AIMS Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 9(9): 24247–24270.
DOI: 10.3934/math.20241180
Received: 11 June 2024
Revised: 30 July 2024
Accepted: 05 August 2024
Published: 16 August 2024

Research article

A sim-learnheuristic algorithm for solving a capacitated dispersion problem under stochastic and non-static conditions

Elnaz Ghorbani[1,2], **Juan F. Gomez**[2], **Javier Panadero**[3] **and Angel A. Juan**[2,*]

[1] Department of Computer Science, Universitat Oberta de Catalunya, 08018 Barcelona, Spain

[2] Research Center on Production Management and Engineering, Universitat Politècnica de València, 03801 Alcoy, Spain

[3] Department of Computer Architecture and Operating Systems, Universitat Autònoma de Barcelona, 08193 Bellaterra, Spain

* **Correspondence:** Email: ajuanp@gmail.com.

Abstract: A fundamental assumption in addressing real-world problems is acknowledging the presence of uncertainty and dynamism. Dismissing these factors can lead to the formulation of an optimal solution for an entirely different problem. This paper presents a novel variant of the capacitated dispersion problem (CDP) referred to as the stochastic and non-static CDP. The main objective of this problem is to strategically position facilities to achieve maximum dispersion while meeting the capacity demand constraint. The proposed approach combines stochastic and non-static elements, introducing a new paradigm to address the problem. This innovation allows us to consider more realistic and flexible environments. To solve this challenging problem, a novel sim-learnheuristic algorithm is proposed. This algorithm combines a biased-randomized metaheuristic (optimization component) with a simulation component (to model the uncertainty) and a machine learning component (to model non-static behavior). The non-static part works by using black box and white box mechanisms to learn the uncertainty with some related facilities' variables. Based on an extended set of traditional benchmarks for the CDP, a series of computational experiments were carried out. The results demonstrate the effectiveness of the proposed sim-learnheuristic approach for solving the CDP under non-static and stochastic scenarios.

Keywords: capacitated dispersion problem; sim-learnheuristic; machine learning; simulation
Mathematics Subject Classification: 68T20, 90-08, 90-10, 90Bxx, 90B36

1. Introduction

Real-world applications of combinatorial optimization algorithms in fields such as logistics, manufacturing, finance, telecommunications, and computer science are vast and diverse, demonstrating the broad impact and significance of solving these complex optimization challenges. In the CDP, the objective is to maximize the dispersion of facilities while considering a capacity constraint that must meet the minimum capacity demand [17]. This NP-hard problem has been defined by [29] in order to model a type of facility location problem with dispersion. Hence, the goal is to choose a subset of facilities with the objective of maximizing their dispersion according to a specified distance measure [9]. The concept of diversity focuses on the measurement of the distances between facilities, which is customized to suit the particular requirements of the application at hand [20]. In the traditional CDP, the quantities demanded and the capacities of the facilities are assumed to be deterministic and known without any ambiguity. Nonetheless, in numerous real-world scenarios, particularly in logistics and transportation, the demand for goods or services may experience randomness, which means that the facilities' capacities might be influenced by uncertain factors. The stochastic capacitated dispersion problem (SCDP) introduces a further layer of realism by incorporating uncertainties. Actually, in many practical situations, the capacity of facilities may not always remain fixed due to a myriad of reasons, such as equipment failures, maintenance requirements, or unforeseen changes in available resources. As such, the SCDP presents a more comprehensive and accurate representation of real-world complexities in supply chain and distribution systems. One of the main challenges in choosing a facility location is ensuring it can handle customer demand. In other words, the capacity of operational facilities should ideally surpass the total demand they need to serve. This condition frequently arises in various logistics-related problems, regardless of the decision level [12].

Furthermore, the capacity parameter in the CDP may also need to be non-static to better reflect real-world scenarios. In real case studies, certain nodes may have limited capacity due to factors such as physical space, equipment availability, or regulatory restrictions. For instance, in a transportation network, some distribution centers might have zero capacity due to maintenance or temporary closures, while others operate at their maximum capacity. This implies that some facilities might unexpectedly exhibit zero capacity, determined by a black box mechanism that considers environmental, operational, or market-related variables [13]. Thus, in this paper, a new stochastic and non-static capacitated dispersion problem (SNS-CDP) is proposed. In the SNS-CDP, the capacities of the facilities may be stochastic or non-static, and a sim-learnheuristic solution approach is proposed to address the complexities introduced in the stochastic and non-static components. Figure 1 illustrates a simplified SNS-CDP where the blue shapes show the selected facilities and the white small circles show the unselected ones. In this figure, facilities are depicted with stochastic or non-static storage capacities.

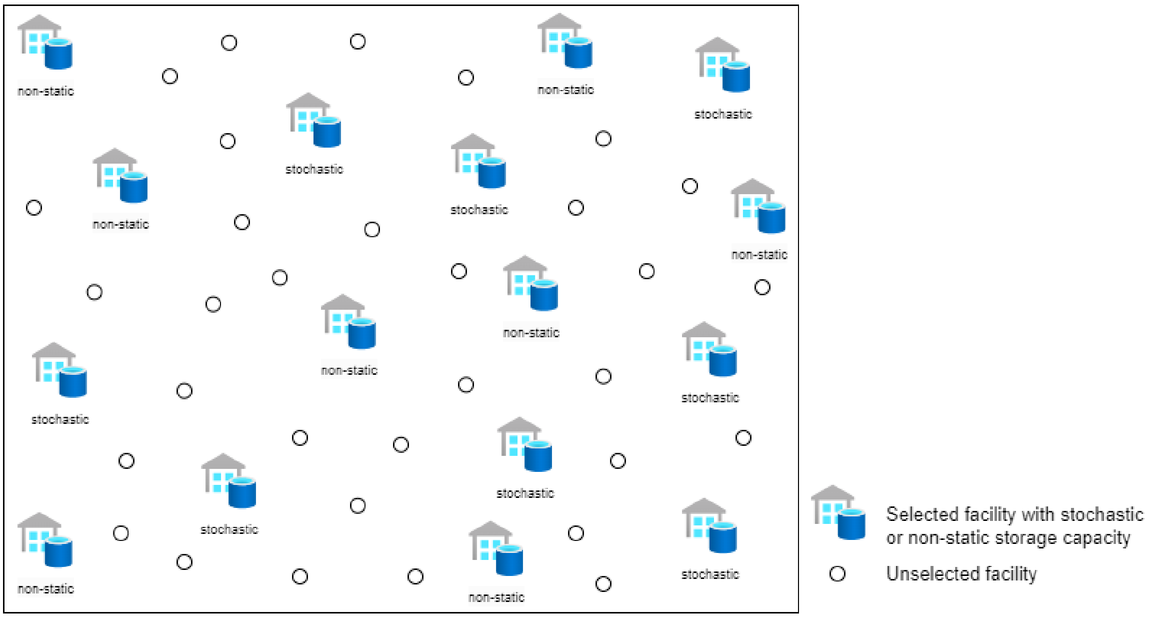

non-static
stochastic
Selected facility with stochastic or non-static storage capacity
Unselected facility

Figure 1. Visualizing a SNS-CDP: Balancing distribution efficiency with resource limitations.

Authors such as [4,27], among many others, have explained why it is important to consider random variables (following certain probability distributions) when modeling real-life capacities (or other inputs of the optimization problem). Assuming these real-life capacities as deterministic values might not properly reflect the uncertainty that characterizes real-life operations. Hence, the main contribution of this paper is designing an innovative sim-learnheuristic algorithm that hybridizes a metaheuristic algorithm, simulation, and machine learning in order to solve the novel SNS-CDP. By defining the type of each facility, the algorithm is able to address the random capacities as well as the non-static ones. In addition to its contribution to the field of facility location and optimization, the proposed sim-learnheuristic algorithm holds significant potential for broader applications in logistics, supply chain management, manufacturing, and telecommunication networks, where both the stochastic and non-static capabilities should be considered to produce solutions closer to reality. The rest of this paper is organized as follows: Section 2 presents related work on the CDP. The mathematical formulation is described in Section 3. In Section 4, our solution methodology is proposed. Following that, in Section 5, we present the numerical findings achieved through the experimentation of our method. Lastly, Section 6 summarizes the main contributions of the paper and outlines potential lines for future research.

2. Related work

Since the stochastic and non-static CDP have not been studied before, we provide some references on the classical version of the CDP as well as on some stochastic or dynamic versions of the more general facility location problem (FLP).

2.1. Previous work on the CDP

Despite the CDP not having been extensively explored in the scientific literature, there are several works on its deterministic version. For instance, Marti et al. [19] presented a heuristic based on the scatter search methodology to solve the CDP. The results demonstrate the superiority of the proposed model and heuristic over existing methods. In another paper, Marti et al. [26] proposed heuristics based on the greedy randomized adaptive search procedure (GRASP) and variable neighborhood descent (VND) methodologies to the CDP. Gomez et al. [12] proposed a multi-start biased-randomized algorithm to solve the deterministic CDP efficiently. Compared to the available instances' solutions in the literature, the proposed approach is highly competitive. In another paper, Mladenovic et al. [24] presented a variable neighborhood search VNS-based heuristics to solve the deterministic CDP. The proposed heuristics are basic VNS, general VNS, and general skewed VNS, and their efficiency has been assessed on benchmark instances. Also, Lu et al. [17] proposed a fast greedy construction parameter-free heuristic algorithm based on tabu search. Additionally, Landete et al. [16] explored the CDP from a mathematical perspective. In this work, the CDP is considered through various mathematical formulations. The study enhanced the formulations by incorporating valid inequalities and variable-fixing procedures. Computational experiments demonstrated the practicality of the proposed approaches for solving diverse instances of the problem in its deterministic version. More recently, Rosati et al. [28] proposed a multi-neighborhood simulated annealing for the CDP. This paper considers two compact mathematical models to achieve good bounds on various instance sizes.

2.2. Stochastic and dynamic facility location problems

The majority of CDP literature predominantly concentrates on the deterministic variant, while the stochastic and non-static CDP models have not been extensively addressed yet. However, a recent research by [11] presented a new simulation-based metaheuristic approach for solving the stochastic CDP. This paper introduces two models for the CDP with stochastic capacities, incorporating soft constraints and penalty costs for violating the total capacity constraint. There has been some research conducted on stochastic and/or dynamic FLPs. For instance, Bieniek [3] addressed the single source capacitated FLP with general stochastic demands. They proposed a unified solution for determining facility locations and customer allocations. Additionally, some papers have employed actual case studies on the dynamic FLP. For instance, Mišković, et al. [23] introduced a robust variant of the dynamic FLP, focusing on optimizing the emergency service network of police special forces units in Serbia. The paper also used the variable neighborhood search method with an efficient local search procedure to solve real-life problem instances. Jena et al. [15] addressed a multi-period FLP with multiple commodities and capacity levels, allowing for facility relocation and temporary closures. The study proposed a formulation for the problem and a hybrid heuristic that combines Lagrangian relaxation and a restricted mixed-integer programming model. In another paper, Marufuzzaman et al. [22] addressed the capacitated dynamic FLP, which aims to satisfy customer demands at minimum cost by determining facility opening, closing, or retention over time. To tackle this challenging problem, the paper proposed a unique hybrid solution algorithm, combining a rolling horizon algorithm with an accelerated Benders decomposition approach. Pearce et al. [25] proposed an approach to solve the budget-constrained dynamic uncapacitated facility location and network design problem optimally. The problem involves constructing or expanding a network and placing facilities within a given budget to meet demands. The study utilizes disaggregated Benders decomposition within a branch-and-cut framework. In another work, Wang et al. [31] introduced a dynamic k-level FLP considering the time factors, and proposed a combinatorial primal-dual approximation algorithm that finds a constant factor approximate solution. Additionally, Oliveira et al. [6] investigated two variants of a dynamic multi-period two-level uncapacitated FLP, where first-level plants serve scattered clients over time via second-level facilities. The study proposed a decomposition approach using Benders decomposition and GRASP. Finally, Silva et al. [30] focused on the dynamic FLP with modular capacities, aiming to minimize location and demand allocation costs over a planning horizon. They proposed three linear relaxation-based heuristics and an evolutionary heuristic that hybridizes a genetic algorithm with a variable neighborhood descent metaheuristic. Recently, Xu et al. [32] explored two ensemble methods for decision-making in stochastic mixed-integer FLP with limited computational budgets, focusing on variance reduction and probabilistic guarantees. Additionally, Ala et al. [1] exploreed the dynamic capacitated FLP for mobile renewable energy charging stations using two-stage stochastic programming. Finally, Aydin et al. [2] addressed the capacitated reliable facility location problem under uncertainty by developing two novel algorithms that integrate sample average approximation and progressive hedging algorithm.

3. Mathematical formulation

In this section, the mathematical formulation of the deterministic CDP proposed by [29] is presented; then, the stochastic and non-static formulations derived from the deterministic model are proposed.

The CDP is mathematically defined as follows: Consider a graph $G = (V, E)$, where V represents a set of n nodes (facilities), and E is a set of edges connecting these nodes. Each node $i \in V$ has an average capacity denoted by c_i, and for every pair of nodes i and j, connected by an edge $(i, j) \in E$, their distance is represented by d_{ij} ($d_{ij} = d_{ji}$). The objective of the CDP is to choose a subset of V, such that the total capacity of the selected nodes is equal to or exceeds a given value B, while simultaneously maximizing the minimum distance among all pairs of selected facilities. Additionally, we introduce a binary variable x_i that is equal to 1 when facility i is selected, and 0 otherwise. Based on these definitions, the mathematical model of the deterministic CDP is the following:

$$\max \min_{i,j \in V, i<j} d_{ij} x_i x_j \tag{3.1}$$

$$\text{s.t.: } \sum_{i \in V} c_i x_i \geq B \tag{3.2}$$

$$x_i \in \{0, 1\}, \quad \forall i \in V. \tag{3.3}$$

Equation (3.1) represents the objective function aiming to maximize the minimum distance among the selected pairs of facilities. Additionally, constraints (3.2) guarantee that the selected nodes meet the minimum capacity requirement (B). Finally, constraints (3.3) refer to the nature of the x_i variable, enforcing that x_i is a binary decision variable. This restriction ensures that $x_i = 1$ when facility i is selected and $x_i = 0$ when it is not. It is assumed that c_i represents the deterministic capacity of facility i.

In the stochastic model of CDP, it is assumed that the storage capacity of each facility is not known in advance. Instead, it is modeled using a probability distribution function. To achieve this, each facility is associated with an independent parameter, denoted by C_i (for all $i \in V$), which follows a non-negative probability distribution, such as the Weibull or log-normal distributions. The formulations of the stochastic variant are as follows:

$$\max \min_{i,j \in V, i<j} d_{ij} x_i x_j \tag{3.4}$$

$$\text{s.t.: } P\Big(\sum_{i \in V} C_i x_i \geq B\Big) \geq \alpha \tag{3.5}$$

$$x_i \in \{0, 1\} \quad \forall i \in V. \tag{3.6}$$

In this model, constraints (3.5) state that the selected facilities collectively have a capacity greater than or equal to the specified minimum requirement with a probability of at least α. When faced with a specific scenario, if the system is incapable of fulfilling the requested demand, there are no available backup or recourse actions that can be employed to revise the situation. Consequently, it becomes essential to calculate the probability of the solution's failure, which refers to the likelihood of the system not being able to provide the necessary service as required. This computation allows us to assess the reliability and performance of the system in uncertain situations, helping to identify potential shortcomings and make informed decisions to improve its overall effectiveness.

In the non-static scenario, we assume that the storage capacities of the facilities are neither deterministic nor stochastic subject to a predefined probability distribution function. We refer to it as non-static because the parameter is defined in a manner that is not mathematically dynamic, meaning it does not depend on time. However, it is also not entirely static. Therefore, in this paper, the term "non-static" denotes the specific nature of the parameter. In this scenario, the probability of obtaining a capacity for node i is assumed to follow a logistic regression model. According to [13], the formulations of the non-static component are as follows:

$$\max \min_{i,j \in V, i<j} d_{ij} x_i x_j \tag{3.7}$$

$$\text{s.t.: } \sum_{i \in V} p_i c_i x_i \geq B \tag{3.8}$$

$$x_i \in \{0, 1\} \quad \forall i \in V. \tag{3.9}$$

In this model, constraint (3.8) ensures that the total capacity of the selected nodes meets the minimum required threshold. This constraint incorporates a probability term p_i following a Bernoulli distribution, $p_i \sim Ber(\phi(c_i, sd_i, od_i))$ where ϕ is a sigmoid function of the logistic regression model. The function ϕ is defined as:

$$\phi(c_i, sd_i, od_i) = \frac{1}{1 + \exp(-(\beta_0 + \beta_1 c_i + \beta_2 sd_i + \beta_3 od_i))} \quad \forall i \in V. \tag{3.10}$$

In Eq (3.10), the function ϕ is a black box model that imitates the behavior of the real world. In this function, the parameter c_i represents the deterministic capacity of node i, sd_i denotes the seasonal demand at node i, and od_i indicates the operational disruption associated with node i. In addition, β_1, β_2, and β_3 are dependent coefficients, while the β_0 is the intercept term. In this model, the variables sd_i and od_i are binary variables, taking on values of either 0 or 1. A value of 1 signifies a promising condition, while a value of 0 denotes a non-promising condition.

In the stochastic-non-static scenario, we recognize that the capacities of facilities do not all follow the same statistical pattern. Some facilities' capacities are modeled stochastically, incorporating variability and uncertainty through a probability distribution function. For other facilities, capacities are considered non-static, meaning they can either be zero or match the deterministic capacity based on the variabilities of seasonal demand and operational disruption. Unlike stochastic capacities, non-static capacities are not described by probability distributions but are assumed to be influenced by a black box model.

4. Our solution approach

In this section, our proposed solution approach is presented. Since the problem described in the previous section is NP-hard, an approximation algorithm is defined to address the complexities of the problem. To address the solution approach for the SNS-CDP, a sim-learnheuristic methodology is proposed. We utilize the power of metaheuristics to handle the deterministic variant of the problem effectively. Additionally, a simulation technique is integrated with the metaheuristics to address the SCDP. Finally, a machine learning component is incorporated to account for the non-static CDP. In the following subsections, each component of our sim-learnheuristic solution approach is described.

4.1. Metaheuristic component

To address the deterministic version of the CDP, a novel hybrid metaheuristic is proposed. Our metaheuristic involves two distinct phases: The initial phase focuses on generating a solution, while the subsequent phase aims to improve it. During the first phase, a constructive heuristic is developed that can adapt based on the specific instance being addressed. This method is repeatedly employed in a classic multi-start pattern to create multiple solutions. In the second phase, the solutions undergo further improvement using a multi-start biased-randomized (MSBR) algorithm. This approach potentially refines the solutions to obtain higher-quality results.

4.1.1. Constructive heuristic

The constructive heuristic, proposed by [10], involves adding elements to a partial solution in a greedy manner. In order to randomly select elements from a list of promising candidates based on the distance objective function, it is incorporated into the GRASP framework [7]. To further enhance these methods, we introduce biased-randomization (BR) techniques [14], enabling the addition of nodes. These BR techniques utilize skewed probability distributions, transforming deterministic heuristics into probabilistic algorithms. As a result, this algorithm can rapidly generate numerous high-quality solutions for the problem.

Our constructive heuristic using a BR technique starts with an initial empty solution S. During the execution of the algorithm, nodes from set V are added to solution S, all while ensuring that the capacity constraint remains unfulfilled. The instance I, which is going to be solved, encompasses important information, such as the capacity of each node (facility) i (c_i), the minimum required capacity level (B), and the distances between pairs of nodes (d_{ij}). The other parameter of the algorithm is δ, which acts as a balancing factor between the distance metric and the capacity aspect within the governing greedy function. Equation (4.1) provides the mathematical expression to calculate the h_i, where d_s is the shortest distance between i and any other nodes j in S, and c_i is the capacity of node i. Parameter δ manages the balance between distance and capacity considerations.

$$h_i = \delta \cdot d_s(i, j) + (1 - \delta) \cdot c_i. \tag{4.1}$$

A flowchart of the algorithm steps is presented in Figure 2. In the algorithm, the procedure begins by initializing an empty set S to represent the evolving solution. The edges within the input graph are sorted in descending order based on their distances, forming the list $edges - list$, which aids edge selection. A biased-randomized selection is used to choose the edge (i_0, j_0) from $edges - list$ using a geometric probability distribution function, and its nodes i_0 and j_0 are added to S. Then, the candidate list CL is formed, comprising nodes from V excluding those in S. The total weight of the solution S is set to the distance between i_0 and j_0. An iterative process begins, where nodes in CL are sorted in descending order based on a certain criterion, and a node i^* is chosen using a geometric selection strategy. A biased random selection of the elements with the geometric probability distribution based on the h values provided by Eq (4.1) returns the element i^* to be added to the solution. Node i^* is added to S and the solution's total weight is updated. Then, the node i is removed from CL. This iterative process continues until S becomes feasible. Finally, the algorithm returns the evolved solution S as the output.

In the non-static model, a white box is applied in order to provide predictions to deal with the uncertainty of the problem. The white box is defined as a function that approximates the black box. In this regard, for the first two nodes, the Eq (4.2) should be used to calculate the h_i instead of Eq (4.1), where $prob(i)$ is the probability of facility i covering the capacity.

$$h_i = \delta \cdot d_s(i, j) + \frac{1 - \delta}{2} \cdot \left(prob(i) \cdot c_i + prob(j) \cdot c_j\right). \tag{4.2}$$

In the algorithm, after generating the S set, the black box updates the capacities of the nodes regarding the facilities condition variables. In the non-static model, the candidate list evaluates the nodes and ranks them in order of preference based on an evaluation function. The evaluation function, defined in Eq (4.3), considers several key factors. The term min_distance(i) represents the minimum distance between node i and the selected facilities. The term max_min_distance is the maximum value of these minimum distances among all candidates in the candidate list. The capacity of a candidate facility is denoted by c_i, while max_cap represents the maximum capacity among all candidate facilities. Additionally, $prob(i)$ signifies the probability that facility i will cover the required capacity. The value of $\delta \in (0, 1)$ is a tuning parameter that is selected based on a value that produces the best outcome.

$$eval(i) = \delta \cdot \frac{min\_distance(i)}{max\_min\_distance(V \setminus S)} + (1 - \delta) \cdot \frac{c_i \cdot prob(i)}{max\_cap(V \setminus S)}. \tag{4.3}$$

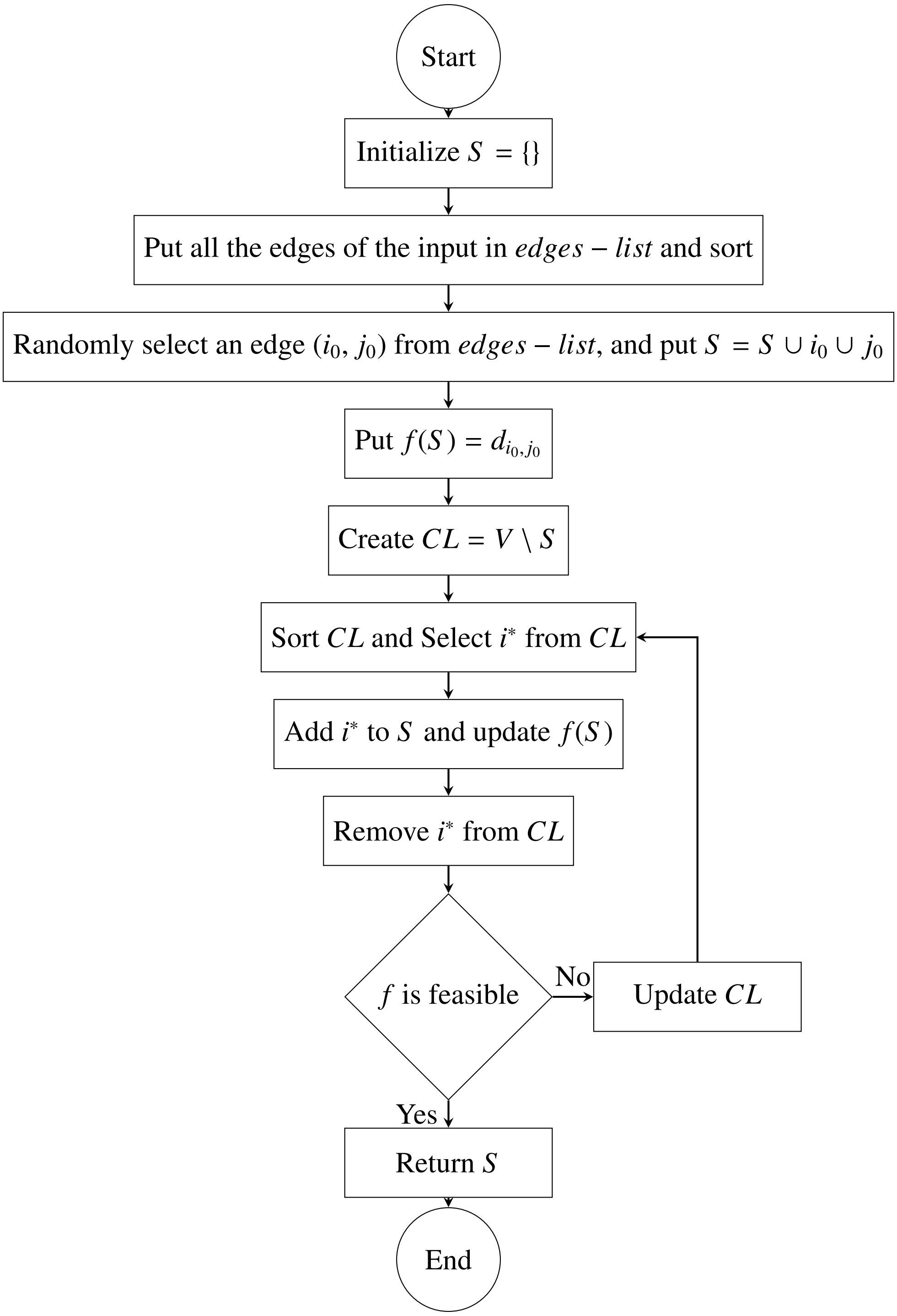

Start
Initialize $S = \{\}$
Put all the edges of the input in $edges - list$ and sort
Randomly select an edge (i_0, j_0) from $edges - list$, and put $S = S \cup i_0 \cup j_0$
Put $f(S) = d_{i_0,j_0}$
Create $CL = V \setminus S$
Sort CL and Select i^* from CL
Add i^* to S and update $f(S)$
Remove i^* from CL
f is feasible
No
Update CL
Yes
Return S
End

Figure 2. The flowchart of the constructive heuristic algorithm.

4.1.2. Multi-start biased-randomized methodology

Figure 3 shows the framework of the MSBR algorithm. This algorithm takes the initial solution as its input, denoted as S, which is generated through the constructive method, and a parameter representing the maximum number of iterations. The algorithm starts by putting the set S in an initial solution set, S_0. Subsequently, the algorithm sets the *counter* to zero. The core of the algorithm emerges in a loop, where each iteration increments the *counter* by one unit. Within this iterative process, the algorithm first identifies the oldest node from S, effectively removing it from S set. Subsequently, it proceeds to select a node, denoted as u^*, with the maximum minimum distance to S. This selected node, u^*, is then incorporated into the solution set S. Following this adjustment, the algorithm evaluates the quality of the updated solution set by calculating the objective function value, denoted as $f(S)$. If the new objective function value is better than the value associated with the initial solution set, $f(S_0)$, the algorithm updates the set S_0 and resets the counter to zero. If, however, the new solution does not obtain an improvement, the algorithm continues its iterative loop. This iterative process persists until the algorithm reaches the maximum specified number of iterations. Upon reaching this threshold, the final solution set S_0 becomes the output of the algorithm, representing the best solution achieved throughout the optimization process. The computational complexity of the MSBR is $O(n^2)$. This complexity arises from the necessity to sort the edge list in increasing distance order and the iterative process of selecting and removing nodes based on the geometric probability distribution until the capacity constraint is satisfied. The second-phase improvement has a complexity of $O(n \log n)$ per iteration, due to the management of the list and neighborhood explorations. Therefore, the overall complexity of the algorithm can be described as $O(n^2 + it_{\max} \cdot n \log n)$, where $it_{\max}$ denotes the maximum number of iterations without improvement.

Our two-phase algorithm comprising the construction and improvement phases, which encompass the constructive step, as well as MSBR improvements, is described in the previous paragraphs. We collectively refer to this two-phase algorithm, MSBR methodology in the rest of this paper. The subsequent subsection describes the sim-learnheuristic algorithm and the integration of MSBR with our simulation and machine-learning frameworks.

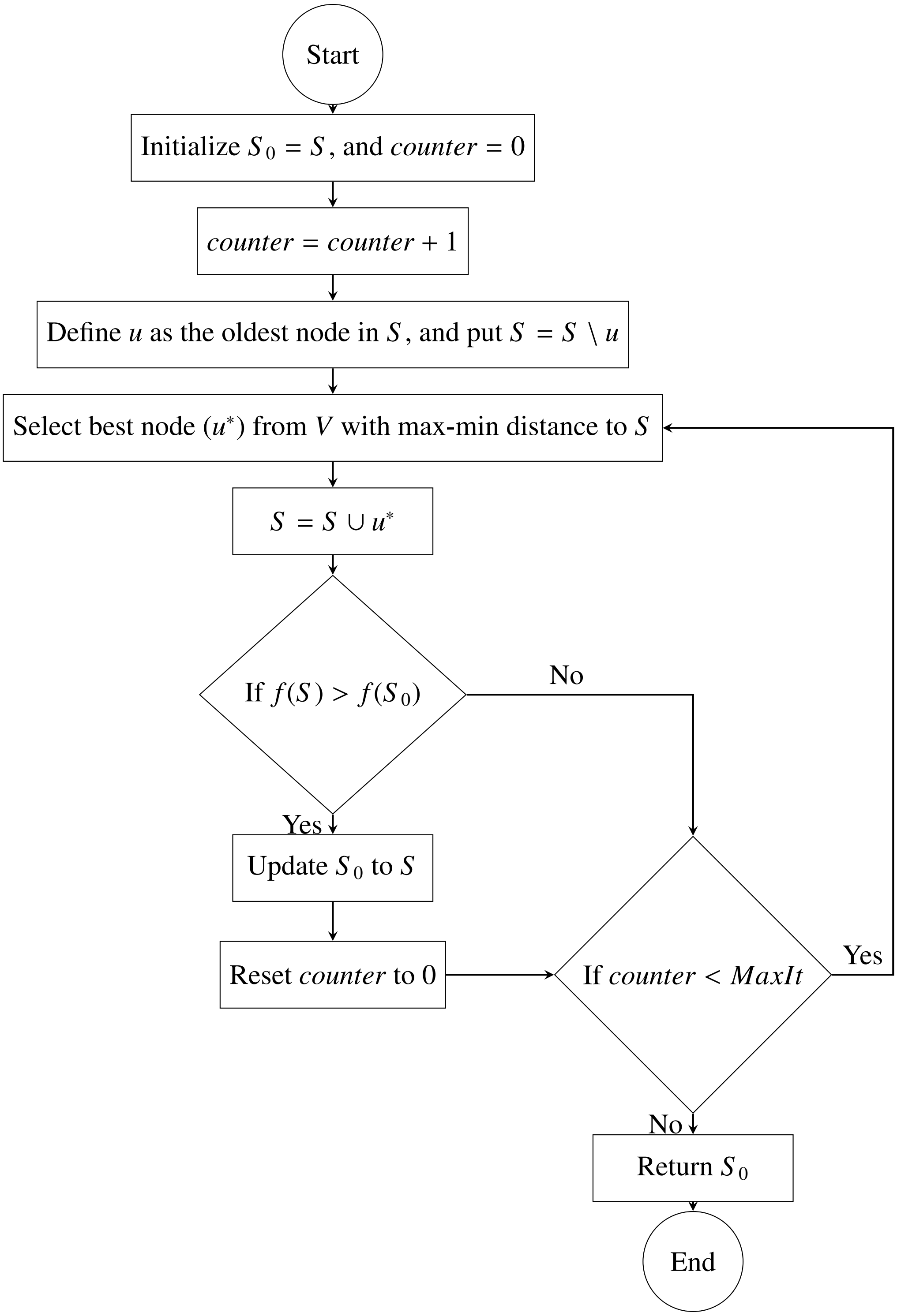

Start
Initialize $S_0 = S$, and $counter = 0$
$counter = counter + 1$
Define u as the oldest node in S, and put $S = S \setminus u$
Select best node (u^*) from V with max-min distance to S
$S = S \cup u^*$
If $f(S) > f(S_0)$
No
Yes
Update S_0 to S
Reset $counter$ to 0
If $counter < MaxIt$
Yes
No
Return S_0
End

Figure 3. The flowchart of the MSBR algorithm.

4.2. Machine learning component

As it has been explained in Section 3, the non-static part of the solution works based on black box and white box mechanisms. The black box imitates the behavior of real life; however, the white box approximates the black box. A multivariate logistic regression machine learning method is applied to predict categorical values based on input variables described in Section 3. The variables used in the logistic regression model are seasonal demand, operational disruption, and deterministic capacity of each facility. This method estimates the probability of a binary dependent variable by modeling the relationship between the dependent and independent variables. The non-static capacities can be zero if the black box model predicts zero or equal to the deterministic capacity, as described in Eq (3.10). The values provided by the black box model are stored in the white box model, which then adjusts the candidate list as outlined in the constructive heuristic section. This process ensures that the capacities in our non-static component are continuously updated.

4.3. Simulation component

In Section 4.1.2, the MSBR algorithm specifically adjusted for addressing the deterministic variant of the CDP is introduced. However, it is not able to deal with the inherent uncertainty prevalent in real-world problems. In this section, the MSBR is improved by incorporating a simulation framework. Algorithm 1 shows the simulation procedure employed within the sim-learnheuristic approach. As mentioned before, in our SNS-CDP, the facilities are categorized into two distinct groups based on their characteristics: Facilities with stochastic or non-static capacities, which are determined through simulation. First, for stochastic facilities, the capacity is determined by generating random numbers utilizing a probability distribution function, specifically- the log-normal distribution function with location μ and scale σ parameters. As discussed in [5, 18], among others, the choice of a log-normal distribution for modeling facility capacities is justified by its suitability in representing positively skewed data, which is typical in capacity-related scenarios. Notice that the specific parameters of the log-normal will be obtained from real-life observations using best-fit statistical techniques. The log-normal distribution is frequently used in practical applications where values are strictly positive and exhibit variability across several orders of magnitude. This characteristic makes it an ideal fit for modeling the stochastic nature of facility capacities in our problem. Second, black box and white box mechanisms are employed to generate the non-static capacities of the corresponding facilities, utilizing three independent variables: c_i, sd_i, and od_i, along with their respective coefficients.

In the context of simulation (Algorithm 1), the key inputs to this algorithm include the solution, its associated nodes, nodes' types, and the maximum number of simulations. The algorithm starts by initializing an empty list named *capacity*. Subsequently, a loop is initiated to iterate through each of the specified maximum simulation iterations. Within this loop, a new variable named *all_nodes_capacity* is created. The primary objective is to populate this list with capacities associated with all facilities in all the iterations. Within another loop, facilities marked for opening_ as determined by the MSBR algorithm_ are considered to be given the capacity values based on their type. If the facility type is stochastic, the capacity is generated as a random number using the log-normal distribution function. In cases where the facility type is non-static, a black_box function generates a probability value based on deterministic capacity, seasonal demand, and operational disruption variables. The resulting capacities are then stored using the white_box function. Upon completing capacity assignments, the capacities of all facilities are aggregated into the *capacity* list, and the cumulative capacities are stored in the *all_nodes_capacity* variable. A major constraint enforced throughout this process is that the collective capacities of all facilities must meet or exceed a predefined demand threshold. However, in instances where this constraint is not met, the algorithm encounters a failure. In the final step, the total capacity is appended to the *capacity* list. Consequently, this list accumulates different total capacity values across different iterations. Finally, the reliability and the final solution are determined. Table 1 describes all the parameters used in the Algorithm 1.

Algorithm 1 Simulation module

```
fail $\leftarrow$ 0
capacity $\leftarrow$ []
while simulation $\leq$ max_simulations do
    all_nodes_capacity $\leftarrow$ 0
    for each node in solution do
        if dict_type[node] is 1 then
            node_capacity $\leftarrow$ StochasticValue[node]
        else
            probValue $\leftarrow$ simulate_blackbox(capacity, operation_disruption,seasonal_demand)
            node_capacity $\leftarrow$ probValue $\times$ capacity
            Add node_capacity to white_box
        end if
        Add node_capacity to all_nodes_capacity
    end for
    if all_nodes_capacity < max_demand then
        fail = fail+1
    end if
    Append all_nodes_capacity to capacity
end while
Update reliability = ( max_simulations - fail) / max_simulations
```

Table 1. Explanation of variables used in the simulation module algorithm.

Variable	Description
fail	Counter for fails
capacity	List storing the capacities of all nodes after each simulation
max_simulations	Maximum number of simulations to run
all_nodes_capacity	Sum of capacities for all nodes in the current simulation
solution	Set of nodes selected to be established
dict_type	Dictionary indicating the type of each node
StochasticValue	Pre-determined function to generate stochastic using log-normal function
simulate_blackbox	Function to calculate probability value based on inputs
operation_disruption	Input parameter for the fit_blackbox function
seasonal_demand	Input parameter for the fit_blackbox function
probValue	Probability value obtained from fit_blackbox
white_box	Data structure to store intermediate results for nodes not of type 1
node_capacity	Capacity of the current node
max_demand	Maximum demand that all nodes' capacity must meet
reliability	Reliability metric for the solution

4.4. Sim-learnheuristic

It is well-established in stochastic literature that excellent solutions obtained for the deterministic version of an optimization problem carry over its quality to its stochastic counterpart, albeit with an understanding that some level of variability comes with the random components [8]. The sim-learnheuristic uses the strength of deterministic optimization to obtain solutions that exhibit a remarkable degree of stability to the stochastic-non-static counterpart. The sim-learnheuristic employs a multi-step process to address SNS-CDP. In the initial stage, it transforms the stochastic-non-static problem into a deterministic problem by substituting expected values for stochastic-non-static parameters. Subsequently, the problem is resolved using the MSBR algorithm, the metaheuristic component, followed by refinement through a short simulation enhanced by the log-normal distribution function and machine learning techniques. This process iterates until the best solutions are achieved within the designated time frame. The pool of the best stochastic and non-static solutions is dynamically updated during this phase. In the next phase, a simulation with a high number of replications is applied to the solutions. This long simulation enables us to attain a more accurate estimation of the solutions. Finally, in the last step, the algorithm computes the reliability and assesses risks associated with the solutions, providing valuable statistical tools for result analysis. Figure 4 shows our sim-learnheuristic approach for solving the SNS-CDP. Based on the mathematical formulations provided for the stochastic CDP in Section 3, specifically the constraint 3.5, the reliability of a solution S is the probability that this solution is feasible. This approach guarantees a comprehensive optimization process, resulting in top-quality solutions with strong reliability assessments.

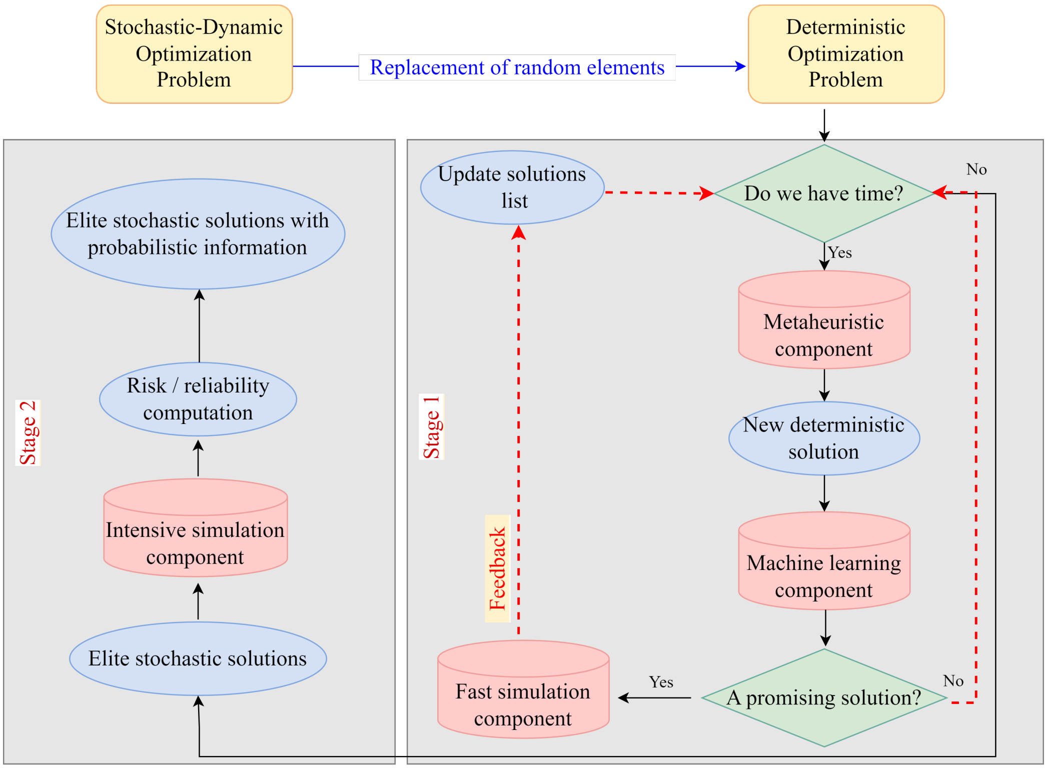

Stochastic-Dynamic Optimization Problem
Replacement of random elements
Deterministic Optimization Problem
Stage 2
Elite stochastic solutions with probabilistic information
Risk / reliability computation
Intensive simulation component
Elite stochastic solutions
Stage 1
Update solutions list
Do we have time?
No
Yes
Metaheuristic component
New deterministic solution
Machine learning component
Feedback
Fast simulation component
Yes
A promising solution?
No

Figure 4. The sim-learnheuristic schema for solving the SNS-CDP.

5. Computational experiments

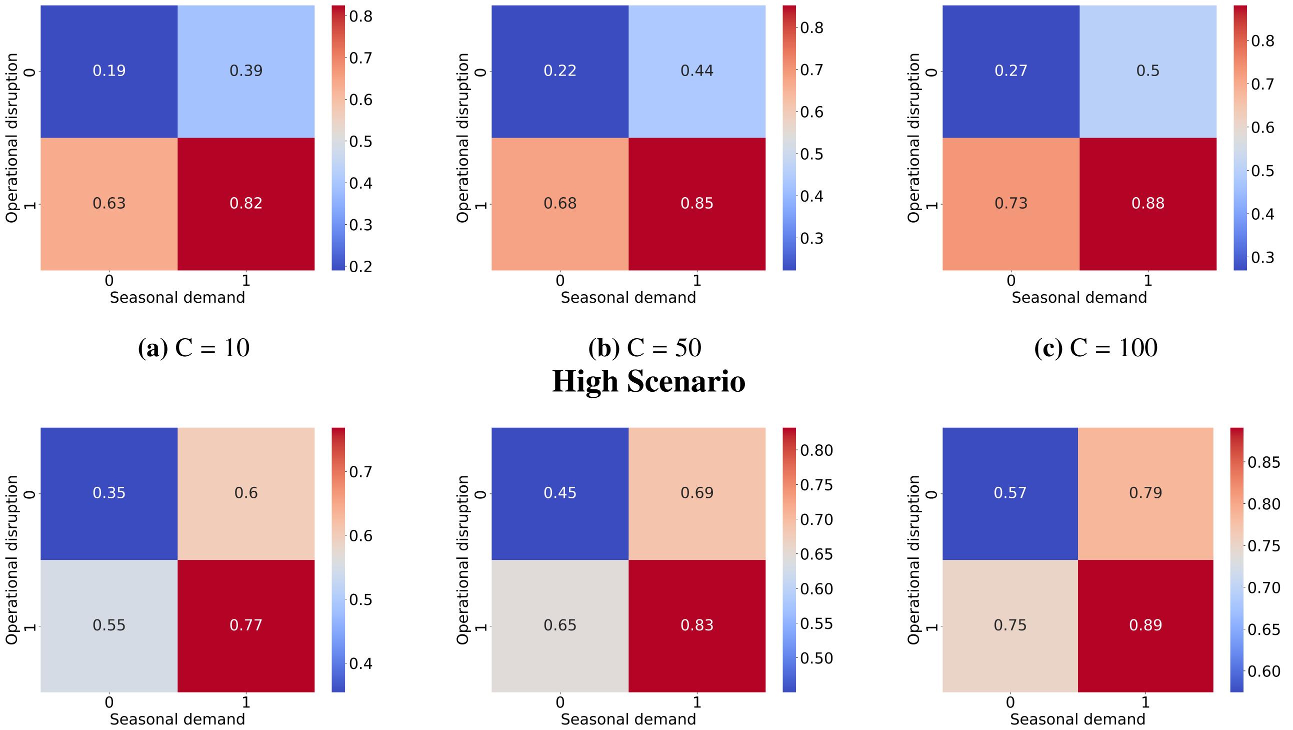
In this section, we explore four distinct variants of the CDP, each characterized by specific assumptions regarding facility behavior. In the first scenario, a deterministic version of the CDP is considered. In the second scenario, it is assumed that the capacity of the facilities follows a log-normal distribution function to define a stochastic scenario. In the third scenario, it is assumed that the capacities of the nodes are non-static using the whitebox_black box mechanism and the multivariate logistic regression model. Finally, in the combination scenario, there are two types of facilities. It is assumed that facilities with odd identification numbers are non-static, while the even ones are stochastic facilities. Additionally, each variant of the CDP has been solved twice. First, we consider the deterministic solution in stochastic, non-static, and stochastic-non-static environments. Then, we solve the stochastic, non-static, and stochastic-non-static problems using MSBR-simulation, MSBR-machine learning, and MSBR-simulation-machine learning, respectively.

Two groups of instances are considered to be solved. First, the GKD benchmark, constructed using Euclidean distances, with node coordinates generated from a uniform distribution ranging between 0 and 10. This dataset is originally from [21] and comprises two subsets: GKD-b (instances with 50 and 150 nodes) and GKD-c (instances with 500 nodes). The MDG benchmark incorporates real numbers randomly selected from a uniform distribution between 0 and 1000. This dataset, initially presented by [7], consists of instances with 500 nodes. In the scenarios with the stochastic capacities (scenarios two and four), the capacity of each facility is depicted as a random variable, denoted as $\hat{C}_i$, using the log-normal probability distribution function with two parameters (μ, σ) that are determined using the deterministic mean capacity parameter of the facilities (c_i). Additionally, the number of short and long simulation iterations in these scenarios is set at 100 and 1000, respectively.

Two sets of parameters for the stochastic and non-static problems have been defined. Table 2 shows the corresponding values for the black box coefficients β_0, β_1, β_2, β_3, and σ level of the log-normal function in low and high scenarios. Figure 5 visualizes the probability of selecting a new node using the black box model based on the chosen set of coefficients for the high and low scenarios. Three different values for the capacity, namely 10, 50, and 100, are considered. Figures 5(a)–(c) depict the visualization with the high-level set of coefficients, while Figures 5(d)–(f) show the visualization with the low-level set of coefficients. Seasonal demand and operational disruption are considered binary variables, each taking values of either 0 or 1. In Figure 5, the cells $(0, 0)$ show the probability of selecting a node under the non-promising condition. Conversely, the cells $(1, 1)$ represent the probability of selecting a node when both variables are in the promising condition.

Table 2. Parameter values for non-static and stochastic components.

	non-static component				stochastic component
	β_0	β_1	β_2	β_3	σ
High	-1.5	0.005	2.0	1.0	10.0
Low	-0.7	0.010	0.8	1.0	5.0



(a) C = 10 **(b)** C = 50 **(c)** C = 100

High Scenario

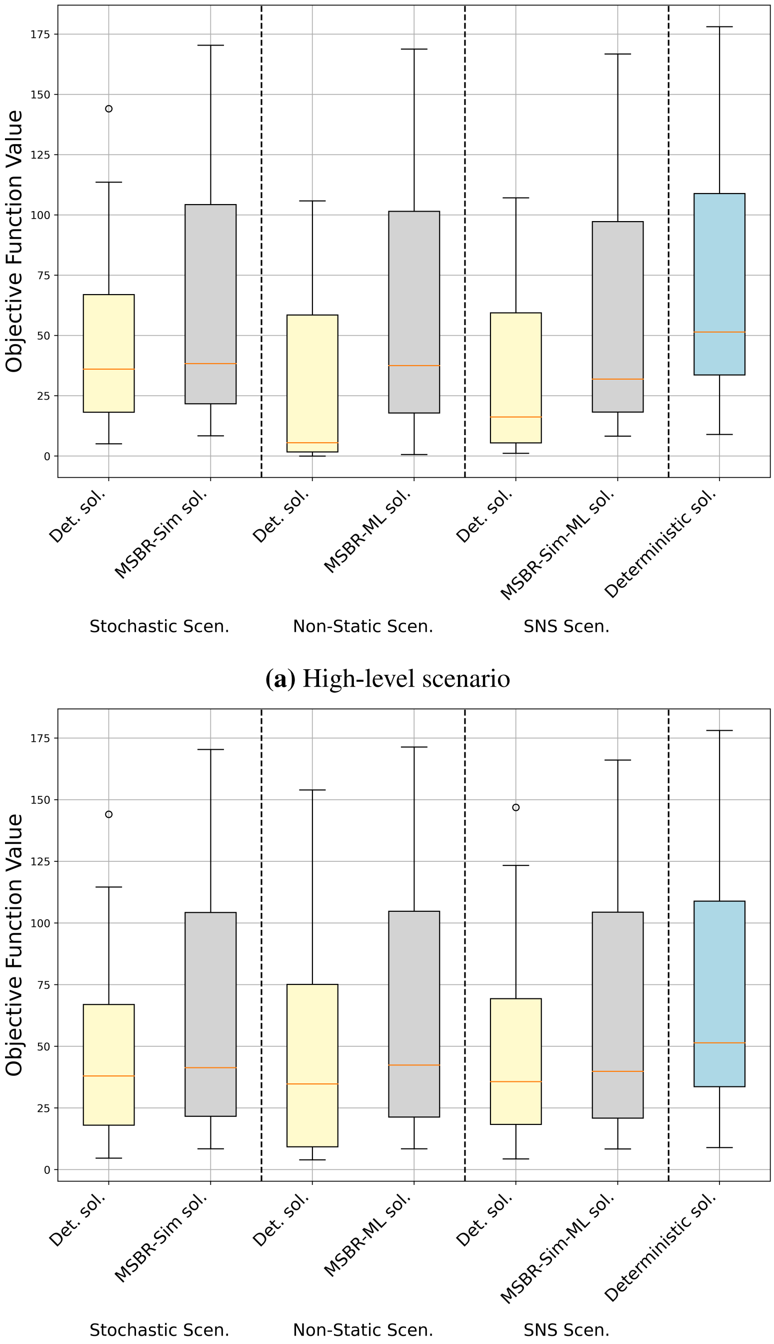
(d) C = 10 **(e)** C = 50 **(f)** C = 100

Low Scenario

Figure 5. Behaviors of the black box using three different capacities and high and low coefficients' values.

The algorithms are implemented using Python 3.10.5 and executed on an Intel(R) Core(TM)i3 CPU @ 2.00GHz with 8 GB of RAM. The sim-learnheuristic code is provided at: `https://github.com/Elnazghorbani/Sim-learnheuristic_CDP`. We imposed a maximum execution time of 60s for smaller and medium-sized instances (those with fewer than 500 nodes) and extended it to 180s for larger instances with 500 nodes. Tables 3 and 4 represent the results achieved for solving the different variants of the CDP for the high and low levels of coefficients, respectively. In the first column of both tables, the instances are introduced, then the deterministic solution using the MSBR algorithm is presented. The deterministic solution in the stochastic environment column reveals the performance of the deterministic solution when tested in a stochastic environment, along with its reliability (percentage of the feasibility of the solution) measure. Then, in the next column, the stochastic objective function values generated by the MSBR-simulation(sim) algorithm are presented. Similar columns are repeated for the non-static and stochastic-non-static scenarios. The solutions in the columns MSBR-machine learning(ml) and MSBR-sim-ml are provided considering the solo stochastic and the stochastic-non-static scenarios, respectively. Finally, the gaps(%) between the MSBR-sim-ml solution and deterministic solution in a non-static environment (a), MSBR-sim (b), and MSBR-ml (c) for all the instances are calculated to compare the sim-learnheuristic with other methodologies.

Figure 6 presents a comparison of solutions across various scenarios for all instances. Figure 6(a) illustrates the behavior of the methodologies with high-level coefficients, while Figure 6(b) shows the results for low-level coefficients. As expected, the deterministic values serve as upper bounds for all solutions. Notably, across solution methodologies—MSBR-sim, MSBR-ml, and MSBR-sim-ml—the objective function values consistently surpass those of deterministic solutions in stochastic, non-static, and combined stochastic-non-static environments for both levels of coefficients. The results underscore the robustness of the sim-learnheuristic algorithm, demonstrating its efficacy alongside MSBR-sim and MSBR-ml approaches in addressing combined uncertainties within the stochastic-non-static scenario.

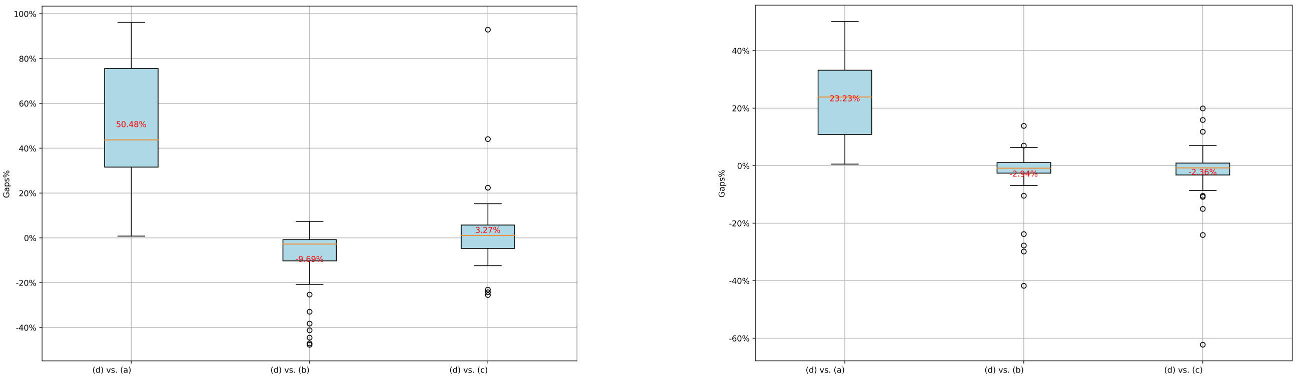


(a) High-level scenario

(b) Low-level scenario

Figure 6. Comparison of solutions across different scenarios for all the instances.

Figure 7 compares the percentage gaps of the MSBR-sim-ml with other methodologies. The data is provided in the last three columns of Tables 3 and 4. These figures highlight the significant positive gaps between the solutions provided by the MSBR-sim-ml and the deterministic solutions in a stochastic-non-static environment for both levels of coefficients. Figure 7(a) demonstrates minimal gaps between MSBR-sim-ml and other methodologies, such as MSBR-sim and MSBR-ml. Additionally, Figure 7(b) confirms the similar behavior of MSBR-sim-ml to MSBR-sim and MSBR-ml in low-level scenarios. Besides, it indicates that even in low-level scenarios, the performance of the sim-learnheuristic is superior to the deterministic solution in a stochastic-non-static environment.



(a) High-level scenario

(b) Low-level scenario

Figure 7. Comparison of percentage gaps between MSBR-Sim-ML and other solution methods.

Table 3. Results of solving the different variants of CDP by MSBR-Sim, MSBR-ML, and MSBR-Sim-ML with the high-level coefficients.

Instance	Deterministic Sol.	Solution to Stochastic Problem				Solution to Non-Static Problem				Solution to Stochastic-Non-Static Problem				Gaps%		
		Deterministic Sol. in Stochastic Env.	Reliability	MSBR-Sim Sol. (a)	Reliability	Deterministic Sol. in Non-Static Env.	Reliability	MSBR-ML Sol. (b)	Reliability	Deterministic Sol. in Stochastic-Non-Static Env. (c)	Reliability	MSBR-Sim-ML Sol. (d)	Reliability	(d)-(c)	(d)-(a)	(d)-(b)
GKD-b_11_n50_b02_m5.txt	147.2	83.5	0.56	142.5	1.00	76.5	0.90	111.3	0.79	86.3	0.59	143.4	0.99	39.8	0.6	22.4
GKD-b_12_n50_b02_m5.txt	178.1	144.1	0.81	170.4	1.00	45.1	0.25	168.8	1.00	84.4	0.47	166.8	0.99	49.4	-2.2	-1.2
GKD-b_13_n50_b02_m5.txt	96.1	58.4	0.61	91.7	1.00	84.6	0.96	92.4	0.96	61.8	0.64	89.2	0.97	30.7	-2.9	-3.7
GKD-b_14_n50_b02_m5.txt	84.6	64.3	0.76	81.2	1.00	28.3	0.34	69.3	0.94	59.3	0.70	79.6	0.99	25.5	-2.0	13.0
GKD-b_15_n50_b02_m5.txt	154.9	111.4	0.72	147.3	1.00	57.2	0.37	121.7	0.83	58.1	0.38	143.7	0.98	59.6	-2.5	15.3
GKD-b_16_n50_b02_m15.txt	77.7	55.2	0.71	76.1	0.98	40.8	0.53	67.0	0.95	46.1	0.59	60.8	0.87	24.2	-25.3	-10.2
GKD-b_17_n50_b02_m15.txt	41.8	21.6	0.52	35.8	0.98	23.2	0.56	33.8	0.98	13.5	0.32	26.9	0.76	50.0	-33.0	-25.5
GKD-b_18_n50_b02_m15.txt	108.5	69.0	0.64	104.0	1.00	99.8	0.92	103.4	1.00	69.1	0.64	96.7	0.99	28.5	-7.5	-6.9
GKD-b_19_n50_b02_m15.txt	119.1	112.0	0.96	114.6	1.00	64.3	0.54	108.6	0.97	87.4	0.73	110.6	0.99	21.0	-3.6	1.8
GKD-b_20_n50_b02_m15.txt	115.3	58.1	0.50	107.4	1.00	105.8	0.92	106.0	1.00	59.6	0.52	104.9	0.97	43.2	-2.4	-1.0
GKD-b_41_n150_b02_m15.txt	163.9	92.1	0.56	156.2	1.00	87.9	0.54	152.8	1.00	93.6	0.58	152.8	1.00	38.8	-2.2	0.0
GKD-b_42_n150_b02_m15.txt	84.0	66.3	0.79	74.0	0.97	7.1	0.08	75.8	1.00	19.7	0.24	79.9	1.00	75.3	7.4	5.1
GKD-b_43_n150_b02_m15.txt	62.5	39.5	0.63	57.2	0.99	6.4	0.10	58.2	1.00	12.6	0.20	54.7	1.00	76.9	-4.6	-6.5
GKD-b_44_n150_b02_m15.txt	100.6	63.8	0.63	95.4	0.99	0.1	0.00	91.7	1.00	5.0	0.10	95.9	1.00	94.8	0.5	4.4
GKD-b_45_n150_b02_m15.txt	104.4	59.0	0.57	99.4	1.00	1.3	0.01	89.6	0.91	17.6	0.17	94.4	0.95	81.3	-5.3	5.1
GKD-b_46_n150_b02_m45.txt	123.3	69.3	0.56	116.6	0.99	62.6	0.51	111.7	1.00	55.9	0.45	115.7	1.00	51.7	-1.0	3.5
GKD-b_47_n150_b02_m45.txt	162.8	112.3	0.69	150.0	0.99	100.3	0.62	148.5	1.00	95.1	0.58	153.8	1.00	38.2	2.5	3.4
GKD-b_48_n150_b02_m45.txt	99.8	70.1	0.70	94.3	0.97	4.0	0.04	86.1	0.99	17.7	0.18	93.1	1.00	81.0	-1.2	7.5
GKD-b_49_n150_b02_m45.txt	166.3	113.6	0.68	165.4	1.00	85.0	0.51	158.7	0.99	107.1	0.64	152.1	0.94	29.6	-8.7	-4.3
GKD-b_50_n150_b02_m45.txt	110.1	65.7	0.60	105.4	0.99	90.8	0.83	100.9	0.96	65.0	0.59	98.8	0.94	34.3	-6.7	-2.1
MDG-b_01_n500_b02_m50.txt	52.0	31.0	0.60	34.0	0.99	0.0	0.00	23.0	0.96	1.2	0.02	23.1	0.97	94.8	-47.1	0.2
MDG-b_02_n500_b02_m50.txt	43.6	25.4	0.58	25.9	1.00	4.1	0.09	29.2	0.97	14.7	0.34	26.4	1.00	44.2	1.9	-10.6
MDG-b_03_n500_b02_m50.txt	43.6	24.5	0.56	29.7	0.96	0.0	0.00	26.4	0.98	1.1	0.03	29.7	0.99	96.2	0.1	11.3
MDG-b_04_n500_b02_m50.txt	44.5	29.3	0.71	31.6	1.00	0.0	0.00	24.0	1.00	3.6	0.08	21.4	0.63	83.3	-47.7	-12.4
MDG-b_05_n500_b02_m50.txt	46.5	25.1	0.69	32.0	0.96	0.1	0.00	23.4	1.00	6.2	0.13	26.5	1.00	76.5	-20.7	11.7
MDG-b_06_n500_b02_m50.txt	45.3	28.7	0.63	35.0	1.00	7.5	0.17	33.2	0.98	20.6	0.46	30.3	0.99	32.0	-15.7	-9.7
MDG-b_07_n500_b02_m50.txt	42.6	24.8	0.58	34.6	0.99	0.0	0.00	21.0	0.86	13.7	0.32	24.5	1.00	44.0	-41.2	14.4
MDG-b_08_n500_b02_m50.txt	48.0	29.6	0.80	38.4	0.99	23.9	0.50	41.3	1.00	23.9	0.70	33.6	0.93	28.8	-14.6	-23.1
MDG-b_09_n500_b02_m50.txt	49.4	37.6	0.76	38.3	1.00	15.4	0.31	32.9	0.98	26.3	0.53	26.5	1.00	0.8	-44.5	-24.3
MDG-b_10_n500_b02_m50.txt	50.9	34.5	0.68	37.5	1.00	0.0	0.00	28.7	1.00	2.2	0.04	27.1	0.98	91.9	-38.3	-5.7
GKD-c_01_n500_b02_m50.txt	9.1	6.9	0.76	8.4	1.00	1.9	0.21	8.5	0.98	4.9	0.54	8.6	1.00	43.4	2.4	0.9
GKD-c_02_n500_b02_m50.txt	9.3	6.0	0.64	9.0	0.97	1.9	0.20	4.6	0.52	3.9	0.42	8.3	0.97	52.5	-8.8	44.1
GKD-c_03_n500_b02_m50.txt	9.2	7.0	0.76	8.6	0.99	3.4	0.37	8.5	1.00	5.6	0.61	8.3	0.99	32.4	-3.4	-2.1
GKD-c_04_n500_b02_m50.txt	9.0	5.1	0.57	8.6	0.99	1.4	0.15	8.4	1.00	2.8	0.31	8.4	0.98	66.8	-2.3	0.3
GKD-c_05_n500_b02_m50.txt	9.0	7.3	0.81	8.8	0.98	4.0	0.44	8.4	1.00	6.2	0.69	8.5	1.00	27.3	-3.7	1.2
GKD-c_06_n500_b02_m50.txt	9.0	6.1	0.67	8.7	0.99	1.8	0.20	7.5	0.90	4.2	0.46	8.5	0.99	51.0	-2.1	11.9
GKD-c_07_n500_b02_m50.txt	9.1	8.0	0.88	8.7	1.00	4.1	0.46	8.3	0.97	6.3	0.70	8.6	0.99	26.4	-0.8	3.5
GKD-c_08_n500_b02_m50.txt	9.3	6.7	0.72	8.9	1.00	4.6	0.49	8.6	1.00	5.8	0.62	8.9	1.00	35.0	0.0	3.4
GKD-c_09_n500_b02_m50.txt	9.0	7.0	0.78	8.7	1.00	2.8	0.31	0.6	0.35	5.9	0.66	8.7	0.99	31.9	0.2	92.9
GKD-c_10_n500_b02_m50.txt	9.1	6.8	0.75	8.8	1.00	0.0	0.00	8.4	0.97	1.2	0.13	8.7	1.00	86.6	-1.2	3.0
Average	71.5	47.2	0.68	65.2	0.99	28.7	0.34	60.3	0.94	31.9	0.43	61.7	0.97	50.5	-9.7	3.3

Table 4. Results of solving the different variants of CDP by MSBR-Sim, MSBR-ML, and MSBR-Sim-ML with the low-level coefficients.

		Solution to Stochastic Problem				Solution to Non-Static Problem				Solution to Stochastic-Non-Static Problem				Gaps%		
Instance	Deterministic Sol.	Deterministic Sol. in Stochastic Env.	Reliability	MSBR-Sim Sol. (a)	Reliability	Deterministic Sol. in Non-Static Env.	Reliability	MSBR-ML Sol. (b)	Reliability	Deterministic Sol. in Stochastic-Non-Static Env. (c)	Reliability	MSBR-Sim-ML Sol. (d)	Reliability	(d)-(c)	(d)-(a)	(d)-(b)
GKD-b_11_n50_b02_m5.txt	147.2	83.5	0.57	142.8	1.00	140.0	0.95	142.9	1.00	89.4	0.61	137.3	1.00	34.9	-4.0	-4.1
GKD-b_12_n50_b02_m5.txt	178.1	144.1	0.81	170.4	1.00	153.2	0.86	171.4	1.00	146.9	0.83	166.1	1.00	11.6	-2.6	-3.2
GKD-b_13_n50_b02_m5.txt	96.1	58.4	0.61	91.9	1.00	65.2	1.00	92.2	1.00	62.1	0.65	92.2	1.00	32.7	0.3	0.0
GKD-b_14_n50_b02_m5.txt	84.6	64.3	0.76	81.2	1.00	75.2	0.89	80.6	1.00	70.4	0.83	81.2	1.00	13.3	0.0	0.7
GKD-b_15_n50_b02_m5.txt	154.9	111.4	0.72	147.3	1.00	143.6	0.93	150.0	0.98	114.0	0.74	148.7	0.99	23.3	0.9	-0.9
GKD-b_16_n50_b02_m15.txt	77.7	55.2	0.71	76.1	0.98	66.6	0.97	74.9	1.00	65.5	0.84	71.3	0.94	8.1	-6.9	-5.1
GKD-b_17_n50_b02_m15.txt	41.8	21.6	0.52	35.8	0.98	26.2	0.97	39.8	0.98	22.3	0.53	34.6	1.00	35.5	-3.4	-15.0
GKD-b_18_n50_b02_m15.txt	108.5	69.0	0.64	103.9	1.00	75.1	0.99	104.3	1.00	71.3	0.66	103.7	0.99	31.2	-0.2	-0.6
GKD-b_19_n50_b02_m15.txt	119.1	114.6	0.96	115.3	1.00	115.2	0.97	115.3	0.96	110.2	0.98	111.6	1.00	1.2	-3.3	-3.3
GKD-b_20_n50_b02_m15.txt	115.3	58.1	0.50	107.3	1.00	60.1	1.00	111.3	1.00	61.3	0.53	111.8	1.00	45.1	4.0	0.5
GKD-b_41_n150_b02_m15.txt	163.9	92.1	0.56	156.2	1.00	154.0	0.95	156.0	0.99	106.1	0.65	154.4	0.99	31.3	-1.1	-1.0
GKD-b_42_n150_b02_m15.txt	84.0	66.3	0.79	74.0	0.97	75.5	0.90	76.3	1.00	71.7	0.85	79.0	1.00	9.2	6.3	3.4
GKD-b_43_n150_b02_m15.txt	62.5	39.5	0.63	57.2	0.99	57.4	0.92	59.8	0.97	40.9	0.66	58.5	0.99	30.0	2.2	-2.3
GKD-b_44_n150_b02_m15.txt	100.6	63.8	0.63	95.4	0.99	55.3	0.55	98.9	1.00	51.7	0.51	95.6	0.99	45.9	0.2	-3.4
GKD-b_45_n150_b02_m15.txt	104.4	59.0	0.57	99.4	1.00	66.3	0.64	102.4	1.00	56.2	0.54	101.5	0.98	44.7	2.1	-0.9
GKD-b_46_n150_b02_m45.txt	123.3	69.3	0.56	116.6	0.99	90.1	0.73	114.3	0.97	65.1	0.53	115.4	1.00	43.6	-1.1	0.9
GKD-b_47_n150_b02_m45.txt	162.8	112.3	0.69	150.0	0.99	144.1	0.89	155.7	1.00	118.4	0.73	157.0	0.99	24.6	4.5	0.8
GKD-b_48_n150_b02_m45.txt	99.8	70.1	0.70	94.3	0.97	72.8	0.73	96.1	1.00	63.9	0.64	93.6	1.00	31.8	-0.7	-2.7
GKD-b_49_n150_b02_m45.txt	166.3	113.6	0.68	165.4	1.00	139.0	0.84	161.8	1.00	123.4	0.74	163.4	1.00	24.5	-1.2	1.0
GKD-b_50_n150_b02_m45.txt	110.1	65.7	0.60	105.4	0.99	68.1	0.97	106.2	1.00	69.0	0.63	106.5	1.00	35.2	1.0	0.3
MDG-b_01_n500_b02_m50.txt	52.0	31.0	0.60	34.0	0.99	12.5	0.24	33.2	0.95	23.9	0.46	39.4	0.95	39.3	13.9	15.9
MDG-b_02_n500_b02_m50.txt	43.6	25.4	0.58	25.9	1.00	15.8	0.36	32.5	0.99	23.2	0.53	26.2	0.98	11.2	1.0	-24.1
MDG-b_03_n500_b02_m50.txt	43.6	24.5	0.56	29.7	0.96	9.5	0.22	25.5	0.96	21.8	0.50	31.9	1.00	31.7	7.0	20.0
MDG-b_04_n500_b02_m50.txt	44.5	33.2	0.73	35.3	1.00	24.0	0.54	40.4	1.00	22.4	0.76	24.9	1.00	10.0	-41.8	-62.3
MDG-b_05_n500_b02_m50.txt	46.5	32.0	0.69	33.1	1.00	31.6	0.68	31.7	1.00	32.4	0.70	32.6	1.00	0.6	-1.5	2.8
MDG-b_06_n500_b02_m50.txt	45.3	32.3	0.71	35.0	0.99	27.2	0.60	29.8	0.99	27.1	0.71	27.4	0.99	1.2	-27.7	-8.6
MDG-b_07_n500_b02_m50.txt	42.6	24.8	0.58	34.6	1.00	19.4	0.46	26.0	1.00	23.8	0.53	28.0	0.99	15.0	-23.8	7.0
MDG-b_08_n500_b02_m50.txt	48.0	38.4	0.80	44.5	1.00	38.0	0.95	44.5	1.00	39.0	0.81	40.3	1.00	3.3	-10.4	-10.4
MDG-b_09_n500_b02_m50.txt	49.4	37.6	0.76	38.3	1.00	30.1	0.91	32.7	1.00	27.2	0.75	29.5	0.99	7.8	-29.8	-10.8
MDG-b_10_n500_b02_m50.txt	50.9	34.5	0.68	37.5	0.96	17.3	0.34	34.3	1.00	31.7	0.62	38.9	0.98	18.5	3.5	11.8
GKD-c_01_n500_b02_m50.txt	9.1	6.9	0.76	8.5	0.97	6.6	0.73	8.5	0.99	6.8	0.74	8.8	0.99	23.3	4.0	3.1
GKD-c_02_n500_b02_m50.txt	9.3	5.6	0.61	9.0	0.99	6.1	0.66	8.7	0.99	5.6	0.60	9.0	1.00	38.1	-0.1	3.2
GKD-c_03_n500_b02_m50.txt	9.2	7.0	0.76	8.6	0.99	8.5	0.92	8.7	0.99	7.1	0.77	8.5	0.97	16.0	-1.8	-2.9
GKD-c_04_n500_b02_m50.txt	9.0	5.1	0.57	8.6	0.98	4.0	0.44	8.8	1.00	4.4	0.48	8.7	0.99	50.2	1.4	-0.6
GKD-c_05_n500_b02_m50.txt	9.0	7.3	0.81	8.8	0.99	7.6	0.96	8.5	1.00	7.8	0.87	8.6	0.99	9.1	-2.3	1.3
GKD-c_06_n500_b02_m50.txt	9.0	6.0	0.67	8.7	1.00	5.8	0.65	8.7	0.99	5.8	0.65	8.6	1.00	32.4	-1.0	-1.3
GKD-c_07_n500_b02_m50.txt	9.1	5.9	0.66	8.7	0.99	8.0	0.95	8.6	1.00	8.1	0.89	8.4	0.99	4.1	-2.6	-1.8
GKD-c_08_n500_b02_m50.txt	9.3	6.7	0.72	8.9	1.00	8.7	0.94	9.0	1.00	6.8	0.73	8.8	1.00	22.5	-1.3	-2.5
GKD-c_09_n500_b02_m50.txt	9.0	4.7	0.52	8.7	1.00	7.8	0.87	8.7	1.00	7.6	0.84	8.7	1.00	12.4	-0.2	-0.3
GKD-c_10_n500_b02_m50.txt	9.1	6.8	0.75	8.8	1.00	7.3	0.81	8.6	0.98	6.5	0.72	8.7	1.00	24.9	-1.2	0.9
Average	71.5	47.7	0.67	65.5	0.99	53.5	0.77	65.7	0.99	48.0	0.68	64.7	0.99	23.2	-2.9	-2.4

6. Conclusions

In this study, novel variants of the CDP have been explored by considering the storage capacity parameter of the facilities as stochastic, non-static, and a combination of stochastic and non-static values. To address the proposed combination problem, a sim-learnheuristic solution approach is developed in order to use the power of machine learning, metaheuristic, and simulation to respond to the stochastic and non-static characteristics of the problem. In this approach, the facilities are classified into two distinct categories based on their unique characteristics. In particular, stochastic facilities are characterized by capacities produced by a log-normal distribution function, while non-static facilities are modeled using a machine learning framework to predict their non-static behavior. Defining both a black box and a white box allows for the creation of a procedure designed to receive real data and generate an approximation thereof. Notably, a multivariate logistic regression model has been applied to perform a classification prediction. The predictions are multiplied by the deterministic capacities. Thus, the non-static capacities can be either zero or equal to the corresponding deterministic capacity value. The logistic regression model incorporates independent factors, each contributing unique coefficients to the modeling process. A constructive multi-start biased-randomized (MSBR) technique is proposed as the metaheuristic component to resolve the problem's deterministic version by replacing the random capacities' expected values. Additionally, a Monte Carlo simulation technique is incorporated to provide the solution for the combination model by a maximum iteration number.

To provide a comprehensive assessment of the results, four different scenarios for the CDP are considered: deterministic, solo-stochastic, solo-non-static, and stochastic-non-static. The deterministic scenario has been answered by using the proposed metaheuristic algorithm. However, the second scenario is resolved by MSBR and the simulation component. Besides, the solo-non-static variant has been answered by a combination of MSBR and our machine-learning component. Finally, the stochastic-non-static scenario has been addressed by our sim-learnheuristic method, which is a combination of the metaheuristic, simulation, and machine-learning components. Two types of comparisons have been conducted. First, we generate deterministic solutions within stochastic, non-static, or stochastic-non-static environments, accompanied by their respective reliability values, to facilitate comparisons with our methodologies. Second, we evaluate the efficiency of our proposed sim-learnheuristic (MSBR-sim-ml) algorithm by comparing it with other solution approaches like MSBR-sim and MSBR-ml. This comparison aims to demonstrate the effectiveness of our methodology in addressing two types of uncertainties.

As expected, the results demonstrated that the deterministic scenario produced the most favorable outcomes. However, the results for the deterministic scenario are not reliable, since it underestimates the real-world constraints. In the first comparison, the sim-learnheuristic outperforms deterministic solutions within the stochastic-non-static environment, as well as the other two solution approaches (MSBR-sim and MSBR-ml). The second comparison highlights the strong performance of the sim-learnheuristic, with only minor gaps observed in its behavior compared with the two other methodologies.

In the future, prospective research should explore deeper the complexities associated with non-static storage capacities. For instance, exploring more intricate models, such as neural networks or other advanced methods, can enhance the predictions. Moreover, employing sophisticated modeling techniques like time-series forecasting has the potential to significantly enhance the predictions of facility capacity. Besides, in future works, a comprehensive analysis can be conducted to discover other factors influencing changes in facility capacity and identify additional variables that may impact the dynamic values. Finally, future studies can improve our understanding of this field by proposing other methodologies to compare the results achieved in this paper using different solution approaches.

Author contributions

A. A. Juan and J. Panadero developed the main idea of the study; E. Ghorbani carried out the research, testing the instances; J. F. Gomez and E. Ghorbani planned the algorithms used in the research and implemented them in software; E. Ghorbani wrote the first draft of the paper; J. Panadero checked the results to make sure they were accurate; J. Panadero and A. A. Juan supervised the project, reviewed the paper, and provided guidance throughout the process. All authors have read and approved the final version of the manuscript for publication.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work has been partially funded by the Spanish Ministry of Science and Innovation (PID2022-138860NB-I00, RED2022-134703-T) as well as by the Generalitat Valenciana (PROMETEO/2021/065).

Conflict of interest

The authors declare no conflict of interest.

References

1. A. Ala, M. Deveci, E. A. Bani, A. H. Sadeghi, Dynamic capacitated facility location problem in mobile renewable energy charging stations under sustainability consideration, *Sustain. Comput. Infor.*, **41** (2024), 100954. https://doi.org/10.1016/j.suscom.2023.100954
2. N. Aydin, A. Murat, B. S. Mordukhovich, Sample intelligence-based progressive hedging algorithms for the stochastic capacitated reliable facility location problem, *Artif. Intell. Rev.*, **57** (2024), 135.
3. M. Bieniek, A note on the facility location problem with stochastic demands, *Omega*, **55** (2015), 53–60. https://doi.org/10.1016/j.omega.2015.02.006
4. M. Chica, A. A. Juan, C. Bayliss, O. Cordón, W. D. Kelton, Why simheuristics? benefits, limitations, and best practices when combining metaheuristics with simulation, *SORT-Stat. Oper. Res. T.*, **44** (2020), 311–334.
5. B. R. Cobb, R. Rumí, A. Salmerón, Inventory management with log-normal demand per unit time, *Comput. Oper. Res.*, **40** (2013), 1842–1851. https://doi.org/10.1016/j.cor.2013.01.017

6. P. B. de Oliveira, R. S. de Camargo, G. de Miranda Júnior, A. X. Martins, A computational study of a decomposition approach for the dynamic two-level uncapacitated facility location problem with single and multiple allocation, *Comput. Ind. Eng.*, **151** (2021), 106964.

7. A. Duarte, R. Marti, Tabu search and grasp for the maximum diversity problem, *Eur. J. Oper. Res.*, **178** (2007), 71–84. https://doi.org/10.1016/j.ejor.2006.01.021

8. E. Ghorbani, T. Fluechter, L. Calvet, M. Ammouriova, J. Panadero, A. A. Juan, Optimizing energy consumption in smart cities' mobility: Electric vehicles, algorithms, and collaborative economy, *Energies*, **16** (2023).

9. F. Glover, K. Ching-Chung, K. S. Dhir, A discrete optimization model for preserving biological diversity, *Appl. Math. Model.*, **19** (1995), 696–701. https://doi.org/10.1016/0307-904X(95)00083-V

10. F. Glover, C. C. Kuo, K. Dhir, Heuristic algorithms for the maximum diversity problem, *J. Inform. Optim. Sci.*, 1998, 19. https://doi.org/10.1080/02522667.1998.10699366

11. J. F. Gomez, A. Martínez-Gavara, J. Panadero, A. A. Juan, R. Martí, A forward-backward simheuristic for the stochastic capacitated dispersion problem, *Mathematics*, **12** (2024), 909. https://doi.org/10.3390/math12060909

12. J. F. Gomez, J. Panadero, R. D. Tordecilla, J. Castaneda, A. A. Juan, A multi-start biased-randomized algorithm for the capacitated dispersion problem, *Mathematics*, **10** (2022).

13. J. F. Gomez, A. R. Uguina, J. Panadero, A. A. Juan, A learnheuristic algorithm for the capacitated dispersion problem under dynamic conditions, *Algorithms*, **16** (2023), 532. https://doi.org/10.3390/a16120532

14. A. Grasas, A. A. Juan, J. Faulin, J. de Armas, H. Ramalhinho, Biased randomization of heuristics using skewed probability distributions: A survey and some applications, *Comput. Ind. Eng.*, **110** (2017), 216–228. https://doi.org/10.1016/j.cie.2017.06.019

15. S. D. Jena, J. F. Cordeau, B. Gendron, Solving a dynamic facility location problem with partial closing and reopening, *Comput. Oper. Res.*, **67** (2016), 143–154. https://doi.org/10.1016/j.cor.2015.10.011

16. M. Landete, J. Peiró, H. Yaman, Formulations and valid inequalities for the capacitated dispersion problem, *Networks*, **81** (2023), 294–315. https://doi.org/10.1002/net.22132

17. Z. Lu, A. Martínez-Gavara, J. K. Hao, X. Lai, Solution-based tabu search for the capacitated dispersion problem, *Expert Syst. Appl.*, **223** (2023), 119856.

18. J. Maisano, A. Radchik, T. Ling, A log-normal model for demand forecasting in the national electricity market, *ANZIAM J.*, **57** (2016), 369–383. https://doi.org/10.21914/anziamj.v57i0.8910

19. R. Marti, A. Martinez-Gavara, J. Sanchez-Oro, The capacitated dispersion problem: An optimization model and a memetic algorithm, *Memetic Comp.*, **13** (2021), 131–146. https://doi.org/10.1007/s12293-020-00318-1

20. A. Martinez-Gavara, T. Corberan, R. Marti, Grasp and tabu search for the generalized dispersion problem, *Expert Syst. Appl.*, **173** (2021), 114703. https://doi.org/10.1016/j.eswa.2021.114703

21. R. Martí, M. Gallego, A. Duarte, A branch and bound algorithm for the maximum diversity problem, *Eur. J. Oper. Res.*, **200** (2010), 36–44. https://doi.org/10.1016/j.ejor.2008.12.023

22. M. Marufuzzaman, R. Gedik, M. S. Roni, A benders based rolling horizon algorithm for a dynamic facility location problem, *Comput. Ind. Eng.*, **98** (2016), 462–469. https://doi.org/10.1016/j.cie.2016.06.029

23. S. Mišković, Z. Stanimirović, I. Grujičić, An efficient variable neighborhood search for solving a robust dynamic facility location problem in emergency service network, *Electron. Notes Discrete Math.*, **47** (2015), 261–268. https://doi.org/10.1016/j.endm.2014.11.034

24. N. Mladenović, R. Todosijević, D. Urošević, M. Ratli, Solving the capacitated dispersion problem with variable neighborhood search approaches: From basic to skewed vns, *Comput. Oper. Res.*, **139** (2022), 105622. https://doi.org/10.1016/j.cor.2021.105622

25. R. H. Pearce, M. Forbes, Disaggregated benders decomposition and branch-and-cut for solving the budget-constrained dynamic uncapacitated facility location and network design problem, *Eur. J. Oper. Res.*, **270** (2018), 78–88.

26. J. Peiró, I. Jiménez, J. Laguardia, R. Martí, Heuristics for the capacitated dispersion problem, *Int. T. Oper. Res.*, **28** (2021), 119–141. https://doi.org/10.1111/itor.12799

27. B. S. Pimentel, G. R. Mateus, F. A. Almeida, Stochastic capacity planning and dynamic network design, *Int. J. Prod. Econ.*, **145** (2013), 139–149. https://doi.org/10.1016/j.ijpe.2013.01.019

28. R. M. Rosati, A. Schaerf, Multi-neighborhood simulated annealing for the capacitated dispersion problem, *Expert Syst. Appl.*, **255** (2024), 124484. https://doi.org/10.1016/j.eswa.2024.124484

29. D. J. Rosenkrantz, G. K. Tayi, S. Ravi, Facility dispersion problems under capacity and cost constraints, *J. Comb. Optim.*, **4** (2000), 7–33.

30. A. Silva, D. Aloise, L. C. Coelho, C. Rocha, Heuristics for the dynamic facility location problem with modular capacities, *Eur. J. Oper. Res.*, **290** (2021), 435–452. https://doi.org/10.1016/j.ejor.2020.08.018

31. L. Wang, Z. Zhang, C. Wu, D. Xu, X. Zhang, Approximation algorithms for the dynamic k-level facility location problems, *Theor. Comput. Sci.*, **853** (2021), 43–56.

32. J. Xu, S. Sen, Ensemble variance reduction methods for stochastic mixed-integer programming and their application to the stochastic facility location problem, *INFORMS J. Comput.*, **36** (2024), 587–599. https://doi.org/10.1287/ijoc.2021.0324