mathematics

MDPI

Article

Ship Selection and Inspection Scheduling in Inland Waterway Transport

Xizi Qiao [1], Ying Yang [2,*], King-Wah Pang [2], Yong Jin [2] and Shuaian Wang [2]

[1] Tsinghua-Berkeley Shenzhen Institute, Tsinghua University, Shenzhen 518055, China; qxz22@mails.tsinghua.edu.cn

[2] Faculty of Business, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong 999077, China; anthony.pang@polyu.edu.hk (K.-W.P.); jimmy.jin@polyu.edu.hk (Y.J.); hans.wang@polyu.edu.hk (S.W.)

* Correspondence: shadow.yang@connect.polyu.hk

Abstract: Inland waterway transport is considered a critical component of sustainable maritime transportation and is subject to strict legal regulations on fuel quality. However, crew members often prefer cheaper, inferior fuels for economic reasons, making government inspections crucial. To address this issue, we formulate the ship selection and inspection scheduling problem into an integer programming model under a multi-inspector and multi-location scenario, alongside a more compact symmetry-eliminated model. The two models are developed based on ship itinerary information and inspection resources, aiming to maximize the total weight of the inspected ships. Driven by the unique property of the problem, a customized heuristic algorithm is also designed to solve the problem. Numerical experiments are conducted using the ships sailing on the Yangtze River as a case study. The results show that, from the perspective of the computation time, the compact model is 102.07 times faster than the original model. Compared with the optimal objectives value, the gap of the solution provided by our heuristic algorithm is 0.37% on average. Meanwhile, our algorithm is 877.19 times faster than the original model, demonstrating the outstanding performance of the proposed algorithm in solving efficiency.

Keywords: inland waterway transport; inspection scheduling; ship selection; mathematical modeling

MSC: 90C10

check for updates

Citation: Qiao, X.; Yang, Y.; Pang, K.-W.; Jin, Y.; Wang, S. Ship Selection and Inspection Scheduling in Inland Waterway Transport. *Mathematics* **2024**, *12*, 2327. https://doi.org/10.3390/math12152327

Academic Editors: Yu-Wang Chen, Mi Zhou and Tao Wen

Received: 20 June 2024
Revised: 14 July 2024
Accepted: 24 July 2024
Published: 25 July 2024

1. Introduction

The maritime sector, responsible for 3% of global greenhouse gas emissions, has seen a 20% increase in emissions over the past decade [1]. Consequently, the sector is now at a critical juncture, confronted with the significant challenge of balancing environmental goals with economic needs. Inland waterway transport is increasingly recognized as a critical component of sustainable maritime transportation, particularly in the context of severe environmental pollution [2–4]. As the world grapples with the dual challenges of environmental degradation and climate change, the importance of inland waterway transport becomes evident due to its low energy consumption, substantial transport capacity, and minimal environmental impact compared to other transport modes [5–8]. This environmentally friendly alternative not only reduces the carbon footprint of freight logistics but also helps to mitigate congestion on overburdened road networks. However, the environmental advantages of inland waterway transport can be significantly undermined if the fuel used by ships fails to meet quality standards, leading to substantial pollution. Additionally, the proximity of these waterways to urban centers and densely populated areas means that pollutants discharged from ships can have a direct and immediate impact on residential environments. This results in not only the degradation of water quality but also compromised air quality, posing potential health hazards to the local population. The frequent traffic of inland waterway ships exacerbates these issues, positioning it as a critical concern for urban environmental management and sustainability efforts.

In response to the pressing need for environmental protection, stringent regulations have been established to control the fuel quality used by inland waterway vessels. In China, for example, the Law of the People's Republic of China on the Prevention and Control of Atmospheric Pollution (Article 63) mandates that inland and river-sea direct vessels must use regular diesel that meets GB252 standards, which specifies a sulfur content of no more than 350 ppm [9]. The law prohibits the use of residual oil and heavy oil, which are commonly used as marine fuel oils but have a high sulfur content and contribute significantly to air pollution. Despite these regulations, compliance remains a significant challenge, primarily due to economic factors. From the ship owners' perspective, the lower cost of substandard marine fuel oil, which is easier and cheaper to produce, often outweighs the considerations of environmental impact. This economic incentive drives both ship owners and fuel suppliers towards cheaper, lower-quality fuels, thereby undermining efforts to promote higher-quality diesel and impeding progress in atmospheric pollution control and green development.

Effective regulation and enforcement are crucial to addressing this issue. To ensure compliance with environmental standards, local maritime authorities implement rigorous inspection regimes. The inspection team conducted a random inspection of inland waterway ships by examining the records in the statutory documents, as well as the retention of certificates including the ship's fuel quality report, fueling invoices, and delivery notes on the inspected vessels. Samples are taken from operating generators to test the sulfur content of the fuel used, and any discrepancies from the standards will be strictly penalized. According to Article 106 of the Law on the Prevention and Control of Atmospheric Pollution [9], violations, such as the use of non-compliant fuel, can result in fines ranging from CNY 10,000 to CNY 100,000, imposed by maritime management agencies and fisheries authorities.

These stringent inspection and enforcement measures are essential for reducing pollutant emissions from vessels and promoting the development of green ships. By ensuring that vessels comply with fuel quality standards, regulatory authorities play a crucial role in mitigating the environmental impact of maritime transportation. Thus, timely and efficient inspection is vital for achieving these environmental goals and advancing the sustainable development of inland waterway transport. Articles focusing on ship inspection primarily concentrate on predicting which ships most urgently need inspection based on ship information [10–13]. However, these studies do not incorporate the information on ship itineraries and the allocation of inspection resources, thereby falling short of providing practical and implementable solutions for inspectors. A few studies that combine ship selection and inspection scheduling are limited to sea ships in maritime transportation [14–17], which has a different problem structure compared with inland waterway scenarios. In the maritime context, ship inspections are primarily categorized into Port State Control (PSC) and Flag State Control (FSC). PSC involves multiple inspectors examining ships at the same port, whereas FSC involves a team of inspectors with the same itinerary conducting inspections at different ports. In inland waterways, due to the relatively high density of ports and smaller scheduling range, it is practical to dispatch multiple inspectors to various ports simultaneously. However, there is currently a lack of research on ship inspection issues in inland waterways. What is more, these existing studies primarily use commercial solvers to solve problems, leading to computational inefficiency and the inability to obtain results in scenarios without the support of commercial solvers.

To address these research gaps, our study makes the following contributions. First, to deal with the issues of ship selection and inspection scheduling in inland waterway transport, we develop an integer programming (IP) model for the multi-inspector and multi-location scenario. With the risk weight and berthing information of merchant ships, the model determines the location assignment and merchant ship inspection decision for each government ship in one day, with the aim of maximizing the total weight of inspected merchant ships. Additionally, we formulate a compact IP model without symmetry to accelerate the problem-solving process. Furthermore, driven by the unique structure and property of the problem, we design a customized heuristic algorithm, the weight-induced inspection scheduling algorithm, that performs well in terms of solution quality and computation time. The algorithm significantly improves the efficiency of solving the problem, laying a foundation for its application to large-scale instances. Lastly, we test the effectiveness of the two IP models and the heuristic algorithm using the Yangtze River in China as a case study. Our models and the algorithm provide efficient solutions for the management of inland waterway transport, ensuring more effective and rational ship inspection scheduling. This enhances inspection efficiency and contributes to the operational security of ships as well as the protection of the inland water environment.

2. Literature Review

The importance of maritime ship inspections cannot be overstated as they play a critical role in ensuring maritime safety and protecting the marine environment [18–20]. Effective inspections prevent substandard ships from operating, which can result in significant monetary savings and environmental protection [21]. Furthermore, post-inspection effects include reduced inspection intervals and fewer reported deficiencies in subsequent inspections [22]. Before conducting an inspection, determining which ships should be selected for inspection among all incoming vessels is a critical issue for authorities, because limited time and resources must be allocated to inspect those ships that are in worse condition [14]. Accordingly, ship inspection can be divided into two main components: ship selection and task scheduling. Ship selection involves identifying which ships to inspect, while task scheduling pertains to the allocation and prioritization of inspection tasks due to limited resources.

During the ship inspection process, any condition not meeting the requirements of the relevant convention is termed a ship deficiency. If critical deficiencies are found, authorities may detain the ship [23]. Predictive models for identifying high-risk ships use both abstract risk levels and specific deficiency or detention conditions as targets. Notable methods include the Bayesian network, support vector machine, logistic regression, and tree-based model [10–13]. For instance, Yang et al. [10] implement a Bayesian network approach to predict ship detention based on inspection data from the Paris MoU. They analyzed key factors such as the number of deficiencies, type of inspection, recognized organization, and ship age, and proposed a strategic game model to determine the optimal inspection rate for port states, which aims to find the optimal balance between the resources spent on inspections and the benefits gained from increased safety and compliance. Some studies consider ships involved in casualties and incidents as indicators of high ship risk and potential future accidents [24,25]. Yan et al. [11] develop a random forest-based model to predict ship detention probabilities, showing it outperformed the selection scheme of ship risk profile [26] in identifying detained ships. However, these models only generate risk scores for ship selection and lack comprehensive inspection guidance, such as ship visit information, arrival times, berthing periods, and port inspection resources. This limitation reduces their practical applicability and feasibility in real-world scenarios. To address these shortcomings, it is essential to integrate ship selection with task scheduling to enhance the efficiency and effectiveness of inspections.

Task scheduling ensures that limited resources are allocated optimally, and inspections are conducted in a timely and organized manner, ultimately improving maritime safety and compliance. In the field of maritime transportation, Rizvanolli and Heise [27] introduce a mixed-integer programming (MIP) model for crew scheduling, optimizing tasks and qualifications to reduce costs and avoid port authority detentions. Leggate et al. [28] fill a gap in maritime crew scheduling, previously underexplored compared to airlines and railways, by proposing MIP formulations for offshore supply vessels, demonstrating real-time solution generation and statistical analysis of key parameters. Addressing inefficiencies in traditional pilot scheduling, Xiao et al. [29] propose a variable neighborhood search approach combined with MIP, reducing operating costs and pilot workloads, thus enhancing job satisfaction. Jia et al. [30] examine integrated vessel traffic and pilot scheduling at seaports, developing an IP model that optimizes navigation channels and anchorage areas, incorporating pilot scheduling to mitigate congestion and improve service through a Lagrangian relaxation algorithm. For pilot workforce management, Giachetti et al. [31] present MIP models for days-on and days-off scheduling, balancing workforce demand, labor requirements, and worker preferences, offering flexible scheduling with extended breaks, thereby improving fairness and job satisfaction.

Compared to studies that consider only one aspect, the number of studies simultaneously addressing both ship selection and inspection scheduling is currently limited. Yan et al. [14] develop an XGBoost model to predict the number of ship deficiencies, considering generic, dynamic, and historical inspection factors. Based on these predictions, a port state control officer scheduling model is proposed to optimize the allocation of inspection resources, aiming at maximizing the predicted total number of detected deficiencies. Qiao et al. [15] construct an IP model to solve the flag state control officer routing and scheduling problem. With the aim of maximizing the total risk weight of inspected ships within a limited budget and human resources, the model determines which ships need to be inspected and the visiting routes of inspectors to various ports. Yan et al. [16] first use a k-nearest neighbor model to predict ship deficiencies and then propose three optimization models to maximize the inspection benefit, which includes the reward for identifying the predicted deficiencies in the inspected ships and the penalty for inspectors' overtime pay. To address the emission control challenges, Luo et al. [17] propose a drone scheduling model for monitoring vessel emissions and design an ant colony algorithm to solve the problem. The model decides the selected ships and the schedule for drones according to the monitoring weight of ships generated from historical data.

To sum up, the existing literature mainly focuses on the ship selection problem or the inspection scheduling problem in maritime transportation. Only a few studies consider the integration of these two aspects. However, these studies primarily focus on the inspection of ocean-going ships, with relatively few applications in inland waterway transport. Yan et al. [14] and Yan et al. [16] consider the scheduling problem of PSC with multiple inspectors staying at the same port. Qiao et al. [15] consider a group of inspectors with the same schedule traveling between multiple ports in FSC. None of them consider the multi-inspector and multi-location scenario, which is suitable for inland waterway inspections. Additionally, Yan et al. [14], Qiao et al. [15], and Yan et al. [16] primarily use solvers to obtain exact solutions, which, although yielding optimal results, exhibit low efficiency when solving large-scale problems.

To address the gaps in existing research, we construct an IP model to solve the integrated ship selection and inspection scheduling problem (ISSISP) in inland waterway transport to maximize the total risk weight of inspected ships, using the solver to obtain the optimal solution and designing a heuristic algorithm to enhance solving efficiency. In Section 3, we present the problem formulation of the ISSISP, including the problem description and model formulation. Section 4 provides a detailed description of the proposed heuristic algorithm. In Section 5, numerical experiments and sensitivity analysis are carried out. In the end, Section 6 makes a conclusion for this study.

3. Problem Formulation

3.1. Problem Description

Considering the ISSISP within a segment of a river during the planning horizon of one day, merchant ships are inspected by government ships at candidate inspection locations. We denote the set of merchant ships by $\mathcal{V}$ and the set of government ships by $\mathcal{M}$. The river segment can be treated as a line, with a set of candidate inspection locations, denoted by $\mathcal{N}$, along its course. For all merchant ships navigating in the waterway throughout the day, there are four types of navigation directions: (1) downstream, (2) upstream, (3) downstream followed by upstream, and (4) upstream followed by downstream. As an example shown in Figure 1, if the direction to the right is considered as the downstream direction, then the sailing directions of merchant ships from v_1 to v_4 correspond to the navigation directions from (1) to (4).

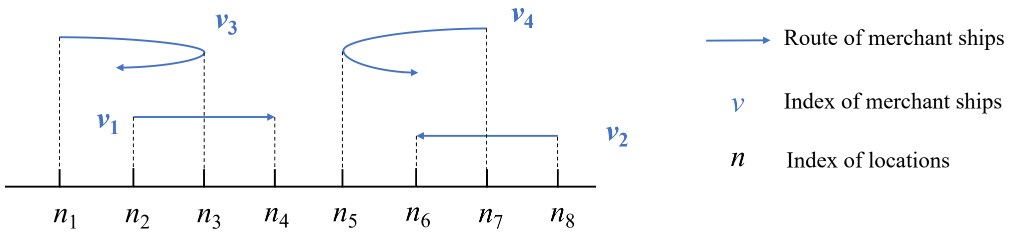


Figure 1. Original problem description diagram.

The fact that inland waterways are often congested and characterized by many sharp bends imposes limitations on the size of merchant ships navigating these routes. Each location $n \in \mathcal{N}$ is assigned a maximum ship size limit $SN_n, n \in \mathcal{N}$ for docking to ensure that the berthed ships do not impede the normal passage of other ships in the adjacent waters. If the size of merchant ship $SV_v, v \in \mathcal{V}$ passing through location n does not exceed the size limit SN_n, then the merchant ship v is permitted to dock at location n and undergo inspection; otherwise, it cannot be inspected at location n. We represent the berth availability of merchant ships by the binary parameter $\delta_v^n, n \in \mathcal{N}, v \in \mathcal{V}$. If merchant ship v can dock and be inspected at location n, then δ_v^n is set to 1; otherwise, it is 0.

We assume that more than one government ship can dock at each location. Each government ship is assigned to one location and remains docked at that location throughout the day, responsible for inspecting some merchant ships docking at that location. Government ships conduct inspections at the assigned locations during the day and adjust their positions at night based on the decisions for the next day. Considering the limitations on personnel efficiency, the maximum number of merchant ships that each government ship can inspect per day is K.

For ship risk assessment, the predicted deficiency number for each merchant ship, based on factors such as ship age, length, depth, last inspection date, company performance, and so on [14], can be normalized into a deficiency proportion. This proportion is defined as the risk weight, representing the ship's overall risk level. A higher predicted deficiency number corresponds to a higher risk weight, indicating a greater priority for inspection. Therefore, to ensure the safety of inland waterway navigation and environmental health, government ships aim to inspect as many high-weight merchant ships as possible. In order to improve the efficiency of the inspection system, the assignment decisions for government ships and the inspection decisions for merchant ships should be made wisely, with the aim of maximizing the total weight of merchant ships inspected by government ships.

3.2. Model Formulation

In this subsection, we first construct an IP model according to the problem setting. However, through the observation of the model, we find that there is symmetry in the model, which is defined as the symmetric IP (SIP) model, leading to redundant calculations in the process of solving it. Hence, we reformulate the SIP model into a more compact form, which is called the compact IP (CIP) model, in order to break the symmetry. The notations used in the two models are summarized in Table 1.

Table 1. Notations used in the model formulation.

	Sets
$\mathcal{V}$	The set of merchant ships
$\mathcal{M}$	The set of government ships
$\mathcal{N}$	The set of candidate inspection locations
	Indices
v	The index for merchant ships in $\mathcal{V}$
m	The index for government ships in $\mathcal{M}$
n	The index for candidate inspection locations in $\mathcal{N}$
	Parameters
w_v	The inspected weight of merchant ship v
δ_v^n	Binary, equals 1 if merchant ship v can be inspected at location n, and 0 otherwise
K	The maximum number of merchant ships inspected by each government ship
	Decision Variables
x_m^n	Binary, equals 1 if government ship m is assigned to location n, and 0 otherwise
y_v^m	Binary, equals 1 if merchant ship v is inspected by government ship m, and 0 otherwise
$\bar{x}^n$	The integer number of government ships assigned to location n
$\bar{y}_v^n$	Binary, equals 1 if merchant ship v is inspected at location n, and 0 otherwise

3.2.1. The SIP Model for the ISSISP

The SIP model is constructed to solve the ISSISP. For assignment decisions, we introduce the binary decision variable x_m^n, which equals 1 if the government ship m is assigned to location n and 0 otherwise. For inspection decisions, we use y_v^m as the binary decision variable, which equals 1 if merchant ship v is inspected by government ship m and 0 otherwise.

The SIP model of the ISSISP can be written as follows:

$$\max \quad \sum_{v\in\mathcal{V}}\sum_{m\in\mathcal{M}} w_v y_v^m \tag{1}$$

$$\text{s.t.} \quad y_v^m - \sum_{n\in\mathcal{N}} \delta_v^n x_m^n \leqslant 0, \qquad \forall v \in \mathcal{V}, \forall m \in \mathcal{M} \tag{2}$$

$$\sum_{v\in\mathcal{V}} y_v^m \leqslant K, \qquad \forall m \in \mathcal{M} \tag{3}$$

$$\sum_{m\in\mathcal{M}} y_v^m \leqslant 1, \qquad \forall v \in \mathcal{V} \tag{4}$$

$$\sum_{n\in\mathcal{N}} x_m^n = 1, \qquad \forall m \in \mathcal{M} \tag{5}$$

$$x_m^n \in \{0,1\}, \qquad \forall m \in \mathcal{M}, \forall n \in \mathcal{N} \tag{6}$$

$$y_v^m \in \{0,1\}, \qquad \forall m \in \mathcal{M}, \forall v \in \mathcal{V}. \tag{7}$$

The objective function (1) maximizes the total weight of the inspected merchant ships. Constraints (2) require that if merchant ship v and government ship m are not simultaneously at any location throughout the day, then merchant ship v cannot be inspected by government ship m. Constraints (3) denote that at most K merchant ships can be inspected by each government ship. Constraints (4) mean that each merchant ship can be inspected at most once. Constraints (5) guarantee that each government ship is assigned to one location. Constraints (6) and (7) are the decision variable constraints.

From the structure of the SIP model, we find that there exists symmetry in the set of indices of government ships due to their homogenization. For instance, if a scheduling decision can be expressed as assigning one government ship to location n and inspect ship v, the contribution to the objective function is the same regardless of which unassigned government ship will execute the schedule. These symmetrical solutions result in extended computational time [32]. An effective approach to tackle this issue is to utilize advanced models that either eliminate or reduce symmetries. Therefore, we build the CIP model to break the symmetry.

3.2.2. The CIP Model for the ISSISP

In the CIP model, we no longer introduce variables related to government ships but directly establish a link between merchant ships and locations. Specifically, we introduce the non-negative integer variable $\bar{x}^n$ as the number of government ships assigned to location n and use $\bar{y}_v^n$ as the binary decision variable, which equals 1 if merchant ship v is inspected at location n and 0 otherwise. Then, we have the following CIP model:

$$
\begin{aligned}
\max \quad & \sum_{v \in \mathcal{V}} \sum_{n \in \mathcal{N}} w_v \bar{y}_v^n && && (8)\\
\text{s.t.} \quad & \bar{y}_v^n \leqslant \delta_v^n && \forall v \in \mathcal{V}, \forall n \in \mathcal{N} && (9)\\
& \sum_{v \in \mathcal{V}} \bar{y}_v^n \leqslant K\bar{x}^n, && \forall n \in \mathcal{N} && (10)\\
& \sum_{n \in \mathcal{N}} \bar{y}_v^n \leqslant 1, && \forall v \in \mathcal{V} && (11)\\
& \sum_{n \in \mathcal{N}} \bar{x}^n = |\mathcal{M}| && && (12)\\
& \bar{x}^n \in \mathbb{Z}_+, && \forall n \in \mathcal{N} && (13)\\
& \bar{y}_v^n \in \{0,1\}, && \forall n \in \mathcal{N}, \forall v \in \mathcal{V}. && (14)
\end{aligned}
$$

The objective function (8) maximizes the total weight of inspected merchant ships. Constraints (9) require that if merchant ship v passed through location n, then merchant ship v cannot be inspected at location n. Constraints (10) mean that the total number of inspected merchant ships at location n is no more than the inspection capacity of location n. Constraints (11) denote that each merchant ship can be inspected at most once. Constraints (12) require that the total number of assigned government ships is $|\mathcal{M}|$. Constraints (13) and (14) are the decision variable constraints.

4. Weight-Induced Inspection Scheduling Algorithm

4.1. Algorithm Design

Based on our analysis of the problem, we have the following observation:

Observation 1. *There are variations in both the number and weights of merchant ships that dock at different locations. Among the ships passing through in a single day, some locations have a low total number of docked ships with small ship weights, while other locations have a high total number of docked ships with large ship weights.*

Inspired by Observation 1, we calculate the distribution of the sum of ship weights at different locations. Analyzing the distribution's variation, it is advisable to prioritize assigning government ships to locations with many high-weight merchant ships. This strategy ensures that, within the constraints of limited inspection capacity, a greater number of high-weight merchant ships can be inspected. Leveraging this unique problem characteristic, we design a tailored weight-induced inspection scheduling (WIS) algorithm to solve the IP model in Section 3.2. The algorithm prioritizes deploying government ships to locations frequented by high-weight merchant ships and iteratively removes inspected merchant ships until all government ships are dispatched.

To be more specific, we first define the set for merchant ships awaiting inspection as $\overline{\mathcal{V}}$, and denote by $\overline{\mathcal{M}}$ the set of unassigned government ships. In each iteration, an unassigned government ship $m^* \in \overline{\mathcal{M}}$ is dispatched to a location $n \in \mathcal{N}$ and is assigned to inspect no more than K merchant ships at location n. We represent by $\overline{\mathcal{V}}_n$ the set of uninspected merchant ships passing through location n and denote by $\overline{\mathcal{V}}_n^K$ the set of uninspected merchant ships with top K weights passing through location n. If the number of uninspected merchant ships docking at location n is smaller than K, then $\overline{\mathcal{V}}_n^K = \overline{\mathcal{V}}_n$. For each location n, u_n is defined as the sum of the weights of merchant ships with top K weights among all uninspected merchant ships passing through location n, i.e.,

$$u_n = \sum_{v \in \overline{\mathcal{V}}_n^K} w_v. \tag{15}$$

A location with the highest u_n, denoted by n^* as shown in Equation (16), is selected for deploying the government ship m^*. Note that there may be multiple locations with the same highest value of u_n, in which case we randomly select one of them as the assigned location n^*.

$$n^* \in \operatorname*{argmax}_{n \in \mathcal{N}} \{u_n | \forall n \in \mathcal{N}\}. \tag{16}$$

After deciding the assigned location n^*, the government ship inspects the merchant ships with the top K weights at location n^* and the set of inspected merchant ships can be expressed as $\mathcal{V}_{n^*}^K$. Subsequently, the inspected merchant ships are removed from the set of uninspected merchant ships, i.e., $\overline{\mathcal{V}} = \overline{\mathcal{V}} \setminus \overline{\mathcal{V}}_{n^*}^K$, and the assigned government ship m^* is removed from the set of unassigned government ships, i.e., $\overline{\mathcal{M}} = \overline{\mathcal{M}} \setminus \{m^*\}$. Then, we design the schedule for the next unassigned government ship. If there is still any unassigned government ship (i.e., the set $\overline{\mathcal{M}}$ is not empty), the process continues iteratively. Otherwise, the algorithm terminates. Algorithm 1 shows the detailed procedure of the proposed WIS algorithm.

Algorithm 1 Weight-induced Inspection Scheduling Algorithm

```
Input: $\mathcal{V}$: the set of merchant ships; $\mathcal{M}$: the set of government ships; $\mathcal{N}$: the set of locations;
    $w_v$: weights of merchant ships; $\delta_v^n$: berth availability of merchant ships; $K$: inspection
    capacity
Output: scheduling dictionary $D$
Initial uninspected merchant ship set $\overline{\mathcal{V}} \leftarrow \mathcal{V}$, unassigned government ship set $\overline{\mathcal{M}} \leftarrow$
    $\mathcal{M}$ and current government ship $m^* \leftarrow 1$
while $m^* \leqslant |\mathcal{M}|$ do
    Update location uninspected ship set $\overline{\mathcal{V}}_n \leftarrow \{v | \delta_v^n = 1, \forall v \in \overline{\mathcal{V}}\}$
    if $|\overline{\mathcal{V}}_n| > K$ then
        Update $\overline{\mathcal{V}}_n^K \leftarrow$ uninspected merchant ships in $\overline{\mathcal{V}}_n$ with top $K$ weights
    else
        Update $\overline{\mathcal{V}}_n^K \leftarrow \overline{\mathcal{V}}_n$
    end if
    Compute the sum of top $K$ weights at location $n$, $u_n \leftarrow \sum_{v \in \overline{\mathcal{V}}_n^K} w_v$
    Update assigned location $n^* \leftarrow \operatorname*{argmax}_{n \in \mathcal{N}} \{u_n | \forall n \in \mathcal{N}\}$
    Update $D[m^*, n^*] \leftarrow \overline{\mathcal{V}}_{n^*}^K$
    $m^* \leftarrow m^* + 1$
end while
```

4.2. Example Illustration

In order to facilitate the understanding of the algorithm, this subsection shows the iterative process of the WIS algorithm based on a specific example. The information of merchant ships can be represented in the form of a two-dimensional table, including the berth availability and weight of each merchant ship, as shown in Table 2. There are five merchant ships ($\mathcal{V} = \{v_1, v_2, v_3, v_4, v_5\}$), two government ships ($\mathcal{M} = \{m_1, m_2\}$) with the inspection capacity of two merchant ships per day ($K = 2$), and four locations ($\mathcal{N} = \{n_1, n_2, n_3, n_4\}$) in the example. "Ship ID" includes the ID for all the ships. "Berth availability" includes the berthing information of each ship in one day. If merchant ship v docks at location n, then it is marked with 1 in the "Berth availability" column. "Weight" corresponds to the risk weight of each merchant ship. "u_n" is the sum of the top K weights of uninspected merchant ships passing through location n.

Table 2. Uninspected merchant ship information in the first iteration of the example.

Ship ID	Berth Availability				Weight
	n_1	n_2	n_3	n_4	
v_1	1	1		1	0.13
v_2		1		1	0.96
v_3		1	1	1	0.24
v_4			1		0.31
v_5			1	1	0.62
u_n	0.13	1.20	0.93	1.58	

At the beginning of the algorithm, the uninspected merchant ship set $\overline{\mathcal{V}}$ is initialized as $\mathcal{V}$, and the unassigned government ship set $\overline{\mathcal{M}}$ is set to be $\mathcal{M}$. In the first iteration, we make the schedule decisions for m_1. First, we can derive the set of merchant ships with the top two largest weights $\overline{\mathcal{V}}_n^K$ for each location (Equations (17)–(20)).

$$\overline{\mathcal{V}}_1^K = \{v_1\}, \tag{17}$$

$$\overline{\mathcal{V}}_2^K = \{v_2, v_3\}, \tag{18}$$

$$\overline{\mathcal{V}}_3^K = \{v_4, v_5\}, \tag{19}$$

$$\overline{\mathcal{V}}_4^K = \{v_2, v_5\}. \tag{20}$$

According to Equation (15), u_n is calculated as the sum of merchant ships with the top K weights passing through location n. That is, for each location, we find merchant ships with the top two largest weights among the merchant ships with 1 marked in the "Berth availability" column and sum their weights up as the value of u_n. If the number of merchant ships with 1 marked in the "Berth availability" column is smaller than two, e.g., n_1, then u_n is the weight of v_1. Then, we can calculate u_n for each location and choose a location with the highest u_n, which is n_4. Accordingly, the set of merchant ships to be inspected is $\overline{\mathcal{V}}_4^K$. Thus, we assign the first government ship m_1 to location n_4 to inspect v_2 and v_5.

After the assignment of m_1, v_2 and v_5 are removed from the uninspected ship set, then we have $\overline{\mathcal{V}} = \{v_1, v_3, v_4\}$. Similar to the above process, we can derive $\overline{\mathcal{V}}_n^K$ as follows:

$$\overline{\mathcal{V}}_1^K = \{v_1\}, \tag{21}$$

$$\overline{\mathcal{V}}_2^K = \{v_1, v_3\}, \tag{22}$$

$$\overline{\mathcal{V}}_3^K = \{v_3, v_4\}, \tag{23}$$

$$\overline{\mathcal{V}}_4^K = \{v_1, v_3\}. \tag{24}$$

In the second iteration, n_3 has the largest u_n among four locations as shown in Table 3. Therefore, we assign m_2 to n_3 to inspect v_3 and v_4. To sum up, the total weight of the inspected merchant ships is 2.13.

Table 3. Uninspected merchant ship information in the second iteration of the example.

Ship ID	Berth Availability				Weight
	n_1	n_2	n_3	n_4	
v_1	1	1		1	0.13
v_3		1	1	1	0.24
v_4			1		0.31
u_n	0.13	0.37	0.55	0.37	

5. Numerical Experiments

In this section, we conduct numerical experiments to evaluate the performance of the three methods, i.e., the SIP model, the CIP model, and the proposed WIS algorithm. Meanwhile, sensitivity analysis is conducted to quantify the impact of uncertainties. All experiments are implemented in Python 3.9, using Gurobi 9.5.2 [33] for solving the SIP model and the CIP model. The experiments are conducted on a MacBook Air equipped with an Apple M2 chip, 16 GB of RAM, and the macOS operating system. The parameters of the models are calibrated in preliminary studies and their final values are presented in the following sections.

5.1. Experimental Design

Taking the Yangtze River as an example, we select twenty locations ($|\mathcal{M}| = 20$) along the river as candidate inspection locations. We focus on the river segment that starts from Chenglingji, flows downstream through intermediate locations, and finally ends at Shanghai. Table 4 presents the distances from Chenglingji to each location, with data sourced from the Yangzhou Navigation Center [34]. Given that the river can be regarded as a line and the locations as points along this line, the distances between any two locations can be calculated based on Table 4.

According to the regulation of the Ministry of Transport [35], the maximum speed for inland waterway ships can not exceed 28 km per hour. Therefore, the speed of each merchant ship, whose unit is kilometers per hour, follows a uniform distribution within the parameter range of $[8, 15]$. The maximum ship size limit SN_n of each location is randomly chosen from the set $\{60, 70, 80, 90, 100\}$ with equal probability. The size of each merchant ship SV_v is a randomly generated integer within the range $[50, 100]$. Based on the parameters set above, we construct the berth availability for each merchant ship, ensuring that there exists at least one location where ship v can dock, i.e., $\sum_{n \in \mathcal{N}} \delta_n^v \geqslant 1, \forall v \in \mathcal{V}$.

Table 4. The distance from Chenlingji to downstream locations.

Location ID	Destination	Distance (km)	Location ID	Destination	Distance (km)	Location ID	Destination	Distance (km)
1	Chenglingji	0	8	Anqing	626	15	Gaogang	1042
2	Honghu	56	9	Chizhou	684	16	Jiangyin	1104
3	Jiayu	101	10	Tongling	724	17	Zhangjiagang	1140
4	Wuhan	228	11	Wuhu	818	18	Nantong	1162
5	Huanggang	325	12	Ma'anshan	861	19	Changshu	1191
6	Huangshi	357	13	Nanjing	907	20	Shanghai	1257
7	Jiujiang	473	14	Zhenjiang	987			

The experimental parameters are listed in Table 5. $[a, b, c]$ denotes a list of numbers generated from a to b with an interval of c. The first 60 experiments are conducted to test the performance of the three methods on different scales of the problem. The experiments with ID from 61 to 150 conduct the sensitivity analysis. To be more specific, the first group (ID from 61 to 90) is used to test the impact of the location number $|\mathcal{N}|$ on the problem results. The second group (ID from 91 to 120) varies in terms of the number of government ships $|\mathcal{M}|$ and the last group (ID from 121 to 150) has variations in the number of merchant ships $|\mathcal{V}|$.

5.2. Performance of the Three Methods

5.2.1. Solution Quality

To test the performance of the three methods, i.e., the SIP model solved by Gurobi, the CIP model solved by Gurobi, and the WIS algorithm, we design instances with different merchant ship numbers ranging from 50 to 1000. The three methods are applied to solve these instances and the results regarding the solution quality are shown in Table 6 and Figure 2.

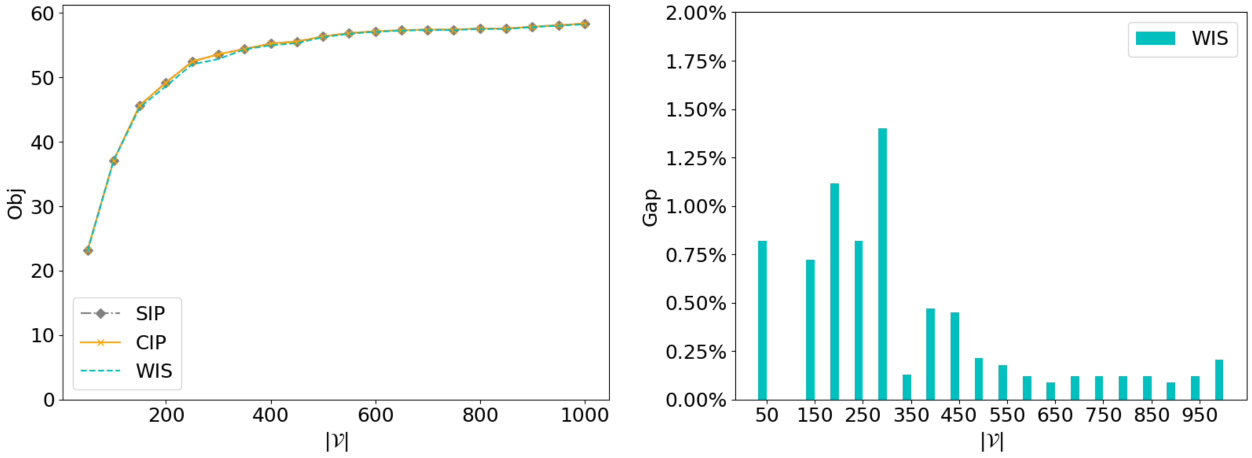
Table 5. Experiment parameter settings.

Experiment ID	$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{M}\rvert$	$\lvert\mathcal{V}\rvert$	K	Method
1–20	20	10	[50, 1000, 50]	6	SIP
21–40	20	10	[50, 1000, 50]	6	CIP
41–60	20	10	[50, 1000, 50]	6	WIS
61–70	[10, 19, 1]	10	500	6	SIP
71–80	[10, 19, 1]	10	500	6	CIP
81–90	[10, 19, 1]	10	500	6	WIS
91–100	20	[1, 10, 1]	500	6	SIP
101–110	20	[1, 10, 1]	500	6	CIP
111–120	20	[1, 10, 1]	500	6	WIS
121–130	20	10	500	[1, 10, 1]	SIP
131–140	20	10	500	[1, 10, 1]	CIP
141–150	20	10	500	[1, 10, 1]	WIS

Table 6. Solution quality of the three methods for instances with different scales.

$\lvert\mathcal{V}\rvert$	**SIP**		**CIP**		**WIS**		
	# Inspected Ships	**Obj**	**# Inspected Ships**	**Obj**	**# Inspected Ships**	**Obj**	**Gap**
50	48	23.17	48	23.17	45	22.98	0.82%
100	60	37.11	60	37.11	60	37.11	0.00%
150	60	45.68	60	45.68	60	45.35	0.72%
200	60	49.19	60	49.19	60	48.64	1.12%
250	60	52.45	60	52.45	60	52.02	0.82%
300	60	53.59	60	53.59	60	52.84	1.40%
350	60	54.41	60	54.41	60	54.34	0.13%
400	60	55.23	60	55.23	60	54.97	0.47%
450	60	55.57	60	55.57	60	55.32	0.45%
500	60	56.35	60	56.35	60	56.23	0.21%
550	60	56.87	60	56.87	60	56.77	0.18%
600	60	57.15	60	57.15	60	57.08	0.12%
650	60	57.31	60	57.31	60	57.26	0.09%
700	60	57.41	60	57.41	60	57.34	0.12%
750	60	57.41	60	57.41	60	57.34	0.12%
800	60	57.58	60	57.58	60	57.51	0.12%
850	60	57.58	60	57.58	60	57.51	0.12%
900	60	57.85	60	57.85	60	57.80	0.09%
950	60	58.10	60	58.10	60	58.03	0.12%
1000	60	58.34	60	58.34	60	58.22	0.21%
525	59.40	52.92	59.40	52.92	59.25	52.73	0.37%

In Table 6, the first column records the number of merchant ships ($\lvert\mathcal{V}\rvert$) for different instances. "# Inspected ships" represents the number of inspected ships in the solution and "Obj" is the objective value, i.e., the total weight of inspected ships. Since both the SIP and CIP models are solved to optimality using Gurobi, the Obj records the optimal objective value of the problem. However, the WIS algorithm is a heuristic algorithm, so the solution obtained is not necessarily an optimal one. To measure the difference between the Obj obtained by WIS and the optimal Obj, we define it as the "Gap" and calculate it by $\text{Gap} = (\text{Obj}_S - \text{Obj}_W)/\text{Obj}_S$, where Obj_S is the Obj of SIP and Obj_W is the Obj of WIS. The last row in the table records the average value of each column.



(**a**) (**b**)

Figure 2. Solution quality of the three methods for instances with different scales. (**a**) Obj of instances solved by the three methods. (**b**) Optimality gap of instances solved by the WIS algorithm.

Figure 2a illustrates the Obj of instances solved by the three methods: SIP (gray dashed line with diamond markers), CIP (orange solid line with cross markers), and WIS (blue dashed line with circular markers). The x-axis represents the instance size ($|\mathcal{V}|$), ranging from 50 to 1000, while the y-axis represents Obj, ranging from 0 to 60. The graph shows that the Obj increases with the increase in the problem scale, i.e., the number of merchant ships. When the scale of the problem is small, the increasing rate of the Obj is rapid, whereas, when the scale of the problem is large, the increasing rate gradually slows down. For the instances with $|\mathcal{V}|$ no more than 300, there are some differences between the curves of the three methods, while, for the instances with $|\mathcal{V}|$ larger than 300, the three curves have a high degree of overlap, indicating a similar performance across the three methods when dealing with large-scale instances. Combined with the specific numerical results in Table 6, we can see that the Obj values of SIP and CIP are identical for the same instance. This is because both models can be solved to optimality by Gurobi. However, the performance of WIS is slightly inferior to the optimal solution in some instances. The Obj of WIS has an average gap of 0.37% from the optimal Obj.

The number of inspected ships in Table 6 can also provide an analytical perspective on the features of the solution. When the number of merchant ships equals 50, the number of inspected ships in the optimal solution is 48, while it equals 45 in the heuristic solution. This indicates that the WIS algorithm cannot inspect a sufficient number of ships, thus resulting in a smaller value of the Obj. In other instances, the number of ships inspected by the three methods reached the overall inspection capacity limit, i.e., $|\mathcal{M}| \times K = 10 \times 6 = 60$, but the weights of the ships selected by SIP and CIP are generally larger compared to WIS. Among all the 20 instances, there exists one instance ($|\mathcal{V}| = 100$) where the Obj of WIS equals the optimal one. For the other instances, there is a certain gap between the optimal Obj and the Obj of WIS.

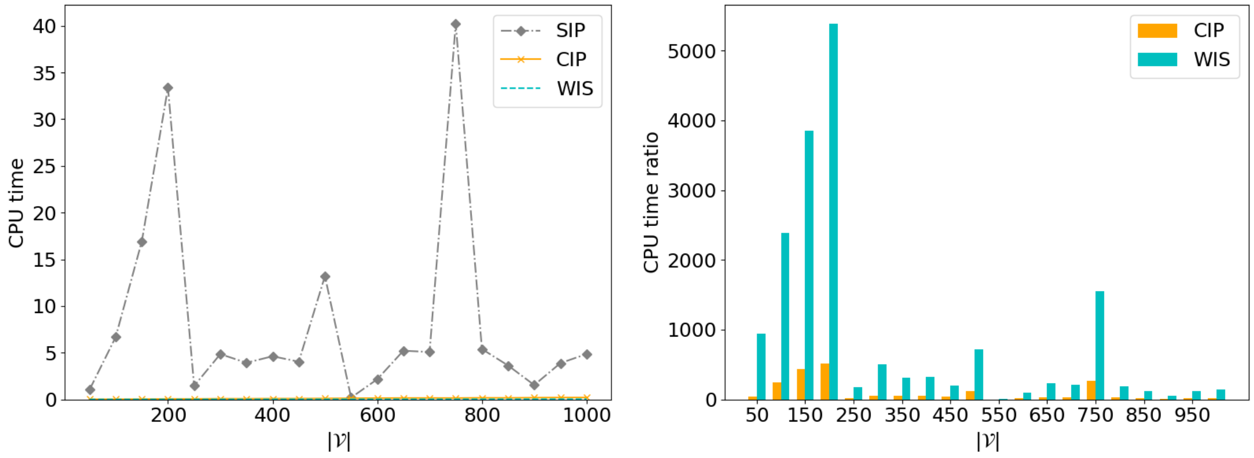
The gaps for all the instances are presented in a bar chart in Figure 2b, which depicts the gap of instances solved by the WIS algorithm, as a function of the instance size. The x-axis represents the instance size, ranging from 50 to 1000, and the y-axis represents the gap, ranging from 0.00% to 2.00%. The bar in blue records the gap of the WIS algorithm. In small-scale instances, we observe a larger gap compared to large-scale instances, due to the identical number of locations set for these instances. When the number of merchant ships $|\mathcal{V}|$ is relatively low, there is a significant difference in the sum of the top K weights of merchant ships at different locations. However, when $|\mathcal{V}|$ is sufficiently high, the difference in the sum of the top K weights becomes smaller. Consequently, the gap between the Obj obtained from the WIS algorithm and the optimal Obj is relatively small.

5.2.2. Computation Time

The computation time of the three methods is recorded in Figure 3 and Table 7. In Table 7, "CPU time" records the time taken by the solver or the algorithm to solve the problem. "CPU time ratio" is the ratio of the CPU time of CIP or WIS to the CPU time of SIP. $\mathrm{CTR}_C = \mathrm{CT}_S/\mathrm{CT}_C$, where CTR_C is the CPU time ratio of CIP, CT_S is the CPU time of SIP, and CT_C is the CPU time of CIP. Similarly, $\mathrm{CTR}_W = \mathrm{CT}_S/\mathrm{CT}_W$, where CTR_W is the CPU time ratio of WIS and CT_W is the CPU time of WIS. The last row in the table records the average value of each column.

Table 7. Computation time of the three methods for problems of different scales.

$\lvert\mathcal{V}\rvert$	SIP	CIP		WIS	
	CPU Time (s)	CPU Time (s)	CPU Time Ratio	CPU Time (s)	CPU Time Ratio
50	1.0351	0.0231	44.81	0.0011	941.00
100	6.6785	0.0274	243.74	0.0028	2385.18
150	16.9242	0.0385	439.59	0.0044	3846.41
200	33.3847	0.0655	509.69	0.0062	5384.63
250	1.4555	0.0609	23.90	0.0081	179.69
300	4.8595	0.0830	58.55	0.0097	500.98
350	3.9341	0.0807	48.75	0.0124	317.27
400	4.6290	0.0932	49.67	0.0143	323.71
450	4.0441	0.0943	42.89	0.0202	200.20
500	13.2110	0.1079	122.44	0.0183	721.91
550	0.1996	0.1264	1.58	0.0196	10.18
600	2.1667	0.1401	15.47	0.0210	103.18
650	5.2260	0.1537	34.00	0.0223	234.35
700	5.0809	0.1555	32.67	0.0235	216.21
750	40.2412	0.1489	270.26	0.0259	1553.71
800	5.4011	0.1733	31.17	0.0280	192.90
850	3.6400	0.1664	21.88	0.0290	125.52
900	1.5481	0.1904	8.13	0.0316	48.99
950	3.8917	0.2064	18.86	0.0330	117.93
1000	4.8689	0.2075	23.46	0.0348	139.91
525	8.1210	0.1172	102.07	0.0183	877.19



(**a**)

(**b**)

Figure 3. Computation time of the three methods for instances with different scales. (**a**) CPU time of instances solved by the three methods. (**b**) CPU time ratio of instances solved by CIP and WIS.

Figure 3a shows the CPU time of instances with different scales solved by the three methods: SIP (gray dashed line with diamond markers), CIP (orange solid line with cross markers), and WIS (blue dashed line with circular markers). The x-axis represents the instance size, ranging from 50 to 1000, while the y-axis represents the CPU time in seconds. Figure 3b depicts the CPU time ratio of instances solved by the CIP and WIS methods. The x-axis represents the instance size and the y-axis represents the CPU time ratio. The blue bars represent the WIS method and the orange bars represent the CIP method.

From Figure 3a and Table 7, we can observe that the CPU time taken to solve the SIP model using Gurobi fluctuates significantly across different problem scales, with an average value of 8.1210 s. In comparison, the CPU time for CIP and WIS can be substantially reduced, with the former averaging at 0.1172 s and the latter at 0.0183 s as shown in Table 7. To quantify the improvement in computational efficiency more distinctly, we calculate the increased ratio of CPU time of CIP and WIS relative to SIP.

Figure 3b visually demonstrates the CPU time ratio for CIP and WIS, highlighting a significant acceleration effect in small-scale instances. The maximum CPU time ratio for CIP reaches 509.69, whereas, for WIS, it reaches 5384.63. Across all instances, the average CPU time ratio for CIP is 102.07, while, for WIS, it is 877.19. This indicates that, although CIP is efficient, WIS shows even greater efficiency in terms of CPU time for the majority of instance sizes.

5.3. Sensitivity Analysis

By conducting sensitivity analysis experiments, we further analyze the sensitivity of the methods to the changes in input parameters, including the location number, the government ship number, and the inspection capacity.

5.3.1. Different Location Numbers

To investigate the impact of varying location numbers on the model's effectiveness, we design instances by altering the location number from 10 to 19 while keeping the government ship number ($|\mathcal{M}| = 10$), the merchant ship number ($|\mathcal{V}| = 500$), and the inspection capacity ($K = 6$) constant.

Figure 4a shows the Obj of instances with different location numbers solved by the three methods. The x-axis represents the number of locations, ranging from 10 to 19, and the y-axis represents Obj. Figure 4b illustrates the gap of instances with different numbers of locations solved by the WIS algorithm. Table 8 shows the solution quality of the three methods.

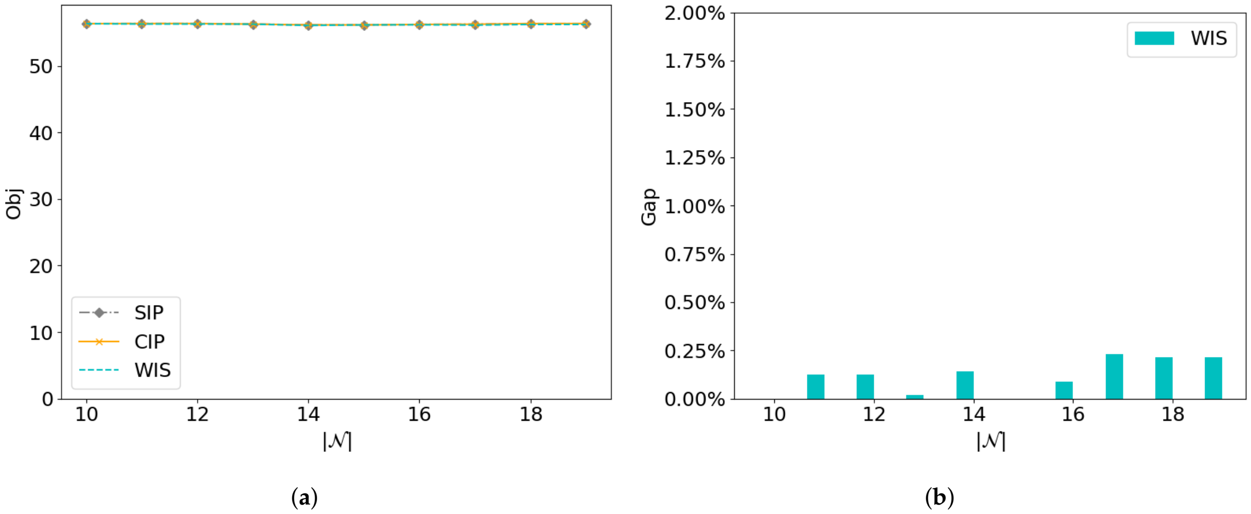


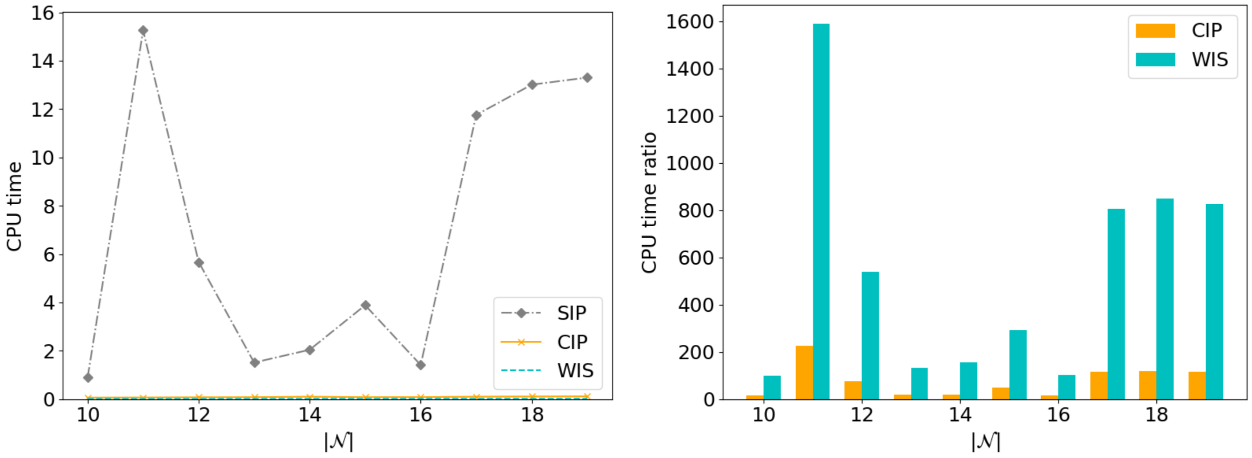
Figure 4. Solution quality of the three methods for instances with different location numbers. (**a**) Obj of instances solved by the three methods. (**b**) Optimality gap of instances solved by the WIS algorithm.

Table 8. Solution quality of the three methods for instances with different location numbers.

$\|\mathcal{N}\|$	SIP		CIP		WIS		
	# Inspected Ships	**Obj**	**# Inspected Ships**	Obj	**# Inspected Ships**	**Obj**	**Gap**
10	60	56.34	60	56.34	60	56.34	0.00%
11	60	56.36	60	56.36	60	56.29	0.12%
12	60	56.36	60	56.36	60	56.29	0.12%
13	60	56.28	60	56.28	60	56.27	0.02%
14	60	56.16	60	56.16	60	56.08	0.14%
15	60	56.16	60	56.16	60	56.16	0.00%
16	60	56.23	60	56.23	60	56.18	0.09%
17	60	56.26	60	56.26	60	56.13	0.23%
18	60	56.35	60	56.35	60	56.23	0.21%
19	60	56.35	60	56.35	60	56.23	0.21%
14.5	60	56.29	60	56.29	60	56.22	0.12%

From the perspective of solution quality, we can know that, as the location number continuously increases, the values of the Obj of the three methods all remain at a stable level as shown in Figure 4a. This indicates that the location number does not significantly affect the value of the Obj, i.e., the total weight of inspected ships. The graph also demonstrates that the objective values remain consistently close across all three methods, regardless of the number of locations. More results from Table 8 reveal that the average Obj of SIP and CIP are both 56.29, while, for WIS, the average Obj is 56.22, with only a small difference of 0.07. The average gap of WIS is 0.12%, which further illustrates that the Obj obtained by WIS has a very small difference compared to the optimal Obj obtained by SIP and CIP as shown in Figure 4b.

Figure 5a illustrates the CPU time required to solve instances with different location numbers using the three methods. The x-axis represents the number of locations and the y-axis represents the CPU time for solving the model in seconds. Figure 5b illustrates the CPU time ratio required to solve instances with different location numbers using the three methods. The x-axis represents the number of locations and the y-axis represents the CPU time ratio. Table 9 records the computation time of the three methods for instances with different location numbers.



(**a**) (**b**)

Figure 5. Computation time of the three methods for instances with different location numbers. (**a**) CPU time of instances solved by the three methods. (**b**) CPU time ratio of instances solved by CIP and WIS.

Table 9. Computation time of the three methods for instances with different location numbers.

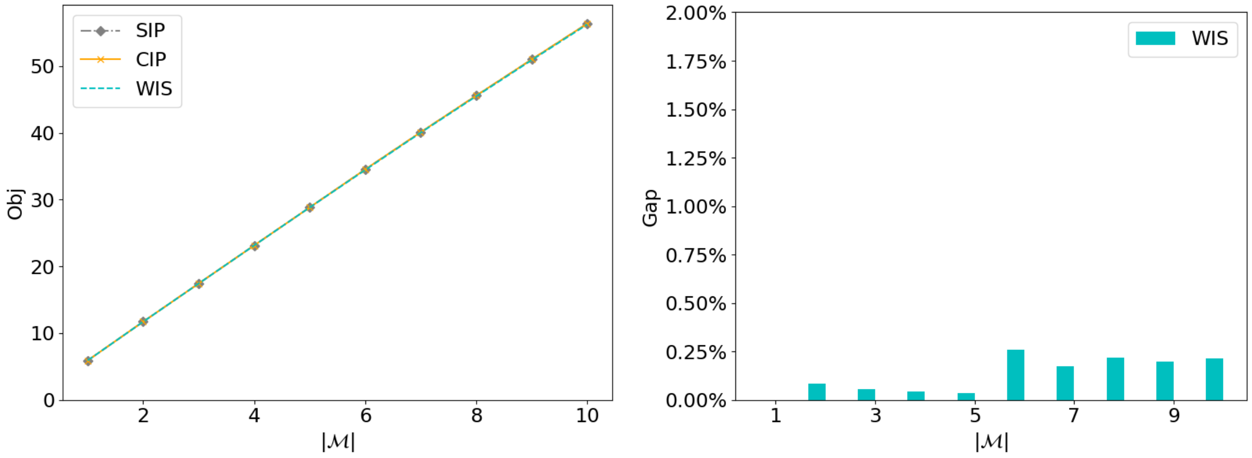
$\lvert\mathcal{N}\rvert$	SIP	CIP		WIS	
	CPU Time (s)	CPU Time (s)	CPU Time Ratio	CPU Time (s)	CPU Time Ratio
10	0.8950	0.0626	14.30	0.0090	99.44
11	15.2549	0.0678	225.00	0.0096	1589.05
12	5.6741	0.0763	74.37	0.0105	540.39
13	1.5139	0.0824	18.37	0.0113	133.97
14	2.0420	0.1015	20.12	0.0131	155.88
15	3.8844	0.0814	47.72	0.0133	292.06
16	1.4118	0.0846	16.69	0.0140	100.84
17	11.7555	0.1018	115.48	0.0146	805.17
18	13.0080	0.1098	118.47	0.0153	850.20
19	13.3006	0.1147	115.96	0.0161	826.12
14.5	6.8740	0.0883	76.65	0.0127	539.31

From the perspective of computation time, Figure 5a shows that the SIP method generally requires significantly more CPU time compared to CIP and WIS. Both CIP and WIS exhibit much lower and relatively stable CPU times across different numbers of locations, indicating their computational efficiency. In comparison, both values of and fluctuations in the CPU time of CIP and WIS are smaller. From Figure 5b, it can be observed that WIS takes less time to solve than SIP and even performs better than CIP. Additionally, Table 9 shows that, when $N = 11$, WIS achieves the maximum CPU time ratio with a value of 1589.05, indicating significantly improves the solution efficiency compared to SIP.

5.3.2. Variable Numbers of Government Ships

To test the influence of variable government ship numbers, we set $|\mathcal{N}| = 20$, $|\mathcal{V}| = 500$, and $K = 6$. The government ship number ranges from 1 to 10, reflecting different amounts of human resources for inspection.

Figure 6a shows the Obj of instances with different government ship numbers solved by the three methods. The x-axis represents the number of government ships, ranging from 1 to 10, and the y-axis represents Obj. Figure 6b illustrates the gap of instances with different numbers of government ships solved by the WIS algorithm. Table 10 shows the solution quality of the three methods for instances with different government ship numbers.



(**a**) (**b**)

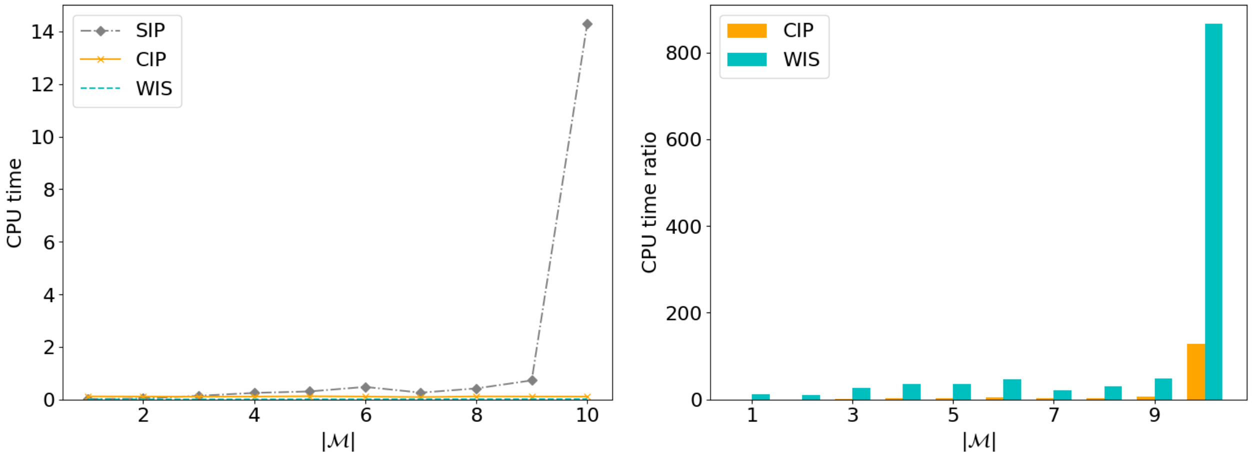
Figure 6. Solution quality of the three methods for instances with different government ship numbers. (**a**) Obj of instances solved by the three methods. (**b**) Optimality gap of instances solved by the WIS algorithm.

Table 10. Solution quality of the three methods for instances with different government ship numbers.

$\|\mathcal{M}\|$	SIP		CIP		WIS		
	# Inspected Ships	Obj	# Inspected Ships	Obj	# Inspected Ships	Obj	Gap
1	6	5.90	6	5.90	6	5.90	0.00%
2	12	11.71	12	11.71	12	11.70	0.09%
3	18	17.43	18	17.43	18	17.42	0.06%
4	24	23.14	24	23.14	24	23.13	0.04%
5	30	28.83	30	28.83	30	28.82	0.03%
6	36	34.52	36	34.52	36	34.43	0.26%
7	42	40.09	42	40.09	42	40.02	0.17%
8	48	45.57	48	45.57	48	45.47	0.22%
9	54	50.99	54	50.99	54	50.89	0.20%
10	60	56.35	60	56.35	60	56.23	0.21%
5.5	33	31.45	33	31.45	33	31.40	0.13%

Figure 6a shows that the Obj continues to become larger with the increase in the government ship number. Table 10 includes the number of inspected ships, from which we can know that the solution of each instance reaches the upper limit of the inspection capacity. This is due to the fact that the number of merchant ships ($|\mathcal{V}| = 500$) is large enough that the total number of merchant ships inspected by each government ship can reach the inspection capacity limit K. The gap of WIS is still small as shown in Figure 6b, with the average gap of 0.13%.

Figure 7a illustrates the CPU time required to solve instances with different government ship numbers using the three methods. The x-axis represents the number of government ships and the y-axis represents the CPU time for solving the model in seconds. Figure 7b illustrates the CPU time ratio of instances solved by the CIP and WIS methods. The x-axis represents the government ship number and the y-axis represents the CPU time ratio. Table 11 records the computation time of the three methods for instances with different government ship numbers.



(**a**) (**b**)

Figure 7. Computation time of the three methods for instances with different government ship numbers. (**a**) CPU time of instances solved by the three methods. (**b**) CPU time ratio of instances solved by CIP and WIS.

Table 11. Computation time of the three methods for instances with different government ship numbers.

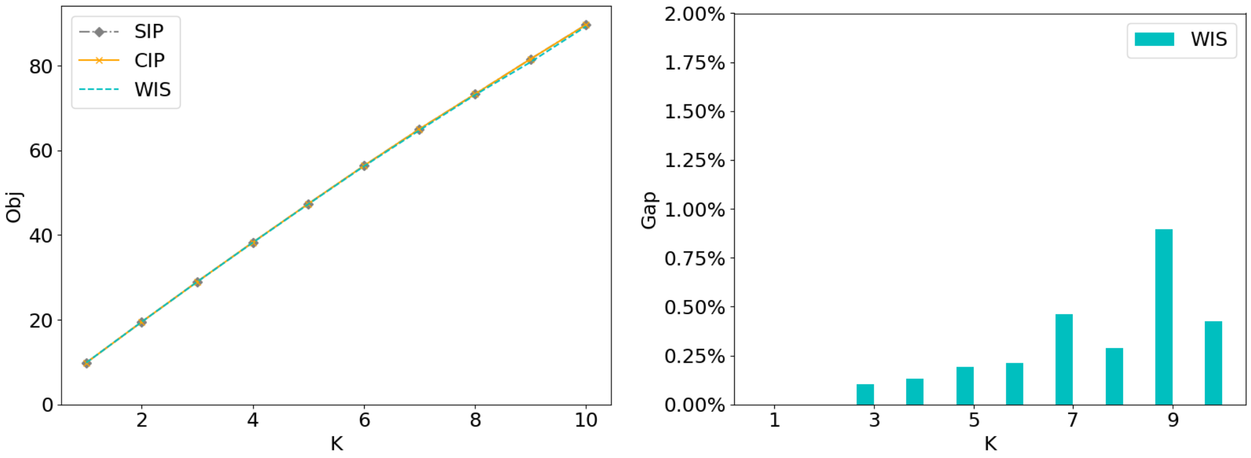
$\lvert\mathcal{M}\rvert$	SIP	CIP		WIS	
	CPU Time (s)	CPU Time (s)	CPU Time Ratio	CPU Time (s)	CPU Time Ratio
1	0.0200	0.1179	0.17	0.0017	11.76
2	0.0375	0.1131	0.33	0.0037	10.14
3	0.1407	0.1073	1.31	0.0051	27.59
4	0.2501	0.1121	2.23	0.0069	36.25
5	0.3067	0.1243	2.47	0.0085	36.08
6	0.4788	0.1121	4.27	0.0102	46.94
7	0.2604	0.0935	2.79	0.0122	21.34
8	0.4201	0.1180	3.56	0.0138	30.44
9	0.7257	0.1144	6.34	0.0151	48.06
10	14.2978	0.1117	128.00	0.0165	866.53
5.5	1.6938	0.1124	15.06	0.0094	180.77

For the computation time of the three methods, from Figure 7a and Table 11, we can see that the CPU time of these three methods is less than one second when $\lvert\mathcal{M}\rvert$ varies from 1 to 9. However, the CPU time of instance with $\lvert\mathcal{M}\rvert = 10$ increases to 14.2978 s. Accordingly, CIP and WIS do not accelerate the solving process significantly in the first nine instances as shown in Figure 7b, but they are outstanding in the last one, with the CPU time ratio of CIP equaling 128.00 and the CPU time ratio of WIS equaling 866.53.

5.3.3. Divergent Inspection Capacities

To find out the influence of divergent inspection capacities, we set $\lvert\mathcal{N}\rvert = 20$, $\lvert\mathcal{M}\rvert = 10$, and $\lvert\mathcal{V}\rvert = 500$. The inspection capacity K ranges from 1 to 10.

Figure 8a shows the Obj of instances with different inspection capacities solved by the three methods. The x-axis represents the inspection capacity, ranging from 1 to 10, and the y-axis represents Obj. Figure 8b illustrates the gap of instances with different inspection capacities solved by the WIS algorithm. Table 12 shows the solution quality of the three methods for instances with different inspection capacities.



(**a**) (**b**)

Figure 8. Solution quality of the three methods for instances with different inspection capacities. (**a**) Obj of instances solved by the three methods. (**b**) Optimality gap of instances solved by the WIS algorithm.

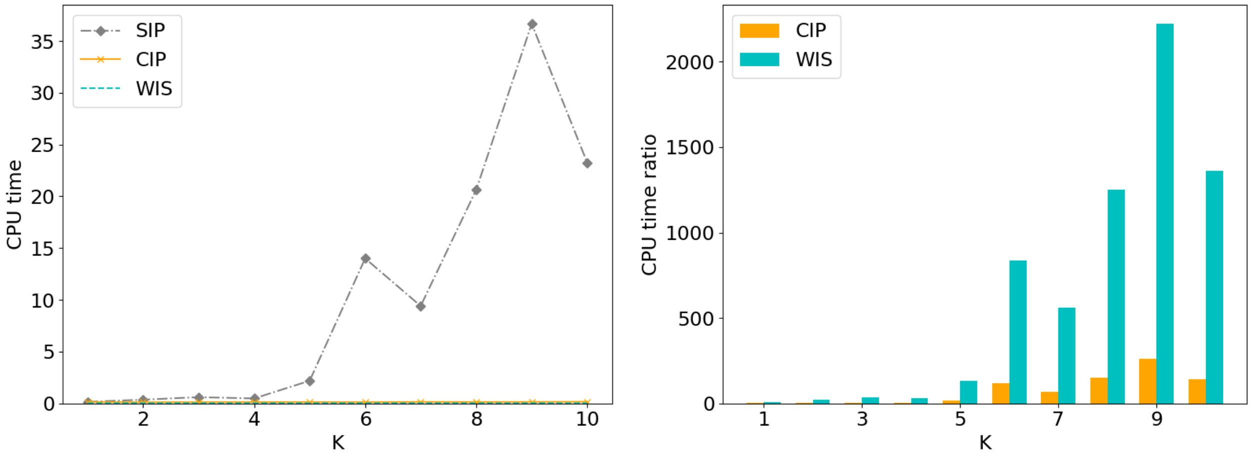
Table 12. Solution quality of the three methods for instances with different inspection capacities.

K	SIP		CIP		WIS		
	# Inspected Ships	Obj	# Inspected Ships	Obj	# Inspected Ships	Obj	Gap
1	10	9.83	10	9.83	10	9.83	0.00%
2	20	19.49	20	19.49	20	19.49	0.00%
3	30	28.98	30	28.98	30	28.95	0.10%
4	40	38.26	40	38.26	40	38.21	0.13%
5	50	47.38	50	47.38	50	47.29	0.19%
6	60	56.35	60	56.35	60	56.23	0.21%
7	70	65.03	70	65.03	70	64.73	0.46%
8	80	73.28	80	73.28	80	73.07	0.29%
9	90	81.56	90	81.56	90	80.83	0.90%
10	100	89.66	100	89.66	100	89.28	0.42%
5.5	55	50.98	55	50.98	55	50.79	0.27%

We can learn from Figure 8a that the Obj increases with the expansion of inspection capacity K. For the same instance, the Obj values obtained by the three methods are similar and the numbers of inspected ships obtained by the three methods share the same value as shown in Table 12. For the ten instances with different inspection capacities, all the optimality gaps of the WIS algorithm are smaller than 1.00% as shown in Figure 8b.

Figure 9a illustrates the CPU time required to solve instances with different inspection capacities using three methods. The x-axis represents the inspection capacity and the y-axis represents the CPU time for solving the model in seconds. Figure 9b illustrates the CPU time ratio of instances solved by the CIP and WIS methods. The x-axis represents the inspection capacity and the y-axis represents the CPU time ratio. Table 13 records the computation time of the three methods for instances with different inspection capacities.

From Figure 9 and Table 13, we can know that, when $K \leqslant 4$, the CPU time of all the three methods is less than 1 s. When $K \geqslant 5$, the CPU time of SIP rapidly goes up, while the CPU time of CIP and WIS remains less than 1 s as shown in Figure 9a. Therefore, the values of the CPU time ratio of CIP and WIS increase a lot for instances with an inspection capacity larger than 4 as shown in Figure 9b.



(**a**) (**b**)

Figure 9. Computation time of the three methods for instances with different inspection capacities. (**a**) CPU time of instances solved by the three methods. (**b**) CPU time ratio of instances solved by CIP and WIS.

To sum up, we conduct numerical experiments on the performance of the three methods, i.e., the SIP model solved by Gurobi, the CIP model solved by Gurobi, and the WIS algorithm.

From the perspective of the solution quality, SIP and CIP can obtain optimal solutions in all instances, so they share the same Obj for the same instance. The optimal Obj increases with the increase in merchant ship number $|\mathcal{V}|$ and gradually converges to a constant when $|\mathcal{V}|$ is large enough. Since the WIS algorithm is a tailored heuristic algorithm, the Obj of WIS is smaller than the optimal Obj for most instances and reaches optimality on one instance among all 20 instances with different $|\mathcal{V}|$. However, gaps of the Obj obtained by WIS are smaller than 1.50%, with an average value of 0.37%. This demonstrates that the quality of the solution obtained by WIS is relatively high. From the perspective of the computation time, the CIP model, which eliminates symmetry, significantly outperforms the SIP model with symmetry in terms of computational efficiency. Compared with the SIP model, the CPU time ratio of the CIP model is 102.07 on average. WIS performs much more prominently because its CPU time ratio has a maximum value of 5384.63 and an average value of 877.19. This means that the solving speed of WIS is hundreds to thousands of times faster than the speed of SIP solved by Gurobi. Hence, the proposed WIS algorithm can obtain a solution with a small gap from an optimal solution at a faster speed, which significantly enhances the solving efficiency. Furthermore, the experimental results of the sensitivity analysis show that the three methods have stable performance under the condition of changing the location number, the government ship number, or the inspection capacity. Changes in the location number have a minimal impact on Obj, whereas increasing the number of government ships or enhancing the inspection capacity can significantly increase Obj. Overall, the experimental results validate the effectiveness of our proposed models and underscore the robustness and reliability of our approach in practical applications.

Table 13. Computation time of the three methods for instances with different inspection capacities.

K	SIP	CIP		WIS	
	CPU Time (s)	CPU Time (s)	CPU Time Ratio	CPU Time (s)	CPU Time Ratio
1	0.1550	0.1362	1.14	0.0164	9.45
2	0.3601	0.1124	3.20	0.0164	21.96
3	0.6072	0.1215	5.00	0.0163	37.25
4	0.4802	0.1258	3.82	0.0163	29.46
5	2.1890	0.1254	17.46	0.0164	133.48
6	14.0011	0.1204	116.29	0.0167	838.39
7	9.4079	0.1422	66.16	0.0168	559.99
8	20.6243	0.1360	151.65	0.0165	1249.96
9	36.6615	0.1403	261.31	0.0165	2221.91
10	23.2500	0.1623	143.25	0.0171	1359.65
5.5	10.7736	0.1323	76.93	0.0165	646.15

6. Conclusions

In the context of the increasingly deteriorating global environment, the maritime sector is responsible for 3% of global carbon emissions alongside a 20% increase in emissions over the past decade, bearing a significant responsibility for environmental protection. Inland waterway transport is essential for sustainable maritime transportation and is regulated strictly for fuel quality. However, economic incentives lead crew members to use cheaper, lower-quality fuels, making government inspections crucial. The existing literature mainly focuses either on predicting the ship deficiency to select the inspected ship efficiently, or optimizing the scheduling process of crews. Although a few studies consider the ship selection and inspector scheduling simultaneously, they are not applicable to the requirements of multi-inspector and multi-location scenarios in inland waterways and they lack efficient heuristic algorithms for solving the problem.

To tackle the ship selection and inspector scheduling problem under the multi-inspector and multi-location scenario, we develop an IP model with symmetry, the SIP model, and a more compact, symmetry-eliminated version, the CIP model, to optimize the ship selection and inspection scheduling problem. These two models are solved to optimality by the commercial solver Gurobi. Meanwhile, by observing the distribution of the sum of ship weights at different locations, we design a heuristic algorithm tailored to the unique structure of the problem, i.e., the WIS algorithm, to solve large-scale problems more efficiently. In the numerical experiments, we use the Yangtze River as an example to conduct the case study. The results show that both solutions of the SIP model and the CIP model reach optimality, with the latter one 102.07 times faster than the former one on average. The gap of the objective value obtained by the WIS algorithm is less than 1.50%, with an average value of 0.37%. As for the computation time, the speed of our algorithm is 877.19 times faster than that of the SIP model solved by Gurobi. This demonstrates the outstanding performance of the WIS algorithm in solving efficiency.

This study presents a novel approach to inland waterway ship inspections by integrating ship selection and inspection scheduling. By leveraging ship visit information, such as their arrival times and berthing periods, alongside port inspection resources, the study aids in decision-making for planning inland ship inspections. This approach provides a more practical and implementable solution, enhancing inspection efficiency and contributing to the protection of the ecological environment in inland waterways. Such a method, in addition to being applicable to multi-inspector and multi-location inspection scenarios in inland waterways, can also address other practical problems, such as the inspection scheduling problem in other fields, the package inspection problem in a warehouse, and the healthcare resource allocation problem. However, the model assumes uniform inspection quality across all inspectors, which may not reflect reality. Incorporating varying levels of inspector expertise and experience could yield more accurate results. Meanwhile, further exploration can consider the information on the ships to predict the importance of inspecting specific ships in advance. This combination of prediction and optimization is able to ultimately improve the overall inspection efficiency for the multi-inspector and multi-location scenario.

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

Funding: This research received no external funding.

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. UNCTAD. Review of Maritime Transport 2023. 2023. Available online: https://unctad.org/publication/review-maritime-transport-2023 (accessed on 30 May 2024).
2. Raza, Z.; Svanberg, M.; Wiegmans, B. Modal shift from road haulage to short sea shipping: A systematic literature review and research directions. *Transp. Rev.* **2020**, *40*, 382–406. [CrossRef]
3. Comi, A.; Polimeni, A. Assessing the potential of short sea shipping and the benefits in terms of external costs: Application to the Mediterranean Basin. *Sustainability* **2020**, *12*, 5383. [CrossRef]
4. Trivedi, A.; Jakhar, S.K.; Sinha, D. Analyzing barriers to inland waterways as a sustainable transportation mode in India: A dematel-ISM based approach. *J. Clean. Prod.* **2021**, *295*, 126301. [CrossRef]
5. Gbako, S.; Paraskevadakis, D.; Ren, J.; Wang, J.; Radmilovic, Z. A systematic literature review of technological developments and challenges for inland waterways freight transport in intermodal supply chain management. *Benchmarking Int. J.* **2024**. [CrossRef]
6. Mohanta, S.; Navandar, Y.V.; Bivina, G.R.; Krishnamurthy, K. Inland water transport: A bibliometric literature review. *Sci. Technol. Libr.* **2023**, *42*, 399–416. [CrossRef]
7. Calderón-Rivera, N.; Bartusevičienė, I.; Ballini, F. Sustainable development of inland waterways transport: A review. *J. Shipp. Trade* **2024**, *9*, 3. [CrossRef]
8. Anku, R.; Pruyn, J.; Thill, C. A review of the state-of-the-art sustainable and climate-resilient inland waterway vessels. In Proceedings of the International Marine Design Conference, Amsterdam, The Netherlands, 2–6 June 2024.

9. National People's Congress of the People's Republic of China. Law of the People's Republic of China on the Prevention and Control of Atmospheric Pollution. 2018. Available online: https://www.mee.gov.cn/ywgz/fgbz/fl/201811/t20181113_673567.shtml (accessed on 26 May 2024).
10. Yang, Z.; Yang, Z.; Yin, J.; Qu, Z. A risk-based game model for rational inspections in port state control. *Transp. Res. Part E Logist. Transp. Rev.* **2018**, *118*, 477–495. [CrossRef]
11. Yan, R.; Wang, S.; Peng, C. An artificial intelligence model considering data imbalance for ship selection in port state control based on detention probabilities. *J. Comput. Sci.* **2021**, *48*, 101257. [CrossRef]
12. Wu, S.; Chen, X.; Shi, C.; Fu, J.; Yan, Y.; Wang, S. Ship detention prediction via feature selection scheme and support vector machine (SVM). *Marit. Policy Manag.* **2022**, *49*, 140–153. [CrossRef]
13. Tian, X.; Wang, S. Cost-sensitive Laplacian logistic regression for ship detention prediction. *Mathematics* **2022**, *11*, 119. [CrossRef]
14. Yan, R.; Wang, S.; Cao, J.; Sun, D. Shipping domain knowledge informed prediction and optimization in port state control. *Transp. Res. Part B Methodol.* **2021**, *149*, 52–78. [CrossRef]
15. Qiao, X.; Yang, Y.; Guo, Y.; Jin, Y.; Wang, S. Optimal routing and scheduling of flag state control officers in maritime transportation. *Mathematics* **2024**, *12*, 1647. [CrossRef]
16. Yan, R.; Yang, Y.; Du, Y. Stochastic optimization model for ship inspection planning under uncertainty in maritime transportation. *Electron. Res. Arch.* **2023**, *31*, 103–122. [CrossRef]
17. Luo, X.; Sun, Z.H.; Qiu, S. Ant colony system based drone scheduling for ship emission monitoring. In Proceedings of the 2021 IEEE Congress on Evolutionary Computation (CEC), Krakow, Poland, 28 June–1 July 2021; pp. 241–247.
18. Xiao, Y.; Qi, G.; Jin, M.; Yuen, K.F.; Chen, Z.; Li, K.X. Efficiency of port state control inspection regimes: A comparative study. *Transp. Policy* **2021**, *106*, 165–172. [CrossRef]
19. Junaidi, A.; Yudo, H.; Ab-Samat, H. Identification of data analysis methods and focus trends in port state control inspections: A comprehensive literature review. *Int. J. Technol.* **2024**, *15*, 179. [CrossRef]
20. Prieto, J.M.; Amor, V.; Turias, I.; Almorza, D.; Piniella, F. Evaluation of Paris MoU maritime inspections using a STATIS approach. *Mathematics* **2021**, *9*, 2092. [CrossRef]
21. Knapp, S.; Bijwaard, G.; Heij, C. Estimated incident cost savings in shipping due to inspections. *Accid. Anal. Prev.* **2011**, *43*, 1532–1539. [CrossRef]
22. Cariou, P.; Mejia, M.Q., Jr.; Wolff, F.C. An econometric analysis of deficiencies noted in port state control inspections. *Marit. Policy Manag.* **2007**, *34*, 243–258. [CrossRef]
23. IMO. Resolution A.1185(33): Procedure for Port State Control. 2023. Available online: https://wwwcdn.imo.org/localresources/en/OurWork/IIIS/Documents/A%2033-Res.1185%20-%20PROCEDURES%20FOR%20PORT%20STATE%20CONTROL,%202023%20(Secretariat)%20(1).pdf (accessed on 10 May 2024).
24. Heij, C.; Knapp, S. Shipping inspections, detentions, and incidents: An empirical analysis of risk dimensions. *Marit. Policy Manag.* **2019**, *46*, 866–883. [CrossRef]
25. Knapp, S.; Heij, C. Improved strategies for the maritime industry to target vessels for inspection and to select inspection priority areas. *Safety* **2020**, *6*, 18. [CrossRef]
26. Tokyo MoU. Information Sheet of the New Inspection Regime (NIR). 2014. Available online: http://www.tokyo-mou.org/doc/NIR-information%20sheet-r.pdf (accessed on 30 May 2024).
27. Rizvanolli, A.; Heise, C.G. Efficient ship crew scheduling complying with resting hours regulations. In *Operations Research Proceedings 2016: Selected Papers of the Annual International Conference of the German Operations Research Society (GOR), Helmut Schmidt University, Hamburg, Germany, 30 August–2 September 2016*; Springer: Cham, Switzerland, 2018; pp. 535–541.
28. Leggate, A.; Sucu, S.; Akartunalı, K.; van der Meer, R. Modelling crew scheduling in offshore supply vessels. *J. Oper. Res. Soc.* **2018**, *69*, 959–970. [CrossRef]
29. Xiao, L.; Wang, Z.; Tan, Z.; Wang, C. A solution method for the maritime pilot scheduling problem with working hour regulations. *Asia-Pac. J. Oper. Res.* **2021**, *38*, 2040015. [CrossRef]
30. Jia, S.; Wu, L.; Meng, Q. Joint scheduling of vessel traffic and pilots in seaport waters. *Transp. Sci.* **2020**, *54*, 1495–1515. [CrossRef]
31. Giachetti, R.E.; Damodaran, P.; Mestry, S.; Prada, C. Optimization-based decision support system for crew scheduling in the cruise industry. *Comput. Ind. Eng.* **2013**, *64*, 500–510. [CrossRef]
32. Darvish, M.; Coelho, L.C.; Jans, R. *Comparison of Symmetry Breaking and Input Ordering Techniques for Routing Problems*; CIRRELT: Québec, QC, Canada, 2020.
33. Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*; Gurobi Optimization, LLC: Beaverton, OR, USA, 2022.
34. Yangzhou Navigation Center. Dongting Lake and Yangtze River Navigation Distance Table. 2017. Available online: https://mp.weixin.qq.com/s/CZ75gKEHD_zdKPpVll2Zbg (accessed on 15 May 2024).
35. Ministry of Transport of the People's Republic of China. Regulations on Vessel Routing in the Jiangsu Section of the Yangtze River (2021). 2021. Available online: https://xxgk.mot.gov.cn/2020/xzgfxwj/202212/t20221206_3719908.html (accessed on 15 May 2024).

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.