Transportation Research Part B 185 (2024) 102984

Contents lists available at ScienceDirect

Transportation Research Part B

journal homepage: www.elsevier.com/locate/trb

The Capacitated Team Orienteering Problem: An online optimization framework with predictions of unknown accuracy

Davood Shiri [a,*], Vahid Akbari [b], Ali Hassanzadeh [c]

[a] *Sheffield University Management School, University of Sheffield, Conduit Road, Sheffield, S10 1FL, United Kingdom*
[b] *Nottingham University Business School, University of Nottingham, Jubilee Campus, Nottingham NG8 1BB, United Kingdom*
[c] *Alliance Manchester Business School, University of Manchester, Booth St W, Manchester M15 6PB, United Kingdom*

ARTICLE INFO

Keywords:
Online vehicle routing
Competitive ratio
Capacitated Team Orienteering Problem
Uncertainty
Online heuristics

ABSTRACT

The Capacitated Team Orienteering Problem (CTOP) is a challenging combinatorial optimization problem, wherein a fleet of vehicles traverses multiple locations, each with distinct prizes, demand weights, and service times. The primary objective is to determine optimal routes for the vehicles that collectively accumulate the highest total prize within capacity and time constraints. The CTOP finds applications across various domains such as disaster response, maintenance, marketing, tourism, and surveillance, where coordinated teams are required to efficiently explore and gather prizes from different sites. The complexity of this problem is further compounded by uncertainties in predicting specific attributes of each location, making it hard to plan routes accurately in advance. In numerous scenarios in practice, subjective predictions for these parameters may exist, but their precise values remain unknown until a location is visited by one of the vehicles. Given the unpredictable nature of these parameters, there is a pressing need for innovative online optimization strategies that can adapt to new information, ensuring the strategic allocation of resources and route planning within set constraints. To address this challenging online optimization problem, we offer a detailed analysis through the lens of theoretical and empirical competitive ratios. We derive an exact tight upper bound on the competitive ratio of online algorithms, and we introduce three novel online algorithms, with two of them achieving optimal competitive ratios. The third algorithm is a polynomial time approximation-based online algorithm with a competitive ratio of $\frac{1}{3.53}$ times the tight upper bound. To evaluate our algorithms, we measure their empirical competitive ratios on randomly generated instances as well as instances from the literature. Our empirical analysis demonstrates the effectiveness of our solutions across a diverse range of simulation scenarios.

1. Introduction

The Orienteering Problem (OP) originates from a hunting game with the goal of maximizing the total collected prize by determining the sequence of visits to a subset of available locations (Tsiligirides, 1984). Its original version is an outdoor sport played in mountainous and unfamiliar terrain, where individuals with strong navigational skills compete. In recent years, this problem and its variants have been studied and implemented in various applications including logistics, transportation and supply chain management. In a logistics setting, a vehicle visits a subset of available demand nodes in an attempt to maximize its total collected prize. As an immediate follow up to the original OP, the Team Orienteering Problem (TOP) formulates the same dynamic

* Corresponding author.
E-mail addresses: d.shiri@sheffield.ac.uk (D. Shiri), vahid.akbari@nottingham.ac.uk (V. Akbari), ali.h@manchester.ac.uk (A. Hassanzadeh).

https://doi.org/10.1016/j.trb.2024.102984
Received 1 September 2023; Received in revised form 6 May 2024; Accepted 24 May 2024
Available online 3 June 2024
0191-2615/© 2024 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

between multiple vehicles with the added constraint that a demand node visited by one vehicle, may not be visited by any other vehicle (Chao et al., 1996b). In the Capacitated Team Orienteering Problem (CTOP), introduced by Archetti et al. (2009), there is a weight (demand) on each item collected from a demand node and a total allowable weight as capacity on each vehicle. All the above mentioned problems are variants of the renowned Traveling Salesman Problem (TSP), and closely related to the general family of Vehicle Routing Problems (VRP).

We investigate the CTOP within an online optimization framework. We assume a travel time between locations, service time for the collection from a demand node, and an overall time budget for each vehicle. Our online optimization framework unfolds in real-time, revealing the weight, service time, and prize associated with each demand node only upon its visit by a vehicle, operating with limited foreknowledge of these parameters. In the deterministic version of the CTOP, given a set of demand nodes with known weights, service times, and prizes, and a set of vehicles with limited capacity and time, the goal is to (i) select a subset of demand nodes for which the total collected prize is maximized, and (ii) to determine the allocation of the demand nodes to the vehicles and their optimal routes under capacity and time constraints (Tarantilis et al., 2013).

In our study, we assume no prior knowledge of the exact value, the probability distribution, or any uncertainty set for the uncertain parameters, and the values are realized over time upon a visit by a vehicle, and only a point estimate (i.e., prediction) is known ex ante. We remark that we do not assume any guarantee on the dispersion or accuracy of the point predictions due to the inherently limited available information. The source of these predictions may vary depending on the context, from a qualitative forecasting method (e.g., *Delphi* method, expert opinion) or a quantitative method (e.g., an extrapolation model based on historical observations).

On an important note and to elaborate more on the process of solving the online CTOP, a subset of the parameters are assumed to be uncertain at the beginning while the online optimization algorithm learns the exact values gradually upon visiting new locations. After arriving at a new location, a new set of exact values are realized and the decisions are updated accordingly. We remark that within our online optimization framework, the two main decisions at any new location which are being made dynamically as we traverse through the network are (i) whether the vehicle serves the location and collects the prize or bypasses and moves on to other locations, and (ii) what route the vehicle takes after leaving the current location. Note that both of these decisions are impacted by the newly observed parameters at the new location.

In this paper, we model the uncertainty in three pivotal parameters associated with each demand node namely, the prize to be collected, the demand weight, and the service (pick-up) time. The intricacies of this type of modeling uncertainty fits quite well within various real-world applications of the CTOP including but not limited to disaster operations management, maintenance operations, sales and marketing, geological exploration, as well as security and police operations. Common across these applications is the use of multiple vehicles to collect prizes from (or service) specific locations. The key operational challenge is the real-time revelation of crucial information upon arrival at these nodes, necessitating immediate strategic adjustments in resource allocation and routing. This dynamic adaptation is vital, whether in coordinating disaster relief, scheduling industrial maintenance, mapping exploration routes, or optimizing security patrols, where the accuracy of prior predictions cannot be guaranteed. We refer to this problem as the *Online Capacitated Team Orienteering Problem* (OCTOP) hereafter.

Our paper is organized as follows. We survey the related literature in Section 2. We introduce the main problem with mathematical notation in Section 3. In Section 4, we prove a tight upper bound on our competitive ratio metric. We introduce and explain our online algorithms in Section 5. We present our computational experiments in Section 6 and conclude the paper in Section 7.

2. Literature review

We review the related literature of the CTOP and its variants in three sub-sections: (i) exact models and heuristic procedures for addressing the deterministic (offline) version, (ii) stochastic and robust optimization approaches, and (iii) related studies on online optimization framework. We highlight our key contributions in this study in the final sub-section.

2.1. Related deterministic literature

The OP was first studied in Tsiligirides (1984) and was investigated further in many papers, e.g., Chao et al. (1996a), Kobeaga et al. (2023). Different extensions of the OP have been investigated in the extant literature, including the problems where the vehicle has a limited capacity (Bock and Sanità, 2015), problems with time windows where visits are constrained by specific time intervals (Vansteenwegen et al., 2009), problems with split delivery options in which a demand node can be serviced multiple times such that each time a portion of the demand of that demand node is satisfied (Wang et al., 2014), arc routing orienteering problems (Riera-Ledesma and Salazar-González, 2017), generalizations with multiple routes/vehicles (Ruiz-Meza et al., 2021; Tarantilis et al., 2013), as well as variants which investigate multi-period planning horizons (Kotiloglu et al., 2017). For comprehensive reviews on OP variants, see Vansteenwegen et al. (2011) and Gunawan et al. (2016).

A generalization of the OP with multiple vehicles is referred to as the TOP which was first introduced as the Multiple Tour Maximum Collection Problem by Butt and Cavalier (1994). In the TOP the multiple vehicles seek to find a set of vehicle routes that maximize the total collected profit while respecting the travel time limit for each vehicle (Chao et al., 1996b; Tang and Miller-Hooks, 2005b; Boussier et al., 2007; Yu et al., 2022a).

Transitioning to a more complex variant, the CTOP introduces vehicle capacities and node weights into the problem. Archetti et al. (2009) first introduced the CTOP and proposed an exact Branch-and-Price method, building upon the formulation by Boussier

Table 1
Overview of studies on variants of orienteering problem in the presence of uncertainty, their problem characteristics, and modeling frameworks. Note that following abbreviations are used under the modeling framework in the table; SP: Stochastic Programming, DDSP: Data-driven Stochastic Programming, RO: Robust Optimization, OO: Online Optimization.

Article(s)	Problem	# Vehicles	Problem type	Capacity constraint?	Skipping a visited demand node as a decision?	Modeling framework
Teng et al. (2004), Tang and Miller-Hooks (2005a), Ilhan et al. (2008), Campbell et al. (2011), Varakantham and Kumar (2013), Papapanagiotou et al. (2014, 2015), Evers et al. (2014b), Verbeeck et al. (2016), Angelelli et al. (2017), Varakantham et al. (2018), Liao and Zheng (2018), Chou et al. (2021), Avraham and Raviv (2023)	OP	Single	Static			SP
Gupta et al. (2015), Dolinskaya et al. (2018), Bian and Liu (2018), Angelelli et al. (2021)	OP	Single	Dynamic			DDSP
Zhang et al. (2014), Zhang et al. (2018)	OP	Single	Dynamic		✓	DDSP
Evers et al. (2014a), Shi et al. (2023)	OP	Single	Dynamic			RO
Demange et al. (2021)	OP	Single	Dynamic			OO
Panadero et al. (2020),Panadero et al. (2023), Song et al. (2020)	TOP	Multiple	Static			SP
Karunakaran et al. (2019), Reyes-Rubiano et al. (2020), Juan et al. (2020)	TOP	Multiple	Dynamic			DDSP
Balcik and Yanıkoğlu (2020), Yu et al. (2022b), Zhang et al. (2023)	TOP	Multiple	Static			RO
This paper	**CTOP**	**Multiple**	**Dynamic**	✓	✓	**OO**

et al. (2007) for the TOP. In their solution methodologies, they employ column generation and dynamic programming techniques. Additionally, Archetti et al. (2009) proposed Tabu Search and Variable Neighborhood Search (VNS) algorithms. In subsequent works, Archetti et al. (2013) presented an exact Branch-and-Price formulation for the CTOP. For other heuristic approaches for solving the CTOP see Tarantilis et al. (2013), Luo et al. (2013), Ben-Said et al. (2019), Gunawan et al. (2021).

In the next two subsections, we review related work that (i) deal with uncertainty in the parameters with stochastic and robust optimization models, and (ii) develop an online optimization framework. Table 1 outlines key studies of OP variants, highlighting their approaches to uncertainty, problem specifics, and modeling techniques.

2.2. Related stochastic and robust optimization literature

Until now, our review has focused on literature that primarily explores deterministic models. There has been a growing interest in recent years in studying the OP variants in the presence of uncertainty, in a subset of input parameters, both from the stochastic programming and robust optimization perspectives. From a stochastic programming lens, uncertain parameters are modeled as random variables with known probability distributions, while in the robust optimization framework, those uncertain parameters are modeled as random variables given a known uncertainty set, due to lack of information on the exact underlying distribution.

Stochastic OP variants have been widely studied in the literature in a static setting, i.e., where decisions are not updated, using various approaches such as Two-stage Stochastic Programming (Teng et al., 2004; Evers et al., 2014b), Branch-and-Cut formulation (Tang and Miller-Hooks, 2005a; Angelelli et al., 2017), Monte Carlo sampling-based procedures (Papapanagiotou et al., 2014, 2015), mixed-integer programming formulations (Varakantham and Kumar, 2013; Varakantham et al., 2018), as well as heuristics (Ilhan et al., 2008; Campbell et al., 2011; Verbeeck et al., 2016; Liao and Zheng, 2018; Chou et al., 2021; Avraham and Raviv, 2023).

There is also a stream of research on the dynamic stochastic variants of the OP. Gupta et al. (2015) focused on a stochastic OP with uncertain service times. They explored both static and dynamic versions of the problem and developed constant factor approximation algorithms for both versions which can be implemented in polynomial time. For heuristic algorithms on the same problem, see Dolinskaya et al. (2018), Bian and Liu (2018), Angelelli et al. (2021), Zhang et al. (2014, 2018). Notably among these works, Zhang et al. (2014) addressed a dynamic OP variant that incorporates time windows and stochastic service times. In their study, while the route for each vehicle is pre-determined, the vehicle is able to skip a demand node after visiting it and realizing its service time. The authors devised a Variable Neighborhood Search heuristic for solving the problem. Building upon their previous work, Zhang et al. (2018) extended the problem by considering entirely dynamic routing decisions. They presented an approximate Dynamic Programming algorithm as an approach to solve this more complex problem, taking into account the dynamic nature of the routing decisions.

The stochastic TOP has also been investigated in both static and dynamic streams. On the static side, Two-stage Stochastic Programming (Song et al., 2020) as well as heuristic approaches (Panadero et al., 2020, 2023) have been applied. On the dynamic stream, the proposed approaches mostly rely on heuristics (Karunakaran et al., 2019; Juan et al., 2020; Reyes-Rubiano et al., 2020; Best and Hollinger, 2019). Note that these studies do not formulate the decision for *bypassing* a node by a vehicle, should that node deemed not-profitable-enough upon the visit.

While the majority of research on Orienteering Problem (OP) variants under uncertainty presumes full knowledge of the probability distributions of uncertain parameters, there are several studies that developed *robust optimization* techniques via worst-case analysis. Among these papers, Evers et al. (2014a), Shi et al. (2023) focus on the OP variants whereas (Balcik and Yanıkoğlu, 2020; Yu et al., 2022b; Zhang et al., 2023) focus on the TOP variants.

Focusing on scholarly works from a stochastic or robust optimization viewpoint, while there have been studies on the OP and its extension with multiple vehicles, the TOP, the literature is quite limited in the case of the CTOP, as there are no studies on the CTOP dealing with uncertainty in the parameters, which warrants a comprehensive investigation of this problem under uncertainty. Furthermore, to the best of our knowledge, there remains a significant gap in the literature regarding efficient solution methodologies for the CTOP within an online optimization framework.

2.3. Related online optimization literature

We review related online optimization literature with competitive ratio as their main metric. Over the years, numerous studies have explored online variants of the TSP, where demands (nodes) are revealed sequentially to the decision maker, rather than being known a priori. In our study of the OCTOP, we integrate the concept of *online demands* into our framework. A differentiator of our approach compared to related studies is twofold: (i) we assume uncertainty in weight, service time, and prize of each demand node, and (ii) these information are disclosed to the vehicles only upon their arrival at the respective nodes. This approach enables us to cover a broader spectrum of real-world situations where the critical attributes of locations remain unknown until physically visited by a team.

The TSP with sequential demands was first studied by Ausiello et al. (2001) through competitive analysis, establishing that no online algorithm could outperform the optimal offline solution by a factor of less than two in the worst-case scenario (i.e., competitive ratio is bounded by 2). Following this foundational work, research expanded to include various adaptations of the online TSP, exploring strategies such as engaging with fair adversaries (Blom et al., 2001), introducing advance notice of demands (Jaillet and Wagner, 2006), resource augmentation and considering extra resources for the online algorithm compared to the offline version (Jaillet and Wagner, 2008), and incorporating service flexibility and rejection capabilities (Jaillet and Lu, 2014, 2011). From a practical point of view, Shirdel and Abdolhosseinzadeh (2018) developed an online heuristic for the TSP with symmetric travel times and triangle inequality, based on Simulated Annealing.

There is a multitude of relevant online problems in the literature. Several online variants of Traveling Repairman Problem (otherwise known as Minimum Latency Problem) have been investigated in Akbari and Shiri (2022, 2021), Zhang et al. (2019), Irani et al. (2004), Krumke et al. (2003). The primary goal in these studies is to minimize the summation of completion times of all demands. Christman et al. (2018) and Feuerstein and Stougie (2001) have analyzed different variants of the dial-a-ride problem with sequential demands from theoretical competitive analysis viewpoints. Several papers including (Voccia et al., 2019; Bertsimas et al., 2019; Van Heeswijk et al., 2019) study the pick-up and delivery problems with sequential demands using empirical analysis. Despite similarity, the underlying assumptions and definitions in these problems distinctly set them apart from the OCTOP.

Another closely related body of research to the OCTOP includes studies on the online prize-collecting TSP and the online quota TSP with sequential demands . Ausiello et al. (2008) investigated the prize-collecting variant through a competitive analysis lens, where each node carries specific weights and penalties. The objective is to amass a predetermined aggregate weight while minimizing the combined travel costs and penalties for bypassed nodes. They establish that no online algorithm can surpass a competitive ratio of two for this problem, and they introduce an algorithm with competitive ratio of $\frac{7}{3}$. Since the quota TSP is a special case of the prize-collecting TSP, with node penalties set to zero, these findings are equally applicable to it. Yu et al. (2014) studied the quota TSP with sequential demands and provided several online algorithms with optimal competitive ratios under specific conditions. In a more recent work, Demange et al. (2021) studied a variant of the OP where demands emerge sequentially across different locales within a territory, each associated with a serviceable time window and a prize indicating its significance. The authors investigate the problem under simplifying assumptions including a fixed minimum interval between consecutive demands and time window lengths equal to the territory's diameter. They offer insights into the competitive ratio of the online algorithms addressing this version of the problem.

Problem Instances	Deterministic CTOP	Stochastic CTOP	Robust CTOP	Online CTOP (OCTOP)
Parameter Description	Deterministic parameters known a priori	Uncertainty in some parameters with known probability distribution	Uncertainty in some parameters with only an uncertainty set known	Uncertain parameters realized over time with limited point estimates known a priori
Prescriptive Approach	Exact mathematical modeling approaches and heuristic algorithms	Scenario-based stochastic optimization models (e.g., SAA)	Robust optimization solution methods (e.g., Distributionally Robust Optimization formulation)	Novel online algorithms, e.g., OTO, IO, AIO

Fig. 1. Different versions of the CTOP based on the uncertainty of parameters and availability of information.

2.3.1. Related online optimization problems with predictions

Recent research within the computer science community has seen a growing interest in TSP variants that incorporate predictions. This body of work adopts a *predictive-model-agnostic* approach, where online algorithms utilize predictions regardless of their origin, acknowledging that these predictions may carry errors of unknown magnitude. The main focus of this line of research is to develop online algorithms that perform close to optimal when the predictions are accurate, while also maintaining theoretical guarantees in terms of worst-case competitive ratios, even with less accurate predictions. Moreover, the goal is to develop algorithms with the versatility to handle varying degrees of predictive accuracy, thereby enhancing their applicability and resilience in real-world scenarios where the accuracy of predictions cannot be guaranteed.

The overarching objective in these studies is to design an online algorithm with three desirable properties: (i) **consistency**, ensuring the algorithm delivers solutions that closely mirror the offline optimal solution under ideal prediction conditions; (ii) **robustness**, guaranteeing the algorithm's output remains within a limit defined by multiplying a positive constant with the offline optimal solution, regardless of the prediction's accuracy; and (iii) **smoothness**, ensuring that the algorithm's solution quality diminishes in a controlled manner as prediction errors increase, maintaining a performance level limited by a function of prediction accuracy times the offline optimal solution.

Research in this domain has predominantly been theoretical, focusing on competitive ratios. Hu et al. (2022) investigated the online TSP with predictions with various prediction models and proposed algorithms that enhance existing results and extend these improvements to the online dial-a-ride problem. Gouleakis et al. (2023) tackled the online TSP with predictions where the input graph is restricted to a line. Chawla and Christou (2023) studied the online TSP with predictions and time-windows constraints, and Bampis et al. (2023) studied the online TSP with predictions on rings, trees, and general metric space.

2.4. Our contributions

Our key contributions in this study are as follows. (i) We introduce a new variant of the CTOP, namely the OCTOP, which models various real-life scenarios, filling a notable gap in the literature. (ii) Building upon the established frameworks in related literature, we adopt the *competitive ratio* metric to study the OCTOP, and we propose a tight upper bound on the competitive ratio for online optimization algorithms. We extend our theoretical analysis and analyze the competitive ratio of online algorithms from resource augmentation, as well as consistency, robustness, and smoothness perspectives. (iii) We develop three novel online algorithms, namely, *One-Time Optimization* (OTO), *Iterative Optimization* (IO), and *Approximation-based Iterative Optimization* (AIO) algorithms. We prove optimal competitive ratios for OTO and IO algorithms. As for AIO, despite exhibiting a comparatively lower competitive ratio, it runs significantly faster than the IO counterpart. (iv) Through extensive empirical analysis, we benchmark our algorithms against optimal offline solutions, offering insights into their effectiveness across various settings, particularly noting the IO algorithm's robust performance with varying prediction accuracies. We conduct our experiments on both randomly generated instances as well as instances from the literature. Fig. 1 shows the distinction between different problem instances discussed in the Literature Review section and the research gap we are targeting in our paper (i.e., under online CTOP).

3. The problem

Let $G = (V_0, E)$ be a complete directed graph. There are N demand nodes indexed from 1 to N and $V = \{1, 2, \dots, N\}$ is the set of demand nodes. The online uncertainty is defined as follows. Each demand node $i \in V$ has a demand to be satisfied which is associated with three positive parameters, namely, weight, service time, and prize denoted by W_i, S_i, and P_i, respectively, which are all initially unknown to the decision maker. Due to the lack of past data, no reliable distributional information about these unknown parameters is at hand. Instead, the decision maker is given predictions which are not necessarily accurate, denoted by $\hat{W}_i$, $\hat{S}_i$, and $\hat{P}_i$. For $i \in V$, the actual values of W_i, P_i, and S_i are revealed once node i is visited by one of the vehicles. We assume

Table 2
Notation and input parameters for the online and offline problem.

Notation	Description	Online inputs	Offline inputs
$G = (V_0, E)$	Input network	✓	✓
$V_0 = \{0, 1, 2, \dots, N\}$	Set of nodes	✓	✓
$0 \in V_0$	Depot node	✓	✓
$V = \{1, 2, \dots, N\}$	Set of demand nodes	✓	✓
E	Set of edges	✓	✓
$K = \{1, 2, \dots, \|K\|\}$	Set of vehicles	✓	✓
C_k	Capacity of vehicle k	✓	✓
T_k	Time limit of vehicle k	✓	✓
t_{ij}	Travel time of edge $(i, j) \in E$	✓	✓
$W_i \in \mathbb{R}^+$	Weight of node $i \in V$		✓
$S_i \in \mathbb{R}^+$	Service time of node $i \in V$		✓
$P_i \in \mathbb{R}^+$	Prize of node $i \in V$		✓
$\hat{W}_i \in \mathbb{R}^+$	Prediction for weight of node $i \in V$	✓	✓
$\hat{S}_i \in \mathbb{R}^+$	Prediction for service time of node $i \in V$	✓	✓
$\hat{P}_i \in \mathbb{R}^+$	Prediction for prize of node $i \in V$	✓	✓
$WE_i \in \mathbb{R}^+$	Normalized error for prediction of weight of node $i \in V$		✓
$SE_i \in \mathbb{R}^+$	Normalized error for prediction of service time of node $i \in V$		✓
$PE_i \in \mathbb{R}^+$	Normalized error for prediction of prize of node $i \in V$		✓
$WE \in \mathbb{R}^+$	Maximal normalized weight error over nodes in V		✓
$SE \in \mathbb{R}^+$	Maximal normalized service time error over nodes in V		✓
$PE \in \mathbb{R}^+$	Maximal normalized prize error over nodes V		✓

complete communication between vehicles in the sense that once the values of online parameters for a node are revealed by one of the vehicles, this information is shared among all the vehicles in real-time. This real-time information sharing is pivotal, as it allows other vehicles to instantly adjust their routes and strategies based on the newly acquired data, ensuring that every vehicle operates with the most current information, thereby optimizing the collective performance of the fleet.

The rest of the problem inputs are deterministic and are defined as follows. The depot node is denoted by 0. As a result, $V_0 = 0 \cup \{1, 2, \dots, N\}$. There are $|K|$ homogeneous vehicles in the graph that are initially positioned at node 0, i.e., the depot. The vehicles are represented by set $K = \{1, 2, \dots, |K|\}$ such that each vehicle k has a known capacity of C_k and a known maximum travel time availability of T_k. For each arc $(i, j) \in E$, we denote the travel time from i to j by t_{ij} which satisfy the triangle inequality and are known to all the vehicles. Note that while we assume travel times satisfy the triangle inequality, in real-life applications where this may not hold, the Floyd–Warshall algorithm can be applied to ensure the inequality is satisfied. Additionally, we note that traversing through a node without visiting it is possible. This remark, along with the aforementioned conditions of the triangle inequality, justifies the use of a complete graph to represent the shortest path between any two nodes (Pallottino, 1984).

In the offline problem, each node $i \in V$ can be visited by one and only one of the vehicles. In the online problem, where online parameters are not known, however, some nodes might be *visited* by multiple vehicles in different iterations of the algorithm. We clarify that this happens due to the iterative use of the offline formulation and since the vehicles can bypass a node without servicing it after learning the actual values of its online parameters. That is, in both the offline and online solutions, a node can be *serviced* by at most one vehicle. In the following mathematical model, since we are solving the offline problem, each node is allowed to be visited by at most one vehicle. The objective is to identify a tour for each vehicle which starts and ends at node 0 such that the capacity and time limit constraints of the vehicles are not violated and total collected prize by all the vehicles is maximized. The notation and input parameters for both the online and offline problems are provided in Table 2.

3.1. A mixed-integer programming formulation for the offline CTOP

Given predictions $\hat{W}_i$, $\hat{S}_i$, $\hat{P}_i$, we can model the offline version of the OCTOP as a Mixed-Integer Programming (MIP) formulation. In the following, we first present the decision variables and then give this MIP model.

Decision Variables:

$$y_i^k = \begin{cases} 1, & \text{if node } i \in V \text{ is visited by vehicle } k \in K. \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

$$x_{ij}^k = \begin{cases} 1, & \text{if edge } (i,j) \in E \text{ is traversed in the direction from } i \text{ to } j \text{ by vehicle } k \in K. \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$$u_i^k \geq 0 : \text{flow amount on node } i \in V_0 \text{ for vehicle } k \in K \tag{3}$$

MIP model:

$$[\text{CTOP}] \ \max \quad \sum_{k \in K} \sum_{i \in V} \hat{P}_i y_i^k \tag{4}$$

$$\text{s.t.} \quad \sum_{i \in V} \hat{W}_i y_i^k \leq C_k, \quad k \in K \tag{5}$$

$$\sum_{(i,j)\in E} t_{ij} x_{ij}^k + \sum_{i\in V} \hat{S}_i y_i^k \le T_k, \quad k \in K \tag{6}$$

$$\sum_{j\in V_0, \neq i} x_{ij}^k - \sum_{l\in V_0, \neq i} x_{li}^k = 0, \quad i \in V_0, k \in K \tag{7}$$

$$\sum_{i\in V} x_{0i}^k = 1, \quad k \in K \tag{8}$$

$$\sum_{i\in V_0, \neq j} x_{ij}^k = y_j^k, \quad k \in K, j \in V \tag{9}$$

$$\sum_{k\in K} y_j^k \le 1, \quad j \in V \tag{10}$$

$$u_j^k \ge u_i^k + 1 - |V_0| * (1 - x_{ij}^k), \quad (i,j) \in E : j \neq 0, k \in K. \tag{11}$$

$$u_i^k \le \sum_{j\in V} y_j^k, \quad k \in K, i \in V \tag{12}$$

$$u_0^k = 0, \quad k \in K \tag{13}$$

$$y_i^k \in \{0,1\} \quad \forall i \in V, k \in K \tag{14}$$

$$x_{ij}^k \in \{0,1\} \quad \forall (i,j) \in E, k \in K \tag{15}$$

$$u_i^k \ge 0 \quad \forall i \in V_0, k \in K \tag{16}$$

Objective function given in (4) maximizes total collected prize. (5) and (6) constraints ensure that capacity and time limit restrictions for each vehicle are not violated. Constraints (7) are flow balance constraints and (8) is to guarantee that all the vehicles start their routes from the depot. Constraints (9) set if a node is visited by a vehicle or not. By constraints (10), each node can be visited by at most one vehicle. Constraints (11)–(13) prevent the formation of subtours in serving nodes. Constraints (14)–(16) define the domains of the decision variables.

4. Tight upper bounds on the competitive ratio

Online optimization problems have been examined in the academic literature using the theoretical framework known as *competitive analysis*, e.g., see Shiri et al. (2023), Gong et al. (2022), Akbari et al. (2021), Ma et al. (2021), Ma and Simchi-Levi (2020), Jaillet and Wagner (2008). The competitive ratio measures the proximity of the solution of the online algorithm under partial information to the offline optimal solution obtained with complete information (Sleator and Tarjan, 1985).

Online algorithms belong to either deterministic or randomized family of algorithms. The output of a deterministic online algorithm remains the same if it is applied to the same online problem instance several times. In contrast, a randomized online algorithm may produce a different output each time that it is executed on the same instance, hence, for randomized algorithms the expected objective function value is considered.

To define the competitive ratio, we denote an arbitrary online algorithm by ALG, i.e., ALG can be deterministic or randomized, and the offline optimal algorithm by OPT. We also represent the solutions of ALG and OPT applied to an instance δ of an arbitrary online problem by $ALG(\delta)$ and $OPT(\delta)$, respectively. For an online maximization problem, we refer to Ma and Simchi-Levi (2020) for defining the competitive ratio for ALG as the infimum of the (expected) objective function value of ALG over the objective function value of OPT for any problem instance $\delta \in \Delta$, i.e.,

$$0 \le \inf_{\delta\in\Delta} \frac{\mathbb{E}(ALG(\delta))}{OPT(\delta)} \le 1. \tag{17}$$

To analyze the OCTOP from a competitive ratio viewpoint, we define below six metrics to study the deviation of predicted online parameters from their respective actual values.

Definition 1. For each node $i \in V$, define the normalized errors for weight (WE_i), service time (SE_i), and prize (PE_i) predictions as the relative discrepancies between the predicted and actual values of these parameters, i.e., $WE_i = \frac{|W_i - \hat{W}_i|}{W_i}$, $SE_i = \frac{|S_i - \hat{S}_i|}{S_i}$, and $PE_i = \frac{|P_i - \hat{P}_i|}{P_i}$, representing the parameters' absolute percentage errors, respectively. Furthermore, the maximum normalized errors across all nodes in V for each parameter are denoted as $WE = \max_{i\in V} WE_i$, $SE = \max_{i\in V} SE_i$, and $PE = \max_{i\in V} PE_i$.

Remark 1. A key feature of the OCTOP which differentiates it from stochastic or robust optimization problems is the fact that no information is assumed about the normalized errors defined in Definition 1. That is, WE_i, SE_i, PE_i, WE, SE, and PE are defined as unknown parameters in this paper.

Utilizing Definition 1, we prove an upper bound on the competitive ratio of online algorithms against the OCTOP, which is sensitive to the accuracy of the predictions, i.e., the unknown normalized errors.

Lemma 1. *No online algorithm has a competitive ratio higher than*

$$Competitive\ ratio = \begin{cases} \frac{1-PE}{1+PE} & if \quad WE = SE = 0 \ \ \& \ \ PE < 1 \\ 0 & \text{otherwise,} \end{cases} \tag{18}$$

for the OCTOP.

Proof of Lemma 1 is provided in Appendix A.

Remark 2. Lemma 1 immediately implies that incorporating randomization into the design of online algorithms for solving the OCTOP does not significantly improve the competitive ratio, especially when dealing with a large number of candidate locations, a common characteristic in the OCTOP.

The nonzero part of the upper bound proved in Lemma 1 holds relevance for decision makers in situations where: (i) the predictions on weight and service time at locations are accurate, e.g., the type of service is identical with a known weight and service time, or (ii) the capacity constraint is relaxed and service times at locations are negligible. In particular, the tight upper bound on the competitive ratio finds merit in security and police operations, where the decision maker can be fed with adversarial false information on the prize at different target locations (Gupta et al., 2020). We will further investigate this upper bound empirically in Section 6.3.1 by testing our algorithms on simulated instances which satisfy the conditions in the nonzero part of the upper bound.

Remark 3. The upper bound in Lemma 1 is also valid for similar online variants of the special cases of CTOP, including the OP and the TOP.

4.1. Resource augmentation

Lemma 1 highlights a challenge for online algorithms in the face of uncertain node weights and service times, due to the competitive ratio's inherent conservatism. Resource augmentation, a strategy discussed in Section 2.3 and introduced by Jaillet and Wagner (2008) for VRP variants, is proposed as a solution to mitigate this issue. We adapt this strategy to the OCTOP's maximization context, examining the effects of granting online algorithms additional time and capacity resources. This exploration aims to determine how investments in these resources could strategically enhance the competitive ratio amidst system uncertainties.

Lemma 2. *Suppose that the time budget of the vehicles in the online and offline algorithms is $\alpha \cdot T$ and T, respectively, and the capacity of the vehicles in the online and offline algorithms is $\beta \cdot C$ and C, respectively, where $\alpha \geq 1$ and $\beta \geq 1$. No online algorithm has a competitive ratio higher than*

$$Competitive\ ratio = \begin{cases} \frac{1-PE}{1+PE} & if \quad WE \leq \frac{\beta-1}{\beta} \ \ \& \ \ SE \leq \frac{\alpha-1}{\alpha} \ \ \& \ \ PE < 1 \\ 0 & \text{otherwise,} \end{cases} \tag{19}$$

for the OCTOP.[1]

Proof of Lemma 2 is provided in Appendix A. Lemma 2 implies that simultaneous resource augmentation on time and capacity results in a nonzero competitive ratio under uncertainty in time and capacity. This improves the upper bound in Lemma 1. However, investing on only one of these two parameters (e.g., time) will not improve the competitive ratio in the presence of uncertainty in the other parameter (e.g., capacity).

4.2. Consistency, robustness, and smoothness

In this section, we present upper bounds on the consistency, robustness, and smoothness of online algorithms which are described in Section 2.3.1. In the problem we introduced in this paper, the OCTOP, as we have three online parameters, i.e., P_i, W_i, and S_i for $i \in V$, we utilize the definition given by Gouleakis et al. (2023), for evaluating problems characterized by multiple online parameters. We remark that we present our results for the resource augmentation case henceforth. This includes its special case without resource augmentation as well.

Definition 2. Let $\eta = (WE, SE, PE)$ be an error metric which is defined as a tuple (Gouleakis et al., 2023), where $WE, SE, PE \geq 0$. An online algorithm is called

- ψ-consistent if its competitive ratio be equal to ψ if $\eta = (0, 0, 0)$.
- χ-robust, if its (worst-case) competitive ratio equals χ among all possible scenarios for η, and
- f-smooth for a continuous function $f(.)$, if its competitive ratio equals $f(\eta)$ for any given scenario for η.

[1] The upper bounds for the cases where resource augmentation is conducted separately only on time availability or capacity of vehicles are trivial given the proof of *Lemma 2*

Given Definition 2, Lemma 1, and Lemma 2, we present tailored upper bounds on the concepts of consistency, robustness, and smoothness of online algorithms for the OCTOP.

Corollary 1. *No online algorithm can be better than* 1*-consistent. Moreover, no online algorithm is robust against the OCTOP. This is because there are scenarios for* η*, i.e.,* $WE > \frac{\beta-1}{\beta}$*, or* $SE > \frac{\alpha-1}{\alpha}$*, or* $PE \geq 1$*, which enforce the competitive ratio of zero to any online algorithm. Where* η *permits nonzero competitive ratio, i.e.,* $WE \leq \frac{\beta-1}{\beta}$*,* $SE \leq \frac{\alpha-1}{\alpha}$ *and* $PE < 1$*, no online algorithm can be better than* $\frac{1-PE}{1+PE}$*-smooth.*

Proof of Corollary 1 is provided in Appendix A. In the next section, we propose three different online algorithms to solve the OCTOP. In Proposition 1 as well as Corollaries 2 and 3, we will comment on the competitive ratio of our algorithms as well as their consistency, robustness, and smoothness.

5. Online algorithms

We introduce three alternative online algorithms. Two of these algorithms are based on the MIP formulation given in Section 3.1 and the third one is an approximation-based algorithm which can be implemented in polynomial time.

5.1. A one-time optimization algorithm

We introduce the following two-phase algorithm. In the first phase, the MIP formulation given in Section 3.1 is applied by setting $W_i = \hat{W}_i$, $S_i = \hat{S}_i$, and $P_i = \hat{P}_i$, to assign a cluster of nodes to the vehicles, as well as an order of servicing the assigned nodes by the vehicles. Let V_k, i.e., $k \in K$, be the set of the assigned demand nodes to vehicle k, and let O_k be the order of visiting the nodes in V_k.

In the second phase, vehicles are dispatched to the assigned nodes with respect to the order specified in the first phase. Once a vehicle arrives at a node, it checks if servicing the node is feasible considering the current capacity and remaining time. If servicing the node is feasible, the vehicle services the node, otherwise, the vehicle skips the node. Hereafter, we refer to this solution as the *one-time optimization* (OTO) algorithm. The pseudo-code of the OTO algorithm is given in Procedure 1.

Proposition 1. *The OTO algorithm matches the upper bounds in Lemma 1, Lemma 2, Corollary 1; hence it has an optimal competitive ratio.*

Proof of Proposition 1 is provided in Appendix A.

Procedure 1: OTO algorithm.

Input:
an instance of the problem ▷ see Table 2
Output: a feasible online solution with an optimal competitive ratio
Phase 1:
  $W_i = \hat{W}_i$
  $S_i = \hat{S}_i$
  $P_i = \hat{P}_i$
  solve the MIP formulation given in §3.1
  find V_k and O_k for $k \in K$
Phase 2:
  while $K \neq \emptyset$**:**
    dispatch vehicle $k \in K$ to the next node $v \in V_k$ with respect to O_k
    observe exact values of W_v, S_v, P_v
    if servicing v is feasible **then:**
      service v
    else:
      skip v
    end if
    update capacity and time
    $V_k = V_k \setminus \{v\}$
    if $V_k = \emptyset$ for $k \in K$ **then:**
      $K = K \setminus \{k\}$ ▷ dispatch k to depot
    end if
  end while

Procedure 2: IO algorithm.

Input:
an instance of the problem ▷ see Table 2
Output: an improved feasible online solution with an optimal competitive ratio
Phase 1: see Phase 1 of Procedure 1
Phase 2:
while $K \neq \emptyset$**:**
  $V' = V \setminus \cup_{k \in K} V_k$
  $OBJ_k = \sum_{i \in V_k} \hat{P}_i$ for $k \in K$
  dispatch vehicle $k \in K$ to the next node $v \in V_k$ with respect to O_k
  observe exact values of W_v, S_v, P_v and update predictions to exact values for node v
  solve the single-vehicle MIP formulation in §3.1 in **real time** (see §5.2)[2]. ▷ this is the rival solution
  find V_k' and O_k'
  $SOL_k = \sum_{i \in V_k'} \hat{P}_i$
  if $OBJ_k \leq SOL_k$ **then:**
    $V_k = V_k'$ and $O_k = O_k'$ ▷ use the rival solution henceforth
    if $v \in V_k$ **then:**
      if servicing v is feasible **then:**
        service v
        $V = V \setminus \{v\}$
      else:
        skip v
      end if
    end if
  else:
    if servicing v is feasible **then:**
      service v
      $V = V \setminus \{v\}$
    else:
      skip v
    end if
  end if
  update capacity and time
  $V_k = V_k \setminus \{v\}$
  if $V_k = \emptyset$ for $k \in K$ **then:**
    $K = K \setminus \{k\}$ ▷ dispatch k to depot
  end if
end while

5.2. An iterative optimization algorithm

The OTO algorithm does not effectively utilize the online information which is obtained dynamically at the demand nodes. This fact motivates us to design an online algorithm whose foundation is the solution of the OTO algorithm such that it maintains the optimal competitive ratio, which also involves gradual refinements that are incorporated into the algorithm, dynamically as the new information is revealed.

To this end, we introduce the *iterative optimization* (IO) algorithm. In the IO algorithm, we first execute the solution of the first phase of the OTO algorithm from which we obtain an assignment of nodes to vehicles, as well as the order of visiting those nodes. Let V_k, i.e., $k \in K$, be the set of the assigned demand nodes to vehicle k, and let O_k be the order of visiting the nodes in V_k. Furthermore, let OBJ_k be the total prize that is expected to be achieved by vehicle k based on this solution, i.e., $OBJ_k = \sum_{i \in V_k} \hat{P}_i$.

As the OTO algorithm solves an orienteering problem instance, there might be a subset of nodes $V' \subseteq V$, such that $V' = V \setminus \cup_{k \in K} V_k$. Hereafter, we decompose this solution into k different single-vehicle sub-problems, namely, δ_k for $k \in K$, i.e., δ_k is an instance of the OCTOP with a single vehicle. We modify the solution in each sub-problem in a way that it maintains the optimality of the solution from the competitive ratio perspective, while it improves the solution wherever possible. For this purpose, each single vehicle k, in sub-problem δ_k, is dispatched to the demand nodes based on O_k. Once the vehicle arrives at node $v \in V_k$ and

[2] This is done using the actual values of online parameters for the unserviced visited nodes and predictions for the unvisited nodes. In this modified MIP formulation, the set of nodes is reduced to $V' \cup V_k$, with the starting depot being node v and the ending depot being node 0.

observes the actual values of W_v, S_v, and P_v, a single-vehicle version of the MIP formulation provided in Section 3.1 is solved. This is done using the actual values of online parameters for the unserviced visited nodes and predictions for the unvisited nodes. In this modified MIP formulation, the set of nodes is reduced to $V' \cup V_k$, with the starting depot being node v and the ending depot being node 0.

We denote the set of nodes that are assigned to be serviced by the vehicle in this new solution by V'_k, the order of servicing these nodes by O'_k, and the total prize that is expected to be achieved from nodes in V'_k by $SOL_k = \sum_{i \in V'_k} \hat{P}_i$. We consider the following two cases.

- If $OBJ_k \le SOL_k$, then V_k and O_k are updated to V'_k and O'_k, respectively. If node v still belongs to V_k (which is recently updated) and it is feasible to service this node, the vehicle services node v, removes v from V, and then moves to the next node based on the updated O_k. Otherwise, the vehicle is directly dispatched to the next node based on the updated O_k, i.e., without servicing node v. In this case, if the obtained solution consists nodes from V' that are not in set V_k, those nodes are added to V_k and are removed from V'. Also, any node which was previously in V_k and was not included in V'_k is added to V'.
- If $OBJ_k > SOL_k$, then V_k and O_k are not updated. The vehicle services node v if servicing this node is feasible, removes i from V, and then moves to the next node based on the current O_k. Otherwise, the vehicle skips node v and moves to the next node based on O_k.

After both of the above cases, when the vehicle departs from node v, this node is removed from V_k. This procedure is repeated until all the vehicles return back to the depot. The pseudo-code of the IO algorithm is given in Procedure 2.

Proposition 2. *The IO algorithm achieves tight upper bounds in Lemmas 1 and 2, as well as the bounds on the consistency, robustness, and smoothness given in Corollary 1; hence it has an optimal competitive ratio.*

Proof of Proposition 2 is provided in Appendix A.

5.2.1. Node exclusion in IO algorithm

To improve the performance of the IO algorithm in scenarios with a large number of nodes, we introduce a *node reduction* technique inspired by the arc reduction strategy in the "Backbone" algorithm outlined in Bertsimas et al. (2019). A key differentiator of our approach compared to the "Backbone" algorithm is the fact that we focus on reducing nodes rather than arcs. Within each iteration of the IO algorithm (line 5:f), an orienteering problem is solved, where the starting point of the vehicle dynamically changes based on where they are located at that point in time. As the cardinality of V' grows, the associated optimization problem may demand excessive computational time. The primary objective of the node reduction heuristic is to strategically eliminate certain nodes from V' that are deemed less probable to contribute to the optimal solution. To do this, we first associate a score to each of the nodes in V' as follows:

$$\bar{S}_i = \frac{\hat{P}_i}{\hat{W}_i \cdot \hat{S}_i \cdot t_{i0} \cdot t_{ci}},$$

where t_{i0} denotes the required time to travel from node i to the depot and t_{ci} is the time to travel from the current node of the corresponding vehicle to node i. We then sort the nodes based on their scores and only include the upper γ percent of the corresponding nodes. An effective value of γ that guarantees a balance between computational time and solution quality can be obtained by parameter tuning using computational experiments.

Corollary 2. *The version of the IO algorithm with node exclusion feature matches the upper bounds in Lemmas 1 and 2, as well as the bounds on the consistency, robustness, and smoothness given in Corollary 1; hence it has an optimal competitive ratio.*

Proof of Corollary 2 is provided in Appendix A. This implies that the OTO and IO algorithms are equivalent from a worst-case theoretical point of view over different scenarios for prediction accuracy. However, we further distinguish between the performance of these algorithms empirically in Section 6. By doing so, we go beyond the theoretical literature of online VRP variants, e.g., the works of Jaillet and Wagner (2006, 2008), Akbari and Shiri (2021), Akbari et al. (2021), Zhang et al. (2019). We explore the distinctions among various online algorithms with *optimal worst-case competitive ratios* based on their average-case performance. This exploration is motivated by Ma et al. (2021), which highlights the importance of analyzing the average-case performance of online algorithms beyond their theoretical competitive ratio, specifically in the context of another online optimization problem within the revenue management domain. We close this subsection by introducing another feature of the IO algorithm which we apply to reduce the running time of this algorithm in real-time.

5.2.2. An acceleration technique for the IO algorithm

In our study, we enhance our iterative optimization model by incorporating an acceleration technique inspired by Bertsimas et al. (2019). This technique leverages the existing route of each vehicle as an initial solution for the optimization model in consecutive iterations, thus facilitating a faster optimization process. By providing the solver with an initial feasible solution, we ensure that our model can swiftly converge to a high quality feasible solution. This acceleration technique holds practical significance as well. Given that the optimization step is performed whenever a decision regarding item collection or bypassing is required, it is imperative to minimize the decision-making time to avoid prolonged waiting periods for nodes. Therefore, the ability to make such decisions rapidly is of utmost importance for efficient operations. This technique is implemented in line 5:f of Procedure 2.

Table 3
Predictions and actual values of prize, weight, and service time for each node in the illustrative example.

Node	Node 1		Node 2		Node 3		Node 4	
	Actual	Predicted	Actual	Predicted	Actual	Predicted	Actual	Predicted
Prize	500	1000	200	150	900	900	800	800
weight	150	350	250	200	300	350	150	200
Service time	50	100	200	100	200	100	100	100

5.3. Polynomial time approximation-based IO algorithm

Although the OTO and IO algorithms achieve optimal competitive ratios, their reliance on solving the MIP formulation makes them impractical for real-time solutions in large-scale scenarios due to the exponential execution time of the formulation. This limitation underscores the importance of an alternative approach that offers competitive ratio guarantee and can operate within polynomial time. Utilizing the 3.53-approximation algorithm for solving the CTOP given in Bock and Sanità (2015) (see Appendix B for a detailed description of this approximation algorithm), we implement a modified version of the IO algorithm in polynomial time. We refer to this algorithm as *Approximation-based IO* (AIO) algorithm. In the first stage of this algorithm, instead of solving the MIP formulation of the CTOP (line 4:d in Procedure 1), we apply the 3.53-approximation algorithm. In the second stage, when solving iterations that involve a single-vehicle scenario (line 5:f of Phase 2 in Procedure 2), with different starting and ending points, we introduce a strategic adjustment. Rather than tackling the sub-problem with varying depots, we standardize the starting and ending points at the actual depot. To navigate from the actual depot to various nodes, we assign a high travel time value, effectively deterring their selection, except for the current vehicle node, which is assigned a minimal travel time to facilitate its preference. This method allows the AIO algorithm to efficiently approximate solutions in polynomial time without the computational burden of exact formulations.

Corollary 3. *The AIO algorithm which can be implemented in polynomial time, has a competitive ratio of at least $\frac{1}{3.53}$ times the tight upper bounds in Lemmas 1 and 2, as well as the bounds on the consistency, robustness, and smoothness given in Corollary 1.*

Proof of Corollary 3 is provided in Appendix A.

5.4. An illustrative example

To elaborate more on the logic of our algorithms, we generate a small example and discuss the implementation of the OTO, IO, and AIO algorithms, alongside the solution for the offline optimal scenario in this example. We consider a single vehicle with a capacity of 500 and a maximum tour duration of 1000. The vehicle is initially positioned at the depot and the network contains four demand nodes. The travel time in any direction between any two locations is fixed at 100. The predictions and actual values of prize, weight, and service time for each demand node are shown in Table 3. We discuss four illustrated routing solutions in Fig. 2:

- **Offline model:** In the offline optimal solution, the actual values of the parameters are known a priori. Hence, the vehicle services node 3 and node 4 and collects a maximum prize of $900 + 800 = 1700$.
- **OTO algorithm:** As the OTO algorithm only has access to the predictions, the MIP formulation in Section 3.1 that is solved in the first phase of the OTO algorithm enforces the vehicle to only visit node 1. In the second phase of the OTO algorithm, since the weight of node 1 is less than the capacity of the vehicle (i.e., $150 < 500$), the vehicle visits node 1 and collects a prize of 500. Since the OTO algorithm has no re-optimization mechanism, even though the vehicle has remaining capacity, it follows the same route that was enforced in the first phase and returns to the depot.
- **IO algorithm:** Similar to the OTO algorithm, in the first phase of the IO algorithm, node 1 is assigned to the vehicle. In the second phase, upon visiting node 1 and observing the actual values of prize, weight, and service time, the vehicle realizes that the actual parameters are less than the predictions. Specifically, the weight of the item is 200 units less than the predicted weight, i.e., the remaining capacity of the vehicle would be 350 rather than 150 if the item at node 1 is collected which implies that re-optimizing the decisions may lead to greater prize collection. Subsequently, the IO algorithm solves the MIP formulation discussed in Section 5.2 for re-optimizing the decisions. The re-optimized decision enforces the vehicle to collect the item from node 1 followed by a visit to node 3. Upon arriving at node 3, the vehicle observes the actual values of the parameters and solves the MIP formulation discussed in Section 5.2 for re-optimizing the decisions. As a result, the vehicle collects the item at node 3 and returns to the depot. Hence, the IO algorithm collects a total prize of $500 + 900 = 1400$.
- **AIO algorithm:** The AIO algorithm approaches this example in a similar fashion to the IO algorithm, except after the realization of the properties of the visited node, instead of an exact formulation, an approximation algorithm is used to re-optimize the vehicle's decision problem. Similar to the eventual route taken by the IO algorithm, AIO visits nodes 1 and 3 before returning to the depot, collecting $500 + 900 = 1400$ as total prize along the way. Note that, as the approximation algorithm favors exploring longer routes within time and capacity limits, we expect to see deviation from the IO's solution in larger instances.

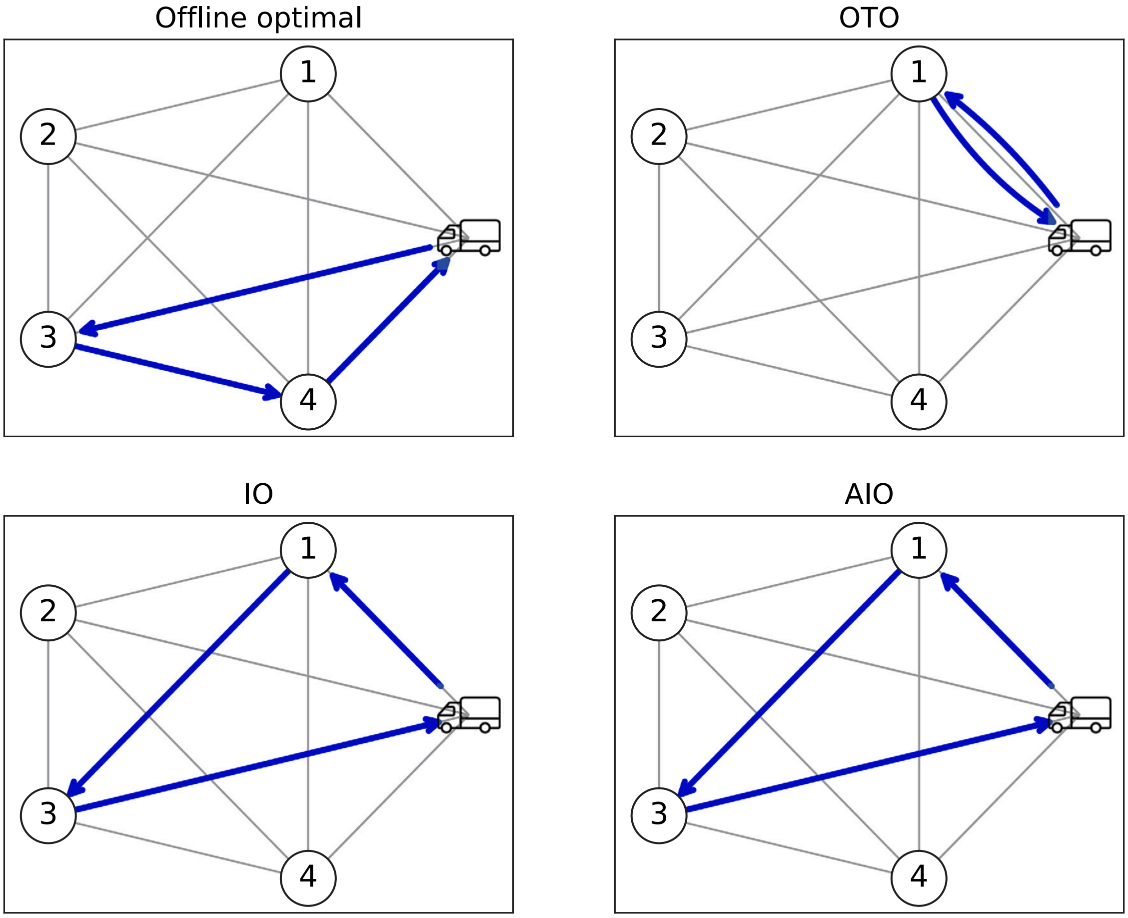

Fig. 2. An illustrative example.

6. Computational experiments

In this section, we discuss our experimental setup, both in terms of hardware as well as models and performance criteria. Subsequently, we present our results based on our extensive computational experiments conducted on randomly generated synthetic instances, followed by analyzing the special case where prize is the only online parameter. We then examine the impact of resource augmentation presented in Section 4.1. To end the section, we validate the performance of our online algorithms using instances sourced from the existing literature.

6.1. Experimental setup and preliminaries

In all our experiments, we coded various models using Python 3.9, executed on a system equipped with an Intel Core i5 processor, 8 GB of RAM, and a 64–bit Windows 10 operating system. We used Gurobi 9.5.0 solver under an academic license for solving our optimization formulations. In the following subsections, we elaborate on the performance criteria for evaluating the empirical performance of our algorithms. We then explain the procedures we have used in simulating both our offline and online models. Lastly, we discuss an important distinction between two concepts, *simulation run time* and *iteration run time*, particularly relevant within the context of online algorithms.

6.1.1. Performance criterion for the empirical analysis

For measuring the computational performance of online algorithms, the notion of *experimental* competitive ratio (ECR) has been widely applied in the literature (Shiri et al., 2024; Yao et al., 2022; Akbari et al., 2021; Shiri et al., 2020; Zhang et al., 2019). Formally, for a given set of test instances $\{\delta_1, \delta_2, \ldots, \delta_\psi\} \subset \Delta$, the ECR of ALG is calculated as

$$\sum_{i=1}^{\psi} \frac{ALG(\delta_i)}{OPT(\delta_i)}. \tag{20}$$

In this paper, we investigate the ECR of our online algorithms on two different data sets. Initially, we examine smaller instances where MIP formulation is solvable, allowing for exact ECR calculation. We refer to these instances as the *synthetic instances* (Section 6.2 and Section 6.3) and we conduct a thorough analysis to determine the algorithms' ECR across tested scenarios. Additionally, to validate the performance of our online algorithms on larger instances, known as *CTOP instances*, drawn from existing literature (Archetti et al., 2009). On the CTOP instances, for the first phase of the OTO and the IO algorithms, that involve solving the offline CTOP using the predicted values of prize, weight and service time, we used the heuristic procedure presented in Appendix C. In the second phase of the IO algorithm implemented on the CTOP instances, we have used the MIP formulation for solving the sub-problems described in Section 5.2 and incorporated the acceleration and node reduction techniques described in Section 5.2.2 and Section 5.2.1 to obtain high quality solutions in real time. Implementation of both phases in the AIO algorithm is inspired by the approximation algorithms presented in Bock and Sanità (2015). Furthermore, we used the best known offline solutions in the literature for computing the ECR values.

6.1.2. Experimental setup for our solution methodologies

Offline experiments. Our datasets, both synthetic and CTOP instances, contain parameters representing offline input values, including the exact values for prize, weight, and service time associated with each demand node. These values, stored in our original data sets, form the basis of our offline experiments and calculating the offline optimal objective function values.

Online experiments. For conducting the online experiments, we need realizations (i.e., scenarios) for normalized deviations of predictions from actual values of the online parameters (see Definition 1). To generate the scenarios for the online predictions, we apply two distinct procedures for introducing prediction errors. The first procedure involves the generation of random prediction errors within specified intervals (i.e., deviations) of 10%, 20%, 50%, and 100% around the exact offline values for each of the parameters prize, weight, and service time for each demand node. For this, we considered the scenarios of $WE = SE = PE = 10\%, 20\%, 50\%, 100\%$. We note that WE, SE and PE are not known to the online algorithms and are solely used to generate the predictions. That is, we generate $\hat{W}_i, \hat{S}_i, \hat{P}_i$ using uniform probability distribution within intervals $[W_i(1-WE), W_i(1+WE)], [S_i(1-SE), S_i(1+SE)], [P_i(1-PE), P_i(1+PE)]$, respectively. We refer to this procedure for simulation of prediction errors as the *bounded randomized error* (BRE) procedure. In the second procedure, to simulate predictions containing false information, we deliberately impose maximal errors of $WE = SE = PE = 10\%, 20\%, 50\%, 100\%$ to all demand nodes where WE, SE and PE are not known to the online algorithms. We refer to this procedure for generating prediction errors as the *forced errors* (FE) procedure. Specifically, for half of the nodes, the predictions are at their worst possible values to simulate false negative predictions, while for the other half, the predictions are at the best possible values to impose false positive predictions.

While both the BRE and FE procedures yielded an average error of 0, the FE procedure introduced data points with more pronounced deviations. To illustrate, consider a hypothetical scenario where the actual service time at a node is 10 min. In the case of a 50% error using the BRE approach, the predicted service time for that node would be a uniformly distributed random value between 5 and 15 min. However, in the FE procedure, the service time for this particular node would be either 5 or 15 min. We remark that we apply WE, SE, PE as online parameters in our analysis for the sake of simulating a comprehensive range of scenarios to trace the performance of our online algorithms as a function of prediction accuracy. That is, WE, SE, PE are not input parameters for our online algorithms in our simulations (see Remark 1 and Table 2). In our analysis, we generated 10 different random scenarios using the BRE and one scenario using FE procedure and tested the average performance of our online algorithms on them. We emphasize that, in the FE procedure, intentional errors were introduced to alter the status of a node from good to bad and vice versa. Consequently, the generation of multiple scenarios was deemed unnecessary, as they would all yield identical outcomes.

6.1.3. Measuring run time in our online algorithms

It is crucial to highlight that the provided run times in Section 6.2, Section 6.3.1, and Section 6.4 pertain to the total simulation run times of the algorithms, rather than traditional total CPU times. In practical implementation, when vehicles reach a demand node, they execute the decision step of their respective algorithms to make the subsequent decision—either accepting the item, collecting it, and proceeding to the next selected demand node (or depot), or rejecting the item and moving on to the next demand node (or depot). Therefore, the run time of the decision step is of paramount importance, as it is impractical to keep the vehicles waiting for an extended duration to execute the algorithm. Consequently, in the IO algorithm, the run time of the decision step (the iterative optimization model) was restricted to 1 min. The acceleration and node reduction techniques played a vital role in facilitating this feature of the IO algorithm on the CTOP instances. Due to its focus solely on checking the feasibility of an item collection, the OTO algorithm executes its decision step in less than 1 s. Similarly, the decision step of the polynomial time AIO algorithm remains less than 1 s in our experiments.

6.2. Synthetic instances

The first data set named synthetic instances consisted of 10 randomly generated instances, each containing 20 demand nodes. In these synthetic instances, travel distances satisfy the triangle inequality. These instances were utilized as preliminary offline samples to assess the efficacy of the online algorithms. Further details about these 10 instances, including average prize, service time and weight at demand nodes ($\bar{P}$, $\bar{S}$ and $\bar{W}$), number of vehicles and their capacity and availability duration (K, T and C) and minimum, maximum and average travel time among all the demand node pairs (min t_{ij}, max t_{ij} and $\bar{t_{ij}}$) are presented in Table 4.

Table 4
Characteristics of the synthetic instances.

Instance	N	$\bar{P}$	$\bar{S}$	$\bar{W}$	K	T	C	min t_{ij}	max t_{ij}	$\bar{t_{ij}}$
SI 1	20	28.35	16.10	17.25	3	240	40	6.08	101.83	49.63
SI 2	20	34.50	13.75	17.45	3	280	40	5.83	123.75	54.29
SI 3	20	25.20	18.25	14.80	4	180	65	0.00	110.54	52.05
SI 4	20	13.40	11.75	19.65	3	280	65	8.25	104.81	49.77
SI 5	20	27.55	18.25	19.10	2	400	65	3.00	123.07	52.04
SI 6	20	20.40	19.30	19.15	3	300	60	7.28	109.79	47.93
SI 7	20	25.10	14.60	13.25	4	240	40	7.07	64.03	31.56
SI 8	20	27.45	25.40	13.60	4	360	30	6.40	115.88	51.41
SI 9	20	26.95	26.05	12.80	3	450	40	6.32	111.13	49.45
SI 10	20	33.90	18.40	18.10	5	280	35	9.00	118.79	54.43
mean	20	26.28	18.19	16.52	3.4	301	48	5.92	108.36	49.26
Std Dev	0	6.11	4.64	2.65	0.84	80.89	13.98	2.63	17.17	6.56

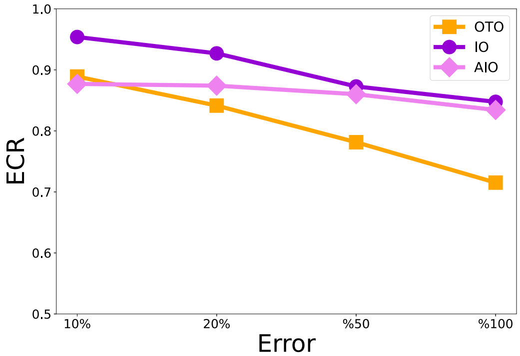

Fig. 3. ECR of synthetic instances under BRE.

Figs. 3 and 4 illustrate the average computational results obtained from applying our online algorithms on 10 synthetic instances with the BRE procedure simulating prediction errors across various scenarios. Fig. 3 presents the ECR, and Fig. 4 displays the simulation run time of the algorithms in seconds. In the context of a maximization problem, the ECR values are capped at 1, as the best solutions generated by an online algorithm cannot surpass the optimal solution of the offline problem.

Fig. 3 illustrates that as the error increases, both OTO and IO algorithms tend to yield solutions of lower quality. However, an intriguing opposite trend is observed for the AIO algorithm, wherein higher errors can lead to higher quality solutions. Furthermore, it is evident that the IO algorithm consistently outperforms both OTO and AIO across all scenarios. While one might anticipate the superiority of IO over AIO, especially in scenarios with lower errors, the superior performance of IO over OTO can be attributed to the absence of a bypassing mechanism for feasible item collections in OTO.

Regarding running times, it is noteworthy that the execution time of the IO algorithm is comparable to that of the OTO algorithm. This suggests that the majority of the IO algorithm's execution time is dedicated to generating initial solutions rather than the iterative procedure. This can be attributed to the acceleration technique discussed in Section 5.2.2.[3] In contrast, the running time of the AIO algorithm remains consistently low irrespective of variations in error. This aligns with expectations, as the AIO algorithm is a polynomial time approximation of IO.

Figs. 5 and 6 present the average results obtained from the synthetic instances utilizing the FE procedure to simulate the prediction errors. Compared to the BRE scenarios, the performance of all three algorithms has been deteriorated under FE scenarios. Notably, in two of the four cases, the IO algorithm outperforms the AIO algorithm. Both IO and AIO outperform the OTO solution in all cases. We remark that the IO algorithm achieves the best overall results considering all error scenarios. Another important observation is that the superiority of IO to OTO is more evident in FE scenarios when compared to BRE. This highlights the importance of incorporating online observations in the decision making process particularly when they are significantly different from predictions.

From a run-time perspective, a similar pattern to the BRE scenarios can be observed in which the run time of the IO algorithm is comparable to that of the OTO algorithm, indicating that the majority of the execution time of the IO algorithm is allocated to generating initial solutions rather than the iterative procedure. The AIO algorithm converges very quickly in all the scenarios due to its polynomial time execution time.

[3] The node exclusion feature discussed in Section 5.2.1 is only incorporated for the CTOP instances.

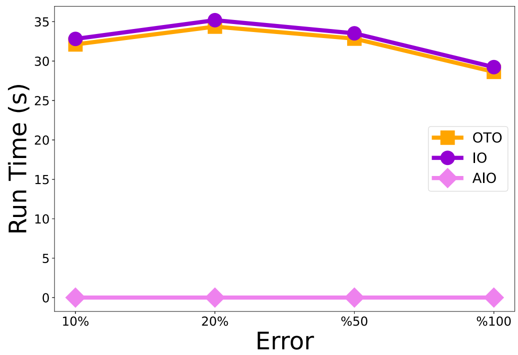

Fig. 4. Simulation runtime (s) of synthetic instances under BRE.

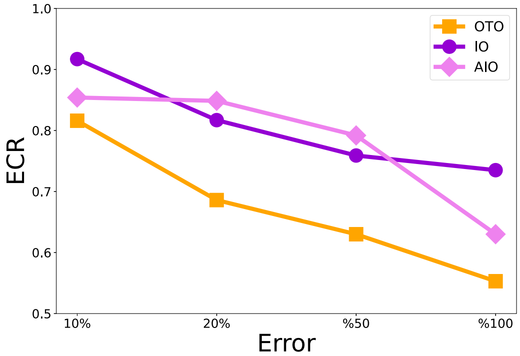

Fig. 5. ECR of synthetic instances under FE.

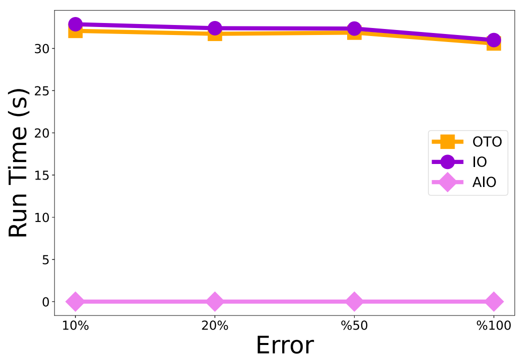

Fig. 6. Simulation runtime (s) of synthetic instances under FE.

6.3. Sensitivity analysis

In this section, we conduct sensitivity analysis on synthetic instances aligned with the theoretical discussions presented in Section 4 and Section 4.1. The choice to focus on the synthetic instances stems from our ability to derive exact ECR values within this subset. This decision ensures the clarity and precision of our sensitivity analysis by leveraging instances where optimal solutions can be determined for mathematical formulations embedded in our online optimization algorithms. Here we aim to isolate the effects of varying parameters on the ECR, ensuring that our sensitivity analysis remains unaffected by the nuances of sub-optimal solutions associated with larger instances.

6.3.1. Revisiting Lemma 1 and Proposition 1 and Corollary 2

We note that our computational experiments in the previous subsections correspond to the scenarios where the upper bound on the competitive ratio fixes at 0. In this subsection we revisit Lemma 1 and Proposition 1 and Corollary 2. Specifically, we focus on the conditions imposing the nonzero part of the upper bound in Lemma 1, i.e., when $WE = SE = 0$. The upper bound in

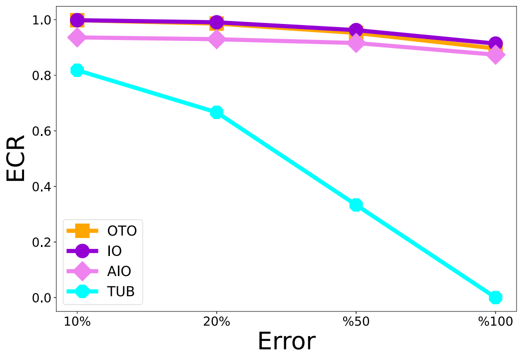

Fig. 7. ECR with BRE and only online prizes.

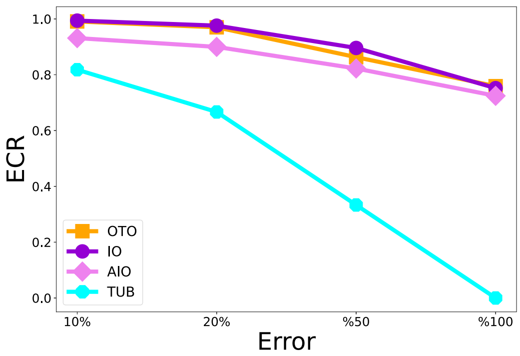

Fig. 8. ECR with FE and only online prizes.

Lemma 1 is the worst-case possible performance that an optimal online algorithm may have against a worst-case instance of the OCTOP. We have specifically demonstrated the characteristics of such worst-case instance in the proof of Lemma 1. Since the OTO and IO algorithms are of optimal competitive ratio according to Proposition 1 and Corollary 2, their ECR values must never fall below the upper bound in Lemma 1. This has been confirmed on the synthetic instances by our empirical experiments in Figs. 7 and 8, where the curve TUB represents the tight upper bound proved in Lemma 1. Interestingly, while the AIO algorithm is not of an optimal competitive ratio, its ECR values remain higher than the worst-case upper bound. Another notable observation from our experiments in this subsection is the fact that where the uncertainty on weight and service time is relaxed, i.e., where $WE = SE = 0$, the performance of the OTO algorithm improves significantly and this algorithm meets the performance of the IO algorithm. This is because in such scenarios all the assigned demand nodes to the vehicles in the first phase of the OTO algorithm, would be feasible collections in the second phase of the algorithm. A similar analysis on the CTOP instances is provided in Appendix D.

6.3.2. Impact of resource augmentation

In this section, we conduct an empirical analysis of resource augmentation, as introduced in Section 4.1. For this analysis, we focus on the Synthetic Instances over the FE scenarios and examine various error cases, including 10%, 20%, 50%, and 100%. Additionally, we consider time (α) and capacity (β) augmentations, ranging from 25% to 100%. Fig. 9 presents the average results over the 10 synthetic instances using the IO algorithm, i.e., we have focused on our best algorithm in this subsection.

The findings from our sensitivity analysis reveal a nuanced relationship between resource augmentation and solution quality. Specifically, increasing the time allocation for vehicles does not necessarily enhance the outcomes. Conversely, increasing vehicle capacity by 50% can improve our solution quality by approximately 35%. Notably, while solely extending time limits shows no positive effect, a combined augmentation of both time and capacity lead to better solutions. For a comprehensive analysis including the impact of error percentages within these augmentation scenarios, refer to Appendix E. This sensitivity analysis serves as a strategic tool within our overall methodology, enabling any given instance to be evaluated for resource value enhancement towards improving the ECR. It offers guidance on which resources (time or capacity in our problem setting) warrant prioritization by the management to optimize outcomes, which in our case investing on capacity augmentation appears to be the preferred choice. This analytical framework thus empowers decision-makers to allocate resources more effectively, ensuring targeted improvements in solution quality.

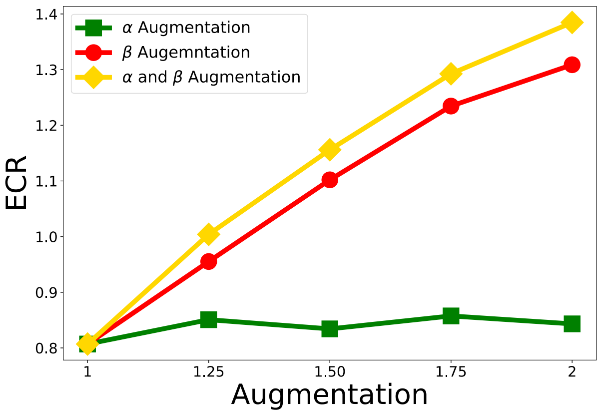

Fig. 9. Impact of resource augmentation; time and capacity denoted by α and β, respectively.

Table 5
Characteristics of the CTOP instances adopted from Archetti et al. (2009).

Instance	N	$\bar{P}$	$\bar{S}$	$\bar{W}$	K	T	C	min t_{ij}	max t_{ij}	$\bar{t_{ij}}$
B 1	100	14.09	10.00	14.58	4	100	100	1.41	91.83	33.95
B 2	50	15.22	10.00	15.54	4	100	100	2.24	85.63	32.43
B 3	75	17.69	10.00	18.19	4	100	100	2.24	85.28	33.23
B 4	100	14.09	10.00	14.58	4	100	100	1.41	91.83	33.95
B 5	150	14.63	10.00	14.90	4	100	100	0.00	91.83	33.47
B 6	199	15.32	10.00	16.01	4	100	100	0.00	91.83	32.91
B 7	120	10.73	10.00	11.46	4	100	100	0.00	114.98	53.62
B 8	100	17.10	10.00	18.10	4	100	100	1.00	96.18	39.47
B 9	150	14.39	10.00	14.90	4	100	100	0.00	91.83	33.47
B 10	199	15.41	10.00	16.01	4	100	100	0.00	91.83	32.91
mean	124.30	14.87	10.00	15.43	4	100	100	0.83	93.31	35.94
Std Dev	49.69	1.89	0.00	1.92	0	0	0	0.95	8.26	6.52

6.4. CTOP instances from the literature

Subsequently, we expanded our computational analysis by applying our online algorithms to the CTOP instances which are larger than the synthetic instances and have been extracted from Archetti et al. (2009). The travel times in CTOP instances also follow the triangle inequality. The characteristics of the CTOP instances are presented in Table 5. Similar to our analysis for the synthetic instances, we applied both the BRE and FE procedures to simulate the prediction errors for generating the realizations of the online inputs for the CTOP instances.

Before testing the performance of the IO algorithm on CTOP instances, in Section 6.4.1, we will first investigate the impact of γ from the node exclusion feature presented in 5.2.1.

6.4.1. Parameter tuning for CTOP instances

Figs. 10 and 11 show the impact of using varying γ values on the quality and total CPU time of the solutions obtained from the IO algorithm on the CTOP instances. Figs. 10 and 11 give the summary of the results over the BRE and FE scenarios, respectively. We recall that γ denotes the percentage of the nodes from V' that are considered in each iteration of the IO algorithm. In both of these figures, we can see that reducing γ does not necessarily lower the quality of the IO algorithm. However, reducing γ can reduce the run time of the algorithms significantly. As a result of this, for our computational experiments over the CTOP instances, we fixed the value of γ to 25%.

6.4.2. Results on the CTOP instances

Fig. 12 presents the comparison of the ECR of the three algorithms (OTO, IO, and AIO) under different prediction error scenarios that are generated using the BRE procedure (10%, 20%, 50%, and 100%) across ten CTOP instances taken from Archetti et al. (2009). We observe that while both IO and OTO exhibit declining performance with higher errors, AIO shows an improvement first when the error increases from 10% to 20% and then faces a slight decline. It is important to observe that despite this decline in the performance of the AIO, it is still able to find better solutions compared to OTO in the case of 50% errors, and better solutions compared to both IO and OTO in the case of 100% error. The reason why AIO slightly outperform IO in the case of 100% errors is perhaps because AIO does not use the node exclusion procedure and investigate all the nodes. Moreover, in the case of 100% error, obtained solutions are not reliable as they contain unreliable predictions.

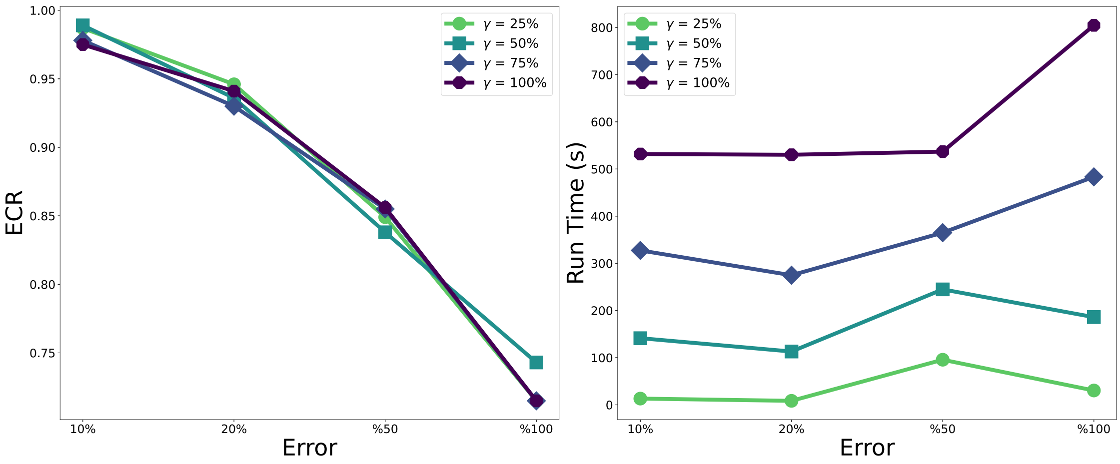

Fig. 10. Impact of γ on solution quality and runtime on BRE instances.

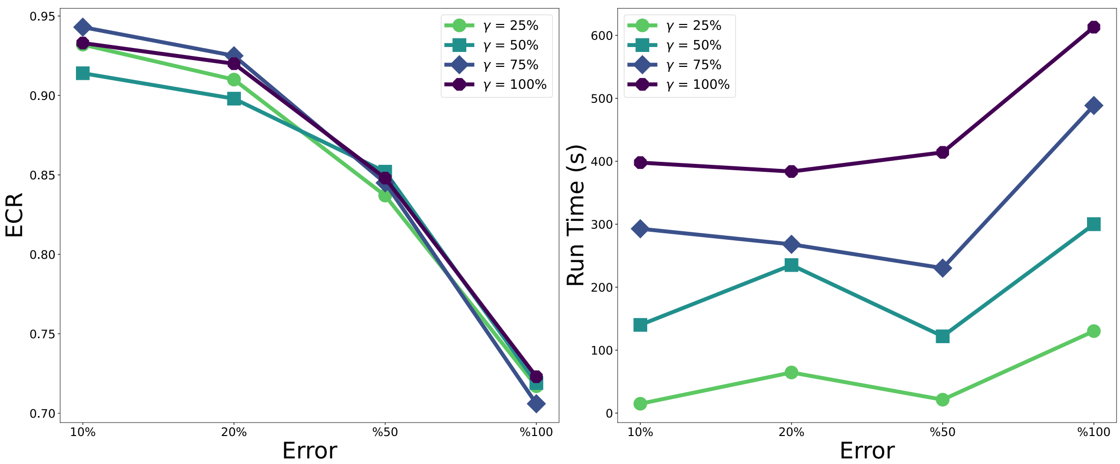

Fig. 11. Impact of γ on solution quality and runtime on FE instances.

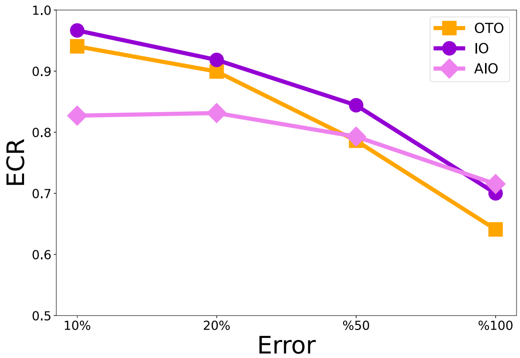

Fig. 12. ECR of CTOP instances under BRE.

Fig. 13 presents a comparison of simulation run times for the same 10 CTOP instances. In contrast to synthetic instances, where initial solution generation constitutes over 99% of the simulation run time, in the CTOP instances, as more nodes are considered, the run time of IO surpasses that of OTO. This underscores the importance of developing acceleration and node reduction techniques. We highlighted in Section 6.1.3 the distinction between simulation run time and iteration run time, noting that each iteration of the IO algorithm is solved within a maximum of 1 min.

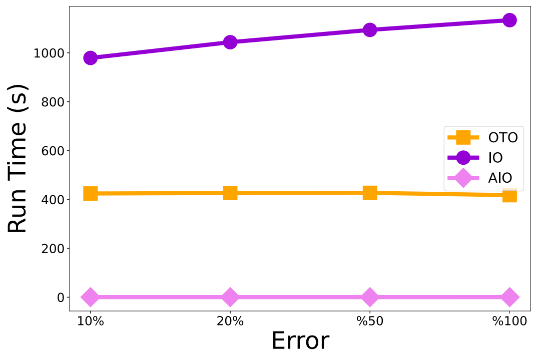

Fig. 13. Simulation runtime (s) of CTOP instances under BRE.

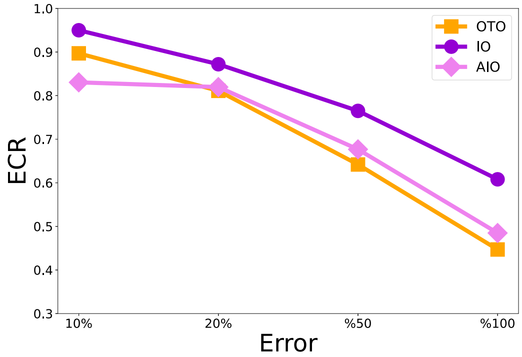

Fig. 14. ECR of CTOP instances under FE.

Figs. 14 and 15 show the average ECR and simulation run times for CTOP instances under varied prediction errors within the FE scenarios. A notable trend emerges in Fig. 14 where the AIO algorithm's performance exceeds that of OTO for errors beyond 10%. Despite this, the IO algorithm consistently outperforms both AIO and OTO across all levels of prediction error. Moreover, this analysis reveals that the gap in performance between IO and OTO widens compared to the BRE case. Furthermore, an increase in prediction error leads to a wider gap, which underscores the importance of leveraging online observations in enhancing decision-making processes.

In terms of simulation run time, it is evident that AIO completes its execution in just a few seconds, owing to its design based on polynomial time algorithms. For OTO, it is observed that prediction errors do not affect simulation run time, with the only variation being in the values used to find the initial solutions. The IO algorithm converges relatively quicker as the prediction error increases. Our conjecture is that with higher prediction errors, the fluctuating parameters may inadvertently simplify the optimization process in each iteration, allowing for more efficient identification of optimal solutions in our IO algorithm. Note that these observations pertain to a single instance according to FE scenarios rather than an average across multiple instances in our BRE scenarios, underscoring the specific context of this performance analysis.

7. Conclusions

In our study, we introduced the Online Capacitated Team Orienteering Problem (OCTOP), a problem that, to the best of our knowledge, has not been studied and addressed in the literature. Our analysis reveals that incorporating *capacity* constraints significantly complicates the problem which requires tailored solution methods. We design our online optimization algorithms to not only address the challenges posed by the capacity limitations, but also the uncertainty in the parameters. In this study, we focus on the uncertainties in prize, weight and service time for each demand node.

In our model, the true value of these uncertain parameters are revealed only when a vehicle visits each demand node. The online algorithm relies on initial point estimates up until that point. Our approach does not assume any inherent accuracy in these predictions, operating without any assurance of their precision. Thus, our online algorithms make decisions based on these initial estimates, effectively navigating the problem space without prior knowledge of the accuracy of the provided data.

We presented a comprehensive worst-case theoretical analysis for the OCTOP which is also valid for the special cases of this problem such as the OP and the TOP. We established a tight upper bound on the competitive ratio for online algorithms as delineated in Lemma 1, incorporating prediction accuracy as a variable through a set of online parameters (Definition 1). This analysis yields vital insights for decision-makers, particularly under conditions where predictions about location attributes such as weights and

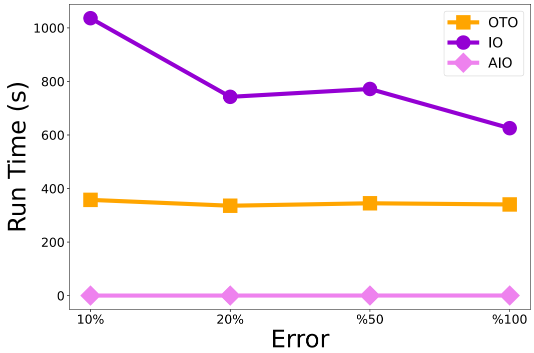

Fig. 15. Simulation runtime (s) of CTOP instances under FE.

service times are accurate or when the impact of constraints on weights and service times can be minimized. Notably, this tight upper bound on competitive ratio holds significance in security and police operations, especially when confronting adversarial misinformation regarding prize values at distinct target sites (Gupta et al., 2020). Moreover, we expanded our investigation to include a review of recent advancements in online optimization with predictions, offering tailored tight upper bounds on the novel concepts of resource augmentation, consistency, robustness, and smoothness within our online algorithms, as discussed in Section 4.1 and Section 4.2.

Next, we introduced three novel online algorithms, namely OTO, IO, and AIO algorithms. In Proposition 1 and Corollary 2, we proved that the OTO and IO algorithms have the optimality properties from a competitive ratio viewpoint. We then applied simulation to measure the performance of our algorithms empirically. In our simulations, prediction errors are utilized to generate scenarios, while the online algorithm relies exclusively on location-specific predictions, oblivious to the accuracy of predictions. This methodology allowed for a comprehensive analysis of the performance of our algorithms across a range of conditions, revealing the superior efficacy of the IO algorithm in most scenarios, as detailed in Section 6. We remark that the IO algorithm produces solutions close to optimal offline problem solutions where predictions are fairly accurate in most of the scenarios. The utilization of the node exclusion feature and the acceleration technique discussed in Section 5.2.1 and Section 5.2.2 enables the IO algorithm to provide real-time solutions for its iterative decisions, making it highly suitable for real-world applications.

Notably, whenever the predictions for demand weights and service times are accurate as tested in Section 6.3.1, or the capacity or time constraints are weakened as investigated in Section 6.4, the OTO algorithm yields results comparable to those of the IO algorithm. However, the OTO algorithm exhibits the least favorable outcomes among our algorithms when capacity and time constraints are stringent and prediction accuracy for demand and service time parameters is imperfect. This limitation of the OTO algorithm further underscores the practical suitability of the IO algorithm.

We designed and developed the AIO algorithm which is a polynomial time online algorithm based on the approximation-based algorithms introduced in Bock and Sanità (2015). While the ECR of the AIO algorithm falls below that of the IO algorithm on average, this algorithm consistently produces reasonable solutions compared to the other two algorithms in our empirical analysis. Specifically, on small-size synthetic instances, when the capacity and time constraints are tightened and the prediction error is high, the AIO algorithm surpasses the IO algorithm based on our observations in Section 6.2.

In closing, we identify several promising directions for future research. A primary avenue involves the development of an online algorithm with a theoretical competitive ratio guarantee that matches the computational run time of the IO algorithm, while also demonstrating superior empirical performance based on the ECR criterion. Notably, the IO algorithm exhibits an ECR decrease in scenarios with high prediction errors, which underscores the challenge of devising an alternative online algorithm capable of sustaining strong performance even under such conditions. As a second potential future research direction, we refer to analyzing a variant of the OCTOP where predictions are not available, emphasizing a lack of prior information about uncertain parameters until a vehicle visits a specific location.

Lastly, exploring the integration of a prediction mechanism directly into our online optimization framework presents a promising direction for future research, particularly in scenarios where high-quality datasets are accessible. This can lead to the development of predictive models that enhance the decision-making process under uncertainty, and accommodating more dynamic and precise adjustments to real-world uncertainties.

CRediT authorship contribution statement

Davood Shiri: Writing – review & editing, Writing – original draft, Methodology, Formal analysis, Conceptualization. **Vahid Akbari:** Writing – review & editing, Writing – original draft, Software, Methodology, Conceptualization. **Ali Hassanzadeh:** Writing – review & editing, Writing – original draft, Software, Methodology, Conceptualization.

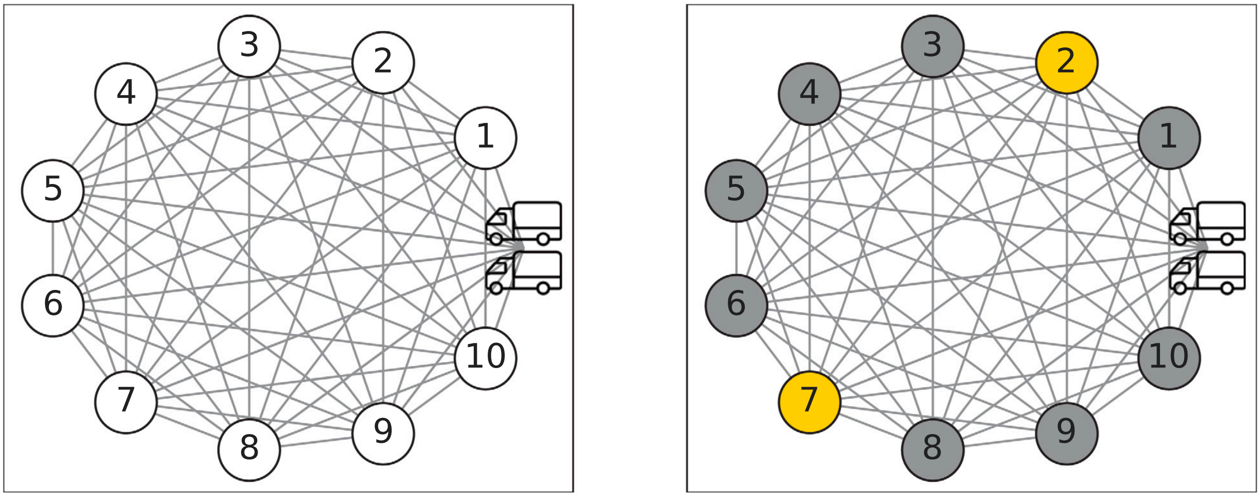

Fig. A.16. Illustration of the instance used in the proof of Lemma 1 for the case where $N = 10$ and $|K| = 2$: In the left figure, the instance inputted to an arbitrary online algorithm is depicted, where the two vehicles cannot distinguish between nodes based on the available predictions. In the right figure, the instance inputted to the offline algorithm is depicted, with the vehicles knowing exactly which two nodes have a higher prize.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Proof of Lemmas, Corollaries and Propositions

Proof of Lemma 1. For proving the upper bound, we follow the standard framework that has been conventionally applied in the literature of the online optimization problems (Ma et al., 2021; Akbari and Shiri, 2021; Ma and Simchi-Levi, 2020; Zhang et al., 2019; Jaillet and Wagner, 2008, 2006; Ausiello et al., 2001). That is, we first construct a problem instance and then prove the upper bound by analyzing the performance of online algorithms on the constructed instance. We remark that we design the instance as a function of prediction accuracy, i.e., the considered instance is parametric and varies based on the prediction errors for different online parameters.

The parametric error-dependent instance. We consider an instance where $|K| < N$ vehicles are initially positioned at the depot node 0. We set $\hat{W}_i = C$, $\hat{S}_i = t$, and $\hat{P}_i = 1$, for all $i \in V$. We let $t_{ij} = 0$ be the travel time between any two nodes in V_0 and we set $C_k = C$ and $T_k = t$ for $k \in K$. Note that, for any $i \in V$, W_i, S_i, P_i, WE_i, SE_i, and PE_i are online parameters and hence are unknown to the online algorithm. We let $V_* \subset V$ be a subset of exactly $|K|$ demand nodes such that, for each $j \in V_*$: $W_j = C$, $S_j = t$, and $P_j = \begin{cases} \frac{1}{1-PE} & \text{if } PE < 1 \\ M & otherwise \end{cases}$, where M is a sufficiently large number. For each node $l \in V \setminus V_*$: $W_l = C + \epsilon_W$, $S_l = t + \epsilon_S$, and $P_l = \frac{1}{1+PE}$, where $\epsilon_W, \epsilon_S \geq 0$. A schematic representation of this instance has been depicted in Fig. A.16 for the case when $N = 10$ and $|K| = 2$.

The analysis. Observe that since $T_k = t$, each vehicle can service at most one demand node in an arbitrary deterministic online algorithm, denoted by ALG_D. Therefore, ALG_D corresponds to a selection of exactly $|K|$ demand nodes from the N demand nodes, i.e., there are $\binom{N}{|K|}$ alternatives for ALG_D. We represent this selection by OPT_{ALG_D}. Note that, in the offline optimum, denoted by OPT, all the information is known a priori, hence all the $|K|$ nodes in V_* are serviced in OPT, i.e., the prize collected by OPT is either $\sum_{j \in V_*} \frac{1}{1-PE} = |K| \cdot (\frac{1}{1-PE})$ if $PE < 1$ or at least M if $PE \geq 1$. Before we proceed with the proof, we propose the following fact. Where OPT_{ALG_D} corresponds to servicing the nodes in $V \setminus V_*$ and $\epsilon_W = \epsilon_S = 0$ (i.e., when $WE_l = SE_l = 0$ for $l \in V \setminus V_*$), the prize collected by OPT_{ALG_D} equals $|K| \cdot \frac{1}{1+PE}$. Also, when $\epsilon_W > 0$ and/or $\epsilon_S > 0$ (i.e., when WE_l and/or SE_l are nonzero) the prize collected by OPT_{ALG_D} equals 0. We now complete the proof by analyzing deterministic and randomized algorithms separately.

- **Proof for deterministic algorithms.** For ALG_D, we consider the instance where OPT_{ALG_D} corresponds to servicing the nodes in $V \setminus V_*$.
- **Proof for randomized algorithms.** We let the $|K|$ demand nodes in V_* be uniformly distributed among the N demand nodes in V. We consider ALG_D against the instance and uniform probability distribution described above. With probability of $\prod_{i=0}^{|K|-1} \frac{N-|K|-i}{N-i}$ all the $|K|$ nodes serviced in OPT_{ALG_D} belong to $V \setminus V_*$. With probability of at most $1 - \prod_{i=0}^{|K|-1} \frac{N-|K|-i}{N-i}$ the prize of OPT_{ALG_D} would be greater than $|K| \cdot \frac{1}{1+PE}$. The lemma follows by Yao's principle (Yao, 1977) when N is sufficiently larger than $|K|$. □

Proof of Lemma 2. We extend the proof of Lemma 1 by applying the effect of resource augmentation on the same instance. Note that when $\epsilon_W \le (\beta-1)\cdot C$ and $\epsilon_S \le (\alpha-1)\cdot t$, the online algorithm will collect an (expected) prize of at most $|K|\cdot\frac{1}{1+PE}$. The Lemma follows since the offline optimum objective function value is $|K|\cdot\frac{1}{1-PE}$ when $PE<1$ and it is M otherwise, i.e., M is a sufficiently large number. □

Proof of Proposition 1. Note that when $WE\le\frac{\beta-1}{\beta}$, $SE\le\frac{\alpha-1}{\alpha}$ and $PE\ge 1$, the upper bound on the competitive ratio fixes at zero. Therefore, we propose the proof for the case where $WE\le\frac{\beta-1}{\beta}$, $SE\le\frac{\alpha-1}{\alpha}$ and $PE_i<1$ for all $i\in V$, i.e., we analyze the case where the solutions outputted by the OTO algorithm are always feasible, but their actual collected prize may deviate from their predicted collectable prize. Let V_* denote the set of nodes serviced by the OTO algorithm. The prize collected by the OTO algorithm would be at least $\sum_{i\in V_*}\frac{1}{1+PE_i}\cdot\hat{P}_i$, whereas the prize collected by the offline optimum would be at most $\frac{1}{1-PE}\cdot\sum_{i\in V_*}\hat{P}_i$ (see Definition 1 in Section 4). Since $\frac{(1-PE)\cdot\sum_{i\in V_*}\hat{P}_i}{\sum_{i\in V_*}(1+PE_i)\cdot\hat{P}_i}\ge\frac{1-PE}{1+PE}$, hence, the proposition. □

Proof of Proposition 2. Based on the same reason discussed in the proof of Proposition 1, we propose the proof for the case where $WE\le\frac{\beta-1}{\beta}$, $SE\le\frac{\alpha-1}{\alpha}$ and $PE_i<1$ for all $i\in V$. Let V_*^{OTO} and V_*^{IO} denote the set of nodes that are serviced by the vehicles in the solutions of the OTO and ALG algorithms, respectively. Note that $\sum_{j\in V_*^{OTO}}\frac{1}{1+PE}\cdot\hat{P}_j\le\sum_{l\in V_*^{IO}}\frac{1}{1+PE}\cdot\hat{P}_l$ due to Line 5:i of Procedure 2. The proof is complete since the objective function value of the offline optimum is not more than $\sum_{j\in V_*^{OTO}}\frac{1}{1-PE}\cdot\hat{P}_j$. □

Proof of Corollary 1. The proof follows based on Lemmas 1 and 2. □

Proof of Corollary 2. We note that incorporating the node exclusion feature will reduce the size of set V' in Procedure 2. Therefore, the total collected reward of the version of the IO algorithm with node exclusion feature would be greater than or equal to the total collected reward of the OTO algorithm. The proof follows similar to the proof of Proposition 2. □

Proof of Corollary 3. In the first stage of IO algorithm, instead of solving the exact formulation of the CTOP, we apply the 3.53-approximation algorithm. In the second stage, when solving iterations that involve a single-vehicle scenario, with different starting and ending points, we introduce a strategic adjustment to use the 3.53-approximation algorithm. Rather than tackling the sub-problem with varying depots, we standardize the starting and ending points at the actual depot. To navigate from the actual depot to various nodes, we assign a high travel time value, effectively deterring their selection, except for the current vehicle node, which is assigned a minimal travel time to facilitate its preference. Therefore, the AIO algorithm achieves a competitive ratio of at least $\frac{1}{3.53}$ times the competitive ratio of the IO algorithm. The proof is complete. □

Appendix B. A 3.53-approximation algorithm for CTOP

The approximation algorithm for the CTOP presented in Bock and Sanità (2015) uses a solution framework for the simpler Orienteering Problem (OP) in its routine, where the goal is to find the most profitable route between nodes in a network without exceeding a specific distance. This approximation algorithm reduces the OP to the k-stroll problem, where the objective is to find a path from a source to a destination by visiting at least k nodes with minimal path length. The procedure in the wrapper algorithm including k-stroll iterates through different values for k and the maximum allowed path length, marking paths that exceed this length as infeasible. In our implementation, we begin by assigning nodes to vehicles through a relaxed optimization that minimizes the weighted sum of the prize to service time and weight ratios, ensuring each node is allocated to exactly one vehicle. The longest round-trip path to the depot from this initial allocation sets the maximum path length, used as an upper bound for the optimal path length for each vehicle. The key to this approach is its reduction to a k-stroll problem, leveraging existing solutions and approximations to address the OP. The algorithm employs a bi-criteria approximation strategy that balances the number of nodes visited against the path length. It integrates proven approximation results for the k-stroll problem, specifically a $2+\epsilon$-approximation, which helps ensure that the path length does not exceed more than twice the optimal plus a small fraction determined by ϵ. By iteratively going through different sets of nodes, and checking the path length constraint, the algorithm refines the incumbent solution. For a more detailed procedure on the approximation algorithm on the OP see Chekuri et al. (2012).

The algorithm presented in Bock and Sanità (2015) expands the OP approximation algorithm by adding a capacity constraint to the vehicle, ensuring the overall collected demand from different nodes does not exceed the vehicle's capacity. This forms the Capacitated Orienteering Problem (COP), where the challenge is to maximize the profit while adhering to both distance and capacity constraints. The algorithm used for COP is then adapted to handle multiple vehicles in CTOP. Here, it applies a greedy approach, sequentially assigning routes to vehicles. It solves the COP for one vehicle at a time, removes the nodes this vehicle will visit from the problem, and repeats the process for the next vehicle. This continues until all vehicles are assigned routes or there are no remaining profitable nodes.

Appendix C. A heuristic algorithm for the offline CTOP

As we outlined in Section 5, the solution to the offline version of the CTOP is a key contributor to our online algorithms and an important input used in various stages of our solution methodology. Recall from Section 3.1 that in the offline version, we assume

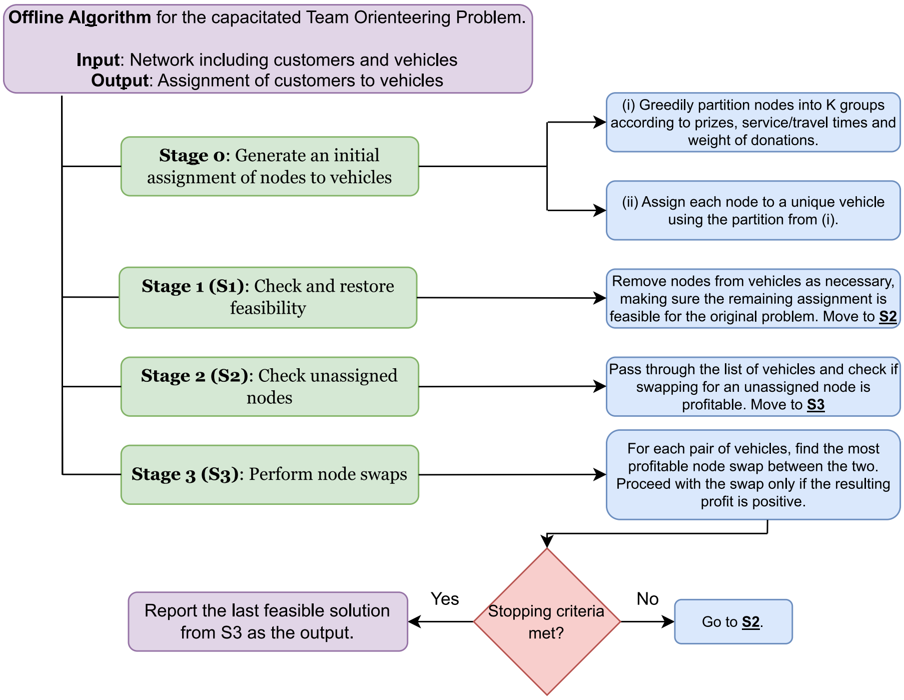

Fig. C.17. Different stages of our proposed algorithm for solving the offline version of CTOP.

all the decisions (i.e., assignment of customers to the vehicles and specific routes) are taken simultaneously and we have either an exact or point estimate value of all the parameters. The mathematical formulation of the offline CTOP is given in (4)–(16).

It is well-established in the literature that the offline CTOP is an NP-hard problem; see Laporte and Martello (1990) for a proof. Therefore, in solving the offline CTOP (or any of the variants of the OP for that matter), there is always a tradeoff between the degree of difficulty of the solution methodology and the quality of the solution. Taking into account this tradeoff for solving the offline problem, as part of our online algorithms, we propose an *improvement-type* heuristic algorithm which we believe it is the best compromise for the case at hand, in that, it produces descent quality offline solutions while using a reasonable amount of time and memory.

Fig. C.17 illustrates different stages of our proposed offline algorithm. At the beginning of the process, we solve a *multiple knapsack problem* with a simple two-phase greedy approach to assign each customer to a distinct vehicle. This initial step (i.e., Stage 0 as denoted in Fig. C.17) provides us with an initial solution, which may or may not be feasible due to the capacity constraints. In stage 1, we carefully pass through different combinations of member customers for each vehicle and we remove customers from each vehicle as necessary to restore feasibility. In stage 2, we pass through the list of unassigned (i.e., removed from their original vehicle in stage 1) customers, and we swap any unassigned customer with one of the assigned customers, only if the swap results in a positive overall profit gain. In stage 3 and the main part of the algorithm, we check every pair of vehicles, $\binom{K}{2}$ in total, and we look for potentially profitable customer swaps between those vehicles. We pass through all pairs at least once. At the end of stage 3, we check for the stopping criteria (i.e., either the optimality gap been less than a certain amount or the number of iterations of the algorithm between stages 2 and 3 exceeding a certain number) and proceed as depicted in Fig. C.17.

Once the stopping criteria are met, we calculate and report the final route within each vehicle as our final solution.

Appendix D. Revisiting Lemma 1 and Proposition 1 and Corollary 2 on the CTOP instances

Figs. D.18 and D.19 illustrate similar experiments on the CTOP instances. In these cases, IO does not exhibit superior performance to OTO when both weight and service time uncertainties are eliminated ($WE = SE = 0$). In the BRE instances, AIO achieves the worst-case performance in all the error cases across both BRE and FE scenarios. Similar to the synthetic instances, IO and OTO demonstrate comparable performances across various error scenarios for both BRE and FE cases.

Appendix E. Resource augmentation: Detailed results

In this section, we present summary of the results of testing the IO algorithm over synthetic instances and FE scenarios. Fig. E.20 depicts the impact of augmenting time (α), Fig. E.21 presents the summary of the results when augmenting capacity (β) and finally, Fig. E.22 illustrates the increase in ECR when augmenting both time and capacity.

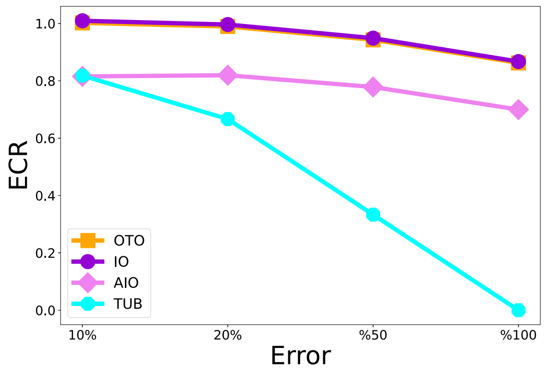

Fig. D.18. ECR with BRE and only online prizes.

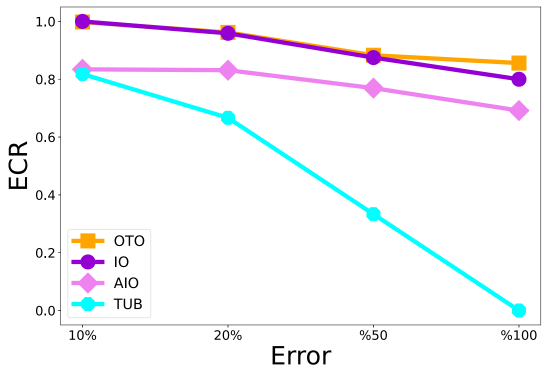

Fig. D.19. ECR with FE and only online prizes.

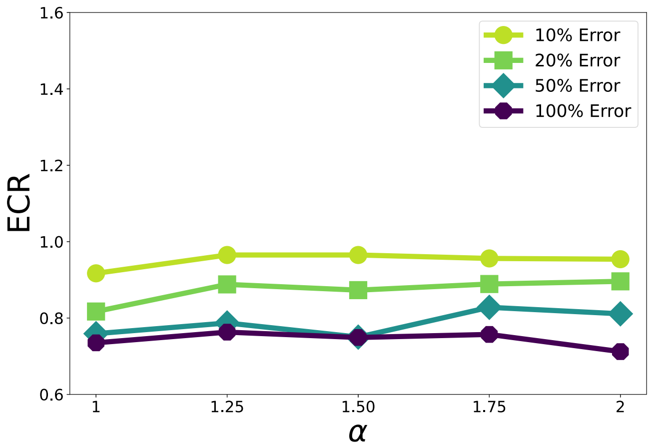

Fig. E.20. Impact of augmenting time on Synthetic Instances over FE scenarios.

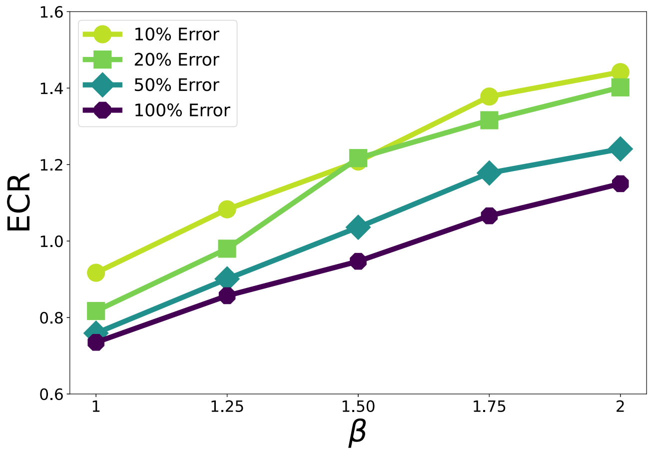

Fig. E.21. Impact of augmenting capacity on Synthetic Instances over FE scenarios.

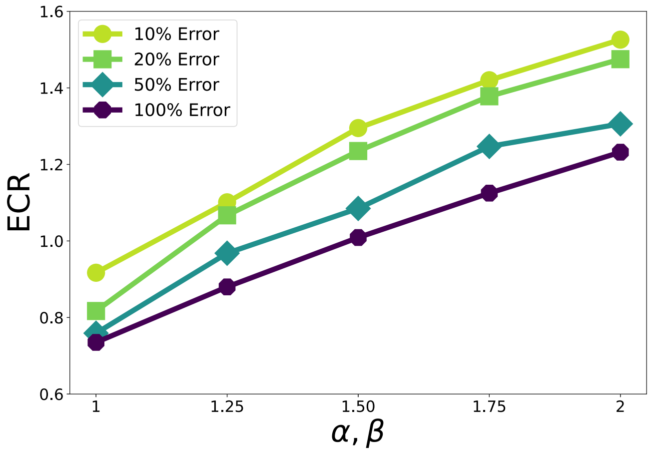

Fig. E.22. Impact of augmenting capacity and time on Synthetic Instances over FE scenarios.

References

Akbari, Vahid, Shiri, Davood, 2021. Weighted online minimum latency problem with edge uncertainty. European J. Oper. Res. 295 (1), 51–65.

Akbari, Vahid, Shiri, Davood, 2022. An online optimization approach for post-disaster relief distribution with online blocked edges. Comput. Oper. Res. 137, 105533.

Akbari, Vahid, Shiri, Davood, Sibel Salman, F., 2021. An online optimization approach to post-disaster road restoration. Transp. Res. B 150, 1–25.

Angelelli, Enrico, Archetti, Claudia, Filippi, Carlo, Vindigni, Michele, 2017. The probabilistic orienteering problem. Comput. Oper. Res. 81, 269–281.

Angelelli, Enrico, Archetti, Claudia, Filippi, Carlo, Vindigni, Michele, 2021. A dynamic and probabilistic orienteering problem. Comput. Oper. Res. 136, 105454.

Archetti, Claudia, Bianchessi, Nicola, Speranza, Maria Grazia, 2013. Optimal solutions for routing problems with profits. Discrete Appl. Math. 161 (4–5), 547–557.

Archetti, Claudia, Feillet, Dominique, Hertz, Alain, Speranza, Maria Grazia, 2009. The capacitated team orienteering and profitable tour problems. J. Oper. Res. Soc. 60, 831–842.

Ausiello, Giorgio, Bonifaci, Vincenzo, Laura, Luigi, 2008. The online prize-collecting traveling salesman problem. Inform. Process. Lett. 107 (6), 199–204.

Ausiello, Giorgio, Feuerstein, Esteban, Leonardi, Stefano, Stougie, Leen, Talamo, Maurizio, 2001. Algorithms for the on-line travelling salesman. Algorithmica 29 (4), 560–581.

Avraham, Edison, Raviv, Tal, 2023. The data-driven time-dependent orienteering problem with soft time windows. EURO J. Transp. Logist. 100112.

Balcik, Burcu, Yanıkoğlu, İhsan, 2020. A robust optimization approach for humanitarian needs assessment planning under travel time uncertainty. European J. Oper. Res. 282 (1), 40–57.

Bampis, Evripidis, Escoffier, Bruno, Gouleakis, Themis, Hahn, Niklas, Lakis, Kostas, Shahkarami, Golnoosh, Xefteris, Michalis, 2023. Learning-augmented online TSP on rings, trees, flowers and (almost) everywhere else. arXiv preprint arXiv:2305.02169.

Ben-Said, Asma, El-Hajj, Racha, Moukrim, Aziz, 2019. A variable space search heuristic for the capacitated team orienteering problem. J. Heuristics 25, 273–303.

Bertsimas, Dimitris, Jaillet, Patrick, Martin, Sébastien, 2019. Online vehicle routing: The edge of optimization in large-scale applications. Oper. Res. 67 (1), 143–162.

Best, Graeme, Hollinger, Geoffrey A., 2019. Decentralised self-organising maps for the online orienteering problem with neighbourhoods. In: 2019 International Symposium on Multi-Robot and Multi-Agent Systems. MRS, IEEE, pp. 139–141.

Bian, Zheyong, Liu, Xiang, 2018. A real-time adjustment strategy for the operational level stochastic orienteering problem: A simulation-aided optimization approach. Transp. Res. E 115, 246–266.

Blom, Michiel, Krumke, Sven O., de Paepe, Willem E., Stougie, Leen, 2001. The online TSP against fair adversaries. INFORMS J. Comput. 13 (2), 138–148.

Bock, Adrian, Sanità, Laura, 2015. The capacitated orienteering problem. Discrete Appl. Math. 195, 31–42.

Boussier, Sylvain, Feillet, Dominique, Gendreau, Michel, 2007. An exact algorithm for team orienteering problems. 4OR 5, 211–230.

Butt, Steven E., Cavalier, Tom M., 1994. A heuristic for the multiple tour maximum collection problem. Comput. Oper. Res. 21 (1), 101–111.

Campbell, Ann M., Gendreau, Michel, Thomas, Barrett W., 2011. The orienteering problem with stochastic travel and service times. Ann. Oper. Res. 186 (1), 61–81.

Chao, I-Ming, Golden, Bruce L., Wasil, Edward A., 1996a. A fast and effective heuristic for the orienteering problem. European J. Oper. Res. 88 (3), 475–489.

Chao, I-Ming, Golden, Bruce L., Wasil, Edward A., 1996b. The team orienteering problem. European J. Oper. Res. 88 (3), 464–474.

Chawla, Shuchi, Christou, Dimitris, 2023. Online time-windows TSP with predictions. arXiv preprint arXiv:2304.01958.

Chekuri, Chandra, Korula, Nitish, Pál, Martin, 2012. Improved algorithms for orienteering and related problems. ACM Trans. Algorithms (TALG) 8 (3), 1–27.

Chou, Xiaochen, Gambardella, Luca Maria, Montemanni, Roberto, 2021. A tabu search algorithm for the probabilistic orienteering problem. Comput. Oper. Res. 126, 105107.

Christman, Ananya, Forcier, William, Poudel, Aayam, 2018. From theory to practice: maximizing revenues for on-line dial-a-ride. J. Comb. Optim. 35 (2), 512–529.

Demange, Marc, Ellison, David, Jouve, Bertrand, 2021. Orienteering problem with time-windows and updating delay. Theoret. Comput. Sci. 863, 1–18.

Dolinskaya, Irina, Shi, Zhenyu Edwin, Smilowitz, Karen, 2018. Adaptive orienteering problem with stochastic travel times. Transp. Res. E 109, 1–19.

Evers, Lanah, Dollevoet, Twan, Barros, Ana Isabel, Monsuur, Herman, 2014a. Robust UAV mission planning. Ann. Oper. Res. 222, 293–315.

Evers, Lanah, Glorie, Kristiaan, Van Der Ster, Suzanne, Barros, Ana Isabel, Monsuur, Herman, 2014b. A two-stage approach to the orienteering problem with stochastic weights. Comput. Oper. Res. 43, 248–260.

Feuerstein, Esteban, Stougie, Leen, 2001. On-line single-server dial-a-ride problems. Theoret. Comput. Sci. 268 (1), 91–105.

Gong, Xiao-Yue, Goyal, Vineet, Iyengar, Garud N., Simchi-Levi, David, Udwani, Rajan, Wang, Shuangyu, 2022. Online assortment optimization with reusable resources. Manage. Sci. 68 (7), 4772–4785.

Gouleakis, Themistoklis, Lakis, Konstantinos, Shahkarami, Golnoosh, 2023. Learning-augmented algorithms for online TSP on the line. In: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, No. 10. pp. 11989–11996.

Gunawan, Aldy, Lau, Hoong Chuin, Vansteenwegen, Pieter, 2016. Orienteering problem: A survey of recent variants, solution approaches and applications. European J. Oper. Res. 255 (2), 315–332.

Gunawan, Aldy, Zhu, Jiahui, NG, Kien Ming, 2021. The capacitated team orienteering problem: a hybrid simulated annealing and iterated local search approach. In: Proceedings of the 13th International Conference on the Practice and Theory of Automated Timetabling-PATAT, Vol. 2.

Gupta, Anupam, Krishnaswamy, Ravishankar, Nagarajan, Viswanath, Ravi, R., 2015. Running errands in time: Approximation algorithms for stochastic orienteering. Math. Oper. Res. 40 (1), 56–79.

Gupta, Sushil, Starr, Martin K., Zanjirani Farahani, Reza, Ghodsi, Mahsa Mahboob, 2020. Prevention of terrorism–an assessment of prior POM work and future potentials. Prod. Oper. Manage. 29 (7), 1789–1815.

Hu, Hsiao-Yu, Wei, Hao-Ting, Li, Meng-Hsi, Chung, Kai-Min, Liao, Chung-Shou, 2022. Online TSP with predictions. arXiv preprint arXiv:2206.15364.

Ilhan, Taylan, Iravani, Seyed M.R., Daskin, Mark S., 2008. The orienteering problem with stochastic profits. IIE Trans. 40 (4), 406–421.

Irani, Sandy, Lu, Xiangwen, Regan, Amelia, 2004. On-line algorithms for the dynamic traveling repair problem. J. Sched. 7 (3), 243–258.

Jaillet, Patrick, Lu, Xin, 2011. Online traveling salesman problems with service flexibility. Networks 58 (2), 137–146.

Jaillet, Patrick, Lu, Xin, 2014. Online traveling salesman problems with rejection options. Networks 64 (2), 84–95.

Jaillet, Patrick, Wagner, Michael R., 2006. Online routing problems: Value of advanced information as improved competitive ratios. Transp. Sci. 40 (2), 200–210.

Jaillet, Patrick, Wagner, Michael R., 2008. Generalized online routing: New competitive ratios, resource augmentation, and asymptotic analyses. Oper. Res. 56, 745–757.

Juan, A.A., Freixes, A., Panadero, J., Serrat, C., Estrada-Moreno, A., 2020. Routing drones in smart cities: A biased-randomized algorithm for solving the team orienteering problem in real time. Transp. Res. Procedia 47, 243–250.

Karunakaran, Deepak, Mei, Yi, Zhang, Mengjie, 2019. Multitasking genetic programming for stochastic team orienteering problem with time windows. In: 2019 IEEE Symposium Series on Computational Intelligence. SSCI, IEEE, pp. 1598–1605.

Kobeaga, Gorka, Rojas-Delgado, Jairo, Merino, María, Lozano, Jose A., 2023. A revisited branch-and-cut algorithm for large-scale orienteering problems. European J. Oper. Res.

Kotiloglu, Serhan, Lappas, Theodoros, Pelechrinis, Konstantinos, Repoussis, P.P., 2017. Personalized multi-period tour recommendations. Tour. Manag. 62, 76–88.

Krumke, Sven O., De Paepe, Willem E., Poensgen, Diana, Stougie, Leen, 2003. News from the online traveling repairman. Theoret. Comput. Sci. 295 (1–3), 279–294.

Laporte, Gilbert, Martello, Silvano, 1990. The selective travelling salesman problem. Discrete Appl. Math. 26 (2–3), 193–207.

Liao, Zhixue, Zheng, Weimin, 2018. Using a heuristic algorithm to design a personalized day tour route in a time-dependent stochastic environment. Tour. Manag. 68, 284–300.

Luo, Zhixing, Cheang, Brenda, Lim, Andrew, Zhu, Wenbin, 2013. An adaptive ejection pool with toggle-rule diversification approach for the capacitated team orienteering problem. European J. Oper. Res. 229 (3), 673–682.

Ma, Will, Simchi-Levi, David, 2020. Algorithms for online matching, assortment, and pricing with tight weight-dependent competitive ratios. Oper. Res. 68 (6), 1787–1803.

Ma, Will, Simchi-Levi, David, Teo, Chung-Piaw, 2021. On policies for single-leg revenue management with limited demand information. Oper. Res. 69 (1), 207–226.

Pallottino, Stefano, 1984. Shortest-path methods: Complexity, interrelations and new propositions. Networks 14 (2), 257–267.

Panadero, Javier, Juan, Angel A., Bayliss, Christopher, Currie, Christine, 2020. Maximising reward from a team of surveillance drones: A simheuristic approach to the stochastic team orienteering problem. Eur. J. Ind. Eng. 14 (4), 485–516.

Panadero, Javier, Juan, Angel A., Ghorbani, Elnaz, Faulin, Javier, Pagès-Bernaus, Adela, 2023. Solving the stochastic team orienteering problem: comparing simheuristics with the sample average approximation method. Int. Trans. Oper. Res..

Papapanagiotou, Vassilis, Montemanni, Roberto, Gambardella, L., 2015. Hybrid sampling-based evaluators for the orienteering problem with stochastic travel and service times. J. Traffic Logist. Eng. 3 (2).

Papapanagiotou, Vassilis, Montemanni, Roberto, Gambardella, Luca Maria, et al., 2014. Objective function evaluation methods for the orienteering problem with stochastic travel and service times. J. Appl. Oper. Res. 6 (1), 16–29.

Reyes-Rubiano, L., Juan, A.A., Bayliss, Christopher, Panadero, Javier, Faulin, J., Copado, P., 2020. A biased-randomized learnheuristic for solving the team orienteering problem with dynamic rewards. Transp. Res. Procedia 47, 680–687.

Riera-Ledesma, Jorge, Salazar-González, Juan José, 2017. Solving the team orienteering arc routing problem with a column generation approach. European J. Oper. Res. 262 (1), 14–27.

Ruiz-Meza, José, Brito, Julio, Montoya-Torres, Jairo R., 2021. A GRASP to solve the multi-constraints multi-modal team orienteering problem with time windows for groups with heterogeneous preferences. Comput. Ind. Eng. 162, 107776.

Shi, Guangyao, Zhou, Lifeng, Tokekar, Pratap, 2023. Robust multiple-path orienteering problem: Securing against adversarial attacks. IEEE Trans. Robot..

Shirdel, G.H., Abdolhosseinzadeh, M., 2018. A simulated annealing heuristic for the online symmetric traveling salesman problem. J. Inf. Optim. Sci. 39 (6), 1283–1296.

Shiri, Davood, Akbari, Vahid, Salman, F. Sibel, 2020. Online routing and scheduling of search-and-rescue teams. OR Spectrum 42 (3), 755–784.

Shiri, Davood, Akbari, Vahid, Salman, F. Sibel, 2024. Online algorithms for ambulance routing in disaster response with time-varying victim conditions. OR Spectrum 1–35.

Shiri, Davood, Akbari, Vahid, Tozan, Hakan, 2023. Online optimisation for ambulance routing in disaster response with partial or no information on victim conditions. Comput. Oper. Res. 106314.

Sleator, Daniel, Tarjan, Robert, 1985. Amortized efficiency of list update and paging rules. Commun. ACM 28, 202–208.

Song, Yongjia, Ulmer, Marlin W., Thomas, Barrett W., Wallace, Stein W., 2020. Building trust in home services—stochastic team-orienteering with consistency constraints. Transp. Sci. 54 (3), 823–838.

Tang, Hao, Miller-Hooks, Elise, 2005a. Algorithms for a stochastic selective travelling salesperson problem. J. Oper. Res. Soc. 56 (4), 439–452.

Tang, Hao, Miller-Hooks, Elise, 2005b. A tabu search heuristic for the team orienteering problem. Comput. Oper. Res. 32 (6), 1379–1407.

Tarantilis, Christos D., Stavropoulou, Foteini, Repoussis, Panagiotis P., 2013. The capacitated team orienteering problem: a bi-level filter-and-fan method. European J. Oper. Res. 224 (1), 65–78.

Teng, S.Y., Ong, Hoon Liong, Huang, Huei Chuen, 2004. An integer L-shaped algorithm for time-constrained traveling salesman problem with stochastic travel and service times. Asia-Pac. J. Oper. Res. 21 (02), 241–257.

Tsiligirides, Theodore, 1984. Heuristic methods applied to orienteering. J. Oper. Res. Soc. 35 (9), 797–809.

Van Heeswijk, Wouter J.A., Mes, Martijn R.K., Schutten, Johannes M.J., 2019. The delivery dispatching problem with time windows for urban consolidation centers. Transp. Sci. 53 (1), 203–221.

Vansteenwegen, Pieter, Souffriau, Wouter, Berghe, Greet Vanden, Van Oudheusden, Dirk, 2009. Iterated local search for the team orienteering problem with time windows. Comput. Oper. Res. 36 (12), 3281–3290.

Vansteenwegen, Pieter, Souffriau, Wouter, Van Oudheusden, Dirk, 2011. The orienteering problem: A survey. European J. Oper. Res. 209 (1), 1–10.

Varakantham, Pradeep, Kumar, Akshat, 2013. Optimization approaches for solving chance constrained stochastic orienteering problems. In: Algorithmic Decision Theory: Third International Conference, ADT 2013, Bruxelles, Belgium, November 12-14, 2013, Proceedings 3. Springer, pp. 387–398.

Varakantham, Pradeep, Kumar, Akshat, Lau, Hoong Chuin, Yeoh, William, 2018. Risk-sensitive stochastic orienteering problems for trip optimization in urban environments. ACM Trans. Intell. Syst. Technol. 9 (3), 1–25.

Verbeeck, Cédric, Vansteenwegen, Pieter, Aghezzaf, E.-H., 2016. Solving the stochastic time-dependent orienteering problem with time windows. European J. Oper. Res. 255 (3), 699–718.

Voccia, Stacy A., Campbell, Ann Melissa, Thomas, Barrett W., 2019. The same-day delivery problem for online purchases. Transp. Sci. 53 (1), 167–184.

Wang, Xingyin, Golden, Bruce, Gulczynski, Damon, 2014. A worst-case analysis for the split delivery capacitated team orienteering problem with minimum delivery amounts. Optim. Lett. 8 (8), 2349–2356.

Yao, Andrew C., 1977. Probabilistic computations: Towards a unified measure of complexity. In: Proceedings of the 18th Annual IEEE Symposium on the Foundations of Computer Science. pp. 222—-227.

Yao, Canqi, Chen, Shibo, Yang, Zaiyue, 2022. Online distributed routing problem of electric vehicles. IEEE Trans. Intell. Transp. Syst. 23 (9), 16330–16341.

Yu, Qinxiao, Adulyasak, Yossiri, Rousseau, Louis-Martin, Zhu, Ning, Ma, Shoufeng, 2022a. Team orienteering with time-varying profit. INFORMS J. Comput. 34 (1), 262–280.

Yu, Qinxiao, Cheng, Chun, Zhu, Ning, 2022b. Robust team orienteering problem with decreasing profits. INFORMS J. Comput. 34 (6), 3215–3233.

Yu, Wei, Liu, Zhaohui, Bao, Xiaoguang, 2014. Optimal deterministic algorithms for some variants of online quota traveling salesman problem. European J. Oper. Res. 238 (3), 735–740.

Zhang, Guowei, Jia, Ning, Zhu, Ning, Adulyasak, Yossiri, Ma, Shoufeng, 2023. Robust drone selective routing in humanitarian transportation network assessment. European J. Oper. Res. 305 (1), 400–428.

Zhang, Shu, Ohlmann, Jeffrey W., Thomas, Barrett W., 2014. A priori orienteering with time windows and stochastic wait times at customers. European J. Oper. Res. 239 (1), 70–79.

Zhang, Shu, Ohlmann, Jeffrey W., Thomas, Barrett W., 2018. Dynamic orienteering on a network of queues. Transp. Sci. 52 (3), 691–706.

Zhang, Huili, Tong, Weitian, Lin, Guohui, Xu, Yinfeng, 2019. Online minimum latency problem with edge uncertainty. European J. Oper. Res. 273 (2), 418–429.