)	Val%	Gap%	T (s)	Val%	Gap%	T (s)
0.10	2	9.97	29.3	6.95	22.5	5.59	5.9	28.57	34.58	2742.3	20.41	13.81	3020.8
0.10	3	16.29	33.2	11.70	30.3	8.23	5.4	28.57	29.83	2722.1	20.41	14.61	3035.8
0.10	4	21.37	35.9	15.17	34.2	12.95	5.9	28.57	28.79	2772.4	18.37	10.63	3015.6
0.10	5	23.31	36.7	17.41	35.8	12.96	6.2	28.57	31.86	2769.0	16.33	11.45	3046.9
0.20	2	21.65	36.5	15.99	33.0	8.00	6.2	22.45	35.30	2893.3	16.33	16.05	3067.6
0.20	3	31.66	41.6	24.05	37.7	12.62	6.1	20.41	30.94	2964.5	14.29	14.66	3126.5
0.20	4	37.87	44.8	29.49	43.1	16.40	6.8	24.49	25.87	2866.6	16.33	8.56	3094.0
0.20	5	40.43	43.3	32.24	42.1	20.49	6.5	24.49	26.48	2876.1	14.29	8.28	3112.4
0.30	2	31.03	44.7	23.16	40.5	9.29	5.9	22.45	29.69	2975.2	16.33	10.15	3075.1
0.30	3	41.80	45.6	33.18	43.9	15.30	7.7	20.41	30.74	2978.2	14.29	9.80	3125.2
0.30	4	47.50	46.0	39.03	45.5	21.91	6.2	24.49	25.40	2915.6	14.29	10.73	3117.5
0.30	5	53.69	48.3	45.18	47.8	26.90	7.4	20.41	23.72	2990.2	14.29	7.41	3105.1
0.40	2	40.15	46.1	32.29	44.1	11.78	5.6	22.45	33.27	2941.4	16.33	10.45	3080.6
0.40	3	52.43	49.3	43.90	48.7	19.86	6.4	22.45	28.00	2950.5	18.37	6.81	3090.8
0.40	4	55.92	48.7	47.34	48.4	23.25	6.3	26.53	21.47	2826.6	16.33	8.14	3102.0
0.40	5	60.25	49.1	52.45	48.7	29.09	7.0	22.45	23.10	2933.4	14.29	8.86	3126.0
Overall		36.60	42.4	29.36	40.4	15.91	6.3	24.23	28.73	2882.3	16.33	10.77	3083.9

Table 2: Comparison of the five upper bounds. Columns "Gap%" report the optimality gap with the best-known primal solution. Columns "T (s)" report the time in seconds. Columns "Val%" give the percentage of instances for which a valid upper bound was produced in dynamic-programming-based bounds.

- We first remark that, when computing bounds UB-Lin and UB-PwLin, if the solution is aborted due to reaching a given time limit, we can use the best upper bound returned by the solver as an upper bound for the HOP. We initially ran our experiments with two possible time limits of 1 hour and 60 seconds. We noticed that there was little difference in the quality of the bound, no matter if the MIP was given the larger or the smaller time limit. Therefore, we decided to use the 60-second time limit.
- During preliminary experiments, we noticed that bound UB-Cont was always computed in less than 30 seconds. We therefore imposed no time limit to compute this bound.
- Different from UB-Lin and UB-PwLin, the labelling algorithms provide valid upper bound only if they finish within the time limit (an early stop provides an invalid upper bound). However, these algorithms turn out to be slow and they do not complete most of the time, even when given a large time limit of 1 hour. Column "Val%" in Table 2 reports the percentage of instances for which the algorithm completed within the time limit and, thus, produced a valid upper bound. Then, in column "Gap%" the average is computed over these instances. Column "T (s)" reports the average time over all instances.

Table 2 shows that finding tight dual bounds for the HOP is hard. The best method, among those that always provide a valid bound, is to solve the continuous relaxation. The average gap obtained by UB-Cont is 15.91 % and the average solution time is 6.3 s.

Dynamic programming bounds are too computationally expensive: even if UB-SSR-2CE's gaps are smaller (10.77 %), this method only yields a valid bound for 16.33 % of the instances and takes a much longer time compared to UB-Cont. Part of the reason for the poor performance of this bound lies in the nature of the HOP, which is not compatible with popular speed-up techniques. For example, most state-of-the-art labelling algorithms for routing problems use bidirectional labelling, in which labels correspond to partial paths both from and to the depot (see, e.g., [32, 33]). In the HOP, however, we can only consider partial paths from a customer to the depot, because we must know the travel time after visiting each customer in the path in order to evaluate the cost of a label.

Regarding the bounds coming from the relaxation of the objective function, as expected UB-PwLin gives lower gaps than UB-Lin (objective (27) is tighter than objective (23)), while the runtimes are comparable. Both bounds, however, have larger gaps compared to UB-Cont. The reason is that the objective function used in these bounds does not penalise enough the risk associated with visiting time-bomb customers, because of the "local hazard" property (see Section 4.1). For example, in best-known integer solutions of the HOP, on average, only 5.49 % of the visited customers are time-bomb. This number increases to 27.46 % in the solutions produced by UB-PwLin. Analogously, in best-known integer solutions of the HOP, time-bomb customers appear in the tour, on average, at 96 % of its length, i.e., at the very end of the tour. By contrast, they appear at 70 % of the tour length, on average, in UB-PwLin's solutions.

6.4 Lower bounds

In this section we investigate the quality of primal feasible solutions found by the black-box non-linear solver and the labelling algorithm described in Section 5. For a given instance and a given lower bound LB, we report two values:

$$G_1\% = 100 \cdot \frac{\text{BKS} - \text{LB}}{\text{BKS}}, \quad G_2\% = 100 \cdot \frac{\text{UB}_\text{c} - \text{LB}}{\text{UB}_\text{c}},$$

where UB_c is the value of the upper bound UB-Cont. Thus, G_1 provides a gap with the best known solution and is influenced by the quality of the primal solutions found. G_2, on the other hand, provides a gap with a chosen upper bound.

		LB-NLModel			LB-NLCModel			LB-NLCModel*			LB-Label		
α^{Tsi}	β^{Tsi}	$\text{G}_1\%$	$\text{G}_2\%$	T (s)	$\text{G}_1\%$	$\text{G}_2\%$	T (s)	$\text{G}_1\%$	$\text{G}_2\%$	T (s)	$\text{G}_1\%$	$\text{G}_2\%$	T (s)
0.10	2	0.00	5.59	69.6	0.00	5.59	27.0	0.00	5.59	102.1	47.14	51.25	3177.7
0.10	3	0.00	8.23	103.0	0.00	8.23	24.7	0.00	8.23	129.5	45.47	51.34	3188.4
0.10	4	0.00	12.95	139.9	0.00	12.95	33.5	0.00	12.95	155.8	45.08	54.07	3182.6
0.10	5	0.00	12.96	216.4	0.00	12.96	26.5	0.00	12.96	192.8	44.23	53.11	3183.4
0.20	2	0.00	8.00	198.9	0.00	8.00	47.6	0.00	8.00	193.3	46.52	52.54	3185.7
0.20	3	1.21	13.74	387.4	0.00	12.62	26.8	0.00	12.62	245.7	46.66	55.35	3190.9
0.20	4	1.94	17.99	557.5	0.00	16.40	35.8	0.00	16.40	507.9	47.63	57.76	3184.7
0.20	5	9.03	28.11	1058.7	0.00	20.49	39.3	0.00	20.49	651.5	46.40	59.38	3182.8
0.30	2	0.00	9.29	282.1	0.00	9.29	9.7	0.00	9.29	105.8	50.18	56.77	3188.8
0.30	3	7.77	22.28	864.6	0.00	15.30	16.1	0.00	15.30	341.2	48.23	58.35	3188.0
0.30	4	14.37	33.89	1406.0	0.00	21.91	28.9	0.00	21.91	1056.2	47.45	61.38	3191.3
0.30	5	11.85	35.91	1643.5	0.00	26.90	240.8	0.30	27.11	1482.3	45.27	61.63	3190.0
0.40	2	5.60	16.97	844.0	0.00	11.78	12.8	0.00	11.78	389.7	50.93	58.80	3193.3
0.40	3	14.62	32.36	1614.1	0.00	19.86	14.8	0.00	19.86	1142.4	49.05	61.39	3191.4
0.40	4	25.42	43.32	1816.9	0.00	23.25	160.5	0.44	23.54	1600.8	48.57	61.88	3188.7
0.40	5	28.60	50.77	2086.4	0.00	29.09	382.7	2.35	30.78	2068.7	46.65	63.69	3191.6
Overall		7.53	22.02	830.6	0.00	15.91	70.5	0.19	16.05	647.9	47.22	57.42	3187.5

Table 3: Upper bounds comparison. Columns "$\text{G}_1\%$" report the optimality gap with the best-known primal solution. Columns "$\text{G}_2\%$" list the optimality gap with the upper bound from the continuous relaxation of the model. Columns "T (s)" report the time in seconds.

We consider the following four lower bounds, which we obtain running either the non-linear solver or the labelling algorithm, with a time limit of 1 hour:

1. LB-NLModel. It is the best feasible solution found by solver Baron, using the model with the lifted MTZ constraints and valid inequality (15). Recall that, because Baron does not support user callbacks and dynamic cut generation, we cannot use valid inequalities (18).

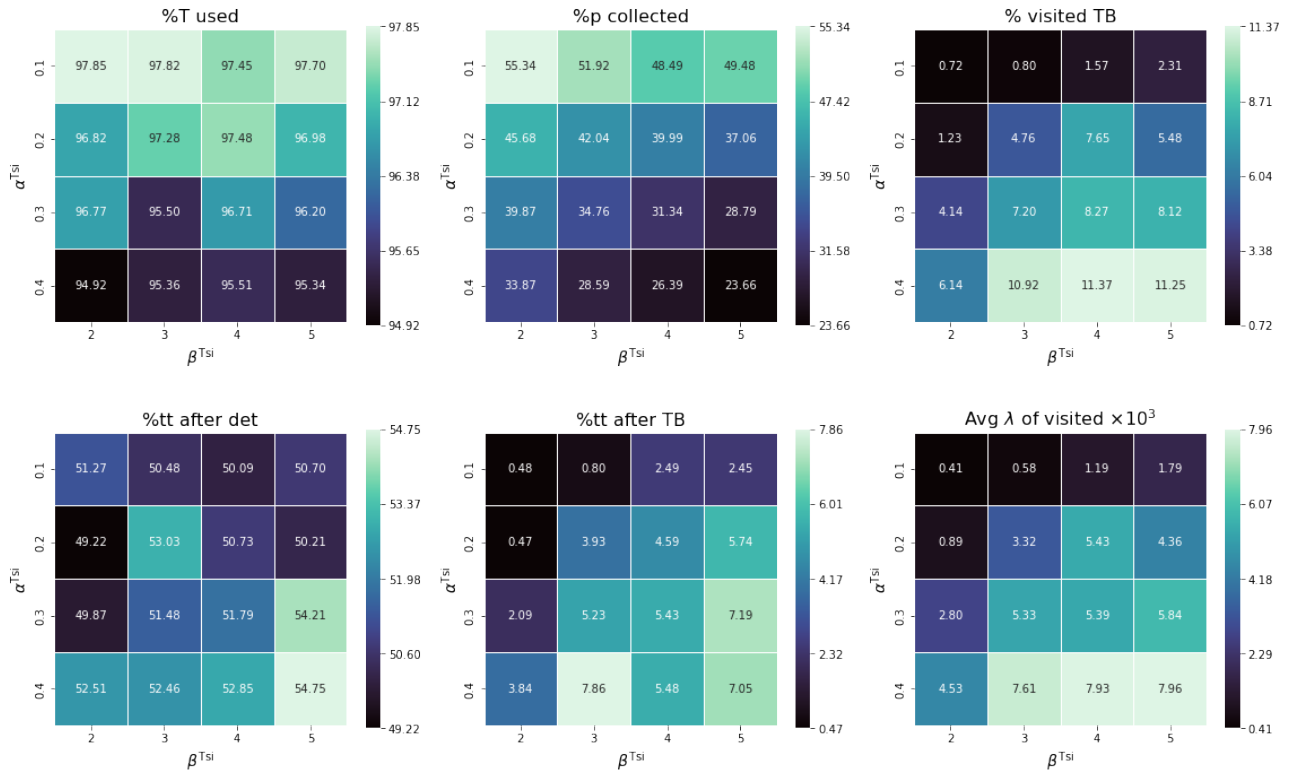


Figure 6: Change in six solution metrics, when instance generation parameters α^{Tsi} and β^{Tsi} change. Each value in the tables is an average over all instances which share the same generation parameters.

2. LB-NLCModel. Similar to bound LB-NLModel, but replacing the objective function with its logarithm, to ensure concavity. The value of the bound is then calculated recomputing the original objective value.

3. LN-NLCModel*. Similar to bound LB-NLCModel, but we disable the lifting of MTZ constraints and we do not add valid inequality (15). Lifting and valid inequality (15), in fact, were identified as effective by the experiments describes in Section 6.2, which used the continuous relaxation as their baseline model. By reporting the results of LN-NLCModel*, we provide evidence that they are also effective for the integer non-linear model.

4. LB-DP. Solution computed using the dynamic programming algorithm presented in Section 5. The algorithm returns the best non-dominated feasible solution found within the time limit.

We note that, while for LB-NLModel we could also report the solver gap (i.e., the gap between the best lower and upper bounds reported by Baron at the end of the computation), we cannot do the same for the two models which use the concave objective function. The reason is that Baron does not provide a way to access the variables' values of the solution which produces the dual bound and, therefore, we cannot recompute the original objective.

Table 3 reports the results of the experiments. Comparing the results under headers LB-NLModel and LB-NLCModel, we note that convexifying the objective function has a considerable impact on the solver's ability to find the optimum. Further comparing these results with those produced by LB-NLCModel* shows that convexification mainly reduces the gaps, while lifting and valid inequalities mainly reduce the compute time. The combination of these two techniques allows to find good quality solutions in short times. Finally, the labelling algorithm is not competitive with model-based approaches, neither in runtime nor in solution quality.

6.5 Analysis of the solutions

In this section, we explore the properties of high-quality solutions and how they vary depending on the instance generation parameters. For each instance, we use the best solution returned by the non-linear solver (LB-NLCModel presented in Section 6.4). Figure 6 shows how the value of six metrics changes when varying instance generation parameters α^{Tsi} and β^{Tsi}. The considered metrics are the following:

- **%T used.** This is the percentage of the time limit T used by the solution. High-quality solutions of the (purely deterministic) OP usually use almost 100% of the time limit [16, 34]. Intuitively, we could expect that the more time-bomb customers are in the instance and the higher is their profit, the lower this value will be. After visiting a time-bomb customer with a high enough profit, it is convenient to go back to the depot without visiting other customers to limit the probability of exploding. Thus, higher values of β^{Tsi} could be correlated with lower %T used. Furthermore, high-quality solutions will visit only a few time-bomb customers. If there are not enough deterministic customers to fill the remaining available time, it might be more convenient to return to the depot earlier. Therefore, we also expect that higher values of α^{Tsi} are correlated with lower %T used. While increasing α^{Tsi}, indeed, causes a decrease in this metric, the pattern is not clear with respect to changes in β^{Tsi}. Further inspection of the solutions revealed that high-quality tours visit time-bomb customers last and tend to use all the available time before. Therefore, an increase in the profits of time-bomb customers (higher values of β^{Tsi}) does not affect the percentage of the time limit used. Only the availability of enough deterministic customers to visit (i.e., lower α^{Tsi}) plays a role.
- **%p collected.** This is the percentage of profit collected by the vehicle over the total profit $\sum_{i \in V} p_i$ in the instance. When more customers are time-bomb (increasing α^{Tsi}) this indicator is lower because the majority of time-bomb customers are not visited and thus a lot of profit is not collected. Increasing β^{Tsi} also lowers this indicator. When the profit of time-bomb customers is higher, in fact, it constitutes a higher share of the total profits. Visiting few time-bomb customers then results in renouncing a higher percentage of the total profit.
- **% visited TB.** It is the percentage of time-bomb customers over all customers visited by a tour. For example, visiting one time-bomb and nine deterministic customers would result in a value of 10%. As we expect, a higher number (higher values of α^{Tsi}) and higher profits (higher values of β^{Tsi}) for time-bomb customers are both associated with more visits.
- **%tt after det** and **%tt after TB**. These metrics report the average value of $\tau_i := 100 \cdot \frac{w_i}{\sum_{(i,j) \in A} t_{ij} x_{ij}}$ computed over visited deterministic and time-bomb customers, respectively. Recall that w_i represents the travel time after customer $i \in V$ and that $\sum_{(i,j) \in A} t_{ij} x_{ij}$ is the tour travel time. Therefore, τ_i is the percentage of travel time after visiting i. Note that if we averaged over all visited customers (without distinguishing between deterministic and time-bomb) then we would expect this value to be close to 50%. We can use these two metrics to confirm our hypothesis that high quality solutions visit time-bomb customers towards the end of the tour: "%tt after TB" is much smaller than "%tt after det". We also note that higher time-bomb profits allow more travel time after visiting time-bomb customers: we can take a slightly higher risk if the reward is also higher.
- **Avg $\boldsymbol{\lambda}$ of visited $\boldsymbol{\times 10^3}$**. This indicator is the average value of λ_i (the parameter of the exponential random variable) over customers visited by the tour. Deterministic customers are assumed to have $\lambda_i = 0$. Because this indicator results in small values, to increase the readability of Figure 6, we multiply it by a factor of 10^3. We note that this indicator increases with both α^{Tsi} and β^{Tsi} and follows quite closely indicator "% visited TB".

7 Conclusions

In this paper we introduced the Hazardous Orienteering Problem, which models a variant of the classical OP in which the parcels picked up at some "time-bomb" customers have a probability of exploding which

depends on how long they travel on the vehicle and, if any parcel explodes, the entire collected profit is lost. The problem has interesting practical applications in cash transportation problems, routing of hazardous material and law enforcement with drones. The peculiar nature of the problem, due to the probability of explosion, makes its natural formulation more intriguing than the one of the classical OP, resulting in the maximisation of a non-concave objective function. This of course has implications on solution procedures: the complexity of the objective function makes the problem harder to solve than the standard OP. Thus, in this paper, we present multiple ways for deriving both upper bounds (based on integer, piece-wise integer and continuous relaxations, as well as on dynamic programming) and lower bounds (based on dynamic programming and black-box non-linear solvers). Our goal is to present and investigate multiple techniques, rather than focusing on one of them and fine-tune its performance. We thus perform an extensive computational analysis to compare the different approaches and determine the most promising ones. Our results show that, in terms of upper bounds, the continuous relaxation is the one leading to the best results. When solving the non-linear model with a black-box solver, transforming the objective function to make it concave, lifting existing inequalities and adding new valid ones has a large impact on the solver's performance. Indeed, a main takeaway of this paper is that modern non-linear solvers can achieve notable results if the user successfully strengthens their formulation. Finally, the analysis of solution features reveals that the solution structure depends on instance parameters, specifically, the number and profitability of time-bomb customers.

We envisage multiple directions for future research. On one side, one might focus on improving the approaches for determining both upper and lower bounds, for example, through stronger relaxations. On the other side, problem generalisations are also of interests, for example the case with multiple vehicles or a multi-objective extensions in which profit collected, travel time and probability of explosion define a three-dimensional Pareto frontier. Another possible extension involves using different probability distributions, tailored for specific application area. One could even introduce more complex modelling choices in which the probability of a catastrophic event does not depend uniquely on the travel time, but also on the specific arcs used after visiting the customer. For example, some roads are more suitable than others for armed robberies; bumpy roads will put lithium ion batteries under more mechanical stress, etc.

Acknowledgements

The work of Alberto Santini has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skodowska Curie grant agreement number 945380.

References

[1] Murat Afsar and Nacima Labadie. 'Team Orienteering Problem with Decreasing Profits'. In: *Electronic Notes in Discrete Mathematics* 41 (2013), pp. 285–293. DOI: 10.1016/j.endm.2013.05.104.

[2] Enrico Angelelli, Claudia Archetti, Carlo Filippi and Michele Vindigni. 'A dynamic and probabilistic orienteering problem'. In: *Computers & Operations Research* 136 (2021). DOI: 10.1016/j.cor.2021.105454.

[3] Enrico Angelelli, Claudia Archetti, Carlo Filippi and Michele Vindigni. 'The probabilistic orienteering problem'. In: *Computers & Operations Research* 81 (2017), pp. 269–281. DOI: 10.1016/j.cor.2016.12.025.

[4] David Applegate, Robert Bixby, Vaek Chvátal and William Cook. *The Traveling Salesman Problem. A computational study*. Vol. 40. Princeton Series in Applied Mathematics. De Gruyter, 2007. DOI: 10.1515/9781400841103.

[5] Egon Balas. 'The prize collecting traveling salesman problem'. In: *Networks* 19 (6 1989), pp. 621–636. DOI: 10.1002/net.3230190602.

[6] Tolga Bekta and Luis Gouveia. 'Requiem for the Miller–Tucker–Zemlin subtour elimination constraints?' In: *European Journal of Operational Research* 236.3 (2014), pp. 820–832.

[7] Ann Campbell, Michel Gendreau and Barrett Thomas. 'The orienteering problem with stochastic travel and service times'. In: *Annals of Operations Research* 186 (2011), pp. 61–81. DOI: doi.org/10.1007/s10479-011-0895-2.

[8] Jianqiang Cheng, Erick Delage and Abdel Lisser. 'Distributionally robust stochastic knapsack problem'. In: *SIAM Journal on Optimization* 24 (3 2014), pp. 1485–1506. DOI: 10.1137/130915315.

[9] Andrea Chiussi, Christos Orlis, Roberto Roberti and Wout Dullaert. 'ATM cash replenishment under varying population coverage requirements'. In: *Journal of the Operational Research Society* (2021). DOI: 10.1080/01605682.2020.1866443.

[10] Nicos Christofides, Aristide Mingozzi and Paolo Toth. 'State-space relaxation procedures for the computation of bounds to routing problems'. In: *Networks* 11.2 (1981), pp. 145–164. DOI: 10.1002/net.3230110207.

[11] Mauro Dell'Amico, Francesco Maffioli and Peter Värbrand. 'On Prize-collecting Tours and the Asymmetric Travelling Salesman Problem'. In: *International Transactions in Operational Research* 2 (3 1995), pp. 297–308. DOI: 10.1111/j.1475-3995.1995.tb00023.x.

[12] Martin Desrochers and Gilbert Laporte. 'Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints'. In: *Operations Research Letters* 10.1 (1991), pp. 27–36.

[13] Regina Elandt-Johnson and Norman Johnson. *Survival models and data analysis*. Wiley, 1980. ISBN: 978-0-471-03174-1.

[14] Lanah Evers, Kristiaan Glorie, Suzanne van der Ster, Ana Isabel Barros and Herman Monsuur. 'A two-stage approach to the orienteering problem with stochastic weights'. In: *Computers & Operations Research* 43 (2014), pp. 248–260. DOI: 10.1016/j.cor.2013.09.011.

[15] Michael Farrington. 'Safety of lithium batteries in transportation'. In: *Journal of power sources* 96.1 (2001), pp. 260–265. DOI: 10.1016/s0378-7753(01)00565-1.

[16] Matteo Fischetti, Juan Jose Salazar Gonzalez and Paolo Toth. 'Solving the orienteering problem through branch-and-cut'. In: *INFORMS Journal on Computing* 10.2 (1998), pp. 133–148. DOI: 10.1287/ijoc.10.2.133.

[17] Marguerite Frank and Philip Wolfe. 'An algorithm for quadratic programming'. In: *Naval Research Logistics* 3 (1-2 1956), pp. 95–110. DOI: 10.1002/nav.3800030109.

[18] Georg Fröhlich, Karl Doerner and Margaretha Gansterer. 'Secure and efficient routing on nodes, edges, and arcs of simple-graphs and of multi-graphs'. In: *Networks* 76.4 (2020), pp. 431–450. DOI: 10.1002/net.21993.

[19] Bruce Golden, Larry Levy and Rakesh Vohra. 'The orienteering problem'. In: *Naval Research Logistics* 34 (3 1987), pp. 307–318.

[20] Maaike Hoogeboom and Wout Dullaert. 'Vehicle routing with arrival time diversification'. In: *European Journal of Operational Research* 275.1 (2019), pp. 93–107. DOI: 10.1016/j.ejor.2018.11.020.

[21] David Houck, Jean-Claude Picard, Maurice Queyranne and Ramakrishna Vemuganti. 'The travelling salesman problem as a constrained shortest path problem: theory and computational experiments'. In: *Opsearch* 17 (1980), pp. 93–109.

[22] Taylan lhan, Seyed Iravani and Mark Daskin. 'The orienteering problem with stochastic profits'. In: *IIE Transactions* 40.4 (2008), pp. 406–421. DOI: 10.1080/07408170701592481.

[23] Diego Lisbona and Timothy Snee. 'A review of hazards associated with primary lithium and lithium-ion batteries'. In: *Process Safety and Environmental Protection* 89.6 (2011), pp. 434–442. DOI: 10.1016/j.psep.2011.06.022.

[24] Silvano Martello and Gilbert Laporte. 'The selective travelling salesman problem'. In: *Discrete and Applied Mathematics* 26 (2 1990), pp. 193–207. DOI: 10.1016/0166-218X(90)90100-Q.

[25] Clair E Miller, Albert W Tucker and Richard A Zemlin. 'Integer programming formulation of traveling salesman problems'. In: *Journal of the ACM (JACM)* 7.4 (1960), pp. 326–329.

[26] Michele Monaci, Cyara Pike-Burke and Alberto Santini. 'Exact algorithms for the 0-1 time-bomb knapsack problem'. In: *Computers & Operations Research* To appear (2022).

[27] Sandra Ulrich Ngueveu, Christian Prins and Roberto Wolfler Calvo. 'Lower and upper bounds for the m-peripatetic vehicle routing problem'. In: *4OR* 8.4 (2010), pp. 387–406. DOI: 10.1007/s10288-010-0148-2.

[28] Christos Orlis, Nicola Bianchessi, Roberto Roberti and Wout Dullaert. 'The Team Orienteering Problem with Overlaps: An Application in Cash Logistics'. In: *Transportation Science* 54.2 (2020), pp. 470–487. DOI: 10.1287/trsc.2019.0923.

[29] Christos Orlis, Demetrio Laganà, Wout Dullaert and Daniele Vigo. 'Distribution with Quality of Service Considerations: The Capacitated Routing Problem with Profits and Service Level Requirements'. In: *Omega* 93 (2020). DOI: 10.1016/j.omega.2019.02.003.

[30] Vassilis Papapanagiotou, Roberto Montemanni and Luca Maria Gambardella. 'Objective function evaluation methods for the orienteering problem with stochastic travel and service times'. In: *Journal of Applied Operational Research* 6.1 (2014), pp. 16–29. ISSN: 1927-0089.

[31] Marvin Rausand, Anne Barros and Arnljot Hoyland. *System Reliability Theory: Models, Statistical Methods, and Applications*. 3rd. Wiley, 2020. ISBN: 978-1-119-37352-0.

[32] Giovanni Righini and Matteo Salani. 'Decremental state space relaxation strategies and initialization heuristics for solving the Orienteering Problem with Time Windows with dynamic programming'. In: *Computers & Operations Research* 36 (4 2009), pp. 1191–1203. DOI: 10.1016/j.cor.2008.01.003.

[33] Giovanni Righini and Matteo Salani. *Dynamic programming for the orienteering problem with time windows*. Tech. rep. 3. Università degli Studi di Milano, 2006. URL: https://air.unimi.it/handle/2434/6449.

[34] Alberto Santini. 'An Adaptive Large Neighbourhood Search algorithm for the Orienteering Problem'. In: *Expert Systems with Applications* 123 (2019), pp. 154–167. DOI: 10.1016/j.eswa.2018.12.050.

[35] Alberto Santini. 'Repository hazardous-orienteering-problem'. In: (2022). DOI: 10.5281/zenodo.7034318. URL: https://github.com/alberto-santini/hazardous-orienteering-problem.

[36] Hanif Sherali, Laora Brizendine, Theodore Glickman and Shivaram Subramanian. 'Low Probability High Consequence Considerations in Routing Hazardous Material Shipments'. In: *Transportation Science* 31.3 (1997). DOI: 10.1287/trsc.31.3.237.

[37] Luca Talarico, Kenneth Sörensen and Johan Springael. 'Metaheuristics for the risk-constrained cash-in-transit vehicle routing problem'. In: *European Journal of Operational Research* 244.2 (2015), pp. 457–470.

[38] Theodoros Tsiligirides. 'Heuristic Methods Applied to Orienteering'. In: *Journal of the Operational Research Society* 35 (1984), pp. 797–809. DOI: 10.1057/jors.1984.162.

[39] Johannes Kornelis Vrijling, Wim Van Hengel and Rob Houben. 'A framework for risk evaluation'. In: *Journal of Hazardous Materials* 43 (1995), pp. 245–261. DOI: 10.1016/0304-3894(95)91197-V.

[40] Shu Zhang, Jeffrey W Ohlmann and Barrett W Thomas. 'Dynamic orienteering on a network of queues'. In: *Transportation Science* 52.3 (2018), pp. 691–706. DOI: 10.1287/trsc.2017.0761.