European Journal of Operational Research 303 (2022) 891–907

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Innovative Applications of O.R.

A time-dependent vessel routing problem with speed optimization

Karl Petter Ulsrud[a], Anders Helgeland Vandvik[a], Andreas Breivik Ormevik[a,*], Kjetil Fagerholt[a], Frank Meisel[b]

[a] Department of Industrial Economics and Technology Management, Norwegian University of Science and Technology, Trondheim NO-7491, Norway
[b] Institute for Business Management, Kiel University, Kiel 24098, Germany

ARTICLE INFO

Article history:
Received 22 November 2021
Accepted 8 March 2022
Available online 12 March 2022

Keywords:
Transportation
Time-dependent vessel routing
Speed optimization
Pickup and delivery
Adaptive large neighborhood search heuristic

ABSTRACT

We study an operational planning problem arising in the offshore oil and gas industry, in which we determine routes, as well as sailing speeds along these routes, for a set of platform supply vessels (PSVs) servicing a given set of delivery and pickup orders such that costs are minimized. The sailing costs, mainly induced by fuel consumption for the PSVs, heavily depend on the chosen sailing speeds. Furthermore, the fuel consumption and the feasible speed ranges for the PSVs are largely affected by weather conditions that may vary over time, resulting in a weather- or *Time-Dependent Vessel Routing Problem with Speed Optimization* (TDVRP-SO). Optional decisions include the postponement of certain orders and the chartering of spot vessels, both associated with additional costs. We present a time-discrete mixed integer programming (MIP) model for the TDVRP-SO. To overcome the challenges of solving large-scale instances of the TDVRP-SO with a commercial MIP solver, we propose an Adaptive Large Neighborhood Search (ALNS) heuristic extended with a local search and a set partitioning model. The ALNS heuristic also includes solving the sub-problem of determining the optimal sailing speeds along each PSV route. Computational tests on instances based on a real planning case from the Norwegian continental shelf show that the ALNS heuristic efficiently provides high-quality solutions. It is also demonstrated that, in contrast to current planning practice, accounting for speed optimization and weather conditions significantly improves the solutions.

1. Introduction

Since the first successful exploration drilling in the southern parts of the North Sea late in the 1960s, the Norwegian petroleum industry has grown and developed into one of the major drivers of Norway's technological and economic growth. Norway's production of oil and gas takes place with platforms at offshore locations on the Norwegian continental shelf, which covers the North Sea, the Norwegian Sea and the Barents Sea. As of 2021, 90 manned platforms or offshore installations (both fixed and floating units) are in operation and contribute to the largest share of the production volumes from the industry (Norwegian Petroleum Directorate, 2021). Each offshore installation can have several hundreds of employees, which naturally leads to large demands for consumable cargoes such as food and beverages and the return of produced waste. Further demand for cargoes is related to maintenance of the installation and the extraction and processing of oil and gas. These cargoes vary greatly in size and form, from small machine parts to large volumes of specialized chemicals.

Supplies to and from the offshore installations are distributed from onshore supply bases along the Norwegian coast by fleets of *Platform Supply Vessels* (PSVs). The PSVs can be extremely costly in operation, with daily charter costs as high as USD 25 000 in some cases (Kisialiou, Gribkovskaia, & Laporte, 2018b). As a result, measures for reducing the operational costs from the offshore logistics is of great importance for the operators in the Norwegian oil and gas industry. At the same time, the environmental performance of operating the PSV fleet has become another field of strong interest to reach the industry's goals of reducing greenhouse gas emissions. Lindstad, Eskeland, & Rialland (2017) report that a single PSV can emit approximately 10 000 tons of CO_2 equivalents per year, depending on the operational profile for the vessel. Detailed operational decision making, e.g., routing and selection of sailing speeds, can play an important role in reducing both costs and emissions from PSV operations.

In this context, we consider a real-life operational planning problem faced on a daily basis by logistics planners involved in the distribution of supplies to and from offshore installations on

* Corresponding author.
E-mail address: andreas.ormevik@ntnu.no (A.B. Ormevik).

https://doi.org/10.1016/j.ejor.2022.03.015

the Norwegian continental shelf. The main task is to determine the routes and schedules for a fleet of PSVs such that total costs are minimized. Each route should start and end at a given onshore supply base (depot), servicing a number of delivery and pickup orders at the platforms in between. In some cases, it might not be possible to service all orders during the planning horizon using only the PSVs in the contracted fleet. The planners then have the option to either postpone the orders or to charter additional PSVs from the spot market on a short-term basis. The total cost to be minimized consists of the sailing costs for the PSVs, in addition to the costs for chartering additional PSVs and the penalty costs for postponing orders. The sailing costs, which mainly amounts to fuel costs for the PSVs, heavily depend on the chosen sailing speeds. Furthermore, the fuel consumption and the feasible speed ranges for the PSVs are largely affected by weather conditions that may vary over time. The time spent on servicing orders at the platforms is also affected by the weather conditions, with longer service times in rough weather conditions. In other words, both service times as well as fuel consumption functions and speed ranges for the PSVs are time-dependent parameters.

We consider the planning problem taking all these practical issues into account in order to determine the following: 1) routes for the PSVs, 2) sailing speeds along these routes, 3) which orders to postpone, and 4) whether to charter in additional PSVs from the spot market, and if so, the routes and sailing speeds also for these. For a time span of 2–3 days, corresponding to the typical duration of the PSV routes and hence the planning horizon, the weather forecast for the areas of operation is usually quite accurate and can be handled as deterministic input to the planning. We name this problem the *Time-Dependent Vessel Routing Problem with Speed Optimization (TDVRP-SO)*.

Problems similar to the TDVRP-SO have received increasing attention in the operations research community in recent years. Early articles focused mostly on the strategic or tactical planning level of offshore logistics, aiming to determine the optimal size for a PSV fleet together with a set of weekly routes and schedules for selected routes between platforms. This problem was first addressed by Fagerholt & Lindstad (2000) under the term Supply Vessel Planning Problem (SVPP). Different extended versions of the SVPP were later studied and solved with various solution methods, e.g., by Halvorsen-Weare, Fagerholt, Nonås, & Asbjørnslett (2012), Norlund, Gribkovskaia, & Laporte (2015), Kisialiou et al. (2018b), Kisialiou, Gribkovskaia, & Laporte (2018a) and Borthen, Loennechen, Fagerholt, Wang, & Vidal (2019).

At a more operational planning level, offshore logistics planning must consider both the delivery of cargo to offshore installations and the pickup of "return cargo" to the onshore base. This makes the TDVRP-SO related to the well-known group of *vehicle routing problems with pickups and deliveries* (VRPPDs). Berbeglia, Cordeau, Gribkovskaia, & Laporte (2007) introduce a threefold classification of such problems based on the servicing pattern present in distribution, namely one-to-one, many-to-many, and one-to-many-to-one problems. The TDVRP-SO is a typical example of the latter as PSVs start their routes from a single onshore supply base and finally return to this base. Similar pickup-and-delivery problems have been studied within offshore logistics by Aas, Gribkovskaia, Halskau, & Shlopak (2007), Gribkovskaia, Laporte, & Shlopak (2008) and Sopot & Gribkovskaia (2014). These articles all consider the problem for a single PSV, whereas the TDVRP-SO creates schedules for a fleet of PSVs.

If available PSV capacity is scarce, situations might occur where only a selection of the requested orders can be serviced on the scheduled voyages. This problem variant was investigated by Cuesta, Andersson, Fagerholt, & Laporte (2017) as the *Vessel Routing Problem with Selective Pickups and Deliveries* (VRPSPD). The formulated arc-flow model was used to determine the most cost-efficient voyage for a single vessel where a penalty cost is charged for those orders that are postponed and hence not serviced on the scheduled voyages. An *Adaptive Large Neighborhood Search* (ALNS) heuristic was proposed for providing high-quality solutions to large problem instances. Order selection is well-studied in many maritime applications, often through another approach of considering the profit-collecting VRPs (PCVRPs), of which Archetti, Speranza, & Vigo (2014) provide a classification of different variants and associated models and solution approaches.

A further issue in offshore logistics are harsh weather conditions that greatly affect the operations. Halvorsen-Weare & Fagerholt (2011) identify wave height as the most influential weather parameter and provide values for forced speed reduction and increased service time of PSVs for different weather states. In an optimization-simulation framework, they add slack to voyages and schedules to obtain solutions that are robust with regards to limit the amount of unserviced demand. A related study is performed by Kisialiou et al. (2018b), investigating the trade-off between minimizing costs and obtaining a certain service level to the installations. Weather conditions can in some cases also force the PSVs to return to the supply base, hence failing to service all requested orders. This can also happen in cases where sudden and more urgent orders occurs, mainly due to equipment failure at an installation that requires a quick response. The problem of managing such disruptions (i.e., recovering to the initial weekly schedules by rerouting the PSVs in the fleet), was first addressed by Albjerk, Danielsen, Krey, Stålhane, & Fagerholt (2016) proposing an exact model formulation, and later extended by Stålhane, Albjerk, Danielsen, Krey, & Fagerholt (2019) with a variable neighborhood heuristic for solving larger case instances.

As weather conditions offshore usually will vary over a planning horizon, the routing problem for PSVs becomes a highly time-dependent (TD) planning problem. Weather conditions impact servicing times at installations as well as sailing times due to forced speed reductions, which in turn affect fuel consumption and, hence, emissions and costs. Time-dependent variants of vehicle routing problems have been investigated for long, with Malandraki & Daskin (1992) as one prominent pioneering study. They provide a definition of the time-dependent vehicle routing problem (TDVRP) and discuss various formulations and solution heuristics for it. Several solution methods have been proposed for TDVRPs in recent years, e.g., branch-and-price (Dabia, Ropke, Van Woensel, & De Kok (2013)), ALNS (Franceschetti et al. (2017)) and hybrid ant colony and local search algorithms (Ma, Wu, & Dai (2017)). For many maritime applications, time-dependency have been handled by linking spatial and temporal positions of PSVs in a time-space node network formulation. Christiansen, Fagerholt, Rachaniotis, & Stålhane (2017) propose such solution approach for the operations planning of bunkering vessels. In a problem arising in the aquaculture industry described by Lianes, Noreng, Fagerholt, Slette, & Meisel (2021), where designated service vessels performs several tasks at sea-based fish farms, both sailing and service times are time-dependent due to varying weather conditions.

Scheduling the PSVs in the TDVRP-SO involves determining the speed for all sailing legs during a voyage. Norlund & Gribkovskaia (2013) consider speed optimization for supply vessels in the offshore logistics through a set of speed selection strategies used for the construction of periodic vessel schedules. In the arc-flow model proposed by Andersson, Fagerholt, & Hobbesland (2015) for a case in roll-on roll-off (RORO) shipping, speed decisions are found as a linear combination of speed alternatives. Modeling voyages in a time-space network allows for handling the usually nonlinear speed optimization problem implicitly either in a mathematical model or a heuristic approach. Such features have been explored previously for instance by Fagerholt, Laporte, & Norstad (2010) and Norstad, Fagerholt, & Laporte (2011) when including

speed optimization in the planning of various applications within deep sea shipping.

The literature review illustrates the great complexity of offshore logistics where many real-life aspects need to be considered to plan PSV operations in a reasonable way. Both order selection and speed optimization must be performed while considering the time-dependency of the decisions due to various weather conditions. The TDVRP-SO that we present in the following, takes these features comprehensibly into account when finding schedules for a fleet of PSVs. The main contributions of this article are listed below:

1. A time-discrete mixed integer programming model for the *Time-Dependent Vessel Routing Problem with Speed Optimization (TDVRP-SO)* that involves relevant features such as speed optimization, accounting for weather conditions, and the option of postponing a selection of orders.
2. The development of a specialized *Adaptive Large Neighborhood Search (ALNS)* heuristic, where both problem-specific destroy and repair heuristics and local search operators are introduced to improve the performance. Furthermore, the ALNS heuristic is integrated with a set partitioning model, recombining promising candidate solutions generated by the heuristic in order to find better solutions in shorter solution time. The ALNS heuristic also includes solving the sub-problem of determining the optimal sailing speeds along each PSV route.
3. We perform various computational experiments based on realistic case data inspired by the current offshore logistics operations on the Norwegian continental shelf. Through this, we show that the proposed ALNS heuristic and its extensions efficiently provides high-quality solutions. It is also demonstrated that, in contrast to current planning practice, accounting for speed optimization and weather conditions significantly improves the solutions.

The further paper is organized as follows: Section 2 provides a detailed description of the TDVRP-SO, illustrated by an example highlighting key features of the problem. Section 3 presents the mathematical formulation of the problem, while the ALNS heuristic developed to solve larger test instances is detailed in Section 4. The test instances generated to represent a real application of the TDVRP-SO are presented in Section 5, along with the required problem specific input data used for computational studies. The further parts of this section summarize and discuss the obtained results. Section 6 concludes the paper, and details on the parameter tuning process needed to run the ALNS heuristic efficiently can be found in Appendix A.

2. Problem description

Solving the TDVRP-SO involves determining routes and schedules for a fleet of platform supply vessels (PSVs) that must service a given set of offshore installations from an onshore supply base (depot). Each offshore installation has an individual demand for pickup and delivery of various cargo, usually stored in deck containers or as bulk cargo (liquids) in designated tanks below deck. An *order* refers to the aggregated quantity of cargoes with similar characteristics requested by each offshore installation. We distinguish between mandatory orders that must be serviced by one of currently planned voyages and optional orders that can be postponed to a later voyage, which, however is associated with a penalty cost. It is assumed that all return cargo (the pickup orders) is optional as it can be stored temporarily at the installations for a short period of time. This leads to three types of orders: *Mandatory delivery (MD), optional delivery (OD)* and *optional pickup (OP)* orders. Since some of the offshore installations do not operate their cranes during nights, these have *restricted opening hours* for when service of the orders (i.e., loading or unloading) can be performed.

The problem is solved on a daily basis, where a plan is made for all PSVs being available for departure from the depot at the next day. The planning period is adapted to the duration of a typical PSV route, which should be at most three days (72 hours). A route or a *voyage* is here defined as a sequence of sailing legs starting at the supply depot, visiting a subset of the offshore installations in the area, and returning to the supply depot before the end of the planning horizon. There is no upper limit on the number of installations visited on a voyage, but each installation can only be visited once by one of the scheduled PSV voyages. A PSV performs the following *activities* on a voyage: (1) Voyage preparations at the supply depot before departure, which takes place during fixed opening hours at the depot and includes tasks such as unloading return cargo (i.e., pickup cargoes) from the previous voyage, bunkering, and loading of the delivery cargoes, (2) sailing between installations and to and from the supply depot, (3) idling at the installations waiting for the commencement of service (only when needed due to rough weather conditions or restricted opening hours at the installations), and (4) handling operations for servicing the orders at an installation (i.e., loading and unloading).

The set of PSVs consists of both a *contracted* fleet having the depot as home port (the "fleet vessels"), and a limited number of available *charter* vessels from a spot market (the "spot vessels"). The latter can only be chartered in cases where the contracted vessels are not able to meet all *mandatory* demand. The PSV fleet is heterogeneous, with individual sailing speeds (both design speed and maximum speed) and machinery layout. Hence, fuel consumption rates and costs of operation also differ between the vessels. Usually, a convex non-linear relationship exists between fuel consumption and sailing speed for vessels, and a speed interval can thus be defined such that lower speed always yields lower fuel consumption within this interval. The PSVs may also have different cargo capacities for different deck and bulk cargoes. In this paper, we consider the deck area to be the limiting resource with regards to vessel capacity. In other words, we assume that bulk capacity constraints will not be violated and, hence, we can express all order sizes in terms of deck cargo units (containers) when generating the test instances later in Section 5.1.

We use a forecast of significant wave heights in the operations area to take weather conditions into account in the planning. High waves reduce the speed of PSVs sailing with the same machinery power, and will also increase the servicing time at installations. This increases the fuel consumption as illustrated in Fig. 1, where some forced speed reduction Δv induced by the weather horizontally shifts the fuel consumption curve. This way, fuel consumption in a certain weather state is found by "translating" the fuel consumption curve known for the vessel operating in calm weather. Additionally, due to safety regulations, wave heights surpassing a certain limit can prohibit service at an installation, forcing the PSVs to idle nearby. As weather conditions can be different for different periods of the planning horizon, speed decisions and, thus, operating costs for performing a PSV voyage depend on the time a leg is sailed.

The objective of the TDVRP-SO is to minimize the sum of monetary costs and penalty costs associated with performing the planned PSV voyages. The monetary costs consist of operating costs for all PSVs, including both variable sailing costs for all vessels used and fixed chartering costs (at a daily rate) for any spot vessels used, and penalty costs are induced by the postponement of optional delivery and pickup orders. The variable sailing costs are assumed to be proportional to the total energy consumption for the PSVs performing the scheduled voyages. This includes mainly the fuel consumption, but can also take into account the use of bat-

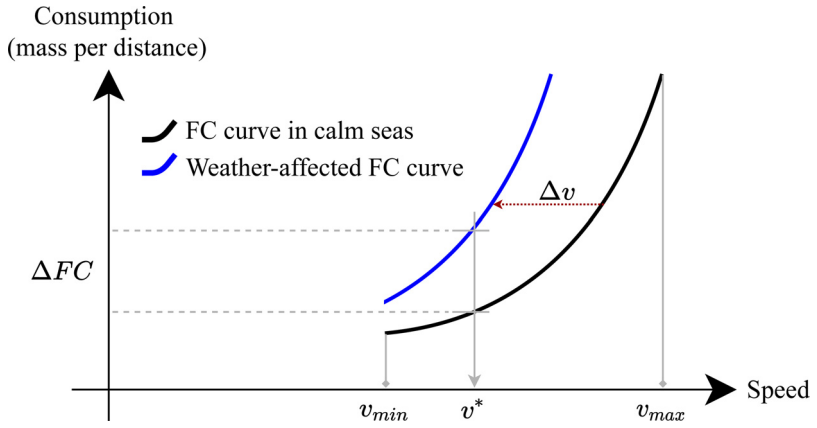

Fig. 1. Illustration of time-dependent fuel consumption (FC). Here, a fixed speed offset Δv induced by weather conditions is assumed for all sailing speeds, shifting the fuel consumption curve horizontally such that maintaining the desired speed v^* becomes considerably more fuel consuming. $v_{\min}$ denotes the starting point of the speed interval in which the fuel consumption is a convex function of sailing speed.

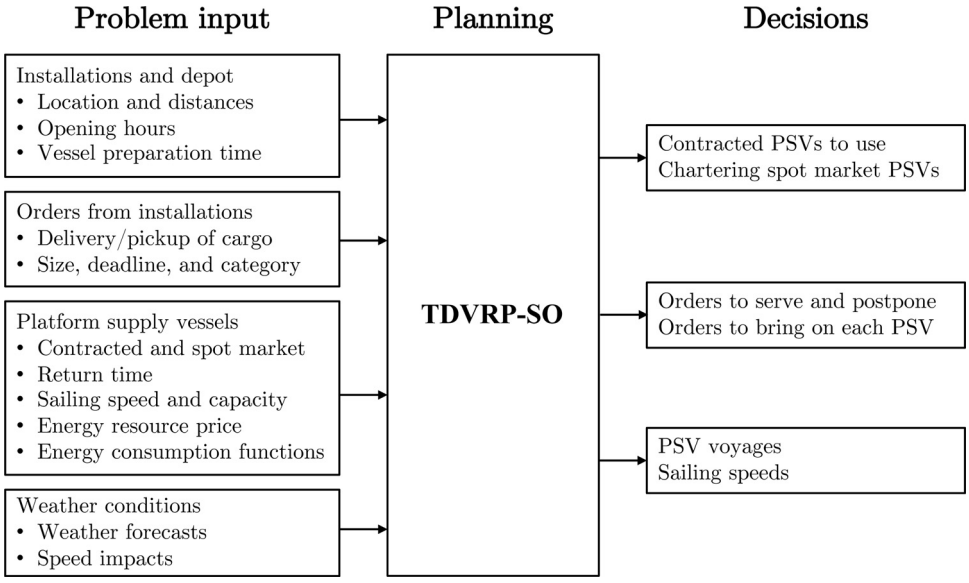

Fig. 2. Workflow of solving the TDVRP-SO.

teries in diesel-electric hybrid machinery configurations becoming more and more frequent for PSVs.

The TDVRP-SO determines cost-optimal voyages (routes) for the PSVs being used, including a complete schedule indicating sailing speeds and required idling or servicing time on all sailing legs. Which orders to deliver and pickup, and which to postpone, is also decided along with the decision of whether or not to charter spot vessels for fulfilling all mandatory delivery orders. Fig. 2 summarizes the solution process for the TDVRP-SO, listing the required input and obtained output and how these can be categorized based on their characteristics.

For a better understanding of the main features of the TDVRP-SO, a small-size example problem and its optimal solution are presented next. In this example, a supply depot services a total of 13 offshore installations. For the next planning period, scheduling the voyages departing on the following day, five installations have placed orders for pickup and delivery of cargo, see Table 1. Installation 1 has limited opening hours, meaning that the PSVs can only service the installation during a "day shift" (hour 10 to hour 22 of the day) whereas all other installations are open 24 hours a day. Two contracted vessels, named *PSV1* and *PSV2*, are available at the depot, with a cargo capacity of 110 and 85 cargo units, respectively. For both vessels, an allowable speed range from 10 knots to 14 knots (maximum sailing speed in fair weather) is specified. A weather forecast is provided, predicting the significant wave heights in the area for the 72 hour planning horizon. We assume the forecast to be *deterministic and static*, meaning that it will not be updated as the PSV voyages last and that the same weather prediction is made for the entire operational area. Neither our definition of the TDVRP-SO nor the solution approaches presented in this paper is restricted by such an assumption, but it is made to simplify the example.

Table 1
Orders placed by the five installations in the example problem. Each order is either a mandatory delivery (MD), an optional delivery (OD) or an optional pickup (OP). Order sizes are given in deck cargo units (containers).

Installation	Opening hours	Order types	Order size
1	10–22	MD	15
		OD	7
2	00–24	MD	25
		OP	30
3	00–24	MD	7
		OD	9
4	00–24	MD	18
		OP	12
5	00–24	MD	20
		OD	7

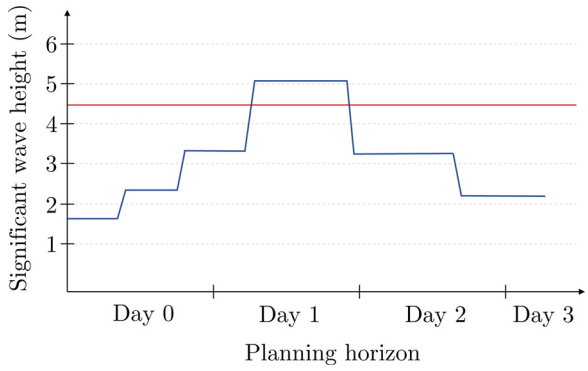

Fig. 3. Weather forecast in the example problem. The horizontal line marks the wave height limit for safe cargo handling at the installations.

The wave height forecast is illustrated in Fig. 3, where the horizontal lines indicate the maximum allowed wave height (4.5m) beyond which the orders cannot be handled at the offshore installations for safety reasons.

A potential solution to the problem is shown in Fig. 4. All orders, both mandatory and optional pickup and delivery cargoes, are serviced. The figure indicates the corresponding voyages of the two vessels and shows the chosen sailing speed for each leg. Note that *PSV1* sails at maximum speed in order to reach installation 1 before it closes at hour 22. Furthermore, the sailing speed is kept relatively high until the last installation (i.e., number 4) has been serviced. This happens to complete all cargo handling activities before the weather conditions worsen at day 2 such that servicing is prohibited, see Fig. 3. On the final sailing leg that leads back to the supply depot, speed is reduced to a cost-efficient minimum as there is no risk of violating the allowed voyage duration of three days.

3. Modeling the TDVRP-SO

The mathematical model for the TDVRP-SO is formulated as a mixed integer programming (MIP) model on a time-space network. We next describe the procedure for generating this network, followed by the presentation of the MIP model.

3.1. Generation of the time-space network

The TDVRP-SO is formulated on a time-space network of nodes and arcs, where a node is characterized by a pair (i, t) with i being an order and t being a completion time of that order. From this, if node (i, t) is part of a PSV's route, it means that the PSV services order i and completes it at time t. Additionally, the model has origin nodes (o, t) and destination nodes (d, t), where o and d both refer to the depot in order to enable that PSVs start and end their routes at the supply base. Two nodes (i, t) and (j, t') can be connected by an arc $\big((i, t), (j, t')\big)$, meaning that a PSV completes order i at time t (and departs the associated installation immediately) and then completes its next scheduled order j at time t'. Such an arc is feasible and included in the arc set, only if the time span t to t' is sufficiently long to let the PSV travel from the installation of order i to the installation of order j (sailing at a chosen speed in the weather conditions present at that time) and perform the service of j.

In other words, the time span includes the duration of both sailing and servicing, and might also include idling time of a PSV, if the installation of order j is not yet opened at the arrival of the vessel or if weather conditions are too harsh. Therefore, an arc $\big((i, t), (j, t')\big)$ represents a feasible option for conducting order j right after order i with respect to constraints on weather conditions, speed choice, and opening hours. By taking up all feasible arcs into a model, we ensure that the optimal solution can be found from the TDVRP-SO model. The systematic generation of the arcs is described next, and the arc generation procedure is also summarized in a pseudocode format in Appendix B.

The first steps in the arc generation procedure finds feasible combinations of orders i and j. As an offshore installation might place up to three different orders (resulting in three different nodes), several symmetric solutions can occur. These solutions all result in a PSV arriving and departing the installation at the same point in time and hence, with the same operating cost associated with the visit. To avoid these symmetries, the arc generation procedure ensures a fixed servicing sequence of orders belonging to

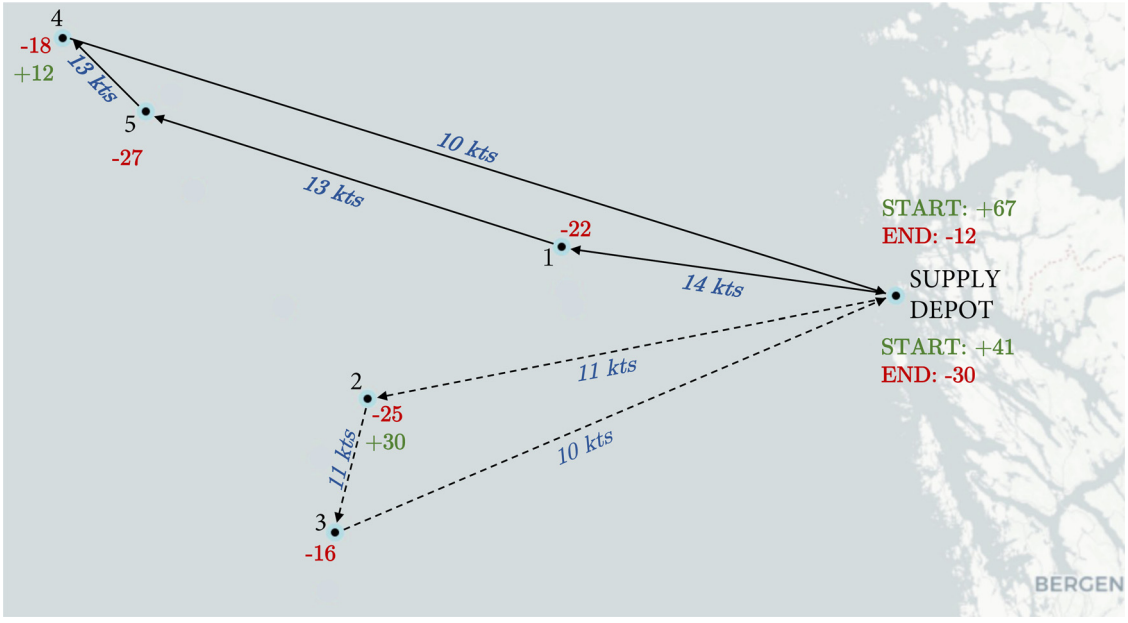

Fig. 4. Two feasible voyages for the example problem. The northernmost voyage is performed by *PSV1*. Positive numbers (+) represent pickup quantities, negative numbers (-) represent deliveries of cargo.

a same installation. We have defined the following hierarchy for services: A mandatory delivery (MD) order is serviced before an optional delivery (OD) order, which in turn is serviced before an optional pickup (OP) order. Thus, the following servicing patterns can occur in cases of multiple orders being placed by a same installation: MD → OD, MD → OP, OD → OP, or MD → OD → OP. Note that delivery orders (if any) are always handled before pickup orders at an offshore installation, which makes sense with regard to satisfying vessel capacity constraints.

Next, a set of feasible start times is determined for the arcs connecting two orders i and j. Initially, simple feasibility checks are performed to limit these start times: If i is the first order to be performed by a PSV, the time it takes to sail directly from the supply depot to the corresponding installation at maximum speed constitutes a lower bound on the earliest possible service start time. Similarly, an upper bound is defined by the minimum sailing duration from i back to the supply depot (at maximum sailing speed) such that the vessel arrives at the supply depot before the end of the planning period. Furthermore, we disregard all start times that are in conflict with the closing of the installation of order i (due to weather conditions or opening hours). To incorporate speed decisions, the arc generation procedure potentially generates multiple arcs for each possible start time, each indicating a possible speed for going from i to j. The resulting possible starting times for servicing order j are found by considering any opening hours or weather restrictions (the installation must be open for service at the time) and the duration of the service. For the final sailing leg of a voyage where a vessel returns to the supply depot after serving the last order, only the most cost-efficient arc (usually corresponding to sailing at the lowest speed, see Fig. 1) that does not violate the maximum duration constraint for a voyage is generated.

Note that both the sailing and servicing time being involved in an arc might be set to zero for some cases. For example, if both orders i and j are located at the same offshore installation, no sailing time is needed to approach order j. Also, if an arc ends at the supply depot, no servicing takes place there and, thus, the service time is zero. Furthermore, we require all PSVs to return to the supply depot before its next voyage is planned to start, i.e., normally either after 48 or 72 hours.

Each feasible arc is associated with a cost that includes the cost of energy (fuel) consumption during sailing, idling and servicing (taking into account the weather conditions at the actual time), as well as any potential chartering costs if the selected vessel is an additional spot vessel used to meet the total demand. The arc generation procedure is repeated for all available PSVs, including the available spot vessel(s). However, some vessels might remain unused in an optimal solution. A zero-cost arc is therefore generated for each vessel, connecting the origin and the destination node representing the supply depot.

As previously discussed, the generation of the time-space network ensures that several problem-specific constraints are respected and hence, can be omitted from the formulation of the mathematical model. Specifically, these constraints relates to the number of visits to an installation and the opening hours of these, the maximum allowed duration of a voyage and the feasible sailing speeds for each leg on a voyage.

3.2. Mathematical formulation

In the following, we present the mathematical notation needed to formulate the TDVRP-SO on the time-space network defined as described in Section 3.1. A set $\mathcal{V}$ of PSVs is available for departures on the forthcoming day (including both contracted and additional spot market vessels), each with a given capacity Q_v. The orders placed by the different offshore installations constitute the set $\mathcal{N}$, which consists of three disjoint subsets $\mathcal{N}^{MD}$, $\mathcal{N}^{OD}$ and $\mathcal{N}^{OP}$, representing the mandatory and optional delivery orders as well as the optional pickup orders, respectively. The arcs generated, as described in Section 3.1, are members of the arc pool $\mathcal{A}_v$ for each available vessel $v \in \mathcal{V}$. The duration of the planning period is discretized into a set of time points $\mathcal{T}$ that need further refining into subsets in order to present the arc-flow model:

- $\mathcal{T}^{S}_{ijv}$ is the set of *start times* of all those arcs that connect orders i (or the origin node o) and j (or the destination node d) for vessel v,
- $\mathcal{T}^{SS}_{ijtv} \subseteq \mathcal{T}^{S}_{ijv}$ contains the *start times* of those arcs that start at order i and end at order j at time t,
- $\mathcal{T}^{SE}_{itjv}$ holds the *end times* of those arcs that start at order i at time t and connect to order j (or the destination node d).

In the following, we use Fig. 5 as an example to show the generation of these subsets of time periods. It shows for two orders i and j the potential connections (arcs) at various times. The shaded horizontal bars represent periods in which servicing is prohibited at the installations, due to either restricted opening hours or rough weather conditions. In this example $\mathcal{T}^{S}_{ijv} = \{1, 2, 3, 6, 7, 8\}$ as these are points in time at which arcs from i to j have their start times. Furthermore, $\mathcal{T}^{SS}_{ij9v} = \{6, 7, 8\}$ is the set of start times that end at node j at time $t = 9$. Finally, $\mathcal{T}^{SE}_{i2jv} = \{3, 4\}$ are end times of arcs that originate at order i at time 2. In general, the different arcs represent different speed options for sailing between the installations where shorter arcs refer to higher chosen speed.

Furthermore, we introduce the following parameters: Each order i is specified by its size S_i, and optional orders $i \in \mathcal{N}^{OD} \cup \mathcal{N}^{OP}$ are also associated with a penalty cost C^{P}_{i} if being postponed. The arc cost, containing both fuel costs and a potential charter cost (an hourly rate for using spot vessels) is denoted $C^{A}_{it\,jt'v}$. Finally, time t^* is the starting point of all voyages, i.e., immediately after vessel preparation at the supply depot has ended on the first day of the planning period.

The binary decision variables $x_{it\,jt'v}$ are used to indicate whether or not arc $\big((i,\,t),\,(j,\,t')\big)$ is selected. Binary variable u_{iv} takes value 1 if order i is serviced by vessel v, 0 otherwise. Continuous variables l^{D}_{iv} and l^{P}_{iv} are introduced to keep track of the onboard cargo quantities of delivery and pickup loads for vessel v *after* servicing an order i.

We can now formulate the complete model for the TDVRP-SO. For a summary of the notation introduced and used in the mathematical model, see Appendix C.

Objective Function

$$\min \sum_{v\in\mathcal{V}} \sum_{((i,t),(j,t'))\in\mathcal{A}_v} C^{A}_{it\,jt'v} x_{it\,jt'v} + \sum_{i\in\mathcal{N}^{OD}\cup\mathcal{N}^{OP}} C^{P}_{i}\left(1 - \sum_{v\in\mathcal{V}} u_{iv}\right) \tag{1}$$

Constraints

$$\sum_{j\in\mathcal{N}} \sum_{t'\in\mathcal{T}^{SS}_{jitv}} x_{jt'itv} - \sum_{j\in\mathcal{N}} \sum_{t'\in\mathcal{T}^{SE}_{itjv}} x_{it\,jt'v} = 0, \quad i \in \mathcal{N}, t \in \mathcal{T}, v \in \mathcal{V} \tag{2}$$

$$\sum_{j\in\mathcal{N}\cup\{d\}} \sum_{t'\in\mathcal{T}^{SE}_{ot^*jv}} x_{ot^*jt'v} = 1, \quad v \in \mathcal{V} \tag{3}$$

$$\sum_{i\in\mathcal{N}\cup\{d\}} \sum_{t\in\mathcal{T}^{S}_{ijv}} \sum_{t'\in\mathcal{T}^{SE}_{itdv}} x_{itdt'v} = 1, \quad v \in \mathcal{V} \tag{4}$$

$$\sum_{v\in\mathcal{V}} u_{iv} = 1, \quad i \in \mathcal{N}^{MD} \tag{5}$$

$$\sum_{v\in\mathcal{V}} u_{iv} \leq 1, \quad i \in \mathcal{N}^{OD} \cup \mathcal{N}^{OP} \tag{6}$$

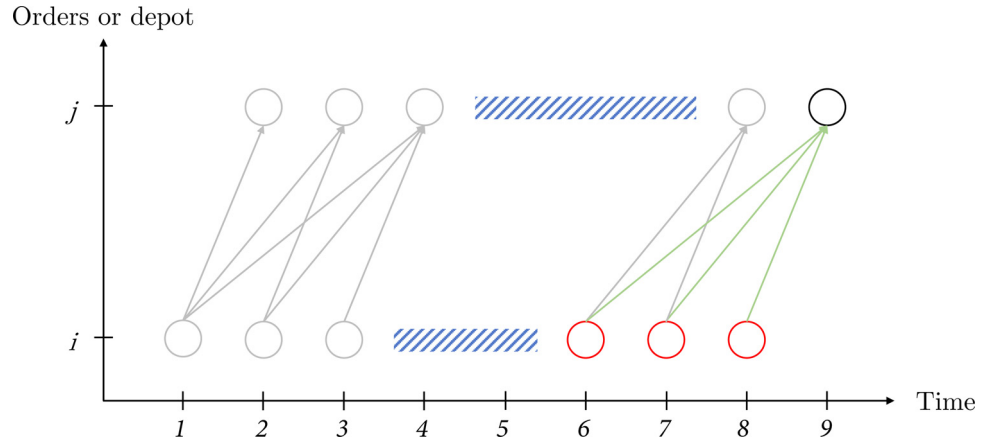

Fig. 5. An example for illustrating sets $\mathcal{T}^{S}_{ijv}$, $\mathcal{T}^{SS}_{ijtv}$, and $\mathcal{T}^{SE}_{it\,jv}$.

$$\sum_{i\in\mathcal{N}\cup\{o\}}\sum_{t\in\mathcal{T}^{S}_{ijv}}\sum_{t'\in\mathcal{T}^{SE}_{it\,jv}} x_{it\,jt'v} = u_{jv}, \quad j\in\mathcal{N}, v\in\mathcal{V} \tag{7}$$

$$l^{D}_{ov} = \sum_{i\in\mathcal{N}^{MD}\cup\mathcal{N}^{OD}} S_i u_{iv}, \quad v\in\mathcal{V} \tag{8}$$

$$l^{D}_{iv} + l^{P}_{iv} \le Q_v u_{iv}, \quad i\in\mathcal{N}, v\in\mathcal{V} \tag{9}$$

$$l^{D}_{jv} \le l^{D}_{iv} - S_j u_{jv} + Q_v\left(1 - \sum_{t\in\mathcal{T}^{S}_{ijv}}\sum_{t'\in\mathcal{T}^{SE}_{it\,jv}} x_{it\,jt'v}\right), \quad i\in\mathcal{N}, j\in\mathcal{N}^{MD}\cup\mathcal{N}^{OD}, v\in\mathcal{V} \tag{10}$$

$$l^{D}_{jv} \le l^{D}_{iv} + Q_v\left(1 - \sum_{t\in\mathcal{T}^{S}_{ijv}}\sum_{t'\in\mathcal{T}^{SE}_{it\,jv}} x_{it\,jt'v}\right), \quad i\in\mathcal{N}, j\in\mathcal{N}^{OP}, v\in\mathcal{V} \tag{11}$$

$$l^{P}_{jv} \ge l^{P}_{iv} + S_j u_{jv} - Q_v\left(1 - \sum_{t\in\mathcal{T}^{S}_{ijv}}\sum_{t'\in\mathcal{T}^{SE}_{it\,jv}} x_{it\,jt'v}\right), \quad i\in\mathcal{N}, j\in\mathcal{N}^{OP}, v\in\mathcal{V} \tag{12}$$

$$l^{P}_{jv} \ge l^{P}_{iv} - Q_v\left(1 - \sum_{t\in\mathcal{T}^{S}_{ijv}}\sum_{t'\in\mathcal{T}^{SE}_{it\,jv}} x_{it\,jt'v}\right), \quad i\in\mathcal{N}, j\in\mathcal{N}^{MD}\cup\mathcal{N}^{OD}, v\in\mathcal{V} \tag{13}$$

$$l^{P}_{dv} = \sum_{i\in\mathcal{N}^{OP}} S_i u_{iv}, \quad v\in\mathcal{V} \tag{14}$$

$$x_{it\,jt'v}\in\{0,1\}, \quad \big((i,t),(j,t')\big)\in\mathcal{A}_v, v\in\mathcal{V} \tag{15}$$

$$u_{iv}\in\{0,1\}, \quad i\in\mathcal{N}\cup\{o,d\}, v\in\mathcal{V} \tag{16}$$

$$l^{D}_{iv}\ge 0, \quad i\in\mathcal{N}\cup\{o,d\}, v\in\mathcal{V} \tag{17}$$

$$l^{P}_{iv}\ge 0, \quad i\in\mathcal{N}\cup\{o,d\}, v\in\mathcal{V} \tag{18}$$

The objective function (1) minimizes the sum of monetary costs (fuel and potentially chartering) for the selected arcs and penalty costs associated with the postponed orders. Constraints (2) preserve the vessel flow for all order nodes, while Constraints (3) and (4) ensure that each vessel departs from and arrives at the supply depot exactly once. All mandatory orders are serviced due to Constraints (5). Optional orders (both pickup and delivery) *might* be serviced according to Constraints (6). Constraints (7) synchronize the number of visits to an order node and the number of services for the same order through ensuring that an order is serviced by its assigned vessel. Load and capacity considerations are handled by Constraints (8) – (14). The total cargo quantity loaded on board a vessel v must equal the sum of orders serviced by the same vessel, see Constraints (8). From (9), the cargo capacity constraints for the vessels are respected at all points during a voyage. Constraints (10) and (11) control the load continuity of delivery loads between installations i and j for all vessels. Constraints (12) and (13) do the same for pickup loads. Constraints (14) ensure that the total cargo unloaded at the depot after a voyage equals the sum of all demands being picked up on the voyage. Constraints (15) – (18) specify the domains of the decision variables.

4. Adaptive large neighborhood search heuristic

Preliminary studies of solving the TDVRP-SO model by the commercial mixed-integer programming solver Gurobi showed that only small instances with up to 12 orders can be solved with good accuracy within a limited runtime of one hour. Therefore, for being able to solve full-scale problem instances based on real data in reasonable time, we have developed an Adaptive Large Neighborhood Search (ALNS) heuristic. The first application of ALNS was presented by Ropke & Pisinger (2006), who use the heuristic to solve a pickup and delivery problem with time windows. Other relevant applications include Cuesta et al. (2017) and Kisialiou et al. (2018b), where ALNS is applied to solve other planning problems related within offshore logistics, which motivates our choice of employing this method here. The key concepts of the heuristic are presented in the next subsections. These include additional destroy and repair heuristics dedicated to the TDVRP-SO, a local search heuristic for improving solutions, solving a speed optimization sub-problem, as well as an integration with a set partitioning model for recombining solutions.

4.1. Overview of the ALNS

The ALNS procedure is presented in Algorithm 1 . The algorithm initializes the current solution x by constructing a feasible solution as is later described in Section 4.2. Successively, destroy and repair heuristics (presented in Section 4.4) are used to generate a candidate solution x' from x. If the candidate solution is promising, a local search is performed, attempting to find improving solutions close to the candidate solution, see Section 4.5. Provided that x' is accepted when applying a simulated annealing-based acceptance criterion that is described in Section 4.3, the current solution x is

Algorithm 1: ALNS.

Input : Total number of ALNS iterations (I^{ALNS}), number of segment iterations (I^S), number of iterations after which the VCP is solved (I^{VCP})
Output: Global best solution, x^*

set current solution x by constructing a feasible solution
set the global best solution, $x^* \leftarrow x$
set the current segment, $m \leftarrow 1$
for $iteration = 1$ to I^{ALNS} **do**
  select a destroy heuristic and a repair heuristic using the adaptive weights w_{dm} in the current segment m
  generate a candidate solution x' from the current solution x using the selected destroy and repair heuristics
  apply local search for improving candidate solution x' if possible
  if x' is accepted by the simulated annealing criterion **then**
    $x \leftarrow x'$
  end
  if I^{VCP} iterations have passed since the VCP was solved **then**
    solve VCP and generate a new candidate solution x''
    if x'' is accepted by the simulated annealing criterion **then**
      $x \leftarrow x''$
    end
  end
  if $f(x) < f(x^*)$ **then**
    $x^* \leftarrow x$
  end
  update scores π_d of the destroy and repair heuristics
  if I^S iterations have passed since last weight update **then**
    update weight $w_{d,m+1}$ for method d to be used in segment $m+1$ based on the scores π_d obtained for each method in segment m.
    calculate new selection probabilities $P(d, m+1)$ for the weights
    update the simulated annealing temperature, T.
    update current segment, $m \leftarrow m+1$
  end
end

set to be x'. Moreover, a set partitioning model termed the Voyage Combination Problem (VCP) is presented in Section 4.6, which is solved every I^{VCP} iterations to use a pool of voyages for generating a new candidate solution x''. This solution is also tested for acceptance by the simulated annealing-based criterion. If the current solution x is better than the so-far best found solution x^*, x^* is updated to this new best solution. Furthermore, the total number of ALNS iterations is divided into *segments* of I^S iterations. Each time I^S iterations have been completed, the weights of the destroy and repair heuristics as well as the cooling temperature used in the simulated annealing-based acceptance criterion are updated. The ALNS terminates after a maximum of I^{ALNS} iterations, returning the best solution found.

4.2. Constructing a feasible initial solution

Constructing an initial solution starts with all orders being unscheduled, which is expressed by a set $\mathcal{U} = \mathcal{N}$. Orders are then inserted sequentially in initially empty voyages until all orders are either assigned to a voyage or placed in a set $\mathcal{P}$ of orders postponed for a later service. The order insertion is performed using a *greedy insertion* heuristic, which evaluates the marginal increase of the objective function due to additional sailing and servicing time and hence, additional fuel consumption, which is induced by the insertion of an order i into the considered voyage. This evaluation is performed for the set of all feasible insertion *positions*, including the insertion into the set $\mathcal{P}$ of postponed orders (and hence, inducing the penalty cost associated with the order). The least costly insertion is then selected, and the procedure is repeated for all remaining orders in the set of unscheduled orders $\mathcal{U}$.

For each possible insertion of orders into a partial solution, a feasibility check needs to be performed. First, the total duration of a voyage after inserting a new order is not allowed to exceed the maximum route duration. This feasibility check requires only one possible sailing speed option for the voyage to remain feasible. Thus, the maximum possible speed (adjusted for any limitations due to weather conditions or opening hours) is applied to all sailing legs in order to check if the newly inserted order can be serviced without violating the maximum duration. Next, the voyage must be feasible with respect to vessel capacity after inserting the order. Inserting a delivery order into a voyage increases the initial onboard load of the PSV, whereas insertion of a pickup order increases the onboard load for the return leg back to the supply depot. Eventually, the insertion of an order must comply to the servicing sequence hierarchy at each installation. This means that inserting an optional pickup or delivery order can never take place without first inserting the mandatory delivery to the same installation in cases where such order exists.

Furthermore, each modification of a voyage results in a new scheduling sub-problem that aims to find the most cost-efficient sailing speeds along the voyage. This needs to be solved for every insertion of orders during the construction heuristic as well as for each iteration of the ALNS after selecting a pair of repair and destroy heuristics in order to compute the cost of a solution. This so-called *Supply Vessel Speed Optimization Problem* (SVSOP) is solved by discretizing arrival times to form a time-space node network, with feasible arcs connecting the nodes in a *Directed Acyclic Graph*. We refer the interested reader to Fagerholt et al. (2010) for a detailed description of this graph based modeling approach. Finding the least costly path through the network resembles a shortest path problem, which we solve simply by applying Dijkstra's algorithm. It is important to note that the SVSOP needs to be solved for each voyage in each cost evaluation step during the ALNS heuristic and, hence, an efficient solution method for the speed-optimizing subproblem is an important factor affecting the overall ALNS performance.

4.3. Large neighborhood search

In each iteration of the ALNS heuristic, the current solution is partially modified and then reconstructed using one pair of destroy and repair heuristics from the set of available heuristics. This section presents the heuristics implemented in our ALNS heuristic, including a set of additional problem-specific heuristics added to exploit the structure of the TDVRP-SO.

Destroy heuristics: We introduce in total six destroy heuristics, the first four (random removal, related removal, worst removal, cluster removal) are inspired by Shaw (1998), Ropke & Pisinger (2006), and Pisinger & Ropke (2007). The latter two heuristics, named spread removal and spot vessel removal, are problem-specific.

Random removal: Specifying an upper limit of q^{ALNS} orders, this destroy heuristic removes orders randomly from both the existing voyages and the set of postponed orders obtained in the previous solution. In other words, this heuristic extends the set of unscheduled orders until a limit $|\mathcal{U}| = q^{ALNS}$ is reached.

Related removal: First proposed by Shaw (1998), the idea of this heuristic is to remove q^{ALNS} *related* orders from the solution. After selecting one order randomly, remaining orders $i \notin \mathcal{U}$ are sorted according to their *relatedness* to the selected order. Here, we define relatedness as the great-circle distance between the orders (installations). We then remove from this list the order at position R. This position is computed by

$$R = y^P \times |\{\text{RemainingOrders} \notin \mathcal{U}\}|,$$

where y is a continuous value drawn from the interval [0,1] and $P > 1$ is a parameter determining the degree of randomness in the selection. Here, larger values for P mean less randomness such that R tends to the first members of the sorted list (i.e., the orders most related to the already selected order) whereas values of P close to 1 lead to a more uniform selection probability for the orders in the list. Furthermore, recall the servicing hierarchy defined for installations with multiple orders, from which removing a mandatory delivery order of an installation means that any optional orders to the same installation automatically must be removed too if such orders have been scheduled to the particular voyage already. This further reduces the size of remaining orders in the sorted list of related orders.

Worst removal: This heuristic aims at finding the single order contributing most to the objective function value. For an order assigned to a voyage, this cost increase is found by solving the SVSOP as described in Section 4.2 and comparing the voyage cost with and without the specific order being included. For a postponed order in set $\mathcal{P}$, the cost increase is set to the penalty cost C_i^P.

Cluster removal: Similar to the *related removal*, the *cluster removal* searches for orders related to each other due to their geographical proximity. However, this heuristic removes entire clusters rather than selecting isolated orders from the relatedness list. Pisinger & Ropke (2007) propose cluster removal to reduce the probability of removing orders here and there at some voyages that would then be likely re-inserted into the exact same positions of these voyages. Removing whole clusters of orders reduces this risk. In this regard, many of the offshore installations in the North Sea can be considered as clusters, as they often are located less than 20 km apart from each other. This makes cluster removal suited for the TDVRP-SO. Clustering is performed for the orders on one of the scheduled voyages using a *k-means* algorithm with $k = 2$. Thus, two orders are selected randomly as centroids, and the remaining orders are split into two clusters based on their closeness to these centroids. Afterwards, the location of two new centroids (the geographical center among the clustered orders) are calculated, and orders are reclustered based on these new centroids. This process is repeated until the location of the two centroids remain unchanged in an iteration. One of the clusters is then selected randomly and removed from the solution. If less than q^{ALNS} orders have been removed, the cluster removal heuristic is repeated for another voyage. If there are no more voyages to evaluate, the heuristic terminates even if less than q^{ALNS} orders have been removed.

Spread removal: Some offshore installations can be far away from each other. Servicing their orders on a same voyage will lead to rather long sailing times and high fuel costs. Hence, a reasonable way to finding more efficient voyages can be to avoid these combinations of orders in a solution. The spread removal sequentially removes those orders that are associated with installations being farthest away from the others (i.e., with the longest minimum distance to the nearest neighbor installation) until q^{ALNS} orders are removed. Fig. 6 illustrates the heuristic, where orders (installations) 1 and 5 are removed sequentially from the scheduled voyage as they are located farthest away from their nearest neighboring orders.

Spot Vessel Removal: This is a simple, problem-specific destroy heuristic that removes *all* orders serviced by a spot vessel in the solution. If there are multiple spot vessels used in a solution, the vessel is chosen randomly. Clearly, this heuristic is only applied if the current solution actually uses spot vessels.

Repair heuristics: After destroying parts of a solution, the whole set of the unscheduled orders $\mathcal{U}$ is subject to a repair process, which attempts to reinsert these orders into the scheduled voyages or, otherwise, postpone them by adding them to set $\mathcal{P}$. We use here four repair heuristics, where the first two are inspired by Ropke & Pisinger (2006) and the other two are designed specificly for the TDVRP-SO.

Basic greedy insertion: For each order, the cost of inserting it at each possible feasible insertion points is evaluated. This is done by solving subproblem SVSOP-subproblem for each such insertion option. If an order is postponed due to a lack of feasible insertion options, it gets assigned the penalty cost again. The order to insert is then found based on the least-cost contribution to the objective function. More formally, if the feasible insertion points of order i are denoted by $s \in \mathcal{S}_i$ (i.e., the positions along a voyage or in the set of postponed orders where the order can be inserted), the basic greedy selects the order to insert through formula

$$\underset{i \in \mathcal{U},\, s \in \mathcal{S}_i}{\operatorname{argmin}} \Delta\text{Cost}(i(s)).$$

However, if the best order to insert is optional and associated with an installation where a mandatory order is yet to be placed, the heuristic instead selects the next best order to insert. This is done to avoid postponing optional orders that might be serviced after servicing the mandatory order to the same installation yet to be inserted.

Regret insertion: Regret-k insertion compares the k best insertions of an order as the sum of cost differences between the best insertion point for the order and the $k-1$ next best insertion positions. As for the basic greedy insertion, the SVSOP-subproblem must be solved for each evaluation of additional costs induced by an insertion. If the order with the largest regret-k value is an optional order, the next best order to insert is selected, and the process continues until the set of unscheduled orders $\mathcal{U}$ is emptied. With $i(s_1)$ denoting the best insertion, and $i(s_{k'})$ the k'-th best insertion for order i, the next order to select is found from the regret-k insertion heuristic as

$$\underset{i \in \mathcal{U}}{\operatorname{argmin}} \sum_{k'=2}^{k} \Big(\Delta\text{Cost}(i(s_1)) - \Delta\text{Cost}(i(s_{k'})) \Big).$$

Maximum penalty cost insertion: This heuristic aims to prevent postponing orders i associated with high penalty costs C_i^P. For this, two lists are created from the set of unscheduled orders: The mandatory orders are first assigned to the voyages, before a list of optional orders sorted by descending penalty costs is created. Insertion of orders is done in a greedy manner, as described for the *basic greedy insertion*, where orders are inserted at their cheapest feasible insertion point. As opposed to the basic greedy insertion, this heuristic does not select the best feasible insertion point among *all* unscheduled orders as the next order to evaluate is already given from the sorted list.

Maximum order size insertion: Insertion of orders takes place in a similar manner as by the *maximum penalty cost insertion*. In this heuristic, however, the orders are inserted into two sorted lists, one for mandatory orders and one for optional orders, based on the *size* of the orders. Mandatory orders are evaluated then before the optional orders, and all orders are inserted in a greedy manner based on the best possible insertion point along a voyage.

Applying noise in the insertion methods: Ropke & Pisinger (2006) find the proposed insertion heuristics to be quite myopic,

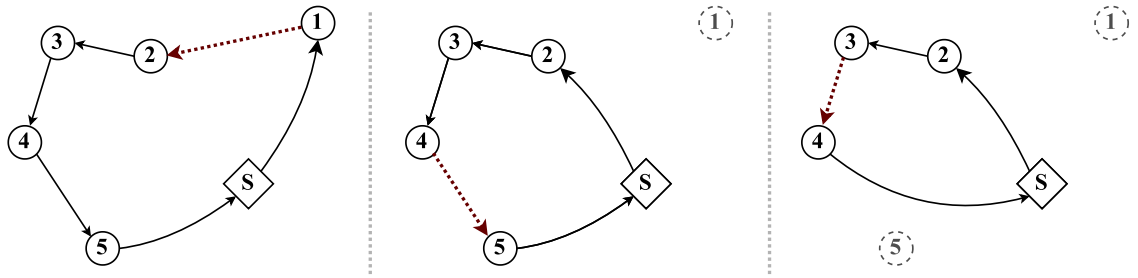

Fig. 6. Two iterations of the *spread removal* heuristic, removing the order on a voyage placed at an installation with the longest minimum distance to the nearest neighboring installation, indicated by the thicker, dashed arrows (disregarding sailing from and to the supply depot, S). In the next iteration, order 4 would have been removed from the voyage.

always searching for local improvements of solutions. To avoid getting stuck in local optima, a 'noise term' can be used to force the heuristics to also explore moves that are not the most effective ones according to the selected repair operator. For each additional cost, C, induced by an insertion of an order (due to longer servicing and sailing times and hence, a larger fuel consumption), a *noise* value is sampled randomly from the interval $[-\max N, \max N]$. We define $\max N = \eta \times \max\{d_{ij}\}$, i.e., as a certain portion of the maximum distance between two offshore installations (η is a tunable parameter). We then proceed with the modified cost $C' = \max\{0, C + noise\}$ when evaluating the performance of a given insertion heuristic.

Acceptance and stopping criteria: We use a simulated annealing-based approach similar to what Ropke & Pisinger (2006) present for accepting candidate solutions even in the case where the objective function value of the candidate solution $f(x')$ is *worse* than the currently best known solution's objective function value $f(x)$. This happens with a probability of $e^{-(f(x')-f(x))/T}$, where $T > 0$ is referred to as the *temperature*. T is decreased from an initial value T_{start} by scaling the current temperature by a fixed cooling rate $\xi \in [0, 1]$ every I^S iteration. In other words, the probability of accepting a non-improving candidate solution is largest during the early phases of the search, allowing for more exploration of the solution space.

4.4. Adaptive selection of destroy and repair heuristics

The selection of a pair of destroy and repair heuristics is controlled by a roulette wheel selection where weights associated with each heuristic are adjusted based on their success rate in previous iterations. All weights are assigned an identical non-zero value at the start of the ALNS algorithm. The adaptivity in the ALNS heuristic then relates to the repeated update of these weights, making the selection of effective heuristics more frequent. For this, the search process is divided into *segments* m, each containing I^S iterations. The weights are updated at the start of a new segment, and for each heuristic d the weights are adjusted according to the relationship

$$w_{d,m} = (1 - r)w_{d,m-1} + r\frac{\pi_d}{\theta_d},$$

where the new value is a weighted average of the previous weight $w_{d,m-1}$ and a *reaction term* consisting of the tunable reaction parameter r, the heuristic score π_d, and the number of times θ_d that the heuristic d has been selected in segment $m - 1$. The score π_d is reset to zero at the start of each new segment and might then increase throughout the iterations of the current segment by so-called *rewards* in three different ways. These rewards are given 1.) if a candidate solution results in global improvement (giving reward σ_1), 2.) if a candidate solution is better than the current solution and has not been accepted before (reward σ_2), or 3.) if a candidate solution has not been accepted before and is worse than the current solution, but is still accepted by the simulated annealing process (reward σ_3). The heuristic score is aggregated throughout the segment based on the sum of obtained rewards. Finally, by introducing I as the set of all destroy and repair heuristics, we set the probability for choosing heuristic $d \in I$ in the roulette wheel selection process of a segment m as

$$P(d, m) = \frac{w_{dm}}{\sum_{\hat{d} \in I} w_{\hat{d}m}}.$$

4.5. Extending the ALNS with a local search

In addition to the regular destroy-and-repair procedure repeated in each iteration of the ALNS, we want to further investigate the neighboring solution space to the current solution at regular intervals. Thus, we introduce a *local search* (LS) with specific local search operators (LSOs) that perform small alterations of the current solution in a given ALNS iteration. Feasibility must still be obtained for all conducted changes, which we ensure by performing the feasibility checks described in Section 4.2.

Local Search Operators: We propose the following LSOs, which are inspired by the work of Gendreau, Hertz, & Laporte (1992) and Korsvik, Fagerholt, & Laporte (2011):

Relocation operators: Relocating of a single order can take place either within a voyage (defined as *intra*-voyage relocate) or between two different voyages (*inter*-voyage relocate). Sequential relocation of several orders can be done within the same iteration but only in case that the installation associated with the selected order has placed several orders that all have been scheduled on the current voyage. This way, feasibility with regards to the limit of at most one visit to each installation is ensured.

Exchange operators: This operator exchanges the positions of orders that might either be currently scheduled in the same voyage (*intra*-voyage exchange) or in two different voyages (*inter*-voyage exchange). Note, if an installation has placed several orders, *all* orders must be exchanged in the same move to ensure feasibility. This is achieved by a so-called *voyage exchange* operator that simply swaps the voyages sailed by the two involved vessels among each other.

Postponement operators: Finally, we have two simple local search operators that either insert so-far postponed orders into a voyage, or postpone currently scheduled orders by performing the opposite move.

Search strategy for the local search: The local search supports the ALNS heuristic in producing high-quality solutions. As the search strategy for the local search can be quite tideous, a sequence of LSOs is performed only when the ALNS has produced promising solutions, defined as solutions with less than β% higher objective function value than the current best solution. Each time the local search is performed, the search strategy follows a sequential procedure: First, one local search operator (LSO) is applied at the time. For each LSO, all possible moves for all orders are found and evaluated based on the same (initial) candidate so-

lution. Second, the most improved solution produced by the first operator is then passed on as input for the next search operator and this process is repeated until all operators have been applied. Third, if no moves leads to improvement for a LSO, the evaluated candidate solution is used as input also for the next LSO. Finally, when the search from the last LSO is completed, the output solution is returned for use in the following iterations of the ALNS.

4.6. Extending the ALNS with a set partitioning model

A full solution created by the ALNS heuristic for a moderately large test instance will usually involve the use of several vessels (i.e., with several voyages created). As a result, a relatively poor solution might contain scheduled voyages for some of the vessels that are more cost-effective than the overall objective function value for the full solution suggests. Inspired by Homsi, Martinelli, Vidal, & Fagerholt (2020), we introduce a set partitioning model where voyages generated in ALNS solutions can be recombined into new and perhaps more cost-effective solutions within short solution time. The problem is solved at every I^{VCP} iterations of the ALNS heuristic. We name this extension of the ALNS the *Voyage Combination Problem* (VCP). For the VCP we introduce the following additional notation: We define the set $\mathcal{K}$ for all offshore installations containing two optional orders (i.e., both OD and OP) but no mandatory orders. Furthermore, $\mathcal{R}_v$ and $\tilde{\mathcal{R}}_{kv} \subseteq \mathcal{R}_v$ are the set of available voyages for vessel v, and the subset of voyages for vessel v visiting installation k while *only* servicing one of the two optional orders, respectively. Two new parameters are C_{rv} as the cost of performing voyage r for vessel v and the binary parameter A_{irv} equal to one if order i is serviced on voyage r by vessel v. The new binary decision variables are defined as y_{rv} indicating whether a voyage r is sailed by vessel v or not, and z_i equal to one if optional order i is *not* serviced by any vessel sailing any voyage, zero otherwise.

The VCP can then be formulated as follows:

$$\min \sum_{v\in\mathcal{V}} \sum_{r\in\mathcal{R}_v} C_{rv} y_{rv} + \sum_{i\in\mathcal{N}} C_i^P z_i \tag{19}$$

$$\sum_{r\in\mathcal{R}_v} y_{rv} \leq 1, \quad v \in \mathcal{V} \tag{20}$$

$$\sum_{v\in\mathcal{V}} \sum_{r\in\tilde{\mathcal{R}}_{kv}} y_{rv} \leq 1, \quad k \in \mathcal{K} \tag{21}$$

$$\sum_{v\in\mathcal{V}} \sum_{r\in\mathcal{R}_v} A_{irv} y_{rv} + z_i = 1, \quad i \in \mathcal{N} \tag{22}$$

$$z_i = 0, \quad i \in \mathcal{N}^{MD} \tag{23}$$

$$y_{rv} \in \{0,1\}, \quad r \in \mathcal{R}_v, v \in \mathcal{V} \tag{24}$$

$$z_i \in \{0,1\}, \quad i \in \mathcal{N} \tag{25}$$

As for the mathematical formulation of the TDVRP-SO, the objective function in (19) minimizes the monetary costs of voyage costs and penalty costs for postponing optional orders. Constraints (20) sets the upper limit on voyages assigned to each vessel, and Constraints (21) ensure that at most one visit to installation $k \in \mathcal{K}$ is performed. Without these constraints, two voyages both visiting the same installation and servicing different optional orders might take place, violating the requirement of a maximum of one visit to an installation. From Constraints (22), all orders must be serviced on at most one voyage (or be postponed), while all mandatory orders must be serviced in accordance with Constraints (23). The binary domains for the decision variables are specified in (24) and (25).

5. Computational experiments: A case from the North Sea

This section presents the results obtained from computational studies on the TDVRP-SO. We start by giving details on the implementation of our heuristic and the generation of test instances, which are based on real offshore logistics operations in the North Sea, see Section 5.1. This is followed by an analysis of ALNS extensions and its competitiveness to a commercial solver (Sections 5.2 and 5.3), as well as the value of speed optimization and of considering weather forecasts (Sections 5.4 and 5.5), providing valuable managerial insights.

5.1. Test instances and implementation details

Our experiments consider Norway's offshore production of oil and gas in the North Sea. For generating test instances, we consider the offshore logistics network related to the Mongstad supply base, situated at the west coast of Norway. Mongstad is one of the largest harbors related to oil-and-gas activities in Europe in terms of cargo throughput, and 27 offshore installations are being serviced by PSVs from this supply base. For each installation, an identification code, the geographical location in terms of latitude and longitude, opening hours and a *standard order size* used to generate various test instances, are given in Table 2. The locations refer to geographical coordinates, opening hours are indicated by the hours each installation can be serviced by the PSVs, and standard order size is the size of a typical order from the installation. Although orders for deck cargo can vary greatly both in shape and dimension, the size of an order is normally expressed at an aggregated level. Demands are specified using square meters of required deck area as unit, but for simplicity, these demands have been converted to a corresponding number of offshore containers in Table 2 and in the model implementation. This conversion eases the estimation of the required service time needed, which is estimated based on the number of containers to be loaded/unloaded, while maintaining a sufficient level of detail in planning the cargo distribution.

The standard order size applies to all three types of orders considered for an installation, namely MD, OD and OP. The relative order sizes between the installations are estimates based on the activity level of these installations (a function of both production volumes and number of employees), where we have set the largest order size for mandatory deliveries equal to 20% of the deck capacity on the contracted fleet vessels. We consider a fleet of five contracted PSVs with cargo capacities of 125–131 cargo units (containers) and specific fuel consumption rates ranging from 540 to 608 kg/hour. The computational study also involves the possibility of chartering a spot vessel, with a similar cargo capacity and a slightly higher fuel consumption due to higher age.

Test instances are generated in the following way: First, we sample from the set of 27 installations a subset of installations I^{orders} that have placed orders in the particular test instance. The number of orders of each type, i.e., MD, OD or OP, is uniformly sampled from intervals representing 50–70%, 20–40% and 20–40% of I^{orders}, respectively. Thus, more orders than installations can be sampled, and the orders are distributed randomly to the sampled subset of installations. Furthermore, the size of each order is determined from a uniform distribution spanning 50–150% of the standard order sizes given in Table 2 for each installation and order type. Finally, the generation process includes the allocation of a sufficient number of available vessels to each test instance (including the spot vessel), large enough to cover the total order size. We generate a total of 60 test instances divided into 12 *instance groups* based on the number of installations I^{orders} as sampled by the instance generation procedure. We denote the test instances based on the number of offshore installations (I), number of orders (O), and number of vessels (V). To distinguish the five instances that

Table 2
Data associated with the installations in the Mongstad case. Standard order sizes are given for MD, OD and OP orders, respectively.

#	Code	Latitude/ Longitude	Opening hours	Standard order sizes	#	Code	Latitude/ Longitude	Opening hours	Standard order sizes
1	TRO	60.64/3.72	07–19	27-14-28	15	KVB	61.07/2.50	00–24	27-14-28
2	TRB	60.77/3.50	07–19	20-10-21	16	VMO	61.04/2.34	00–24	20-10-21
3	TRC	60.88/3.60	07–19	14-7-14	17	WEL	61.04/2.34	00–24	22-11-23
4	CPR	60.74/3.61	00–24	22-11-23	18	VFB	60.78/2.89	00–24	27-14-28
5	SEN	60.95/3.58	00–24	23-12-24	19	WEP	60.85/2.65	00–24	22-11-23
6	SDO	60.85/3.62	00–24	18-9-19	20	HUL	60.85/2.65	00–24	20-10-21
7	SEQ	60.89/3.67	00–24	23-12-24	21	STA	61.25/1.85	07–19	20-10-21
8	OSE	60.48/2.82	00–24	27-14-28	22	STB	61.20/1.82	00–24	20-10-21
9	OSB	60.48/2.82	00–24	14-7-14	23	STC	61.29/1.90	00–24	27-14-28
10	OSC	60.60/2.77	00–24	14-7-14	24	GFA	61.17/2.18	00–24	20-10-21
11	OSO	60.70/2.93	00–24	20-10-21	25	GFB	61.20/2.20	00–24	20-10-21
12	SSC	60.70/2.93	00–24	17-9-18	26	GFC	61.20/2.27	00–24	27-14-28
13	OSS	60.38/2.79	00–24	20-10-21	27	SOD	60.90/3.81	00–24	22-11-23
14	DSD	60.08/2.63	00–24	15-8-16					

Table 3
Test instances used for the computational study, grouped by the number of installations in each instance. There are five test instances within each group. The number of orders might vary, as shown, but the number of available fleet vessels remains equal. Chartering the spot vessel is also possible for all instances in all instance groups.

Installations	Orders	MD	OD	OP	Fleet vessels	Notation
5	5–7	3	1–2	1–2	1	5-(5–7)-1-X
7	8–9	4–5	2	2	1	7-(8–9)-1-X
9	9–11	5–6	2–3	2–3	1	9-(9–11)-1-X
11	12–15	6–7	3–4	3–4	2	11-(12–15)-2-X
13	14–18	7–9	3–5	3–5	2	13-(14–18)-2-X
15	15–21	8–10	3–6	3–6	2	15-(15–21)-2-X
17	18–23	9–11	4–6	4–6	3	17-(18–23)-3-X
19	19–25	10–13	4–7	4–7	3	19-(19–25)-3-X
21	23–28	11–14	5–8	5–8	3	21-(23–28)-3-X
23	24–31	12–16	5–9	5–9	4	23-(24-21)-4-X
25	26–33	13–17	5–10	5–10	4	25-(26–33)-4-X
27	28–37	14–18	6–10	6–10	5	27-(28–37)-5-X

Table 4
Four weather states (WS) based on significant wave heights (SWH) and the impact on the PSVs in terms of enforced reduction of speed (Δv), as well as relative increases of service times (ΔT^{SERV}) and fuel consumption ($\Delta f^{S,I}$) for both servicing and idling.

WS	SWH (m)	Δv	ΔT^{SERV}	$\Delta f^{S,I}$
0	≤ 2.5	0	0	0
1	(2.5, 3.5]	0	20%	20%
2	(3.5, 4.5]	-2 knots	30%	30%
3	> 4.5	-3 knots	*Forced idling*	100%

belong to a group, we append a fourth index X. Thus, we present the test instances as I-O-V-X. Table 3 displays the characteristics of the different instance groups, and how the number of orders of different types might vary for instances involving the same number of installations. All instances in the same instance group have the same number of vessels available for operation.

A key question of the further data generation is how to capture the impact of various weather conditions on vessel speed, fuel consumption, and service times of the PSVs. We use here the approach proposed by Halvorsen-Weare & Fagerholt (2011), who defined four discretized weather states (wave height intervals). Key information of these weather states is provided in Table 4. We see that rough weather states WS 2 and WS 3 enforce that vessels have to reduce their speed by 2 and 3 knots, respectively. Furthermore, service times are increased by 20% to 30% in WS 1 and 2, whereas service operations cannot take place under WS 3 at all. The relative increase in fuel consumption from more harsh weather is also given from the table. Furthermore, fuel consumption at a given speed in a given weather condition is computed using the cubic fuel-speed relationship presented by Norlund & Gribkovskaia (2017). From this, if the wave conditions lead to a speed offset of 2 knots, the specific fuel consumption (in kg/hour) for sailing e.g., in 12 knots is assumed to be equal to the consumption for sailing in 14 knots in fair weather.

We construct from these weather states three *weather scenarios* specifying how weather changes throughout the planning period, see Fig. 7. The *Fair* weather scenario involves small wave heights (WS 0) throughout the entire planning period, such that vessels are not affected by weather at any time. In the *Mixed* weather scenario, the wave heights are steadily increasing during the first half of the planning horizon before they decrease in the second half. The period of roughest weather (WS 3) lasts for approximately 8 hours. The *Rough* weather scenario is characterized by moderate wave heights (WS 1) in the beginning, which then drastically worsens in the next days such that, finally, weather state WS 3 is reached, where installations cannot be serviced any more. Since the geographical operating area is not very large, we assume for all scenarios that weather is spatially static, meaning that all parts of the geographical region experience the same weather conditions at the same time.

Furthermore, a set of vessel-specific parameters is required as input to the TDVRP-SO. The speed interval (low-high) in which the fuel consumption relationship proposed by Norlund & Gribkovskaia (2017) is valid, is set to 10–14 knots. Based on current values at the time the computational studies were conducted, fuel costs are set to 276 USD/ton, and the spot vessel might be chartered for 608 USD/hour. Servicing time at an installation is determined by the cargo handling rate (lifting of container units), which we set to 10 minutes per unit. We apply these values identically for all vessels.

Lastly, the penalty cost associated with postponing optional orders must be given. Here, a trade-off must be obtained between a frequent use of spot vessels on one hand, and congestion of orders to service at later voyages on the other. Therefore, we set the penalty cost of an optional order as the sum of servicing costs and sailing from the supply base to the installation and back (at the design speed of the PSV), i.e., the fuel cost associated to a voyage that exclusively services this particular order. Setting the penalty cost for postponing an optional order this way satisfies two criteria: 1)

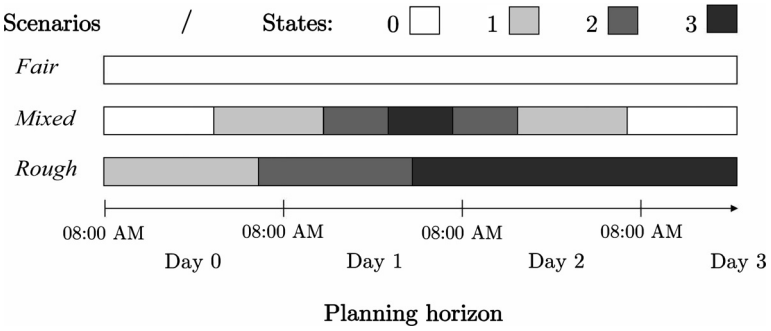

Fig. 7. Visualization of the applied weather scenarios, detailing which weather states are experienced at what times during the planning horizon.

An optional order should be serviced by a PSV on a voyage given that capacity and voyage duration constraints can be respected. 2) A spot vessel should never be chartered solely to service an optional order.

With regard to implementation, the ALNS heuristic, with its local search and solving of the Supply Vessel Speed Optimization (sub-)Problem (SVSOP), is implemented in Java 11.0.2. The time-space network generation is implemented in Python 3.8.5. The mathematical formulation from Section 3 and the voyage combination problem in Section 4.6 are implemented in Gurobi 9.1.1. We perform all tests on a Lenovo ThinkSystem SD530 computer running on Linux CentOS 7, with two 3.6 GHz Intel Xeon Gold 6244 processors with 8 cores and 384GB RAM. The maximum solution time for the ALNS heuristic is set to 600 seconds per instance, as this is considered to be the maximum solution for this problem in practice. Gurobi is allowed to run for 3600 seconds (one hour) when solving the mathematical model in order to increase the number of obtained solutions for benchmarking. As the ALNS heuristic involves randomized components, we solve each test instance five times and report the average objective values and solution times when discussing the ALNS performance. We identify the SVSOP-subproblem of Section 4.2 as being computationally expensive, which is why we utilize parallel computations to increase the efficiency of the heuristic. This is done by evaluating pairs of destroy and repair heuristics in parallel. By storing all solutions of the subproblem in a cache, we ensure that each specific subproblem (with given vessels and voyages) is solved only once. A detailed discussion of the tuning of all parameters that are involved in the ALNS is provided in Appendix A.

5.2. Assessment of the ALNS extensions

To compare the basic ALNS heuristic with the extended variants including either the local search described in Section 4.5, or the Voyage Combination Problem (VCP) (i.e., the set partitioning model presented in Section 4.6), or both, we experiment with four different configurations of the ALNS for all test instance groups. Three performance measures are reported, namely the coefficient of variation (which is a measure of the consistency in the solution quality provided), the solution time per instance, and a gap indicating the quality of the obtained solutions. We define this gap as the relative difference between the objective value achieved by a considered ALNS configuration compared to the best objective value obtained by any configuration. We report this gap as an average over the five runs per instance and the five instances per group.

The comparison of the ALNS configurations is summarized in Table 5. We observe that all extension configurations on average reduce both the gap and the coefficient of variation compared to the basic ALNS heuristic. The increase in solution time is deemed acceptable, as all instances are solved within the specified time limit of 600 seconds.

The impact of introducing each of the two extensions in isolation is evaluated by considering the results of configurations 'ALNS + LS' and 'ALNS + VCP' in Table 5. When the local search is included in the ALNS heuristic, we find that more than 90% of the updates of the current best solution are found by the local search, with *voyage exchange* as the best performing operator. Also the intra-voyage relocate and exchange operators contribute to finding better solutions. Furthermore, we observe that the best solution to an instance is found through the local search in 70%, through the destroy and repair heuristics in 25%, and through the construction heuristic in 5% of the instances.

The set partitioning extension to the ALNS heuristic (i.e., the VCP) is also found effective in providing solutions of good quality in relatively short time. The VCP is solved at predefined intervals during the search, controlling the number of iterations between each run of the VCP. Test instance groups 5–11 are solved to best quality within 200 iterations of the pure ALNS, and, thus, the VCP does not contribute to finding new and better solutions for these instances. However, the VCP provides 18% of the best solution updates during the full search for the larger instances from groups 13–27, which indicates that this approach is well suited for the ALNS when solving larger problem instances.

Furthermore, we observe that the full extension (ALNS + LS + VCP) provides the best and most consistent solutions for all test instances with up to 19 installations. For the larger instances, gaps are at most 0.31% and consistently below the gap of any other ALNS-configuration, which shows that applying both extension leads to the best solution quality on average. The increase of solution times by approximately 15% for the full extension compared to the configuration including only local search is considered acceptable. In the remainder of this paper, we thus refer to the fully extended configuration 'ALNS + LS + VCP' as the ALNS heuristic, and all further computational results presented in this paper are obtained using this ALNS-configuration.

5.3. Comparing the ALNS heuristic with a commercial solver

We now compare the ALNS to the solutions provided by the solver Gurobi 9.1.1 based on several performance indicators. For each instance group, we report the average objective value (**Obj.**) value across the five instances within an instance group. Note that this might refer to the optimal solution, if found for an instance within the runtime limit of 3600 seconds, or, otherwise, to the best integer feasible solution found up to that time. We also report the average lower bound (**LB**) provided by Gurobi for each test instance group and the average runtime measured in seconds. For the ALNS heuristic, we also observe the objective function value, next to the coefficient of variation of these values for the five solution runs per instance, and the solution time. Furthermore, we report a gap ($\mathbf{Gap}^{Obj.}$) that measures the relative difference in the objective function values achieved by Gurobi and the ALNS heuris-

Table 5
Comparison of results from running the ALNS and its extensions local search (LS) and Voyage Combination Problem (VCP). **CV** is the coefficient of variation averaged for all runs in each instance group. **Gap** is defined as the relative difference between the best solution from *any* configuration for each test instance in an instance group, and the averaged value from five runs of the same instance using a particular ALNS configuration. Average solution times for each instance are given in seconds.

Instance	ALNS			ALNS + LS			ALNS + VCP			ALNS + LS + VCP		
Group	CV	Gap	Time	CV	Gap	Time	CV	Gap	Time	CV	Gap	Time
5	0.00%	0.00%	0.9	0.00%	0.00%	1.4	0.00%	0.00%	1.0	0.00%	0.00%	1.6
7	0.00%	0.00%	2.5	0.00%	0.00%	4.1	0.00%	0.00%	1.1	0.00%	0.00%	4.5
9	0.00%	0.00%	4.5	0.00%	0.00%	7.5	0.00%	0.00%	5.5	0.00%	0.00%	8.3
11	0.00%	0.00%	18.9	0.00%	0.00%	42.0	0.00%	0.00%	23.2	0.00%	0.00%	50.3
13	0.15%	0.07%	32.3	0.00%	0.00%	72.2	0.15%	0.07%	47.1	0.00%	0.00%	84.0
15	0.16%	0.34%	37.2	0.02%	0.02%	76.1	0.10%	0.13%	56.7	0.00%	0.00%	91.1
17	0.21%	0.17%	56.7	0.03%	0.01%	149.5	0.19%	0.12%	95.5	0.00%	0.00%	167.7
19	0.20%	0.21%	70.9	0.04%	0.03%	189.6	0.15%	0.11%	132.0	0.00%	0.00%	224.3
21	0.29%	0.80%	84.8	0.20%	0.48%	174.2	0.21%	0.21%	161.2	0.13%	0.19%	195.2
23	0.45%	0.69%	131.7	0.17%	0.16%	333.0	0.28%	0.28%	225.7	0.02%	0.05%	363.9
25	0.26%	0.35%	141.8	0.04%	0.10%	391.5	0.20%	0.19%	273.6	0.03%	0.07%	442.0
27	0.80%	1.81%	141.8	0.58%	0.92%	304.0	0.36%	0.36%	223.8	0.23%	0.31%	387.8
Avg.	0.23%	0.51%	61.1	0.09%	0.20%	145.4	0.14%	0.17%	104.0	0.03%	0.07%	168.4

Table 6
Comparison of Gurobi (run for 3600 seconds per instance) and ALNS. **Obj.** represents average objective values, and **Gap**$^{Obj.}$ shows the relative change of the objective value between Gurobi and ALNS. The averaged lower bounds provided by Gurobi are shown by **LB**. **CV** is the coefficient of variation (relative standard deviation) and **Time** is the average solution time (in seconds) for each instance. All numerical values represent the average across all test instances in an instance group.

Instance group	Gurobi			ALNS heuristic			
	Obj.	*LB*	Time	*Obj.*	CV	Gap$^{Obj.}$	Time
5	2217.6	2217.6	5.9	2217.6	0.00%	0.00%	1.6
7	2095.0	2095.0	32.9	2095.0	0.00%	0.00%	4.5
9	5627.5	5627.6	48.5	5627.5	0.00%	0.00%	8.3
11	3517.5	3420.3	2417.2	3517.5	0.00%	0.00%	50.3
13	3994.7	3682.9	3567.7	3970.5	0.00%	-0.60%	84.0
15	12477.9	7296.0	3600.0	8375.5	0.00%	-32.88%	91.1
17	8567.2	3749.6	3600.0	4966.1	0.00%	-42.03%	167.7
19	8988.4	3994.8	3600.0	5153.5	0.00%	-42.66%	224.3
21	13325.4	7069.2	3600.0	9265.8	0.13%	-30.47%	195.2
23	-	4092.2	3600.0	5962.1	0.02%	-	363.9
25	-	4342.4	3600.0	6630.5	0.03%	-	442.0
27	-	4648.8	3600.0	8389.9	0.23%	-	387.8

tic, where a negative value means that the solution from the ALNS heuristic yields the lowest objective function value.

The results are summarized in Table 6. It can be seen that Gurobi can solve some instances with up to 13 installations to optimality within a runtime of one hour. Furthermore, feasible solutions are produced within the same time limit for all instances up to group 21, although the objective function values are potentially weak as is indicated by the large differences compared to the lower bounds. For larger instances, Gurobi cannot find feasible solutions within one hour (indicated by '-' in the table), except for one instance in the group with 23 installations. With regard to ALNS heuristic, we see that the heuristic clearly outperforms Gurobi. It achieves the same optimal solutions for the smaller instances and provides solutions of much better quality for the medium-sized instances with up to 21 installations. Savings in cost are as high as 30% to 42% per instance group. For the largest instances, the ALNS heuristic finds solutions within just a few minutes per instance. The corresponding lower bounds and the fact that the total cost of these solutions are comparable to those of the medium sized instances indicates that these are high-quality solutions. The coefficient of variation is very small in all cases, indicating that the performance of the ALNS heuristic is consistent.

5.4. Value of allowing speed optimization

The cost effects of allowing speed optimization can be evaluated by solving selected instances once with speed being fixed and identical for all vessels and once with the full flexibility of making speed decisions as in the previous experiments. For the fixed speed, we let vessels travel at their *design speed* of 12 knots for all legs of their routes, as this is consistent with the current practice in many real applications. We admit that lowering this fixed speed down to 10 knots, which is the lowest allowed speed, would lead to a further fuel reduction. At the same time, less serviced orders (more postponement of orders) would be observed, resulting in higher penalty cost. The design speed of 12 knots usually represents a reasonable trade-off between fuel efficiency and achieved order service level.

Table 7
The reductions in costs by performing speed optimization on five of the largest test instances, evaluated for each of the three weather scenarios.

	Weather scenario		
Instance	Fair	Mixed	Rough
19-21-3-2	24.01%	20.90%	18.52%
21-24-3-1	22.52%	19.18%	16.28%
23-27-4-1	22.87%	19.07%	16.32%
25-29-4-1	23.63%	19.95%	17.06%
27-32-5-1	22.41%	18.90%	16.78%
Average	23.09%	19.60%	16.99%

We randomly select five test instances among the largest instance groups, and solve these for each of the three weather scenarios presented in Section 5.1 both with speed optimization and without (i.e., using the service speed of 12 knots), and the corresponding results are presented in Table 7. For each instance and each weather scenario, the table gives the relative reduction of cost achieved by including speed decisions compared to using a fixed sailing speed on all sailing legs.

Table 8
The realized costs (**Costs**) and number of missed orders (**#MO**) of solutions to five large test instances when planning with and without weather forecasts. The impact of weather planning is evaluated for both the *Mixed* and the *Rough* weather scenario.

	Weather scenario *Mixed*				Weather scenario *Rough*			
	Assuming *Fair*		Weather planning		Assuming *Fair*		Weather planning	
Instance	**Costs**	**#MO**	**Costs**	**#MO**	**Costs**	**#MO**	**Costs**	**#MO**
19-21-3-2	10602.7	5	6062.9	0	15896.3	5	7035.5	0
21-24-3-1	12740.6	6	6595.7	0	18238.3	6	7648.2	0
23-27-4-1	9951.8	4	7052.7	0	11799.0	4	8100.5	0
25-29-4-1	11599.3	3	8277.9	0	9504.6	1	9445.6	0
27-32-5-1	18084.4	8	10807.5	0	20991.8	5	12308.5	0
Average	**12595.7**	**5.2**	**7759.4**	**0**	**15286.0**	**4.2**	**8907.7**	**0**

From the results in Table 7, significant reductions in overall costs are observed when speed optimization is allowed across all instances and weather scenarios. No clear correlation between the size of an instance and the cost reductions are found, however, it seems like the value of speed optimization decreases with worsened weather conditions. The speed-optimized voyages last longer due to lower average sailing speed, and by reviewing Fig. 7, we see that a later return time to the supply depot means longer periods of sailing in larger wave heights for the scenarios *Mixed* and *Rough*. This relative increase of the fuel consumption limits the benefits of the speed optimization given the particular weather input used for our study.

Although not presented here, we have also observed similar reductions in the relative CO_2 emissions for all test instances, as this can be easily calculated from the amount of fuel consumed, which is the largest contributor to the objective function value for all instances.

5.5. Value of considering weather forecasts

In addition to the large potential for cost reductions provided by speed optimization, the realized costs of scheduled voyages strongly depend on the experienced weather conditions and whether these are taken into account when planning the PSV operations.

To quantify the importance of what we term *weather planning*, the best solutions from the selected five large test instances for both the *Mixed* and the *Rough* weather scenario are compared to the performance of the schedules made by assuming *Fair* weather throughout the planning period. If fair weather is assumed but harsher weather conditions in fact are experienced, the actual fuel costs are underestimated as maintaining the scheduled speed will require more machinery power when sailing in higher waves. Furthermore, the highest sailing speeds might not be possible to maintain due to rough weather, causing delays to the schedules. If a vessel then arrives at an installation outside its opening hours, the order penalty cost applies to the objective function as prohibited service results in a *missed order* (MO).

Table 8 shows the results obtained from ignoring or considering weather conditions in the routing and scheduling of PSVs. We observe that for both weather scenarios considered, i.e., for *Mixed* and *Rough* weather, all occurrences of missed orders are avoided when the real weather conditions are taken into account in the planning. When the weather is mistakenly assumed to be without impact on the vessel's performance (*Fair* weather), several orders are not serviced due to delays in the realization of the scheduled voyages (i.e., when applying the schedule to the same test instance while considering the *actual* weather conditions). Eventually, when weather planning is applied, the average cost for the five instances are reduced by 38% and 45% for the *Mixed* and *Rough* weather scenarios, respectively. Put differently, to *not* account for weather when scheduling voyages can have costly consequences.

6. Concluding remarks

In this paper we have studied an operational planning problem arising in the offshore oil and gas industry, in which we determine routes, as well as sailing speeds along these routes, for a set of platform supply vessels (PSVs) servicing a given set of delivery and pickup orders at offshore installations such that costs are minimized. The sailing costs, mainly induced by fuel consumption for the PSVs, heavily depend on the chosen sailing speeds. Furthermore, the fuel consumption and the feasible speed ranges for the PSVs are largely affected by weather conditions that may vary over time. This results in a weather- or Time-Dependent Vessel Routing Problem with Speed Optimization (TDVRP-SO). Optional decisions include the postponement of certain orders and the chartering of spot vessels, both associated with additional costs.

A mixed integer programming (MIP) model for the TDVRP-SO has been formulated, defined on a time-space network. As the MIP solver is limited to solving instances with at most 16 orders for 13 offshore installations within a reasonable time limit of one hour, we have proposed an Adaptive Large Neighborhood Search (ALNS) heuristic for the TDVRP-SO. To improve initially constructed solutions, the ALNS heuristic applies an adaptive selection of destroy and repair heuristics where some are problem-specific heuristics based on the characteristics of offshore logistics planning. Additionally, two extensions are introduced to further improve the performance of the ALNS heuristic, namely local search and a set partitioning model that selects useful combinations of voyages from previous ALNS iterations. The ALNS heuristic also includes solving the sub-problem of determining the optimal sailing speeds along each route.

Our experiments on 60 test instances generated based on real data from the Norwegian continental shelf show that the ALNS heuristic is able to find optimal solutions in short computational times to all instances that could be solved to optimality by the MIP solver. The experiments also show that including speed optimization results in significant cost savings and CO_2 emission reductions of approximately 20%, compared to the more common approach of applying a fixed design speed for each voyage. Furthermore, three scenarios for forecasted wave heights were used to examine the impact of considering weather conditions to the planning. Our results highlight the importance of considering the correct weather conditions when scheduling voyages as, otherwise, the solutions may substantially underestimate the true cost of conducting the planned voyages under actual weather conditions. Ignoring weather conditions in the planning might even result in some orders not being serviceable, due to weather-induced delays in the actual schedule execution.

Acknowledgments

The authors would like to thank our contact persons at Equinor for providing valuable insight to the current practice of their offshore logistics operations, and for elaborating on the importance of considering the operational aspects presented in this article.

Appendix A. Parameter tuning for ALNS

The developed ALNS heuristic requires a proper setting of its parameters to deliver best-possible solutions. For this, we apply a systematic tuning approach for the key parameters to obtain appropriate values for them. To avoid overfitting, we generated an extra set of ten test instances, used solely for the parameter tuning, with 9 to 27 installations. Those parameters that were tuned are listed in Table 9. We started by solving the ten generated test instances using the initial values of the parameters. We then re-run while varying one parameter at a time, testing five different settings for each parameter. Having identified the best performing value of a parameter, this value was fixed and the process continued for the next parameter.

Table 9
Systematically tuned ALNS-parameters in the order they were tuned. Initial values inspired by Ropke & Pisinger (2006) and Liu, Tao, & Xie (2019).

Parameter	Initial Value	Final Value	Description
$\hat{q}$	[5%, 15%]	[15%, 50%]	Percentage of orders to remove, picked uniformly at random in the interval
σ_1	33	33	ALNS score for finding new globally optimal solution
σ_2	9	9	ALNS score for finding new locally optimal solution
σ_3	13	1	ALNS score for finding new solution
r	0.1	0.1	ALNS reaction parameter
η	0.250	0.025	ALNS noise control parameter
P	5	7	Determinism parameter

All further ALNS-parameters have been set through a trial-and-error process, which lead to the values shown in Table 10.

Appendix B. Arc generation procedure for the time-space network

See Algorithm 2

Algorithm 2: Arc-Generation Procedure.

Input : Set of available vessels, set of orders, start depot, end depot, vessel preparation end time, vessel return time
Output: Set of arcs

initialize the set of arcs as an empty set
for v in the set of all available vessels **do**
  for i in the set of all orders and the depot **do**
    for j in the set of all orders and the depot **do**
      if moving from i to j is not allowed **then**
        skip to next j
      end
      for t^s in the set of time points from preparation end to return time **do**
        if i is not the depot **and** $t^s <$ earliest arrival time at i from depot **then**
          skip to next t^s
        else if $t^s +$ min sailing duration from j to depot $>$ return time **then**
          skip to next t^s
        else if installation with order i is closed at t^s **then**
          skip to next t^s
        end
        calculate the set of possible arrival times t^{arr} at order or depot j
        if j is an order **then**
          calculate the servicing duration required for j
          calculate the servicing start times with no idling before start
          if servicing of order j cannot be performed without idling **then**
            calculate the first possible servicing start time
            add the time point to the set of servicing start times
          end
          for t^{ss} in the set of service start times **do**
            calculate end time t' by adding service duration to t^{ss}
            calculate the arc cost given the start, arrival, servicing, and end times
            add the arc $\big((i, t), (j, t')\big)$ and its arc cost to the set of arcs
          end
        else `// applies to the final sailing leg back to depot`
          calculate the arc costs for all arrival times
          set the end time t' equal to the arrival time with the cheapest cost
          add the arc $\big((i, t), (j, t')\big)$ and its arc cost to the set of arcs
        end
      end
    end
  end
end

Appendix C. Notation used to mathematically formulate the TDVRP-SO

Sets and Indices

$\mathcal{V}$	- set of available vessels v
$\mathcal{N}$	- set of orders i
$\mathcal{N}^{MD}$	- subset of $\mathcal{N}$ consisting of mandatory delivery orders
$\mathcal{N}^{OD}$	- subset of $\mathcal{N}$ consisting of optional delivery orders
$\mathcal{N}^{OP}$	- subset of $\mathcal{N}$ consisting of optional pickup orders
$\mathcal{A}_v$	- set of arcs $\big((i, t), (j, t')\big)$ for vessel v
$\mathcal{T}$	- set of all discrete time points, t, in the planning horizon
$\mathcal{T}^S_{ijv}$	- subset of $\mathcal{T}$ with start times for arcs between i and j for vessel v
$\mathcal{T}^{SS}_{ijtv}$	- subset of $\mathcal{T}$ with specific start times for arcs between i and j with end time t for vessel v
$\mathcal{T}^{SE}_{itjv}$	- subset of $\mathcal{T}$ with specific end times for arcs between i and j with start time t for vessel v

Parameters

S_i	- size of order i
Q_v	- maximum load capacity of vessel v
$C^A_{itjt'v}$	- fuel consumption and potential chartering cost for vessel v on arc $\big((i, t), (j, t')\big)$
C^P_i	- penalty cost of not servicing optional order i
o	- supply depot at the beginning of a voyage, modeled as an origin node
d	- supply depot at the end of a voyage, modeled as a destination node
t^*	- time at which vessel preparation ends

Table 10
Overview of further ALNS-parameter values.

Parameter	Value	Description
k	3	Regret parameter set for the regret insertion heuristic.
κ	0.2	Lower threshold for the adaptive weights.
ξ	$0.2\% \times T_{start}$	Simulated annealing cooling rate.
T_{start}	-	Simulated annealing temperature, set such that the probability of accepting a candidate solution is 50% if the candidate solution is less than 5% worse than the current solution.
I^{ALNS}	5000	Number of iterations for the ALNS heuristic throughout the conducted computational studies.
I^{S}	100	Number of iterations in one ALNS segment.
I^{VCP}	-	Number of iterations between each time the Voyage Combination Problem (VCP) is solved. The parameter is set to 200 for the first five completions, 500 for the next two, and finally set to 1000. Hence, the VCP is solved ten times within 5000 ALNS iterations.
β	20%	Maximum objective value gap between the current best solution, x, and the candidate solution x' in order to initiate the local search extension of the ALNS. In other words, we require $x' \leq (1+\beta)x$ for the local search in Section 4.5 to be applied.
γ	4	Number of discrete time points per hour. Time intervals of 15 minutes is shorter than the sailing time in 97.5% of the voyage legs when sailing in 14 knots, meaning that departure and arrival takes place at different points in time, inducing a sailing cost also for the shortest sailing legs.

Decision Variables

$x_{it\,jt'v}$ - $\begin{cases} 1, & \text{if arc}\big((i,t),(j,t')\big)\text{is used by vessel } v \\ 0, & \text{otherwise} \end{cases}$

u_{iv} - $\begin{cases} 1, & \text{if order } i \text{ is serviced by vessel } v \\ 0, & \text{otherwise} \end{cases}$

l^{D}_{iv} - delivery load on board vessel v *after* servicing order i

l^{P}_{iv} - pickup load on board vessel v *after* servicing order i

References

Aas, B., Gribkovskaia, I., Halskau, Ø., & Shlopak, A. (2007). Routing of supply vessels to petroleum installations. *International Journal of Physical Distribution & Logistics Management, 37*(2), 164–179.

Albjerk, N., Danielsen, T., Krey, S., Stålhane, M., & Fagerholt, K. (2016). A vessel pickup and delivery problem from the disruption management in offshore supply vessel operations. In *International conference on computational logistics* (pp. 50–64). Springer.

Andersson, H., Fagerholt, K., & Hobbesland, K. (2015). Integrated maritime fleet deployment and speed optimization: Case study from roro shipping. *Computers & Operations Research, 55*, 233–240.

Archetti, C., Speranza, M. G., & Vigo, D. (2014). Chapter 10: Vehicle routing problems with profits. In *Vehicle routing: Problems, methods, and applications, second edition* (pp. 273–297). SIAM.

Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., & Laporte, G. (2007). Static pickup and delivery problems: A classification scheme and survey. *Top, 15*(1), 1–31.

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

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

Cuesta, E. F., Andersson, H., Fagerholt, K., & Laporte, G. (2017). Vessel routing with pickups and deliveries: An application to the supply of offshore oil platforms. *Computers & Operations Research, 79*, 140–147.

Dabia, S., Ropke, S., Van Woensel, T., & De Kok, T. (2013). Branch and price for the time-dependent vehicle routing problem with time windows. *Transportation Science, 47*(3), 380–396.

Fagerholt, K., Laporte, G., & Norstad, I. (2010). Reducing fuel emissions by optimizing speed on shipping routes. *Journal of the Operational Research Society, 61*(3), 523–529.

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

Franceschetti, A., Demir, E., Honhon, D., Van Woensel, T., Laporte, G., & Stobbe, M. (2017). A metaheuristic for the time-dependent pollution-routing problem. *European Journal of Operational Research, 259*(3), 972–991.

Gendreau, M., Hertz, A., & Laporte, G. (1992). New insertion and postoptimization procedures for the traveling salesman problem. *Operations Research, 40*(6), 1086–1094.

Gribkovskaia, I., Laporte, G., & Shlopak, A. (2008). 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.

Halvorsen-Weare, E. E., & Fagerholt, K. (2011). Robust supply vessel planning. In *International conference on network optimization* (pp. 559–573). Springer.

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

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

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

Kisialiou, Y., Gribkovskaia, I., & Laporte, G. (2018b). Robust supply vessel routing and scheduling. *Transportation Research Part C: Emerging Technologies, 90*, 366–378.

Korsvik, J. E., Fagerholt, K., & Laporte, G. (2011). A large neighbourhood search heuristic for ship routing and scheduling with split loads. *Computers & Operations Research, 38*(2), 474–483.

Lianes, I. M., Noreng, M. T., Fagerholt, K., Slette, H. T., & Meisel, F. (2021). The aquaculture service vessel routing problem with time dependent travel times and synchronization constraints. *Computers & Operations Research, 134*, 105316.

Lindstad, H. E., Eskeland, G. S., & Rialland, A. (2017). Batteries in offshore support vessels–pollution, climate impact and economics. *Transportation Research Part D: Transport and Environment, 50*, 409–417.

Liu, R., Tao, Y., & Xie, X. (2019). An adaptive large neighborhood search heuristic for the vehicle routing problem with time windows and synchronized visits. *Computers & Operations Research, 101*, 250–262.

Ma, Z.-J., Wu, Y., & Dai, Y. (2017). A combined order selection and time-dependent vehicle routing problem with time widows for perishable product delivery. *Computers & Industrial Engineering, 114*, 101–113.

Malandraki, C., & Daskin, M. S. (1992). Time dependent vehicle routing problems: formulations, properties and heuristic algorithms. *Transportation Science, 26*(3), 185–200.

Norlund, E. K., & Gribkovskaia, I. (2013). Reducing emissions through speed optimization in supply vessel operations. *Transportation Research Part D: Transport and Environment, 23*, 105–113.

Norlund, E. K., & Gribkovskaia, I. (2017). Environmental performance of speed optimization strategies in offshore supply vessel planning under weather uncertainty. *Transportation Research Part D: Transport and Environment, 57*, 10–22.

Norlund, E. K., Gribkovskaia, I., & Laporte, G. (2015). Supply vessel planning under cost, environment and robustness considerations. *Omega, 57*, 271–281.

Norstad, I., Fagerholt, K., & Laporte, G. (2011). Tramp ship routing and scheduling with speed optimization. *Transportation Research Part C: Emerging Technologies, 19*(5), 853–865.

Norwegian Petroleum Directorate (2021). Recent activity. Website. https://www.norskpetroleum.no/en/developments-and-operations/recent-activity/.

Pisinger, D., & Ropke, S. (2007). A general heuristic for vehicle routing problems. *Computers & Operations Research, 34*(8), 2403–2435.

Ropke, S., & Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science, 40*(4), 455–472.

Shaw, P. (1998). Using constraint programming and local search methods to solve vehicle routing problems. In *International conference on principles and practice of constraint programming* (pp. 417–431). Springer.

Sopot, E., & Gribkovskaia, I. (2014). Routing of supply vessels to with deliveries and pickups of multiple commodities. *Procedia Computer Science, 31*, 910–917.

Stålhane, M., Albjerk, N., Danielsen, T., Krey, S., & Fagerholt, K. (2019). A variable neighbourhood search heuristic for disruption management in offshore oil and gas logistics. *Journal of the Operational Research Society, 70*(4), 588–600.