

Evaluating robustness of metaheuristics in stochastic variants of the CVRPTW

Konrad Pempera

Abstract—Modern logistics systems operate under uncertain and dynamically changing conditions, particularly in relation to customer demand and delivery time windows. This paper investigates a stochastic variant of the Capacitated Vehicle Routing Problem with Time Windows (CVRPTW), where both time-window boundaries and customer demands are perturbed across multiple scenarios. A complete MILP model integrating vehicle capacity constraints, soft time windows, and penalty-based feasibility relaxation is formulated. To solve large-scale stochastic instances, Tabu Search (TS) and Artificial Bee Colony (ABC) metaheuristics are developed and validated against an exact Gurobi solver and a greedy constructive baseline. The experimental evaluation, conducted over 1,200 stochastic scenarios derived from Solomon benchmarks, demonstrates that metaheuristics consistently outperform the greedy approach, with the ABC algorithm achieving a significant average cost reduction of 23.74%. Beyond average performance, the metaheuristics exhibit reduced variability and effectively mitigate extreme high-cost outcomes, ensuring higher operational reliability. Crucially, the observed reduction in variance and the mitigation of the most severe scenarios are highly significant for real-world applications, as they minimize operational risks and ensure predictable service levels in volatile environments. These results confirm that the proposed metaheuristics provide stable routing performance under uncertainty, primarily by prioritizing time-window compliance over simple distance minimization, which reduces penalties by up to 31.3%. The findings highlight a strategic trade-off where accepting slightly longer routes ensures significantly higher service predictability.

Keywords—CVRPTW; robustness; stochastic; time windows; MILP; TabuSearch; ABC

I. INTRODUCTION

IN an era of rapidly changing supply chains, demand is growing for flexible transportation services that serve store networks, pickup points, construction sites, and temporary events. Transport companies deliver daily materials and components whose sizes may vary from day to day. Customers typically specify acceptable delivery and unloading time windows (e.g. for safety or operational reasons), so each visit must not only occur at a scheduled time but also complete unloading within a designated time window.

Time windows do not need to be static – they can be modified due to delays, demand fluctuations, or customer requests. In such conditions, planning methods capable of

maintaining solution quality under changing input data are particularly important. While existing literature often addresses these challenges through formal robust optimization using uncertainty sets [1], [2] or focuses on specific stochastic elements like travel times [3], the approach presented in this work demonstrates characteristics of *operational robustness* through a different lens. The methodological innovation of this work lies in replacing the classical point-based evaluation with a comprehensive analysis across a broad spectrum of 1200 stochastic scenarios. This approach enables the assessment of algorithmic performance under diverse operational conditions ranging from highly favorable to adverse, providing a more complete representation of solution stability and reliability. By analyzing the behavior of the developed metaheuristic solutions, structural features of routes are identified, such as their inherent potential to mitigate delay propagation, which remain invisible in standard deterministic benchmarks. The algorithm's ability to intelligently explore the solution space means it adapts better to disruptions than constructive approaches, which enables reduction of operational costs and decreased frequency of time window violations.

Taking into account the fleet's limited capacity, operational time, route planning and load allocation become a combinatorial problem burdened with many realistic operational constraints: vehicle capacities (capacitated), time windows of availability, and variable demand at delivery points. In this paper, a mathematical model reflecting the described conditions is proposed, four solution methods are compared: an exact method (Gurobi solver) for small-scale instances, a greedy constructive baseline, and two tailored metaheuristics — Tabu Search (TS) and Artificial Bee Colony (ABC) — adapted to large-scale stochastic instances. An additional value of this work is the ability to observe to what extent these methods differ in terms of robustness to uncertainty and how they behave under extreme conditions. The experiments were conducted on benchmark data extended with stochastic modifications of customer time windows and demands.

A. Research hypotheses

To evaluate the effectiveness and stability under uncertainty robustness of the proposed approaches, the following research hypotheses (H1 - H5) were formulated:

H1. Metaheuristic algorithms (TS and ABC) provide solutions of comparable quality to exact method (Gurobi solver).

Konrad Pempera is with Department of Control and Quantum Computing, Wrocław University of Science and Technology, Wrocław, Poland (e-mail: konrad.pempera@pwr.edu.pl, ORCID: 0009-0002-0701-9639).



- H2.** Metaheuristics achieve a lower objective value than the greedy algorithm across a set of stochastic scenarios.
- H3.** The metaheuristics exhibit smaller variability of results across scenarios, which indicates higher stability and operational robustness with changing input data.
- H4.** The metaheuristics minimize the total penalty associated with time-window violations compared to the greedy approach. Under the assumption of a continuous service model without waiting times, the goal is to reduce the magnitude of unavoidable delay propagation.
- H5.** The metaheuristics reduce the loss tail — that is, the frequency and severity of extreme-cost scenarios. Fewer extreme cases indicate greater reliability of the method under difficult operating conditions.

The above hypotheses address not only average solution quality, but also their stability and reliability under uncertainty. The analysis aims to demonstrate that a reduced sensitivity to randomness, visualized through steeper ECDF curves, directly translates to higher service reliability compared to the potentially erratic performance of the greedy baseline. This allows the analysis to move beyond nominal optimization results and assess algorithm behavior in the presence of stochastic disturbances — a crucial factor in practice.

B. Contributions and Target Audience

Value is provided by this research for three primary groups of stakeholders. For researchers, a methodology for evaluating metaheuristics through the lens of performance distributions and ECDF curves is introduced, moving beyond simple average-case analysis. For industry, evidence-based insights are derived from 1200 diverse operational scenarios. Rather than focusing on a single best case, it is demonstrated how different algorithmic approaches maintain service-level stability across a wide variety of daily fluctuations, offering a more reliable basis for real-world decision-making. For solution providers, the clear competitive advantage of reliability-oriented routing is demonstrated, as it ensures schedule predictability even in the face of the inherent stochasticity of modern supply chains.

The remainder of this paper is organized as follows. Section II reviews the related literature on vehicle routing problems with time windows, with particular emphasis on stochastic and robust approaches, while also covering emerging artificial intelligence and machine learning applications in vehicle routing. Section III introduces the mathematical formulation of the considered capacitated vehicle routing problem with time windows. Section IV describes the experimental setup and presents a comprehensive evaluation of the proposed methods under 1200 stochastic scenarios. Finally, Section V concludes the paper and outlines directions for future research.

II. LITERATURE REVIEW

The Vehicle Routing Problem (VRP) belongs to the class of NP-hard problems and represents one of the most extensively studied topics in logistics and transportation. Its basic version, first proposed by Dantzig and Ramser in 1959, assumes the optimization of vehicle routes serving a set of customers while

minimizing the total travel cost [4]. Since then, many extensions have been developed, incorporating constraints such as vehicle capacity (CVRP [5]), service times, Multi-criteria VRP [6], and time windows within which customers can be served — the so-called Vehicle Routing Problem with Time Windows (VRPTW) [7], [8]. The capacitated version with time windows (CVRPTW) better reflects the conditions of real-world transportation systems, where resource availability and time requirements play a crucial role.

Modern logistics systems operate under increasingly dynamic conditions — not only in terms of variable transportation times and fluctuating fleet sizes but also regarding demand and changing time window boundaries. For example, in grocery retail networks, fresh products must be delivered regularly and in appropriate quantities to avoid losses related to product shelf life. In the construction industry, material deliveries must be synchronized with work schedules, as any delay may lead to downtime and costly disruptions. These situations demonstrate that routing models must account for both variability and uncertainty in operational data.

Traditional VRPTW and CVRPTW models assume deterministic data — fixed demand, travel times, and rigid time windows [9], [10]. However, in practice, these parameters frequently change, limiting the effectiveness of methods based solely on deterministic data. Consequently, there is growing interest in stochastic models and approaches that aim to ensure greater solution robustness against data variability.

The literature regarding stochastic variants of VRP has seen intensive development — reviews indicate numerous studies that address random demand and stochastic travel times [11], [12]. One example is the work by Pavone et al. [13], where a VRPTW model accounts for the random arrival of requests, with the time window starting depending on when the demand appears. However, in most such studies, time windows remain deterministic, even though in real-world systems they may shift dynamically.

In parallel, within the field of robust optimization, research has been conducted to safeguard the planning process against data uncertainty. While classical approaches, such as the RVRPTW model by Munari et al. [1], utilize uncertainty sets for demand and travel times, recent advancements have introduced Distributionally Robust Optimization (DRO) [14]. DRO approaches, as analyzed in data-driven contexts [2], aim to find solutions that remain feasible under the worst-case probability distribution within a specific ambiguity set. Similarly, Shen et al. [3] analyzed electric vehicle routing problems with uncertain demand and energy consumption. Although DRO and data-driven robust models provide strong theoretical guarantees, they often lead to overly conservative solutions and high computational complexity when applied to large-scale instances with complex time-window interdependencies.

Parallel to exact and metaheuristic methods, a growing body of research explores Artificial Intelligence (AI) and Machine Learning (ML) for VRP. Recent works utilize Deep Reinforcement Learning (DRL) and Transformer-based architectures [15], [16] to learn routing policies directly from data. While these AI-driven models offer rapid solution generation, they often struggle with the rigid feasibility requirements of time

windows and the high variance of stochastic environments without extensive retraining [17].

Despite the rich literature, significant research gaps remain. Many works treat random demand or uncertain travel times as stochastic variables, while few consider variability in time windows themselves. Bomboi et al. [12] showed that in VRPTW with random travel times, uncertainty strongly affects route feasibility. Errico et al. [18] analyzed stochastic service times but under deterministic time windows. An interesting extension was proposed by Spliet and Gabor [19], introducing the TWAVRP problem where time window boundaries are decisions made before demand realization. However, the literature still lacks works combining variable, random time windows with stochastic demand in the CVRPTW variant. Scenario-based approaches, where numerous random instances are generated and solved separately to assess stability and method behavior under uncertainty, are also rarely encountered. One of the newer directions includes works utilizing contextual data — for example, Serrano et al. [20] analyzed travel times dependent on environmental features.

The systematic review by Liu et al. [21] indicates that approximately 86% of studies on VRPTW employ approximate methods, mainly heuristics and metaheuristics, with the majority being tested only on deterministic data.

Furthermore, the literature increasingly recognizes simheuristics as a general methodology that extends metaheuristics through simulation to solve stochastic combinatorial optimization problems [22]. While deterministic models are often oversimplified versions of real-life systems, simulation-based stochastic analysis allows for a more granular assessment of risk and reliability criteria. This approach, frequently utilizing Monte Carlo simulation, is particularly effective for vehicle routing where the incorporation of stochastic information regarding demand or travel times significantly increases the reliability and service level of the resulting routing plans [23]. By evaluating promising solutions across various realizations, researchers can identify robust and resilient route configurations that maintain high performance across a wide spectrum of conditions, providing a practical alternative to overly rigorous theoretical models.

Despite the notable progress achieved in the literature, a clear research gap can still be identified in the following areas:

- Modeling random time window boundaries as variables drawn from probability distributions,
- Integrating the stochastic nature of time windows with random demand,
- Evaluating algorithm stability through large-scale empirical stress-tests rather than nominal average-case analysis,
- Bridging the gap between computationally expensive Distributionally Robust Optimization and the practical need for operationally stable metaheuristic solutions.

This work positions itself as an empirical alternative to formal robust optimization and AI-driven policies. By utilizing a scenario-based framework, it allows for a direct assessment of operational robustness across 1200 diverse realizations. This approach provides valuable insights into the stability of visit sequences and the mitigation of delay propagation, offering

a pragmatic foundation for handling uncertainty in real-world routing problems.

III. PROBLEM FORMULATION

This section presents the mathematical formulation of the Capacitated Vehicle Routing Problem with Time Windows (CVRPTW). The model is implemented as a two-index flow formulation, emphasizing operational time continuity and soft time windows with specific penalty factors.

A. Problem Definition and Notation

We define a set of customers $\mathcal{N} = \{1, 2, \dots, N\}$ and the depot as node 0. The complete set of locations is $\mathcal{D} = \{0\} \cup \mathcal{N}$. The movement between any two locations $i, j \in \mathcal{D}$ incurs a travel cost $d_{i,j}$. Each customer $i \in \mathcal{N}$ is characterized by a demand y_i , a service duration s_i , and a time window $[a_i, b_i]$.

The model assumes a fleet of V_{max} identical vehicles with capacity Q . Model follows a no-wait policy, where the departure from a location must occur immediately after the service is completed, and travel to the next node begins without delay.

TABLE I: Revised Notation Summary

Symbol	Meaning
$x_{i,j}$	Binary variable: 1 if an arc $i \rightarrow j$ is traversed
t_i	Continuous variable: arrival time at location i
u_i	Continuous variable: cumulative load after serving location i
e_i, l_i	Non-negative variables: early and late time window violations
$d_{i,j}, s_i$	Travel distance and service time at location i
$[a_i, b_i]$	Time window (start and end) for location i
Q, V_{max}	Vehicle capacity and maximum fleet size
C_{dist}	Factor for total travel distance
C_{time}	Factor for total operational time
C_{pen}	Global penalty factor for time violations
β_e, β_l	Specific weights for early (e_i) and late (l_i) arrivals

B. Mathematical Model

The mathematical model consists of an objective function that minimizes the total operational cost, subject to a set of constraints that ensure feasibility. The objective function comprises three main components: travel costs, operational time costs, and penalty costs for time window violations.

1) *Objective Function:* The objective function (1) minimizes a total cost:

$$\min_{x,e,l} Z = C_{dist} \cdot D + C_{time} \cdot T + C_{pen} \cdot P \quad (1)$$

where:

- $D = \sum_{i \in \mathcal{D}} \sum_{j \in \mathcal{D}} d_{i,j} x_{i,j}$ is the total distance.
- $T = D + \sum_{i \in \mathcal{N}} s_i$ is the total operational time.
- $P = \sum_{i \in \mathcal{N}} (\beta_e e_i + \beta_l l_i)$ is the total weighted penalty for time window violations.

2) *Constraints*: The objective function (1) is minimized subject to the following constraints:

Routing and Flow Constraints:

$$\sum_{j \in \mathcal{D}, j \neq i} x_{i,j} = 1, \quad \forall i \in \mathcal{N} \quad (2)$$

$$\sum_{i \in \mathcal{D}, i \neq j} x_{i,j} = 1, \quad \forall j \in \mathcal{N} \quad (3)$$

$$\sum_{j \in \mathcal{N}} x_{0,j} \leq V_{max} \quad (4)$$

$$\sum_{j \in \mathcal{N}} x_{0,j} = \sum_{j \in \mathcal{N}} x_{j,0} \quad (5)$$

Constraint (2) ensures that exactly one vehicle leaves each customer location, while constraint (3) guarantees that exactly one vehicle enters each location. Together, these two equations ensure that every customer in set $\mathcal{N}$ is visited exactly once.

Constraint (4) restricts the total number of active routes to the available fleet size V_{max} by limiting the number of vehicles exiting the depot. Finally, constraint (5) maintains the flow balance at the depot, ensuring that the number of vehicles returning to the warehouse is equal to the number of vehicles that originally departed.

Capacity and Subtour Elimination Constraints:

$$u_j \geq u_i + y_j - Q(1 - x_{i,j}), \quad \forall i \in \mathcal{N}, j \in \mathcal{N}, i \neq j \quad (6)$$

$$y_i \leq u_i \leq Q, \quad \forall i \in \mathcal{N} \quad (7)$$

Constraint (6) serves a dual purpose: it tracks the cumulative load of the vehicle and eliminates illegal subtours (cycles that do not include the depot). If the vehicle traverses the arc from location i to location j ($x_{i,j} = 1$), the equation forces the cumulative load at node j (u_j) to be at least the sum of the load already present at node i (u_i) and the demand of the new customer y_j . If the arc is not used ($x_{i,j} = 0$), the term $Q(1 - x_{i,j})$ makes the constraint redundant.

Constraint (7) defines the feasible domain for the cumulative load variables. It ensures that the load at any node i is at least equal to that customer's specific demand y_i and never exceeds the maximum vehicle capacity Q . This ensures that no vehicle is assigned more cargo than it can physically carry.

Temporal Continuity and No-Wait Constraints:

$$x_{i,j} = 1 \implies t_j = t_i + s_i + d_{i,j}, \quad \forall i \in \mathcal{N}, j \in \mathcal{N}, i \neq j \quad (8)$$

$$x_{0,j} = 1 \implies t_j = d_{0,j}, \quad \forall j \in \mathcal{N} \quad (9)$$

Constraint (8) establishes the timeline for the vehicle's route between any two customer locations. Unlike standard VRPTW formulations that use inequalities ($\geq$), this model employs a strict equality to enforce a no-wait policy. If an arc $i \rightarrow j$ is selected ($x_{i,j} = 1$), the arrival time at node j (t_j) is exactly the sum of the arrival time at the previous node i (t_i), the service duration s_i , and the travel time $d_{i,j}$. This implies that the vehicle must depart for the next destination immediately after completing service.

Constraint (9) defines the initial arrival time for the first customer in any route. If a vehicle travels directly from the depot to location j , the arrival time t_j is set equal to the travel

distance (or time) $d_{0,j}$, assuming the vehicle departs from the depot at $t_0 = 0$.

Soft Time Window Penalties:

$$e_i \geq a_i - t_i, \quad \forall i \in \mathcal{N} \quad (10)$$

$$l_i \geq (t_i + s_i) - b_i, \quad \forall i \in \mathcal{N} \quad (11)$$

Constraints (10) and (11) define the violation variables used to implement soft time windows. Unlike "hard" time windows that would strictly forbid any arrival outside the $[a_i, b_i]$ range, these constraints allow for deviations at a cost, which is determined by the penalty factors in the objective function.

Constraint (10) calculates the early arrival violation e_i . If the vehicle arrives at location i at time t_i before the window opens ($t_i < a_i$), e_i takes the value of the difference. Because e_i is defined as a non-negative variable, it remains zero if the vehicle arrives after a_i .

Constraint (11) measures the lateness violation l_i . In this specific model, a visit is considered late if the service completion time ($t_i + s_i$) exceeds the upper bound of the time window b_i . This reflects operational environments where the vehicle must not only arrive but also finish unloading within the customer's specified time frame. These penalty variables are then weighted by coefficients β_e and β_l respectively, allowing the model to prioritize the avoidance of late completions over early arrivals, or conversely.

This formulation provides a complete MILP model for the CVRPTW, solvable using commercial solvers or metaheuristic approaches for large-scale instances.

IV. EXPERIMENT

This section describes the experimental setting, including the dataset, scenario generation procedure, implementation details, and evaluation metrics. Then the experimental results for each hypothesis are presented.

A. Dataset and Implementation Description

All experiments were conducted on benchmark instances derived from the Solomon CVRPTW dataset. The test set includes representative instances from the three main Solomon families: clustered (C101, C201), random (R101, R201), and mixed (RC101, RC201), covering both short- and long-horizon variants. Each original Solomon instance was extended into a stochastic scenario set using a custom scenario generator. For every instance, 200 stochastic scenarios were created, resulting in a total of 1200 scenarios. The generator applies random perturbations to customer time windows and demands, modelling uncertain operational conditions such as delays, demand variability, or temporal disturbances.

The implementation was developed in C#, and the experimental setup consists of four methods: Gurobi (an exact solver for small-scale validation), and three heuristic approaches: a greedy baseline, Tabu Search (TS), and Artificial Bee Colony (ABC).

The Greedy Constructive Heuristic serves as the baseline, building routes iteratively by selecting the next customer based on a modified distance metric that incorporates estimated

penalties for potential time-window violations. A customer is added only if vehicle capacity and return-to-depot feasibility are maintained, otherwise, a new route is initiated.

The Tabu Search (TS) metaheuristic performs a local search starting from the greedy solution. It explores the neighborhood using operators such as *swap*, *insert*, and *invert*. To escape local optima and prevent cycling, recently visited solutions are stored in a Tabu List. The algorithm includes an aspiration criterion and a diversification mechanism that generates a new random solution when a state of stagnation is detected. It is controlled by the parameter *iteration_limit* - maximum number of iterations with no improvement).

The Artificial Bee Colony (ABC) algorithm is a population-based approach where the search is divided among employed, onlooker, and scout bees. Employed bees explore the neighborhood of existing food sources (solutions), while onlookers prioritize higher-quality solutions based on their fitness. Scout bees ensure exploration by abandoning exhausted food sources and replacing them with new random solutions. The initial population is seeded with the greedy solution and randomized configurations to ensure broad coverage of the search space.

The framework was designed for fully parallel execution. For every scenario, all methods were executed, and the system recorded objective values, total travel costs, time-window penalties, vehicle operation times, the number of routes, and the final route structure (GTR). All metrics were exported to a CSV file for comprehensive statistical analysis.

B. Computational Environment

All experiments were conducted on a Linux-based workstation equipped with a 13th Gen Intel Core i7-13700K processor and 32 GB of RAM. The system operated under Ubuntu 24.04.2 LTS, using the standard .NET runtime environment. Parallel execution was enabled through the Task Parallel Library, allowing scenario simulations to be distributed across the available CPU cores.

C. Parameter tuning

The parameter tuning for the Tabu Search (TS) and Artificial Bee Colony Optimization (ABC) algorithms was conducted on six representative Solomon instances (C101, C201, R101, R201, RC101, RC201). While the tuning process was executed for each instance individually, the results are presented in an aggregated format as the observed performance trends were highly consistent across the different files. The primary objective of this procedure was the identification of the optimal combination of parameters to minimize the objective function value across diverse stochastic scenarios.

For the Tabu Search metaheuristic, a total of 1620 experimental runs were performed. This covered 90 unique configurations, where the following parameters were varied:

- *Neighborhood operators*: *swap*, *insert*, and *invert*.
- *Tabu list size*: {10, 20, 30, 40, 50, 60}.
- *Iteration limit*: {50, 100, 150, 200, 250, 300}.

Each configuration was tested with three independent repetitions to ensure statistical robustness. Similarly, for the

Artificial Bee Colony algorithm, 1290 experimental runs were carried out across unique test settings. The tuning space for ABC included:

- *Neighborhood/Mutation operators*: *swap*, *2-opt*, *or-opt*, and *random*.
- *Population size* (Food sources): {50, 100, 200, 500}.
- *Stagnation limit*: {150, 300}.

These settings were evaluated across 5 random scenarios for each instance, with three repetitions per configuration. By systematically traversing these parameter sets, a balance between exploration of the solution space and computational efficiency was achieved, providing a foundation for the large-scale comparative analysis.

1) *ABC Parameter Analysis*: The experimental results for the ABC algorithm tuning are detailed in Table II. The data indicates that the selection of the neighborhood operator and number of food sources significantly influence the quality of the solutions obtained across the stochastic scenarios.

TABLE II: ABC Algorithm Tuning Results

Parameter	Value	Mean	Min	Max
Operator	2opt	71600	10908	253908
Operator	oropt	70152	9253	253656
Operator	rand	68288	8931	252686
Operator	swap	68284	8615	253501
Limit	150	69604	8760	253540
Limit	300	69558	8615	253908
FoodSources	50	67151	8615	247995
FoodSources	100	69623	9442	250646
FoodSources	200	71969	10014	253908

The *swap* operator achieved the lowest mean objective value (68284), followed closely by the randomized approach (*rand*) at 68288. Notably, in the worst-case scenarios (*Max*), the *rand* strategy outperformed all other operators. This suggests that a randomized selection of operators enhances diversification, when the search encounters stagnation, the invocation of more aggressive moves can facilitate the escape from local optima. Conversely, the primary disadvantage of the *rand* strategy is the risk of prematurely exiting a promising region of the search space when a milder operator might have yielded further refinement.

Regarding the *Limit* parameter, the performance variation between values of 150 and 300 was marginal, indicating that within this range, the parameter does not exert a decisive influence on the optimization outcome. In the analysis of population size (*FoodSources*), the smallest tested value of 50 proved to be the most effective. This efficiency is likely due to the fact that a smaller population allows the algorithm to execute more evolutionary cycles within the fixed computational time limit, resulting in more rapid convergence toward high-quality solutions.

Based on these findings, the configuration selected for the subsequent large-scale comparative analysis consists of the *swap* operator, a *Limit* of 300, and 50 *FoodSources*.

2) *TabuSearch Parameter Analysis*: The analysis of the parameter tuning results for the Tabu Search (TS) algorithm is summarized in Table III.

TABLE III: TabuSearch Algorithm Tuning Results

Parameter	Value	Mean	Min	Max
Operator	swap	70662	8131	255719
Operator	insert	72496	9324	256783
Operator	invert	74315	11524	259106
TabuSize	10	74267	8745	259106
TabuSize	20	73302	8366	256374
TabuSize	30	72044	8159	256869
TabuSize	40	71519	8200	254121
TabuSize	50	71249	8131	254245
TabuSize	60	72132	8131	254245
Iterations	50	72138	8567	256769
Iterations	100	72150	8159	257350
Iterations	150	72241	8131	257751
Iterations	200	73509	8336	258439
Iterations	250	72475	8555	259106
Iterations	300	72599	8370	257363

Regarding the neighborhood operators, the `swap` operator yielded the most favorable results across all metrics, significantly outperforming `insert` and `invert`. This suggests that for the addressed vehicle routing problem, interchanging two nodes within the route structure is the most effective mechanism for navigating toward high-quality regions of the solution space.

A clear performance trend is observed regarding the tabu list size (*TabuSize*), where increasing the size from 10 to 50 consistently reduced the mean objective value by preventing cycling and encouraging broader exploration of the search space. This expanded memory better equips the algorithm to escape local optima; however, increasing the *TabuSize* further to 60 resulted in a slight performance degradation. This suggests a threshold where an excessively restrictive list over-constrains the available move set, potentially hindering intensification within promising regions of the solution space.

In contrast, the results for the iteration limit (the number of non-improving moves before triggering diversification) remained relatively stable across the tested range. The mean values exhibited minimal variance between 50 and 300 iterations, suggesting that the algorithm either reaches a stable local optimum quickly or that the diversification mechanism effectively resets the search independently of the specific threshold value within this interval. However, the value of 150 iterations achieved the absolute minimum objective value across the tests.

Consequent to this analysis, the final experimental setup for the Tabu Search metaheuristic utilizes the `swap` operator, a *TabuSize* of 50, and an iteration limit of 150. These settings ensure a balance between intensive local exploitation and

the necessary exploration to maintain solution quality under stochastic conditions.

D. Experiment A: Metaheuristic Validation against Exact Methods (H1)

The first experiment verifies Hypothesis H1 by comparing the performance of Tabu Search (TS) and Bee Colony Optimization (ABC) against the Gurobi exact solver. To ensure a fair comparison of efficiency, the metaheuristic algorithms operated under a strict 30-second time limit as a stopping criterion, while the Gurobi solver was allowed up to 10 minutes (600 seconds) per instance to find an optimal solution. Due to the high computational complexity of exact methods, this experiment was conducted on a reduced dataset consisting of 120 scenarios with 13 customer locations each.

The results, as shown in Table IV, demonstrate the high efficiency of the implemented metaheuristics. The nature of the Solomon instances significantly influenced the solver's performance. Instances with narrow time windows proved to be computationally very expensive and sometimes reached a limit of 10 minutes without achieving full convergence. Out of the 120 analyzed scenarios, Gurobi failed to provide the optimal solution in 4 cases, where both Tabu Search and ABC managed to find lower cost solutions. Specifically, Tabu Search matched Gurobi's result in 116 scenarios and outperformed it in 4, while the ABC algorithm matched Gurobi in 113 cases and also outperformed it in 4.

Despite the significantly shorter execution time (a constant 30 seconds vs. up to 600 seconds for Gurobi), the metaheuristics consistently achieved results identical to or better than those provided by the exact solver. This parity in solution quality highlights a major advantage of metaheuristics. That is capability of finding optimal solutions in a fraction of the time required by exact methods. This is particularly evident in scenarios with tight constraints where Gurobi struggled to explore the full search space within the 10-minute limit. While the average time for Gurobi was approximately 381 seconds—with some simpler instances solved quickly and others hitting almost 600-second threshold. Both Tabu Search and Bee Colony Optimization reached the same or superior objective values in just 30 seconds. The observed stability and efficiency confirm that the implemented algorithms are highly effective in solving the addressed optimization problem, providing a reliable and significantly faster alternative for practical, large-scale routing in dynamic environments.

E. Experiment B: Metaheuristic Performance vs. Greedy Baseline (H2)

The second experiment verifies Hypothesis H2, which states that metaheuristic methods (Tabu Search and Bee Colony Optimization) achieve a lower average objective value than the greedy algorithm across a broad set of scenarios. The experiments were carried out on 1200 scenarios, and the summary of the relative improvements (Gaps) is presented in Table V. The metrics Gap_{min} and Gap_{max} are derived from the absolute extreme values recorded across the entire scenario

TABLE IV: Detailed performance comparison across instance classes (13 locations).

Instance Class	Gurobi (Mean)	Tabu Search (Mean)	ABC (Mean)
C101 (wide)	6147	6147	6147
C201 (narrow)	28465	28461	28461
R101 (wide)	1117	1117	1118
R201 (narrow)	5538	5525	5525
RC101 (wide)	1276	1276	1277
RC201 (narrow)	7183	7183	7183
Avg. Time [s]	381.50	30.00	30.00

set, representing the best-case and worst-case performance improvements, respectively. Similarly, the percentile gaps reflect the relative advantage maintained by metaheuristics at specific thresholds of the cost distribution.

A significant reduction in the objective function value is demonstrated by the metaheuristic approaches compared to the greedy baseline. As shown by the Gap_{avg} metric, the average operational cost is lowered by 23.74% by the ABC algorithm and by 15.50% by Tabu Search. This downward trend is consistently reflected across the entire performance distribution. In typical operational scenarios, represented by the median (p_{50}), a reduction of 38.72% is achieved by the ABC algorithm, while a 24.52% improvement is provided by Tabu Search.

The robustness of the metaheuristics is particularly evident in the analysis of the distribution tail, representing highly adverse conditions. For the 95th percentile (p_{95}), the cost threshold is lowered by 11.48% by the ABC method. Furthermore, the maximum operational risk (Gap_{max}) is reduced by 11.57% and 7.15% by the ABC and Tabu Search algorithms, respectively. Notably, the highest local performance gain is observed in the minimum cost values (Gap_{min}), where an improvement of 39.54% is reached by the Tabu Search metaheuristic, indicating its high efficiency in identifying high-quality solutions in favorable scenarios.

Overall, Hypothesis H2 is strongly confirmed. Significantly lower objective values are achieved by metaheuristic algorithms across the entire distribution of analyzed stochastic scenarios. The results suggest that the integration of ABC and TS provides a much more effective and reliable alternative to traditional constructive approaches.

F. Experiment C: Variability and Robustness Analysis (H3)

The third experiment investigates Hypothesis H3, which states that metaheuristics produce results with lower variability across stochastic scenarios than the greedy algorithm.

The primary indicator of operational robustness in this analysis is the dispersion of objective values, measured by the standard deviation across stochastic realizations. As presented in Table VI, a significant reduction in variability is achieved by metaheuristic approaches, particularly in clustered and wide-window instances. For the C101 instance, the standard deviation is reduced from 3505 for the greedy baseline to

TABLE V: Descriptive statistics of objective values for Greedy, Tabu Search, and Bee Colony Optimization.

Method	Mean	Gap_{avg} [%]	Gap_{min} [%]	Gap_{max} [%]
Greedy	81262	-	-	-
TabuSearch	68667	15.50	39.54	7.15
ABC	61974	23.74	36.02	11.57
Method	p_{50} [%]	p_{75} [%]	p_{90} [%]	p_{95} [%]
Greedy	-	-	-	-
TabuSearch	24.52	23.97	6.87	7.25
ABC	38.72	39.15	11.07	11.48

TABLE VI: Standard deviation of objective values across stochastic realizations for each benchmark instance.

Method	C101	C201	R101	R201	RC101	RC201
Greedy	3505	4486	523	935	509	1215
TabuSearch	1732	3783	284	1423	247	1252
ABC	1563	3790	150	1248	178	1288

1563 for the ABC algorithm, representing a 55.4% increase in predictability. Similarly, in the R101 scenario, the dispersion is lowered from 523 to 150 by the ABC method. These results indicate a significantly lower sensitivity to stochastic perturbations when metaheuristic search is employed.

The spatial distribution of customers and the nature of time constraints are observed to influence the stability of the generated solutions, as further illustrated by the Empirical Cumulative Distribution Function (ECDF) plots in Fig. 1. Analysis of these results reveals that in clustered instances (C101, C201), geographical groupings are effectively exploited. The ECDF curves for ABC and Tabu Search are positioned significantly to the left of the Greedy curve and exhibit much steeper slopes. This steepness directly corresponds to the lower standard deviation values reported in Table VI, confirming that metaheuristics consistently identify high-quality solutions within a narrow cost range.

While the advantage persists in most randomized and mixed instances, the search space complexity in highly constrained environments (e.g., R201, RC201) leads to a different dispersion pattern. In these cases, the metaheuristic algorithms explore a wider variety of feasible high-quality regions, which may result in a slightly higher standard deviation compared to the greedy baseline. However, as evidenced by the Gap_{avg} discussed in Experiment B, this minor increase in variability is accompanied by a substantial reduction in the absolute objective value.

As shown in all plots in Fig. 1, the ECDF curves for the metaheuristics (particularly ABC) rise much more rapidly than the Greedy baseline. This confirms that better solutions are obtained more frequently and with higher reliability. The flatter curve of the Greedy algorithm suggests a high sensitivity to randomness, where a single unfavorable random draw can lead to a disproportionately poor outcome.

The combined evidence from the per-instance dispersion metrics and ECDF analysis confirms Hypothesis H3: substantially more stable and predictable performance is provided by metaheuristic algorithms. They generate solutions with lower variability across the majority of problem categories, making them far better suited for real-world logistics where service reliability and adherence to strict time constraints are critical.

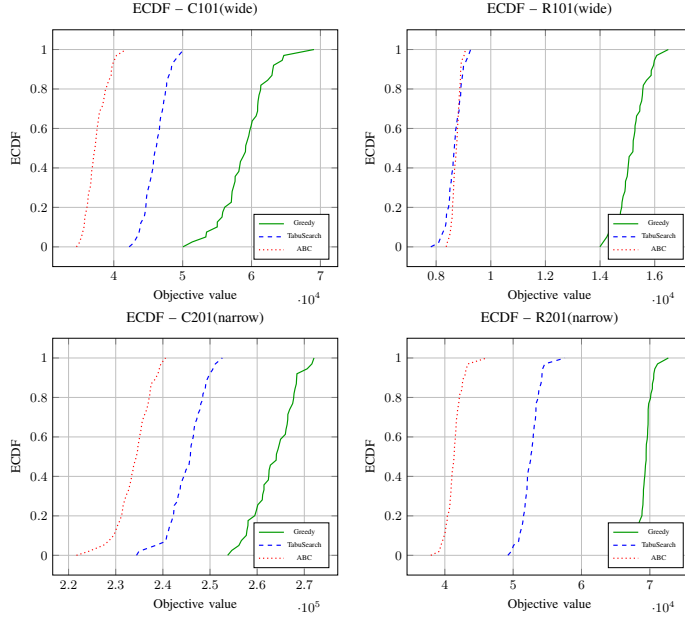


Fig. 1: ECDF comparison for four representative Solomon instances

G. Experiment D: Analysis of Time-Window Violation Penalties (H4)

The fourth experiment focuses on the penalty component of the objective function, which reflects violations of customer time windows under stochastic perturbations. In this setting, the objective function assigns a weight of 1 to travel distance, 1 to total operation time, and a weight of 2 to the total penalty associated with time-window violations. This formulation places a strong emphasis on limiting tardiness, prioritizing temporal feasibility over route length efficiency.

The results presented in Table VII indicate that metaheuristic algorithms produce significantly lower penalty values than the greedy baseline. Under the assumption of a continuous service model—where vehicles proceed directly between locations without waiting—penalties often arise from the cumulative propagation of delays across the route. However, both Tabu Search and the Artificial Bee Colony (ABC) algorithm consistently reduce the magnitude of these violations compared to the greedy approach.

This behaviour can be explained by the search dynamics of the methods. The greedy algorithm constructs routes sequentially and commits early to locally feasible insertions. In a "no-waiting" scenario, this lack of foresight leads to fragile sequences where a single late arrival triggers a domino effect of time-window violations for all subsequent customers. In

contrast, metaheuristics explore a broader neighbourhood of moves, allowing for the reordering of visits to find sequences that better absorb temporal perturbations. By optimizing the global structure of the route rather than making local, immediate choices, ABC and Tabu Search significantly mitigate delay propagation, reducing the average penalty.

Table VII summarizes the decomposition of the objective function. The results show that metaheuristics significantly reduce penalty values, decreasing from the greedy baseline of 72146 to 58949 for Tabu Search and 49548 for the ABC algorithm—a reduction of 18.3% and 31.3%, respectively. Notably, it is observed that the metaheuristic solutions result in higher values for total route cost and vehicle operation time compared to the greedy baseline. This increase in operational components is, however, significantly outweighed by the substantial reduction in penalties. Consequently, a superior overall performance is achieved, with the ABC algorithm reaching a total objective value of 61,974 compared to the baseline of 81,262. These results confirm that ABC and Tabu Search are better equipped to handle the temporal uncertainties of the CVRPTW by prioritizing the avoidance of time-window violations over the simple minimization of travel distance.

Therefore, Experiment D provides important evidence regarding Hypothesis H4: the metaheuristic approach outperforms the greedy baseline primarily through superior time-window compliance. Under conditions of uncertainty, where temporal feasibility is weighted, metaheuristics prove to be better equipped to handle delay propagation and stochastic perturbations typical of real-world CVRPTW instances.

TABLE VII: Comparison of average objective components for Greedy and Tabu Search

Method	RouteCost	Penalty	OperationTime	Objective
Greedy	2725	72146	6391	81262
TabuSearch	3026	58949	6692	68667
ABC	4380	49548	8046	61974

H. Experiment E: Tail Loss Reduction (H5)

The fifth experiment examines Hypothesis H5, which states that metaheuristic algorithms reduce the loss tail—defined as the frequency and severity of extreme-cost scenarios. The loss tail corresponds to the upper region of the cost distribution, where stochastic perturbations and accumulated delays produce exceptionally high objective values. Reducing this tail is crucial in real-world applications where worst-case performance determines operational feasibility.

The experimental results, summarized in Table V, indicate substantial improvements in the upper bounds of the cost distribution. A significant contraction of the loss tail is achieved by the metaheuristic approaches compared to the greedy baseline. Specifically, the maximum objective value (Gap_{max}) is reduced by 11.57% by the ABC algorithm and by 7.15% by TabuSearch. These figures demonstrate that metaheuristics generate far fewer scenarios with excessively

high costs, effectively mitigating the extreme outcomes that are most disruptive to logistics planning.

The mitigation of operational risk is further evidenced by the percentile gaps in the upper distribution. For the most demanding 10% (p_{90}) and 5% (p_{95}) of scenarios, the ABC algorithm maintains a consistent advantage of approximately 11.1–11.5% over the greedy approach. This indicates that even under extreme stochastic pressure, the metaheuristic framework prevents the "cost explosion" characteristic of constructive heuristics.

These differences are further reinforced by the Empirical Cumulative Distribution Function (ECDF) plots. The metaheuristic curves are observed to be significantly steeper and reach full cumulative coverage much earlier than the greedy baseline. While the greedy approach results in flatter curves that extend toward much higher costs, the rapid convergence of the ABC and Tabu Search curves indicates a lower sensitivity to unfavorable random realizations. For instance, the significant reduction in the objective value threshold at p_{95} directly corresponds to a much thinner tail and a lower probability of encountering a worst-case scenario.

Overall, the results strongly support Hypothesis H5. It is demonstrated that metaheuristics not only improve average performance but also significantly mitigate the risk of extreme unfavorable outcomes. By effectively managing the stochastic search space, these algorithms ensure predictable behavior even under difficult operating conditions, providing a more reliable foundation for robust transportation planning.

V. CONCLUSION

In this study, the effectiveness and reliability of metaheuristic algorithms were evaluated within the context of the stochastic Capacitated Vehicle Routing Problem with Time Windows (CVRPTW). Based on an extensive experimental framework including 1200 unique scenarios, it was demonstrated that metaheuristic approaches provide a significantly more consistent and high-quality alternative to traditional constructive methods.

The experimental results confirmed that the Artificial Bee Colony (ABC) algorithm achieved the highest performance, outperforming both Tabu Search and the greedy baseline across all evaluated metrics. A mean cost reduction (Gap_{avg}) of 23.74% was reached by the ABC algorithm, compared to 15.50% for Tabu Search. A critical finding of this research is the mitigation of operational risks associated with extreme-cost outcomes, as evidenced by a 11.57% reduction in the maximum objective value (Gap_{max}) and the stabilization of costs in the upper percentiles (p_{95}). It was observed that while metaheuristics may generate solutions with slightly higher total route costs and vehicle operation times, this increase is effectively compensated by a substantial decrease in time-window penalties, which were reduced by up to 31.3%. Furthermore, the quality of the metaheuristic solutions for smaller instances was validated through a comparison with the Gurobi exact solver, providing empirical evidence of the high efficiency of the implemented algorithms.

A. Practical Implications

The findings of this study offer key insights for logistics management and the design of automated routing systems. The application of metaheuristic optimization yields a huge advantage over traditional constructive methods in real-world transportation environments characterized by uncertainty.

- **Enhancing Service Reliability:** The significant reduction in the objective function's upper percentiles (p_{95}) demonstrates that metaheuristic algorithms provide a much higher degree of predictability. For logistics providers, this translates to a lower frequency of significant delays, thereby increasing customer satisfaction and strengthening service level agreement (SLA) compliance.
- **Strategic Trade-off Management:** A critical practical observation is the identification of the trade-off between route length and penalty avoidance. It is demonstrated that by accepting slightly higher vehicle operation times, a disproportionately large reduction in time-window penalties can be achieved. This insight allows decision-makers to justify more conservative, stability-oriented routing strategies that prioritize timely arrivals over the simple minimization of travel distance.
- **Operational Risk Mitigation:** The Gap_{max} metric indicates that metaheuristic approaches effectively mitigate the impact of extreme stochastic disruptions. By reducing the maximum observed costs by over 11%, these algorithms ensure that operational expenditures remain within stable bounds even under adverse conditions. This approach prevents the significant cost increase typically inherent in greedy dispatching methods.
- **Decision Support and Risk Assessment:** Scenario-based evaluation provides a solid foundation for operational planning. It allows logistics managers to analyze how delivery networks perform under various uncertain conditions, making it possible to identify vulnerabilities and ensure stability before problems arise in actual operations.

B. Research Roadmap and Future Directions

The current work establishes a necessary foundation for the development of adaptive, intelligent logistics systems. To address the evolving research frontier in optimization, the following strategic roadmap is proposed:

- **Short-term: Algorithmic Expansion and Constraint Enrichment.** In the immediate term, the research scope is to be expanded by incorporating additional metaheuristics, such as Simulated Annealing or Genetic Algorithms, to further evaluate the performance of diverse search strategies. Furthermore, the problem formulation will be extended to include more complex constraints, such as heterogeneous fleets and multi-depot configurations, ensuring the model more closely mirrors the intricacies of real-world logistical challenges.
- **Mid-term: Intelligent Neighborhood Control by Reinforcement Learning (RL).** The integration of Reinforcement Learning is envisioned for the dynamic selection of neighborhood operators during the optimization process. It is anticipated that the intelligent control of

the neighborhood generation process will allow for the strategic deployment of aggressive operators, such as `invert`, when the search encounters stagnation. Conversely, milder operators, such as `swap` or `insert`, will be prioritized when the search traverses promising regions of the solution space, ensuring a more effective balance between exploration and exploitation.

- **Mid-to-Long term: Data-Driven Parameter Optimization.** Bridging the gap between mid- and long-term objectives, a data-driven approach will be implemented. This involves the development of a neural network architecture designed to analyze instance-specific features to automatically select the best parameters for the optimization algorithms. By tailoring parameters to the unique characteristics of each instance, an improvement in both the stability and efficiency of the search process is expected.
- **Long-term: Neural Combinatorial Optimization.** The final stage of the roadmap is directed toward Neural Combinatorial Optimization, where deep neural networks are developed to generate solutions directly as end-to-end solvers. The high-quality results and the extensive scenario-based dataset established in this paper will serve as the base for training these models. This transition will enable the instantaneous generation of feasible routes in highly volatile environments where traditional iterative optimization may prove computationally prohibitive.

In summary, the implemented algorithms have proven their effectiveness in managing uncertainty within logistics. The transition from static, greedy-based planning to reliable, learning-augmented metaheuristics represents the most promising pathway toward autonomous and resilient transportation systems.

ACKNOWLEDGMENT

The authors would like to thank PhD Radosław Idzikowski and M.Sc. Eng. Michał Jaroszczuk for their appropriate and constructive suggestions to improve this article.

REFERENCES

- [1] P. Munari, A. Moreno, J. De La Vega, D. Alem, J. Gondzio, and R. Morabito, "The robust vehicle routing problem with time windows: compact formulation and branch-price-and-cut method," *Transportation Science*, vol. 53, no. 4, 02 2018.
- [2] Y. Zhang, Z. Zhang, A. Lim, and M. Sim, "Robust data-driven vehicle routing with time windows," *Operations Research*, vol. 69, pp. 469–485, 03 2021.
- [3] Y. Shen, L. Yu, and J. Li, "Robust electric vehicle routing problem with time windows under demand uncertainty and weight-related energy consumption," *Complex System Modeling and Simulation*, vol. 2, no. 1, pp. 18–34, 2022.
- [4] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management Science*, vol. 6, no. 1, pp. 80–91, 1959.
- [5] R. Idzikowski, J. Rudy, and M. Jaroszczuk, "Solving a vehicle routing problem for a real-life parcel locker-based delivery," in *System Dependability - Theory and Applications*, W. Zamojski, J. Mazurkiewicz, J. Sugier, T. Walkowiak, and J. Kacprzyk, Eds. Cham: Springer Nature Switzerland, 2024, pp. 69–79.
- [6] —, "Multi-criteria vehicle routing problem for a real-life parcel locker-based delivery," *International Journal of Electronics and Telecommunications*, vol. vol. 71, pp. 1–1, 07 2025.
- [7] P. Toth and D. Vigo, *Vehicle Routing: Problems, Methods, and Applications*, 2nd ed. Philadelphia, PA: SIAM, 2014.
- [8] B. L. Golden, S. Raghavan, and E. A. Wasil, *The Vehicle Routing Problem: Latest Advances and New Challenges*, ser. Operations Research/Computer Science Interfaces Series. New York: Springer, 2008, vol. 43.
- [9] K. Braekers, K. Ramaekers, and I. Nieuwenhuys, "The vehicle routing problem: State of the art classification and review," *Computers & Industrial Engineering*, vol. 99, 12 2015.
- [10] A. Yernar and C. Turan, "Recent developments in vehicle routing problem under time uncertainty: a comprehensive review," *Bulletin of Electrical Engineering and Informatics*, vol. 14, pp. 1263–1275, 04 2025.
- [11] J. Oyola, H. Arntzen, and D. L. Woodruff, "The stochastic vehicle routing problem, a literature review, part i: models," *EURO Journal on Transportation and Logistics*, vol. 7, no. 3, pp. 193–221, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2192437620300649>
- [12] F. Bomboi, C. Buchheim, and J. Puente, "On the stochastic vehicle routing problem with time windows, correlated travel times, and time dependency," *4OR: A Quarterly Journal of Operations Research*, vol. 20, 2022.
- [13] M. Pavone, N. Bisnik, E. Frazzoli, and V. Isler, "A stochastic and dynamic vehicle routing problem with time windows and customer impatience," *MONET*, vol. 14, pp. 350–364, 06 2009.
- [14] H. Rahimian and S. Mehrotra, "Distributionally robust optimization: A review," 08 2019.
- [15] W. Kool, H. van Hoof, and M. Welling, "Attention, learn to solve routing problems!" 2019. [Online]. Available: <https://arxiv.org/abs/1803.08475>
- [16] J. Li, Y. Ma, R. Gao, Z. Cao, A. Lim, W. Song, and J. Zhang, "Deep reinforcement learning for solving the heterogeneous capacitated vehicle routing problem," *IEEE Transactions on Cybernetics*, vol. 52, no. 12, pp. 13 572–13 585, 2022.
- [17] I. Bello, H. Pham, Q. V. Le, M. Norouzi, and S. Bengio, "Neural combinatorial optimization with reinforcement learning," 2017. [Online]. Available: <https://arxiv.org/abs/1611.09940>
- [18] F. Errico, G. Desaulniers, M. Gendreau, W. Rei, and L.-M. Rousseau, "The vehicle routing problem with hard time windows and stochastic service times," *Transportation Science*, vol. 50, no. 2, pp. 607–626, 2016.
- [19] R. Spliet and A. F. Gabor, "The time window assignment vehicle routing problem," *Transportation Science*, vol. 48, no. 3, pp. 435–455, 2014.
- [20] B. Serrano, A. M. Florio, S. Minner, M. Schiffer, and T. Vidal, "Contextual stochastic vehicle routing with time windows," 2024. [Online]. Available: <https://arxiv.org/abs/2402.06968>
- [21] X. Liu, Y.-L. Chen, L. Y. Por, and C. S. Ku, "A systematic literature review of vehicle routing problems with time windows," *Sustainability*, vol. 15, no. 15, 2023. [Online]. Available: <https://www.mdpi.com/2071-1050/15/15/12004>
- [22] A. A. Juan, J. Faulin, S. E. Grasman, M. Rabe, and G. Figueira, "A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems," *Operations Research Perspectives*, vol. 2, pp. 62–72, 2015. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S221471601500007X>
- [23] U. Ritzinger, J. Puchinger, and R. Hartl, "A survey on dynamic and stochastic vehicle routing problems," *International Journal of Production Research*, vol. 54, pp. 215–231, 01 2016.