https://doi.org/10.1016/j.trb.2018.10.011

Drone Scheduling to Monitor Vessels in Emission Control Areas

Jun Xia[a], Kai Wang[b], Shuaian Wang[b,*]

[a] *Sino-US Global Logistics Institute, Antai College of Economics & Management, Shanghai Jiao Tong University, China*

[b] *Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, Hong Kong*

Abstract

The use of drones to monitor the emissions of vessels has recently attracted wide attention because of its great potentials for enforcing regulations in emission control areas (ECAs). Motivated by this potential application, we study how drones can be scheduled to monitor the sailing vessels in ECAs, which is defined as a drone scheduling problem (DSP) in this paper. The objective of the DSP is to design a group of flight tours for drones, including the inspection sequence and timings for the vessels, such that as many vessels as possible can be inspected during a given time period while prioritizing highly weighted vessels for inspection. We show that the DSP can be regarded as a generalized team orienteering problem, which is known to be NP-hard, and deriving solutions for this problem can be more difficult because additional complicated features, such as time-dependent locations, multiple trips for a drone, and multiple stations (or depots), are addressed simultaneously. To overcome these difficulties, we model the dynamics of each sailing vessel using a real-time location function in a deterministic fashion. This approach allows us to approximately represent the problem on a time-expanded network, based on which a network flow-based formulation can be formally developed. To solve this proposed formulation, we further develop a Lagrangian relaxation-based method that can obtain near-optimal solutions for large-scale instances of the problem. Numerical experiments based on practically generated instances with 300 time points and up to 100 vessels are conducted to validate the effectiveness and efficiency of the proposed method. Results show that our method derives tight upper bounds on optimal solutions, and can quickly return good feasible solutions for the tested instances. We also conduct experiments based on realistic tracking data to demonstrate the usefulness of our solutions, including those for the cases considering the uncertainty of vessel locations.

Keywords: drone scheduling; emission control area; time-expanded network; Lagrangian relaxation.

*The corresponding author

Email addresses: `lgtxiaj@sjtu.edu.cn` (Jun Xia), `chwangkai@gmail.com` (Kai Wang),

1. Introduction

Over 80% of global trade volumes are carried by oceangoing vessels (Xiao et al. 2015, Ng 2015). These shipping activities emit large amounts of exhaust gases including carbon dioxide (CO_2), nitrogen oxides (NO_x), and sulfur oxides (SO_x) (see, e.g., UNCTAD 2017, Zheng et al. 2017). According to the International Maritime Organization (IMO) (IMO 2018), shipping operations account for nearly 10–15% of anthropogenic SO_x emissions around the world, with most of these emissions coming from densely populated coastal regions (Transport and Environment 2018). For example, marine transport in 2012 generated 50% and 45% of the overall SO_x emissions in Hong Kong and Los Angeles, respectively (Environmental Protection Department 2015, Starcrest 2011). In these regions, SO_x emissions lead to significant environmental problems along with serious health impacts (see, e.g., Kirschstein and Meisel 2015, Corbett et al. 2007), which result in strong motivations to ameliorate the polluted environment.

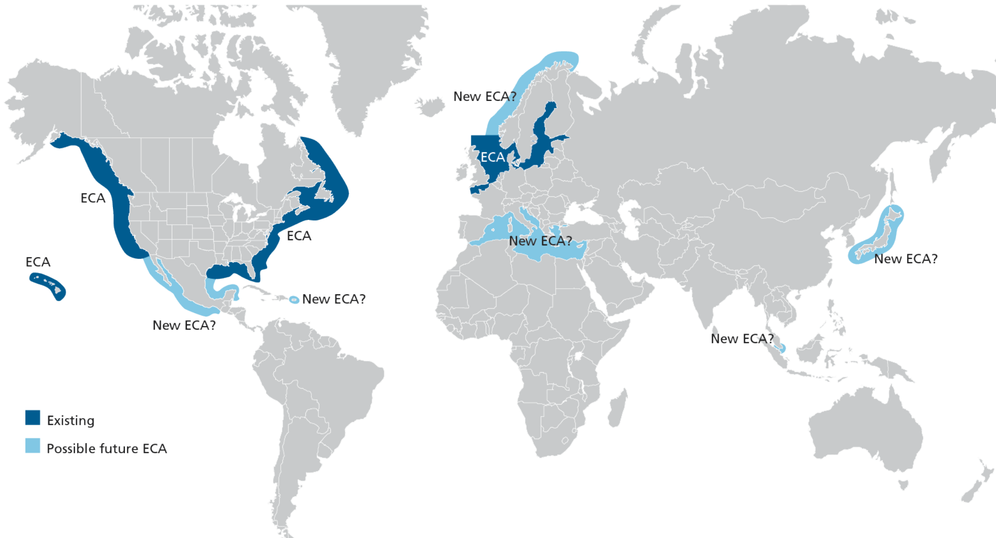


Figure 1: Existing and possible future emission control areas (Safety4Sea 2018)

The high SO_x emissions is mainly attributed to the consumption of heavy fuel oil which has an approximate sulfur percentage of 3.5%. To reduce the SO_x emitted from vessels in coastal regions, local governments and the IMO adopt a variety of regulations; for instance, Hong Kong limits the use of sulfur in its port area and Los Angeles offers a subsidy for purchasing low-sulfur fuel (Environmental Protection Department 2015, Starcrest 2011). As shown in Figure 1, IMO has designated four *emission control areas* (ECAs) since 2015 and may set up more ECAs in the near future. Vessels in ECAs must use fuels containing less than 0.1% of sulfur. According to the

wangshuaian@gmail.com (Shuaian Wang)

Marine Environment Protection Committee (MEPC 2016), the sulfur percentage must be below 0.5% around the world after 2020. Although relevant emission regulations have been in effect in a number of coastal regions, their enforcements are far from effective (Martek 2018). Since using low-sulphur fuels leads to high operation costs, many shipowners take the risk of not complying with the regulations. Such non-compliance ratios even reach up to 12.3% according to historical cases (OECD 2018). Therefore, the enforcement of relevant regulations must be enhanced to reduce SO_x emissions in ECAs.

A promising way to improve the regulation enforcement in ECAs is the use of drones to inspect vessels' compliance with the regulations (Ward and Kobe 2015, Ship Efficiency Review 2016, Green4Sea 2016). With the rapid development of related technologies for unmanned aerial vehicles (UAVs), drones are being more commonly used in the shipping industry (Peters 2016). Recently, Martek Marine, a drone technology supplier, signed a 72 million USD contract with the European Maritime Safety Agency (Port Technology 2017). In the contract, Martek Marine promises to develop durable drones (over 50 kilometers for a flight trip) to sample vessels' gases using self-equipped sensors. In that way, on-shore supervision managers can timely monitor the emission levels of the target vessels and check their compliance with designated regulations (Green4Sea 2016, Ship Efficiency Review 2016). In addition, the Danish Environmental Protection Agency has also started to research and develop sensors with more accurate inspection ability (Marine Electronics 2015). Other relevant practices to improve the airborne monitoring are also seen in Ward and Kobe (2015). These technological advancements will promote the practical uses of drones in the near future.

Regarding the use of drones to monitor vessels' emissions in ECAs, this study strives to validate this application from an operational perspective, in which relevant decisions have to be made by considering the following aspects:

- Maintaining a drone fleet is costly and maximizing its utilization is of practical interest (see, e.g., OECD 2018). Conducting a superficial inspection of only one vessel in each flight trip is inefficient. Hence, the decision should assign multiple inspection tasks for each operated tour to increase the utilization of the drone fleet.

- Compared with the limited fleet size of drones, the number of vessels for inspection in ECAs can be very large in certain time periods (see, e.g., Marine Traffic 2018), which implies that, sometimes, not all vessels are guaranteed to be inspected. Hence, the decision needs to answer how a subset of vessels can be selected for inspections, such that the effectiveness of monitoring operations can be maximized.

- One important feature in this application is that sailing vessels' locations vary from time to time, causing the flying times of drones between any two vessels to change dynamically. Furthermore, each inspection tour is restricted by the endurance of the drone's battery, such that a drone has to return to a base station to have its battery replaced before it runs out of power. Hence, both the sequence and the timing for operating the inspection tasks have to be optimized appropriately.

Specifically, the core of the decision is to assign each drone a sequence of inspection tasks with time schedules, which yield a set of flight tours constrained by battery powers. Owing to the limited size of the drone fleet, the optimized tours should include as many vessels as possible that are prioritized for inspection, which can be targeted as a maximization of the weighted number of vessels that have been inspected during a given time period. In this paper, we refer to this decision problem as a *drone scheduling problem* (DSP) which will be investigated later.

Consider a special case of DSP where all target vessels are anchored, and there is only one base station from which each drone is allowed to operate the inspection tour at most once. Then, the DSP is specialized into a team orienteering problem (TOP), which is a generalization of the orienteering problem (OP) known to be NP-hard (see, e.g., Golden et al. 1988). Notably, the DSP extends the TOP by considering that the locations of nodes, where vehicles collect rewards, are generalized to be time-dependent. Therefore, although the TOP and its variants have been studied with many effective models and methods (see, e.g., Butt and Ryan 1999, Boussier et al. 2007), most of them cannot be directly applied to tackle the DSP in a generalized setting.

Besides TOP, several studies also focus on routing UAVs in a number of applications, such as military surveillance (see, e.g., Murray and Karwan 2010, Xia et al. 2017) and logistics delivery operations (see, e.g., Murray and Chu 2015, Wang et al. 2017, Carlsson and Song 2017). For military surveillance, problems are often considered with dynamics and uncertainties, such as considering the dynamic appearances of new targets or considering the uncertain information collected from locations that are short of communications. For logistics delivery, drones cooperate with trucks to deliver packages to geographically located customers. However, a drone is often restricted to carrying only one parcel, which simplifies the drone routing decision to an assignment of customers to drones following a one-to-one relationship, because each flight trip of a drone can be seen as a trivial combination of a forward trip and a backward trip. The DSP studied in this work serves as a more general problem for routing drones because many additional generalized features are addressed at the same time, such as multiple tasks being assigned to each drone and multiple round trips being allowed to operate for each drone at multiple stations. What's more, the DSP involves the dynamic changes of vessels' locations, a condition that is fundamentally different from those that consider targets with fixed locations.

The contributions of this study can be summarized as follows. First, motivated by the application of drones to inspect the emissions of vessels in ECAs, we derive a new optimization problem that routes drones among sailing vessels to maximize the weighted number of inspected vessels. This problem is of both practical and academic interest because of its potential impact on improving the enforcement of ECA regulations, as well as the involvement of specific features that generalize the well-known TOP from the literature. Second, for this problem, we model the dynamics of vessels using deterministic real-time location functions. By this way, we can represent the DSP on a time-expanded network and develop for it a well-structured mixed-integer linear programming (MILP) formulation that can be potentially solved by developing efficient decomposition techniques. Third, to solve the proposed model formulation, we develop a Lagrangian relaxation-based method that experimentally solves the proposed formulation with 300 time points and up to 100 vessels. For the subset of tested instances with no more than 80 vessels, desirable feasible solutions with close optimality gaps (less than 3% on average) can be obtained quickly in 20 iterations.

The remainder of this paper is organized as follows. Section 2 reviews recent works that are related to the problem. The DSP is described in Section 3 and is formulated as an MILP model based on a time-expanded network in Section 4. In Section 5, we further develop a Lagrangian relaxation-based method for the problem. Numerical experiments are presented in Section 6, followed by a discussion on extended solutions for handling the uncertainty of vessel locations in Section 7. Conclusion and discussion of future works are presented in Section 8.

2. Literature review

In this section, we provide comprehensive reviews on existing operations research (OR) practices related to ECAs in Section 2.1, the state-of-the-art techniques for solving the TOP in Section 2.2, and the existing studies for routing drones in various applications in Section 2.3, respectively.

2.1. Related works on OR practices considering ECA regulations

To the best of our knowledge, there are only a limited number of OR applications related to ECAs, with most of them focused on optimizing the tactics of the ship operator in response to ECA regulations. Given that shipping operators are requested to use high-cost clean fuels in ECAs, existing OR works have mainly been contributed to optimize the speed of vessels, the objective of which is to minimize total fuel cost while following the ECA regulations. Different from conventional studies without ECA regulations (see, e.g., Meng et al. 2016, He et al. 2017), Doudnikoff and Lacoste (2014) show that vessels prefer to sail in low speeds inside ECAs but fast outside to catch up with the predetermined port call schedules and this practice results in an increase in CO_2 emissions. In addition to speed optimization for vessels, Fagerholt et al. (2015)

further take into account the decision on vessels' sailing paths in nearby regions of ECAs. An optimization model is developed to select paths from a realistic candidate path set. Their results show that ECA regulations may lead vessels to sail long distance paths, in order to save on total fuel cost. Sailing paths and speeds are also jointly optimized in Fagerholt and Psaraftis (2015), where the proposed model additionally determines the crossover points intersected by the vessels' paths and the boundary of an ECA.

The DSP can be seen as another OR practice related to ECAs. This work attempts to help supervision departments enforce the ECA regulations.

2.2. Related works on TOP

The DSP generalizes a conventional TOP that aims to design round-trips for a set of vehicles, such that the total collected rewards from visited nodes are maximized. One may refer to Vansteenwegen et al. (2011) and Gunawan et al. (2016) for a comprehensive review on recent state-of-the-arts for the TOP.

Both exact and heuristic methods are studied for the TOP. Boussier et al. (2007) propose a set-partition formulation by generating all possible routes, whose linear programming relaxation can be effectively solved by a column generation technique. With the obtained relaxation bounds, the authors develop a branch-and-price algorithm that optimally solves the problems with up to 100 nodes. To improve the computational efficiency, Keshtkaran et al. (2016) introduce a bounded bidirectional dynamic programming technique, where possible partial paths are extended in both forward and backward directions. To speed up partial path explorations, they embed a called decremental state space relaxation rule to dynamically control the number of visits to each included node. Valid cuts from classic vehicle routing problems, such as subset-row inequalities, are also utilized to strengthen the relaxation bounds. Bianchessi et al. (2018) propose a new two-index compact formulation for the TOP, based on which a branch-and-cut algorithm with newly developed valid cuts are proposed, and the problems with up to 100 nodes are solved. To tackle larger instances, Lin and Vincent (2012) develop a standard simulated annealing method that finds new best solutions for some benchmark instances with up to 288 nodes. Dang et al. (2013) study a particle swarm optimization algorithm enhanced with a faster evaluation process. The proposed algorithm achieves a stable performance on extensive test instances having up to 399 nodes.

Routing drones can be regarded as an extension of TOP for real applications, in which drones equipped with sensors are dispatched to patrol a set of targets, such that total rewards collected from the patrolling operations can be maximized. Compared with the classical TOP and its variant problems (see, e.g., Archetti et al. 2014, Ke et al. 2016), some generalization features are addressed in the DSP. First, in the DSP, a drone is allowed to operate more than one trips from multiple stations, a situation that generalizes the TOP by considering multiple depots and designing one or more routes for each vehicle to collect rewards. Second, the DSP can also be seen as a TOP that considers dynamically located nodes, which implies that the traveling time between any two nodes will be dependent on what time the travel happens. These generalization features make the problem more complicated, such that existing models and methods for the TOP cannot be directly applied to the DSP.

2.3. Related works on routing drones in various applications

Drones have been successfully applied for military use. Shetty et al. (2008) study the routing of a fleet of drones to destroy a designated set of differently prioritized targets. They propose a two-phase solution framework that involves solving a target assignment subproblem for each drone in the first phase and then solving a traveling salesman problem to gain a routing plan in the second phase. Considering a more complex environment in which new targets may appear dynamically, Murray and Karwan (2010) develop an integer programming model that reassigns drones to the updated set of tasks in response to any changes in the battlefield. Mufalli et al. (2012) consider the routing of drones for military surveillance missions, where drones carry sensors and collect information from designated targets. Their proposed model decides the sensors for each drone by incorporating payload capacity constraints. A fleet of drones are then routed following these constraints to maximize the information gained from the surveillances. Uncertainty on the information collected from the surveillances is also considered. For this consideration, Xia et al. (2017) develop a region-sharing strategy that dynamically routes drones to collect information rather than sticking to a predetermined routing plan. The proposed strategy is proven to be effective in a modern battlefield where communications between drones and ground stations are often blocked.

Cooperation between drones and trucks is investigated in last-mile delivery operations, which aim to improve operational efficiency. Murray and Chu (2015) study a joint scheduling problem for both drones and trucks, where drones are dispatched to service customers near the depot and trucks are mainly responsible for delivering far packages. Wang et al. (2017) investigate another cooperation manner between drones and trucks, where trucks are allowed to carry drones along the working routes. In this way, drones can fly from the trucks to visit those customers who are far from the depots. Similar cooperations between drones and trucks are also considered by Carlsson and Song (2017) and Agatz et al. (2018). Carlsson and Song (2017) theoretically analyze the delivery efficiency improved from the cooperation and prove that the potential improvement is dependent on the square root of the ratio of the truck's speed and the drone's speed. Agatz et al. (2018) investigate a new variant of the traveling salesman problem by considering the collaboration with a drone. They model the problem as an integer program and develop a heuristic method based on local search and dynamic programming. Improvements from the collaboration are also shown in their numerical experiments. Dorling et al. (2017) study a package delivery problem, in which packages are delivered using drones only. By considering some practical constraints, including limited endurance time and limited carried weights, they regard the problem as a multi-trip vehicle routing problem with side constraints.

From the aforementioned works, we notice that drones are routed among fixedly located nodes, and the distances between the geographical nodes are all fixed in their problems. A fundamental difference of our case from these earlier studies is that the locations of the vessels for inspection are time-dependent. This generalization makes routing drones much more difficult.

3. Problem description

In this section, we formally describe the DSP. Relevant notations for the problem are presented in Section 3.1. Vessels' real-time location coordinates are defined in Section 3.2. Calculation of drone's real-time flying time between pair of vessels and stations is discussed in Section 3.3. The objective and constraints for the problem are given in Section 3.4.

3.1. Notations

Consider an ECA that includes a set of base stations K where drones stop for battery replacements. Let τ_0 denote the time needed for each battery replacement operation. Each station $k \in K$ is associated with a fixed location coordinate (α_k, β_k). A set of identical drones, denoted by M, are operated to inspect the real-time emission condition of the vessels in the ECA for a given planning period denoted by $[0, T_{\max}]$. Let m_k denote the number of drones initially allocated at station k. Suppose each drone is able to fly in a maximum speed s and its maximum endurance time is up to Q minutes owing to its battery power limit.

Let V denote the set of vessels, and each vessel $v \in V$ is assumed to sail at speed s_v, where $s_v < s$. Within the planning period, we define $[e_v, l_v]$ as a time interval when vessel v is sailing inside the ECA, which can be regarded as a time window for inspecting vessel v. The inspection time for each vessel $v \in V$ is τ_v. We assume vessels have different weights of importance for inspection. For example, vessels with a non-compliance history should be highly weighted, while vessels that have been previously inspected in other zones can be weighted with lower values. We denote by w_v the weight (or revenue) for inspecting each vessel $v \in V$.

A drone can start and end an inspection tour at the same station or at different stations. The itinerary of a drone mainly consists of three operations, including battery replacement at stations, emission inspections over vessels, and flying trips between two adjacent vessel inspection tasks.

When a drone flies to a vessel, it will follow the vessel's sailing track for a short period of time in order to conduct emission inspections. A drone that has returned to a station is allowed to restart a new tour after its used battery is replaced with a fully charged one. An example of feasible tours is illustrated in Figure 2, where two stations denoted by $K = \{k_1, k_2\}$ and six sailing vessels (candidates to be inspected) denoted by $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ are included in the ECA. In the example, a feasible tour can either form a loop based on a single station (i.e., $k_2 \to v_1 \to v_2 \to k_2$ for drone B) or form a path between two stations (i.e., $k_1 \to v_3 \to v_4 \to k_2$ and $k_2 \to v_5 \to v_6 \to k_1$ for drone A). The DSP is to design for drones a group of such tours with corresponding time schedules (hereafter called scheduled tours), with the aim of inspecting as many vessels as possible during a given time period while prioritizing highly weighted vessels for inspection.

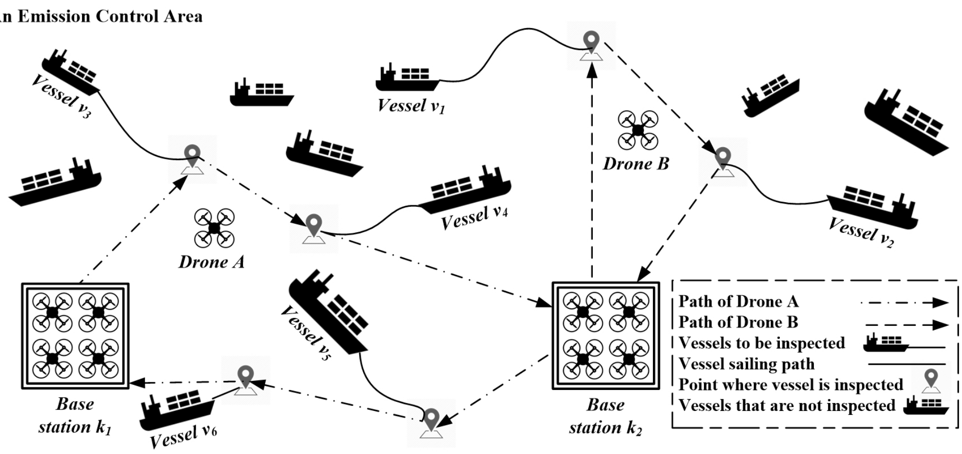


Figure 2: Example of drone's feasible tours in an ECA

3.2. Real-time location coordinates of the vessel

The DSP differs from the conventional TOP mainly in the aspect that the vessel locations are not fixed during the planning horizon. In other words, each scheduled tour is constituted by a sequence of real-time locations of the vessels when they are being inspected. To design good scheduled tours, one has to identify each vessel's real-time location at any time of the planning period. In the problem, we target on incoming vessels, each of which has a port to call in the ECA. Sailing tracks of these vessels are relatively fixed, compared with the cases of freely sailing vessels. Hence, we assume vessels' real-time locations during the next short time period can be predicted based on the vessels' sailing statuses (e.g., speed and course), which are precisely accessed by an automatic identification system (see, e.g., Marine Traffic 2018). On the basis of this assumption, the location coordinates of each vessel $v \in V$ at any time $t \in [e_v, l_v]$, denoted by $(\alpha_v(t), \beta_v(t))$, are priorly known.

Figure 3 gives an example to describe the variation of a vessel's location coordinates during the given period, where the horizontal axis stands for the time and the vertical axis stands for the coordinate value. As noted in the figure, during time window $[e_v, l_v]$, vessel v may stay at a fixed location (α_p, β_p) in port or sail along a designated direction in the prediction. Given the predicted real-time locations for all target vessels, we are now allowed to estimate the flying time of the drone between each two moving vessels in the system.

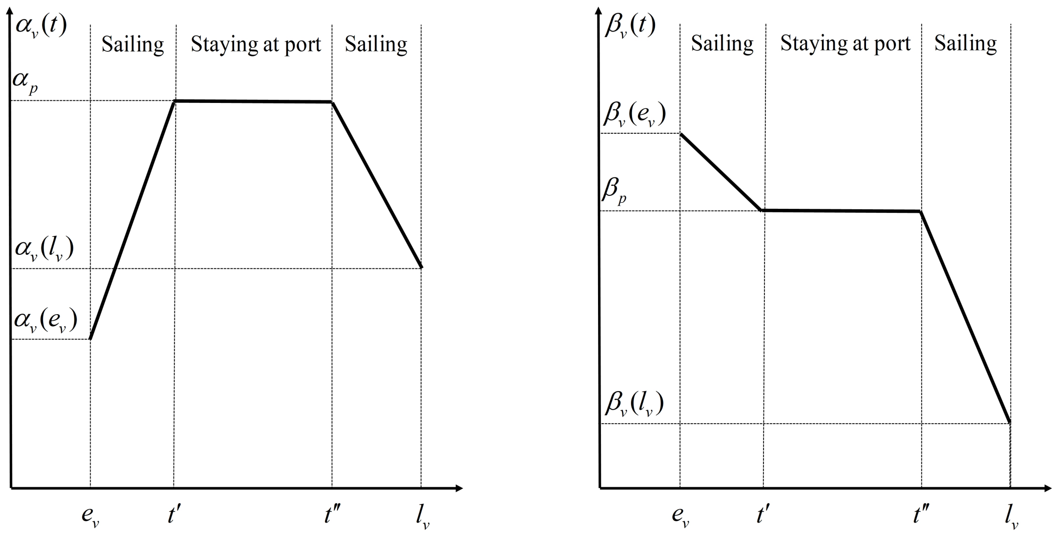


Figure 3: An example of vessel's real-time location coordinate

3.3. Calculate drone's flying time in a tour

At any time of the planning period, the drone's flying time between two vessels depends on the real-time locations of the vessels in the ECA. Before calculating the flying time, we first show the following optimal strategy for a drone to finish a given sequence of inspection tasks.

Proposition 1. *Given a tour (i.e., the base stations and the set of ordered vessels for inspection), there exists an optimal schedule such that the inspection of each vessel is finished as early as possible.*

Proof. Given an inspection tour by $\{k \to v_1 \to ... \to v_n \to k'\}$, we first observe that it is always beneficial to finish the tour as early as possible. Suppose the tour starts from station k at time t^0, and the drone can fly in different ways (i.e., different speeds and directions) to inspect the vessels, potentially resulting in two different schedules (i.e., schedule 1 and schedule 2) with their return times to station k' represented by t^1 and t^2, respectively, where $t^1 < t^2$. Suppose schedule 2 is in an optimal solution for the DSP, then schedule 2 in the solution can be replaced by schedule 1, which forms another feasible solution with the same objective value. Thus, schedule 1 is also in an optimal solution for the DSP.

Following the above observation, we next show that a drone can finish a tour earlier if the inspection of the last vessel could be completed earlier. Suppose vessel v_n is the last vessel to be inspected in the tour, and there are two schedules that respectively finish the inspection of vessel v_n in two different locations (i.e., location 1 and location 2) at times t^1 and t^2, where $t^1 < t^2$ (see the illustrative example in Figure 4). Suppose the flying time for a drone to return to the end station from location 2 is denoted by $\bar{t}$. Thus, for the drone that finishes the inspection at location 1, it can always follow the sailing track of vessel v_n from location 1 to location 2 and then fly back to the end station at time $t^1+\bar{t}+(t^2-t^1)s_{v_n}/s$, which will be earlier than $t^2+\bar{t}$, namely the return time after finishing the last inspection at location 2.

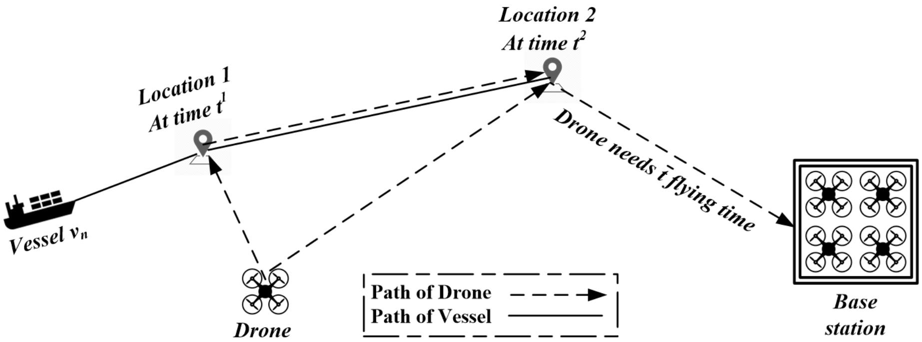


Figure 4: Illustration of the two possible locations to inspect the last vessel in a tour

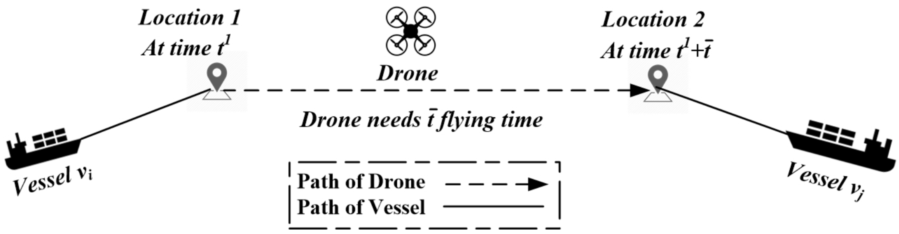


Figure 5: Illustration of the locations to inspect two adjacent vessels in a tour

Next, we prove that if a drone wants to inspect the next vessel earlier, it has to finish the inspection task of current vessel earlier, which is valid for each pair of adjacent inspection tasks in a given tour. Suppose a drone flies for a time length of $\bar{t}$ from location 1 (at time t^1), where inspection of vessel v_i is finished, to a next vessel v_j at location 2 (at time $t^1+\bar{t}$) (see the illustrative example in Figure 5). If the drone finishes the inspection of vessel v_i at t' where $t' < t^1$, it can fly along the sailing track of vessel v_i to location 1 and then fly to location 2 at time $t'+\bar{t}+(t^1-t')s_{v_i}/s$ that is earlier than $t^1+\bar{t}$. Given that the drone arrives location 2 earlier, it can further fly to vessel v_j for an earlier inspection.

By summarizing the above three segments of discussions, we conclude that inspecting every next vessel as early as possible is an optimal condition for the problem. □

Proposition 1 implies that given any sequence of inspection tasks, there always exists such an optimal schedule that requires the inspection of each vessel to be finished as early as possible. This property significantly simplifies the design of optimal time schedule for each tour (i.e., the timings of the vessels to be inspected). Therefore, whenever and wherever a drone is located, the drone is always motivated to fly in a most direct path with the maximum speed to somewhere the next target vessel is potentially located. Based on that, we can estimate the flying time using the manner of the two-dimensional Euclidean distance calculation.

Let $\tau_{v,v'}(t)$ denote the flying time needed for a drone that leaves vessel v at time t and reaches the next vessel v' for inspection. Given that the real-time location of vessel v at time t is $(\alpha_v(t), \beta_v(t))$, and after the drone's flying trip to vessel v', the real-time location of vessel v' is $(\alpha_v(t+\tau_{v,v'}(t)), \beta_v(t+\tau_{v,v'}(t)))$. Then, the minimum value of $\tau_{v,v'}(t)$ can be determined by solving the following equation.

$$\left[s \cdot \tau_{v,v'}(t)\right]^2 = \left[\alpha_v(t) - \alpha_{v'}(t + \tau_{v,v'}(t))\right]^2 + \left[\beta_v(t) - \beta_{v'}(t + \tau_{v,v'}(t))\right]^2. \tag{1}$$

Furthermore, denote by $\tau_{k,v}(t)$ the flying time needed from station k at time t to vessel v, and by $\tau_{v,k}(t)$ the flying time needed from vessel v at time t to station k. Given the real-time locations before and after a flying trip, similarly, minimum values of $\tau_{k,v}(t)$ and $\tau_{v,k}(t)$ can also be easily determined by solving the following equations, respectively.

$$\begin{aligned} [s \cdot \tau_{k,v}(t)]^2 &= [\alpha_k - \alpha_v(t + \tau_{k,v}(t))]^2 + [\beta_k - \beta_v(t + \tau_{k,v}(t))]^2, \quad (2) \\ [s \cdot \tau_{v,k}(t)]^2 &= [\alpha_v(t) - \alpha_k]^2 + [\beta_v(t) - \beta_k]^2. \quad (3) \end{aligned}$$

Proposition 2. *Equations* (1), (2), *and* (3) *always have positive roots.*

Proof. Since the right-hand-side of (3) is a positive constant, equation (3) must have an unique positive root. We next prove the existence of positive roots for (1) and (2), both of which can be equivalent to finding a positive value of τ such that the following function satisfies $f(\tau) = 0$.

$$f(\tau) = [s \cdot \tau]^2 - [a - a(\tau)]^2 - [b - b(\tau)]^2, \tag{4}$$

where a drone initially locates at (a, b) and inspects the next vessel at location $(a(\tau), b(\tau))$ after a time period of τ.

Note that $(a(0), b(0))$ is the initial location of the next inspected vessel at $\tau = 0$. Let $\ell_0 = \sqrt{[a - a(0)]^2 + [b - b(0)]^2}$ represent the Euclidean distance between (a, b) and $(a(0), b(0))$. Thus, by following the triangle inequality, there is $\ell_0 + s_v \cdot \tau > \sqrt{[a - a(\tau)]^2 + [b - b(\tau)]^2}$. Substituting this inequality into (4), we have

$$\begin{aligned} f(\tau) &= [s \cdot \tau]^2 - [a - a(\tau)]^2 - [b - b(\tau)]^2, \\ &> [s \cdot \tau]^2 - (\ell_0 + s_v \cdot \tau)^2, \\ &> (s^2 - s_v^2)\tau^2 - 2s_v\ell_0 \cdot \tau - \ell_0^2. \end{aligned}$$

Since $s > s_v$, we can always find a large enough τ' to guarantee $f(\tau') > 0$. Together with the initial condition that $f(0) = -\ell_0^2 < 0$, there is at least one root in $(0, \tau')$. This proposition is thus proved. □

Proposition 2 theoretically implies that any target vessel staying in the ECA can be inspected, if drone's battery is sufficiently endurable. Motivated by Proposition 1, if there exist multiple positive roots for an equation, the calculated flying time should take the value of the smallest one. Based on (1)–(3), drone's flying time among vessels and stations at any time of the planning period can be uniquely determined.

3.4. Objective and constraints

The DSP aims to design for drones a set of scheduled tours for monitoring the emissions of vessels. The objective of the problem is to maximize the total weights of the vessels that have been inspected (i.e., the weighted number of inspected vessels) during a given time period. The output decision has to observe the following constraints, which are necessary to define a feasible solution of this problem.

Tour feasibility constraints. A feasible scheduled tour must start from a station and end by returning to the same station or another. The maximum round-trip time of the scheduled tour is bounded by Q minutes.

Inspection constraints. Each vessel can be inspected at most once during the given time period. In other words, a vessel can be included in exactly one tour if this vessel is to be inspected.

Safety constraints. At each station, the time gap between two adjacent launches or landings of drones must be no smaller than a safety threshold (set as 1 minute in our study). These constraints are employed to avoid the situation where too many drones are operated at the same time over the same station, which may lead to a crash.

4. Time-expanded network formulation

In this section, we formulate the DSP as a network flow-based problem with integer flow restrictions. In Section 4.1, we construct the underlying network by expanding vessels and stations with additional time dimensions. In Section 4.2, an MILP formulation is proposed to represent the DSP based on this constructed time-expanded network.

4.1. Construct a time-expanded network

Time-expanded network is widely adopted in representing various applied problems, such as service network design and railway timetabling (see, e.g., Caprara et al. 2002, Crainic et al. 2014, Ng and Lo 2016). We represent the DSP by constructing a time-expanded network, where vessels and stations are associated with disjoint time points in $T = \{1, ..., T_{\max}\}$ to form the network nodes. The length of gap between each two adjacent time points can be differently set (e.g., 1 or 5 minutes), which coordinates the network scale and the discretization accuracy on time. In our study, we set the length of gap as 1 minute. Provided with the evenly distributed time points, we construct vessel-time nodes in $N = \{(v, t)|v \in V, t \in T_v\}$, where $T_v = \{e_v, e_v + 1, ..., l_v\}$ is the set of discretized time points included in vessel v's time window, and construct station-time nodes in $N_0 = \{(k, t)|k \in K, t \in T\}$.

Given each pair of vessel-time nodes (v, t) and (v', t'), we establish an arc from (v, t) to (v', t') if $\{t' - t - 1 < \tau_{v,v'}(t) + \tau_{v'} \leq t' - t\}$ holds. The constructed arc stands for a flying trip that leaves vessel v at time t for the next inspection over vessel v'; the inspection operation over vessel v' (with time $\tau_{v'}$) is included in this arc. Hence, holding arc (i.e., an arc adjacently connecting (v, t) to $(v, t + 1)$ for any $v \in V$ and $t \in T$) is not generated.

A flying trip starting from the location of vessel v at time t to the location of station k can be represented by a directed arc from (v, t) to (k, t') for some t' satisfying $\{t' - t - 1 < \tau_{v,k}(t) + \tau_0 \leq t' - t\}$. Note that the battery replacement operation (with time τ_0) is already included in the arc. Thus, any newly arriving drone at a station is allowed to immediately start off the next tour. Similarly, we construct an arc by connecting (k, t') to (v, t) if $\{t - t' - 1 < \tau_{k,v}(t) + \tau_v \leq t - t'\}$ is satisfied, which implies a flying trip from station k to vessel v started off at time t'. Note that the inspection operation over vessel v is in the arc, and hence a holding arc between each two adjacent vessel-time nodes with respect to the same vessel is not generated.

Let A denote the arc set including all the generated arcs. An acyclic time-expanded network, denoted by $G = (N \cup N_0, A)$, is then constructed to formulate the problem. For ease of presentation, we define i as the node index, and let u_i and t_i denote node i's corresponding vessel (or station) and time point, respectively. Given $e(i, j)$ denoted as the arc connecting node i to node j, we assign its arc weight by $w_{i,j} = \frac{1}{2}(w_{u_i} + w_{u_j})$, where $w_u = 0$ if $u \in K$. An illustrative example of the time-expanded network is presented in Figure 6, which is a time scheduled representation of the inspection tours in Figure 2.

In network G, a path from one station-time node to another corresponds to a possible scheduled tour for a drone by subjecting it to the aforementioned tour feasibility constraints. Since any return arc to a station-time node has included the battery replacement time τ_0, the maximum time length for a feasible path is given by $Q_0 = Q + \tau_0$. Based on the network representation, the DSP can be formulated as selecting a number of paths, such that inspection constraints and safety constraints are further observed, and the total weights of the selected paths are maximized.

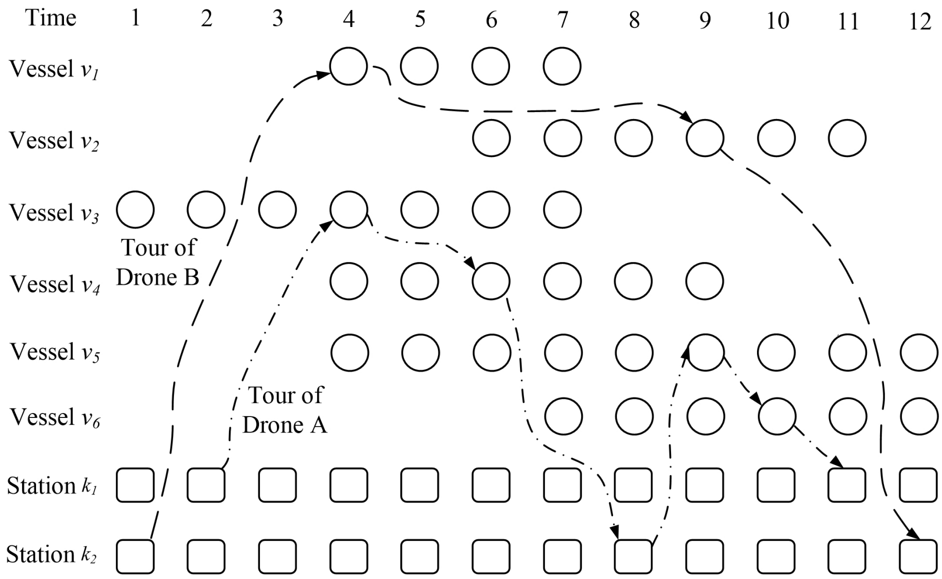


Figure 6: An example of three tours on a time-expanded network

4.2. MILP formulation

Basing on network G, we present an arc-based formulation in which a path is decided by a set of arc type binary variables $x_{i,j}$. Let $x_{i,j} = 1$ if arc $e(i,j)$ is included by a path, and $x_{i,j} = 0$ otherwise. The number of drones staying at station k at time t is denoted by an integer variable $y_{k,t}$. Let $y_{k,0} = m_k$ for $k \in K$. Note that $y_{k,0}$ is not a variable but a constant parameter used in the model. Define continuous variable q_i as the remaining working time of a drone on arrival at node i of the network. Based on the three sets of decision variables $\{x_{i,j}|e(i,j) \in A\}$, $\{y_{k,t}|k \in K, t \in T\}$, and $\{q_i|i \in N \cup N_0\}$, the DSP is formulated as follows.

$$\text{(F)}: \quad \max \sum_{e(i,j)\in A} w_{i,j} x_{i,j}, \tag{5}$$

$$\text{s.t.} \quad \sum_{j\in N} x_{i,j} - \sum_{j\in N} x_{j,i} = y_{u_i,t_i-1} - y_{u_i,t_i}, \qquad \forall i \in N_0, \tag{6}$$

$$\sum_{j\in N\cup N_0} x_{i,j} - \sum_{j\in N\cup N_0} x_{j,i} = 0, \qquad \forall i \in N, \tag{7}$$

$$q_i - (t_j - t_i)x_{i,j} + Q_0(1 - x_{i,j}) \geq q_j, \qquad \forall i \in N, \forall j \in N \cup N_0, \tag{8}$$

$$Q_0 - (t_j - t_i)x_{i,j} \geq q_j, \qquad \forall i \in N_0, \forall j \in N \tag{9}$$

$$\sum_{i \in N \cup N_0} \sum_{j \in N: u_j = v} x_{i,j} \leq 1, \qquad \forall v \in V, \tag{10}$$

$$\sum_{j \in N} x_{i,j} + \sum_{j \in N} x_{j,i} \leq 1, \qquad \forall i \in N_0, \tag{11}$$

$$x_{i,j} \in \{0, 1\}, \qquad \forall e(i,j) \in A, \tag{12}$$

$$y_{k,t} \in \{0, 1, ..., |M|\}, \qquad \forall k \in K, \forall t \in T, \tag{13}$$

$$q_i \geq 0, \qquad \forall i \in N \cup N_0. \tag{14}$$

The objective function (5) maximizes the weights of all included arcs, which correspond to total revenues obtained by inspecting the vessels. Constraints (6) calculate the number of drones staying on each station-time node by linking the decision variables in $\{x_{i,j}|e(i,j) \in A\}$ and $\{y_{k,t}|k \in K, t \in T\}$. Note that the number of drones in any station at any time must be ranged by $[0, |M|]$. Constraints (7) enforce that the arrival and departure of a drone on each vessel-time node must be balanced. Constraints (8) and (9) capture the charge depletion of the battery during the flight and update the remaining time of the drone upon arrival at each node, where the remaining time will recover to Q_0 after the used battery is replaced with a fully charged one upon visiting any station-time node. Constraints (7)–(9) are the tour feasibility constraints. Constraints (10) restrict that each vessel can be inspected at most once, which corresponds to the inspection constraints. Constraints (11) are the safety constraints that allow at most one drone to take off and land at each station-time node, given that the length of disjoint time gap and the safety time threshold are both equal to 1 minute in our study. The types and feasible domains of the decision variables are defined in (12)–(14). Note that the value of each variable $y_{k,t}$ is determined by a set of binary variables $x_{i,j}$ in (9), which is guaranteed to be an integer. Therefore, relaxing each $y_{k,t} \in \{0, 1, ..., |M|\}$ by $0 \leq y_{k,t} \leq |M|$ does not affect the integrity of this variable.

Model (F) is an MILP formulation whose size grows in the scale of network G. Although an off-the-shelf MILP solver such as CPLEX is available for a direct solution, our preliminary numerical tests show that the MILP solver is limited to solving problems with up to 40 vessels and the computations can be rather time-consuming. We are thus motivated to study a new solution approach that can be more effective and efficient for solving (F).

5. A Lagrangian relaxation-based method

Lagrangian relaxation-based techniques are widely applied to solve various operations research problems (i.e., Fisher 2004, Cacchiani et al. 2012), in which a Lagrangian subproblem is formed by dropping complicating constraints and penalizing violations of the dropped constraints in the objective function. In most situations, the Lagrangian subproblem has a decomposable structure and can be solved by a number of separated easier decisions. Knowing that optimal objective of a Lagrangian subproblem provides a valid bound (an upper bound in our case) on the optimal value of original problem, it is always available to measure the obtained feasible solutions with known optimality gaps.

In this section, we present a Lagrangian relaxation-based solution method to solve (F). In Section 5.1, we relax the inspection constraints from (F) to obtain a Lagrangian subproblem, which is then solved by equivalently optimizing a set partitioning-like formulation based on a limited number of columns. In Section 5.2, a lower bounding strategy is discussed. In Section 5.3, we propose a subgradient optimization procedure to iteratively converge the best-known upper bound to the best-known lower bound for the DSP.

5.1. Lagrangian relaxation

Relaxing Constraints (10) of (F) and bringing them into the objective function with associated Lagrangian multipliers $\pi_v \geq 0$ for each $v \in V$, we obtain the Lagrangian subproblem LR(Π), where $\Pi = \{\pi_v | v \in V\}$ denotes the vector of Lagrangian multipliers.

$$\text{LR}(\Pi): \quad \max \quad \sum_{(i,j)\in A} w_{i,j}x_{i,j} + \sum_{v\in V}\pi_v(1 - \sum_{i\in N\cup N_0}\sum_{j\in N:u_j=v} x_{i,j}), \tag{15}$$

$$\text{s.t.} \quad \sum_{j\in N} x_{i,j} - \sum_{j\in N} x_{j,i} = y_{u_i,t_i-1} - y_{u_i,t_i}, \qquad \forall i \in N_0, \tag{16}$$

$$\sum_{j\in N\cup N_0} x_{i,j} - \sum_{j\in N\cup N_0} x_{j,i} = 0, \qquad \forall i \in N, \tag{17}$$

$$q_i - (t_j - t_i)x_{i,j} + Q_0(1 - x_{i,j}) \geq q_j, \qquad \forall i \in N, \forall j \in N \cup N_0, \tag{18}$$

$$Q_0 - (t_j - t_i)x_{i,j} \geq q_j, \qquad \forall i \in N_0, \forall j \in N \tag{19}$$

$$\sum_{j\in N} x_{i,j} + \sum_{j\in N} x_{j,i} \leq 1, \qquad \forall i \in N_0, \tag{20}$$

$$x_{i,j} \in \{0, 1\}, \qquad \forall e(i,j) \in A, \tag{21}$$

$$0 \leq y_{k,t} \leq |M|, \qquad \forall k \in K, \forall t \in T, \tag{22}$$

$$q_i \geq 0, \qquad \forall i \in N \cup N_0. \tag{23}$$

To solve LR(Π), we reformulate it into a set partitioning-like formulation, LR(Π, R), where R denotes a path set of network G and each path $r \in R$ connects an origin station-time node $o(r)$ to a destination station-time node $d(r)$. With battery power restrictions defined in (18) and (19), the maximum time length of each path is limited by Q_0, i.e., $t_{d(r)} - t_{o(r)} \leq Q_0$ holds for each $r \in R$. The weight of path r in the Lagrangian subproblem is determined by $\rho_r = \sum_{v\in V(r)} (w_v - \pi_v)$, where $V(r)$ represents the subset of vessels that are inspected in path r. Binary variable θ_r is defined for each $r \in R$. Let $\theta_r = 1$ if path r is operated, and otherwise let $\theta_r = 0$. Then, the set partitioning-like formulation is given by

$$\text{LR}(\Pi, R): \quad \max \quad \sum_{r \in R} \rho_r \theta_r + \sum_{v \in V} \pi_v, \tag{24}$$

$$\text{s.t.} \quad \sum_{r \in R: o(r) = i} \theta_r - \sum_{r \in R: d(r) = i} \theta_r = y_{u_i, t_i - 1} - y_{u_i, t_i}, \qquad \forall i \in N_0, \tag{25}$$

$$\sum_{r \in R: \{o(r) = i \text{ or } d(r) = i\}} \theta_r \leq 1, \qquad \forall i \in N_0, \tag{26}$$

$$\theta_r \in \{0, 1\}, \qquad \forall r \in R, \tag{27}$$

$$0 \leq y_{k,t} \leq |M|, \qquad \forall k \in K, \forall t \in T. \tag{28}$$

Note that constraints (17)–(19) are not included in $\text{LR}(\Pi, R)$ since they are naturally satisfied for each generated path in R. Constraints (25) are reformulated from (16), which calculate the number of drones remained at each station-time node. Constraints (26) are the safety constraints reformulated from (20), which allow at most one path started or ended on each station-time node. Constraints (27)–(28) define the types and feasible domains of the decision variables, where each $y_{k,t}$ is also relaxed to be continuous without affecting its integral condition.

We next solve the Lagrangian subproblem $\text{LR}(\Pi, R)$ to derive a valid upper bound on the optimal solution of (F). Note that solving $\text{LR}(\Pi, R)$ by enumerating all the possible paths of R is intractable because the size of R can be extremely large. Let $l^*(i, j, \Pi)$ denote the largest weight path from station-time node i to station-time node j, where arc weights of the underlying network are determined according to vector Π. Let $L^*(\Pi) = \{l^*(i, j, \Pi) | i \in N_0, j \in N_0, 0 < t_j - t_i \leq Q_0\}$ denote a path set that only includes the largest weight paths for all possible station-time node pairs. If there are multiple largest weight paths, all of them are included in $L^*(\Pi)$. Now, we have Proposition 3 to show that the optimal solutions of $\text{LR}(\Pi, R)$ and $\text{LR}(\Pi, L^*(\Pi))$ must have equal objective values.

Proposition 3. *Optimal value of LR*$(\Pi, L^*(\Pi))$ *is equal to the optimal value of LR*(Π, R).

Proof. We prove this proposition by showing an evidence that the paths selected in any optimal solution of $\text{LR}(\Pi, R)$ must be included in $L^*(\Pi)$.

Note that there are at most one path started or ended on each station-time node (i.e., the safety constraints), therefore, any feasible solution of $\text{LR}(\Pi, R)$ allows at most one path that can be operated between each two station-time nodes. Suppose for some pair of station-time nodes, a path r excluded from $L^*(\Pi)$ is selected in an optimal solution of $\text{LR}(\Pi, R)$. Then, we can replace this path by the largest weight path $l^*(o(r), d(r), \Pi)$ to form a new solution without violating any constraints of $\mathrm{LR}(\Pi, R)$ while having the solution objective improved because $\rho_{l^*(o(r),d(r),\Pi)} > \rho_r$. This newly formed solution contradicts the original assumption that an optimal solution involves a path excluded from $L^*(\Pi)$, and this proposition is thus proved by contradiction. $\square$

According to Proposition 3, obtaining the optimal value of $\mathrm{LR}(\Pi, R)$ can be realized in two phases, where in the first phase we construct the largest weight path set $L^*(\Pi)$ whose size is roughly proportional to the square of the station-time node number in G, and in the second phase we solve $\mathrm{LR}(\Pi, L^*(\Pi))$ based on the provided path set $L^*(\Pi)$. Since G is an acyclic network, the largest weight path from any $i \in N_0$ to $j \in N_0$ with $0 < t_j - t_i \le Q_0$ can be found in $O(|V|^2|Q_0|)$ using a dynamic programming approach (see Proposition 4). Given $L^*(\Pi)$, we employ an existing MILP solver to find an optimal solution of $\mathrm{LR}(\Pi, L^*(\Pi))$. According to our preliminary numerical tests, $\mathrm{LR}(\Pi, L^*(\Pi))$ can be solved quickly by CPLEX because of the limited size of $L^*(\Pi)$.

Proposition 4. *A dynamic programming approach finds the largest weight path from any $i \in N_0$ to $j \in N_0$ with $0 < t_j - t_i \le Q_0$ in a time complexity of $O(|V|^2|Q_0|)$.*

Proof. According to the construction of G under Proposition 1, each vessel-time node can be linked by only one vessel-time node for each different previously inspected vessel. Therefore, for each vessel-time node, there can be at most $|V| - 1$ candidate arcs linked by previous vessel-time nodes. In the dynamic programming, since G is acyclic in the time dimension, we can update the largest weight partial path ended with each vessel-time node in a total of $|t_j - t_i|$ separated steps along with the increase of time points. Given that each step involves the update of the largest weight partial paths on $|V|$ nodes with respect to the same time point and the update on each node requires at most $|V|$ candidate exploration checks, time complexity $O(|V|^2)$ is hence needed for one step. Because of the battery power restriction, i.e., $0 < |t_j - t_i| \le Q_0$, the proposed dynamic programming procedure can proceed for at most $|Q_0|$ steps. Therefore, the overall time complexity of the dynamic programming is $O(|V|^2|Q_0|)$. $\square$

5.2. Lower bound solution

The optimal value of $\mathrm{LR}(\Pi)$ is an upper bound on the optimal value of (F), which can be used to gauge the quality of a feasible solution (or lower bound solution) in our problem. Based on the progressive information of solving a Lagrangian subproblem, we next discuss a solution strategy on finding feasible solutions of (F), which is integrated as a lower bounding step in the Lagrangian relaxation-based method.

It is worth noting that, after relaxing (10), solution of a Lagrangian subproblem may result in an infeasible path which inspects a vessel several times. Therefore, in the set partitioning-like formulation, not all paths in R and even in $L^*(\Pi)$ can constitute a feasible solution to (F). Ideally, if an optimal solution of $\mathrm{LR}(\Pi, L^*(\Pi))$ is obtained to include a set of operated paths that do not violate any constraints of (10), these paths constitute an optimal solution to (F). By this property, we are motivated to find lower bound solutions based on those feasible paths that are already in the set $L^*(\Pi)$. Let $\bar{L}^*$ denote the set of feasible paths in $L^*(\Pi)$, the paths of which are ruled out by executing a feasibility check on each path of $L^*(\Pi)$. Given $\bar{L}^*$, a lower bound solution can be derived by solving the following MILP formulation:

$$\mathrm{LB}(\bar{L}^*): \quad \max \sum_{r\in\bar{L}^*} \rho'_r\theta_r, \tag{29}$$

$$\text{s.t.} \quad \sum_{r\in\bar{L}^*:o(r)=i} \theta_r - \sum_{r\in\bar{L}^*:d(r)=i} \theta_r = y_{u_i,t_i-1} - y_{u_i,t_i}, \qquad \forall i \in N_0, \tag{30}$$

$$\sum_{r\in\bar{L}^*} \sum_{i\in N(r):u_i=v} \theta_r \leq 1, \qquad \forall v \in V, \tag{31}$$

$$\sum_{r\in\bar{L}^*:o(r)=i \text{ or } d(r)=i} \theta_r \leq 1, \qquad \forall i \in N_0, \tag{32}$$

$$\theta_r \in \{0,1\}, \qquad \forall r \in \bar{L}^*, \tag{33}$$

$$0 \leq y_{k,t} \leq |M|, \qquad \forall k \in K, \forall t \in T. \tag{34}$$

where $\rho'_r = \sum_{v\in V(r)} w_v$ and $N(r)$ is a subset of nodes visited by path r. Other parameters and variables used in $\mathrm{LB}(\bar{L}^*)$ are the same as those defined for $\mathrm{LR}(\Pi, L^*(\Pi))$. Similar with the solution to $\mathrm{LR}(\Pi, L^*(\Pi))$, $\mathrm{LB}(\bar{L}^*)$ can also be quickly solved by CPLEX because $\bar{L}^*$ is a further reduced set of $L^*(\Pi)$ that is known to be with a limited size.

5.3. A subgradient optimization procedure

Given any vector Π, an upper bound on the optimal value of (F) can be derived by solving a Lagrangian subproblem $\mathrm{LR}(\Pi)$, and a feasible solution (i.e., a lower bound) of (F) can be obtained by solving $\mathrm{LB}(\bar{L}^*)$. Next, we apply a well-known subgradient optimization procedure to finding near-optimal multipliers for the Lagrangian dual problem. The algorithmic procedure is presented in Algorithm 1.

In a real practice, because of the dynamic change of vessel's status, prediction error on a vessel's real-time information can be enlarged if execution of the plan is delayed a lot. Therefore, the proposed method is required to be capable of finding good lower bound solutions within a particular limited period of time, in order to guarantee effectiveness of the planning solution.

In this regard, at the lower bounding step, we allow $\bar{L}^*$ to accumulate feasible paths of $L^*(\Pi)$ from multiple consecutive iterations. Let $\bar{L}^*_\delta$ be the set of feasible paths in $L^*(\Pi^\delta)$, where Π^δ is the

Lagrangian multiplier vector in the δth iteration. In each iteration of the subgradient algorithm, we construct the feasible path set by $\bar{L}^* = \bar{L}^* \cup L^*(\Pi^\delta)$ in Step 2. When $\bar{L}^*$ grows to a considerable size that leads to a significantly slow computation, we refresh $\bar{L}^* = \varnothing$. In the algorithm, the size of $\bar{L}^*$ is restricted by $|\bar{L}^*| \leq 3|L^*(\Pi)|$, according to our preliminary tests showing that adding any more paths over that size barely improves the lower bound.

Algorithm 1 Subgradient optimization procedure

Step 1. Initialization. Multipliers in Π are initialized by zeros. The best-known upper bound and lower bound are initialized by $ub = +\infty$ and $lb = -\infty$. Iterator is set by $\delta = 0$;

Step 2. Upper bounding. Let Π^δ denote the vector of multipliers used in the δth iteration. Solve LR(Π^δ) and obtain its optimal solution objective value as $ub(\Pi^\delta)$. Update the best-known upper bound by $ub = ub(\Pi^\delta)$ if a smaller upper bound is found. Set $\bar{L}^* = \bar{L}^*(\Pi^\delta)$;

Step 3. Lower bounding. Solve LB($\bar{L}^*$) to gain a lower bound solution whose objective value is obtained as lb^δ. Update the best-known lower bound by $lb = lb^\delta$ if a better solution is detected;

Step 4. Multiplier updating. If constraints (10) are violated by the optimal solution of LR(Π^δ), update $\Pi^{\delta+1} \leftarrow \Pi^\delta$ using the formula below

$$\pi_v^{\delta+1} \leftarrow \max\left\{0, \pi_v^\delta + \frac{\lambda\left(ub(\Pi^\delta) - lb\right)}{\|\Delta\|^2}\Delta_v\right\}, \tag{35}$$

where given $\hat{x}_{i,j} \in \mathbf{x}^\delta$ as the arc-design variables of the optimal solution to LR(Π^δ), $\Delta_v = 1 - \sum_{i \in N \cup N_0} \sum_{j \in N: u_j = v} \hat{x}_{i,j}$ and $\|\Delta\|^2 = \sum_{v \in V} (\Delta_v)^2$. The constant parameter λ is a scalar chosen between 0 and 2. We set $\lambda = 1$ in this work;

Step 5. Stop criteria checking. The algorithm stops if at least one of the following conditions is activated: (i) Constraints (10) are satisfied by $\mathbf{x}^\delta$; (ii) Iterator exceeds a maximum number $\delta_{\max}$; (iii) Running time of the algorithm exceeds a predefined time limit $t_{\max}$; (iv) Gap between lb and ub is small enough. Otherwise, set $\delta \leftarrow \delta + 1$ and move to Step 2 for the next iteration.

In addition, the subgradient optimization procedure may sometimes oscillate between two non-optimal vectors of multipliers, causing the iterated upper bound values fail to converge. To avoid this situation, we half the value of the scalar λ in (35) if current best upper bound does not improve for a certain number of iterations.

6. Numerical experiments

This section reports on the results of the experiments conducted to show the effectiveness and efficiency of the Lagrangian relaxation-based method. The solution algorithm is implemented in C using the CPLEX of version 12.6. Experiments are performed on an Intel Xeon (2.1 GHz) Desktop PC with 16 GB RAM.

In Section 6.1, we introduce the test data generated for the DSP. An overview on the performance of the Lagrangian relaxation-based method is shown in Section 6.2, and its performance on obtaining fast solutions is reported in Section 6.3. Section 6.4 involves a sensitivity analysis on solutions produced in different parametric settings. A case study based on realistic data of vessel locations is provided in Section 6.5. The effectiveness of our solution in response to uncertain vessel locations is examined in Section 6.6.

6.1. Generation of test data

We generate the test data based on practical conditions of the Pearl River Delta (PRD) of China[1]. Relevant data for the ECA, the vessels to be inspected, the base stations, and the deployed drones are defined and generated as follows.

Figure 7: Map of the ECA in Pearl River Delta

ECA. The ECA considered in our experiment is based on the case of PRD, which is roughly shaped as a rectangular zone with a length of 170 nautical miles and a width of 20 nautical miles (North of England P&I Association 2018) as shown in Figure 7. We represent this rectangular zone ranged horizontally by $[-85, 85]$ and vertically by $[0, 20]$ in a two-dimensional coordinate system. Vessels that are to be inspected are those new incoming vessels in the ECA, with each vessel planning to call one of the three core ports, including Hong Kong port (HK) in Hong Kong,

[1] China established three ECAs within its territorial waters in 2015, including the PRD ECA. The ECAs of China were established according to China's domestic laws rather than by the IMO. Therefore, they are not plotted in Figure 1.

Yantian port (YT) in Shenzhen, and Chiwan port (CW) in Shenzhen. According to the real geographical locations of the ports, we fix their coordinates in the two-dimensional coordinate system as HK-$(20, 0)$, YT-$(40, 0)$, CW-$(0, 0)$, respectively.

Vessels. New incoming vessels that sail inside the ECA to call any one of the above three ports are candidates for the emission inspection. At the beginning of a planning period, we suppose vessels are randomly located in the rectangular zone, which follows an even distribution. We assume vessels in the ECA generally sail in a constant speed between 5 and 10 knots, and real-time locations of a sailing vessel are roughly tracked by a straight path directing from the vessel's current location to its destination location for the port call. Considering that calling a port normally lasts approximately 6–48 hours, which is usually longer than the length of a planning time period (i.e., 5 hours in our case), we suppose all target vessels in the ECA have identical time windows $[0, T_{\max}]$ for the inspection. We also generate the importance weight (or revenue) of inspecting a vessel by a random integer value in $[5, 15]$.

Stations and drones. The base stations of a drone are located along the coastline of the PRD. In the two-dimensional coordinate system, a station's vertical coordinate is fixed to be 0 and its horizontal coordinate is randomly determined within $[-85, 85]$, which follows an even distribution. We assume a station initially stores five drones, and each drone can fly at a maximum speed of 30 knots for at most 120 minutes. Moreover, the time for inspecting each vessel's emission is set to 5 minutes. The time for replacing a drone's battery is also assumed to be 5 minutes.

Table 1: Description of the generated instances in each class

ClassID	$\lvert V\rvert$	$\lvert K\rvert$	$\lvert M\rvert$	$\lvert N \cup N_0\rvert$	$\lvert A\rvert$	$\lvert L^*\rvert$
v20k1	20	1	5	6300	85918	10017
v20k2	20	2	10	6600	93197	17267
v20k3	20	3	15	6900	100418	32772
v40k1	40	1	5	12300	339209	14682
v40k2	40	2	10	12600	355852	49469
v40k3	40	3	15	12900	370260	67411
v60k1	60	1	5	18300	765071	12189
v60k2	60	2	10	18600	790915	41102
v60k3	60	3	15	18900	809579	85862
v80k1	80	1	5	24300	1358247	18685
v80k2	80	2	10	24600	1382376	35975
v80k3	80	3	15	24900	1412064	82570
v100k1	100	1	5	30300	2134459	19609
v100k2	100	2	10	30600	2178150	73806
v100k3	100	3	15	30900	2210219	94374

Consider a planning period of 5 hours (i.e., $T_{\max} = 300$ minutes), which has a gap of 1 minute between each two adjacent time points. Test instances are generated and clustered in different classes, including the number of vessels $|V| \in \{20, 40, 60, 80, 100\}$ and the number of stations $|K| \in \{1, 2, 3\}$. We denote as "vakb" a class of instances that are associated with a vessels and b stations. Each class contains five randomly generated instances.

Table 1 summarizes the information averaged over the five instances for each class. The number of vessels, number of stations, and number of deployed drones are shown in columns "$|V|$," "$|K|$," and "$|M|$," respectively. Columns "$|N \cup N_0|$" and "$|A|$" report the average node and arc number of each time-expanded network constructed by an instance in each class. Column "$|L^*|$" reports the average number of paths that are generated in $L^*(\Pi)$. All these data are used to measure the scale of the test instances generated for each class.

6.2. Overview of the solution performance

For the Lagrangian relaxation-based method proposed in Section 5, we now compare its performance with a benchmark method that uses an optimization solver to directly solve the model (F). We adopt ILOG CPLEX 12.6 as the optimization solver. The Lagrangian relaxation-based method stops if the number of iterations exceeds 1000 or the percentage gap between the best-known upper bound and lower bounds is smaller than 0.1%. Both methods are confined to a run-time limit of 7200 seconds.

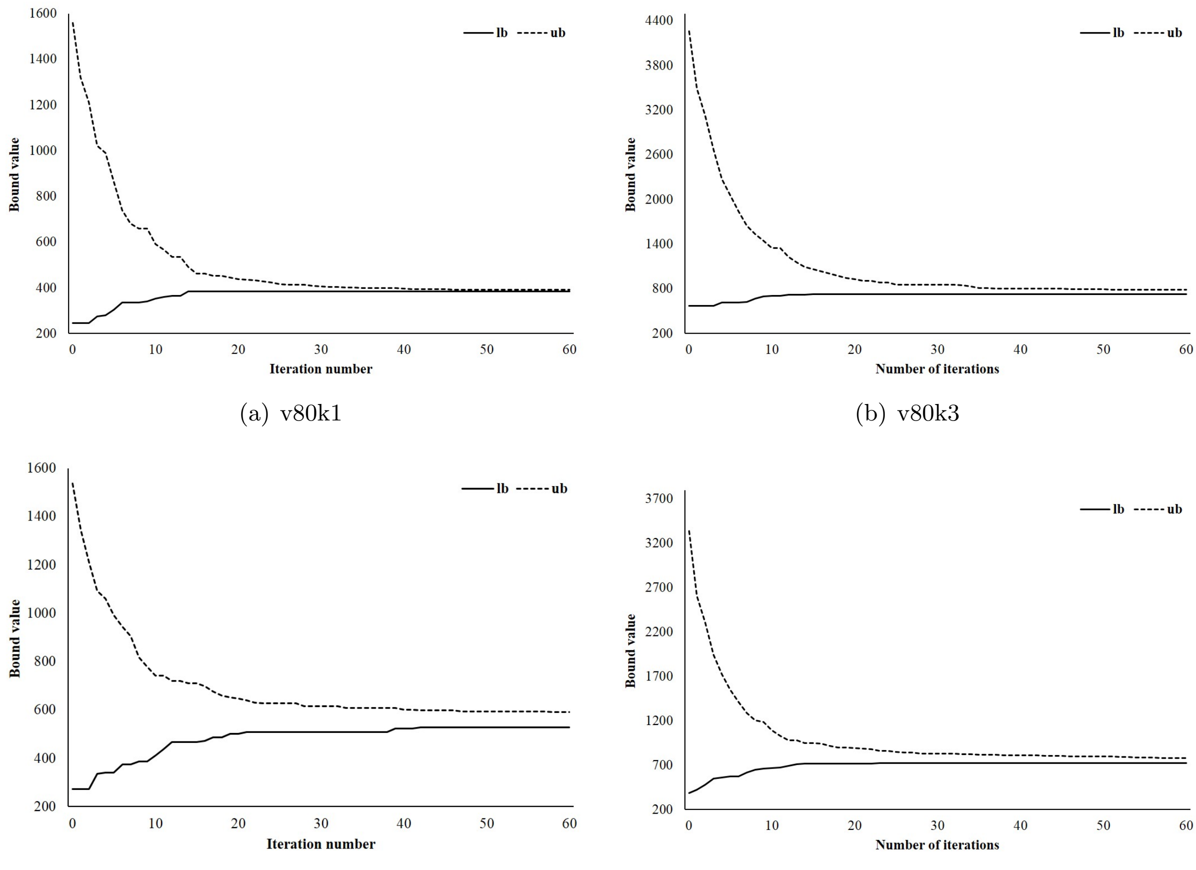
Table 2 compares the results produced by the Lagrangian relaxation-based method and by CPLEX. For each method, we report in column "Obj" the average solution objectives over the five test instances for each class, and best objectives are marked in bold. Column "Gap(%)" reports the average percentage gap between the best-known lower bound lb and the best-known upper bound ub found by each method (i.e., Gap(%)$= 100(ub - lb)/ub$). Column "Time" reports the computation time (in seconds) used by each method for the solution. We report in column "Opt/Fea" the number of instances that are solved for proven optimality (Opt) / the number of instances that are solved with feasible solutions (Fea) by CPLEX for each class. For the Lagrangian relaxation-based method, we report in column "MaxGap(%)" the maximum gap value of $100(ub - lb)/ub$ obtained among the five instances for each class and in column "Iter" the average number of iterations that are processed by the subgradient optimization procedure for the solution. The symbol "–" indicates that no feasible solutions are found within the time limit.

Table 2 shows that the Lagrangian relaxation-based method solves most of the test instances with up to 60 vessels to near-optimality (less than 1%) within the time limit, significantly outperforming CPLEX. In smaller classes of instances with 20 and 40 vessels, although CPLEX can sometimes obtain the same feasible solutions as those obtained by Lagrangian relaxation-based method, few of these solutions are proven to be optimal within the time limit. This shortcoming stems from the fact that CPLEX can hardly derive even acceptable upper bounds for these instances, thus leading to large average percentage gaps that range from 10.67% to 86.24%. For some classes of instances with 60 vessels and all the classes of instances with 80 and 100 vessels, CPLEX fails to obtain any feasible solutions within the given time limit. By contrast, for these larger instances, the Lagrangian relaxation-based method can still obtain solutions with an average percentage gap mostly no greater than 5%. For the largest class of instances v100k3 which are also the hardest set of instances in the experiment, the Lagrangian relaxation-based method can also achieve acceptable solutions with an average percentage gap of approximately 7% within the time limit.

Table 2: Solution performance on solving the instances for each test class

ClassID	CPLEX				Lagrangian				
	Obj	Gap(%)	Time	Opt/Fea	Obj	Gap(%)	MaxGap(%)	Time	Iter
v20k1	**81.6**	38.02	7200	0/5	**81.6**	0.16	0.28	433	211
v20k2	**144.4**	30.41	7200	0/5	**144.4**	0.16	0.28	866	306
v20k3	**172.8**	11.63	4342	2/5	**172.8**	0.28	0.58	1683	256
v40k1	**184.2**	54.27	7200	0/5	**184.2**	0.10	0.10	757	136
v40k2	243.4	42.08	7200	0/5	**244.6**	0.54	1.23	6804	331
v40k3	**354.2**	10.67	6324	1/5	**354.2**	0.53	0.71	7200	219
v60k1	**144.4**	86.24	7200	0/5	**144.4**	0.17	0.31	1521	402
v60k2	n/a	n/a	7200	0/0	**308.4**	0.86	2.38	6886	308
v60k3	n/a	n/a	7200	0/0	**308.4**	1.63	3.32	7200	155
v80k1	n/a	n/a	7200	0/0	**274.2**	0.18	0.41	3025	389
v80k2	n/a	n/a	7200	0/0	**408.0**	0.71	2.06	6372	373
v80k3	n/a	n/a	7200	0/0	**448.0**	2.40	4.36	7200	141
v100k1	n/a	n/a	7200	0/0	**499.0**	4.77	10.56	7200	194
v100k2	n/a	n/a	7200	0/0	**570.6**	5.62	8.32	7200	75
v100k3	n/a	n/a	7200	0/0	**719.0**	7.05	8.52	7200	66

Table 2 also indicates that the classes of instances with more stations are harder than those with fewer stations. This conclusion is based on the observation that average percentage gaps and average computation times both increase in the number of $|K|$ among the solved classes of instances with the same $|V|$. For example, as shown in Table 2, for the classes of instances with 60 vessels and with the number of stations increased from 1 to 3, the average percentage gap increases from 0.17% to 1.63% and the average computation time increases from 1521 seconds to 7200 seconds. This result is attributed to the fact that when $|K|$ is larger, the time-expanded network has more origin-destination node pairs, which lead to additional computation time on running a dynamic programming to obtain a larger path set L^* in each iteration of the subgradient optimization procedure. These variations of the sizes of $|L^*|$ based on different $|K|$ can also be seen from the final column of Table 1. Moreover, when $|L^*|$ increases, the lowering bounding step of the Lagrangian relaxation-based method will probably require CPLEX to run an MILP with a larger path set $\bar{L}^*$, which also slows down the convergence of the overall solution algorithm.



(a) v80k1

(b) v80k3

(c) v100k1

(d) v100k3

Figure 8: Convergence of the Lagrangian relaxation-based method on solving the instances with 80 and 100 vessels

We next examine the convergence of the Lagrangian relaxation-based method for solving the large instances having 80 and 100 vessels. Figure 8 illustrates the variations of the best-known lower and upper bounds obtained for the first 60 iterations. Four instances respectively selected from v80k1, v80k3, v100k1, and v100k3 are illustrated in the figure. The results show that the Lagrangian relaxation-based method for solving these large instances can converge rapidly at early iterations because of the constant increase of lower bounds and the decrease of upper bounds. The lower bound is rarely improved after around 20 iterations, whereas the upper bound can be further improved until the bound gap is close enough. This finding implies that, although the convergence rate of the Lagrangian relaxation-based method may depend on the scale of the instances (as seen in Table 2), its convergence effect at each iteration is independent of the instance's scale. The convergence condition likewise suggests that we practically only need to run the Lagrangian relaxation-based method for a limited number of iterations (e.g., 20 observed from Figure 8) for solution output, because the obtained best-known lower bound solution after that number of iterations is already close to those further solutions that can be potentially detected by continuing the iterative optimization procedure.

6.3. Performance of the Lagrangian relaxation-based method on finding fast solutions

To prove computationally that the proposed Lagrangian relaxation-based method is of practical interest in producing quick and good solutions, we further examine its performance on finding lower bound solutions in a small number of iterations. Table 3 compares the results produced by the Lagrangian relaxation-based method with different maximum iteration number restrictions (subject to 5, 10, and 20 iteration) and by CPLEX with different computation time limits (subject to 600 and 1800 seconds). In each solution scenario, we report in columns "Obj" and "Time" the average solution objective and average computation time for each class of instances, respectively. Column "Gap(%)" reports the average percentage gap between the best-known lower bound found by the Lagrangian relaxation-based method within the designated iterations and the best-known upper bound found by the method in 7200 seconds. The average solution objectives obtained by CPLEX with time limits of 600 and 1800 seconds are reported in columns "Obj(600s)" and "Obj(1800s)," respectively. In these two columns, we report "(n)" if there are $n \in \{1, 2, 3, 4\}$ instances that are solved with feasible solutions for an instance class. We report symbol "–" if no feasible solutions are found. The best objective value is marked in bold in the table.

As shown in Table 3, for the smallest classes of instances with 20 vessels, both CPLEX and the Lagrangian relaxation-based method can quickly find lower bound solutions with equal objective values. When the number of vessels increases to 40, feasible solutions are barely found by CPLEX within 1800 seconds. However, the Lagrangian relaxation-based method can still find good solutions with Gap(%) less than 1% in at most few minutes. For more and larger classes of instances for which CPLEX cannot find any feasible solutions, the Lagrangian relaxation-based method can also quickly output acceptable solutions after 10 iterations for most of the instances with no more than 80 vessels. Their average computation times required for processing the 10 iterations are no greater than 10 minutes. For the largest classes of instances with 100 vessels, the Lagrangian relaxation-based method can still solve the instances with $|K| = 1$ very efficiently. For the remaining instances with $|K| \in \{2, 3\}$, the Lagrangian relaxation-based method is able to output quick solutions with

Table 3: Compare the Lagrangian relaxation-based method and CPLEX on obtaining fast solutions

ClassID	Lagrangian (Iter=5)			Lagrangian (Iter=10)			Lagrangian (Iter=20)			CPLEX	
	Obj	Gap(%)	Time	Obj	Gap(%)	Time	Obj	Gap(%)	Time	Obj(600s)	Obj(1800s)
v20k1	**81.6**	0.16	5	**81.6**	0.16	11	**81.6**	0.16	24	**81.6**	**81.6**
v20k2	**144.4**	0.16	14	**144.4**	0.16	28	**144.4**	0.16	58	**144.4**	**144.4**
v20k3	**172.8**	0.28	27	**172.8**	0.28	52	**172.8**	0.28	108	**172.8**	**172.8**
v40k1	**184.2**	0.09	12	**184.2**	0.09	34	**184.2**	0.09	80	-	(4)
v40k2	241.2	1.76	57	243.4	0.97	169	**244.6**	0.54	465	-	(4)
v40k3	353.0	0.87	116	**354.2**	0.52	317	**354.2**	0.52	754	-	-
v60k1	143.2	1.34	13	**144.4**	0.17	33	**144.4**	0.17	68	-	(4)
v60k2	295.6	5.00	71	307.2	1.26	203	**308.4**	0.85	463	-	-
v60k3	295.6	5.74	195	307.2	2.03	513	**308.4**	1.63	1172	-	-
v80k1	235.2	12.93	17	265.4	2.46	47	**274.2**	0.18	138	-	-
v80k2	358.4	11.72	45	395.2	3.05	128	**408.0**	0.71	383	-	-
v80k3	410.8	9.08	107	436.4	4.70	332	**447.6**	2.47	997	-	-
v100k1	343.8	34.28	24	413.8	20.72	60	**480.8**	8.18	204	-	-
v100k2	420.0	30.59	111	502.6	16.05	324	**559.0**	7.54	1170	-	-
v100k3	547.2	29.26	123	659.8	14.70	414	**711.2**	8.05	1484	-	-

average percentage gaps ranging from 14.70% to 16.05% in 10 iterations. However, to obtain better solutions (with an average percentage gap less than 10%), the algorithm needs more than 1000 seconds to process a total of 20 iterations for the solution.

Results in Table 3 computationally demonstrate that our proposed Lagrangian relaxation-based method is capable of producing fast and good solutions for practical use. For the classes of instances with no more than 80 vessels, desirable feasible solutions with close optimality gaps (less than 3% on average) can be obtained in 20 iterations.

6.4. Sensitivity tests for solutions produced in different settings

In the DSP, many input parameters are assumed to be known in advance. Sensitivity analysis on some parameters, such as the endurance of a drone's battery Q and the number of deployed drones $|M|$ for the planning, are of real practical interest to be studied.

As mentioned earlier, a bottleneck in the use of drones for monitoring vessels in the ECA is induced by the limit of battery's endurance. In general, using durable drones can cover more vessels for the inspection, i.e., a vessel that is originally too far to inspect can now be included in a longer scheduled tour. Figure 9 illustrates the variation of the solution objectives based on different endurance levels of a drone, where $Q \in \{30, 60, 90, 120, 150, 180\}$. Results of the three instances selected from v60k1, v80k1, and v100k1 are reported and tracked by dotted, dashed, and solid lines, respectively, in the figure.

Figure 9 shows that the total importance weight from finished inspections is obviously increasing in the value of Q. The trend of the curves show that the obtained solution can be more significantly improved when Q increases from 30 to 60 minutes in the instance of v100k1. This result implies that the practitioners should be motivated to upgrade their drones to be more enduring if the drones' original endurance is limited. Moreover, given that the drones can fly for 60 minutes, increasing the drone's endurance ability can be still helpful in increase the effectiveness of inspections for all the three instances, and the increment of effectiveness is roughly proportional to the endurance ability of a drone.

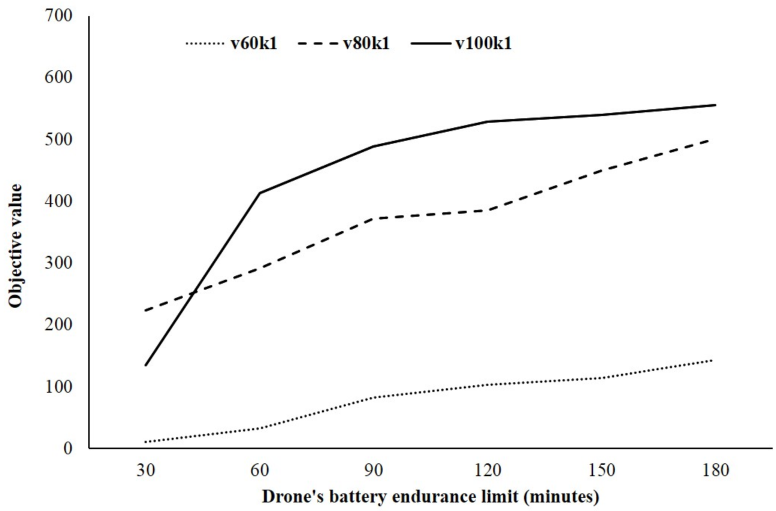


Figure 9: Variation of solution objectives based on different battery endurance limits

We also conduct sensitivity tests for the solutions based on different numbers of deployed drones. Figure 10 illustrates the variations of solution objectives of three instances that are selected from v60k1, v80k1, and v100k1 for the tests. In each instance, the number of deployed drones varies from 1 to 7. The variations in solution objectives are tracked by dotted, dashed, and solid lines, respectively for, the instances of v60k1, v80k1, and v100k1.

Figure 10 shows that deploying more drones in the system can help increase the solution objective. However, the improvement effect weakens for all cases. Rationally, there exists a maximum size for the fleet of drones, in which case deploying any additional drone will no longer improve the objective. As shown in Figure 10, for the instance of v60k1, the solutions with respect to more than two deployed drones roughly have the same solution objective. For the instance of v80k1, the maximum size for the fleet of drones should be around four, after which the solution objective no longer improves. For the instance of v100k1, the objective value continues improving in the number of deployed drones because more vessels are to be inspected. Hence, the maximum size for the fleet of drones must reach a number greater than seven. The solution sensitivity testing toward different numbers of deployed drones can also be employed to determine an economic size for the drone fleet, which is a trade-off number by considering the improved effectiveness of inspections and the increased cost brought by deploying any additional drone in the system.

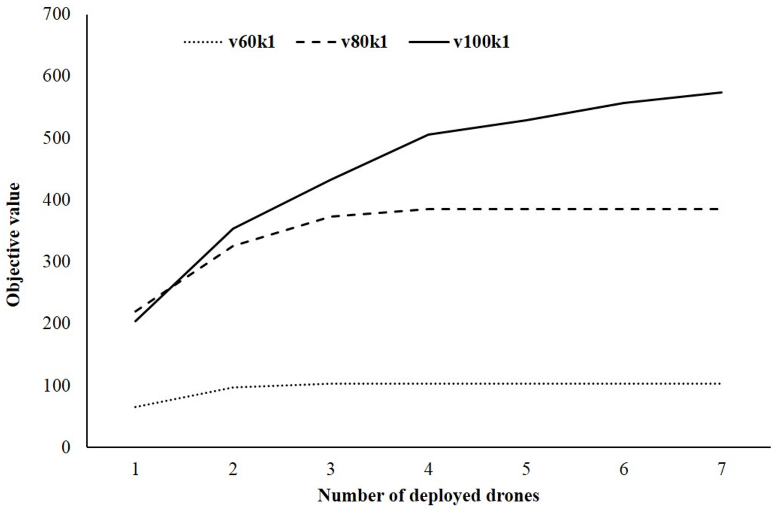


Figure 10: Variation of solution objectives based on different deployed drone numbers

6.5. A case study based on realistic vessel locations

A case study based on realistic vessel location data is provided to examine the DSP solution. We capture the location data of 20 vessels sailing inside the PRD zone from the automatic identification system, with the time between 7am and 5pm on the day of April 23, 2018. The real-time location of each vessel is collected every 3 minutes. The time period is discretized by 200 time units. Table 4 presents the vessel information, where columns "e_v" and "l_v" are known as the earliest and latest time in the planning period to inspect each vessel. Location data is recorded by longitude and latitude coordinates. We report in columns "(long, lat)$_1$" and "(long, lat)$_2$" the real-time longitude and latitude coordinates of each vessel at e_v and at l_v. The importance weight to inspect each vessel is given in column "w_v".

Based on the realistic data, we consider three scenarios to test the DSP solutions. The first is a base scenario considering a single drone base station near HK, whose (long, lat) coordinates are given as (114.2, 22.2), and two drones are deployed. The second scenario is extended from the base scenario, in which we enforce that each scheduled tour of drone has only one vessel to inspect. Solution of this scenario can be obtained by running our algorithm based on a modified time-expanded network, where all arcs between vessel-time nodes are removed. In the third scenario, we include an additional base station near CW, with its (long, lat) coordinates fixed as (113.9, 22.5). No extra drones are initially assigned to CW. In all scenarios, suppose each drone can fly at a maximum speed of 50 knots. Drone's endurance time is set to 180 minutes (i.e., 60 time units). The time for battery replacement and for inspecting each vessel's emission are both set to 6 minutes (i.e., 2 time units). To keep safety, only one takeoff or landing operation is allowed within each unit time gap. The flying distance between any two locations is computed based on the longitude-latitude system.

Table 4: Vessel location information from the automatic identification system

vesID	e_v	l_v	(long, lat)$_1$	(long, lat)$_2$	w_v
1	0	200	(115.7648, 22.1244)	(113.9978, 21.9606)	9
2	0	200	(113.6590, 22.6906)	(113.6590, 22.6906)	6
3	22	200	(112.2582, 21.0000)	(113.6992, 21.6238)	12
4	0	200	(115.1477, 22.6217)	(115.1477, 22.6223)	11
5	107	152	(115.6915, 21.7184)	(114.6279, 21.3918)	10
6	0	200	(113.2773, 21.8341)	(113.2772, 21.8342)	5
7	0	200	(114.1736, 22.1468)	(114.3988, 21.8596)	13
8	0	200	(113.6650, 22.7431)	(113.6650, 22.7431)	9
9	0	200	(113.9115, 21.9928)	(113.9110, 21.9928)	14
10	0	105	(114.9545, 22.1662)	(115.9969, 22.4363)	7
11	105	200	(115.9936, 22.3878)	(115.0738, 22.0925)	13
12	0	200	(113.9138, 22.0128)	(113.9133, 22.0122)	12
13	0	200	(113.1858, 21.9587)	(113.1858, 21.9587)	6
14	0	200	(113.6712, 22.7367)	(113.6712, 22.7368)	12
15	0	114	(115.0276, 22.1681)	(115.9687, 22.3358)	14
16	0	85	(115.1677, 22.2293)	(115.9931, 22.4543)	12
17	0	200	(113.5007, 23.0596)	(114.0402, 21.9674)	7
18	0	200	(114.7369, 22.6056)	(114.7370, 22.6056)	9
19	0	200	(113.5733, 22.8176)	(113.5735, 22.8176)	14
20	0	200	(113.5373, 22.9935)	(113.5373, 22.9935)	5

Table 5 illustrates the solutions of the three scenarios, each of which is obtained by running the Lagrangian relaxation-based method for at most 20 iterations. The designed tours in each solution have information of IDs (i.e., A1 or B2) and sequences of (vessel/station, time) pairs. Each tour ID indicates the operation sequence of the tour for some drone. For example, A1 indicates the first tour operated by drone "A". From Table 5, we see that there are eight tours designed to complete the inspections of 17 vessels out of the total 20 vessels in Scenario-1. The total weighted vessel number is obtained as 176, which is significantly better than the solution objective of Scenario-2 (129) and is slightly worse than that of Scenario-3 (181). The significant objective improvement from Scenario-2 to Scenario-1 demonstrates the strength of allowing to operate multiple inspection tasks in one tour for the solution. Moreover, the use of an extra drone base station also helps to improve the solution effectiveness, because more feasible inspection tours can be considered for the solution. For example, the vessel 20, which is not inspected in the solution of Scenario-1, can now be included by the tour A2, with CW being its destination station, in Scenario-3.

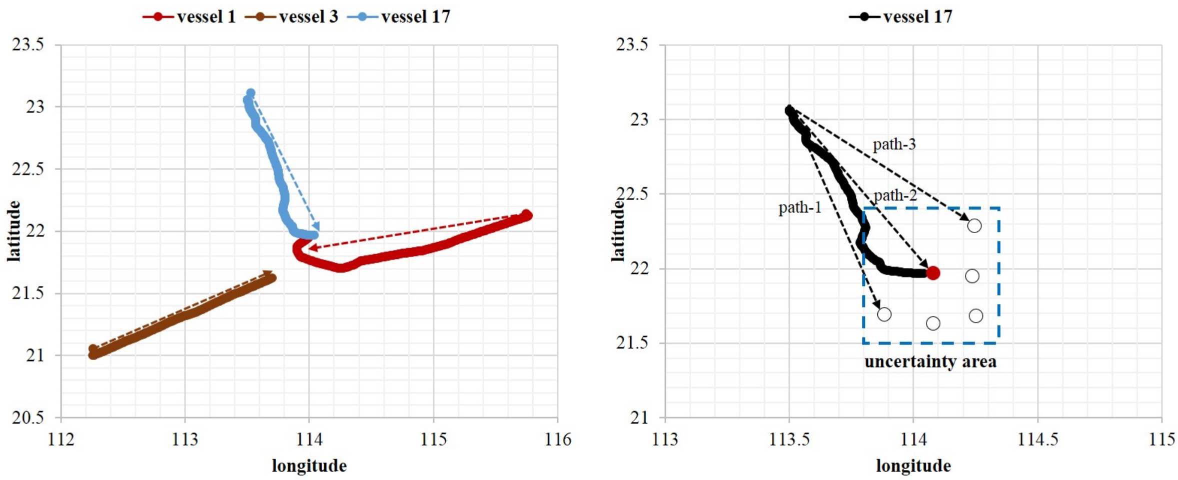
Table 5: Solutions based on different scenarios

Scenario-1 (Obj = **176**)	Scenario-2 (Obj = **129**)	Scenario-3 (Obj = **181**)
A1: (HK,1)→(16,32)→(HK,63)	A1: (HK,1)→(18,19)→(HK,37)	A1: (HK,1)→(16,32)→(HK,63)
A2: (HK,65)→(7,69)→(HK,73)	A2: (HK,38)→(12,48)→(HK,58)	A2 (HK,64)→(2,83)→(19,89)
A3: (HK,81)→(2,100)→(14,104)	A3: (HK,59)→(7,63)→(HK,67)	→(20,96)→(14,105)→(CW,115)
→(8,107)→(19,112)→(HK,135)	A4: (HK,68)→(19,91)→(HK,114)	A3 (CW,116)→(8,126)→(CW,136)
A4: (HK,137)→(6,162)→(3,173)→(HK,198)	A5: (HK,115)→(1,129)→(HK,143)	A4 (CW,137)→(17,144)→(6,163)
B1: (HK,3)→(10,29)→(15,33)→(HK,60)	A6: (HK,144)→(11,172)→(HK,200)	→(3,174)→(HK,199)
B2: (HK,61)→(18,79)→(4,91)→(HK,117)	B1: (HK,2)→(15,28)→(HK,54)	B1: (HK,2)→(15,28)→(10,32)→(HK,59)
B3: (HK,118)→(5,143)→(HK,168)	B2: (HK,56)→(9,67)→(HK,78)	B2: (HK,60)→(7,64)→(18,83)
B4: (HK,169)→(12,179)→(17,182)	B3: (HK,79)→(14,99)→(HK,119)	→(4,95)→(HK,121)
→(9,185)→(1,190)→(HK,200)	B4: (HK,120)→(17,134)→(HK,148)	B3: (HK,122)→(1,136)→(5,150)→(HK,174)
	B5: (HK,149)→(3,174)→(HK,199)	B4: (HK,176)→(12,186)→(9,189)→(HK,200)

Based on Scenario-3, we next look into how the initial allocation of drones to base stations influences the solution, when the total number of used drones are fixed. Suppose three drones are allocated to HK and CW, thus generating four different allocation combinations, i.e., (HK:3,CW:0), (HK:2,CW:1), (HK:1,CW:2), and (HK:0,CW:3). For the first two cases, where no less than two drones are allocated to the station of HK, the 20 vessels are all inspected in both solutions, obtaining the total weighted vessel number as 200. For the solution of (HK:1,CW:2), vessel 10 will not be covered for inspection, and the obtained weighted vessel number is reduced to 193. When all the three drones are allocated to the station of CW at the beginning, neither vessel 10 or vessel 16 is included in any inspection tour, resulting in the solution objective further reduced to 181. It is hence seen that the initial allocation of drones can be an active factor to influence the solution effectiveness. This fact motivates experienced practitioners to reposition drones during non-operating hours, so as to increase the number of inspected vessels.

6.6. Solution analysis considering the uncertainty of vessel locations

In the model, real-time locations of vessels are estimated based on their sailing courses and these estimations are assumed to be accurate over the planning horizon. In practice, the estimated paths may deviate from the actual paths during the planning horizon if vessels do not strictly follow their preset courses. Consequently, the solution derived from our model using the estimated vessel locations shall be revalidated for the actual realized locations. In this subsection, we examine the robustness of our method in response to the uncertainty of vessels' actual paths. The experiments are based on the practical dataset used in Section 6.5. We treat the real-time paths of vessels in the dataset as the actual paths, which are not used in our algorithm. When estimating vessels' real-time locations, we construct auxiliary straight-line paths to replace the actual paths. The obtained solution on the estimated vessel locations is then examined on the setting with actual vessel locations.



(a) Actual paths (b) Estimated paths

Figure 11: Illustration of the actual and estimated paths of vessels

The deviation of estimating vessels' actual locations is realized as follows: (i) Figure 11(a) illustrates the actual sailing paths of three vessels (vessel 1, vessel 3, and vessel 17 in Table 4) from the dataset, shown by the solid-line tracks, which are not used in our algorithm. Instead, we plot straight lines to connect their origin and destination locations, shown as dashed-line tracks, which are utilized to estimate the real-time location of each vessel at any time point of the planning horizon. (ii) To obtain a path with uncertain deviations, we identify the possible estimated sailing path based on the vessel's possible destination locations at the end of the planning horizon. Specifically, we define for each vessel a squared uncertainty area to capture its possible destination locations. As illustrated in Figure 11(b), for vessel 17 starting from its origin location $(113.5007, 23.0596)$, there are a set of candidate paths (i.e., path-1, path-2, and path-3) for estimation, each of which ends at a possible destination location within the given uncertainty area. Only one candidate path is chosen for each vessel for the estimation.

The central location of the squared area is denoted by (α_v, β_v). Each possible location in the squared area has its longitude in $[\alpha_v - \gamma \cdot d_v, \alpha_v + \gamma \cdot d_v]$ and its latitude in $[\beta_v - \gamma \cdot d_v, \beta_v + \gamma \cdot d_v]$, where d_v indicates the maximum estimation deviation and γ is a ratio to scale the coverage for this squared area. In the experiments, for each vessel v, we define the central location of the area by $\alpha_v = \alpha_v(l_v)$ and $\beta_v = \beta_v(l_v)$, where $(\alpha_v(l_v), \beta_v(l_v))$ is known as the last location of the vessel during the planning period (i.e., the red-point location in Figure 11(b)), namely, the location (long, lat)$_2$ of Table 4. For each sailing vessel v, we set $d_v = 1$ and $\gamma \in (0, 1)$. If a vessel keeps staying in a port area during the planning period, we set $d_v = 0$ showing that we are certain about the location of this vessel. To simulate our solution with the uncertainty, we estimate the path of each vessel using an auxiliary straight line which ends at a random location inside the uncertainty area. With the vessel locations estimated from these paths, we derive a solution using the Lagrangian relaxation-based algorithm. The obtained solution is recorded as a group of vessel inspection sequences, and then we examine the performance of operating these inspection sequences on the actual vessel locations. Note that, when operating on actual vessel paths, drones may miss some last vessels in the sequences and return to stations before running out of battery, due to the distance estimation gaps between the estimated and the actual vessel locations. In this case, we allow drones to skip those last inspection tasks and return to stations ahead of time.

Table 6: Analysis of the solution on estimated vessel paths with uncertain deviations

	Instance ID	Obj^a	Obj^e	Realized ratio	No. of vessels missed	No. of tours missed
$\gamma = 0.1$	E1	169	176	0.96	1	0
	E2	172	172	1.00	0	0
	E3	176	176	1.00	0	0
	E4	168	168	1.00	0	0
	E5	148	173	0.86	2	0
	Avg.			**0.96**	**0.60**	**0.00**
$\gamma = 0.3$	E6	164	175	0.94	1	0
	E7	170	170	1.00	0	0
	E8	158	171	0.92	1	0
	E9	153	173	0.88	2	0
	E10	154	179	0.86	2	0
	Avg.			**0.92**	**1.20**	**0.00**
$\gamma = 0.5$	E11	159	177	0.90	2	1
	E12	156	165	0.95	1	0
	E13	158	171	0.92	1	0
	E14	164	176	0.93	1	0
	E15	155	180	0.86	2	0
	Avg.			**0.91**	**1.40**	**0.20**

We perform the experiments following the Scenario-1 setting defined in Section 6.5, where one station and two drones are involved. Table 6 illustrates the solutions computed under different sets of estimated vessel paths. The uncertainty level γ is selected from $\{0.1, 0.3, 0.5\}$. We generate five instances for each selected value of γ, where the estimated vessel paths in each instance are determined randomly. For each instance, we report in columns "Obj^a" and "Obj^e" the solution objectives with respect to actual vessel paths and estimated vessel paths, respectively. The next column reports the realized ratio computed by ($\mathrm{Obj}^a/\mathrm{Obj}^e$). The last two columns report the number of vessels missed and the number of tours missed, by comparing the solution on actual paths and that on estimated paths.

As shown in Table 6, for the instances with $\gamma = 0.1$, about 96% of the total weighted vessel numbers with respect to the solution on estimated paths can be realized by operating the predetermined inspection sequences on actual vessel paths. This average ratio reduces to 92% and 91% for the groups of instances with $\gamma = 0.3$ and $\gamma = 0.5$, respectively. The reductions of the total collected weight from "Obj^e" to "Obj^a" mainly attributes to the fact that some planned vessel inspections are not operated, while the number of such missed inspections grows with the uncertainty level (γ) of the path estimation. Furthermore, given that the theoretically best objective is 176 (the result of Scenario-1 in Section 6.5), only one instance out of the 15 instances reaches the "optimal" objective value. Despite the loss of optimality, the values reported in column "Obj^a" still significantly outperforms the Scenario-2 solution (with objective 129) reported in 6.5, in which case drones are restricted to inspect only one vessel in each tour.

7. Discussion on two extended solutions for the DSP with uncertain vessel locations

In this study, we assume that the sailing speed and course of a vessel are fixed over the whole planning horizon. However, vessels' real-time locations are uncertain in practice, for instance, a vessel may occasionally adjust its sailing speed and course rather than stick to a given plan for safety or congestion reasons [e.g., avoidance of ship collisions (Qu et al. 2011, Weng et al. 2012)]. As seen in Section 6.6, the deviated estimations of vessel locations lead to some preset inspections being missed. In order to decrease this negative effect, one is encouraged to incorporate the uncertainty of vessel paths in designing the inspections tours. In this section, we briefly discuss two simple extensions of our method, a sample average-based solution and a rolling horizon-based solution, to capture the uncertainty of vessel locations.

7.1. A sample average-based solution

Sample average-based methods are widely applied in stochastic operation systems, such as seaport terminals, airports, and warehouses. Given a set of candidate solutions, the quality of each solution is repeatedly examined by a large number of random settings, and the solution with respect to the best overall performance is selected for the output. For the DSP, the randomness of the problem is mainly from the uncertain sailing paths of vessels. Suppose we have obtained 15 candidate solutions derived from different settings of path estimations (e.g., solutions of the instances E1-E15 in Table 6). Instead of using a single objective value to evaluate each solution, we randomly generate significantly many test instances (e.g., one thousand instances) for calculating the average objective for each of the 15 solutions. The evaluation of a given solution over each specific test instance can be realized by running the Lagrangian relaxation-based algorithm on a reduced time-expanded network, a similar case with the calculation of a solution based on actual vessel locations in Section 6.6.

7.2. A rolling horizon-based solution

Rolling horizon implementation is known to be effective in tackling discrete-time stochastic dynamic optimization problems in response to the uncertain future information (see, e.g., Sethi and Sorger 1991). The method resorts to a group of dynamic decisions according to the real-time information to reduce the impact of uncertainty on the solution. For the DSP, the inspection tours in a solution could be adjusted by multiple decisions that are made at different time points of the planning horizon. Drones can immediately follow the updated tours when their current vessel inspections are finished. For an example planning horizon with length of 5 hours, we can dynamically change each drone's inspection tours every 60 minutes, according to current vessels' sailing data from the tracking system. To do so, tours assigned to each drone are timely re-optimized and updated if the actual real-time locations of vessels are far from their previous estimations.

To implement the rolling horizon-based solution, at each time point when the information is updated, we have to solve a real-time DSP, where current statuses of drones and vessels shall be accessed for the input. In order to realize the real-time scheduling for drones, in the model we need to differentiate the flows of drones on the time-expanded network, such that locations and remaining working time of each operated drone can be captured. Specifically, to describe the real-time problem, we refer to each path as an inspection tour for a particular drone, that is, tour design variables in the model should be drone-specific (i.e., using binary variable $x^h_{i,j} \in \{0, 1\}$ to determine whether drone h passes arc $e(i, j)$ on the network). Moreover, the tour feasibility constraints are associated with specific drones as well, and a feasible path in the solution may start from a vessel-time node, which represents a partial sequence of inspection tasks that are operated by a working drone with remaining battery powers. Since the revised formulation of the real-time DSP is structurally similar with the model (F), we can also apply the Lagrangian relaxation-based method for the solution.

8. Conclusion

This paper examines a drone scheduling problem that develops for drones a set of scheduled tours to inspect vessels in emission control areas. To address the time-dependent locations of the sailing vessels, we construct a time-expanded network, and based on that develop for the problem a mixed-integer linear programming formulation. A Lagrangian relaxation-based method is proposed to solve larger instances for the problem. On the basis of random instances generated from the case of the Pearl River Delta, we conduct numerical experiments to examine the effectiveness and efficiency of the proposed solution method. Experimental results show that the Lagrangian relaxation-based method significantly outperforms a commercial solver and can derive tight upper bounds for the formulations with 300 time points and up to 100 vessels. For those instances with no more than 80 vessels, the proposed method can derive near-optimal solutions (with optimality gaps less than 3%) in only 20 iterations. Experiments based on realistic tracking data are also conducted to demonstrate the usefulness of our solutions.

In this work, we assume the power usage of drones is not affected by the flying speed, and our proposition suggests that drones should fly at their maximum speeds to generate optimal scheduled tours. However, in reality, flying too fast may lead to stronger air resistances, causing drone's endurance time to be shortened. For this practical consideration, a future study can be conducted to show how the battery consumption is affected by drone's flying speed. Based on that, the inspection tours can be optimized by regarding drone's speed as an additional decision. This extension will further improve the power management for drones, which is valuable in increasing both safety and efficiency of drone usages in various applications.

Acknowledgments

The authors are grateful to the two reviewers for helpful comments. This research is supported by the National Natural Science Foundation of China (71701128, 71831008). Shuaian Wang is sponsored by Environment and Conservation Fund Project 92/2017.

References

Agatz, N., Bouman, P. and Schmidt, M. (2018), Optimization approaches for the traveling salesman problem with drone, *Transportation Science* **52**(4), 739–1034.

Archetti, C., Speranza, M. G., Corberán, Á., Sanchis, J. M. and Plana, I. (2014), The team orienteering arc routing problem, *Transportation Science* **48**(3), 442–457.

Bianchessi, N., Mansini, R. and Speranza, M. G. (2018), A branch-and-cut algorithm for the team orienteering problem, *International Transactions in Operational Research* **25**(2), 627–635.

Boussier, S., Feillet, D. and Gendreau, M. (2007), An exact algorithm for team orienteering problems, *A Quarterly Journal of Operations Research* **5**(3), 211–230.

Butt, S. E. and Ryan, D. M. (1999), An optimal solution procedure for the multiple tour maximum collection problem using column generation, *Computers & Operations Research* **26**(4), 427–441.

Cacchiani, V., Caprara, A. and Fischetti, M. (2012), A Lagrangian heuristic for robustness, with an application to train timetabling, *Transportation Science* **46**(1), 124–133.

Caprara, A., Fischetti, M. and Toth, P. (2002), Modeling and solving the train timetabling problem, *Operations Research* **50**(5), 851–861.

Carlsson, J. G. and Song, S. (2017), Coordinated logistics with a truck and a drone, *Management Science* **64**(9), 3971–4470.

Corbett, J. J., Winebrake, J. J., Green, E. H., Kasibhatla, P., Eyring, V. and Lauer, A. (2007), Mortality from ship emissions: A global assessment, *Environmental Science & Technology* **41**(24), 8512–8518.

Crainic, T. G., Hewitt, M., Toulouse, M. and Vu, D. M. (2014), Service network design with resource constraints, *Transportation Science* **50**(4), 1380–1393.

Dang, D.-C., Guibadj, R. N. and Moukrim, A. (2013), An effective PSO-inspired algorithm for the team orienteering problem, *European Journal of Operational Research* **229**(2), 332–344.

Dorling, K., Heinrichs, J., Messier, G. G. and Magierowski, S. (2017), Vehicle routing problems for drone delivery, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **47**(1), 70–85.

Doudnikoff, M. and Lacoste, R. (2014), Effect of a speed reduction of containerships in response to higher energy costs in sulphur emission control areas, *Transportation Research Part D: Transport and Environment* **28**, 51–61.

Environmental Protection Department (2015), Controlling emissions from marine sector. Accessed January 20, 2018. http://www.epd.gov.hk/epd/english//environmentinhk/air/prob_solutions/controlling-emissions-marine-sector.html.

Fagerholt, K., Gausel, N. T., Rakke, J. G. and Psaraftis, H. N. (2015), Maritime routing and speed optimization with emission control areas, *Transportation Research Part C: Emerging Technologies* **52**, 57–73.

Fagerholt, K. and Psaraftis, H. N. (2015), On two speed optimization problems for ships that sail in and out of emission control areas, *Transportation Research Part D: Transport and Environment* **39**, 56–64.

Fisher, M. L. (2004), The Lagrangian relaxation method for solving integer programming problems, *Management Science* **50**(12), 1861–1871.

Golden, B. L., Wang, Q. and Liu, L. (1988), A multifaceted heuristic for the orienteering problem, *Naval Research Logistics* **35**(3), 359–366.

Green4Sea (2016), New agreement to use drones for ship emission monitoring. Accessed January 23, 2018. https://www.green4sea.com/new-agreement-to-use-drones-for-ship-emission-monitoring/.

Gunawan, A., Lau, H. C. and Vansteenwegen, P. (2016), Orienteering problem: A survey of recent variants, solution approaches and applications, *European Journal of Operational Research* **255**(2), 315–332.

He, Q., Zhang, X. and Nip, K. (2017), Speed optimization over a path with heterogeneous arc costs, *Transportation Research Part B: Methodological* **104**, 198–214.

IMO (2018), International Maritime Organization. Accessed January 20, 2018. http://www.imo.org/en/Pages/Default.aspx.

Ke, L., Zhai, L., Li, J. and Chan, F. T. (2016), Pareto mimic algorithm: An approach to the team orienteering problem, *Omega* **61**, 155–166.

Keshtkaran, M., Ziarati, K., Bettinelli, A. and Vigo, D. (2016), Enhanced exact solution methods for the team orienteering problem, *International Journal of Production Research* **54**(2), 591–601.

Kirschstein, T. and Meisel, F. (2015), GHG-emission models for assessing the eco-friendliness of road and rail freight transports, *Transportation Research Part B: Methodological* **73**, 13–33.

Lin, S.-W. and Vincent, F. Y. (2012), A simulated annealing heuristic for the team orienteering problem with time windows, *European Journal of Operational Research* **217**(1), 94–107.

Marine Electronics (2015), Drones lead the way in emissions compliance. Accessed January 23, 2018. http://www.marinemec.com/news/view,drones-lead-the-way-in-emissions-compliance_39183.htm.

Marine Traffic (2018). Accessed January 24, 2018. https://www.marinetraffic.com/en/ais/home/centerx:89.9/centery:9.2/zoom:4.

Martek (2018), The future of ship engine emission monitoring is up in the air. Accessed January 21, 2018. https://www.martek-marine.com/blog/future-ship-engine-emission-monitoring-air/.

Meng, Q., Du, Y. and Wang, Y. (2016), Shipping log data based container ship fuel efficiency modeling, *Transportation Research Part B: Methodological* **83**, 207–229.

MEPC (2016), Marine Environment Protection Committee, 70th session, 24-28 October 2016. Accessed March 3, 2018. http://www.imo.org/en/MediaCentre/MeetingSummaries/MEPC/Pages/MEPC-70th-session.aspx.

Mufalli, F., Batta, R. and Nagi, R. (2012), Simultaneous sensor selection and routing of unmanned aerial vehicles for complex mission plans, *Computers & Operations Research* **39**(11), 2787–2799.

Murray, C. C. and Chu, A. G. (2015), The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery, *Transportation Research Part C: Emerging Technologies* **54**, 86–109.

Murray, C. C. and Karwan, M. H. (2010), An extensible modeling framework for dynamic reassignment and rerouting in cooperative airborne operations, *Naval Research Logistics* **57**(7), 634–652.

Ng, M. W. (2015), Container vessel fleet deployment for liner shipping with stochastic dependencies in shipping demand, *Transportation Research Part B: Methodological* **74**, 79–87.

Ng, M. W. and Lo, H. K. (2016), Robust models for transportation service network design, *Transportation Research Part B: Methodological* **94**, 378–386.

North of England P&I Association (2018), China: Emission control areas update. Accessed January 23, 2018. http://www.nepia.com/insights/industry-news/china-emission-control-areas-starupdatestar/.

OECD (2018), Reducing sulphur emissions from ships. Accessed January 20, 2018. https://www.itf-oecd.org/sites/default/files/docs/sulphur-emissions-shipping.pdf.

Peters, G. (2016), How drones are changing the maritime industry. Accessed January 23, 2018. http://www.ship-technology.com/features/featurehow-drones-are-changing-the-maritime-industry-4865807/.

Port Technology (2017), World's largest maritime drone contract awarded. Accessed January 23, 2018. https://www.porttechnology.org/news/worlds_largest_maritime_drone_contract_awarded.

Qu, X., Meng, Q. and Li, S. (2011), Ship collision risk assessment for the Singapore Strait, *Accident Analysis & Prevention* **43**(6), 2030–2036.

Safety4Sea (2018), INTERTANKO launches bunker surcharge clauses for ECAs. Accessed January 20, 2018. https://www.green4sea.com/intertanko-launches-bunker-surcharge-clauses-ecas/.

Sethi, S. and Sorger, G. (1991), A theory of rolling horizon decision making, *Annals of Operations Research* **29**(1), 387–415.

Shetty, V. K., Sudit, M. and Nagi, R. (2008), Priority-based assignment and routing of a fleet of unmanned combat aerial vehicles, *Computers & Operations Research* **35**(6), 1813–1828.

Ship Efficiency Review (2016), Agreement signed to tackle emissions monitoring via drones. Accessed January 23, 2018. http://www.shipefficiencyreview.com/agreement-signed-to-tackle-emissions-monitoring-via-drones/.

Starcrest (2011), Port of Los Angeles inventory of air emissions 2010. Accessed January 20, 2018. https://www.portoflosangeles.org/pdf/2010_Air_Emissions_Inventory.pdf.

Transport and Environment (2018), Air pollution from ships. Accessed January 20, 2018. https://www.transportenvironment.org/what-we-do/shipping/air-pollution-ships.

UNCTAD (2017), Review of Maritime Transport. Accessed January 20, 2018. http://unctad.org/en/PublicationsLibrary/rmt2017_en.pdf.

Vansteenwegen, P., Souffriau, W. and van Oudheusden, D. (2011), The orienteering problem: A survey, *European Journal of Operational Research* **209**(1), 1–10.

Wang, X., Poikonen, S. and Golden, B. (2017), The vehicle routing problem with drones: Several worst-case results, *Optimization Letters* **11**(4), 679–697.

Ward, V. R. and Kobe, S. (2015), Best practices airborne MARPOL Annex VI monitoring. Accessed January 23, 2018. https://www.trafi.fi/filebank/a/1482762219/d80cdd7de80e58885ce5a4dd0af86c02/23541-Best_Practices_Airborne_MARPOL_Annex_VI_Monitoring.pdf.

Weng, J., Meng, Q. and Qu, X. (2012), Vessel collision frequency estimation in the Singapore Strait, *The Journal of Navigation* **65**(2), 207–221.

Xia, Y., Batta, R. and Nagi, R. (2017), Controlling a fleet of unmanned aerial vehicles to collect uncertain information in a threat environment, *Operations Research* **65**(3), 674–692.

Xiao, Y., Fu, X., Ng, A. K. and Zhang, A. (2015), Port investments on coastal and marine disasters prevention: Economic modeling and implications, *Transportation Research Part B: Methodological* **78**, 202–221.

Zheng, S., Ge, Y.-E., Fu, X., Nie, Y. M. and Xie, C. (2017), Modeling collusion-proof port emission regulation of cargo-handling activities under incomplete information, *Transportation Research Part B: Methodological* **104**, 543–567.