RAIRO-Oper. Res. 60 (2026) 625–642
https://doi.org/10.1051/ro/2025160

RAIRO Operations Research
www.rairo-ro.org

A TWO-STAGE APPROACH TO THE MULTIPLE-AGENT ORIENTEERING PROBLEM WITH STOCHASTIC WEIGHT AND CAPACITY CONSTRAINTS

Xinrui Liu[1], Xinggang Luo[1,2,*], Pengli Ji[3], Zhongliang Zhang[2] and Yuning Hu[2]

Abstract. The Multiple-Agent Orienteering Problem with Capacity Constraints (MAOPCC) is one kind of routing problem that finds applications in both tourism and transportation industries. The MAOPCC aims to find feasible routes with maximum profit while considering time constraints. In this paper, we extend the MAOPCC to the Multiple-Agent Orienteering Problem with Stochastic Weight and Capacity Constraints (MAOPCCSW) to address the uncertainty in practical situations. The problem is solved using a two-stage stochastic model with recourse and hard time constraints. The model considers the effect of stochastic weights on the expected total profit value during the modeling stage. The two-stage model is solved with Sample Average Approximation (SAA), which converges to the optimal solution with a high computational cost. Therefore, to solve large instances, a heuristic method is developed, which utilizes the problem structure of the MAOPCCSW and explicitly considers relevant uncertainties. In our experimental analysis, we demonstrate the effectiveness of the MAOPCCSW method, which outperforms both the standard deterministic method and the deterministic method amended with real-time information.

Mathematics Subject Classification. 90C59.

Received February 6, 2024. Accepted December 10, 2025.

1. Introduction

The Multiple-Agent Orienteering Problem with Capacity Constraints (MAOPCC) [30] is a variant of the Orienteering Problem (OP) [26], which involves developing one path through nodes associated with profits and arcs associated with weights, respectively. Different from the OP, in the MAOPCC, multiple routes are developed. Additionally, each node has a service time and a capacity constraint, and multiple visitors can collect profits from the same node. The capacity of a node represents the maximum number of visitors or users it can serve simultaneously. Figure 1 shows an illustrative example of the MAOPCC with three visitors and four attractions. In Plan 1, all visitors choose to start from Attraction A for the biggest profit. As the capacity limit of each attraction equals 2, visitor 1 in Plan 1 must wait while visiting Attractions A and C. In contrast, visitors in Plan 2 can visit more attractions with optimization. To solve the MAOPCC, Wang *et al.* [30] established a non-linear optimization model for the MAOPCC and presented two algorithms according to the characteristics

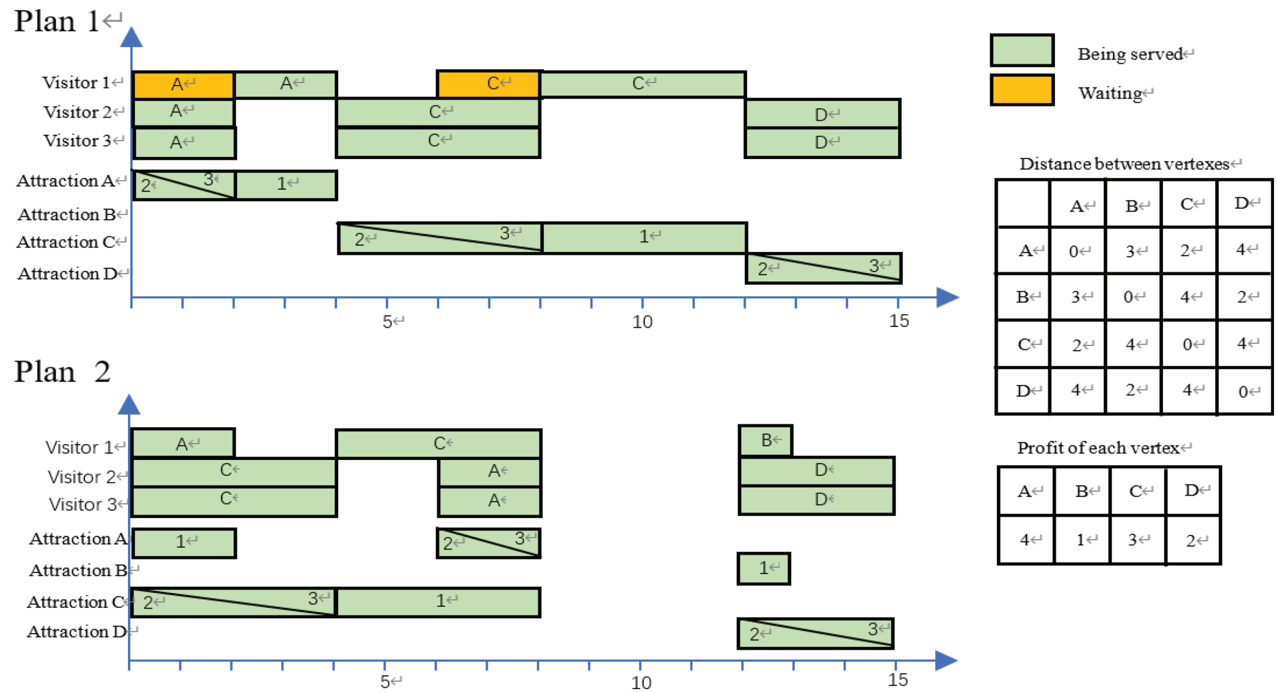
Keywords. Multiple-agent orienteering problem, stochastic, two-stage, sample average approximation.

[1] School of Management, Hangzhou Dianzi University, 310018 Zhejiang, P.R. China.

[2] Experimental Center of Data Science and Intelligent Decision, Hangzhou Dianzi University, Hangzhou 310018, P.R. China.

[3] Zhejiang Lab, 310018 Zhejiang, P.R. China.

*Corresponding author: xgluo@hdu.edu.cn



Distance between vertexes

	A	B	C	D
A	0	3	2	4
B	3	0	4	2
C	2	4	0	4
D	4	2	4	0

Profit of each vertex

A	B	C	D
4	1	3	2

FIGURE 1. An illustrative example of the MAOPCC.

of the problem. Based on the work of Wang *et al.* [30], Luo *et al.* [17] developed a MILP model of the MAOPCC and presented more efficient algorithms to solve it.

Tourism is one of the most important application fields of the MAOPCC [17]. In a tourism system, several attractions with capacity are located in different places within a region. Optimizing visitor itineraries helps reduce waiting times and enables them to visit more attractions within a limited time. The MAOPCC has applications in various scenarios, such as itinerary arrangements for tourist attractions, arranging visit itineraries for theme park excursions, and mapping out routes for touring museums. The MAOPCC can also be applied in other fields. For example, it can be used to schedule freight orders for truck drivers, optimizing their delivery routes under warehouse capacity limit to reduce the waiting time of drivers and improve the transportation efficiency of freight companies.

In practice, the weights of arcs of the MAOPCC are often difficult to be accurately estimated. For example, the travel time between two nodes in a city is influenced by the traffic, morning and evening peaks, weather circumstances, traffic accidents, and other factors that can cause travel time uncertainty. Under uncertainty, the solution of a deterministic problem may be ineffective because it is suboptimal or infeasible. Therefore, it is necessary to consider the MAOPCC with uncertainty.

Recent studies have addressed the multiple-agent trip planning problem under various constraints [16, 17, 22, 27]. However, none have explored this problem under uncertainty. To bridge this gap, we investigate the MAOPCC with stochastic weights, where each arc possesses a known probability distribution. We refer to this variant as the MAOPCC with Stochastic Weights (MAOPCCSW). This study extends the deterministic MAOPCC model by incorporating stochastic elements. Unlike the conventional one-stage deterministic formulation, we establish a two-stage recourse model for the MAOPCCSW, in which the first-stage decisions are constrained by a second-stage optimization problem [1]. To solve the MAOPCCSW, we introduce two solution approaches: (1) the Sample Average Approximation (SAA), a classic methodology in stochastic programming; and (2) a problem-specific heuristic algorithm designed to exploit the MAOPCCSW's structure. Computational experiments demonstrate the effectiveness of both approaches, highlighting their advantages over deterministic MAOPCC solutions and their extensions.

The main contributions of this work are summarized as follows: (1) We establish the mathematical model for the MAOPCCSW, which is an extension of the deterministic MAOPCC; (2) To facilitate solving the problem, we develop a two-stage stochastic model with recourse and hard time constraints, and utilize the SAA method to solve the model; (3) As the SAA method cannot solve large scale problem, we develop a heuristic algorithm which can solve large problem within reasonable time.

The rest of the paper is structured as follows. Section 2 reviews the related research. In Section 3, the MAOPCCSW is formulated, and a two-stage model is established. The SAA method applied to the two-stage model is described in Section 4. The heuristic approach for solving the MAOPCCSW is described in Section 5. In Section 6, the results of computational experiments are provided. Concluding remarks are given in Section 7.

2. Literature review

The vehicle routing problem (VRP) is one of the classical routing problems, which is indeed the generalized version of the traveling salesman problem (TSP) [9]. Moreover, the OP can be seen as a selective VRP, in which the weights of arcs represent travel cost or resource consumption (travel time, service time, or fuel consumption), and the profit of nodes represents the importance or revenues of them. The aim is to find a path that maximizes the profits collected by visiting the selected nodes within the time limit that confines the total weights used in the route. The OP is a classic operations research problem, and its solving algorithms include exact and heuristic ones. For details on these algorithms, please refer to the overview papers of Vansteenwegen *et al.* [28] and Gunawan *et al.* [10].

Different from the OP, several paths with different time limits are developed simultaneously and the capacity limit of each node is considered in the MAOPCC. To solve this problem, Wang *et al.* [30] established a nonlinear model, and designed a Branch and Bound (B&B) algorithm, a Sequential Algorithm (SA), and a Probabilistic Iterated Local Search (PILS) for solving the MAOPCC. Based on the work of Wang *et al.* [30], Luo *et al.* [17] developed a MILP model of the MAOPCC. They designed a new B&B algorithm and a Variable Neighborhood Search (VNS) algorithm based on the linear model to solve the MAOPCC and verified the performance of the algorithms through numerical experiments.

To our knowledge, there is still no existing literature to study the MAOPCCSW. However, there are some studies related to the OP with stochastic weights (OPSW). In the following content, we briefly review the existing methods used to solve the OPSW and its variants, which are classified as stochastic programming and robust optimization.

2.1. Stochastic programming

Stochastic programming is used to handle uncertainty and randomness in decision-making. Tang and Miller-Hooks [24] proposed a chance-constrained model for the OPSW, in which, based on the discrete probability distribution, the total duration of travel must meet the time limit with a predefined probability. They proposed an exact algorithm and a heuristic algorithm to solve the model. Teng *et al.* [25] and Campbell *et al.* [4] introduced the recourse model of the OPSW. In the two-stage recourse model for the routing problem, the first-stage decision develops the initial routes based on the uncertain weights between nods. This initial planning is then recourse in the second stage, where the actual costs are realized and evaluated against the predefined routes. The second-stage recourse cost evaluates the penalty incurred by the nodes that cannot be visited in the actual situation of the second-stage. This model aims to strike a balance between optimality in the first-stage and adaptability in the second-stage, making robust decisions under uncertainty. Based on this, Teng *et al.* [25] considered the penalty proportional to the profit exceeding the time limit as the second-stage cost and developed an integer L-shaped solution method to solve the problem. Similar to Teng *et al.* [25], Campbell *et al.* [4] also considered the penalty of profits on the nodes which cannot be visited in the second-stage before the deadline. Time constraints in their work were considered as soft: even if the total weight of the tour exceeds the total time limit, nodes can still be accessed. They developed an exact algorithm and a variable neighborhood search heuristic algorithm for solving the model. In addition to the penalty function, there is another fundamental difference between the models proposed by Teng *et al.* [25] and Campbell *et al.* [4]. Specifically, Teng *et al.* [25] restricted their analysis to discrete probability distributions exclusively. On the other hand, Campbell *et al.* [4] expanded the scope of their model by incorporating various continuous probability distributions, accommodating a wider range of uncertainties where the stochastic weights can follow continuous probability. Evers *et al.* [7] presented an SAA approach for the OPSW and a heuristic for solving it. They proposed a profit penalty which is the total profit of the inaccessible nodes in the first-stage, but the profit penalty is not involved in this model, which allows the solution to involve all the nodes that are possible to reach, even if the probability of reaching some of those nodes is very low. Solak *et al.* [23] presented multi-stage and two-stage models under endogenous uncertainty. They used an SAA with Lagrangian relaxation and decomposition for decomposing the original model into several subproblems and developed a sub-gradient algorithm for solving the subproblems. However, they treated the uncertain variables as discrete variables and divided them into several levels, which would influence the feasibility judgment as dividing travel time into levels might make feasible solutions become infeasible in the MAOPCC. Apap and Grossmann [3] developed a model which has endogenous and exogenous uncertainties at the same time, called multi-stage stochastic programming. They presented several propositions which can greatly reduce the number of non-anticipativity constraints. They developed a sequential scenario decomposition heuristic to solve the problem. Similar to Solak *et al.* [23], they also used Lagrangian relaxation, decomposition, and sub-gradient methods. Ilhan *et al.* [13] proposed an exact parametric solution technique and a Pareto-based bi-objective genetic algorithm for solving the OP with stochastic profit. Different from Evers *et al.* [7], Ilhan *et al.* [13] tried to maximize the probability that the sum of profit is greater than or equal to a predefined target. Therefore, the mean and variance of the profit are the two main factors to be considered. They used the convex efficient frontier to handle the two factors. Wang *et al.* [29] also treated the weight of arcs as an uncertain variable. They proposed the uncertain team OP with time windows and used uncertainty theory to solve it. They developed a formula to describe the uncertainty of inserting one node into the route. However, inserting one node into the route of one visitor might influence the time use of other visitors for the MAOPCCSW, which increases the difficulty of describing the time constraint with uncertain theory. Similar to the multi-stage model presented by Solak *et al.* [23], Dolinskaya *et al.* [5] presented the adaptive orienteering problem with stochastic travel times (AOPST). They treated the uncertainty problem as a dynamic problem and divided it into planning and execution problems. Furthermore, they divided the planning problem into a master problem which selects the subset of the nodes, and a pathfinding subproblem which finds a path with the founded nodes subset. In the execution problem, they updated the solution with the dynamically updated data. Hasannia Kolaee and Mirzapour Al-e Hashem [15] developed a practical optimization model for the Medical Tour Centers (MTCs) and used a scenario-based two-stage stochastic optimization approach to address the uncertainty. They developed a meta-heuristic which combined Progressive Hedging Algorithm (PHA) and Genetic Algorithm (GA) to solve the problem.

2.2. Robust optimization

Evers *et al.* [6] presented a robust optimization method for the OP. Their approach ensured solutions were robust against varying weight uncertainties and handled all weight realizations within a predefined uncertainty set. However, planned profits may reduce due to conservative worst-case considerations. The trade-off involves balancing robustness and profitability based on risk tolerance and business needs. Specific agile policies are combined to respond to the impact of weight implementation during itinerant execution. Makui *et al.* [18] developed a robust optimization model to deal with the uncertainty and utilized a Benders decomposition algorithm to tackle large-scale computational complexity. Aazami and Saidi-Mehrabad [2] and Saeedi Mehrabad *et al.* [19] studied the supply chain problems and proposed different algorithms to solve the corresponding problem. Heydari and Aazami [11] studied a job shop scheduling problem and used the ε-constraint method to solve it. Ke *et al.* [14] proposed a proportion based on a robust optimization method that assumes that a certain proportion of uncertainty coefficients in each solution can be changed. The novel algorithm optimizes the deterministic model to achieve a balance between optimality and feasibility when the coefficients change. They solved the team orienteering problem with interval data with the method and proved the usefulness of the proposed robust optimization approach with numerical experiments. Yu *et al.* [32] solved a robust variant of the team orienteering problem with decreasing profits. Assuming that the service time of customers is uncertain and the goal is to develop robust routes for vehicles to maximize collected profits, they proposed a two-index robust formulation based on the constraints of dynamic programming recursive equation. They proposed the Branch and Price algorithm and Tabu search algorithm to solve the problem and demonstrated the effectiveness of the algorithms.

3. Two-stage approach

First, a nominal mathematical model for the MAOPCCSW is established in Section 3.1, which is consistent with the formula used by Luo *et al.* [17]. Next, a two-stage model for the MAOPCCSW is developed in Section 3.2, which is easier to solve. In order to unify the statements of weight and time, we use traveling time to represent weight in the following paper.

3.1. Optimization of the MAOPCCSW

Consider a set of nodes H and visitors N and let $|H|$ and $|N|$ be their cardinalities, respectively. Note that H does not includes the initial departure location (attraction 0) and the final destination (attraction $|H|+1$). Let $H^+ = H \cup \{0\} \cup \{|H|+1\}$ be the set of nodes with starting point and ending point. Let tl_i be the time limite of visitor i, st_j and SC_j be the serving time and the serving load set of node j, respectively. Let R_{ik} be the profit for visitor i visiting attraction k. For each arc (i,j), we associate an uncertain value ξ_{ij} representing the random traveling time of it.

The central decision variables of the mathematical model are outlined as follows: x_{ijk}: $x_{ijk}=1$ if visitor i uses the route from attraction j to attraction k, and 0 otherwise.

The additional decision variables in the model are outlined as follows: z_{ijm}: $z_{ijm}=1$ if visitor i utilize resource m of attraction j, and 0 otherwise. q_{iljm}: $q_{iljm}=1$ if visitor i is arranged before visitor l on the resource m of attraction j, and 0 otherwise. t_{ij}: The time of service beginning for attraction j for visitor i.

The formulation of the MAOPCCSW is presented below:

$$\max \sum_{i\in N}\sum_{j\in H^+}\sum_{k\in H^+} R_{ik}x_{ijk} \tag{1}$$

Subject to:

$$t_{ij} \le tl_i \qquad \forall i \in H,\ \ \forall j \in H^+ \tag{2}$$

$$\sum_{k\in H^+\setminus\{0\}} x_{i0k} = 1 \qquad \forall i \in N \tag{3}$$

$$\sum_{j\in H^+\setminus\{|H|+1\}} x_{ij(|H|+1)} = 1 \qquad \forall i \in N \tag{4}$$

$$\sum_{j\in H^+\setminus\{|H|+1\}} x_{ijk} = \sum_{j\in H^+\setminus\{0\}} x_{ikj} \le 1 \qquad \forall i \in N,\ \ \forall k \in H,\ \ j \ne k \tag{5}$$

$$\sum_{k\in H^+} x_{ijk} = \sum_{m\in \mathrm{SC}_j} z_{ijm} \qquad \forall i \in N,\ \ \forall j \in H,\ \ j \ne k \tag{6}$$

$$t_{ik} - t_{ij} \ge M(x_{ijk}-1) + \xi_{jk} + st_j \qquad \forall i \in N,\ \ \forall j \in H^+ \setminus \{|H|+1\},\ \ \forall k \in H^+ \setminus \{0\},\ \ j \ne k \tag{7}$$

$$q_{iljm} + q_{lijm} \ge z_{ijm} + z_{ljm} - 1 \qquad \forall i,l \in N,\ \ \forall j \in H,\ \ \forall m \in \mathrm{SC}_j,\ \ i \ne l \tag{8}$$

$$q_{iljm} + q_{lijm} \le 0.5 \times (z_{ijm} + z_{ljm}) \qquad \forall i,l \in N,\ \ \forall j \in H,\ \ \forall m \in \mathrm{SC}_j,\ \ i \ne l \tag{9}$$

$$t_{lj} - t_{ij} \geq st_j - M\left(1 - \sum_{m\in \mathrm{SC}_j} q_{iljm}\right) \qquad \forall i,l \in N,\ \ \forall j \in H,\ \ i \neq l \tag{10}$$

$$t_{ij} \geq 0 \qquad \forall i \in N,\ \ \forall j \in H^+ \tag{11}$$

where M is a large positive number. The model aims to maximize the total profit collected by visitors. Constraint (2) guarantees that the time for every visitor returns to the ending point does not violate the time constraint. Constraints (3) and (4) guarantee that all visitors commence their routes from the starting point and ultimately reach the ending point. Constraint (5) guarantees the flow consistency of each visitor and preserves that each attraction is visited only once by any visitor. Constraint (6) guarantees that when a visitor visits an attraction, he accesses only one of the available resources. Constraint (7) ensures that the visitation of an attraction by a visitor is bound by the time of their previous attraction visit. Constraints (8) and (9) enforce an order relation for two visitors to use the same resource at the same attraction. Constraint (10) ensures that a visitor starts to be served after the previous visitor who visited the same attraction and used the same resource. Besides, the presence of any subloop within the routes is effectively avoided through constraint (10). Constraint (11) guarantees that the commencement time for each visitor to receive services at each attraction is strictly greater than 0. This model can be used to solve deterministic MAOPCC and is called the deterministic model in this paper.

3.2. Two-stage approach

We develop a two-stage model for the MAOPCCSW based on the idea of the recourse model proposed by Evers *et al.* [7] and Solak *et al.* [23]. In the first-stage of the model, routes are developed, and in the second-stage, recourse actions are implemented based on the realization of traveling time uncertainty. If certain nodes in the first-stage routes cannot be visited due to the actual realization of traveling time uncertainty, the sum of the profits of those inaccessible nodes is referred to as the profit shortage. In practice, each attraction that is assigned in routes and cannot be visited in the second-stage will disappoint visitors and affect their satisfaction. Therefore, similar to Teng *et al.* [25], we adopt the profit penalty to prevent inserting redundant nodes in routes.

The two-stage model aims to maximize the profit in the first-stage, considering the profit shortage caused by uncertain traveling time realizations. In the second-stage, the constraint guarantees that the time consumption for each visitor is not larger than the upper bound becomes a soft constraint, which means the actual time for any visitor to go to the ending point can be larger than the upper bound. However, in the two-stage model, visitors are not allowed to continue visiting node j when the expected time to reach the ending point from node j is larger than the bound.

ξ_{jk} is used to denote the arc realization of the random variables between any two attractions j and k. For notational convenience, ξ^l is the set of all the arc realizations of the lth scenario, and ξ is the set of all of the ξ^l. The two-stage model is developed as follows:

$$\max \sum_{i\in N}\sum_{j\in H^+}\sum_{k\in H^+} R_{ij}x_{ijk} + E_\xi(v(x,\xi^l)) \tag{12}$$

s.t. Constraints (4)–(11).

Where $v(x,\xi^l)$ is the function of profit shortage for routes x and traveling time realizations ξ^l. $E_\xi(v(x,\xi^l))$ denotes the expected value of function $v(x,\xi^l)$. The constraint of time bound is ineffective in two-stage model as profit punishment prevents counting redundant nodes in routes of the first-stage.

For the given x and ξ^l, let $t_{ij}^{\xi^l}$ be the corresponding variable of t_{ij} under ξ^l and α be the penalty coefficient, then profit shortage can be calculated through the following optimization subproblem, where

$$y_{ij}^{\xi^l} = \begin{cases} 1 & \text{if node } j \text{ is in the route of visitor } i, \\ & \text{but cannot be visited in the second stage under } \xi^l \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

(RECOURSE)

$$v\left(x, \xi^l\right) = \max \sum_{i \in N} \sum_{j \in H^+} R_{ij} \left(-\alpha y_{ij}^{\xi^l}\right) \tag{14}$$

subject to

$$M\left(x_{ijk}-1\right) - M y_{ik}^{\xi^l} + t_{ij}^{\xi^l} + st_j + \xi_{jk} + st_k + \xi_{k(|H|+1)} \leq tl_i \qquad \forall i \in N, \quad \forall j,k \in H^+ \tag{15}$$

$$y_{ik}^{\xi^l} \geq y_{ij}^{\xi^l} + x_{ijk} - 1 \qquad \forall i \in N, \quad \forall j,k \in H^+ \tag{16}$$

$$t_{ik}^{\xi^l} - t_{ij}^{\xi^l} \geq M(x_{ijk}-1) + \xi_{jk} + st_j \qquad \forall i \in N, \quad \forall j,k \in H^+ \tag{17}$$

$$t_{lj}^{\xi^l} - t_{ij}^{\xi^l} \geq st_j - M\left(1 - \sum_{m \in \mathrm{SC}_j} q_{iljm} + y_{ij}^{\xi^l}\right) \qquad \forall i,l \in N, \quad \forall j \in H^+ \tag{18}$$

$$t_{ij}^{\xi^l} \geq 0 \qquad \forall i \in N, \quad \forall j \in H^+. \tag{19}$$

Similar to constraints (7)–(10) in the deterministic model, constraints (17) and (18) calculate the start time under ξ^l with decision variable x_{ijk} and q_{iljm}, which have been determined in the first-stage. Note that, in constraint (18), if $y_{ij}^{\xi^l}$ equals 1, which means visitor i cannot occupy the resource of node j, then visitor i will not influence the start time of visitor l who visits node j after visitor i. Constraint (15) finds inaccessible nodes through the calculation of the expected time for going to the ending point. One node is inaccessible for a visitor if the expected return time after being served at the node is larger than the time limit. Constraint (16) ensures that all nodes assigned after the node which cannot be visited in the second-stage are inaccessible. Constraint (19), the same as the deterministic model, ensures that starting time should be non-negative.

4. Sample average approximation

Sample Average Approximation (SAA) is used for solving stochastic optimization problems. It uses a set of sampled scenarios to approximate the objective value of the stochastic problem [20]. Let $\xi^1, \ldots, \xi^S$ be the set of S identically distributed realizations of ξ. Then, the expected shortage can be estimated through the average profit of the S scenarios:

$$g_S(x) = \frac{1}{S} \sum_{l=1}^{S} v\left(x, \xi^l\right). \tag{20}$$

Then, our SAA model can be expressed as below:

$$\max \sum_{i \in N} \sum_{j \in H^+} \sum_{k \in H^+} R_{ij} x_{ijk} + \left(\frac{1}{S} \sum_{l=1}^{S} \sum_{i \in N} \sum_{j \in H^+} R_{ij} \left(-\alpha y_{ij}^{\xi^l}\right)\right) \tag{21}$$

subject to Constraints (3)–(6), Constraints (8) and (9), Constraints (15)–(19).

Let v^* and v_S be the true and estimated objective value of the two-stage model, respectively. According to Solak *et al.* [23], v_S is a valid upper statistical bound of v^*, and according to Shapiro [21], v_S converges to v^* when $S \to \infty$, if conditional sampling is used. As a consequence, selecting a larger value of S yields a more accurate approximation of the true objective function. Nevertheless, since the computational complexity of the SAA problem grows exponentially with S, it may be more efficient to select a smaller sample size S, and solve several SAA problems with i.i.d. samples.

Let P denotes the total number of SAA problems that have been solved, and let v_S^p and x_S^p, $p = 1, 2, \ldots, P$, represents the optimal solution of the pth replication. We assume that x_S^p represents the first-stage decision, which is of practical importance for the routing problem. Once a feasible solution $x_S^p \in \mathcal{X}$ is obtained by solving the SAA problem, a new sample $\xi^1, \ldots, \xi^{S'}$ can be obtained, and $g_{S'}(x_S^p)$ can be approximated by the unbiased estimator:

$$g_{S'}\left(x_S^p\right) = \frac{1}{S'} \sum_{l=1}^{S'} v\left(x_S^p, \xi^l\right) \tag{22}$$

where S' is typically larger than S, as the computational time required to estimate the objective value for a given solution will be less than that required to solve the SAA problem. On the other hand, this phase is still difficult as it requires solving a multi-stage problem with endogenous uncertainty where only the first-stage decisions are known, hence, any solution procedure must be efficient in calculating $g_{S'}(x_S^p)$. Additionally, one would be interested in estimating the quality of x_S^p. The optimality gap can be calculated to estimate the solution quality: $v^* - g_{S'}(x_S^p)$, where $g_{S'}(x_S^p)$ is calculated in (22), and v^* can be approximated by $\bar{v}_S^P$:

$$\bar{v}_S^P = \frac{1}{P} \sum_{p=1}^{P} v_S^p. \tag{23}$$

The sampling procedure can be terminated once the optimality gap estimate is sufficiently small or after performing all P replications, and the best solution among the SAA solutions can be selected using an appropriate criterion. However, the variance of the optimality gap estimator is also important, and must be considered in determining the quality of a solution:

$$\left(\sigma^2_{\bar{v}_S^P} + \sigma^2_{g_{S'}\left(x_S^p\right)}\right)^{1/2} \tag{24}$$

where $\sigma^2_{\bar{v}_S^P}$ and $\sigma^2_{g_{S'}(x_S^p)}$ are the estimates of the variances for the estimators of v^* and $g_{S'}(x_S^p)$, respectively, which are calculated as:

$$\sigma^2_{\bar{v}_S^P} = \frac{1}{(P-1)P} \sum_{p=1}^{P} \left(v_S^p - \bar{v}_S^P\right)^2 \tag{25}$$

$$\sigma^2_{g_{S'}\left(x_S^p\right)} = \frac{1}{(S'-1)S'} \sum_{l=1}^{S'} \left(v\left(x_S^p, \xi^l\right) - g_{S'}\left(x_S^p\right)\right)^2. \tag{26}$$

The expected value calculated through function (22) can be directly used to evaluate the quality of the solutions. Additionally, the standard deviation helps assessing the fluctuation of the solutions under various scenarios.

The overall procedure of the SAA algorithm is presented in Figure 2. To find a solution, routes for visitors are enumerated in the first stage and the expected profit shortage is calculated through S scenarios in the second stage. After finding P solutions, S' scenarios are used to obtain the best solution among the P solutions.

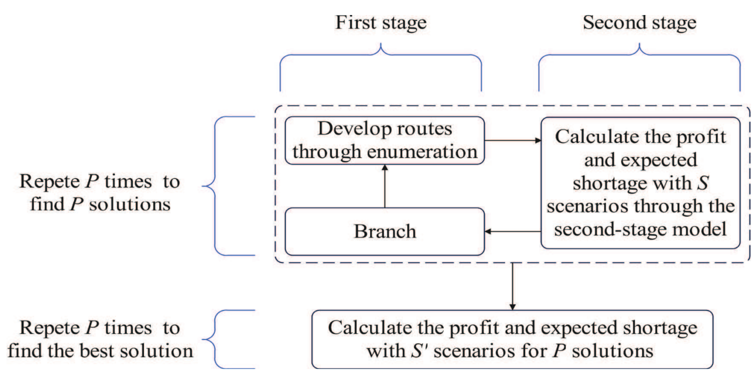


FIGURE 2. Sample average approximation process.

5. Variable neighborhood search

Due to the reliance on solving large Mixed-Integer Linear Programs (MILPs), the solution to the SAA problem, as described in the previous section, can only be obtained within a reasonable computation time for small instances. A heuristic method is developed to address this limitation and solve the MAOPCCSW for larger instances. This heuristic takes advantage of the specific structure of the MAOPCCSW, enabling more efficient and practical solutions for larger problem sizes. The heuristic for solving the MAOPCCSW is based on the framework proposed by Luo *et al.* [17]. Besides, the heuristic method utilizes the elements of two heuristic methods: the randomization concept inspired by Tsiligirides [26] and the scoring metric introduced by Golden *et al.* [8]. These two elements have shown promising results in the heuristic for the deterministic OP. With suitable adjustments, they can effectively handle the uncertainties in solution construction and evaluation. By incorporating these elements into the MAOPCCSW heuristic, we gain valuable insights into the impact of uncertainty on solutions and obtain reliable solutions.

Typically, the VNS algorithm commences by utilizing the initial solution generated through the constructive heuristic. The local search procedure tries to improve the solution with neighborhood structures. To prevent the solution from falling into a local optimum, the shaking procedure randomly deletes some nodes of the solution for further improvement.

5.1. Solution representation

We use $|H|+|N|$ partial permutations to represent the solution with $|N|$ visitors and $|H|$ attractions. The route of a visitor, $T_k(k=1,\ldots,|N|)$, is expressed as a permutation of attractions (visitor-link). Owing to the resource limit constraint, each visitor is served in a certain order at every attraction, which is expressed by a permutation of visitors (resource-link), $R_l(l=1,\ldots,|H|)$. When a visitor visits an attraction, it has a position on the visitor-link and resource-link at the same time. For example, visitor i, who starts from the starting node, visits Attraction 1 and Attraction 2 by sequence and finally returns to ending point, has a visitor-link $T_i=\{1,2\}$. Correspondingly, because of the resource limit of each attraction, visitor i must be presented in the resource-link of Attraction 1 $R_1=\{i\}$.

5.2. Construction phase

The constructive heuristic begins with an empty solution and progressively inserts solution components into the current partial solution to achieve a well-optimized initial solution. As the MAOPCCSW involves complex constraints within its model, the constructive heuristic algorithm must guarantee feasibility while adding new components in partial solution. Typically, during each iteration, the constructive heuristic utilizes diverse formulations to evaluate and choose solution components for insertion into the current solution.

We adopt the Fast_LH algorithm proposed by Luo *et al.* [17] to solve the deterministic MAOPCC. The Fast_LH adds a new attraction to the route of the visitor who has finished their visit to another attraction and places it in the last position of the route. Note that, the attraction is only inserted on the last position of the corresponding visitor-link in each iteration to reduce computation time. The Fast_LH uses the neighbor profit of an attraction j_1 to select the next attraction for a visitor to visit, which is the total profit of the attractions in set $\delta^{\xi}_{j_1}$. However, in the MAOPCCSW, the evaluation also needs to consider the uncertainty of the solution. Therefore, like the SAA approach, we generate S scenarios for evaluation, as expressed as follows:

$$j_1^s = \operatorname{argmax}\left(\sum_{l=1}^{S}\sum_{j\in\delta^{\xi^l}_{j_1}} R_j/\Delta^{\xi}_{j_1}\right), \qquad j_1\in J_1 \tag{27}$$

where $\delta_{j_1}^{\xi}$ is the set which consists of indices representing attractions that are not visited within a certain range of attraction j_1 under realizations set ξ, which is determined by

$$\delta_{j_1}^{\xi} = \{j|\xi_{j_1 j} \leq r, j \in J^u\} \tag{28}$$

$$r = D^{\xi}/|H|^2 \tag{29}$$

where J_1 constitutes a collection of indices denoting the attractions that are currently available for insertion. $\Delta_{j_1}^{\xi} = \xi_{mj_1} + st_{j_1} + w_{j_1}^{\xi}$, ξ_{mj_1} represents the uncertain traveling time between the previous node m on the visitor-link and the node j_1 to be inserted, st_{j_1} denotes the duration or processing time needed for visitors to visit the attraction, and $w_{j_1}^{\xi}$ is the waiting time with realizations set ξ. J^u comprises attractions that have not been visited. D^{ξ} refers to traveling time summation of all possible pairs of attractions under ξ. As the traveling time of arcs are uncertain, S' scenarios are used in the construction procedure to deal with the uncertainty. By using the Fast_LH algorithm, nodes are continually inserted into the visitor-links until there are no insertions left that can increase the profit estimated by S' scenarios.

5.3. Improvement phase

The improvement phase starts by utilizing the initial solutions produced through the constructive heuristic. When a certain number of iterations are completed without finding a better solution, the algorithm initiates the shaking procedure to jump out the local optimum. The algorithm stops when it employs multiple rounds of the shaking procedure and does not find any better solution. In the improvement phase, we adopt the inserting-based strategy (IBS) which is presented by Luo *et al.* [17] as the local search procedure of VNS. In this section, the profit density of a visitor-link is the ratio of the profit of the visitor-link and the time it uses. Critical profit density is a specified threshold of profit density used to classify visitor-links. If the profit density of a visitor-link exceeds the critical profit density, it is considered as a developed link; otherwise, it is categorized as an undeveloped link. An undeveloped link with the lowest profit density is defined as the selected link. The main idea of the IBS is that, because of the low profit density of the selected link, it has more potential to be improved. therefore, the algorithm would focus on the selected link and try to improve it. In the preparation procedure for each iteration, if the selected link becomes a developed link in the last iteration, a new link is selected for development in the current iteration.

5.3.1. Neighborhood structures

There are four neighborhood structures in IBS: Inserting, Changing, Deleting, and Swapping.

Inserting. This neighborhood structure randomly selects attractions not yet visited by the selected visitor-link and inserts them into the last positions of both the selected link and the corresponding resource-link. The time constraint of the selected link can be violated in this structure.

Changing. At the beginning of the process, this neighborhood structure selects a single attraction from the selected link with equation (30):

$$j_1^s = \operatorname{argmax}\left(\sum_{j\in\delta_{j_1}^{\xi}} P_j/\Delta_{j_1}^{\xi}\right), \qquad j_1 \in J' \tag{30}$$

where J' includes the indices of all attractions located in the selected link; $\delta_{j_1}^{\xi}$ is the set containing the indices of attractions that have been visited and located within a specific range of j_1; $\Delta_{j_1}^{\xi} = \xi_{ij_1}^{l} + \xi_{j_1 k}$; i and k are the indexes of attractions, where i is the pre-order node of attraction j_1 and k is the post-order node of attraction j_1.

After selection, the position of j_1^s on the selected link and corresponding resource-link are changed to all possible positions. If the modification leads to a reduction in the time required for the selected link to complete its routing, the modified solution is accepted. Similar to Inserting, this process also allows the time-bound constraints of the selected links to be violated and repeats Changing process if one modified solution is accepted.

Deleting. In this neighborhood structure, attractions are removed from the end of the selected link one by one until the total time utilized by the link no longer exceeds the predefined time limit.

Swapping. This neighborhood aims to achieve one of the objectives by exchanging attractions that have been visited with those that remain unexplored within a randomly selected visitor-link: (1) increase the total profit collected by the visitor, (2) if the profits of the visited and non-visited attractions are equal, this neighborhood seeks to reduce the total travel time of the corresponding link. This structure ends when no feasible swapping is attainable or a single feasible swapping has been successfully identified for one of the visitor-links.

For each iteration of the local search, the four neighborhood structures are launched consecutively and try to search for a better solution. Similar to the construction phase, S' scenarios are used to deal with the uncertainty and evaluate each modification. The new solution is accepted when the average profit under S' scenarios is larger than the old one.

Algorithm 1. Local_search(Current Solution (CS)).

```
CS' = Inserting(CS, selected_link);
CS' = Changing(CS', selected_link);
CS' = Deleting(CS', selected_link);
if CS' > CS or (reward density of CS == reward density of CS') then
    CS = CS';
else if CS' == CS and (reward density of CS > reward density of CS') then
    CS = CS';
end if
CS' = Swapping(CS);
if CS' > CS or (reward density of CS == reward density of CS') then
    CS = CS';
else if CS' == CS and (reward density of CS > reward density of CS') then
    CS = CS';
end ifreturn CS
```

5.3.2. Shaking

During the shaking procedure, the solution is randomly modified to jump out from potential local optima. There are two different types of shaking in the shaking process: local shaking and overall shaking. The local shaking process is utilized to jump out from the local optima if the selected link fails to become developed after certain number of iterations. After a specific number of local shakings, if the algorithm cannot find a better solution, the overall shaking process will be launched.

Local shaking. As a first step, the allocated time limit for the selected link is interchanged with the time restriction of a different undeveloped link, characterized by a comparatively smaller time constraint. If the selected link remains undeveloped after the exchange, this procedure will randomly remove several attractions which are arranged in the selected link. Additionally, in this procedure, a tabu list for the selected link is cleared which is used to avoid repeated insertion of one attraction during the Inserting step.

Overall shaking. In a manner similar to the previous procedure, this procedure randomly eliminates nodes that have been visited by each visitor. Subsequently, the heuristic algorithm randomly inserts unvisited attractions into each visitor-link.

6. Numerical experiments

In this section, we will assess the advantages of considering uncertainty in the MAOPCCSW. We will employ the SAA formulation outlined in Section 3 and the MAOPCCSW heuristic presented in Section 5 to solve the small case of the MAOPCCSW. For large instances, the two-stage model with SAA cannot provide solutions within reasonable computational time. Therefore, only the MAOPCCSW heuristic is used in this circumstance. First, we will provide a description of the data utilized in our experiments. Following that, the settings of the SAA method and the MAOPCCSW heuristic are discussed. Finally, we will discuss the results from both approaches. For comparison, deterministic solution approach developed by Luo *et al.* [17] is also used in the stochastic environment, denoted as the MAOPCC approach in the tables. Similar to Luo *et al.* [17], we use the exact algorithm for the smallest instances and heuristic algorithm for other instances to solve the deterministic model (denoted as MAOPCC in tables). Besides the deterministic approach, similar to Evers *et al.* [7], we also use re-optimization (Re-Opt), which utilizes real-time information. When a visitor reaches the ending point, we will try to add nodes between the ending point and the pre-order point regarding the expected traveling time. The SAA method and heuristic method are implemented in Java on Intel (R) Core (TM) i7 CPU, 2.4 GHz, and 8.00 GB RAM. CPLEX 12.1 optimizes MIP.

6.1. Data

We use the datasets generated by Luo *et al.* [17]. They developed 30 instances and divided them into 3 datasets. The scales of these datasets are 7 nodes (including 5 attractions) with 10 visitors, 12 nodes (including 10 attractions) with 20 visitors, 22 nodes (including 20 attractions) with 30 visitors, and 32 nodes (including 30 attractions) with 40 visitors. For each scale, 10 instances were generated as a group. The ID format "number of nodes-number of visitors-case sequence number" was used to represent these instances. Among these datasets, instances with 12 nodes are small instances and the others are large instances. Since these instances are deterministic, we assume that the traveling time is normally distributed with a standard deviation of 15% of the expected value.

6.2. Parameter setting

For the SAA procedure, same as Evers *et al.* [7], in each run, we use $S = 8$ scenarios to estimate the expected profit shortage. This procedure is repeated for $P = 10$ replications to find the best first-stage solution in these SAA problems. To evaluate the first-stage solutions of P replications, we use $S' = 1000$ scenarios to test the solutions. For the VNS algorithm, we employed the same algorithm parameters as in Luo *et al.* [17]. For specific details, please refer to Luo *et al.* [17].

6.3. Experiments

Firstly, we will outline the process of obtaining the results from the SAA method for a given instance. Next, the characteristics of using the two-stage model with the SAA process and the MAOPCCSW heuristic method and the performance of related routes are discussed based on the experiment results. Throughout the presentation of results, all reported values will be rounded to two decimals for clarity and consistency.

To illustrate the results of the two-stage model, we use the second instance of seven nodes with a normal distribution traveling time. Table 1 displays the results from ten separate runs of the SAA method. In the first column, the objective value of each run is presented. According to the definition of Objective (21), this is the average value of overall profit based on eight scenarios in each run. The overall objective function of our proposed two-stage recourse model consists of two components. The first component is the first-stage profit, while the second component is the average recourse cost incurred in the second-stage. Both of these quantities are listed in the second and third columns, respectively. It is important to emphasize that the objective value and the cost of the second-stage are approximated using a limited set of eight scenarios in a given run. To better estimate the expected profit, we simulate 1000 scenarios to get the expected shortage. Columns 4 and 5 contain the calculated values representing the expected profit shortage and expected profit, respectively. The

TABLE 1. SAA results for instance 7-10-2.

Obj.	First-stage profit	Second-stage cost	Exp. Shortage	Exp. Profit
54.43	55.00	0.58	1.10	53.90
55.80	65.00	9.20	12.34	52.66
48.90	65.00	16.10	17.28	47.72
42.00	65.00	23.00	23.07	41.93
49.00	50.00	1.00	1.07	48.93
53.50	65.00	11.50	12.48	52.52
53.50	65.00	11.50	12.58	52.42
47.35	60.00	12.65	22.43	37.58
53.90	70.00	16.10	19.99	50.01
51.20	65.00	13.80	18.77	46.23

value in column 5 represents the expected profit, combining the profit from the first-stage with the expected recourse cost. Note that, we use the B&B algorithm proposed by Luo *et al.* [17] for getting the optimal solution of small-size instances for the deterministic model, re-optimization model, and two-stage model with the SAA method.

The two-stage model, as described earlier, is applied to ten different instances of the first data set with seven nodes. The best solutions found by the two-stage model are listed in the fifth column of Table 2. Table 2 also contains the results of the MAOPCCSW heuristic, which are displayed in the fourth column.

As shown in Table 2, compared with the solution obtained from the deterministic MAOPCCs, the Re-Opt strategy, the two-stage model with SAA and heuristic algorithm have better performance on the profit. Especially, for the Re-Opt strategy, the re-inserting procedure for each scenario increases the profit and standard deviation at the same time as different scenarios would insert different attractions and cause the increasing of variance. The two-stage model with the SAA method and heuristic for solving the MAOPCCSW always outperforms the deterministic and Re-Opt method in the aspect of expected profit. Moreover, the two-stage model tends to exhibit a higher level of stability and predictability in terms of profit values. In most instances, the standard deviation of the total profit obtained from the two-stage tour is comparatively lower than the standard deviation observed in the expected profit from both the deterministic and Re-Opt tours. Comparing the two-stage model with the MAOPCCSW heuristic, the MAOPCCSW heuristic uses S' scenarios when constructing routes while the two-stage model with the SAA method only uses S scenarios in the construction phase. Therefore, as shown in the fourth and fifth column, the routes got from heuristic algorithm have better performance on expected profit value and standard deviation. Another notable advantage of heuristics is their ability to significantly reduce computation time compared to the two-stage model: the heuristic algorithm can solve all instances of data set 1 within 1000 seconds, while the entire two-stage model (based on ten runs) may take about ten hours. Increasing the number of scenarios in the two-stage model results in longer computation times but leads to convergence towards the actual best-expected profit. Additionally, the two-stage model offers the advantage of providing an upper bound estimation for the solution.

Tables 3 and 4 contain the results for large-size instances in which the two-stage model with the SAA method cannot get a solution within reasonable computational time. Different from Table 2, the variable neighborhood search algorithm proposed by Luo *et al.* [17] is used to develop the route for the deterministic MAOPCC (since the exact algorithm is unable to get a solution within reasonable computational time), and the computing times are displayed in Figures 3 and 4. As shown in these figures, the H-MAOPCCSW algorithm exhibits superior computational efficiency across most large-scale instances. Furthermore, the time advantage of H-MAOPCCSW over comparative methods becomes more pronounced as the instance scale increases, highlighting its scalability. As for the quality of solutions, compared to the deterministic MAOPCC tour, the Re-Opt policy does not yield substantial advantages in terms of expected profits. This is also related to the tour construction method of

TABLE 2. SAA and heuristic results for the first dataset.

Instance	MAOPCC			MAOPCCSW Re-Opt			H-MAOPCCSW			Two-stage model with SAA			Two-stage model UB
	Expected profit (standard deviation)												
7-10-1	115.80	(7.55)	14 743.71 s	116.28	(8.22)	14 769.22 s	118.22	(6.73)	0.97 s	117.87	(6.44)	170.19 h	120.36
7-10-2	48.93	(9.44)	193.61 s	50.01	(10.23)	230.65 s	52.42	(9.38)	1.57 s	53.90	(9.28)	17.99 h	50.66
7-10-3	133.72	(10.66)	1655.58 s	133.87	(11.64)	1665.64 s	145.50	(8.87)	0.35 s	134.30	(7.12)	19.66 h	159.87
7-10-4	209.23	(12.92)	667.37 s	211.94	(14.43)	670.14 s	220.93	(9.66)	0.40 s	213.19	(8.32)	18.03 h	225.58
7-10-5	47.05	(5.01)	11 367.16 s	46.83	(6.00)	11 371.27 s	47.10	(3.78)	1.18 s	47.10	(3.78)	150.34 h	48.54
7-10-6	182.15	(10.24)	1199.41 s	182.88	(10.48)	1211.41 s	187.45	(7.14)	0.34 s	185.04	(6.44)	15.33 h	191.37
7-10-7	72.42	(7.82)	13 516.8 s	73.39	(8.35)	13 565.65 s	75.31	(4.91)	0.45 s	74.22	(4.53)	177.50 h	76.21
7-10-8	107.52	(11.32)	6078.53 s	108.39	(12.43)	6124.76 s	110.53	(8.06)	0.27 s	109.19	(7.50)	69.34 h	113.57
7-10-9	135.17	(9.67)	14 365.06 s	136.53	(10.07)	14 372.76 s	139.51	(8.72)	0.42 s	137.20	(8.55)	170.61 h	141.98
7-10-10	44.53	(2.92)	14 433.93 s	44.60	(3.60)	14 478.04 s	44.69	(2.61)	0.28 s	44.69	(2.61)	170.53 h	45.24

TABLE 3. Heuristic results for the second dataset.

Instance	MAOPCC			MAOPCCSW Re-Opt			H-MAOPCCSW		
	Expected profit (standard deviation)								
12-20-1	460.59	(7.65)	32.53 s	461.74	(8.43)	39.65 s	471.61	(3.86)	17.70 s
12-20-2	357.17	(2.99)	44.77 s	357.27	(4.53)	52.98 s	364.49	(1.09)	17.32 s
12-20-3	440.00	(11.96)	10.56 s	441.09	(11.90)	17.80 s	449.38	(7.92)	14.71 s
12-20-4	321.56	(9.84)	15.81 s	321.82	(6.84)	19.03 s	328.88	(6.12)	9.42 s
12-20-5	609.09	(5.98)	140.43 s	610.50	(9.30)	150.03 s	629.82	(4.91)	72.77 s
12-20-6	571.43	(8.35)	9.51 s	573.14	(11.62)	9.90 s	574.18	(7.24)	7.97 s
12-20-7	594.22	(6.25)	37.67 s	596.05	(10.04)	46.81 s	598.30	(5.56)	20.68 s
12-20-8	864.70	(5.30)	23.17 s	864.88	(8.28)	29.28 s	875.88	(5.13)	17.46 s
12-20-9	490.86	(3.04)	52.26 s	492.37	(3.46)	53.42 s	492.62	(1.03)	29.46 s
12-20-10	644.24	(4.60)	29.49 s	645.87	(3.97)	31.90 s	663.46	(1.97)	18.87 s

TABLE 4. Heuristic results for the third dataset.

Instance	MAOPCC			MAOPCCSW Re-Opt			H-MAOPCCSW		
	Expected profit (standard deviation)								
22-30-1	1599.32	(19.54)	4900.47 s	1601.87	(20.82)	4981.73 s	1609.83	(16.71)	1091.47 s
22-30-2	1419.30	(12.99)	2006.99 s	1420.07	(13.29)	2083.82 s	1424.49	(6.54)	928.22 s
22-30-3	1481.95	(9.24)	3910.45 s	1484.32	(10.51)	4074.82 s	1487.80	(7.50)	1797.57 s
22-30-4	1607.94	(12.75)	4910.48 s	1608.42	(14.38)	5014.96 s	1609.73	(9.46)	2410.78 s
22-30-5	1572.88	(19.04)	1384.73 s	1575.15	(20.71)	1648.86 s	1584.16	(13.52)	532.00 s
22-30-6	1601.73	(10.59)	4204.84 s	1604.63	(13.51)	4370.06 s	1606.52	(9.28)	1550.40 s
22-30-7	1852.32	(12.88)	6720.34 s	1853.67	(15.18)	6871.48 s	1859.26	(5.70)	1812.46 s
22-30-8	1858.19	(12.42)	7341.45 s	1860.88	(13.73)	7383.25 s	1869.63	(6.48)	2503.74 s
22-30-9	1465.06	(9.38)	2727.12 s	1465.84	(10.80)	2899.03 s	1469.25	(7.54)	1140.57 s
22-30-10	1659.41	(21.99)	1958.40 s	1661.53	(22.43)	2190.21 s	1670.64	(14.94)	1290.81 s

the deterministic MAOPCC, which selects nodes in a manner that effectively utilizes almost all available time limits without considering the impact of nodes not selected in the solution. Therefore, the visitor may be in a disadvantageous position when considering inserting attractions in the Re-Opt policy. However, when dealing with the MAOPCCSW, we explicitly considered these impacts in advance. Note that the deviation gap between

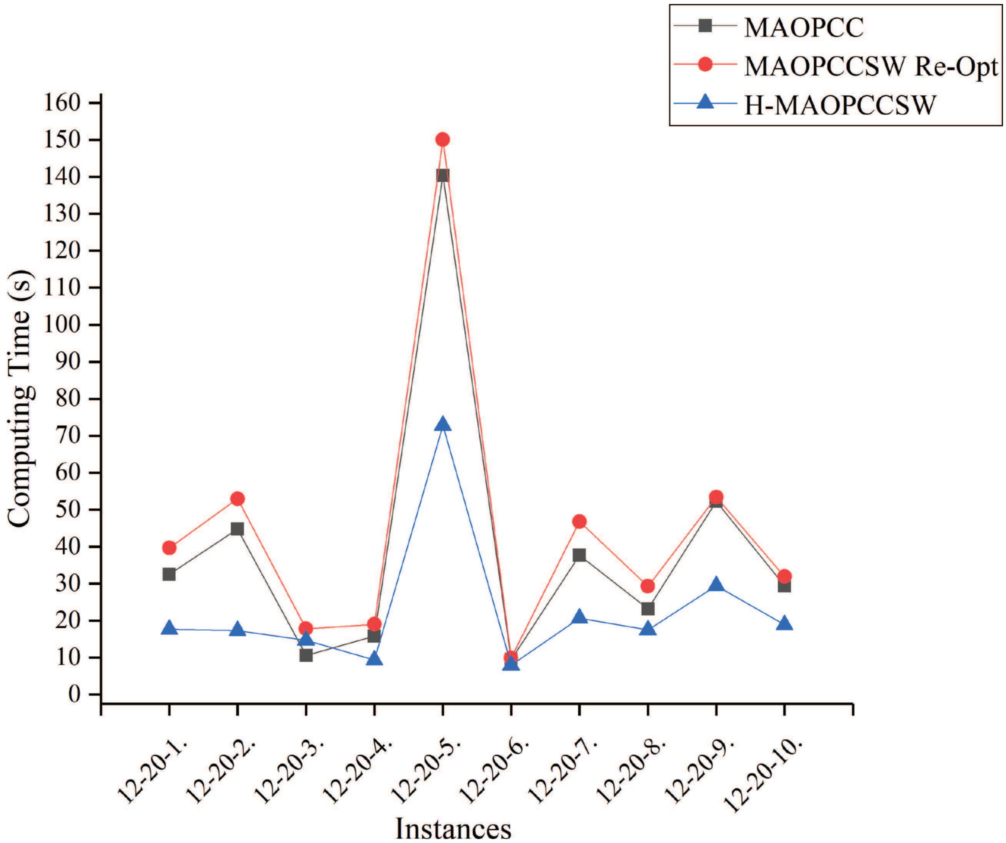


FIGURE 3. Computing time of the second dataset.

the deterministic MAOPCC and the MAOPCCSW heuristic in the third dataset is larger than that in the second dataset. Especially, for the seventh, eighth and tenth instance in the third dataset, the standard deviation of the deterministic MAOPCC is much larger than that of the MAOPCCSW heuristic. This phenomenon can occur on the MAOPCCSW, since the MAOPCC is a complex problem and small changes in distance may cause significant modifications of the solution, which greatly increases the standard deviation. Moreover, as shown in Figures 3 and 4, the H-MAOPCCSW algorithm can solve the problem with less computing time.

The average results of the three datasets are presented in Table 5. Notably, as the scale of the problem increases, the SAA procedure cannot solve the problem. Compared to the other two algorithms, the proposed H-MAOPCCSW algorithm can find a solution with higher expected value within lower computing time.

7. CONCLUSION

In this research, we use the idea of the two-stage recourse model with profit shortage to tackle the multiple-agent orienteering problem with stochastic weight and capacity constraints (MAOPCCSW) by a two-stage model. The total profit of nodes that cannot be accessed due to the weight realization in the second-stage is defined as the profit shortage. With the profit penalty in the second-stage, we can prevent inserting redundant attractions in the first-stage.

To address the MAOPCCSW, we begin by utilizing the sample average approximation (SAA) method for the two-stage model. To express the profit obtained under specific travel and weight realization scenarios, we

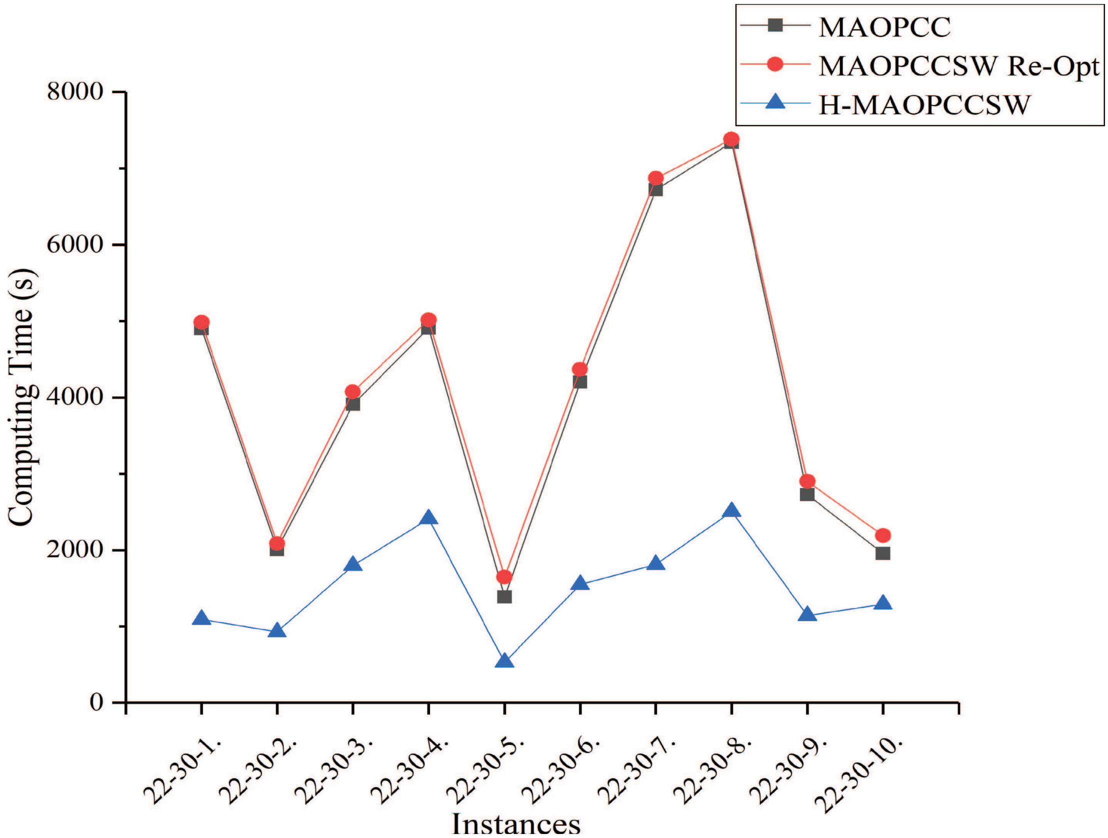


FIGURE 4. Computing time of the third dataset.

TABLE 5. Average results for the datasets.

Instance	Expected profit (standard deviation)												Average two-stage model UB
	MAOPCC			MAOPCCSW Re-Opt			H-MAOPCCSW			Two-stage model with SAA			
7-10	109.65	(8.76)	7822.12 s	110.47	(9.55)	7845.95 s	114.17	(6.99)	0.62 s	111.67	(6.46)	97.94h	117.34
12-20	535.39	(6.60)	39.62 s	536.47	(7.84)	45.08 s	544.86	(4.48)	22.64 s	–	–	–	–
22-30	1611.81	(14.08)	4006.53 s	1613.64	(15.54)	4151.82 s	1619.13	(9.77)	1505.80 s	–	–	–	–

Notes. "–" means the SAA algorithm cannot solve the problem within 192 h.

introduce a linearization expression. This expression enables us to approximate the profit with greater efficiency and accuracy, facilitating the solution process for this complex problem. Linearization is helpful to the SAA method of the MAOPCCSW. The objective value obtained through SAA will converge to the best-expected profit in the two-stage model, but this convergence process can be time-consuming. Therefore, for solving large cases, we develop the MAOPCCSW heuristic method, which utilizes the characteristics of the MAOPCCSW and considers the relevant uncertainties. The heuristic is based on the variable neighborhood search (VNS) algorithm for solving the MAOPCC and we adjust the evaluation approach of the construction phase and improvement phase. Our MAOPCCSW heuristic algorithm can provide high-quality solutions with less computational time.

The calculation results show that our two-stage model with the SAA method can provide high-quality solutions compared with the deterministic MAOPCC solutions and solutions with the Re-Opt policy. At the same time, the two-stage model can also provide an upper bound for the estimate of the solution. Besides, the MAOPCCSW heuristic can solve the MAOPCCSW with less computational time and higher quality.

Future research may focus on a multi-stage mode. The solution in the first phase is to carry out the initial tour as long as possible and add or delete nodes to the tour according to the predefined online strategy. In this way, the expected profit may increase further, because the solution of the two-stage model is a part of the solution space of the multi-stage model. For example, the realization of the stochastic weight can be achieved each time when a visitor travels through the arc and the routes can be adjusted according to the realized arc weight. In addition, there are differences in the behavior of different customers in a service system [12], so another research potential is to consider the behavior and preferences of heterogeneous tourists in the modeling. For example, different types of visitors exhibit varying levels of patience, leading to differences in their tolerable waiting times. Taking this variation into account can make the problem more aligned with real-world scenarios.

Funding

This research was financially supported by the National Natural Science Foundation of China (Grant No. 72371086 and 72101068), Zhejiang Provincial Natural Science Foundation of China (Grant No. LZ20G010001), Zhejiang Provincial Philosophy and Social Science Planning Project, China (Grant No. 23SYS11ZD and 21NDJC072YB).

Data availability statement

The code used in this paper is available online in a Github repository: https://github.com/darkness399/A-TWO-STAGE-APPROACH-TO-THE-MAOPCCSW [31].

References

[1] A.A. Aazami and M. Saidi-Mehrabad, Benders decomposition algorithm for robust aggregate production planning considering pricing decisions in competitive environment: a case study. *Sci. Iranica* **26** (2019) 3007–3031.

[2] A. Aazami and M. Saidi-Mehrabad, A production and distribution planning of perishable products with a fixed lifetime under vertical competition in the seller–buyer systems: a real-world application. *J. Manuf. Syst.* **58** (2021) 223–247.

[3] R.M. Apap and I.E. Grossmann, Models and computational strategies for multistage stochastic programming under endogenous and exogenous uncertainties. *Comput. Chem. Eng.* **103** (2017) 233–274.

[4] A.M. Campbell, M. Gendreau and B.W. Thomas, The orienteering problem with stochastic travel and service times. *Ann. Oper. Res.* **186** (2011) 61–81.

[5] I. Dolinskaya, Z.E. Shi and K. Smilowitz, Adaptive orienteering problem with stochastic travel times. *Transp. Res. Part E-Logistics Transp. Rev.* **109** (2018) 1–19.

[6] L. Evers, T. Dollevoet, A.I. Barros and H. Monsuur, Robust UAV mission planning. *Ann. Oper. Res.* **222** (2014) 293–315.

[7] L. Evers, K. Glorie, S. van der Ster, A.I. Barros and H. Monsuur, A two-stage approach to the orienteering problem with stochastic weights. *Comput. Oper. Res.* **43** (2014) 248–260.

[8] B.L. Golden, Q. Wang and L. Liu, A multifaceted heuristic for the orienteering problem. *Nav. Res. Logistics (NRL)* **35** (1988) 359–366. https://onlinelibrary.wiley.com/doi/abs/10.1002/1520-6750%28198806%2935%3A3%3C359%3A%3AAID-NAV3220350305%3E3.0.CO%3B2-H.

[9] A. Goli, A. Aazami and A. Jabbarzadeh, Accelerated cuckoo optimization algorithm for capacitated vehicle routing problem in competitive conditions. *Int. J. Artif. Intell.* **16** (2018) 88–112. https://api.semanticscholar.org/CorpusID:64860002.

[10] A. Gunawan, H.C. Lau and P. Vansteenwegen, Orienteering problem: a survey of recent variants, solution approaches and applications. *Eur. J. Oper. Res.* **255** (2016) 315–332.

[11] M. Heydari and A. Aazami, Minimizing the maximum tardiness and makespan criteria in a job shop scheduling problem with sequence dependent setup times. *J. Ind. Syst. Eng.* **11** (2018) 134–150.

[12] Y. Hua, T. Che, C. Yang and M. Hu, Customer no-show reduction in web-based appointment service: investigations of non-attendance behaviors. *Serv. Ind. J.* **44** (2024) 538–562.

[13] T. Ilhan, S.M.R. Iravani and M.S. Daskin, The orienteering problem with stochastic profits. *IIE Trans.* **40** (2008) 406–421.

[14] L. Ke, Z. Xu, Z. Feng, K. Shang and X. Qian, Proportion-based robust optimization and team orienteering problem with interval data. *Eur. J. Oper. Res.* **226** (2013) 19–31.
[15] M.H. Kolaee and S.M.J.M. Al-e Hashem, Stochastic medical tourism problem with variable residence time considering gravity function. *RAIRO-Oper. Res.* **56** (2022) 1685–1716.
[16] J. Li, Z. Zhang, D. Song, B. Yang and L. Zhou, Cruise onboard itinerary planning for multi passengers with service venue capacity and time-window constraints. *Comput. Oper. Res.* **176** (2025) 106944.
[17] X.-G. Luo, X.-R. Liu, P.-L. Ji, X.-Z. Shang and Z.-L. Zhang, Trip planning for visitors in a service system with capacity constraints. *Comput. Oper. Res.* **148** (2022) 105974.
[18] A. Makui, M. Heydari, A. Aazami and E. Dehghani, Accelerating benders decomposition approach for robust aggregate production planning of products with a very limited expiration date. *Comput. Ind. Eng.* **100** (2016) 34–51.
[19] M.S. Mehrabad, A. Aazami and A. Goli, A location–allocation model in the multi-level supply chain with multi-objective evolutionary approach. *J. Ind. Syst. Eng.* **10** (2017) 140–160.
[20] V. Norkin, G. Pflug and A. Ruszczyński, A branch and bound method for stochastic global optimization. *Math. Program. Ser. B* **83** (1998) 425–450.
[21] A. Shapiro, Inference of statistical bounds for multistage stochastic programming problems. *Mathematical Methods Of Oper. Res.* **58** (2003) 57–68.
[22] J. Slootweg, R. van der Mei, C.J. Jagtenberg and F. Ottenhof, Centralized multi-visitor trip planning with activity reservations in crowded destinations. *Comput. Oper. Res.* **167** (2024) 106633.
[23] S. Solak, J.-P.B. Clarke, E.L. Johnson and E.R. Barnes, Optimization of R&D project portfolios under endogenous uncertainty. *Eur. J. Oper. Res.* **207** (2010) 420–433.
[24] H. Tang and E. Miller-Hooks, Algorithms for a stochastic selective travelling salesperson problem. *J. Oper. Res. Soc.* **56** (2005) 439–452.
[25] S.Y. Teng, H.L. Ong and H.C. Huang, An integer l-shaped algorithm for time-constrained traveling salesman problem with stochastic travel and service times. *Asia-Pac. J. Oper. Res.* **21** (2004) 241–257.
[26] T. Tsiligiridis, Heuristic methods applied to orienteering. *J. Oper. Res. Soc.* **35** (1984) 797–809.
[27] N.S.S.S.N. Usha Devi and R. Mohan, VNSMAS: a constraint-based portfolio profit maximization. *Comput. Oper. Res.* **170** (2024) 106769.
[28] P. Vansteenwegen, W. Souffriau and D. Van Oudheusden, The orienteering problem: a survey. *Eur. J. Oper. Res.* **209** (2011) 1–10.
[29] J. Wang, J. Guo, J. Chen, S. Tian and T. Gu, Uncertain team orienteering problem with time windows based on uncertainty theory. *IEEE Access* **7** (2019) 63403–63414.
[30] W. Wang, H.C. Lau and S.-F. Cheng, Exact and heuristic approaches for the multi-agent orienteering problem with capacity constraints, in 2017 IEEE Symposium Series on Computational Intelligence (SSCI) (2017). 10.1109/SSCI.2017.8285329.
[31] L. Xinrui, L. Xinggang, J. Pengli, Z. Zhongliang and Y. Hu, Java code for "a two-stage approach to the multiple-agent orienteering problem with stochastic weight and capacity constraints" (2025). https://github.com/darkness399/A-TWO-STAGE-APPROACH-TO-THE-MAOPCCSW.
[32] Q. Yu, C. Cheng and N. Zhu, Robust team orienteering problem with decreasing profits. *INFORMS J. Comput.* **34** (2022) 3215–3233.