Branch-and-price for clash-free periodic supply vessel planning problem with split delivery and variable service time

Elham Mardaneh [†‡§], Mojtaba Heydar [*] [†‡§], Ryan Loxton [†‡§]

Abstract

Efficient scheduling and routing of vessels is crucial in the oil and gas industries. In this paper, we consider a periodic supply vessel planning problem in which the weekly demands at multiple offshore facilities are satisfied with a fleet of heterogeneous vessels. Preemptive service at the base, variable service at facilities, and split delivery are allowed. The objective is to find the optimal fleet composition and weekly trips for vessels that minimize the total travel and fixed costs. The scheduling challenge is modelled by an arc-flow, path-flow, and set-partitioning formulations. For the first time in the literature of periodic supply vessel planning problem, we propose a branch-and-price algorithm. Simulation results for the North West Shelf project in Australia show that the proposed approach can generate high-quality solutions to large, industrial-scale problem instances.

Keywords: Transportation, Branch-and-price, OR in maritime industry, Supply vessel planning, Periodic routing

1 Introduction

Vessels are the main mode of transport for cargo delivery in offshore oil and gas industries around the world. They are capital-intensive equipment with about 25000 USD daily charter costs Kisialiou et al. (2019). In this paper, we consider a new variant of periodic supply vessel planning (PSVP) problem that identifies the optimal fleet composition of vessels and determines the weekly schedule which repeats itself for utilised vessels. The classic PSVP problem includes multiple offshore oil and gas facilities with weekly cargo demands that must be served by a fleet of heterogeneous vessels for cargo delivery. All vessels depart from and return to a single onshore depot, called the base hereafter. The base and facilities receive service during given time windows. The start time and the end time of a service at the base or facilities should be within a time window. This problem belongs to industrial shipping which is different from liner and tramp shipping Homsi et al. (2020). According to Kisialiou et al. (2018, 2019) the PSVP problem is $\mathcal{NP}$-hard.

The aim of this paper is to enhance the PSVP problem by considering practical features such as split delivery, variable service time at facilities and preemptive service at the base. In our PSVP problem, the turnaround time for the vessels at the base does not coincide with a given time window. This means that potentially the service will need to be paused during closing hours and pushed into the next time window. Moreover, due to safety regulations and limited number of personnel, the weekly schedule of vessels needs to be clash-free. This imposes at most one vessel at each facility at any time. This problem is inspired by Woodside, the largest oil and gas producer in Australia and reflects its operations in the North West Shelf region of Western Australia, where the supply vessels are based in Karratha (see Figure 1).

The periodic supply vessel planning problem at hand is unique among the class of industrial shipping problems. The operational features which make this problem different from the classical PSVP include:

- Split delivery is permitted.

[*]email: mojtaba.heydar@curtin.edu.au

[†]School of Electrical Engineering, Computing, and Mathematical Sciences, Curtin University, Bentley WA 6102, Australia.

[‡]ARC Training Centre for Maintenance through Data Science

[§]Curtin Centre for Optimisation and Decision Science

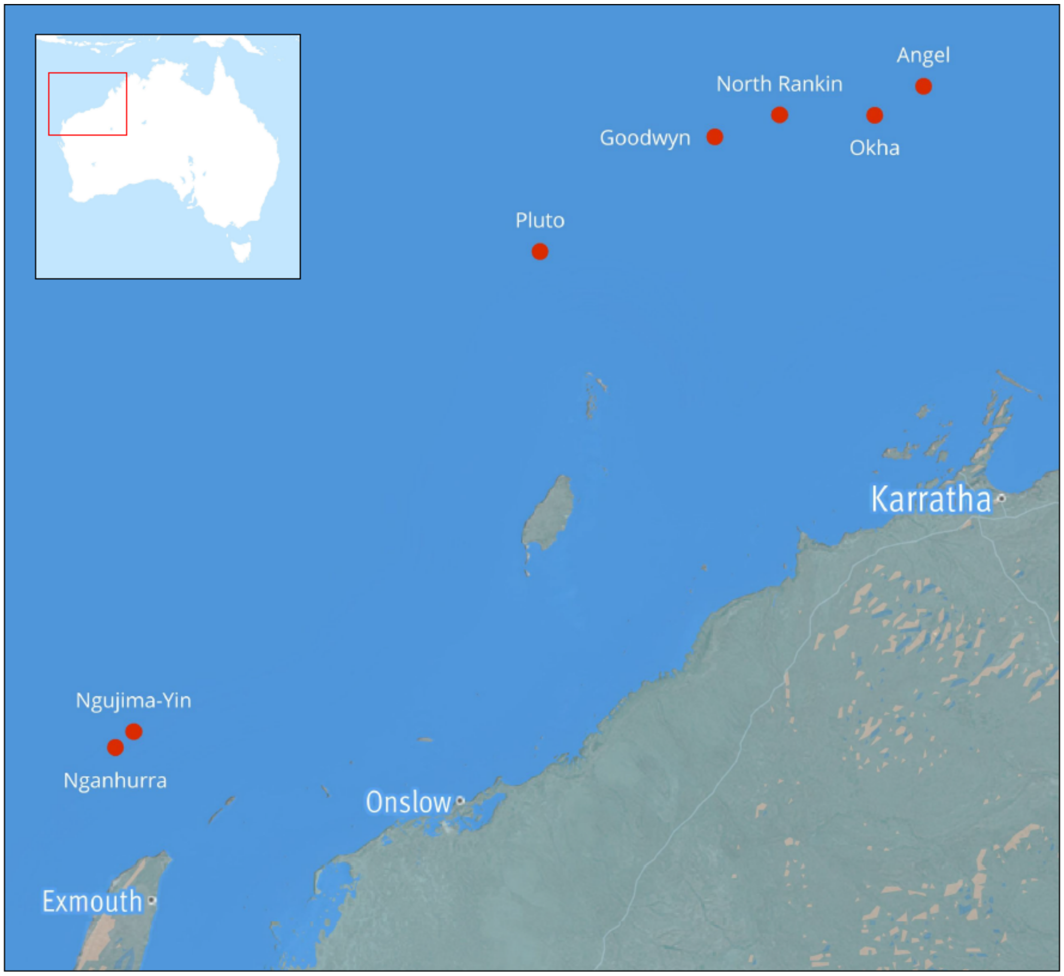

Figure 1: Woodside's offshore facilities in the North West Shelf region Mardaneh et al. (2017)

- Each facility has fixed and variable service times which are different for each vessel type. The fixed service time is related to positioning of the vessel and the variable service time is defined by arrival time and load quantity.
- Offshore facilities can accommodate at most one vessel at any given time (clash-free).
- There is a preemptive preparation or service time at the base that varies depending on the vessel types. The service time at the base which mainly spans the maintenance and vessel preparation is a function of its start time and vessel type. Unlike (Halvorsen-Weare et al., 2012) where there is only a fixed departure time, in this problem the departure time from the base can be anytime within the base time window. This is equivalent to the flexible departure time which is a recent practice in major oil and gas companies (Kisialiou et al., 2018).
- The actual duration of a trip is a function of its departure time from the base and the cargo delivery at facilities. A trip is a sequence of offshore facilities visited by a vessel in a pre-specified time, which starts and ends at the base.

It should be noted that in traditional PSVP problem the duration of trips is two to three days (Halvorsen-Weare et al., 2012), whereas in our problem the duration of a trip can be up to seven days which further increases the complexity of the problem.

Figure 2 illustrates an example of periodic supply vessel planning with seven-day long period. There are one onshore base and five offshore facilities serviced by three trips. Each facility has only one time window on each day shown by hollow circles. Vessel 1 on trip 1 departs the base on day one, and services facility f_1 on day two, facility f_3 on day three, and facility f_4 on day four. Finally, it returns

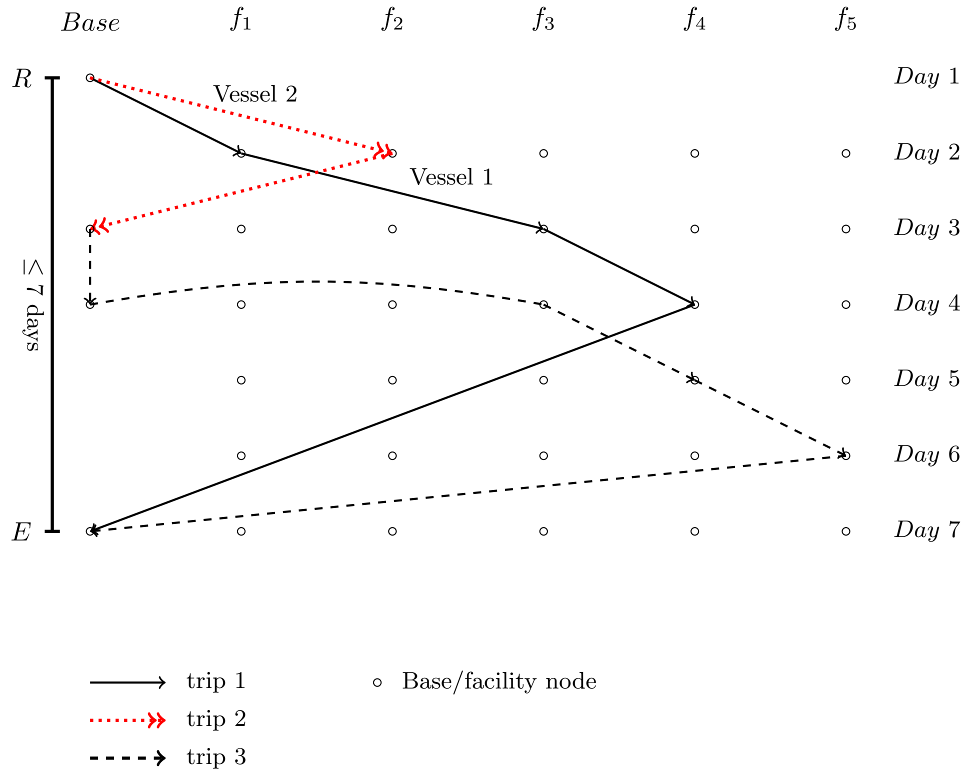

Figure 2: An example of three trips of two vessels with split deliveries and multiple trips.

to the base on day seven. Vessel 2 on trip 2 departs the base on day one, services facility f_2 on day two, and returns to the base on the following day. It undergoes a base service on day three and four, departs the base on day four, and services three facilities f_3, f_4, and f_5 on three consecutive days. It returns to the base at the end of period on day seven. trip 1 spans over the length of the period, which is seven days in this example. trip 2 starts and finishes in less than a week and the length of trip 3 is less than one period, but it starts in the middle of the week and finishes at the end of the period. Recall that split delivery of cargo at the facilities is allowed. For example, facility f_3 has two deliveries on days three and four, respectively, which are fulfilled through trips 1 and 3. Similarly, demand of facility f_4 is fulfilled by two deliveries on days four and five.

The objective of interest is to minimize the total cost of vessel trips and the fixed (charter) cost of the fleet. The optimal fleet cost results in determining the optimal fleet composition. The constraints that must be taken into account include time windows at the base and facilities, trip duration, visit number and deliveries of the facilities, vessel capacities, and clash-free visits of facilities. In this paper, we first present the arc-flow and path-flow formulations of the problem and then develop a branch-and-price algorithm to solve the set-partitioning formulation.

1.1 Literature review

This problem belongs to the family of periodic vehicle routing problems in which customers identify sets of acceptable visit days over a time horizon (e.g., a week) and are served by vehicles that make single-day round trips (Kisialiou et al., 2018). This family of problems is similar to PSVP in some aspects. The depot from which capacitated vehicles depart is similar to the base in PSVP. The demand of each customer is known and will be fully satisfied by vehicles (Campbell and Wilson, 2014).

However, the PSVP is more complicated than the routing problems. For example, the period length in supply vessel planning problem is one week, while in the routing problems the period length is typically one day (Halvorsen-Weare et al., 2012; Kisialiou et al., 2018). Other factors that make our proposed supply vessel planning problem significantly challenging are split delivery, clash-free visits, long trip duration, the base time window, variable service duration, fleet composition and preemptive service at base. All these features have not been studied collectively in periodic supply vessel planning or even routing problems.

This study also relates to several variants of vehicle routing problem (VRP). For a reach survey on VRP, see Braekers et al. (2016). Features that are common with VRP include: time windows in which services must start and finish (Andersson et al., 2011; Koç et al., 2016); split deliveries where facilities can be served by multiple vessels (Desaulniers, 2010; Andersson et al., 2011; McNabb et al., 2015); limited trip duration since supply vessel trips are usually limited to one week (Hernandez et al., 2014); vehicle-site suitability since some off-shore facilities can only accommodate certain types of vessels (Zare-Reisabadi and Mirmohammadi, 2015); and multiple-trips since vessels may perform multiple round-trips per week (Hernandez et al., 2016; Afshar-Nadjafi and Afshar-Nadjafi, 2017; Huang et al., 2021). A recent study by (Li et al., 2020) considers the synchronised VRP with split delivery, variable service time and multiple time windows. They develop a branch-and-price-and-cut algorithm to tackle the problem. Although, our problem has some common features with the problem in (Li et al., 2020), there are significant differences that prohibits applying their proposed solution methodology to the problem in this paper. These differences are as follows: we consider fixed cost for using a vehicle, prescribed number of visits for facilities, heterogeneous vehicles, and multiple trips. Moreover, we allow vehicles to visit facilities more than once and deliver a continuous amount of cargo.

The number of previous studies on the periodic supply vessel planning is somewhat limited. Fagerholt and Lindstad (2000) considered the basic version of the periodic supply vessel planning problem. Gribkovskaia et al. (2008) considered a single vessel planning problem and developed a tabu search as well as several construction heuristics as a solution methodology. In another study, Halvorsen-Weare et al. (2012) extended the work of Fagerholt and Lindstad (2000) to consider other operational and service features such as capacity constraint at the base, minimum and maximum trip duration, and spread constraints. They presented a trip-based (referred to as voyaged-based) solution method working in two phases. In the first phase, all feasible trips are generated. Then, the second phase uses the generated trips in a set covering model with the spread of departures. Since, the periodic vessel planning problem is $\mathcal{NP}$-hard, several heuristic methods have been proposed to solve large-scale instances. Shyshou et al. (2012) developed a large neighbourhood search heuristic for determining the fleet composition as well as vessel schedules. Borthen et al. (2018) adapted the hybrid genetic algorithm of Vidal et al. (2012) to account for some features of periodic vessel planning problem such as spanning trips over multiple periods. In a recent study, similar to current study, Kisialiou et al. (2018) considered flexible departure times from the base. They developed a trip-based formulation where trips were defined for vessel types instead of individual vessels. As the trip-based formulation was not capable of solving large-size instances, they developed an adaptive variable neighbourhood heuristic. Borthen et al. (2019) considered a bi-objective periodic supply vessel planning problem where the objectives were to minimize the cost of weekly plans for platform supply vessels while maximizing the compatibility of the current plan with previous plans. Then, they adapted the genetic search algorithm of Borthen et al. (2018) to solve real-size instances.

This study overlaps with the other types of maritime transportation problem such as ship routing, fleet composition, and berth allocation. Maritime pickup and delivery problem was studied by Andersson et al. (2011) and Stålhane et al. (2012), and fuel oil distribution problem was addressed in Agra et al. (2013). Christiansen et al. (2017) developed arc-flow and path-flow formulations for routing and scheduling fuel vessels to supply anchored ships. An integrated berth allocation, fleet composition, and periodic supply vessel planning problem was studied by Cruz et al. (2019).

This paper is the literature's first study that presents a branch-and-price algorithm for clash-free periodic supply vessel planning problem with split delivery, variable service time at facilities and preemptive service at base. However, in our rigorous literature survey, we could find only two studies which developed branch-and-price for ship routing and scheduling problem (Homsi et al., 2020; Stålhane et al., 2012). These studies modelled the ship routing problem as a pickup-delivery problem in which ship may have different starting locations.

1.2 Contributions

The purpose of this paper is to minimize the total travel cost and fixed cost of the fleet for a periodic supply vessel planning problem with the following characteristics:

1. Variable service time at facilities with multiple time windows.

2. Preemptive service time at base which is a function of service start time.
3. Split delivery with limited trip duration.
4. One vessel at a facility at any time.

We present the arc-flow and path-flow formulations of the problem and develop a branch-and-price algorithm to solve the set-partitioning model. We propose a special label-extension algorithm to solve the pricing problem and to find the non-dominant solutions.

The remainder of this paper is organized as follows. Section 2 presents the description of the problem and develops the arc-flow and path-flow formulations of it in Sections 2.1 and 2.2, respectively. The Dantzig-Wolfe decomposition is developed in Section 3. Column generation and label extension are discussed in Section 4. The branch-and-price approach is presented in Section 5 where detailed explanations are provided. Then, Section 6 describes our computational study on large-scale instances generated from a real-world practice of the problem. Concluding remarks are given in Section 7.

2 Problem Description

Our scheduling problem involves a set of offshore oil and gas facilities and a set of vessels for cargo transport. All vessels share the same base and must undergo a thorough base service (which includes maintenance, safety check, refueling and cargo loading) before leaving the base. Since the base is only open for part of each day, the duration of the base service depends on when the service starts—if it starts late in the day, then potentially the service will need to be paused during closing hours and pushed into the next day. Vessel trips take no longer than T hours (e.g. 168 hours or 7 days).

Let 0 denote the vessel base and let $\mathcal{F}$ denote the set of offshore facilities. Our goal is to design an optimal schedule of length T, and since this schedule can start at any time, we model the problem over a $\mathbb{T}$ time horizon. Note that $\mathbb{T} \bmod T = 0$. Let $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ denote a network model for the problem, where $\mathcal{N}$ is the node set and $\mathcal{A}$ is the arc set. The node set $\mathcal{N}$ can be expressed as

$$\mathcal{N} = \{0\} \cup \{d\} \cup \mathcal{K}_B \cup \{\cup_f \mathcal{K}_f\},$$

where 0 is the designated start node for vessels, d is the designated end node for vessels, $\mathcal{K}_B$ is a set of nodes representing the vessel base and $\mathcal{K}_f$ is a set of nodes representing offshore facility f (each node in $\mathcal{K}_f$ corresponds to a different time window for facility $\mathcal{F}$). There are multiple nodes in $\mathcal{K}_B$ because each vessel must use a different node for each base service. Note that d corresponds to the base 0; that is, vessels start and end at the base.

In the mathematical model, a typical vessel schedule (consisting of multiple trips) would start at node 0 for the initial base service, then depart 0 and visit multiple offshore nodes sequentially for cargo delivery before returning to a base node in $\mathcal{K}_B$ for the next base service, after which a new trip can begin (if required). The vessel will eventually finish at node d after the end of the final trip.

Let $\mathcal{V}$ denote the set of vessels in the problem. The sets defined above are summarized in Table 1.

Table 1: Set definitions in mathematical models

$\mathcal{F}$	Set of offshore facilities
$\mathcal{N}$	Set of nodes in the network model
$\mathcal{A}$	Set of arcs in the network model
$\mathcal{K}_B$	Set of nodes representing the vessel base
$\mathcal{K}_f$	Set of nodes representing offshore facility f
$\mathcal{V}$	Set of vessels

Let δ_0^v denote the service duration of vessel v at the vessel base, and let $\left[a_l^0, b_l^0\right]$ denote the l^{th} opening period of the vessel base. We assume that the base's open and close times are the same on each day, giving $a_{l+1}^0 = a_l^0 + 24$ and $b_{l+1}^0 = b_l^0 + 24$ for each l. Furthermore, $b_l^0 - a_l^0 = b_{l+1}^0 - a_{l+1}^0$ for each $l \in \left\{1, 2, \ldots, \lfloor \frac{2T}{24} \rfloor - 1\right\}$.

Each offshore node $i \in \mathcal{K}_f,\ f \in \mathcal{F}$, has a service time window $[a_i, b_i]$, defining the interval during which the different vessels can service the node, and a time $\delta_{\text{fix},i}^v,\ v \in \mathcal{V}$, defining the fixed time for

Table 2: Parameter definitions in mathematical models

δ_0^v	Service duration of vessel v at vessel base
$[a_l^0, b_l^0]$	The l^{th} opening period of the vessel base
$\delta_{\text{fix},i}^v$	Fixed time for service visit by vessel v at node i (representing an offshore facility)
$[a_i, b_i]$	Service time window for vessel v at node i
Q^v	Deck-space of vessel v
t_{ij}^v	Travel time for vessel v traversing (i, j)
c_{ij}^v	Fixed cost coefficient for vessel v traversing arc (i, j)
c_{fix}^v	Fixed cost of using vessel v
q_f	Weekly cargo demand at facility f
n_f	Weekly visit frequency at facility f
ξ_f^v	Minimum off-load by vessel v at facility f
s_f^v	Off-loading rate for vessel v at facility f

service visits at that node (these fixed times incorporate vessel positioning and set-up but not cargo off-loading time). Note that all time windows and service duration are given in units of hours.

The characteristics of each vessel $v \in \mathcal{V}$ are expressed by the following parameters: deck-space of the vessel, travel time required to traverse an arc, fixed cost coefficient corresponding to an arc, and fixed cost of using the vessel. Moreover, the constraints at each offshore facility $f \in \mathcal{F}$ are expressed by the following parameters: weekly cargo demand, weekly visit frequency, minimum off-load for vessel $v \in \mathcal{V}$, and off-loading rate for vessel v at facility f. These parameters are summarized in Table 2.

In the mathematical model, each vessel can perform multiple trips per week and each trip is preceded by a base service. This service is paused while the base is closed. Servicing at offshore facilities, however, cannot be paused and must begin immediately after arrival.

We define two sets $\mathcal{T}_1^{lv}$, $\mathcal{T}_2^{lv}$ and parameter λ^v as follows:

$$\lambda^v = \lceil \frac{\delta_0^v}{b_1^0 - a_1^0} \rceil \tag{1}$$

$$\mathcal{T}_1^{lv} = \begin{cases} \left[a_l^0, \min\left\{b_l^0, b_{l+\lambda^v-1}^0 - a_{l+\lambda^v-1}^0 - \delta_0^v + b_{l+\lambda^v-2}^0\right\}\right) & \text{if } \lambda^v \geq 2 \\ \left[a_l^0, b_l^0 - \delta_0^v\right) & \text{if } \lambda^v = 1 \end{cases} \tag{2}$$

$$\mathcal{T}_2^{lv} = \begin{cases} \left[\min\left\{b_l^0, b_{l+\lambda^v-1}^0 - a_{l+\lambda^v-1}^0 - \delta_0^v + b_{l+\lambda^v-2}^0\right\}, b_l^0\right] & \text{if } \lambda^v \geq 2 \\ \left[b_l^0 - \delta_0^v, b_l^0\right] & \text{if } \lambda^v = 1 \end{cases} \tag{3}$$

where, $\mathcal{T}_1^{lv}$ indicates the time interval within the l^{th} time window that if a vessel starts the base service, it spans λ^v days, and $\mathcal{T}_2^{lv}$ indicates the time interval within the l^{th} time window that if a vessel starts a base service, it spans $\lambda^v + 1$ days. Note that $\left(\bigcup_l \mathcal{T}_1^{lv}\right) \bigcap \left(\bigcup_l \mathcal{T}_2^{lv}\right) = \emptyset$ holds. Suppose that the planning horizon is 14 days, $\mathbb{T} = 14$, the base is open every day from 6:00AM to 6:00PM and $\delta_0^v = 21$. Hence,

$$a_1^0 = 6, b_1^0 = 18, a_2^0 = 30, b_2^0 = 42, a_3^0 = 54, b_3^0 = 66, \ldots, a_{14}^0 = 318, b_{14}^0 = 330.$$

According to formulation (1), $\lambda^v = \lceil \frac{21}{18-6} \rceil = 2$. Then, $\mathcal{T}_1^{1v}$ is the time interval within the first time window [6:00AM,6:00PM] that if vessel v starts the base service, it spans day one and day two. $\mathcal{T}_2^{1v}$ is the time interval within the first time window that if vessel v starts the base service, it spans day one, day two and day three. According to formulations (2) and (3),

$$\mathcal{T}_1^{lv} = \left[a_1^0, \min\left\{b_1^0, b_2^0 - a_2^0 - \delta_0^v + b_1^0\right\}\right) = \left[a_1^0, \min\left\{18, 42 - 30 - 21 + 18\right\}\right) = [6, 9)\,,$$

$$\mathcal{T}_2^{1v} = \left[\min\left\{b_1^0, b_2^0 - a_2^0 - \delta_0^v + b_1^0\right\}, b_1^0\right] = [9, 18]\,.$$

Now, if vessel v starts the base service at 7:30AM, it takes $\delta_0^v + b_1^0 - a_1^0 = 21 + 18 - 6 = 33$ hours to finish the service at 4:30PM of next day. If vessel v starts the base service at 10:00AM, it takes $\delta_0^v + b_1^0 - a_1^0 + b_2^0 - a_2^0 = 21 + 18 - 6 + 42 - 30 = 45$ hours to finish the service at 7:00AM of two days later.

2.1 Arc-Flow Model

In the arc-flow formulation the variables represent flows on individual arcs of network $\mathcal{G}$. The decision variables in the model are defined below:

- S = schedule start time;
- E = schedule end time;
- y_{ij}^v = binary variable indicating whether vessel v traverses link (i,j) (where $y_{ij}^v = 1$ if this occurs and $y_{ij}^v = 0$ otherwise);
- q_{ij}^v = cargo flow transported by vessel v along link (i,j);
- $u_{\text{start},i}^v$ = service start time of vessel v at node i;
- $u_{\text{end},i}^v$ = service end time of vessel v at node i;
- $z_{l,i}^v$ = binary variable indicating whether $u_{\text{start},i}^v \in \mathcal{T}_1^{lv}$ at node $i \in \{0\} \cup \mathcal{N}_B$ (where $z_{l,i}^v = 1$ if this occurs and $z_{l,i}^v = 0$ if $u_{\text{start},i}^v \in \mathcal{T}_2^{lv}$);
- $w_{l,i}^v$ = binary variable indicating whether $u_{\text{start},i}^v \in \mathcal{T}_2^{lv}$ at node $i \in \{0\} \cup \mathcal{N}_B$ (where $w_{l,i}^v = 1$ if this occurs and $w_{l,i}^v = 0$ if $u_{\text{start},i}^v \in \mathcal{T}_1^{lv}$);
- ${z'}_{l,i}^v$ = binary variable indicating whether $u_{\text{end},i}^v \in \left[a_l^0, b_l^0\right]$ at node $i \in \{0\} \cup \mathcal{N}_B$ (where ${z'}_{l,i}^v = 1$ if this occurs and ${z'}_{l,i}^v = 0$ otherwise).
- ϕ_i^{vw} = binary variable indicating whether vessels v and $w \neq v$ visit node i corresponding to facility $f \in \mathcal{K}_f$ (where $\phi_i^{vw} = 1$ if this occurs and $\phi_i^{vw} = 0$ otherwise); and
- ψ_i^{vw} = binary variable indicating whether vessels v and $w \neq v$ visit node i corresponding to facility $f \in \mathcal{K}_f$, and vessel v completes its service at node i before vessel w arrives at the node (where $\psi_i^{vw} = 1$ if this occurs and $\psi_i^{vw} = 0$ otherwise).

We define $u_{\text{start},i}^v = u_{\text{end},i}^v = 0$ if vessel v never visits node i. In addition, for any vessel $v \in \mathcal{V}$, we define $u_{\text{start},0}^v = 0$ and $u_{\text{start},d}^v = 0$ if vessel v is never used.

The objective function defined below is the combination of the travel cost and the vessels fixed cost:

$$\text{Total Cost} = \sum_{v \in \mathcal{V}} \sum_{(i,j) \in \mathcal{A}} c_{ij}^v y_{ij}^v + c_{\text{fix}}^v \sum_{j \in \mathcal{N}} y_{0j}^v. \tag{4}$$

The overall objective function (4) should be minimized subject to the constraints described below. Note that in the constraint descriptions, $M > 0$ is a sufficiently large constant.

$$0 \le q_{ij}^v \le Q^v y_{ij}^v, \qquad v \in \mathcal{V}, (i,j) \in \mathcal{A}, \tag{5}$$

$$\sum_{j \in \mathcal{N}} a_i y_{ji}^v \le u_{\text{start},i}^v \le u_{\text{end},i}^v \le \sum_{j \in \mathcal{N}} b_i y_{ji}^v, \qquad v \in \mathcal{V}, f \in \mathcal{F}, i \in \mathcal{K}_f, \tag{6}$$

$$M\left(z_{l,i}^v + w_{l,i}^v - 1\right) + a_l^0 \le u_{\text{start},i}^v \le b_l^0 + M\left(1 - z_{l,i}^v - w_{l,i}^v\right), \qquad l \in \left\{1, 2, \ldots \left\lfloor \frac{\mathbb{T}}{24} \right\rfloor \right\}, i \in \{0\} \cup \mathcal{K}_B \tag{7}$$

$$M\left({z'}_{l,i}^v - 1\right) + a_l^0 \le u_{\text{end},i}^v \le b_l^0 + M\left(1 - {z'}_{l,i}^v\right), \qquad l \in \left\{1, 2, \ldots \left\lfloor \frac{\mathbb{T}}{24} \right\rfloor \right\}, i \in \{0\} \cup \mathcal{K}_B, \tag{8}$$

$$0 \le S \le u_{\text{start},0}^v + M\left\{1 - \sum_{j \in \mathcal{N}} y_{0j}^v\right\}, \qquad v \in \mathcal{V}, \tag{9}$$

$$E \ge u_{\text{start},d}^v, \qquad v \in \mathcal{V}, \tag{10}$$

$$0 \le E - S \le T, \tag{11}$$

$$\sum_{j \in \mathcal{N}} y_{ji}^v \le 1, \qquad v \in \mathcal{V}, f \in \mathcal{F}, i \in \{d\} \cup \mathcal{K}_B \cup \mathcal{K}_f, \tag{12}$$

$$\sum_{j \in \mathcal{N}} y_{j0}^v = \sum_{j \in \mathcal{N}} y_{dj}^v = 0, \qquad v \in \mathcal{V}, \tag{13}$$

$$\sum_{l} \left(z_{l,i}^v + w_{l,i}^v\right) = \sum_{f} \sum_{j \in \mathcal{K}_f} y_{ij}^v, \qquad v \in \mathcal{V}, i \in \{0\} \cup \mathcal{K}_B, \tag{14}$$

$$\sum_{l} {z'}^{v}_{l,i} = \sum_{f} \sum_{j \in \mathcal{K}_f} y^v_{ij}, \qquad v \in \mathcal{V}, i \in \{0\} \cup \mathcal{K}_B, \tag{15}$$

$$\sum_{j \in \mathcal{N}} y^v_{ji} = \sum_{j \in \mathcal{N}} y^v_{ij}, \qquad v \in \mathcal{V}, f \in \mathcal{F}, i \in \mathcal{K}_B \cup \mathcal{K}_f, \tag{16}$$

$$\sum_{j \in \mathcal{N}} q^v_{ji} - \sum_{j \in \mathcal{N}} q^v_{ij} \geq \sum_{j \in \mathcal{N}} \xi^v_f y^v_{ji}, \qquad v \in \mathcal{V}, f \in \mathcal{F}, i \in \mathcal{K}_f, \tag{17}$$

$$\sum_{v \in \mathcal{V}} \sum_{j \in \mathcal{N}} \sum_{i \in \mathcal{K}_f} q^v_{ji} - \sum_{v \in \mathcal{V}} \sum_{j \in \mathcal{N}} \sum_{i \in \mathcal{K}_f} q^v_{ij} = q_f, \qquad f \in \mathcal{F}, \tag{18}$$

$$\sum_{v \in \mathcal{V}} \sum_{j \in \mathcal{N}} \sum_{i \in \mathcal{K}_f} y^v_{ji} \geq n_f, \qquad f \in \mathcal{F}, \tag{19}$$

$$u^v_{\text{end},i} + t^v_{ij} \leq u^v_{\text{start},j} + M\{1 - y^v_{ij}\}, \qquad v \in \mathcal{V}, i \in \mathcal{N} \setminus \{d\}, j \in \mathcal{N}, \tag{20}$$

$$u^v_{\text{end},i} - u^v_{\text{start},i} \geq \delta^v_{\text{fix},i} \sum_{j \in \mathcal{N}} y^v_{ji} + \frac{\sum_{j \in \mathcal{N}} q^v_{ji} - \sum_{j \in \mathcal{N}} q^v_{ij}}{s^v_f}, \qquad v \in \mathcal{V}, f \in \mathcal{F}, i \in \mathcal{K}_f, \tag{21}$$

$$u^v_{\text{end},i} - u^v_{\text{start},i} \geq \delta^v_0 \sum_{j \in \mathcal{N}} y^v_{ij} + \left(a^0_{l+1} - b^0_l\right) \left((\lambda^v - 1) \sum_{l} z^v_{l,i} + \lambda^v \sum_{l} w^v_{l,i} \right), \qquad v \in \mathcal{V}, i \in \{0\} \cup \mathcal{K}_B, \tag{22}$$

$$\phi^{vw}_i \geq \psi^{vw}_i, \qquad v \in \mathcal{V}, w \in \mathcal{V}, i \in \mathcal{K}_f, f \in \mathcal{F}, \tag{23}$$

$$\phi^{vw}_i \geq 1 - M\left\{2 - \sum_{j' \in \mathcal{N}} y^v_{j'i} - \sum_{j' \in \mathcal{N}} y^w_{j'i}\right\}, \qquad v \in \mathcal{V}, w \in \mathcal{V}, f \in \mathcal{F}, i \in \mathcal{K}_f, \tag{24}$$

$$2\phi^{vw}_i \leq \sum_{j' \in \mathcal{N}} y^v_{j'i} + \sum_{j' \in \mathcal{N}} y^w_{j'i}, \qquad v \in \mathcal{V}, w \in \mathcal{V}, f \in \mathcal{F}, i \in \mathcal{K}_f, \tag{25}$$

$$u^v_{end,i} \leq u^w_{\text{start},i} + M\{2 - \phi^{vw}_i - \psi^{vw}_i\}, \qquad v \in \mathcal{V}, w \in \mathcal{V}, f \in \mathcal{F}, i \in \mathcal{K}_f, \tag{26}$$

$$u^v_{\text{start},i} \geq u^w_{\text{end},i} - M\{1 - \phi^{vw}_i\} - M\psi^{vw}_i, \qquad v \in \mathcal{V}, w \in \mathcal{V}, f \in \mathcal{F}, i \in \mathcal{K}_f. \tag{27}$$

The lower and upper bounds for the continuous-valued decision variables are defined by constraints (5)-(11). In particular, constraints (5) ensure that cargo flow is non-negative, cannot exceed the vessel's capacity and is zero if the corresponding link is not traversed. Constraints (6) state that vessel must start and end the service at facilities when they are open. When a vessel starts a trip from the base node, its service starts and ends when the vessel base is open (Constraints (7) and (8)). The schedule start time, end time and duration are expressed by constraints (9)-(11). Constraints (12)-(16) are defined to impose that each vessel trip starts and finishes at base nodes and does not include "jumps" or "cycles". Constraints (12) state that each vessel can visit the vessel base and the offshore facility multiple times, but every new visit must use a different node. Constraints (13) ensure that a vessel cannot return to the start node and cannot leave the end node. When a vessel starts a trip, it must undergo a base service (constraints (14) and (15)). Flow balance equations (16) guarantee that a vessel cannot start a schedule or finish at facilities or at base nodes other than start and end nodes. Constraints (17), (18) and (19) govern the minimum offload on each trip, the weekly cargo demand and the visit frequency at facilities, respectively. Time sequencing constraints are defined by constraints (20). Constraints (21) ensure that if a vessel visits a facility, then the service duration spans the fixed service time and variable unloading time. The base service time that may be interrupted due to base closure is defined by constraints (22). The final group of constraints (23)-(27) state that different vessels do not visit the same offshore facility at the same time. This is necessary because of safety regulations and personnel availability. Specifically, ϕ^{vw}_i and ψ^{vw}_i are defined by constraints (23)-(25) and clash-free visits at facilities are expressed by constraints (26) and (27).

Our optimization model can now be defined as follows: Minimize objective (4) subject to constraints (5)-(27). This problem is a mixed-integer linear programming problem that can, in principle, be solved using standard software such as CPLEX. The main barrier to solving the problem for real industry scenarios is dimensionality. In the next section, we present the path-flow formulation of the problem. In the path-flow formulation the decision variables correspond to flow on paths of network $\mathcal{G}$, and are frequently associated with set-covering, set-packing, or set-partitioning models.

Table 3: Sets and input parameters of path-flow model

Sets	
$\mathcal{R}$	ordered set of all feasible trips, ordered by departure time
$\mathcal{N}_r$	ordered set of nodes on trip $r \in \mathcal{R}$, ordered by arrival time at the node (representing an offshore facility)
$\mathcal{N}$	set of all facility nodes, $\bigcup_{f \in \mathcal{F}} \mathcal{K}_f$
$\mathcal{R}_v \subseteq \mathcal{R}$	set of all trips for vessel $v \in \mathcal{V}$ from the base
Parameters	
$\hat{e}_r$	minimum journey end time of vessel $v_r \in \mathcal{V}$ on trip r
τ_r	start time of trip r from the base
$\hat{u}^{v_r}_{\text{arrival},i}$	earliest arrival time at node i on trip r
$\hat{u}^{v_r}_{\text{start},i}$	earliest service start time at node i on trip r
$\hat{u}^{v_r}_{\text{end},i}$	earliest service finish time at node i on trip r
c_r	cost of trip $r \in \mathcal{R}$
o^r_i	binary parameter indicating whether node i is covered by trip r

2.2 Path-Flow Model

To develop the path-flow model, we need to define some other concepts and secondary parameters. A trip of a vessel starts at the base and after visiting some nodes, and off-loading the cargo, it returns to the base at the end of the trip. We define a trip as an ordered set of nodes $r = \{0, i_1, i_2, \dots, i_j, \dots, d\}$ plus a vessel (v_r) and start time (τ_r), where subscript j indicates the order of a node in trip r. There is a cost associated with each trip r traveled by vessel v_r and is defined as $c_r = \sum_{i,j \in r} c^{v_r}_{ij}$. Due to vessel capacity and distance constraints, some trips cannot be covered by some vessels. Therefore, we define the set of feasible trips for vessel v as $\mathcal{R}_v$. There is a minimum journey end time $\hat{e}_r$ for each trip r by vessel v_r. The value of $\hat{e}_r = \hat{u}^{v_r}_{\text{start},d}$ depends on τ_r. To calculate the parameter $\hat{e}_r$, we introduce some additional parameters. Let $\hat{u}^{v_r}_{\text{arrival},i_j}$ and $\hat{u}^{v_r}_{\text{start},i_j}$ be the earliest arrival and start time at node i_j for vessel v_r taking trip r, respectively. Recall that $\lambda^v = \lceil \frac{\delta^v_0}{b^0_1 - a^0_1} \rceil$ is the required time to adjust the service end time of vessel v at the base (service at the base is preemptive). Then we can calculate $\hat{e}_r$ using the set of equations as given below:

$$
\begin{cases}
\hat{u}^{v_r}_{\text{arrival},i_j} = \begin{cases} \tau_r + \delta^{v_r}_0 + t^{v_r}_{0,i_j} + \lambda^{v_r}(a^0_2 - b^0_1) & ; \quad j = 1 \\ \hat{u}^{v_r}_{\text{end},i_{j-1}} + t^{v_r}_{i_{j-1},i_j} & ; \quad j = 2, \dots, |\mathcal{N}_r| \end{cases} \\
\hat{u}^{v_r}_{\text{start},i_j} = \max(a_{i_j}, \hat{u}^{v_r}_{\text{arrival},i_j}) \\
\hat{u}^{v_r}_{\text{end},i_j} = \hat{u}^{v_r}_{\text{start},i_j} + \delta^{v_r}_{\text{fix},i_j} + \frac{\xi^{v_r}_{f:i_j \in \mathcal{K}_f}}{s^{v_r}_{f:i_j \in \mathcal{K}_f}} \\
\hat{e}_r = \hat{u}^{v_r}_{\text{end},i_{|\mathcal{N}_r|}} + t^{v_r}_{i_{|\mathcal{N}_r|},d}
\end{cases}
$$

To govern the visit frequency of facilities, we introduce a binary parameter o^r_i which indicates whether node i corresponding to facility f is covered by trip r ($o^r_i = 1$ if this occurs and $o^r_i = 0$ otherwise). These additional parameters are summarized in Table 3. Note that a trip $r \in \mathcal{R}_v$ is infeasible for vessel v if $\hat{u}^{v_r}_{\text{start},d} - \hat{u}^{v_r}_{\text{start},0} > T$ or $\sum_{f \in \mathcal{F}: \mathcal{K}_f \cap \mathcal{N}_r \neq \emptyset} \xi^{v_r}_f > Q^{v_r}$. The trips defined here can be used in the modeling of path-flow formulation of the problem.

The decision variables in the model are defined below:

- x_r = binary variable indicating whether trip r is selected (where $x_r = 1$ if this occurs and $x_r = 0$ otherwise);
- $q^{v_r}_i$ = cargo flow delivered at node $i \in \mathcal{N}_r$ by vessel v_r of trip r;
- $u^{v_r}_{\text{start},i}$ = service start time of vessel v_r of trip r at node $i \in \mathcal{N}_r$;
- $u^{v_r}_{\text{end},i}$ = service end time of vessel v_r of trip r at node $i \in \mathcal{N}_r$;
- e_r = end time of trip r;
- $z^{v_r}_l$ = binary variable indicating whether vessel v_r of trip r starts the base service during time interval $\mathcal{T}^{lv_r}_1$ (where $z^{v_r}_l = 1$ if $u^{v_r}_{\text{start},0} \in \mathcal{T}^{lv_r}_1$ and $z^{v_r}_l = 0$ if $u^{v_r}_{\text{start},0} \in \mathcal{T}^{lv_r}_2$);

- $w_l^{v_r}$ = binary variable indicating whether vessel v_r of trip r starts the base service during time interval $\mathcal{T}_2^{lv_r}$ (where $w_l^{v_r}=1$ if $u_{\text{start},0}^{v_r} \in \mathcal{T}_2^{lv_r}$ and $w_l^{v_r}=0$ if $u_{\text{start},0}^{v_r} \in \mathcal{T}_1^{lv_r}$);
- ${z'}_l^{v_r}$ = binary variable indicating whether vessel v_r of trip r finishes the base service during time interval $\left[a_l^0, b_l^0\right]$ (where ${z'}_l^{v_r}=1$ if $u_{\text{end},0}^{v_r} \in \left[a_l^0, b_l^0\right]$ and ${z'}_l^{v_r}=0$ if $u_{\text{end},0}^{v_r} \notin \left[a_l^0, b_l^0\right]$);
- $\phi_i^{v_r v_{r'}}$ = binary variable indicating whether vessel v_r of trip r and vessel $v_{r'}$ of trip r' visit node $i \in \mathcal{N}_r \cap \mathcal{N}_{r'}$ (where $\phi_i^{v_r v_{r'}}=1$ if this occurs and $\phi_i^{v_r v_{r'}}=0$ otherwise); and
- $\psi_i^{v_r v_{r'}}$ = binary variable indicating whether vessel v_r of trip r and vessel $v_{r'}$ of trip r' visit node $i \in \mathcal{N}_r \cap \mathcal{N}_{r'}$, and vessel v_r completes its service at node i before vessel $v_{r'}$ arrives at the node (where $\psi_i^{v_r v_{r'}}=1$ if this occurs and $\psi_i^{v_r v_{r'}}=0$ otherwise).

Similar to the arc-flow model, the objective function of the path-flow model consists of travel cost and the vessel fixed cost. These quantities are defined as follows:

$$\text{Total Cost} = \sum_{r\in\mathcal{R}} c_r x_r + \sum_{v\in\mathcal{V}} c_{\text{fix}}^v \min\,(1, \sum_{r:v_r=v} x_r) \tag{28}$$

where, $c_r = \sum_{i,j\in\mathcal{N}_r} c_{ij}^{v_r}$. The overall objective function (28) should be minimized subject to the constraints described below. Note that in the constraint descriptions, $u_{\text{start},d}^{v_r} = e_r$ and $M > 0$ is a sufficiently large constant.

$$a_i x_r \le u_{\text{start},i}^{v_r} \le u_{\text{end},i}^{v_r} \le b_i x_r, \qquad r\in\mathcal{R}, i\in\mathcal{N}_r, \tag{29}$$

$$M\left(z_l^{v_r} + w_l^{v_r} - 1\right) + a_l^0 \le \tau_r \le b_l^0 + M\left(1 - z_l^{v_r} - w_l^{v_r}\right), \qquad r\in\mathcal{R}, l\in\left\{1,2,\ldots\left\lfloor\frac{\mathbb{T}}{24}\right\rfloor\right\}, \tag{30}$$

$$M\left({z'}_l^{v_r} - 1\right) + a_l^0 \le u_{\text{end},0}^{v_r} \le b_l^0 + M\left(1 - {z'}_l^{v_r}\right), \qquad r\in\mathcal{R}, l\in\left\{1,2,\ldots\left\lfloor\frac{\mathbb{T}}{24}\right\rfloor\right\}, \tag{31}$$

$$0 \le S \le \tau_r + M(1-x_r), \qquad r\in\mathcal{R}, \tag{32}$$

$$E \ge e_r x_r, \qquad r\in\mathcal{R}, \tag{33}$$

$$0 \le E - S \le T, \tag{34}$$

$$\sum_l (z_l^{v_r} + w_l^{v_r}) = x_r, \qquad r\in\mathcal{R}, \tag{35}$$

$$\sum_l {z'}_l^{v_r} = x_r, \qquad r\in\mathcal{R}, \tag{36}$$

$$\xi_f^{v_r} x_r \le q_i^{v_r}, \qquad r\in\mathcal{R}, f\in\mathcal{F}, i\in\mathcal{K}_f\cap\mathcal{N}_r, \tag{37}$$

$$\sum_{i\in\mathcal{N}_r} q_i^{v_r} \le Q^{v_r} x_r, \qquad r\in\mathcal{R}, \tag{38}$$

$$\sum_{r\in\mathcal{R}} \sum_{i\in\mathcal{K}_f\cap\mathcal{N}_r} q_i^{v_r} \ge q_f, \qquad f\in\mathcal{F}, \tag{39}$$

$$\sum_{r\in\mathcal{R}} \sum_{i\in\mathcal{K}_f\cap\mathcal{N}_r} o_i^r x_r \ge n_f, \qquad f\in\mathcal{F}, \tag{40}$$

$$u_{\text{end},i_j}^{v_r} + t_{i_j i_{j+1}}^{v_r} \le u_{\text{start},i_{j+1}}^{v_r} + M(1-x_r), \qquad r = \{0, i_1, \ldots, i_j, \ldots, d\}, \tag{41}$$

$$u_{\text{end},i}^{v_r} - u_{\text{start},i}^{v_r} \ge \delta_{\text{fix},i}^{v_r} x_r + \frac{q_i^{v_r}}{s_{f:i\in\mathcal{K}_f}^{v_r}} - M(1-x_r), \qquad r\in\mathcal{R}, i\in\mathcal{N}_r, \tag{42}$$

$$u_{\text{end},0}^{v_r} - \tau_r \ge \delta_0^{v_r} x_r + \left(a_{l+1}^0 - b_l^0\right)\left((\lambda^{v_r}-1)\sum_l z_l^{v_r} + \lambda^{v_r}\sum_l w_l^{v_r}\right), \qquad r\in\mathcal{R}, l\in\left\{1,2,\ldots\left\lfloor\frac{\mathbb{T}}{24}\right\rfloor\right\}, \tag{43}$$

$$e_r \le \tau_{r'} + M(2 - x_r - x_{r'}), \qquad v\in\mathcal{V}, r, r'\in\mathcal{R}_v : \tau_r \le \tau_{r'}, \tag{44}$$

$$e_r = \hat{e}_{r'} x_r + \sum_{r\in\mathcal{N}_r} \left(u_{\text{end},i}^{v_r} - \hat{u}_{\text{end},i}^{v_r} x_r\right), \qquad r\in\mathcal{R}, \tag{45}$$

$$\phi_i^{v_r v_{r'}} \ge \psi_i^{v_r v_{r'}}, \qquad r, r'\in\mathcal{R}, f\in\mathcal{F}, i\in\mathcal{K}_f, \tag{46}$$

$$\phi_i^{v_r v_{r'}} \geq 1 - M(2 - o_i^r x_r - o_i^{r'} x_{r'}), \qquad r, r' \in \mathcal{R}, f \in \mathcal{F}, i \in \mathcal{K}_f, \tag{47}$$

$$2\phi_i^{v_r v_{r'}} \leq o_i^r x_r + o_i^{r'} x_{r'}, \qquad r, r' \in \mathcal{R}, f \in \mathcal{F}, i \in \mathcal{K}_f, \tag{48}$$

$$u_{\text{end},i}^{v_r} \leq u_{\text{start},i}^{v_{r'}} + M(2 - \phi_i^{v_r v_{r'}} - \psi_i^{v_r v_{r'}}), \qquad r, r' \in \mathcal{R}, f \in \mathcal{F}, i \in \mathcal{K}_f, \tag{49}$$

$$u_{\text{start},i}^{v_r} \geq u_{\text{end},i}^{v_{r'}} - M(1 - \phi_i^{v_r v_{r'}}) - M\psi_i^{v_r v_{r'}}, \qquad r, r' \in \mathcal{R}, f \in \mathcal{F}, i \in \mathcal{K}_f. \tag{50}$$

Constraints (29) state that vessel must start and finish the service at facilities when they are open. Constraints (30) and (31) ensure that the base service of a trip starts and finishes when the vessel base is open. The schedule start time, end time and duration are defined by constraints (32)-(34). When a vessel starts a trip, it must undergo a base service (constraints (35) and (36)). Constraints (37)-(40) govern the minimum offload on each trip, the vessel capacity, the weekly cargo demand and the visit frequency at facilities, respectively. Time sequencing constraints are defined by constraints (41). Constraints (42) ensure that if a vessel visits a facility, then the service duration spans the fixed service time and variable unloading time. The base service time that may be interrupted due to base closure is defined by constraints (43). Constraints (44) guarantee that a vessel is on a single trip at any given time. The end time of a trip is measured by the difference between actual service finish time and minimum service finish time at each node on the trip (constraints (45)). The final group of constraints (46)-(50) ensure that different vessels do not visit the same offshore facility at the same time. $\phi_i^{v_r v_{r'}}$ and $\psi_i^{v_r v_{r'}}$ are defined by constraints (46)-(48) and clash-free visits at facilities are ensured by constraints (49) and (50).

Solving the resulting path-flow formulation is computationally intractable because unlike classical PSVP problem, generating the trips a priori is impossible. This is due to the fact that cargo deliveries are cot discrete and service duration is variable. Hence, we do not move in this direction. Instead, we apply the column generation technique to make the problem tractable. As pointed out by Barnhart et al. (1998), through column generation, we only consider those variables in the linear programming (LP) relaxation that are very likely to appear in the optimal solution. In Section 3, we present the Dantzig-Wolfe decomposition of the periodic supply vessel planning problem under study.

3 Dantzig-Wolfe Decomposition

Dantzig-Wolfe decomposition technique (Dantzig and Wolfe, 1960) reformulates a linear programming problem, with huge number of variables, and generates a master problem and a pricing problem. The pricing problem can be an optimization problem such as mixed-integer linear program (Costa et al., 2019). In what follows, we present the master problem, the pricing problem, and branching decisions.

3.1 The master problem

We define the following notations used in the master problem as well as the pricing problem.

- θ_r^t: a binary parameter that indicates whether vessel $v_r \in \mathcal{V}$ taking trip $r \in \mathcal{R}$ is in operation during time interval t.
- $\theta_{ri}^{\prime t}$: a binary parameter that indicates whether vessel $v_r \in \mathcal{V}$ taking trip $r \in \mathcal{R}$ is at node $i \in \mathcal{N}_r$ during time interval t.
- $h_{ij}^{v_r}$: a binary parameter that indicates whether vessel $v_r \in \mathcal{V}$ taking trip $r \in \mathcal{R}$ traverses link $(i, j) \in \mathcal{A}$.
- $u_{\text{start},0}^{v_r}$: a continuous parameter that indicates the start time of vessel $v_r \in \mathcal{V}$ taking trip $r \in \mathcal{R}$ at the base.
- $u_{\text{end},0}^{v_r}$: a continuous parameter that indicates the end time of vessel $v_r \in \mathcal{V}$ taking trip $r \in \mathcal{R}$ at the base.
- m_r^f: an integer parameter that indicates the number of times that facility $f \in \mathcal{F}$ is visited on trip $r \in \mathcal{R}$.
- $u_{\text{start},d}^{v_r}$: a continuous parameter that indicates the arrival time of vessel $v_r \in \mathcal{V}$ taking trip $r \in \mathcal{R}$ at the base at end of trip.

- $q_i^{v_r}$: a continuous parameter that indicates the amount of cargo delivered by vessel $v_r \in \mathcal{V}$ taking trip $r \in \mathcal{R}$ at node $i \in \mathcal{N}_r$.

- g_v: a binary decision variable that indicates whether vessel $v \in \mathcal{V}$ is utilized.

The master problem will be as follows

$$\min \quad \sum_{r\in\mathcal{R}} c_r x_r + \sum_{v\in\mathcal{V}} c_{\text{fix}}^{v_r} g_v \tag{51}$$

subject to

$$\sum_{r\in\mathcal{R}} \sum_{i\in\mathcal{K}_f\cap\mathcal{N}_r} q_i^{v_r} x_r \geq q_f, \qquad f \in \mathcal{F}, \tag{52}$$

$$\sum_{r:v_r=v} \theta_r^t x_r \leq 1, \qquad t \in \{0,1,2,\ldots,T\}, v \in \mathcal{V}, \tag{53}$$

$$\sum_{r\in\mathcal{R}} \theta'^t_{ri} x_r \leq 1, \qquad t \in \{0,1,2,\ldots,T\}, f \in \mathcal{F}; i \in \mathcal{K}_f, \tag{54}$$

$$S - u_{\text{start},0}^{v_r} x_r \leq 0, \qquad r \in \mathcal{R}, \tag{55}$$

$$E - u_{\text{start},d}^{v_r} x_r \geq 0, \qquad r \in \mathcal{R}, \tag{56}$$

$$E - S \leq T, \tag{57}$$

$$\sum_{r\in\mathcal{R}} m_r^f x_r \geq n_f, \qquad f \in \mathcal{F}, \tag{58}$$

$$\sum_{i\in\mathcal{N}_r} \sum_{j\in\mathcal{N}_r} \sum_{r:v_r=v} h_{ij}^{v_r} x_r \leq \overline{M} g_v, \qquad v \in \mathcal{V}. \tag{59}$$

The objective function (51) minimizes the travel cost and fixed cost of using a vessel. Constraints (52) satisfy the demands of facilities. Overlapping of the same vessel on multiple trips is excluded through constraints (53) according to which, each vessel can be used at most once during any interval. Constraints (54) guarantee that at any time there is at most one vessel at each facility. Constraints (55)-(57) limit the duration of all trips. Visit frequency of each facility is satisfied by constraints (58). Constraints (59) ensure that a vessel is used if it only takes at least one trip. The coefficient $\overline{M}$ in this constraint is defined to guarantee the feasibility of the relaxed problem. Its quantity and coefficient of g_v in the objective function impact the difficulty of the model. One possible value for $\overline{M}$ that guarantees feasibility is $|\mathcal{N}| = TW|\mathcal{F}|$ which means if trip r visits all nodes in the network, the constraint is feasible.

An optimal solution of the relaxed master problem provides a lower bound for the branch-and-bound tree. Since this master problem has a substantial number of decision variables, we apply the column generation technique. In the column generation process, we consider a subset of decision variables forming the RMP. Once solved to optimality, the dual variables for constraints (52)-(59) will be used in the pricing problem (see subsection 3.2) in order to find the new decision variables x_r with negative reduced costs. The process continues until no such variable exists.

3.2 The pricing problem

The role of the pricing problem is to find new promising variables (columns) for the master problem. As mentioned before, at each iteration, the dual variables of the RMP are used to find columns with the least reduced cost. First, let us define the following dual variables:

- α_f: dual variable for constraint (52) for facility $f \in \mathcal{F}$.

- β_t^v: dual variable for constraint (53) for each vessel and time interval $t \in \{0,1,2,\ldots,T\}$.

- γ_{fi}^t: dual variable for constraint (54) for each facility f, each node $i \in \mathcal{K}_f$ and time interval t.

- π_v: dual variable for constraint (55) for each vehicle $v \in \mathcal{V}$.

- σ_v: dual variable for constraint (56) for each trip $r \in \mathcal{R}$.

- κ: dual variable for constraint (57).
- μ_f: dual variable for constraint (58) for facility $f \in \mathcal{F}$.
- η_v: dual variable for constraint (59) for each vehicle $v \in \mathcal{V}$.

Recall that the binary decision variable y_{ij}^v, $v \in \mathcal{V}$ indicates whether vessel v travels through the link connecting nodes i and j. The reduced cost of variable x_r defines the objective function of the pricing problem for each vessel v and trip r as follows:

$$\begin{aligned}\min \quad & \sum_i \sum_j c_{ij}^{v_r} y_{ij}^{v_r} - \sum_{i \in \mathcal{K}_f \cap \mathcal{N}_r} \alpha_f q_i^{v_r} - \sum_t \beta_t^{v_r} \theta_r^t \\ & - \sum_t \sum_{i \in \mathcal{K}_f \cap \mathcal{N}_r} \gamma_i^t \theta'^t_{ri} - \pi_{v_r} u_{\text{start},0}^{v_r} - \sigma_{v_r} u_{\text{start},d}^{v_r} - \sum_f \mu_f m_r^f - \sum_{i \in \mathcal{N}_r} \eta_{v_r} y_{0,i}^{v_r} \end{aligned} \tag{60}$$

subject to:

constraints (30), (31), (35), (36) and (43),

$$\sum_f \sum_{i \in \mathcal{K}_f \cap \mathcal{N}_r} y_{0,i}^{v_r} = 1, \tag{61}$$

$$\sum_{\substack{j \in \mathcal{N}_r \\ j \neq i}} y_{ij}^{v_r} - \sum_{\substack{j \in \mathcal{N}_r \\ j \neq i}} y_{ji}^{v_r} = 0, \qquad i \in \mathcal{N}_r, \tag{62}$$

$$\sum_{i \in \mathcal{N}_r} y_{i,d}^{v_r} = 1, \tag{63}$$

$$q_i^{v_r} \leq Q^{v_r} \sum_{j \in \mathcal{N}_r} y_{ij}^{v_r}, \qquad i \in \mathcal{K}_f, \tag{64}$$

$$\sum_{i \in \mathcal{N}_r} q_i^{v_r} \leq Q^{v_r}, \tag{65}$$

$$\xi_f^{v_r} y_{ij}^{v_r} \leq q_i^{v_r}, \qquad i, j \in \mathcal{K}_f \cap \mathcal{N}_r f \in \mathcal{F}, \tag{66}$$

$$q_j^{v_r} \leq q_f \sum_{\substack{i \in \mathcal{N}_r \cap \mathcal{K}_f \\ i \neq j}} y_{ij}^{v_r}, \qquad j \in \mathcal{K}_f \cap \mathcal{N}_r, f \in \mathcal{F}, \tag{67}$$

$$m_r^f \geq \sum_{i \in \mathcal{N}_r} \sum_{\substack{j \in \mathcal{K}_f \\ i \neq j}} y_{ij}^{v_r}, \qquad f \in \mathcal{F}, \tag{68}$$

$$a_i \sum_{j \in \mathcal{N}_r} y_{ij}^{v_r} \leq u_{\text{start},i}^{v_r} \leq u_{\text{end},i}^{v_r} \leq b_i \sum_{j \in \mathcal{N}_r} y_{ij}^{v_r}, \qquad i \in \mathcal{N}_r, \tag{69}$$

$$u_{\text{end},i}^{v_r} + t_{ij}^{v_r} \leq u_{\text{start},j}^{v_r} + M(1 - y_{ij}^{v_r}), \qquad i, j \in \mathcal{N}_r, \tag{70}$$

$$u_{\text{end},i}^{v_r} - u_{\text{start},i}^{v_r} \geq \delta_{\text{fix},i}^{v_r} \sum_{j \in \mathcal{N}} y_{ij}^{v_r} + \frac{q_i^{v_r}}{s_f^{v_r}}, \qquad i \in \mathcal{K}_f \cap \mathcal{N}_r, f \in \mathcal{F}, \tag{71}$$

$$\sum_l (z_l^{v_r} + w_l^{v_r}) = 1, \tag{72}$$

$$\sum_l z'^{v_r}_l = 1, \tag{73}$$

$$M\left(z_l^{v_r} + w_l^{v_r} - 1\right) + a_l^0 \leq u_{\text{start},0}^{v_r} \leq b_l^0 + M\left(1 - z_l^{v_r} - w_l^{v_r}\right), \qquad l \in \left\{1, 2, \dots \left\lfloor \frac{\mathbb{T}}{24} \right\rfloor \right\}, \tag{74}$$

$$M\left(z'^{v_r}_l - 1\right) + a_l^0 \leq u_{\text{end},0}^{v_r} \leq b_l^0 + M\left(1 - z'^{v_r}_l\right), \qquad l \in \left\{1, 2, \dots \left\lfloor \frac{\mathbb{T}}{24} \right\rfloor \right\}, \tag{75}$$

$$u_{\text{end},0}^{v_r} - u_{\text{start},0}^{v_r} \geq \delta_0^{v_r} + \left(a_{l+1}^0 - b_l^0\right)\left((\lambda^{v_r} - 1)\sum_l z_l^{v_r} + \lambda^{v_r} \sum_l w_l^{v_r}\right), \tag{76}$$

$$l \in \left\{1, 2, \dots \left\lfloor \frac{\mathbb{T}}{24} \right\rfloor \right\}, \tag{77}$$

$$u^{v_r}_{\text{start},d} - u^{v_r}_{\text{start},0} \le T; \tag{78}$$

$$u^{v_r}_{\text{start},0} \le t + M(1 - \theta^t_r); \qquad t \in \{0, 1, 2, \dots, T\}, \tag{79}$$

$$u^{v_r}_{\text{start},d} \ge t - M(1 - \theta^t_r); \qquad t \in \{0, 1, 2, \dots, T\}, \tag{80}$$

$$u^{v_r}_{\text{start},i} \le t + M(1 - \theta'^t_{ri}), \qquad t \in \{0, 1, 2, \dots, T\}, i \in \mathcal{N}_r, \tag{81}$$

$$u^{v_r}_{\text{end},i} \ge t - M(1 - \theta'^t_{ri}), \qquad t \in \{0, 1, 2, \dots, T\}, i \in \mathcal{N}_r. \tag{82}$$

The mixed integer linear programming (MILP) problem defined by (60)-(82) should be solved for every vessel v and every trip r in order to find new columns for the restricted master problem. Since this MILP is difficult to solve, we modify the label extension algorithm by Desaulniers (2010), which is based on Elementary Shortest Path Problem with Resource Constraints (ESPPRC), and will discuss it in Section 4.1.

4 Column Generation

The master problem presented in subsection 3.1 has many decision variables and it is intractable to list each one. Therefore, we employ the column generation approach to iteratively add new columns (decision variables) to the linear programming (LP) relaxation of (51)-(59). Starting with a limited number of columns at node 0 of the branch-and-bound tree, at each iteration, this LP is solved, and the dual variable of each constraint is obtained. This LP provides a lower bound for each node of the branch-and-bound tree. Then, the dual variables are passed onto pricing problem to find new variables x_r with negative reduced costs. If there is no such variable, the current optimal solution is an optimal solution for the master problem. For further details on column generation see (Desaulniers et al., 2006; Desaulniers, 2010; Feillet, 2010). During the process, branching is required if the solution of any integer variable is fractional.

4.1 A label extension algorithm for solving sub-problem

As mentioned the sub-problem given by (60)-(82) has the specifications of Elementary Shortest Path Problem with Resource Constraints (ESPPRC). The ESPPRC is $\mathcal{NP}$-hard, so is our pricing problem (Costa et al., 2019; Dror, 1994). Since the seminal work of Feillet et al. (2004) for solving ESPPRC, many researches used a variant of the algorithm for solving pricing problems in the context of vehicle routing problem (Stålhane et al., 2012; Hernandez et al., 2016; Desaulniers, 2010; Dell'Amico et al., 2006). In this paper, we propose a forward dynamic programming algorithm based on the algorithms developed by Hernandez et al. (2016) and Desaulniers (2010). In our approach, for each feasible path from node 0 to node $i \in \mathcal{N}$, a label L_i is defined. A label for node i is *any feasible path from node 0 to node i.* We define any label $L_i = (\overline{C}_i, \text{pos}(i), \rho_i, \epsilon_i, q_i, Q_i, l_i, \mathbf{N}_i, \mathbf{V}_i, \mathbf{R}_i)$ with $|\mathcal{N}|+2|\mathcal{F}|+7$ components. In this representation, $\overline{C}_i$ is the reduced cost of the partial path from node 0 to node i, $\text{pos}(i) = i$ indicates the position of label i, ρ_i is the service start time at i, ϵ_i is the service end time at i, q_i is the amount of delivery to node i, Q_i is the accumulated delivery up to node i, $l_i = \epsilon_i - \rho_0$ is the duration of the path up to i, $\mathbf{N}_i$ is a $(1 \times |\mathcal{F}|)$ integer vector which shows the number of times that facility f is visited up to i, $\mathbf{V}_i$ is a $(1 \times |\mathcal{N}|)$ binary vector which shows the unreachability of a node form node i, (this set is very important since it reduces the computation time of the dynamic programming approach), and $\mathbf{R}_i$ is a $(1 \times |\mathcal{F}|)$ vector that shows the remaining demand of facility f at node i. Vector $\mathbf{N}_i$ enables us to keep track of required visit frequency of each facility. The reason of adding vector $\mathbf{R}_i$ to the label is that a label may visit a facility more than once and during each visit a fixed amount of cargo must be delivered which fulfills a portion of a demand. The main constraints are capacity of vessels, time window at each node, and duration of the path. In our labeling setting, components ρ_i and ϵ_i are associated with time windows $[a_i, b_i]$, component Q_i is associated with vessel capacity, and l_i guarantees that constraint (78) is not violated.

In this algorithm, we start from the base, node 0, and for each discretized time instant t such that

$0 \leq t < T$, we define a label as

$$L_0^t = (0, 0, \rho_0^t, \epsilon_0^t, 0, 0, l_0^t, (\underbrace{0, 0, \ldots, 0}_{|\mathcal{F}|}), (\underbrace{0, 0, \ldots, 0}_{|\mathcal{N}|}), (\underbrace{q_{f_1}, q_{f_2}, \ldots, q_{f_{|\mathcal{F}|}}}_{|\mathcal{F}|})), \qquad t \in \{0, 1, 2, \ldots, T-1\}.$$

Then, we update parameter ϵ_0^t based on the service start time at node 0. If $\rho_0^t \in \cup_{\ell \in \mathcal{L}} \mathcal{T}_1^{\ell v}$, then ϵ_0^t is calculated by

$$\epsilon_0^t = \rho_0^t + \delta_0^v + (a_2^0 - b_1^0)(\lambda^v - 1),$$

and if $\rho_0^t \notin \cup_{\ell \in \mathcal{L}} \mathcal{T}_1^{\ell v}$, or equivalently, $\rho_0^t \in \cup_{\ell \in \mathcal{L}} \mathcal{T}_2^{\ell v}$, then

$$\epsilon_0^t = \rho_0^t + \delta_0^v + (a_2^0 - b_1^0)\lambda^v.$$

Since this algorithm is a constructive one, a new label L_j should be extended from label L_i in order to form an updated partial path. After extension the new label $L_j = (\overline{C}_j, \text{pos}(j), \rho_j, \epsilon_j, q_j, Q_j, l_j, \mathbf{N}_j, \mathbf{V}_j, \mathbf{R}_j)$ is updated as follows:

$$\begin{aligned}
&\text{pos}(j) = j \\
&\rho_j = \max\{\epsilon_i + t_{ij}^v, a_j\} \\
&\epsilon_j = \min\{\rho_j + \frac{q_j}{s_{f:j\in\mathcal{K}_f}} + \delta_{\text{fix},j}^v, b_j\} \\
&q_j = \max\left\{\xi_{f:j\in\mathcal{K}_f}^v, \min\{\mathbf{R}_i(f)_{j\in\mathcal{K}_f}, (b_j - \rho_j - \delta_{\text{fix},j}^v)s_{f:j\in\mathcal{K}_f}, Q^v - Q_i\}\right\} \\
&Q_j = Q_i + q_j \\
&l_j = \epsilon_j - \rho_0 \\
&\begin{cases} \mathbf{N}_j(f) = \mathbf{N}_i(f) & ; \quad j \notin \mathcal{K}_f \\ \mathbf{N}_j(f) = \mathbf{N}_i(f) + 1 & ; \quad j \in \mathcal{K}_f \end{cases} \\
&\begin{cases} \mathbf{R}_j(f) = \mathbf{R}_i(f) & ; \quad j \notin \mathcal{K}_f \\ \mathbf{R}_j(f) = \mathbf{R}_i(f) - q_j & ; \quad j \in \mathcal{K}_f. \end{cases}
\end{aligned}$$

In above process, ρ_j guarantees that the service at node j starts when the node is open, and the vessel should wait if it arrives before the time-window. Similarly, ϵ_j ensures that the time window constraint and the fix service-time requirement are met. The quantity delivered at node j takes into account minimum delivery, remaining demand of each facility, and available capacity of the vessel. As mentioned earlier, $\mathbf{N}_j(f)$ and $\mathbf{R}_j(f)$ ensure that the number of visits and the remaining demand of facility f is updated at node j. Using these values, the reduced cost of feasible extension L_j is updated as follows:

$$\overline{C}_j = \begin{cases}
\overline{C}_i + c_{ij}^v - \sum_{t=\epsilon_i}^{t=\epsilon_j} \beta_t - \sigma_v u_{\text{start},d}^v, & \text{if } j = d, \\
\begin{cases} \overline{C}_i + c_{\text{fix}}^v + +c_{ij}^v - \alpha_{f:j\in\mathcal{K}_f} q_j - \sum_{t=\epsilon_i}^{t=\epsilon_j} \beta_t - \sum_{t=\epsilon_i+t_{ij}}^{t=\epsilon_j} \gamma_j \\ -\pi_v u_{\text{start},0}^v - \mu_{f:j\in\mathcal{K}_f} \mathbf{N}_j(f) - \eta_v, \end{cases} & \text{if } i = 0, \\
\overline{C}_i + c_{ij}^v - \alpha_{f:j\in\mathcal{K}_f} q_j - \sum_{t=\epsilon_i}^{t=\epsilon_j} \beta_t - \sum_{t=\epsilon_i+t_{ij}}^{t=\epsilon_j} \gamma_j - \mu_{f:j\in\mathcal{K}_f} \mathbf{N}_j(f) & \text{else.}
\end{cases}$$

Therefore, the definition and calculation of the reduced cost depends on the location of node j.

Another parameter of the label that must be updated after extension is $\mathbf{V}_j$, the set of nodes that are unreachable from node j once the partial path is extended to node j. It should be noted that if node j is unreachable from node i, then $\mathbf{V}_i(j) = 1$; otherwise, $\mathbf{V}_i(j) = 0$, i.e., label L_j can be extended from label L_i. The vector $\mathbf{V}_j$ is updated as follows:

1. $V_j(j) = 1$;
2. $V_j(k) = 1$, if $V_i(k) = 1$ and $k \neq i, j$;
3. if $\max\{\epsilon_j + t_{jk}^v, a_k\} > b_k$, then $V_j(k) = 1$;
4. if $b_k - \max\{\epsilon_j + t_{jk}^v, a_k\} < \delta_{\text{fix},k}^v + \frac{\xi_{f:k\in\mathcal{K}_f}^v}{s_{f:k\in\mathcal{K}_f}}$, then $V_j(k) = 1$;

5. if $Q_j + \xi^v_{f:k\in\mathcal{K}_f} > Q^v$, then $V_j(k) = 1$;

6. $V_j(k) = 1$, if $N_i(f) \geq n_f, k \in \mathcal{K}_f$;

7. $V_j(k) = 1$, if $R_i(f) = 0, k \in \mathcal{K}_f$;

8. if none of the above conditions were met, then $V_j(k) = 0$.

Item 1 is clear. Item 2 states that if node i cannot be extended to node k, any extension of it cannot be extended to node k either. Item 3 checks the time window of the reachable nodes. Item 4 checks if there is enough time for delivery at reachable nodes. If there is not enough time, then node k is not reachable from node j. Item 5 checks the capacity of the vessel for reachable nodes. Items 6 and 7 check the visit and demand requirements of node k before extension from node j. If a node passes all these checks, it is a reachable node. Otherwise, it is unreachable and we do not have to extend to that node any further. Moreover, when extending a label L_i from node i to node j as label L_j, the following feasibility checks must be investigated:

1. Label L_j is infeasible if $\rho_j \geq b_j$ or $b_j - \max\{\epsilon_i + t^v_{ij}, a_j\} < \delta^v_{\text{fix},j} + \frac{\xi^v_{f:j\in\mathcal{K}_f}}{s_{f:j\in\mathcal{K}_f}}$.

2. Label L_j is infeasible if $Q_j > Q^v$.

3. Label L_j is infeasible if $l_j > T$.

In case 1, if in label L_j, service starts after time window b_j or there is not enough time for minimum delivery, then label L_j is regarded as an infeasible extension. Cases 2 and 3 guarantee that the vessel capacity and schedule duration constraints are not violated.

In label extension algorithm usually a huge number of labels may be created that slows down the algorithm performance. To avoid this, a set of dominance rules are set up to eliminate dominated labels (Hernandez et al., 2016; Desaulniers, 2010). Consider two partial labels from the base, node 0, to node j as L_j and L'_j. Label L_j dominates label L'_j if following conditions hold:

$$\begin{cases} 1. & l_j \leq l_{j'}, \\ 2. & Q_j \geq Q_{j'}, \\ 3. & \overline{C}_j \leq \overline{C}_{j'}, \\ 4. & \mathbf{V}_j \leq \mathbf{V}_{j'}, \\ 5. & \mathbf{N}_j \leq \mathbf{N}_{j'}, \mathbf{N}_j(f) \geq n_f \text{ and } \mathbf{N}_{j'}(f) \geq n_f, \forall f \\ 6. & \mathbf{R}_j(f) \leq \mathbf{R}_{j'}(f), \forall f. \end{cases}$$

To prove the above dominance rules, we follow the approach presented by Huang et al. (2021) which is through induction. Without loss of generality, we assume that extensions from L_j and L'_j are labels L_k and L'_k, respectively. Then, we just need to show that L'_j will return dominated value. Since the proofs for all rules are similar, for brevity we just prove the first rule. For rule 1, it is easy to show that

$$\begin{aligned} l_j \leq l'_j \Rightarrow \quad & \epsilon_j - \rho_0 \leq \epsilon'_j - \rho_0, \\ & \epsilon_j \leq \epsilon'_j, \\ & \epsilon_j + t^v_{jk} \leq \epsilon'_j + t^v_{jk} \Rightarrow \\ & \Rightarrow \begin{cases} (1): & \epsilon_j + t^v_{jk} \leq \epsilon'_j + t^v_{jk} \leq a_k \Rightarrow \rho_k = \rho'_k = a_k \\ \\ (2): & a_k \leq \epsilon_j + t^v_{jk} \leq \epsilon'_j + t^v_{jk} \Rightarrow \rho_k = \epsilon_j + t^v_{jk} \leq \epsilon'_j + t^v_{jk} = \rho'_k \\ \\ (3): & \epsilon_j + t^v_{jk} \leq a_k \leq \epsilon'_j + t^v_{jk} \Rightarrow \rho_k = a_k \leq \epsilon'_j + t^v_{jk} = \rho'_k \end{cases} \\ & (1), (2), (3) \Rightarrow \rho_k \leq \rho'_k. \end{aligned}$$

Before calculating the service finish time at node k, we need to find q_k.

Case 1: From

$$q_k = \max\left\{\xi_{f:k\in\mathcal{K}_f}, \min\{\mathbf{R}_j(f)_{k\in\mathcal{K}_f}, (b_k - \rho_k - \delta^v_{\text{fix},k})s_{f:k\in\mathcal{K}_f}, Q^v - Q_j\}\right\},$$

Algorithm 1 The proposed label-setting algorithm

1: Create all initial labels
2: Set $UL_0 = \cup_m\{L_0^m\}$, $SL_0 = \emptyset$, $m = 1, 2, \dots, \lfloor \frac{\mathbb{T}}{24} \rfloor (b_1^0 - a_1^0)$
3: Apply dominance check and keep only dominant labels.
4: **for** $i \in \mathcal{N}$ **do**
5: Set $UL_i = \emptyset$ and $SL_i = \emptyset$.
6: **end for**
7: **while** $\bigcup_{i\in\mathcal{N}} UL_i \neq \emptyset$ **do**
8: Choose a label $L_i \in UL_i$, where $UL_i \neq \emptyset$.
9: **for all** $(i, j) \in \mathcal{A}$ **do**
10: Extend L_i to create L_j.
11: **if** L_j is feasible **then**
12: Update $UL_j = UL_j \cup \{L_j\}$.
13: Discard from the set $UL_j \cup SL_j$ the dominated labels.
14: **end if**
15: Set $UL_i = UL_i \setminus \{L_i\}$ and $SL_i = SL_i \cup \{L_i\}$.
16: **end for**
17: A label $L \in SL_d$ with minimum $\overline{C}$ provides a shortest path from 0 to d.
18: **end while**

only the second value in the min function depends on time, and if $q_k = (b_k - \rho_k - \delta^v_{\text{fix},k})s_{f:k\in\mathcal{K}_f}$, then by substituting it in service end time equation, we have

$$\left.\begin{aligned} \epsilon_k &= \min\{\rho_k + \tfrac{q_k}{s_{f:k\in\mathcal{K}_f}} + \delta^v_{\text{fix},k}, b_k\} = \min\{\rho_k + \tfrac{(b_k-\rho_k-\delta^v_{\text{fix},k})s_{f:k\in\mathcal{K}_f}}{s_{f:k\in\mathcal{K}_f}} + \delta^v_{\text{fix},k}, b_k\} \\ \epsilon'_k &= \min\{\rho'_k + \tfrac{q'_k}{s_{f:k\in\mathcal{K}_f}} + \delta^v_{\text{fix},k}, b_k\} = \min\{\rho'_k + \tfrac{(b_k-\rho'_k-\delta^v_{\text{fix},k})s_{f:k\in\mathcal{K}_f}}{s_{f:k\in\mathcal{K}_f}} + \delta^v_{\text{fix},k}, b_k\} \end{aligned}\right\} \Rightarrow \epsilon_k = \epsilon'_k = b_k.$$

Case 2: Let $q_k = \xi_{f:k\in\mathcal{K}_f}$. By substituting it in service end time equation, we have

$$\epsilon_k = \min\{\rho_k + \frac{\xi_{f:k\in\mathcal{K}_f}}{s_{f:k\in\mathcal{K}_f}} + \delta^v_{\text{fix},k}, b_k\}$$

and

$$\epsilon'_k = \min\{\rho'_k + \frac{\xi_{f:k\in\mathcal{K}_f}}{s_{f:k\in\mathcal{K}_f}} + \delta^v_{\text{fix},k}, b_k\}.$$

Since $\rho_k \leq \rho'_k \Rightarrow \rho_k + \frac{\xi_{f:k\in\mathcal{K}_f}}{s_{f:k\in\mathcal{K}_f}} + \delta^v_{\text{fix},k} \leq \rho'_k + \frac{\xi_{f:k\in\mathcal{K}_f}}{s_{f:k\in\mathcal{K}_f}} + \delta^v_{\text{fix},k}$, we have three possibilities as follows

$$\begin{cases} (1) \quad \rho_k + \frac{\xi_{f:k\in\mathcal{K}_f}}{s_{f:k\in\mathcal{K}_f}} + \delta^v_{\text{fix},k} \leq \rho'_k + \frac{\xi_{f:k\in\mathcal{K}_f}}{s_{f:k\in\mathcal{K}_f}} + \delta^v_{\text{fix},k} \leq b_k \quad \Rightarrow \epsilon_k \leq \epsilon'_k \\ (2) \quad b_k \leq \rho_k + \frac{\xi_{f:k\in\mathcal{K}_f}}{s_{f:k\in\mathcal{K}_f}} + \delta^v_{\text{fix},k} \leq \rho'_k + \frac{\xi_{f:k\in\mathcal{K}_f}}{s_{f:k\in\mathcal{K}_f}} + \delta^v_{\text{fix},k} \\ (3) \quad \rho_k + \frac{\xi_{f:k\in\mathcal{K}_f}}{s_{f:k\in\mathcal{K}_f}} + \delta^v_{\text{fix},k} \leq b_k \leq \rho'_k + \frac{\xi_{f:k\in\mathcal{K}_f}}{s_{f:k\in\mathcal{K}_f}} + \delta^v_{\text{fix},k} \end{cases}$$

From (1), (2), and (3) it is easy to show that $\epsilon_k \leq \epsilon'_k \Rightarrow \epsilon_k - \rho_0 \leq \epsilon'_k - \rho_0 \Rightarrow l_k \leq l'_k$.

Case 3. If $q_k = Q^v - Q_j$, the values of ϵ_k and ϵ'_k depend on ρ_k and ρ'_k. Therefore, the proof is similar to case 2.

Case 4. In this case, we have $q_k = \mathbf{R}_j(f)_{k\in\mathcal{K}_f}$, therefore, as in cases 2 and 3, we have $\epsilon_k \leq \epsilon'_k \Rightarrow \epsilon_k - \rho_0 \leq \epsilon'_k - \rho_0 \Rightarrow l_k \leq l'_k$ and this completes the proof for rule 1.

Now that the label extension and dominance rules are discussed, we can present the extension procedure illustrated in Algorithm 1. Let us define UL_i as the set of un-scanned labels at node i and SL_i as the set of scanned labels at node i. Lines 1 through 6 present all initialization. In line 10, labels are extended from UL_i. The dominance rule is applied in line 13. Sets UL_i and SL_i are then updated in line 15. Once all label from node 0 to the destination are created, the one with the minimum reduced cost provides a new column for the master problem.

5 Branch-and-Price

The combination of branch-and-bound and column generation, resulting in the so-called branch-and-price, has gained numerous attention over the last three decades and is one of the leading exact algorithms for solving routing problems (Costa et al., 2019). The branch-and-price algorithm works iteratively on two sub-problems called master problem and pricing problem. At each iteration, the linear relaxation of a restricted master problem (RMP) with a subset of variables is solved by column generation to provide lower bounds for the branch-and-bound algorithm, and by using the pricing problem new variables are defined and added to the RMP. If the new RMP does not yield an integer solution, we branch on fractional variables in the branch-and-bound tree.

5.1 Initial solution

To start the column generation, we provide the master problem with initial feasible columns at node 0 of the branch-and-bound tree. We consider two heuristics to generate solution and initiate the column generation. In the first heuristic, we consider one vessel for each facility $f \in \mathcal{F}$, that visits the facility n_f times in different time windows. Therefore, the first master problem includes $|\mathcal{F}|$ trips that originate from the base and terminate at the base. In this approach, depending on the facility demand, q_f, a vessel v is assigned to the facility such that $Q^v \geq q_f$. Please note that if a facility's demand is bigger than capacity of any vessel, the second heuristic described below can still generate feasible solutions.

The second heuristics is depicted in Algorithm 2. This algorithm creates two sorted lists of facility demands and vessel capacities (lines 1-2). In the first list, the higher the demand, the higher the priority. Vessels are sorted based on the decreasing order of their capacities. Next, we create an empty set of trips. The main loop of the algorithm starts from line 5, which keeps track of the unsatisfied $TotalDemand$. Then, we select the first vessel from the list of vessels with the remaining positive capacity and create an empty $TempTrip$ (lines 6-7). Next, we check the possibility of adding a node from the list of available facilities and deliver Q_i of cargo. This value is calculated based on the remaining cargo on vessel, $\delta^v_{\text{fix},i}$, and available time until the end of the time window. The time window and its associated node are determined based on the vessel arrival time at the facility. For example, if the vessel arrives before or within the time window on day i, then the ith node of the facility will be considered. Otherwise, the $(i+1)$th node is the node of interest, meaning that the vessel must wait until the next day. The next step is to update the remaining capacity of the vessel and the unfulfilled demand of the facility. Once the node is selected, it will be added to $TempTrip$ (lines 9-16). This process continues until no more nodes can be added. In this case, $TempTrip$ is added to $Trips$ and the algorithm repeats from line 7. At the end of the algorithm, a list of feasible trips will be available that are added to the columns of RMP at node 0.

5.2 Branching

When the solution of LP relaxation at each node of branch-and-bound tree is not binary, a branching is required. Our branching decision is on whether a particular arc (i, j) is included in trip r or not. To further explore, let $\tilde{\mathbf{x}}$ be a solution of LP relaxation at node 0 of the branch-and-bound tree. In this solution, we choose x_{rv} that its value is closest to 0.5 to branch on. In this variable, there exist an arc (i, j) that makes x_{rv} fractional and its flow (equivalent to y^v_{ij} in the sub-problem) is fractional as well. In this case, we branch as follows. On a first branch, we impose $y^v_{ij} = 0$ meaning that arc (i, j) should be excluded from every trip, and on a second branch we impose $y^v_{ij} = 1$ which ensures that each trip should pass through arc (i, j).

5.3 Valid Inequality

During branch-and-price process, any pair of columns (r, r') added to RMP may violate the trip duration constraint. Therefore, for such a pair a valid inequality will be added to RMP. More precisely, let Γ denote the set of pair of incompatible trips that violate constraint (57):

$$\Gamma = \left\{ (r, r') | u^{v'_r}_{\text{start},d} - u^{v_r}_{\text{start},0} > T \right\}, \tag{83}$$

Algorithm 2 The second heuristic algorithm for generating initial solutions

Let $SLDT$ denote a sorted list of tuple of facilities based on their demands.
Let $SLCT$ denote a sorted list of tuple of vessels and their capacities.
Let $Trips = \emptyset$ denote the set of created trips with their associated attributes.
Set $TotalDemand = \sum_{f \in \mathcal{F}} q_f$.
while $TotalDemand > 0$ **do**
  for v in $SLCT$ **do**
    Define a temporary empty trip $TempTrip$.
    for f in $SLDT$ **do**
      if $duration < 168 - 0.4 \times maxDistance$ **then**
        if $0 < SLDT[f][1] \leq SLCT[v][1]$ **then**
          Calculate the estimated arrival time at the facility f.
          Based on the arrival time of vessel v at facility f, select a node associated with the time winndow.
          calculate the delivery amount $Q_i,\ i \in \mathcal{K}_f$.
          $SLDT[f][1] \longleftarrow SLDT[f][1] - Q_i$.
          $SLCT[v][1] \longleftarrow SLCT[v][1] - Q_i$
          Update $TempTrip$.
        end if
      else if ($duration < 168 - 0.4 \times maxDistance$) and ($TempTrip \neq \emptyset$) **then**
        $Trips \longleftarrow Trips \cup TempTrip$.
        $TempTrip = \emptyset$.
      end if
    end for
  end for
end while
Return the set $Trips$

then the constraint $x_r + x'_r \leq 1$ is a valid inequality. After adding this valid inequality to RMP, we need to modify the pricing problem since the dual variable of constraint (83) will impact the reduced cost. Let ζ_s denote the dual variable corresponding to constraint (83) for $s \in \Gamma$. Define

$$A^*(s) = \left\{ (i,j) \in \mathcal{A} | (i,j) \in r \oplus r' \right\},$$

as the set of arcs in solution s that appears in trips r and r'. Then the value of $\overline{C}_j$ in the label extension algorithm will be updated as $\overline{C}_j = \overline{C}_j - \sum_{s \in \Gamma : (i,j) \in A^*(s)} \zeta_s$.

6 Experimental Results and Discussion

In order to test the performance of our branch-and-price algorithm on periodic supply vessel planning problem, we randomly generated medium to large instances based on the real-world data related to a project that the first and the third authors participated. Text files of these instances are available from the authors upon request. The branch-and-price algorithm was coded into Python 3.8 and the MILP solver of CPLEX was called. All instances were solved using CPLEX 20.1 on a laptop computer with Windows 10 Enterprise 64 bit, a Core i7 processor, and 16 GB RAM.

To generate the instances, we considered different levels for four important factors of the problem, namely, number of facilities $|\mathcal{F}|$, demand, number of time windows per facility TW, and number of visit per facility n_f. It is assumed that the length of time windows is up to 12 hours. Please note that the longer the time window the more possibilities for trips and hence the more complex pricing problem. These factors and their levels are shown in Table 4. Number of nodes and size of the network can be obtained based on the number of facilities and time windows, that is $|\mathcal{N}| = TW|\mathcal{F}|$. It should be noted that we have two levels for n_f. In the first level, all facilities are restricted to require only one visit, and in the second level, the number of visits can be 1 or 2, each with 0.5 probability. At each level of demand, the q_f is generated form $DU[\underline{l}, \overline{u}]$ where $\underline{l}$ and $\overline{u}$ are lower and upper limits of demand intervals, respectively. Considering all these levels, we have $2 \times 3 \times 3 \times 2 = 36$ different classes. For each class we generated 5 different instances. In all experiments, we start with offshore supply vessel (OSV) and platform supply vessel (PSV) as the two types of vessels in the initial solution. The parameters of each vessel type is given in Table 5. Please note that the traveling cost c^v_{ij} is calculated by fuel consumption multiplied by speed multiplied by t^v_{ij}.

In the first experiment, we considered $|\mathcal{F}| = 5$ and $q_f \sim DU[1, 200]$. Depending on the number of TW, the numbers of nodes in the network are 35, 50, and 70. The outputs are shown in Table 6.

Table 4: Levels of different factors for generating instances

$\lvert\mathcal{F}\rvert$	5, 10
q_f	[1, 200], [201, 500], [501, 800]
TW	7, 10, 14
n_f	1, [1, 2]

Table 5: Information for vessels used in the numerical study

	Q^v	δ_0^v	fuel consumption (litre/nautical mile)	speed (nautical mile/hour)
OSV	220	10	40	10
PSV	850	21	54	10

Table 7 and Table 8 show the outputs for $|\mathcal{F}| = 5$ facilities when their demands are $q_f \sim DU[201, 500]$ and $q_f \sim DU[501, 800]$, respectively. In a similar way, in another set of experiments, we considered the same scenarios of facilities $|\mathcal{F}| = 10$. Considering the same values for TW, the resulting networks would have 70, 100, 140 nodes (see Tables 9 - 11). In all output tables, the first column from the left is the instance number. The next three columns are number of time windows, minimum number of visits, and number of nodes, which together with the demand size define the size of the instance. The last seven columns (the outputs) are total number of nodes explored in the branch-and-bound tree, total number of columns generated in the column generation process, total number of cuts, the optimality gap, and computation time. In our experiments we let the algorithm run for 7200 seconds. Otherwise, the algorithm stops as soon as an integer solution is found. In the "multiple visits" ("multiple trips") column, "Y" indicates if there is at least one facility with multiple visits (one vessel with multiple trips), and "N" means no multiple visits (multiple trips) observed. Also reported in tables are the average optimality gap (%) and the average computation time (seconds).

The results from Table 6 indicate that we obtained 0.00 gap in 27 (out of 30) cases with the average run time of less than 2 hours. The table also shows that in some cases we have multiple visit even if the minimum required number of visits is 1. It can also be observed that, based on the optimality gap, the class of instances 16-22 is the most difficult one among all classes in this table. In Table 7 the demands of $|\mathcal{F}|$ facilities are increased. In this case, we obtained 0.00 optimality gap for 24 cases with the average gap of 2.35% and the average computation time of 794 seconds. The results also reveal that increasing n_f from 1 to $[1, 2]$ increases the problem difficulty. Moreover, it can be observed that instances with 10 time windows are easier than those with 7 or 14 time windows. Finally, in this level of demand, the class of instances 26-30 is the most difficult one. The results associated with the third and last level of demand for $|\mathcal{F}| = 5$ are tabulated in Table 8. The structure of this table is the same as the first two tables already discussed. We were able to solve 15 instances to optimality, but the the average gap is 2.30% which is the minimum among all three levels of demand. The average computation time is 5 minutes less which shows the efficiency of the proposed branch-and-price approach. According to this table, the instances with 10 time windows tend to be easier to solve. It is worth noting that the size of instances considered in these three tables are comparable with the medium sized instances considered by Li et al. (2020) and Huang et al. (2021). Overall, comparing Tables 6 - 8 as we increase the minimum facility demand q_f the computation time decreases, and so does the optimality gap.

Now we discuss the second experiment. As mentioned earlier, in this experiment, we assumed that $|\mathcal{F}| = 10$, and there are three levels or intervals from which the demand of facility f, q_f, is randomly selected. The results for the first set of second experiments are given in Table 9 in which $q_f \sim DU[1, 200]$. From all 30 instances solved in this set, we were able to solve 23 of them with 0% optimality gap with an average of 3.45% gap and 5120.3 seconds. Like before, here, the instances with 10 time windows have the lowest gap, which indicates they are easier to solve within 2-hour time limit. The results for the second set in which $q_f \sim DU[201, 501]$ are given in Table 10. Here, only 4 instances have 0% optimality gap, and the average gap of all 30 instances is 26.06% which is quite high. This indicates that these instances are difficult. The last table in our numerical study shows the results for the third set in which $q_f \sim DU[501, 800]$ (see Table 11). The average optimality gap for this set is 3.06% which is obtained in less than 28 minutes. Comparing the impact of TW reveals that the quality of solutions in instances with 10 time windows are better than the instances with seven and 14 time windows, respectively (0.36% versus 2.20% and 4.49%). The size of instances considered in these three tables are comparable with the large sized instances considered by Li et al. (2020) and Huang et al. (2021), and hence they are difficult to solve.

Table 6: Results for instances with $|\mathcal{F}| = 5$ and $q_f \sim DU[1, 200]$

Instance	TW	n_f	$\|\mathcal{N}\|$	nodes explored	no. columns	no. cuts	gap(%)	multiple visits	multiple trips	time (s)
1	7	1	35	61	296	0	0.00	Y	N	7200
2				99	328	0	0.00	Y	N	7200
3				85	468	0	0.00	N	N	7200
4				3	380	0	0.00	N	N	1085
5				133	316	0	0.00	N	N	7200
							0.00			**5977**
6	7	[1,2]	35	13	317	40	0.00	Y	N	2123
7				1	332	2	0.00	Y	N	852
8				1	221	0	0.00	Y	N	369
9				3	276	0	0.00	Y	Y	314
10				5	177	7	0.00	Y	N	3133
							0.00			**1358.2**
11	10	1	50	11	328	0	0.00	N	N	7200
12				3	344	0	0.00	Y	N	2281
13				67	380	0	0.00	Y	N	7200
14				9	295	0	0.00	N	N	1621
15				3	309	0	0.00	Y	N	1733
							0.00			**4007**
16	10	[1,2]	50	3	187	0	0.00	Y	N	1022
17				7	192	0	10.66	N	N	2534
18				3	253	16	48.29	N	N	913
19				3	223	0	0.00	Y	N	1213
20				3	126	31	0.00	Y	N	146
							11.79			**1165.6**
21	14	1	70	5	368	15	0.00	Y	N	5853
22				3	271	0	0.00	Y	N	4683
23				3	305	0	0.00	Y	N	6343
24				19	535	30	0.00	Y	N	7200
25				10	426	0	0.00	N	N	7200
							0.00			**6255.8**
26	14	[1,2]	70	7	380	43	28.60	N	N	6502
27				45	336	0	0.00	N	N	7200
28				1	298	0	0.00	N	N	7200
29				44	264	0	0.00	Y	N	7200
30				3	312	0	0.00	Y	N	4174
							5.72			**6455.2**
							2.92			**4203.13**

7 Conclusion

In this paper, we tackled a new variant of periodic supply vessel planning problem in which preemptive service at the base, variable service at facilities, and split delivery are incorporated. The optimization problem considered in this paper was motivated by oil and gas company Woodside, the largest producer in Australia. We modelled the problem with arc-flow, path-flow and set-partitioning formulations. The optimisation models for this problem are extremely challenging, hence we developed a branch-and-price algorithm to solve its set-partitioning model for instances of realistic dimensions. In this approach, we constructed the initial vessel trips heuristically and added them to the master problem. The solution of linear programming relaxation of the master problem selects vessels and trips to minimise total travel and fixed costs. Then, a modified label extension algorithm capturing problem structure was devised to generate additional vessel trips in the pricing problem.

We considered 180 medium to large-scale example problems. The experimental results reveal that we can find an integer solution for about 73.3% cases of medium-sized instances, i.e., when the number of facilities is 5, with gap zero within 2 hours of computation time. This value is 48.9% for large-sized cases. Several observations can be made from the experimental study. First, increasing the demand from the second level to the third level increases the chance of finding an integer solution with gap of zero. This is because the number of possibilities in generating trips for high demand is less compared to low demand. Second, it is better to include ten time windows in the schedule as there is always a good chance to obtain a low optimality gap in a reasonable amount of time. Third, the middle level of demand increases the problem intractability. The final observation is that the model is able to handle split delivery and multiple trips whenever possible and required.

For future work, the model can be extended to consider other features such as tide constraint and berth capacity constraint at the base. The former type of constraint is important as the tide can affect the arrival (departure) of the vessel at (from) the base.The current model could also be extended for

Table 7: Results for instances with $|\mathcal{F}| = 5$ and $q_f \sim DU[201, 500]$

Instance	TW	n_f	$\|\mathcal{N}\|$	nodes explored	no. columns	no. cuts	gap(%)	multiple visits	multiple trips	time (s)
1	7	1	35	5	156	0	0.00	Y	N	80
2				3	156	0	0.00	N	N	64
3				3	136	0	0.00	Y	N	83
4				3	116	0	0.00	Y	N	70
5				7	151	0	0.00	Y	N	135
							0.00			**86.4**
6	7	[1,2]	35	3	128	2	11.01	Y	N	202
7				7	160	5	0.29	Y	N	397
8				13	164	61	0.00	Y	N	404
9				21	352	22	10.96	Y	N	408
10				1	128	0	0.00	Y	N	41
							4.39			**290.4**
11	10	1	50	23	272	102	0.00	Y	N	705
12				7	196	29	0.00	Y	N	289
13				23	263	15	0.00	Y	N	756
14				3	107	0	0.00	Y	N	140
15				27	348	116	0.00	Y	N	672
							0.00			**512.4**
16	10	[1,2]	50	1	132	0	0.00	Y	N	175
17				1	151	0	0.00	Y	N	245
18				11	124	2	0.00	Y	N	228
19				5	100	0	0.00	Y	Y	58
20				19	219	64	3.60	Y	N	1714
							0.72			**484**
21	14	1	70	25	399	48	0.00	Y	N	2946
22				3	116	0	0.00	Y	N	541
23				7	176	32	0.00	Y	N	490
24				3	144	30	0.00	Y	N	650
25				31	228	45	6.92	Y	N	1737
							1.38			**1272.8**
26	14	[1,2]	70	7	135	1	0.00	Y	N	901
27				1	164	0	0.00	Y	N	1718
28				3	112	0	0.00	Y	N	653
29				1	111	0	0.00	Y	N	129
30				11	227	49	37.97	Y	N	7200
							7.59			**2120.2**
							2.35			**794.4**

the uncertain environments requiring other algorithms and solution methodologies. Finally, models and algorithms for the problem with additional features such as shared vessels could be developed.

Acknowledgements

The second author is partially supported by the Australian Research Council through the Centre for Transforming Maintenance through Data Science (grant number IC180100030).

References

B. Afshar-Nadjafi and A. Afshar-Nadjafi. A constructive heuristic for time-dependent multi-depot vehicle routing problem with time-windows and heterogeneous fleet. *Journal of King Saud University-Engineering Sciences*, 29(1):29–34, 2017.

A. Agra, M. Christiansen, and A. Delgado. Mixed integer formulations for a short sea fuel oil distribution problem. *Transportation Science*, 47(1):108–124, 2013.

H. Andersson, M. Christiansen, and K. Fagerholt. The maritime pickup and delivery problem with time windows and split loads. *INFOR: Information Systems and Operational Research*, 49(2):79–91, 2011.

C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. Savelsbergh, and P. H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998.

T. Borthen, H. Loennechen, X. Wang, K. Fagerholt, and T. Vidal. A genetic search-based heuristic for a fleet size and periodic routing problem with application to offshore supply planning. *EURO Journal on Transportation and Logistics*, 7(2):121–150, 2018.

Table 8: Results for instances with $|\mathcal{F}| = 5$ and $q_f \sim DU[501, 800]$

Instance	TW	n_f	$\|\mathcal{N}\|$	nodes explored	no. columns	no. cuts	gap(%)	multiple visits	multiple trips	time (s)
1	7	1	35	33	169	111	1.23	Y	N	389
2				21	154	20	7.33	Y	N	214
3				30	197	101	9.46	Y	N	427
4				9	62	65	0.13	Y	N	58
5				13	74	5	4.67	Y	N	72
							4.56			**232**
6	7	[1,2]	35	9	65	0	1.20	N	N	92
7				1	10	0	0.00	Y	N	5
8				3	54	0	6.60	N	N	41
9				3	78	20	0.00	Y	N	47
10				1	66	19	0.00	Y	N	26
							1.56			**42.2**
11	10	1	50	1	10	0	0.00	Y	N	5
12				1	10	0	0.00	Y	N	3
13				11	62	23	0.00	Y	N	100
14				1	10	0	0.00	Y	N	5
15				15	166	38	0.60	Y	N	343
							0.12			**91.2**
16	10	[1,2]	50	1	10	0	0.00	Y	N	5
17				1	10	0	0.00	Y	N	5
18				21	134	3	2.30	Y	N	543
19				1	42	0	0.00	Y	N	27
20				35	242	212	4.40	Y	N	724
							1.34			**260.8**
21	14	1	70	1	10	0	0.00	Y	N	5
22				1	10	0	0.00	Y	N	5
23				1	10	0	0.00	Y	N	5
24				17	62	2	6.90	Y	N	180
25				19	105	37	8.80	Y	N	949
							3.14			**228.8**
26	14	[1,2]	70	1	46	0	0.00	Y	N	139
27				1	10	0	0.00	Y	N	5
28				21	134	3	2.32	Y	N	360
29				17	98	49	5.53	Y	N	335
30				65	330	200	7.65	Y	N	3178
							3.10			**803.4**
							2.30			**276.4**

T. Borthen, H. Loennechen, K. Fagerholt, X. Wang, and T. Vidal. Bi-objective offshore supply vessel planning with costs and persistence objectives. *Computers & Operations Research*, 111:285–296, 2019.

K. Braekers, K. Ramaekers, and I. Van Nieuwenhuyse. The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering*, 99:300–313, 2016.

A. M. Campbell and J. H. Wilson. Forty years of periodic vehicle routing. *Networks*, 63(1):2–15, 2014.

M. Christiansen, K. Fagerholt, N. P. Rachaniotis, and M. Stålhane. Operational planning of routes and schedules for a fleet of fuel supply vessels. *Transportation Research Part E: Logistics and Transportation Review*, 105:163–175, 2017.

L. Costa, C. Contardo, and G. Desaulniers. Exact branch-price-and-cut algorithms for vehicle routing. *Transportation Science*, 53(4):946–985, 2019.

R. Cruz, A. B. Mendes, L. Bahiense, and Y. Wu. Integrating berth allocation decisions in a fleet composition and periodic routing problem of platform supply vessels. *European Journal of Operational Research*, 275(1):334–346, 2019.

G. B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operations Research*, 8(1): 101–111, 1960.

M. Dell'Amico, G. Righini, and M. Salani. A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection. *Transportation Science*, 40(2):235–247, 2006.

G. Desaulniers. Branch-and-price-and-cut for the split-delivery vehicle routing problem with time windows. *Operations Research*, 58(1):179–192, 2010.

G. Desaulniers, J. Desrosiers, and M. M. Solomon. *Column Generation*, volume 5. Springer Science & Business Media, 2006.

Table 9: Results for instances with

Instance	TW	n_f	$\|\mathcal{N}\|$	nodes explored	no. columns	no. cuts	gap(%)	multiple visits	multiple trips	time (s)
1	7	1	70	17	381	2	0.00	Y	N	1215
2				49	690	257	8.83	Y	N	5561
3				51	816	509	0.00	Y	N	5431
4				3	408	0	0.00	Y	N	744
5				13	524	31	0.00	Y	N	1873
							1.77			**2964.8**
6	7	[1,2]	70	5	392	34	0.00	Y	N	1878
7				21	704	94	6.10	Y	N	6537
8				1	276	23	0.00	Y	N	1072
9				21	620	71	13.41	Y	N	5877
10				10	372	82	2.42	Y	N	7200
							4.39			**4512.8**
11	10	1	100	2	260	0	0.00	Y	N	1398
12				19	556	79	0.00	Y	N	6842
13				7	344	0	0.00	Y	Y	1696
14				5	244	14	0.00	Y	N	1577
15				15	720	1	0.00	Y	N	4698
							0.00			**3242.2**
16	10	[1,2]	100	1	440	3	0.00	Y	N	7200
17				8	504	32	0.00	Y	N	7200
18				3	596	6	0.00	Y	N	7200
19				11	608	20	0.00	Y	N	6832
20				13	716	0	0.00	Y	N	6506
							0.00			**6987.6**
21	14	1	140	1	420	0	0.00	Y	N	7200
22				1	328	0	0.00	Y	N	7200
23				29	122	16	5.50	Y	N	272
24				1	364	0	0.00	Y	N	7200
25				1	412	0	0.00	Y	N	7200
							1.10			**5814.4**
26	14	[1,2]	140	1	372	0	0.00	Y	N	7200
27				1	332	0	0.00	Y	N	7200
28				1	420	1	62.30	Y	N	7200
29				1	360	22	4.80	Y	N	7200
30				1	356	0	0.00	Y	N	7200
							13.42			**7200**
							3.45			**5120.3**

M. Dror. Note on the complexity of the shortest path models for column generation in vrptw. *Operations Research*, 42(5):977–978, 1994.

K. Fagerholt and H. Lindstad. Optimal policies for maintaining a supply service in the norwegian sea. *Omega*, 28(3):269–275, 2000.

D. Feillet. A tutorial on column generation and branch-and-price for vehicle routing problems. *4OR*, 8(4):407–424, 2010.

D. Feillet, P. Dejax, M. Gendreau, and C. Gueguen. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks: An International Journal*, 44(3):216–229, 2004.

I. Gribkovskaia, G. Laporte, and A. Shlopak. A tabu search heuristic for a routing problem arising in servicing of offshore oil and gas platforms. *Journal of the Operational Research Society*, 59(11): 1449–1459, 2008.

E. E. Halvorsen-Weare, K. Fagerholt, L. M. Nonås, and B. E. Asbjørnslett. Optimal fleet composition and periodic routing of offshore supply vessels. *European Journal of Operational Research*, 223(2): 508–517, 2012.

F. Hernandez, D. Feillet, R. Giroudeau, and O. Naud. A new exact algorithm to solve the multi-trip vehicle routing problem with time windows and limited duration. *4OR*, 12(3):235–259, 2014.

F. Hernandez, D. Feillet, R. Giroudeau, and O. Naud. Branch-and-price algorithms for the solution of the multi-trip vehicle routing problem with time windows. *European Journal of Operational Research*, 249(2):551–559, 2016.

G. Homsi, R. Martinelli, T. Vidal, and K. Fagerholt. Industrial and tramp ship routing problems: Closing the gap for real-scale instances. *European Journal of Operational Research*, 283(3):972–990, 2020.

Table 10: Results for instances with $|\mathcal{F}| = 10$ and $q_f \sim DU[201, 500]$

Instance	TW	n_f	$\|\mathcal{N}\|$	nodes explored	no. columns	no. cuts	gap(%)	multiple visits	multiple trips	time (s)
1	7	1	70	24	630	148	59.50	Y	N	6758
2				38	648	227	56.40	Y	N	6336
3				28	619	95	33.10	Y	N	3681
4				12	408	167	24.80	Y	N	2418
5				8	384	72	66.80	Y	N	1237
							48.12			**4086**
6	7	[1,2]	70	6	308	9	11.9	Y	N	1414
7				4	143	9	0.00	Y	N	3207
8				1	284	14	13.80	Y	N	2126
9				5	188	0	0.00	Y	Y	2445
10				17	435	37	0.00	Y	N	1612
							5.14			**2160.8**
11	10	1	100	8	448	160	52.60	Y	N	3640
12				36	600	384	27.80	Y	N	7200
13				40	516	339	27.80	Y	N	7200
14				54	688	390	50.80	Y	N	7200
15				33	568	249	21.50	Y	N	4803
							36.10			**6008.3**
16	10	[1,2]	100	5	84	12	16.50	Y	N	1201
17				29	628	260	8.90	Y	N	7200
18				13	436	59	0.00	Y	N	3275
19				3	299	33	19.50	Y	N	1553
20				20	348	26	3.50	Y	N	7200
							9.68			**4085.8**
21	14	1	140	1	180	0	52.60	Y	N	7200
22				34	452	383	28.60	Y	N	5500
23				1	160	0	36.60	Y	N	1296
24				42	544	419	33.10	Y	N	7200
25				18	340	112	45.30	Y	N	4382
							39.24			**5115.6**
26	14	[1,2]	140	12	396	142	19.40	Y	N	6131
27				1	248	43	7.90	Y	N	4964
28				10	456	84	25.80	Y	N	6419
29				8	248	49	35.60	Y	N	6299
30				5	236	45	1.80	Y	N	2587
							18.10			**5278.20**
							26.06			**4455.78**

N. Huang, J. Li, W. Zhu, and H. Qin. The multi-trip vehicle routing problem with time windows and unloading queue at depot. *Transportation Research Part E: Logistics and Transportation Review*, 152:102370, 2021.

Y. Kisialiou, I. Gribkovskaia, and G. Laporte. The periodic supply vessel planning problem with flexible departure times and coupled vessels. *Computers & Operations Research*, 94:52–64, 2018.

Y. Kisialiou, I. Gribkovskaia, and G. Laporte. Supply vessel routing and scheduling under uncertain demand. *Transportation Research Part C: Emerging Technologies*, 104:305–316, 2019.

Ç. Koç, T. Bektaş, O. Jabali, and G. Laporte. Thirty years of heterogeneous vehicle routing. *European Journal of Operational Research*, 249(1):1–21, 2016.

J. Li, H. Qin, R. Baldacci, and W. Zhu. Branch-and-price-and-cut for the synchronized vehicle routing problem with split delivery, proportional service time and multiple time windows. *Transportation Research Part E: Logistics and Transportation Review*, 140:101955, 2020.

E. Mardaneh, Q. Lin, R. Loxton, and N. Wilson. Cargo scheduling decision support for offshore oil and gas production: a case study. *Optimization and Engineering*, 18(4):991–1008, 2017.

M. E. McNabb, J. D. Weir, R. R. Hill, and S. N. Hall. Testing local search move operators on the vehicle routing problem with split deliveries and time windows. *Computers & Operations Research*, 56:93–109, 2015.

A. Shyshou, I. Gribkovskaia, G. Laporte, and K. Fagerholt. A large neighbourhood search heuristic for a periodic supply vessel planning problem arising in offshore oil and gas operations. *INFOR: Information Systems and Operational Research*, 50(4):195–204, 2012.

M. Stålhane, H. Andersson, M. Christiansen, J.-F. Cordeau, and G. Desaulniers. A branch-price-and-cut method for a ship routing and scheduling problem with split loads. *Computers & Operations Research*, 39(12):3361–3375, 2012.

Table 11: Results for instances with $|\mathcal{F}| = 10$ and $q_f \sim DU[501, 800]$

Instance	TW	n_f	$\lvert\mathcal{N}\rvert$	nodes explored	no. columns	no. cuts	gap(%)	multiple visits	multiple trips	time (s)
1	7	1	70	9	140	26	14.60	Y	N	370
2				9	144	119	0.47	Y	N	335
3				19	252	124	0.01	Y	N	870
4				5	164	66	0.00	Y	N	317
5				23	224	284	13.25	Y	N	852
							5.67			**548.8**
6	7	[1,2]	70	3	100	37	0.00	Y	N	153
7				7	140	0	0.00	Y	Y	323
8				3	76	0	6.32	Y	N	83
9				3	92	0	0.00	Y	Y	356
10				11	168	50	0.00	Y	Y	782
							1.26			**339.4**
11	10	1	100	15	200	130	3.81	Y	N	7200
12				17	120	45	0.00	Y	N	583
13				35	264	95	8.40	Y	N	1774
14				15	160	113	9.10	Y	N	1027
15				27	264	337	0.00	Y	N	1845
							4.26			**2485.8**
16	10	[1,2]	100	1	124	0	0.00	Y	N	163
17				1	40	0	0.00	Y	N	56
18				5	132	45	1.71	Y	N	298
19				1	40	0	0.00	Y	N	36
20				7	260	102	0.00	Y	N	1295
							0.34			**369.6**
21	14	1	140	25	276	327	0.40	Y	N	3073
22				13	296	89	12.50	Y	N	3371
23				13	296	89	12.90	Y	N	3468
24				33	292	275	0.00	Y	N	5371
25				33	324	493	0.00	Y	N	5409
							5.16			**4138.4**
26	14	[1,2]	140	1	164	14	0.00	Y	N	830
27				5	128	17	8.30	Y	N	2247
28				1	68	0	0.00	Y	N	565
29				9	136	4	0.00	Y	N	7200
30				1	68	0	0.00	Y	N	310
							1.66			**2230.4**
							3.06			**1685.4**

T. Vidal, T. G. Crainic, M. Gendreau, N. Lahrichi, and W. Rei. A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. *Operations Research*, 60(3):611–624, 2012.

E. Zare-Reisabadi and S. H. Mirmohammadi. Site dependent vehicle routing problem with soft time window: Modeling and solution approach. *Computers & Industrial Engineering*, 90:177–185, 2015.