

Article

Not peer-reviewed version

A Branch-and-Price Approach to the Platform Supply Vessel Routing and Scheduling Problem with Uncertain Demand

[Bin Ji](#), Jing Liu, [Samson S. Yu](#)*

Posted Date: 14 April 2026

doi: 10.20944/preprints202604.0863.v1

Keywords: branch-and-price; platform supply vessels; routing and scheduling; uncertain demand



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

A Branch-and-Price Approach to the Platform Supply Vessel Routing and Scheduling Problem with Uncertain Demand

Bin Ji ¹, Jing Liu ¹ and Samson S. Yu ^{2,*}

¹ School of Traffic & Transportation Engineering, Central South University, Changsha 410075, China

² School of Engineering, Deakin University, Melbourne, 3125, Australia

* Correspondence: samson.yu@deakin.edu.au

Abstract

With the expansion of offshore oil and gas exploration into deep-water regions, the efficient scheduling of Platform Supply Vessels (PSVs) is critical to offshore operations. The Platform Supply Vessel Routing and Scheduling Problem (PSVRSP) is an NP-hard combinatorial optimization problem, which is further complicated by uncertainty in offshore demand. Existing studies reveal a methodological gap: heuristic approaches cannot guarantee optimality, while exact algorithms often ignore demand uncertainty. To address this gap, this study proposes a Branch-and-Price (B&P) method for the Platform Supply Vessel Routing and Scheduling Problem with Uncertain Demand (PSVRSP-UD). A scenario-based Mixed-Integer Linear Programming (MILP) model is formulated, in which demand uncertainty is captured using Latin Hypercube Sampling (LHS) combined with Cholesky Decomposition and Sample-Based Reduction (SBR). Based on Dantzig–Wolfe Decomposition, the proposed B&P algorithm integrates NG-Route labeling and a two-level branching strategy to achieve global optimization. Computational experiments show that the B&P algorithm outperforms CPLEX in both computational efficiency and solution quality. Sensitivity analyses examine the impacts of scenario number, demand fluctuation, and weight coefficients on the results. The new results in this study can provide a practical decision-support tool for offshore logistics operations.

Keywords: branch-and-price; platform supply vessels; routing and scheduling; uncertain demand

MSC: 90-10

1. Introduction

Against the backdrop of global oil and gas exploration and development expanding into deep and ultra-deep waters, supply vessels serve as the core logistics link connecting onshore supply bases with offshore facilities such as drilling platforms and production platforms [1]. Their routing and scheduling optimization directly determines the continuity and economic efficiency of offshore oil and gas operations. Given the limited space on offshore installations, regular deliveries of equipment and materials, as well as the removal of waste materials, rely heavily on Platform Supply Vessel (PSV) services. As one of the costliest resources in offshore logistics, PSVs require high operational efficiency and rational scheduling for effective cost control in oil and gas companies. Therefore, the Platform Supply Vessel Routing and Scheduling Problem (PSVRSP) has become a central topic in upstream supply chain optimization for the offshore industry [2].

The PSVRSP is essentially a complex combinatorial optimization problem, whose difficulty arises from the interplay of multi-dimensional constraints and various sources of uncertainty. In

terms of constraints, scheduling must comply with visit-frequency requirements of offshore platforms, PSV capacity limits, onshore base service capacity (maximum daily departures), PSV-specific characteristics (power, speed, cargo-capacity heterogeneity), and time-window restrictions related to operational windows and voyage duration limits. In terms of uncertainty, volatile demand from offshore platforms and disruptions caused by adverse weather conditions often invalidate pre-planned schedules, leading to extra costs such as vessel chartering and route rerouting. Moreover, the PSVRSP is inherently NP-hard. As the number of offshore facilities increases, the solution space expands exponentially, rendering traditional experience-based scheduling and basic optimization methods inadequate for real-world applications.

Existing studies have achieved considerable progress in the optimization of PSVRSP. Early research mainly focused on deterministic environments, where scheduling models were typically formulated using Mixed-Integer Linear Programming (MILP) and solved with heuristic algorithms such as genetic algorithms and tabu search, providing preliminary optimization for fleet composition and route planning. With a deeper understanding of practical operational scenarios, the impacts of demand uncertainty have attracted increasing attention. Researchers have adopted approaches including two-stage stochastic programming, robust optimization, and simulation–optimization embedded with metaheuristics to incorporate demand volatility, thereby improving the practicality and robustness of scheduling solutions. Nevertheless, the application of exact solution methods in PSVRSP remains relatively limited. A notable contribution was made by Friedberg [3], who introduced a Branch-and-Price (B&P) algorithm for an MILP model of PSVRSP, in which supply vessels operated in coordination with bunkering vessels as offshore hubs. Their work validated the effectiveness of decomposition-based exact methods for PSV routing problems. However, their model was developed under a deterministic setting and did not account for demand uncertainty. To date, few exact algorithms have been proposed for PSVRSP under stochastic demand, leaving a clear methodological gap in the literature.

Despite the advances covering both deterministic and stochastic settings as well as various heuristic approaches, critical research gaps still exist. Most studies addressing demand uncertainty rely on heuristics or metaheuristics, which cannot guarantee solution optimality. Meanwhile, the few studies that employ exact algorithms ignore demand variability and thus deviate from real-world operational conditions. Therefore, developing an exact solution framework that integrates demand uncertainty and enables rigorous global optimization is of great theoretical significance and practical value for enhancing the reliability and cost-effectiveness of offshore logistics. Building on the existing literature, this study focuses on the complexity and practical requirements of the PSVRSP. By incorporating multiple operational constraints and demand uncertainty, this paper aims to develop more efficient exact optimization approaches, so as to provide scientific and practical decision support for offshore oil and gas companies and facilitate the refined management and sustainable development of offshore logistics.

2. Literature Review

This section provides a systematic review of the literature related to the Platform Supply Vessel Routing and Scheduling Problem (PSVRSP) in both deterministic and uncertain environments.

Early research on PSVRSP can be traced back to Fagerholt [4], who first established a Mixed-Integer Linear Programming (MILP) model for the periodic routing and scheduling of heterogeneous platform supply vessels, considering pickup and delivery tasks for deck cargo. This work laid the foundation for subsequent modeling of supply vessel operations. Aas et al. [5] further expanded this model by incorporating constraints related to deck-space availability on offshore platforms. For large-scale instances, Gribkovskaia et al. [6] designed a tabu search heuristic to improve computational efficiency. Shyshou et al. [7] focused on simplified versions of the problem and developed customized heuristics. Halvorsen-Weare et al. [8] formulated an MILP model that limited voyage duration and controlled departure times from the supply base to improve schedule feasibility. Although the existing literature has increasingly addressed multi-commodity, multi-vessel, and complex

operational settings, most early studies relied on heuristic algorithms to obtain feasible solutions. Friedberg et al. [3] introduced an important advancement by applying a Branch-and-Price (B&P) algorithm to an MILP model for platform supply vessel routing and scheduling, in which supply vessels operated alongside refueling vessels as offshore hubs. This study demonstrated the strength of decomposition techniques for platform supply vessel routing problems.

Research on the Supply Vessel Planning Problem (SVPP) under uncertainty includes papers addressing uncertain weather conditions, uncertain demand, and the combination of both uncertain demand and uncertain weather conditions. Most studies adopt a two-stage framework that integrates simulation and optimization and can be divided into two categories. The first category constructs robust schedules using deterministic models enhanced by either probabilistic constraints to accommodate uncertain demand [9] or time slack to absorb delays caused by rough weather [8]. The second category estimates the average undelivered demand via simulation and then incorporates penalty costs into a deterministic optimization model [10]. Following a similar philosophy, Cruz et al. [11] generate candidate routes under stochastic demand and travel times and then use reliability levels to build robust weekly schedules. Meanwhile, Ksciuk et al. [12] provides a comprehensive review of uncertainty handling in maritime routing, including supply vessel scheduling.

Adverse weather is widely recognized as the primary cause of schedule failures [13]. To improve robustness, Halvorsen-Weare et al. [8] propose a set-covering model with mandatory time slack between consecutive voyages. However, they do not verify actual robustness through simulation. Halvorsen-Weare et al. [10] later extend this work by using discrete-event simulation to compute expected unmet demand and assign penalties accordingly. For uncertain demand, common strategies include enforcing voyage-level chance constraints to guarantee service reliability or inflating demand forecasts to accommodate variability. Notably, Halvorsen-Weare et al. combine demand inflation and time buffers to address both demand and weather uncertainty simultaneously.

Nevertheless, these approaches have notable drawbacks. Most of these methods do not aim to minimize the actual expected total cost, which can result in economically inefficient schedules. By comparison, Monte Carlo Sampling has proven effective for discrete stochastic optimization problems where full scenario enumeration is intractable, including stochastic vehicle routing and supply chain design [14]. It is also widely used to evaluate solutions within heuristic frameworks to handle both NP-hardness and scenario intractability [15].

Despite these advances, most studies addressing the stochastic Supply Vessel Routing and Scheduling Problem (SVRSP) rely on heuristics or simulation-based methods. Few employ exact solution methods such as Branch-and-Price, and even fewer integrate such methods with Monte Carlo sampling to address stochastic demand while minimizing expected total cost under time-window constraints. To fill this gap, this study proposes a Branch-and-Price algorithm embedded with Monte Carlo sampling to solve the SVRSP with uncertain demand and time windows, with the objective of minimizing the expected total cost.

3. Problem Description

A PSVRSP typically involves several core components, including an onshore supply base, a set of offshore platforms that generate orders, and a fleet of Platform Supply Vessels (PSVs), which must fulfill all service orders. In practical offshore logistics operations, each platform may issue multiple orders, and each order exhibits diverse loading and unloading requirements as well as time windows during which vessels must initiate service.

In practical offshore logistics operations, the PSVRSP can be described on a graph $\mathcal{G}(\mathcal{V}, \mathcal{A})$, where the node set $\mathcal{V}$ represents order locations, and the arc set $\mathcal{A}$ represents sailing legs, each associated with the sailing distance between pairs of orders. Let $\mathcal{C}$ denote the set of offshore platforms that require service, and $\mathcal{O}$ the set of all orders to be performed. Let r be the unit cargo-handling rate at each offshore platform, and $\mathcal{K}$ the set of available PSVs. In this work, a trip is defined as a complete voyage cycle performed by a single vessel, consisting of three sequential tasks:

(i) departing empty from the onshore supply base; (ii) serving orders offshore; and (iii) returning to the onshore supply base to unload any pickup cargo collected during the trip.

Each vessel $k \in \mathcal{K}$ is allowed to complete at most L consecutive trip. The entire sequence of trips performed by a vessel is collectively referred to as a route. Figure 1 illustrates the structure of the transportation network.

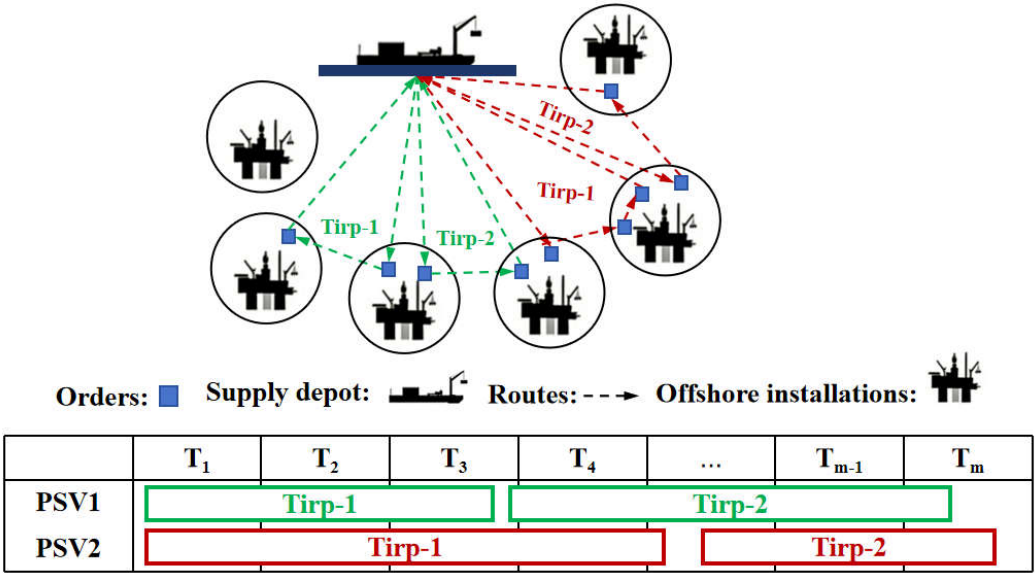


Figure 1. Illustrative diagram of a PSVRSP.

3.1. Demand Uncertainty Modeling and Scenario Generation

In this section, we present a method to approximate the probability distribution of demand at offshore installations. We assume that demands at different installations are mutually independent; that is, demand at each installation follows an independent probability distribution, which may include normal, Erlang, or Weibull distributions [9]. Based on such probabilistic representations, several modeling approaches can be employed, including robust optimization, chance-constrained programming, and scenario-based stochastic programming. Robust and chance-constrained models provide protection against uncertainty but may lead to overly conservative decisions or complex probabilistic constraints [16]. By comparison, scenario-based stochastic programming discretizes the underlying demand distribution into a finite set of representative realizations, which supports direct comparisons across different demand outcomes. Given this balance between modeling fidelity and computational tractability, this study adopts scenario-based representations. To incorporate this uncertainty into the model, demand is treated as a random variable, and a combination of scenario generation and scenario reduction approach is employed.

To generate a diverse set of demand scenarios, this study adopts an approach that combines LHS and Cholesky decomposition [17]. LHS provides stratified sampling that ensures more uniform coverage of multidimensional distribution space. However, when sampling multiple random variables, their inherent correlation may reduce representativeness of the generated scenarios. To address this issue, Cholesky decomposition is incorporated to weaken inter-order demand correlations, thereby improving independence and diversity of the scenario set.

Suppose there are $|\mathcal{N}_O|$ orders, and their demands are represented by $X_1, \dots, X_{|\mathcal{N}_O|}$. The cumulative distribution function for the demand at order n is expressed as follows,

$$P_n = F_n(X_n). \tag{1}$$

Using this formulation, a total of M scenarios is generated through the process. First, an ordered sampling matrix is constructed via the LHS method. Next, the Cholesky decomposition is applied to adjust the sampled vectors and mitigate correlation among port demands. The final output is an M -scenario demand matrix for the $|\mathcal{N}_O|$ orders, completing the multi-scenario generation procedure.

To improve computational efficiency and reduce redundant scenarios, the stepwise backward reduction (SBR) algorithm [18] is applied. This algorithm iteratively computes the distances between scenarios and removes the scenario with the smallest distance to others until the initial scenarios are reduced to the predefined number. The final output consists of representative scenarios and their associated probabilities. Specifically, $\mathcal{S}$ denotes the scenario set $\mathcal{S} = \{1, 2, \dots, |\mathcal{S}|\}$, and each scenario $s \in \mathcal{S}$ occurs with probability p_s .

3.2. Mathematical Modeling

In this section, we present an MILP formulation for the PSVRSP-UD. Before we proceed, we first introduce some pertinent notation. Table 1 lists the sets, parameters, and variables involved in the model.

In this study, duplicated supply base nodes are used to model the multi-trip scheduling problem for a single vessel [19]. By constructing an ordered chain of depot-copy nodes to characterize the trip sequence, cross-trip service is avoided without introducing additional trip-index variables. To ensure the sequential use of duplicated supply base nodes, the arc set $\mathcal{A}$ is defined as $\mathcal{A} = \mathcal{A}_{\mathcal{O}\mathcal{O}} \cup \mathcal{A}_{\mathcal{D}\mathcal{O}} \cup \mathcal{A}_{\mathcal{O}\mathcal{D}} \cup \mathcal{A}_{\mathcal{D}\mathcal{D}}$, where $\mathcal{A}_{\mathcal{O}\mathcal{O}} = \{(i, j): i, j \in \mathcal{O}, i \neq j\}$ represents arc set between customer nodes, $\mathcal{A}_{\mathcal{D}\mathcal{O}} = \{(d, j): d \in \mathcal{D}, j \in \mathcal{O}\}$ represents arc set from duplicated supply base nodes to customer nodes, $\mathcal{A}_{\mathcal{O}\mathcal{D}} = \{(i, d): d \in \mathcal{D}, i \in \mathcal{O}\}$ represents arc set from customer nodes to duplicated supply base nodes, $\mathcal{A}_{\mathcal{D}\mathcal{D}} = \{(0^l, 0^{l+1}): l = 1, 2, \dots, L\}$ represents sequential arc set between consecutive duplicated supply base nodes. This definition ensures that supply vessels can only visit duplicated supply base nodes in strict trip sequence within the planning horizon, and any non-sequential jump across trips is prohibited.

Table 1. Notation of sets, parameters and variables.

Tag	Description
Set	
$\mathcal{O}$	Set of orders, $\mathcal{O} = \{1, 2, \dots, \mathcal{O} \}$
$\mathcal{C}$	Set of offshore platforms that request pickup and/or delivery orders, $\mathcal{C} = \{1, 2, \dots, \mathcal{C} \}$
$\mathcal{D}$	Set of duplicated supply base nodes, $\mathcal{D} = \{0^1, 0^2, \dots, 0^{L+1}\}$, where 0^1 denotes the departure base for the first trip and 0^{L+1} denotes the return base for the last trip
$\mathcal{V}$	Set of all nodes, $\mathcal{V} = \mathcal{O} \cup \mathcal{D}$
$\mathcal{A}$	Set of feasible arcs
$\mathcal{K}$	Set of platform supply vessels, $\mathcal{K} = \{1, 2, \dots, \mathcal{K} \}$
$\mathcal{S}$	Set of demand scenarios, $\mathcal{S} = \{1, 2, \dots, \mathcal{S} \}$
Parameter	
p_s	Probability of scenario s
q_i^s	Cargo quantity required for order i at platform c^i under scenario s
r	Unit cargo handling rate
C_t	Operating cost per unit time of a vessel
C_f	Fuel price per unit time
B_k	Light weight of vessel k
E	Daily fuel consumption of a vessel sailing at maximum speed under full load
c^i	Platform where order i is located
d_{ij}	Sailing distance between platform c^i and c^j , with $d_{ij} = 0$ if $c^i = c^j$
τ_{ij}	Sailing time from c^i to c^j
s_i	Service time of order i
$[a_i, b_i]$	Time window for service of order i

v	Sailing speed of platform supply vessels
Q	Capacity of platform supply vessel k
T	Planning horizon, typically one week
L	Maximum number of trips allowed for a single vessel within the planning horizon

Variable

l_i^{ks}	Continuous variable representing the load of vessel k departing from platforms c^i of order i under scenario s
t_i^k	Continuous variable representing the leave time of order i by vessel k
x_{ij}^k	Binary variable equal to 1 if vessel k traverses arc (i, j) and 0 otherwise

The PSVRSP-UD model aims to generate feasible routes and schedules for platform supply vessels (PSVs) to minimize the total cost while satisfying all order fulfillment and operational constraints. As PSVs represent one of the costliest resources in offshore logistics, and in response to the requirements of energy conservation and emission reduction in international shipping, the total cost consists of vessel operating cost and carbon emission cost, where α is a weight coefficient that balances the importance of the two cost components. The carbon emission cost adopts the fuel consumption function proposed by Psaraftis and Kontovas [20], which comprehensively considers the nonlinear relationship among demand scenario probability, vessel speed, payload, and fuel consumption. The complete MIP model of PSVRSP-UD is as follows.

$$\text{Min } \alpha \sum_{k \in \mathcal{K}} C_t(t_{0^{L+1}}^k - t_0^k) + (1 - \alpha) \sum_{s \in \mathcal{S}} p_s \sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{A}} [C_f \cdot E_k \cdot v^3 \cdot (B_k + l_{ik}^s)^{\frac{2}{3}} \cdot \tau_{ij} \cdot x_{ij}^k], \alpha \in [0,1], \quad (2)$$

Subject to

$$\sum_{(0^1, j) \in \mathcal{A}} x_{0^1 j}^k \leq 1 \quad \forall k \in \mathcal{K}, \quad (3)$$

$$\sum_{(i, 0^{L+1}) \in \mathcal{A}} x_{i 0^{L+1}}^k \leq 1 \quad \forall k \in \mathcal{K}, \quad (4)$$

$$\sum_{(0^l, j) \in \mathcal{A}} x_{0^l j}^k = \sum_{(i, 0^l) \in \mathcal{A}} x_{i 0^l}^k \quad \forall k \in \mathcal{K}, l = 2, \dots, L, \quad (5)$$

$$\sum_{(i, j) \in \mathcal{A}} x_{ij}^k = \sum_{(j, i) \in \mathcal{A}} x_{ji}^k \quad \forall k \in \mathcal{K}, i \in \mathcal{O}, \quad (6)$$

$$\sum_{k \in \mathcal{K}} \sum_{(i, j) \in \mathcal{A}} x_{ij}^k = 1 \quad \forall k \in \mathcal{K}, j \in \mathcal{O}, \quad (7)$$

$$\sum_{(i, j) \in \mathcal{A}} x_{ij}^k \leq \sum_{(j, i) \in \mathcal{A}} x_{ji}^k \quad \forall k \in \mathcal{K}, i \in \mathcal{D} / \{0^1, 0^{L+1}\}, \quad (8)$$

$$l_i^{ks} \leq Q \quad \forall k \in \mathcal{K}, i \in \mathcal{V}, s \in \mathcal{S}, \quad (9)$$

$$l_i^{ks} + q_j^s \leq l_j^{ks} + M(1 - x_{ij}^k) \quad \forall (i, j) \in \mathcal{A}, k \in \mathcal{K}, s \in \mathcal{S}, \quad (10)$$

$$l_{0^l}^{ks} = 0, \forall k \in \mathcal{K}, s \in \mathcal{S}, l = 2, \dots, L, \quad (11)$$

$$\tau_{ij} = \frac{d_{ij}}{v} \quad \forall i, j \in \mathcal{V}, \quad (12)$$

$$s_i = \frac{q_i^s}{r} \forall i \in \mathcal{O}, s \in \mathcal{S}, \quad (13)$$

$$a_i - M \left(1 - \sum_{(i,j) \in \mathcal{A}} x_{ij}^k \right) \leq t_i^k \forall i \in \mathcal{V}, k \in \mathcal{K}, \quad (14)$$

$$t_i^k \leq b_i + M \left(1 - \sum_{(i,j) \in \mathcal{A}} x_{ij}^k \right) \forall i \in \mathcal{V}, k \in \mathcal{K}, \quad (15)$$

$$t_i^k + \tau_{ij} + s_j \leq t_j^k + M(1 - x_{ij}^k) \forall (i,j) \in \mathcal{A}, k \in \mathcal{K}, \quad (16)$$

$$x_{ij}^k \in \{0,1\} \forall i,j \in \mathcal{V}, k \in \mathcal{K}, \quad (17)$$

$$l_i^{ks} \geq 0 \forall i \in \mathcal{V}, k \in \mathcal{K}, s \in \mathcal{S}, \quad (18)$$

$$T \geq t_i^k \geq 0 \forall i \in \mathcal{V}, k \in \mathcal{K}, \quad (19)$$

The objective function (2) is to minimize the expected value of the total cost over all demand scenarios. Constraint (3) is the out-degree constraint for the starting duplicated supply base node. Constraint (4) is the in-degree constraint for the ending duplicated supply base node. Constraint (5) is the flow conservation constraint for intermediate duplicated supply base nodes, ensuring the orderly connection between consecutive trips. Constraint (6) is the flow balance constraint for order nodes. Constraint (7) is the uniqueness constraint, ensuring each order is served exactly once. Constraint (8) ensures that each vessel must visit the duplicated supply base nodes in sequence. Constraint (9) is the maximum payload constraint, ensuring the vessel load does not exceed capacity Q under any demand scenario. Constraint (10) is the payload update constraint, ensuring the load of vessel k under scenario s is properly updated when traversing arc (i,j) . Constraint (11) is the payload reset constraint, ensuring the vessel completes unloading at each intermediate duplicated supply base node. Constraint (12) defines the calculation of vessel sailing time. Constraint (13) defines the calculation of order service time. Constraints (14)-(15) are order service time window constraints, ensuring the service start time falls within the specified interval. Constraint (16) is the time continuity constraint, ensuring the start time of service for order j is no earlier than the completion time of service for order i plus the sailing time from i to j . Constraints (17)-(19) are variable definition constraints.

4. Proposed Branch-and-Price Algorithm for The PSVRSP-UD

To precisely solve the PSVRSP-UD, this study designs and implements a B&P algorithm [21]. Its core idea is to embed Column Generation (CG) within a Branch-and-Bound (B&B) framework [22]. By dynamically generating feasible trips with negative reduced cost, it avoids the combinatorial explosion caused by explicitly enumerating all routes, thereby achieving efficient global optimal solution of large-scale integer programming models. The following sequentially presents the model reformulation, pricing subproblem solution, branching strategy, and integer solution repair mechanism.

4.1. Model Reformulation via Dantzig-Wolfe Decomposition

The PSVRSP-UD is essentially analogous to the Multi-Trip Vehicle Routing Problem with Time Windows (MTVRPTW). For such multi-trip routing and scheduling problems, two typical set-partitioning models have been widely studied. The first model uses a complete vessel route as a column, directly representing a consecutive multi-trip sequence from departure to return [23]. Although structurally intuitive, it leads to an explosion of the column space due to the combination of trip connections and time schedules, making it impractical for real-scale instances. The second

model takes a single trip as the basic column unit and does not explicitly model the full route of a vessel [24]. It controls the non-overlapping trips of the same vessel via constraints, drastically reducing problem complexity and difficulty. This study adopts the second modeling approach and follows the exact solution paradigm for multi-trip problems.

Before presenting the formal set-partitioning model, we first provide a formal definition of a trip, which serves as the foundation for the subsequent column generation and pricing problem.

Definition 1. A trip r is composed of an ordered node sequence cus_r , a planned start time t_r^{start} , a maximum postponable time t_r^{relax} and a planned end time t_r^{end} such that:

- The total loading demand on the sequence does not exceed the vessel's capacity constraint;
- The vessel can start service at t_r^{start} , serve all orders according to the given sequence, return to the onshore supply base, and satisfy all time windows and sailing time constraints;
- The planned start time t_r^{start} and maximum postponable time t_r^{relax} jointly form the feasible departure time interval of the trip, covering all valid departure times under the visiting sequence.

The cost of trip r under scenario s ($s \in \mathcal{S}$) is denoted c_r^s , and the set of trips is denoted Ω . Based on this definition, the PSVRSP-UD can be modeled as a set-partitioning problem with mutual exclusion constraints, where y_r denotes whether trip r is selected, and the set to be covered is the order demand of offshore platforms. Mutual exclusion constraints ensure that selected trips can be assigned to K PSVs without overlapping in time for the same PSV. This paper enforces that at most K PSVs are used at any time. By discretizing time, a Boolean indicator u_{tr} is defined to indicate whether trip r covers time t , such that the number of occupied vessels at any time does not exceed the fleet size.

The resulting set-partitioning formulation of the PSVRSP-UD is presented as follows:

$$\min \sum_{r \in \Omega} \sum_{s \in \mathcal{S}} p_s c_r^s y_r, \quad (20)$$

Subject to

$$\sum_{r \in \Omega} a_{ir} y_r = 1 \quad \forall i \in \mathcal{O}, \quad (21)$$

$$\sum_{r \in \Omega} u_{tr} y_r \leq K \quad \forall t \in \{0, \dots, T \times 10^{nb}\}, \quad (22)$$

$$y_r \in \{0, 1\} \quad \forall r \in \Omega. \quad (23)$$

where a_{ir} indicates whether order i is covered by trip r , nb is the number of decimal digits for time precision. Constraint (21) ensures each order is served once. Constraint (22) ensures the number of PSVs used at any time intervals does not exceed the fleet size limit.

To avoid an explosion of constraints due to overly fine time precision, this study adopts a coarse time interval relaxation strategy. If the relaxed solution is feasible, it is optimal; otherwise, the granularity is refined. Specifically, the planning horizon $[0, T]$ is partitioned as follows. Let $l_{\min}$ be a small value guaranteeing that the duration of any feasible trip is greater than $l_{\min}$. The interval is partitioned as $\Delta_t = [l_{\min} \times t, l_{\min} \times (t + 1)]$ $t \in \{0, \dots, \lfloor \frac{T}{l_{\min}} \rfloor - 1\}$, and Constraint (22) is replaced by constraint (24).

$$\sum_{r \in \Omega} b_{tr} y_r \leq K \quad \forall t \in \{0, \dots, \lfloor \frac{T}{l_{\min}} \rfloor\}, \quad (24)$$

where $b_{tr} \in [0, 1]$ is the time proportion of trip r within interval Δ_t . Since $l_{\min}$ is smaller than any trip duration, $l_{\min}$ can accurately compute the start time of any trip without loss of precision. If the optimal relaxed solution is infeasible (violating (22)), then $l_{\min}$ is halved and recomputed.

The relaxed linear master problem (RLMP) is formulated as follows.

$$\min \sum_{r \in \hat{\Omega}} \sum_{s \in \mathcal{S}} p_s c_r^s y_r, \quad (25)$$

Subject to

$$\sum_{r \in \hat{\Omega}} a_{ir} y_r \geq 1 \quad \forall i \in \mathcal{O}, \quad (26)$$

$$\sum_{r \in \hat{\Omega}} b_{tr} y_r \leq K \quad \forall t \in \{0, \dots, \lfloor \frac{T}{l_{\min}} \rfloor\}, \quad (27)$$

$$1 \geq y_r \geq 0 \quad \forall r \in \hat{\Omega}. \quad (28)$$

4.2. Pricing Subproblem: NG-Route Labeling Algorithm

The pricing subproblem seeks a trip with negative reduced cost under all scenarios. Given dual values $\pi_i (i \in \mathcal{O})$ and $\psi_t (t \in \{0, \dots, \lfloor \frac{T}{l_{\min}} \rfloor\})$ corresponding to Constraints (26) and (27) of the RLMP, the reduced cost for $r (r \in \hat{\Omega})$ is presented as follows.

$$\sum_{s \in \mathcal{S}} p_s c_r^s - \sum_{i \in \mathcal{O}} a_{ir} \pi_i + \sum_{t \in \{0, \dots, \lfloor \frac{T}{l_{\min}} \rfloor\}} b_{tr} \psi_t, \quad (29)$$

This subproblem can be viewed as the Elementary Shortest Path Problem with Resource Constraints (ESPPRC) [25], typically solved by a dynamic programming labeling algorithm. However, for the PSVRSP-UD studied in this paper, the traditional labeling algorithm faces two challenges:

- the search space of the pricing subproblem corresponds to trips, and the time dual variable ψ_t leads to different reduced costs for different departure times of the same order sequence;
- multi-scenario demand constraints require each label to track resource consumption and feasibility under all demand scenarios, significantly increasing label dimension and pruning difficulty.

To improve computational efficiency and avoid this combinatorial difficulty, this paper introduces the NG-Route relaxation strategy [26] into the labeling algorithm, combined with the group of labels and representative label mechanisms [27], reducing computational complexity without losing optimality.

4.2.1. Definition of NG-Labels

The ESPPRC is defined on a directed graph $\mathcal{G}'(\mathcal{O} \cup \{o, d\}, \mathcal{A})$, where nodes o and d represent the onshore supply base as the path start and end, respectively. Under the NG-Route relaxation framework, a neighborhood set $\mathcal{NG}_i \subseteq \mathcal{O} \cup \{d\}$ is predefined for each node $i \in \mathcal{O}$, consisting of the closest nodes to i . Path extension is only allowed within the neighborhood, thereby significantly compressing the state space. Meanwhile, to avoid generating numerous duplicate labels with the same path but different times due to different departure times, this paper further introduces the group of labels and representative label mechanism. The core logic includes:

- All labels in the same group correspond to an identical visiting sequence path, differing only in departure time;
- Only one representative label is retained per group, and the postponable time $rt(\cdot)$ describes the flexible range of all feasible departure times within the group, such that every feasible trip in the same group can be derived from the representative label.

On this basis, an NG-Label characterizes the partial path state of the vessel under multi-scenario demand, time windows, and capacity constraints. Compared with traditional labels, the NG-Label not only records path information but also contains a neighborhood set to control the extension direction, further improving efficiency.

Each label $L_i(r(L_i), \eta(L_i), f(L_i), l(L_i), st(L_i), et(L_i), rt(L_i), \bar{c}(L_i), NG(L_i))$ stores the following data:

- $r(L_i)$ is the path from the origin to $\eta(L_i)$.
- $\eta(L_i)$ is the current node of the path $r(L_i)$.
- $f(L_i)$ is the parent label from which L_i is extended.
- $l(L_i) = \{l^s(L_i) | s \in S\}$ is the vessel load in each demand scenario, where $l^s(L_i)$ denotes the vessel load upon arrival at node $\eta(L_i)$ along path $r(L_i)$ under scenario s , representing the most critical state information in the label.
- $st(L_i)$ is the start time for the trip.
- $et(L_i)$ is the trip end time (the departure time from the previous node).
- $rt(L_i)$ is the maximum time that the label can be postponed while remaining feasible.
- $\bar{c}(L_i)$ is the reduced cost of the path, including travel cost, dual values of nodes, and dual value contribution of the time interval from $st(L_i)$ to $et(L_i)$.
- $NG(L_i)$ is the neighborhood set of the current node $\eta(L_i)$.

4.2.2. Label Extension Rules

In the extension phase, each label extension is checked for feasibility according to the following constraints.

Neighborhood Constraint: Only extend to nodes in the neighborhood NG_i of the current node $\eta(L_i)$.

$$j \in NG(L_i) \quad (30)$$

Time window constraint: Ensure the vessel's arrival time does not exceed the latest service time of the node.

$$et(L_i) + \tau_{ij} \leq b_j \quad \forall j \in NG(L_i) \quad (31)$$

Load constraint: Vessel load must not exceed the maximum capacity under all scenarios.

$$l^s(L_i) + q_j^s \leq Q \quad \forall s \in S, j \in NG(L_i) \quad (32)$$

where $l^s(L_i)$ is the current load in scenario s .

If all constraints are satisfied, a new label $L_j(r(L_j), \eta(L_j), f(L_j), l(L_j), st(L_j), et(L_j), rt(L_j), \bar{c}(L_j), NG(L_j))$ is generated with updated information.

$$r(L_j) = r(L_i) \cup \{j\} \quad (33)$$

$$\eta(L_j) = j \quad (34)$$

$$f(L_j) = L_i \quad (35)$$

$$l^s(L_j) = l^s(L_i) + q_j^s \quad \forall s \in S \quad (36)$$

$$st(L_j) = st(L_i) \quad (37)$$

$$et(L_j) = \max\{a_j + s_j, et(L_i) + \tau_{ij} + s_j\} \quad (38)$$

$$rt(L_j) = \min\{rt(L_i), b_j - et(L_j)\} \quad (39)$$

$$\begin{aligned}
& \bar{c}(L_j) \\
&= \bar{c}(L_i) + \sum_{s \in S} p_s c_{ij}^s - \pi_j \\
&+ \sum_{\bar{t} = \text{ind}(et(L_i))}^{\bar{t} = \text{ind}(et(L_j))} \frac{\min(et(L_j), \sup(\Delta_{\bar{t}})) - \max(et(L_i), \inf(\Delta_{\bar{t}}))}{\max(\epsilon, \sup(\Delta_{\bar{t}}) - \inf(\Delta_{\bar{t}}))} \psi_{\bar{t}}
\end{aligned} \tag{40}$$

$$NG(L_j) = \mathcal{NG}_j \tag{41}$$

Equations (33)-(41) update the information of the newly generated label L_j , including the path extended to the new node $\eta(L_j)$, the current node position, the parent label, the vessel's load upon across all scenarios, the departure time from $\eta(L_j)$, the trip start time, the updated arrival and departure time at the new node, the maximum feasible postponement time, and the updated neighborhood set. In Equation (40), $\text{ind}(et(L_i))$ denotes the index of the discretized time interval that contains the time $et(\cdot)$. Let $\inf(\Delta_{\bar{t}})$ and $\sup(\Delta_{\bar{t}})$ represent the lower and upper bounds of the time interval $\Delta_{\bar{t}}$, respectively. A small positive constant ϵ is introduced in order to avoid a division by 0 when $\sup(\Delta_{\bar{t}}) - \inf(\Delta_{\bar{t}}) = 0$. With this definition, the dual variables $\psi_{\bar{t}}$ are accumulated proportionally according to the proportion of time interval $\Delta_{\bar{t}}$ included in $[et(L_i), et(L_j)]$.

4.2.3. Dominance Rule

In the labeling algorithm, the design of the dominance rule determines computational efficiency. The core idea is that if one label at the same node is not better than another in terms of resource consumption and cost, it cannot lead to a better solution and can be pruned early. Traditional dominance strategies are usually based on the deterministic shortest path framework. However, given the multi-scenario demand and temporal flexibility of labels in the PSVRSP-UD, the following dominance criterion is designed. A label L_i^1 dominates another label L_i^2 if all following conditions are satisfied simultaneously.

$$r(L_i^1) \subseteq r(L_i^2) \tag{42}$$

$$f(L_i^1) = f(L_i^2) \tag{43}$$

$$l^s(L_i^1) \leq l^s(L_i^2) \quad \forall s \in S \tag{44}$$

$$et(L_i^1) \leq et(L_i^2) \tag{45}$$

$$rt(L_i^1) \geq rt(L_i^2) \tag{46}$$

$$\bar{c}(L_i^1) \leq \bar{c}(L_i^2) \tag{47}$$

This dominance rule reduces the number of labels and improves the efficiency of the pricing subproblem without losing optimality.

4.3. Branching Strategy and Integer Solution Repair

Solving the RLMP via CG only obtains a linear relaxation solution. To obtain an optimal schedule satisfying integer constraints, CG must be embedded within a B&B framework to form a complete B&P algorithm. However, in multi-trip scheduling problems such as the PSVRSP-UD, the traditional arc branching strategy has obvious limitations. Even if all arc flows are integer-valued, trip variables may still be fractional. This occurs because the same visiting sequence can form multiple feasible trips at different departure times, making arc flows unable to uniquely determine the time

schedule of trips. To address this issue, this section designs a two-level branching strategy tailored to the problem's characteristics, as shown in Figure 2.

If a fractional-flow arc exists, the conventional arc branching strategy is applied. If all arc flows are integer, a repair procedure is initiated to construct an integer trip schedule without time conflicts. All feasible trip structures in the current fractional solution are treated as virtual customers, constructing an auxiliary VRPTW subproblem. Each virtual customer corresponds to a trip node sequence cus_r , with time window $[t_r^{start}, t_r^{start} + t_r^{relax}]$ representing the feasible start interval of the sequence; service time and demand are set to 0; the fleet size is consistent with the original problem. The goal is to assign a conflict-free service sequence to these virtual customers, i.e., to construct a set of integer trips without overlapping times. The auxiliary VRPTW problem is solved by CG. If a feasible integer solution is obtained, the global optimal upper bound is updated and the current node is pruned. If repair fails, no integer solution can be formed based on the structures in the current solution, implying at least one arc in these structures does not belong to the optimal solution. A standard branching rule is applied on unconstrained arcs, creating two branches that force the arc flow to 0 and 1, respectively. If no feasible solution exists after all arcs with flow 1 are forbidden, the current node is directly pruned.

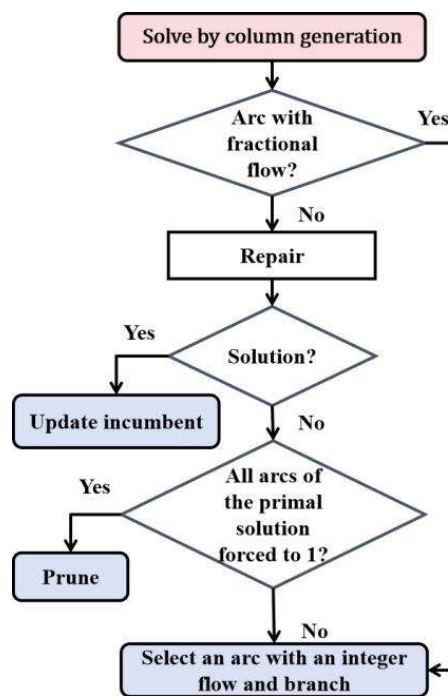


Figure 2. Proposed two-level branching strategy.

This two-level branching strategy effectively resolves the special issue of integer arc flows but fractional trip variables in multi-trip scheduling, guaranteeing the algorithm converges to the global optimal solution.

5. Numerical Experiments

To verify the effectiveness of the proposed model and algorithm, various computational experiments are conducted. The flow-based MILP model was solved using ILOG CPLEX 12.9, implemented in C++, with a time limit of 3600 seconds. All experiments were conducted on a Windows 10 system with an Intel(R) Core(TM) i5-8500 CPU @ 3.0 GHz and 8 GB RAM.

5.1. Test Instances and Parameter Settings

This section specifies the test instances and parameter settings adopted for PSVRSPUD. The benchmark instances are constructed based on real-world operational data from an oil and gas

company operating in Brazilian waters, as reported in Silva’s work [28]. A total of 6 offshore platforms are selected to form the test network. The sailing time from the onshore supply base to each platform ranges from 16 to 19 hours, and the average sailing time between any two platforms is 1.1 hours. During the planning horizon, each platform generates a random number of service orders. For each order, the expected demand is uniformly distributed in the interval [70, 150]. To capture demand uncertainty, a scenario generation and reduction method is employed to construct a set of representative demand scenarios denoted by $\mathcal{S}$. Each scenario $s \in \mathcal{S}$ is assigned with an occurrence probability p_s . The actual demand under each scenario follows a normal distribution with the expected demand as the mean and a standard deviation $\sigma \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$. Each standard deviation corresponds to a distinct demand variation level, which enables a comprehensive evaluation of the impact of different demand uncertainty degrees on scheduling decisions. For time-related parameters, the service time windows are generated in accordance with the classic Solomon scheme. The earliest arrival time at each node is determined by the sailing distance, and the width of each time window ranges from 30 to 70 hours, which does not exceed the total planning horizon of 168 hours.

Each instance is named according to a consistent rule that includes the number of platforms, orders, vessels, and the standard deviation of demand. For illustration, the instance name c6-o12-v3-d3-5 indicates an instance with 6 platforms (c), 12 orders (o), 3 vessels (v), a demand standard deviation of 0.3 (d), and the replicate index (n) 5.

5.2. Experimental Results and Analysis

5.2.1. Performance of the B&P Algorithm

The performance of the proposed B&P algorithm is compared with the flow-based model solved by CPLEX 12.9. Experiments in this section are conducted under a fixed demand standard deviation ($\sigma = 0.3$) and a weight coefficient ($\alpha = 0.5$), considering 10 demand scenarios. In each instance, statistical comparisons are performed on multiple metrics, including objective function value (Obj), computational time (T), upper bound (UB), lower bound (LB), gap ($GAP = \frac{UB-LB}{LB} \times 100\%$), and relative deviation ($RD = \frac{UB_{B\&P}-UB_{CPLEX}}{UB_{CPLEX}} \times 100\%$).

The comparative results of the algorithms on different test instances are summarized in Tables 2 and 3. For small-scale instances with 4-8 orders, both CPLEX and B&P can obtain optimal solutions within the 3600-second time limit. In terms of computational efficiency, the B&P algorithm exhibited a significant advantage. The objective values are exactly the same, and the relative deviation (RD) is 0%, which verifies that the optimal solutions obtained by the B&P algorithm are consistent with those of CPLEX. In terms of computational efficiency, the B&P algorithm demonstrates a remarkable advantage. For instances with 4 orders, the average solution time of B&P is only 0.012 seconds, while CPLEX requires approximately 0.25 seconds, making B&P more than 95% faster on average. For instances with 8 orders, the computational time of CPLEX increases sharply to a range of 522.5 seconds to 1586.91 seconds, with an average of 461.97 seconds. In contrast, the B&P algorithm solves the same instances in only 0.02 seconds to 1.7 seconds, with an average of merely 0.31 seconds. These results also indicate that B&P’s advantage in computational efficiency becomes more pronounced with modest increases in problem size.

Table 2. Comparison of B&P and CPLEX results for small-scale instances.

Instance	CPLEX		B&P		RD(%)
	Obj	T(s)	Obj	T(s)	
c6-o4-v2-d3-1	25,432,210.37	0.24	25,432,210.37	0.01	0.00
c6-o4-v2-d3-2	26,710,724.33	0.29	26,710,724.33	0.01	0.00
c6-o4-v2-d3-3	25,521,074.33	0.33	25,521,074.33	0.01	0.00
c6-o4-v2-d3-4	24,782,629.51	0.17	24,782,629.51	0.02	0.00
c6-o4-v2-d3-5	24,834,059.25	0.24	24,834,059.25	0.01	0.00

c6-o8-v2-d3-1	53,468,579.29	1,586.91	53,468,579.29	0.82	0.00
c6-o8-v2-d3-2	52,032,791.70	734.77	52,032,791.70	1.70	0.00
c6-o8-v2-d3-3	52,806,246.00	907.11	52,806,246.00	0.30	0.00
c6-o8-v2-d3-4	54,222,141.25	867.13	54,222,141.25	0.15	0.00
c6-o8-v2-d3-5	52,919,017.96	522.5	52,919,017.96	0.02	0.00
AVG	39,272,947.40	461.97	39,272,947.40	0.31	0.00

For medium- and large-scale instances (involving 12 to 24 orders), CPLEX 12.9 fails to obtain optimal solutions within the 3600-second time limit. The average GAP of CPLEX reaches 97.38%, and this gap tends to increase with the instance size, resulting in a rapid decline in solution quality for larger instances—even failing to output valid UB and LB values for some large-scale instances. This indicates the limited ability of CPLEX to solve large-scale PSVRSP-UD using flow-based models. In contrast, the proposed B&P algorithm consistently obtains global optimal solutions, and the computed optimal solutions fall within the UB range reported by CPLEX. In terms of computational efficiency, the average runtime of the B&P algorithm is only 461.95 seconds, while CPLEX is forced to terminate at the 3600-second time limit. The RD analysis shows that the B&P algorithm achieves an average RD of -4.18%, indicating that its optimal solutions are even better than the upper bounds of CPLEX in most cases. For example, the instance c6-o20-v3-d3-4 achieves an RD of -12.74%, showing the most significant cost advantage, while instances c6-o12-v3-d3-3 and c6-o16-v3-d3-4 achieve an RD of 0.00%, indicating their optimal solutions are consistent with CPLEX’s upper bounds. These results collectively verify the effectiveness and practical relevance of the proposed approach in complex and uncertain offshore logistics environments.

In summary, the B&P algorithm significantly outperforms CPLEX in both solution quality and computational efficiency. It can obtain exact solutions within a reasonable computational time, exhibits robust convergence behavior, making it highly suitable for complex real-world PSVRSP applications.

Table 3. Comparison of B&P and CPLEX results for medium- and large-scale instances.

Instance	CPLEX				B&P				RD (%)
	UB	LB	GAP _{CPL} (%)	T(s)	UB	LB	GAP _{B&P} (%)	T(s)	
c6-o12-v3-d3-1	75,737,827.4 9	3,925,699.9 6	94.28	3609.7 0	72,882,926.25	72,882,926.2 5	0.00	33.4	-3.77
c6-o12-v3-d3-2	77,713,692.0 9	4,099,999.9 7	94.50	3609.4 2	74,498,014.61	74,498,014.6 1	0.00	6.81	-4.14
c6-o12-v3-d3-3	74,717,891.3 7	3,829,999.9 6	94.44	3610.8 0	74,717,891.37	74,717,891.3 7	0.00	20.47	0.00
c6-o12-v3-d3-4	77,570,204.7 8	3,979,999.9 7	95.24	3610.1 8	77,570,204.78	77,570,204.7 8	0.00	16.28	0.00
c6-o12-v3-d3-5	74,274,463.1 3	3,789,999.9 6	94.23	3608.5 4	73,724,533.93	73,724,533.9 3	0.00	12.43	-0.74
c6-o16-v3-d3-1	103,319,547. 14	1,419,999.9 9	98.69	3606.7 0	102,281,751.6 0	102,281,751. 60	0.00	416.04	-1.00
c6-o16-v3-d3-2	92,914,482.4 8	1,279,999.9 9	98.67	3607.7 2	89,223,758.69	89,223,758.6 9	0.00	382.9	-3.97
c6-o16-v3-d3-3	106,689,087. 95	2,219,999.9 9	97.91	3608.2 1	100,673,785.1 9	100,673,785. 19	0.00	244.69	-5.64
c6-o16-v3-d3-4	91,296,267.6 7	2,899,999.9 9	96.84	3608.0 0	91,293,953.82	91,293,953.8 2	0.00	163.97	0.00
c6-o16-v3-d3-5	104,033,923. 82	1,519,999.9 9	98.72	3608.0 7	98,012,638.27	98,012,638.2 7	0.00	137.55	-5.79

c6-o20-v3-d3-1	122,445,304.67	2,409,999.99	98.11	3609.75	113,177,774.57	113,177,774.57	0.00	481.07	-7.57
c6-o20-v3-d3-2	117,544,619.02	2,249,999.99	98.80	3610.23	109,487,790.33	109,487,790.33	0.00	367.92	-6.85
c6-o20-v3-d3-3	136,524,805.65	2,009,999.99	98.52	3608.18	131,382,840.15	131,382,840.15	0.00	682.35	-3.77
c6-o20-v3-d3-4	119,657,794.44	2,169,999.99	98.22	3610.56	104,412,686.33	104,412,686.33	0.00	1560.81	-12.74
c6-o20-v3-d3-5	-	-	-	3624.08	168,345,412.26	168,345,412.26	0.00	2627.07	-
c6-o24-v4-d3-1	161,747,185.53	349,999.99	99.79	3626.18	147,194,030.48	147,194,030.48	0.00	313.78	-9.00
c6-o24-v4-d3-2	145,716,786.60	1,339,999.99	99.08	3625.65	144,869,150.42	144,869,150.42	0.00	142.19	-0.58
c6-o24-v4-d3-3	-	-	-	3,611.88	145,462,488.39	145,462,488.39	0.00	677.11	-
c6-o24-v4-d3-4	148,898,512.90	869,999.99	99.42	3609.70	140,673,070.67	140,673,070.67	0.00	535.70	-5.52
c6-o24-v4-d3-5	-	-	-	3609.42	131,850,277.58	131,850,277.58	0.00	416.50	-
AVG	107,694,258.63	2,374,452.92	97.38		102,710,400.09	102,710,400.09	0.00	461.95	-4.18

5.2.2. Sensitivity Analysis on Number of Scenario

To investigate the influence of the number of demand scenarios on solutions under demand uncertainty, this section conducts a sensitivity analysis on the 4-port instance c6-o4-v2-d3-n, varying the number of scenarios from 10 to 100. The total cost and CPU time are illustrated in Figure 3. It shows that the total cost exhibits a significant downward trend as the number of scenarios increases from 10 to 100, rather than remaining robust as previously described. Specifically, the average total cost of the five instances decreases from 25,456,139.56 (10 scenarios) to 14,426,007.87 (100 scenarios), with a total reduction of approximately 43.33%. This indicates that increasing the number of scenarios can better capture the characteristics of demand uncertainty, and the optimal scheduling decisions are gradually adjusted to be more cost-effective as the scenario number increases.

In terms of computational efficiency, the algorithm’s runtime shows a stable positive correlation with the number of scenarios. For example, the CPU runtime of c6-o4-v2-d3-1 increases from 0.24 seconds to 2.61 seconds; for c6-o4-v2-d3-5, the runtime rises from 0.14 seconds to 1.94 seconds. On average, the runtime of the five instances increases from 0.214 seconds to 2.218 seconds, which is a tenfold increase but still remains at a low level, ensuring high computational efficiency.

Overall, the results indicate that the scenario setting adopted in this study can balance the trade-off between solution quality and computational efficiency.

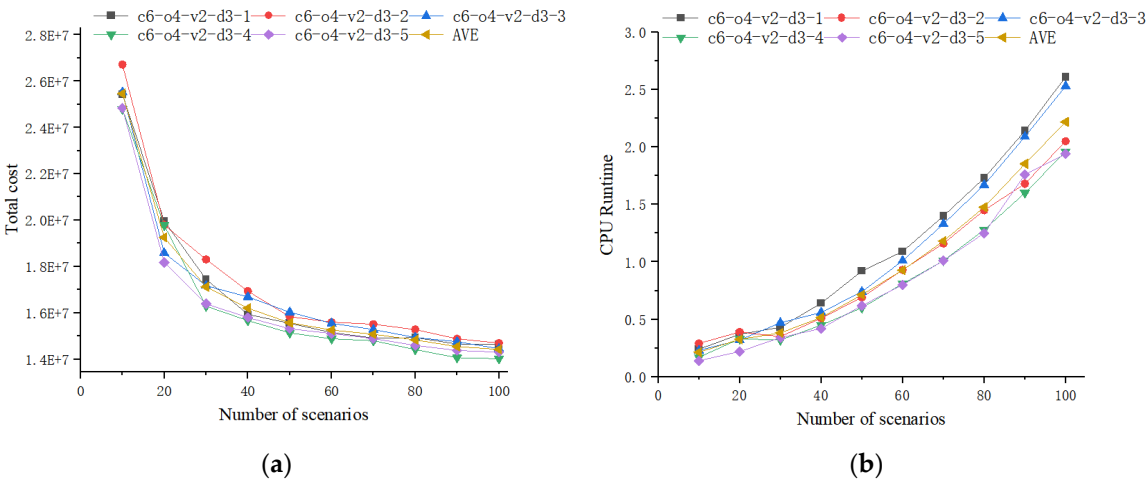


Figure 3. Sensitivity analysis for the number of scenarios: (a) Total cost; (b) Total cost.

5.2.3. Sensitivity Analysis on Demand Fluctuation Levels

To investigate the impacts of demand uncertainty on scheduling results, this section conducts a sensitivity analysis under different levels of demand fluctuation. Tests are carried out on 8-order instance c6-o8-v2-n as an example; the standard deviation of demand is varied. Specifically, σ is set to 0.1, 0.2, 0.3, 0.4, and 0.5, representing increasing uncertainty levels. All other parameters remain unchanged. The detailed results are reported in Table 4.

In terms of solution quality, the total cost generally shows an upward trend as demand uncertainty increases. As σ rises from 0.1 to 0.5, the average total cost increases from 52,513,500.63 to 52,998,710.12, indicating that higher demand volatility requires more conservative scheduling and additional capacity reserves to ensure feasibility, thereby raising the overall operational cost. At the instance level, most test problems show a gradual rise in objective value with increasing σ , reflecting the additional economic burden induced by demand uncertainty. Regarding computational efficiency, an overall decline in solution time can be observed as σ increases. The average computing time decreases monotonically from 1256.17 seconds at $\sigma=0.1$ to 800.82 seconds at $\sigma=0.5$. This phenomenon suggests that stronger demand uncertainty leads to a more constrained solution space and tighter feasibility conditions, which in turn accelerates the convergence of the algorithm and reduces the overall solution time without significantly compromising solution quality.

In summary, increasing demand fluctuation results in higher total expected cost due to the need for more robust scheduling arrangements, while simultaneously improving the computational efficiency of the proposed B&P algorithm.

Table 4. Sensitivity analysis for demand variability.

Instance	0.1		0.2		0.3		0.4		0.5	
	Obj	T(s)	Obj	T(s)	Obj	T(s)	Obj	T(s)	Obj	T(s)
c6-o8-v2-1	53,391,416.8	1,920.6	53,430,007.6	1,572.9	53,430,007.6	1,572.9	53,792,988.8	1,244.1	53,829,472.1	1,206.1
c6-o8-v2-2	51,441,580.1	698.5	51,530,934.0	747.6	51,530,934.0	747.6	52,123,967.2	660.9	52,215,260.6	641.9
c6-o8-v2-3	52,385,311.7	1,207.8	52,595,867.3	1,180.2	52,595,867.3	1,180.2	53,016,814.1	835.5	53,227,180.1	785.6
c6-o8-v2-4	54,129,789.9	1,030.9	54,176,259.6	888.7	54,176,259.5	888.7	54,268,422.0	719.4	54,588,344.6	604.6
c6-o8-v2-5	51,219,404.6	1,422.9	51,190,494.6	1,323.3	51,190,494.6	1,323.3	51,132,735.9	1,168.3	51,133,293.2	765.8
AVE	52,513,500.6	1,256.2	52,584,712.6	1,142.6	52,738,275.6	1,087.1	52,866,985.6	925.7	52,998,710.1	800.8

5.2.4. Sensitivity Analysis on Weight Coefficient α

To explore the impact of economic objectives and low-carbon objectives on the optimal scheduling scheme, a sensitivity analysis is conducted with respect to the weight coefficient α . Tests are carried out on 8-order instance c6-o8-v2-d3-n as an example; all other parameters are held constant. Specifically, α is set to five typical levels: 0, 0.25, 0.5, 0.75, 1, where a larger α indicates greater emphasis on minimizing operational time cost, and a smaller α highlights the control of carbon emissions. The corresponding results of sailing time and sailing distance of the optimal scheduling scheme under different weight configurations are presented in Figure 4.

Experimental results show that the weight coefficient α has a mild but stable impact on both sailing time and distance, without causing significant fluctuations or monotonic trends. When α varies from 0 to 1, the sailing time (time in best cost) maintains a relatively steady level across all 8-order instances. Specifically, the sailing time ranges approximately between 127.59 and 138.40, with no abrupt increases or decreases. For example, the time for instance c6-o8-v2-d3-2 remains 127.59 when $\alpha = 0.25$ and $\alpha = 1$, while the time for c6-o8-v2-d3-3 reaches the maximum of 138.80 when $\alpha = 0.25$, showing only slight variations across different α levels.

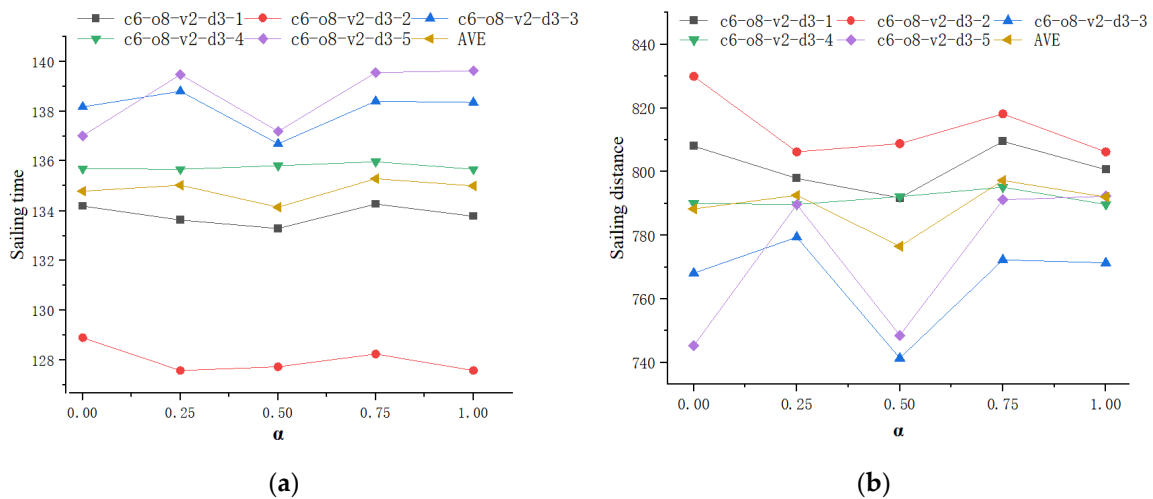


Figure 4. Sensitivity analysis for the weight coefficient: (a) Sailing time of the optimal scheduling scheme; (b) Sailing distance of the optimal scheduling scheme.

In terms of sailing distance (distance in best cost), it also remains stable with the change of α . The distance ranges between 741.38 and 829.96, with no obvious monotonic trend as α increases or decreases. For most instances, the distance fluctuates slightly within a narrow range across different α levels; for example, the distance of c6-o8-v2-d3-4 varies only between 789.71 and 795.14, with a maximum variation of less than 0.7%.

Although α adjusts the trade-off priority between operational time cost and carbon emission cost, the optimal scheduling scheme remains robust and efficient. The stable sailing time and distance across different α levels indicate that adjusting the weight coefficient does not alter the core structure of the optimal route. In practical applications, decision-makers can flexibly select an appropriate α according to corporate emission-reduction policies, operational budgets, or energy-saving requirements, without causing obvious deterioration in sailing efficiency or route rationality.

6. Conclusions

This study addresses the platform supply vessel routing and scheduling problem under demand uncertainty, aiming to minimize the expected total cost (including vessel operating costs and carbon emission costs) while satisfying multi-dimensional operational constraints such as vessel capacity, time windows, and multi-trip limits. A scenario-based MILP model is developed, which integrates stochastic offshore demand through multi-scenario methods. Specifically, demand uncertainty is

represented by a finite set of representative scenarios generated using LHS combined with Cholesky decomposition, and the uncertainty is incorporated into the MILP by enforcing feasibility under all scenarios while minimizing the expected objective value weighted by scenario probabilities. The SBR algorithm is further adopted to prune redundant scenarios, ensuring computational tractability without compromising solution fidelity. To efficiently solve this NP-hard problem, a B&P algorithm is proposed that incorporates Dantzig–Wolfe decomposition, NG-Route labeling with group labels and representative label mechanisms, and a two-level branching strategy to accelerate convergence and ensure global optimality.

Systematic numerical experiments on instances generated from real-world offshore logistics data demonstrate that the proposed B&P method achieves optimal solutions and outperforms CPLEX in all test instances. For small-scale instances, the B&P algorithm obtains the same optimal solutions as CPLEX but with significantly higher computational efficiency. For medium- and large-scale instances, CPLEX fails to converge to optimal solutions within the time limit, whereas the B&P algorithm consistently achieves global optimality with a zero optimality gap. Sensitivity analyses reveal that a reasonable number of representative demand scenarios can effectively characterize demand uncertainty, and that increasing the number of scenarios improves solution quality without significant loss of computational efficiency. Higher demand fluctuation increases the expected total cost due to the need for more conservative scheduling but accelerates algorithm convergence by tightening the solution space. In addition, the weight coefficient balancing operational and low-carbon objectives has a mild but stable impact on scheduling results, allowing flexible adjustment according to practical decision needs. These results collectively verify the effectiveness and practical relevance of the proposed approach in complex and uncertain offshore logistics environments.

This work opens several directions for future research on the PSVRSP. First, the current modeling framework assumes that demands at different offshore platforms are mutually independent, whereas spatial or temporal correlations may exist in practical offshore operations. Incorporating demand correlations into scenario generation may better capture real-world demand dynamics. In addition, the current study focuses only on demand uncertainty and does not consider other common offshore uncertainties, such as adverse weather conditions. Future research may integrate multi-source uncertainties to further enhance the robustness of scheduling solutions. Furthermore, the algorithm's performance may be improved by incorporating more advanced pruning strategies or parallel computing techniques to handle larger-scale instances with more complex operational constraints.

Author Contributions: Conceptualization, B.J. and J.L.; methodology, B.J. and J.L.; software, J.L.; validation, B.J., J.L. and S.Y.; formal analysis, J.L.; investigation, J.L.; resources, B.J.; data curation, J.L.; writing—original draft preparation, B.J. and J.L.; writing—review and editing, B.J. and S.Y.; visualization, J.L.; supervision, B.J. and S.Y.; project administration, B.J.; funding acquisition, B.J. All authors have read and agreed to the published version of the manuscript.

Funding: Please add: This research was funded by National Natural Science Foundation of China, grant number No. 72371250; Natural Science Outstanding Youth Foundation of Hunan Province, China, grant number No. 2024JJ4073 and The Science and Technology Innovation Program of Hunan Province, China, grant number No. 2024RC3011.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Guo, L.; Song, W.; Long, Y. Maritime service network design with oil-fueled and LNG-fueled mixed vessel fleet under uncertainty: Overview and future research directions. *Transp. Econ. and Manag.* **2025**, *3*, 423–431.

2. Shu, Y.; Guo, Y.; Du, Y.; Wang, S. A Mixed-Integer Linear Programming Framework for Multi-Period Offshore Wind Personnel Logistics: Integrating Routing, Scheduling, and Personnel Inventory Management. *Mathematics* **2026**, *14*, 978.
3. Friedberg, D.O.; Uglane, V.T. Routing and Scheduling of Platform Supply Vessels; Norwegian University of Science and Technology: Trondheim, Norway, 2013.
4. Fagerholt, K.; Lindstad, H. Optimal policies for maintaining a supply service in the Norwegian Sea. *Omega* **2000**, *28*, 269–275.
5. Aas, B.; Gribkovskaia, I.; Halskau Sr, Ø.; Shlopak, A. Routing of supply vessels to petroleum installations. *Int. J. Phys. Distrib. Logist. Manag.* **2007**, *37*, 164–179.
6. Gribkovskaia, I.; Laporte, G.; Shlopak, A. A tabu search heuristic for a routing problem arising in servicing of offshore oil and gas platforms. *J. Oper. Res. Soc.* **2008**, *59*, 1449–1459.
7. Shyshou, A.; Gribkovskaia, I.; Laporte, G.; Fagerholt, K. A large neighbourhood search heuristic for a periodic supply vessel planning problem arising in offshore oil and gas operations. *INFOR* **2012**, *50*, 195–204.
8. Halvorsen-Weare, E.E.; Fagerholt, K.; Nonås, L.M.; Asbjørnslett, B.E. Optimal fleet composition and periodic routing of offshore supply vessels. *Eur. J. Oper. Res.* **2012**, *223*, 508–517.
9. Kisialiou, Y.; Gribkovskaia, I.; Laporte, G. Supply vessel routing and scheduling under uncertain demand. *Transport. Res. C: Emerg. Technol.* **2019**, *104*, 305–316.
10. Halvorsen-Weare, E.E.; Fagerholt, K. Optimization in offshore supply vessel planning. *Optim. Eng.* **2017**, *18*, 317–341.
11. Cruz, R.; Mendes, A.B.; Bahiense, L. Schedule robustness in the periodic supply vessels planning problem with stochastic demand and travel time. *Int. Trans. Oper. Res.* **2024**, *31*, 2338–2365.
12. Ksciuk, J.; Kuhlemann, S.; Tierney, K.; Koberstein, A. Uncertainty in maritime ship routing and scheduling: A literature review. *Eur. J. Oper. Res.* **2023**, *308*, 499–524.
13. Aas, B.; Halskau Sr, Ø.; Wallace, S.W. The role of supply vessels in offshore logistics. *Marit. Econ. & Logist.* **2009**, *11*, 302–325.
14. Salehi Sadghiani, N.; Torabi, S.A.; Sahebjamnia, N. Retail supply chain network design under operational and disruption risks. *Transport. Res. E: Logist. Transport. Rev.* **2015**, *75*, 94–113.
15. Alrefaei, M.H.; Andradottir, S. Simulated annealing algorithm with constant temperature for discrete stochastic optimization. *Manag. Sci.* **1999**, *45*, 772–786.
16. Christiansen, M.; Fagerholt, K. Robust ship scheduling with multiple time windows. *Nav. Res. Logist.* **2002**, *49*, 611–625.
17. Hu, J.; Li, H. A new clustering approach for scenario reduction in multi-stochastic variable programming. *IEEE Trans. Power Syst.* **2019**, *34*, 3813–3825.
18. Nikzad, E.; Bashiri, M.; Oliveira, F. Two-stage stochastic programming approach for the medical drug inventory routing problem under uncertainty. *Comput. & Ind. Eng.* **2019**, *128*, 358–370.
19. Avolio, M.; Di Francesco, M.; Fuduli, A.; Gorgone, E.; Calvo, R.W. Multi-day routes in a multi-depot vehicle routing problem with intermediate replenishment facilities and time windows. *Comput. & Oper. Res.* **2025**, *182*, 107084.
20. Psaraftis, H.N.; Kontovas, C.A. Ship speed optimization: Concepts, models and combined speed-routing scenarios. *Transp. Res. Part C: Emerg. Technol.* **2014**, *44*, 52–69.
21. Zhu, Z.; Zhang, S.; Cao, Z. Exact and heuristic methods for the integrated optimisation of vessel deployment and liner shipping routing schedule. *Int. J. Syst. Sci.: Oper. & Logist.* **2025**, *12*, 2505920.
22. Ji, B.; Zhou, S.; Zhang, D.; Yu, S.S. A branch-and-price-based heuristic for the vehicle routing problem with two-dimensional loading constraints and time windows. *Int. Trans. Oper. Res.* **2024**, *31*, 658–691.
23. Ceselli, A.; Righini, G.; Salani, M. A column generation algorithm for a rich vehicle-routing problem. *Transp. Sci.* **2009**, *43*, 56–69.
24. Hernandez, F.; Feillet, D.; Giroudeau, R.; Naud, O. A new exact algorithm to solve the multi-trip vehicle routing problem with time windows and limited duration. *4OR* **2014**, *12*, 235–259.

25. Ji, B.; Zhang, D.; Zhang, Z.; Yu, S.S.; Van Woensel, T. The generalized serial-lock scheduling problem on inland waterway: A novel decomposition-based solution framework and efficient heuristic approach. *Transport. Res. E: Logist. Transport. Rev.* **2022**, *168*, 102935.
26. Baldacci, R.; Mingozzi, A.; Roberti, R. New route relaxation and pricing strategies for the vehicle routing problem. *Oper. Res.* **2011**, *59*, 1269–1283.
27. Hernandez, F.; Feillet, D.; Giroudeau, R.; Naud, O. Branch-and-price algorithms for the solution of the multi-trip vehicle routing problem with time windows. *Eur. J. Oper. Res.* **2016**, *249*, 551–559.
28. Silva, V.A.; Wang, A.; Ferreira Filho, V.J.M.; Gounaris, C.E. Routing and scheduling of platform supply vessels in offshore oil and gas logistics. *Comput. & Oper. Res.* **2024**, *164*, 106556.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.