# A Vessel Pickup and Delivery Problem from the Disruption Management in Offshore Supply Vessel Operations

Nils Albjerk[1], Teodor Danielsen[1], Stian Krey[1], Magnus Stålhane[1(✉)], and Kjetil Fagerholt[1,2]

[1] Department of Industrial Economics and Technology Management, Norwegian University of Science and Technology, Trondheim, Norway
magnus.staalhane@iot.ntnu.no

[2] Norwegian Marine Technology Research Institute (MARINTEK), Otto Nielsens veg 10, 7052 Trondheim, Norway

**Abstract.** This paper considers a vessel pickup and delivery problem that arises in the case of disruptions in the supply vessel logistics in the offshore oil and gas industry. The problem can be modelled as a multi-vehicle pickup and delivery problem where delivery orders are transported by supply vessels from an onshore supply base (depot) to a set of offshore oil and gas installations, while pickup orders are to be transported from the installations back to the supply base (i.e. backload). We present both an arc-flow and a path-flow formulation for the problem. For the path-flow formulation we also propose an efficient dynamic programming algorithm for generating the paths, which represent feasible vessel voyages. It is shown through a computational study on various realistic test instances provided by a major oil and gas company that the path-flow model is superior with respect to computational performance.

**Keywords:** Disruption management · Offshore supply · Vehicle routing

## 1 Introduction

Norway is a major oil and gas producer with a total petroleum production of about 230 million $Sm^3$ (standard cubic meters) in 2015 [8]. This production takes place from offshore installations on the Norwegian continental shelf with about 60 oil and gas fields. To ensure continuous production, the offshore installations are supplied with different equipment and material by specialized offshore supply vessels (OSVs). The OSVs represent one of the largest cost elements in the upstream supply chain, where the annual costs of one OSV amount to millions of USDs.

The oil and gas companies operating on the Norwegian continental shelf usually have a long-term plan for supplying its offshore installations, where a set of voyages are to be sailed on a weekly basis by a given chartered fleet of OSVs.

© Springer International Publishing Switzerland 2016
A. Paias et al. (Eds.): ICCL 2016, LNCS 9855, pp. 50–64, 2016.
DOI: 10.1007/978-3-319-44896-1_4

A voyage performed by an OSV starts at the onshore supply base, then the OSV visits and services a set of offshore installations in a pre-determined sequence, before returning to the supply base. In addition to bringing all types of products that are needed to the offshore installations, the OSVs also carry backload from the installations to the supply base. Each voyage is scheduled to take two or three days and each OSV usually completes two or three voyages each week. Figure 1 illustrates a weekly plan where three OSVs are scheduled to visit five installations.

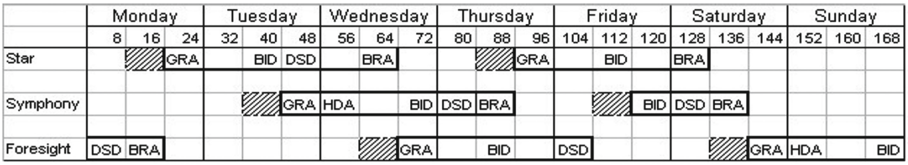


**Fig. 1.** Illustration of a weekly plan including three OSVs (Star, Symphony and Foresight) and five offshore installations (GRA, BID, DSD, BRA, HAD). Each voyage is represented by a rectangle, where the shaded area represents the time at the supply base [10].

The offshore installations are located in a part of the world where weather conditions can be harsh, especially during the winter season. Sometimes wave heights may limit both an OSV's sailing speed and its ability to perform unloading/loading operations at the installations. Another major source of disruptions to the plan comes from unexpected orders or extra high demand from the offshore installations, which especially occurs after periods with bad weather where the installations have not been serviced for some days. To mitigate the effects of these disruptions, the planners may have to deviate from the planned voyages and re-route the OSVs. They also have the possibility to charter an extra OSV from the spot market to handle the disruptions, though at a very high cost.

In this paper we study the problem of how to determine the OSV voyages for the next days after a disruption has occurred. The goal is to return to the long-term plan before the next voyage is planned to start for each OSV. Using the example from Fig. 1, suppose that the planner on Monday morning receive reports saying that the weather will be bad for the next two days, resulting in increased sailing and service times at the offshore installations. The planner then needs to determine how to adjust the next voyage for the OSVs Star (starting on Monday) and Symphony (starting on Tuesday) so that they hopefully can start on their next voyages on Thursday and Friday, respectively. These decisions affect both the service level perceived by the offshore installations, in case of delays in their services, and the sailing and chartering costs. Hence, the objective of the problem is to minimize these costs, while at the same time maintain a sufficient service level to the offshore installations and avoid delays of the OSVs that cause knock-on effects to the long-term plan.

Several papers address routing in the upstream supply chain for the offshore oil and gas industry. [5] study a pickup and delivery problem that arises in the service of offshore installations in the Norwegian Sea. Unlike the problem studied in this paper, they consider the routing of only one OSV. [11] extend the problem by taking into account demands for multiple commodities and the stowage of these commodities in dedicated compartments onboard the OSVs. They present a mathematical model of the problem and a heuristic to provide high quality solutions in a short amount of time.

As we can see in Sect. 2, the problem studied in this paper can be modelled as a multi-vehicle pickup and delivery problem where the delivery orders are transported from the supply base and the pickup orders (i.e. backload) are returned to the depot. The offshore installations might be visited once or twice, either conducting pickup and delivery simultaneously or at different points in time, possibly by different OSVs. Using the classification proposed by [1], our problem can be viewed as a special version of the 1-M-1|P-D|m, i.e. a one-to-many-to-one pickup and delivery problem with multiple vehicles. Problems of this type have been studied in the literature, though in very different contexts than ours. [3] studies a problem arising in reverse logistics, and, as this paper, propose both an arc-flow and path-flow model for the problem. [9] studies the same problem, but also takes into account time limits on the vehicles. However, in contrast to our problem neither allow customers to be visited twice and they do not consider the possibility of chartering an extra vessel/vehicle.

Relatively few studies regarding disruption management in ship routing exist, and to the authors' knowledge there exist no publications on disruption management in offshore supply logistics. However, in container liner shipping there exist a few studies, such as [2], which consider the vessel schedule recovery problem. Different recovery actions are proposed in the case of disruptions, such as increasing speed, canceling deliveries, and swapping port visits. A model considering sailing costs, delays, and misplaced cargo is presented, and it is run with data from real life cases. [7] proposes a mathematical model for simultaneous rescheduling of ships and cargo in a container liner network. Poor weather conditions, port congestion, low port productivity, towage, tidal windows, and several other sources of disruptions are mentioned. The model's suggested recovery actions include changing the departure or arrival time at ports, transshipment of cargo between ports, and speed adjustments. The possible measures to handle disruptions in [2,7] are to some extent the same as the ones available in our problem, such as canceling orders and re-routing. However, increasing speed to reduce delays will be less effective in our problem due to shorter sailing distances and is therefore not included, while the possibility of chartering additional OSVs from the spot market is an additional option available in offshore supply.

The contribution of this paper is to propose and test two mathematical models of the problem, i.e. an arc-flow and a path-flow model. For the path-flow model we also propose an efficient dynamic programming algorithm for generating the paths (feasible OSV voyages). It is shown that the path-flow model is superior to the arc-flow model with regards to computational performance.

Section 2 provides a formal description of the problem together with an arc-flow and a path-flow model of the problem. The dynamic programming algorithm for generating all feasible paths are presented in Sect. 3, while computational experiments are shown in Sect. 4. Finally, some concluding remarks are given in Sect. 5.

## 2 Problem Description and Mathematical Models

The problem studied in this paper can be formulated on a graph $G = (\mathcal{N}, \mathcal{A})$. At a given point in time there are $n$ cargoes that must be transported from the onshore depot to different offshore installations, while at the same time $m$ cargoes need to be transported from (possibly different) installations and back to the onshore depot. The set of nodes $\mathcal{N} = \{0, ..., n+m+1\}$ contains two nodes representing the onshore supply depot (nodes 0 and $n+1$) and one node for each cargo to be picked up or delivered. The set $\mathcal{N}$ can be divided into the set $\mathcal{N}^P = \{1, ..., n\}$ containing all pickup nodes, and the set $\mathcal{N}^D = \{n+2, ..., n+m+1\}$ containing all delivery nodes. We use the term *sibling nodes* to denote a pickup and a delivery node that is associated with the same offshore installation. The set of arcs $\mathcal{A}$ consists of arcs between all node pairs, with the following exceptions: there are no arcs entering node 0, no arcs leaving node $n+1$, and for sibling nodes there are no arcs from the pickup node to the delivery node since it is always preferable to deliver before picking up at an installation.

Each cargo occupies a given number of square meters on the deck of an OSV, and for each node $i \in \mathcal{N}^D$ we denote this area $D_i$, while for each node $i \in \mathcal{N}^P$ we denote it $P_i$. In addition, a penalty cost $C_i^R$ is incurred if node $i \in \mathcal{N}$ cannot be serviced on the planned upcoming set of voyages and must be postponed. This penalty can be set differently for each cargo depending on the importance of its delivery or pickup.

To transport the cargoes, a set of OSVs is available. Let $\mathcal{V} = \{1, ..., k+1\}$ denote the set of OSVs, where $k+1$ represents an OSV chartered from the spot market. Each OSV $v$ has a total deck capacity $Q_v$ measured in square meters, and a cost $C_{vij}^S$ and time $T_{vij}$ associated with sailing from node $i$ to node $j$, and servicing node $j$. Note that both of these parameters are weather dependent, however, we assume that they are known at the time of planning. Since we are only planning the next voyage, i.e. the next couple of days, this is a reasonable assumption. For example, if we know that the weather will be bad in the next couple of days we adjust $C_{vij}^S$ and $T_{vij}$ accordingly. Further, let $T_v^{MIN}$ be the time OSV $v$ is available to begin the next voyage, and let $T_v^{MAX}$ be the planned departure time of the subsequent voyage for OSV $v$. However, we do allow the OSV to return back to the depot up to $\Gamma$ hours after $T_v^{MAX}$, but at a cost of $C_v^D$ per hour. The OSV $k+1$ must be chartered for a whole number of time periods (days), where the length of a time period is denoted by the parameter $H$, and the daily time charter rate is represented by $C^{TC}$.

### 2.1 Arc-Flow Model

The variable $x_{vij}$ equals 1 if OSV $v$ sails arc $(i, j)$, and 0 otherwise. The auxiliary variable $y_{vi}$ equals 1 if OSV $v$ visits node $i$, and 0 otherwise. If the visit to node $i$

is postponed (i.e. not serviced on the voyage of any of the OSVs), the variable $u_i$ equals 1, and 0 otherwise. The cargo load variables $l_{vij}$ equal the load measured in square meters on OSV $v$ when sailing arc $(i, j)$. If the arc is not traversed, the corresponding load variable is equal to 0. The number of hours OSV $v$ arrives at the depot after $T_v^{MAX}$ is represented by the variable $t_v^D$. The number of whole days that the OSV $k+1$ from the spot market needs to be chartered is denoted by $t^{TC}$. To simplify the notation, the constraints are defined using sets of nodes, even though some constraints may contain combinations of the indices $v$, $i$, and $j$ for which the corresponding variable $x_{vij}$ does not exist. In these cases the missing variable can be assumed to take the value 0. The operational planning and disruption management problem can be formulated as follows:

**Objective**

$$\min \sum_{v \in \mathcal{V}} \sum_{(i,j) \in \mathcal{A}} C_{vij}^S x_{vij} + C^{TC} t^{TC} + \sum_{i \in \mathcal{N}} C_i^R u_i + \sum_{v \in \mathcal{V} \setminus \{k+1\}} C_v^D t_v^D \tag{1}$$

**subject to**:

$$\sum_{i \in \mathcal{N} \setminus \{0\}} x_{v0i} = 1 \qquad v \in \mathcal{V} \tag{2}$$

$$\sum_{i \in \mathcal{N} \setminus \{n+1\}} x_{vi(n+1)} = 1 \qquad v \in \mathcal{V} \tag{3}$$

$$\sum_{j \in \mathcal{N}} x_{vji} - \sum_{j \in \mathcal{N}} x_{vij} = 0 \qquad v \in \mathcal{V}, i \in \mathcal{N} \setminus \{0, n+1\} \tag{4}$$

$$y_{vi} - \sum_{j \in \mathcal{N} \setminus \{i\}} x_{vij} = 0 \qquad v \in \mathcal{V}, i \in \mathcal{N} \tag{5}$$

$$\sum_{v \in \mathcal{V}} y_{vi} + u_i = 1 \qquad i \in \mathcal{N} \setminus \{0, n+1\} \tag{6}$$

$$l_{vij} \leq (Q_v - P_j) x_{vij} \qquad v \in \mathcal{V}, i \in \mathcal{N}, j \in \mathcal{N}^P \tag{7}$$

$$l_{vij} \leq Q_v x_{vij} \qquad v \in \mathcal{V}, i \in \mathcal{N}, j \in \mathcal{N}^D \tag{8}$$

$$l_{vij} \geq P_i x_{vij} \qquad v \in \mathcal{V}, i \in \mathcal{N}, j \in \mathcal{N}^P \tag{9}$$

$$l_{vij} \geq D_j x_{vij} \qquad v \in \mathcal{V}, i \in \mathcal{N}^D, j \in \mathcal{N}^D \tag{10}$$

$$l_{vij} \geq (P_i + D_j) x_{vij} \qquad v \in \mathcal{V}, i \in \mathcal{N}^P, j \in \mathcal{N}^D \tag{11}$$

$$\sum_{i \in \mathcal{N}} l_{vij} + P_j x_{vjh} - l_{vjh} + Q_v x_{vjh} \leq Q_v \qquad v \in \mathcal{V}, j \in \mathcal{N}^P, h \in \mathcal{N} \tag{12}$$

$$\sum_{i \in \mathcal{N}} l_{vij} - D_j x_{vjh} - l_{vjh} + Q_v x_{vjh} \leq Q_v \qquad v \in \mathcal{V}, j \in \mathcal{N}^D, h \in \mathcal{N} \tag{13}$$

$$\sum_{j \in \mathcal{N}^D} D_j y_{vj} - l_{v0i} + Q_v x_{v0i} \leq Q_v \qquad v \in \mathcal{V}, i \in \mathcal{N} \tag{14}$$

$$l_{vi(n+1)} - \sum_{j \in \mathcal{N}^P} P_j y_{vj} + Q_v x_{vi(n+1)} \leq Q_v \qquad v \in \mathcal{V}, i \in \mathcal{N} \tag{15}$$

$$t^{TC} \geq \left( \sum_{(i,j) \in \mathcal{A}} T_{(k+1)ij} x_{(k+1)ij} \right) \frac{1}{H} \tag{16}$$

$$T_v^{MIN} + \sum_{(i,j) \in \mathcal{A}} T_{vij} x_{vij} - T_v^{MAX} \leq t_v^D \qquad v \in \mathcal{V} \backslash \{k+1\} \tag{17}$$

$$t_v^D \leq \Gamma \qquad v \in \mathcal{V} \backslash \{k+1\} \tag{18}$$

$$\sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{S}} x_{vij} \leq |\mathcal{S}| - 1 \qquad v \in \mathcal{V}, \mathcal{S} \subset \mathcal{N}, |\mathcal{S}| \geq 2 \tag{19}$$

$$x_{vij} \in \{0,1\} \qquad v \in \mathcal{V}, (i,j) \in \mathcal{A}, \tag{20}$$

$$y_{vi} \in \{0,1\} \qquad v \in \mathcal{V}, i \in \mathcal{N} \tag{21}$$

$$u_i \in \{0,1\} \qquad i \in \mathcal{N} \tag{22}$$

$$t_v^D \geq 0 \qquad v \in \mathcal{V} \backslash \{k+1\} \tag{23}$$

$$t^{TC} \in \mathbb{Z}^+ \tag{24}$$

The objective function (1) consists of four parts. The first term summarizes the costs related to sailing and servicing nodes for all OSVs, while the second term expresses the cost related to chartering an OSV from the spot market. The third and fourth terms are artificial costs that penalize orders that are postponed and OSVs that return to the onshore supply depot later than planned. Constraints (2) and (3) ensure that all voyages begin and end at the depot, while constraints (4) conserve the flow through the problem defining network. The auxiliary variables are set by constraints (5), and constraints (6) ensure that all nodes are either serviced by an OSV or the visit is postponed until a later voyage. Further, constraints (7)–(11) ensure that the capacity of each OSV is not violated on any arc along its route, while constraints (12) and (13) are the cargo flow conservation constraints. Since the model does not distinguish between cargo that is to be delivered to an installation and backload, constraints (14) ensure that the total amount of cargo to be delivered to installations on a voyage equals the load on-board when the OSV leaves the depot. Similarly, constraints (15), together with constraints (7) and (8), ensure that the load on-board, when the OSV arrives at the depot, equals the total amount of picked up cargo on a voyage. If an OSV is chartered from the spot market, constraint (16) calculates the time it is used, and rounds up to the nearest whole day. Constraints (17) calculate the delay of each OSV when returning to the depot, while constraints (18) assure that the delay is not more than $\Gamma$ hours for each OSV in the long-term fleet. Finally, constraints (19) are the subtour eliminating constraints, and constraints (20)–(24) define the domain of each set of variables.

### 2.2 Path-Flow Model

Arc-flow models are well suited to describe a problem, however, they often perform inferior to path-flow models due to the large number of constraints and a relatively weak linear programming bound. In this section we describe a path-flow model obtained by applying Dantzig-Wolfe decomposition to the Arc-flow model presented above. A path through the graph $G$ for vessel $v$ is considered feasible if it satisfies constraints (2)–(5), (7)–(21), (23) and (24).

Let the set $\mathcal{R}_v$ contain all feasible paths for OSV $v$, and let the parameter $A_{ri}$ be equal to 1 f node $i$ is included on path $r$, and 0 otherwise. The cost associated with sailing and servicing all nodes on path $r$ for OSV $v$ is denoted by $C^S_{vr}$. This includes the sailing costs ($C^S_{vij}$) and any penalty costs from returning to the depot late ($C^D_v$). In addition, let $C^S_{(k+1)r}$ for all paths $r$ associated with the OSV chartered from the spot market include the time charter costs $C^{TC}$, in addition to the fixed charter costs. Further, let variable $\lambda_{vr}$ equal 1 if OSV $v$ use path $r$, and 0 otherwise. As for the arc-flow model, if the visit to node $i$ is postponed, the variable $u_i$ equals 1, and 0 otherwise. Using this notation the path-flow model can be formulated as follows:

$$min \sum_{v \in \mathcal{V}} \sum_{r \in \mathcal{R}_v} C^S_{vr} \lambda_{vr} + \sum_{i \in \mathcal{N}} C^R_i u_i \tag{25}$$

subject to:

$$\sum_{v \in \mathcal{V}} \sum_{r \in \mathcal{R}_v} A_{ri} \lambda_{vr} + u_i = 1 \qquad i \in \mathcal{N} \backslash \{0, n+1\} \tag{26}$$

$$\sum_{r \in \mathcal{R}_v} \lambda_{vr} \leq 1 \qquad v \in \mathcal{V} \tag{27}$$

$$\lambda_{vr} \in \{0, 1\} \qquad v \in \mathcal{V}, r \in \mathcal{R}_v \tag{28}$$

$$u_i \in \{0, 1\} \qquad i \in \mathcal{N} \tag{29}$$

The objective function (25) corresponds to (1) for the arc-flow model, and sums the costs related to the voyages for all OSVs and the costs associated with nodes that are postponed until a later voyage. Constraints (26) ensure that all nodes are either serviced by an OSV or postponed until a later voyage, while constraints (27) ensure that each OSV sails at most one voyage. Finally, constraints (28) and (29) put binary restrictions on the variables.

## 3 Path Generation Using Dynamic Programming

In this section we describe how we generate all feasible paths through graph $G$ which are needed to solve the path-flow model described in Sect. 2.2. All paths are generated for each OSV $v$ through $|\mathcal{N}|$ *stages*, where $|\mathcal{N}|$ is the number of nodes in the network. The chosen approach applies full enumeration of possible paths with removal of infeasible and dominated paths.

Algorithm 1 shows the pseudocode for the generation of paths, and is based on the labeling algorithm described in [6]. In this approach all partial paths are encoded using labels which stores each (partial) path and the accumulation of resources along the path. Let $M_p$ be the set of all labels representing paths of length $p$. The algorithm begins by creating an initial label representing a path starting at the depot node, and an empty set of labels representing complete feasible paths $\mathcal{R}$. Then, while $p$ is less than or equal to the number of nodes in $G$, we create new labels $L'$ by extending all labels $L$ representing a path of length $p$ to all nodes $i \in \mathcal{N}$. If the label $L'$ is feasible it is added to the set $M_{p+1}$, unless it is extended to the depot node, in which case it is added to the set $\mathcal{R}$. Once all labels in $M_p$ have been extended to labels in $M_{p+1}$ we check all pairs of labels in $M_{p+1}$ to see whether we can remove some labels due to dominance, and the counter $p$ is updated. Finally, all labels in $\mathcal{R}$ are returned, and their corresponding paths are added to the path-flow model.

In the following we explain what data is stored in a label, how a label is extended, what constitutes a feasible extension of a label, and under what circumstances we can say that one label dominates another.

**Algorithm 1.** Pseudocode for dynamic voyage generation

**procedure** VOYAGEGENERATOR<br>
    Create *initial label*<br>
    Add *initial label* to *initial stage* $M_1$<br>
    $\mathcal{R} = \emptyset$<br>
    $p = 1$<br>
    **while** $p \leq |\mathcal{N}|$ **do**<br>
        **for** all *labels* $L$ in *stage* $M_p$ **do**<br>
            **for** all nodes $i$ in $\mathcal{N}$ **do**<br>
                Create new label $L'$ by extending label $L$ to node $i$<br>
                **if** $L'$ is feasible **then**<br>
                    **if** $i = n + 1$ **then**<br>
                        Add $L'$ to $\mathcal{R}$<br>
                    **else**<br>
                        Add $L'$ to $M_{p+1}$<br>
                    **end if**<br>
                **end if**<br>
            **end for**<br>
        **end for**<br>
        Remove all dominated states from $M_{p+1}$<br>
        $p = p + 1$<br>
    **end while**<br>
    Return $\mathcal{R}$<br>
**end procedure**

### 3.1 Label Data

In each stage, new labels are created containing the following data:

- $i$ - The current node
- $R$ - The predecessor label
- $V$ - The set of nodes visited
- $C$ - The sailing and service cost
- $T$ - The sailing and service time
- $\pi^D$ - The maximum deck capacity occupied at any point along the path
- $\pi^P$ - The deck capacity needed for backload

In the following we use $i(L)$ to denote the current node $i$ for label $L$ and similarly use $R(L)$, $V(L)$, $C(L)$, $T(L)$, $\pi^D(L)$ and $\pi^P(L)$ for the rest of the label data.

The initial label represents an OSV starting at the depot pickup node. The initial state has no predecessor, which is denoted by $null$.

$$L_0 = \{0,\ null,\ \emptyset,\ 0,\ 0,\ 0,\ 0\},$$

### 3.2 Label Extension

When extending a label along an arc $(i(L),\ j)$, a new label $L'$ is created at node $j$. The label data are updated as follows:

$$i(L') = j \tag{30}$$

$$R(L') = L \tag{31}$$

$$V(L') = V(L) \bigcup \{j\} \tag{32}$$

$$T(L') = T(L) + T^S_{vi(L)j} \tag{33}$$

$$C(L') = C(L) + C^S_{vi(L)j} + \begin{cases} max\{C^D_v * (T(L') + T^{MIN}_v - T^{MAX}_v), 0\}, & \text{if } j = n+1 \\ 0, & \text{otherwise} \end{cases} \tag{34}$$

$$\pi^D(L') = \begin{cases} max\{\pi^D(L) + D_j, \pi^P(L)\}, \text{ if } j \in \mathcal{N}^D \\ max\{\pi^D(L), \pi^P(L) + P_j\}, \text{ if } j \in \mathcal{N}^P \end{cases} \tag{35}$$

$$\pi^P(L') = \begin{cases} \pi^P(L), \text{ if } j \in \mathcal{N}^D \\ \pi^P(L) + P_j, \text{ if } j \in \mathcal{N}^P \end{cases} \tag{36}$$

Equations (30) and (31) update the current node and the predecessor for $L'$. The new current node is marked as visited in Eq. (32). The time and cost data are updated in Eqs. (33) and (34). The capacity data are updated in Eqs. (35) and (36) according to whether node $j$ is a delivery or a pickup node. Figure 2 illustrates how the capacity data are updated along a path.

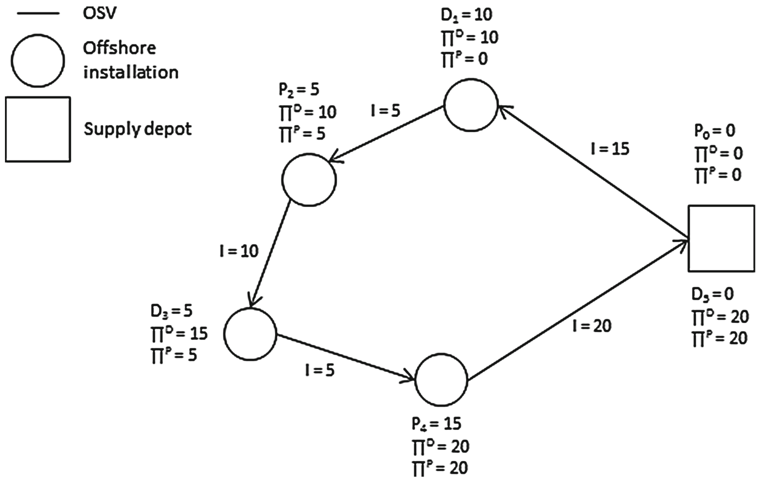


**Fig. 2.** Illustration showing how capacity data are updated during label extension. Requested delivery and backload size is given for each delivery and pickup node, respectively. The current amount of cargo carried on deck of the OSV is given for each arc.

**Proposition 1.** *Let $L_f$ be the final label associated with a path $r$. Then $\pi^D(L_f)$ equals the maximum load onboard the OSV on path $r$.*

*Proof.* Let the maximum load on path $r$ be carried on arc $(i^*,\ j^*)$. Then the maximum load on $r$ is

$$l_{i^*j^*} = \sum_{j\in\Omega^P} P_j + \sum_{j\in\Theta^D} D_j,$$

where $\Omega^P \subseteq \mathcal{N}^P$ is the set of pickup nodes already visited and $\Theta^D \subseteq \mathcal{N}^D$ is the set of delivery nodes not yet visited.

Consider the labels $L_1$ and $L_2$ with current nodes $i_1$ and $i_2$, respectively. $L_1$ is the predecessor of $L_2$, and both labels are predecessors of $L_f$. Let $\mathcal{N}_f^D$ denote the set of delivery nodes in the path of $L_f$. Assume, without the loss of generality, that the nodes visited on path $r$ are numbered in the sequence they are visited. Then, the load on deck after visiting node $i_1$ and $i_2$, denoted by $l_{i_1}$ and $l_{i_2}$, respectively, is

$$l_{i_1} = \sum_{j\in\mathcal{N}_f^D} D_j - \sum_{0\leq j\leq i_1} D_j + \sum_{0\leq j\leq i_1} P_j, \text{ and} \tag{37}$$

$$l_{i_2} = \sum_{j\in\mathcal{N}_f^D} D_j - \sum_{0\leq j\leq i_2} D_j + \sum_{0\leq j\leq i_2} P_j. \tag{38}$$

Assume that either $l_{i_1}$ or $l_{i_2}$ is the maximum load on $r$. The difference in load is

$$l_{i_1} - l_{i_2} = \sum_{i_1 < j \leq i_2} P_j - \sum_{i_1 \leq j \leq i_2} D_j \tag{39}$$

If $l_{i_1} > l_{i_2}$, then

$$\pi^D(L_2) = max\{\pi^D(L_1) + \sum_{i_1 \leq j \leq i_2} D_j, \pi^P(L_1) + \sum_{i_1 < j \leq i_2} P_j\}$$

$$= \pi^D(L_1) + \sum_{i_1 \leq j \leq i_2} D_j, \text{ and}$$

$$l_{i^* j^*} = \sum_{j \in \mathcal{N}_f^D} D_j - \sum_{0 \leq j \leq i_1} D_j + \sum_{0 \leq j \leq i_1} P_j$$

If $l_{i_2} > l_{i_1}$, then

$$\pi^D(L_2) = max\{\pi^D(L_1) + \sum_{i_1 \leq j \leq i_2} D_j, \pi^P(L_1) + \sum_{i_1 < j \leq i_2} P_j\}$$

$$= \pi^P(L_1) + \sum_{i_1 < j \leq i_2} P_j, \text{ and}$$

$$l_{i^* j^*} = \sum_{j \in \mathcal{N}_f^D} D_j - \sum_{0 \leq j \leq i_2} D_j + \sum_{0 \leq j \leq i_2} P_j$$

Not depending on which of $l_{i_1}$ and $l_{i_2}$ is greatest, the maximum load can be expressed as

$$l_{i^* j^*} = \sum_{j \in \Omega^P} P_j + \sum_{j \in \Theta^D} D_j = \pi^D(L_f).$$

Thus, Proposition 1 is correct.

Additional remark: if node $i^*$ were a delivery node, then the load on board before visiting $i^*$ would be greater than $l_{i^* j^*}$. Thus, $i^*$ is a pickup node. Similar, if $j^*$ were a pickup node, then the load on board after visiting $j^*$ would be greater than $l_{i^* j^*}$. Thus, $j^*$ is a delivery node.

### 3.3 Feasible Extension

An extension of state $L$ to state $L'$ along an arc $(i(L), j)$ is feasible if the following hold:

$$j \notin V(L) \tag{40}$$

$$k \notin V(L), \text{ if } j \in \mathcal{N}^D \text{ and } j \text{ and } k \text{ are siblings.} \tag{41}$$

$$T(L) + T^S_{vi(L)j} \leq T^{MAX}_v + \Gamma - T^{MIN}_v, \text{ if } j = n+1 \tag{42}$$

$$T(L) + T^S_{vi(L)j} + T^S_{vj(n+1)} \leq T^{MAX}_v + \Gamma - T^{MIN}_v, \text{ if } j \neq n+1 \tag{43}$$

$$max\{\pi^D(L) + D_j, \pi^P(L)\} \leq Q_v, \text{ if } j \in \mathcal{N}^D \tag{44}$$

$$max\{\pi^D(L), \pi^P(L) + P_j\} \leq Q_v, \text{ if } j \in \mathcal{N}^P \tag{45}$$

If the inequality (40) holds, node $j$ is not already visited in path in label $L$. Inequality (41) assures that the pickup node should never be visited before the delivery node for the same installation. A *sibling* of a delivery node is defined as the corresponding pickup node of the installation, if one exists. Likewise, the sibling of a pickup node is the delivery node of the installation. As stated in inequality (42), the time needed to service the nodes in a path sailed by OSV $v$ should not exceed the time available to $v$, where $\tau$ is the maximum allowed delay on a voyage. If node $j$ is not the depot delivery node, there should also be enough time available to sail back to the onshore supply depot. This is assured by inequality (43). The inequalities (44) and (45) hold if the load on deck at any point of the path does not exceed the available deck capacity of the OSV.

### 3.4 Label Domination

At each stage, all labels dominated by anoter label are identified and removed. The label dominance criteria are:

**Proposition 2.** *The label $L_1$ dominates $L_2$ if*

$$V(L_1) = V(L_2), \tag{46}$$

$$i(L_1) = i(L_2), \tag{47}$$

$$C(L_1) \leq C(L_2), \tag{48}$$

$$T(L_1) \leq T(L_2). \tag{49}$$

Note that $V(L_1) = V(L_2)$ also implies that $\pi^P(L_1) = \pi^P(L_2)$ and $\pi^D(L_1) = \pi^D(L_2)$. Because the cost and time and non-decreasing and separable resources, it can easily be shown that Proposition 2 is a valid dominance criterion.

## 4 Computational Study

In this section we present a comparison of the computational performance of the two models. In Sect. 4.1 we describe the test instances used, while in Sect. 4.2 we present the computational results.

### 4.1 Test Instances

The test instances are based on data supplied by Statoil, the major Norwegian oil and gas company, and consist of an onshore supply depot and a set of offshore installations. As of today, each onshore supply depot services up to 13 offshore installations (28 nodes). However, when looking at recovery planning, it is rare that all the installations are to be visited within the short planning horizon. We have thus created instances with 4 to 8 installations (10 to 18 nodes). These instances are summarized in Table 1, which gives the id of each instance (ID), the name of the associated depot (Depot), the number of nodes (# Nodes), and the number of OSVs (# OSVs).

Three versions of each instance are tested:

- **No disruptions** All vessels have normal sailing speeds and all cargoes are of (roughly) normal size. These instances are denoted using the standard IDs from Table 1.
- **Reduced sailing speed** due to adverse weather conditions. This is done by reducing the speed of each vessel from ten to five knots, and thus, increasing the sailing time. The increase in sailing time will also affect the sailing costs. The ID for these instances have a superscript "S" for speed, e.g. $M_{12}^S$.
- **Large cargo sizes**, which often is the case after periods where adverse weather conditions have made supplying the offshore installations impossible or too costly. This is done by setting the demand and backload amount to the triple of the size used in the non-disrupted case to ensure that deck capacity becomes a binding constraint. The ID for these instances have a superscript "L" for load, e.g. $M_{12}^L$.

The reason for these choices of disruptions is that both speed reductions and more cargo to transport are the most common consequences of bad weather conditions. Either the speed of each vessel is reduced for a period of time, or the cargo has piled up both at the depot and at the installations because the OSVs have been prevented from sailing for a few days.

**Table 1.** Test instances used to compare the arc-flow and the path-flow models. #OSVs includes one spot vessel.

| ID | Depot | # Nodes | # OSVs |
|---|---|---|---|
| $M_{10}$ | Mongstad | 10 | 2 |
| $M_{12}$ | Mongstad | 12 | 2 |
| $Å_{14}$ | Ågotnes | 14 | 2 |
| $F_{16}$ | Florø | 16 | 2 |
| $Å_{18}$ | Ågotnes | 18 | 3 |

### 4.2 Test Results

The arc-flow and path-flow models have been tested on a computer running Windows 7 with an Intel i7-3770 3.40 GHz CPU and 16 GB of RAM. The dynamic programming algorithm for the apriori generation of paths for the path-flow model is implemented in Java, while the arc-flow and path-flow models are implemented in Xpress-IVE 1.24.04 with Xpress-Mosel 3.6.0 and solved with Xpress-Optimizer 21.01.04 [4]. For both models we set an upper time limit of one hour (3,600 s).

The results of the computational tests can be seen in Table 2. For the arc-flow model the optimality gap (Opt. gap) and the computational time (Time) are given, while for the path-flow model we only give the computing time (Time).

**Table 2.** Comparison of the arc-flow and path-flow solution methods. For the arc-flow model the optimality gap (Opt. gap) and the solution time is presented and for the path-flow model the total solution time, including the time it takes to generate all paths, is presented.

| ID | Arc-flow | | Path-flow |
|---|---|---|---|
| | Opt. gap | Time [s] | Time [s] |
| $M_{10}$ | 0.0 % | 0.1 | 0.13 |
| $M_{10}^S$ | 0.0 % | 7.2 | 0.06 |
| $M_{10}^L$ | 0.0 % | 3.0 | 0.03 |
| $M_{12}$ | 0.0 % | 0.4 | 0.51 |
| $M_{12}^S$ | 68.9 % | >3 600 | 0.38 |
| $M_{12}^L$ | 0.0 % | 709 | 0.04 |
| $Å_{14}$ | 0.0 % | 0.6 | 13.1 |
| $Å_{14}^S$ | 63.3 % | >3 600 | 13.0 |
| $Å_{14}^L$ | 38.7 % | >3 600 | 0.56 |
| $F_{16}$ | 0.0 % | 0.4 | 529.3 |
| $F_{16}^S$ | 77.0 % | >3 600 | 553.4 |
| $F_{16}^L$ | 46.4 % | >3 600 | 101.3 |
| $Å_{18}$ | 69.6 % | >3 600 | >3 600 |
| $Å_{18}^S$ | 94.8 % | >3 600 | >3 600 |
| $Å_{18}^L$ | 68.6 % | >3 600 | 1 222 |

The reason for this, is that in the instances where the path-flow model exceeds the time limit it does so while generating paths, and thus we do not have any optimality gap in those instances. As can be seen from the results, except for the two instances $Å_{14}$ and $F_{16}$, the path-flow model finds the optimal solution in less time than the arc-flow model. The path-flow model is also able to find the optimal solution in six instances where the arc-flow model cannot within the given time limit. This shows that the path-flow model is superior to the arc-flow model with respect to the computational performance for this problem.

## 5 Concluding Remarks

In this paper we have studied a vessel pickup and delivery problem that arises in the case of disruptions in the supply vessel logistics in the offshore oil and gas industry. We have shown that the problem can be modelled as a multi-vehicle pickup and delivery problem and proposed two alternative formulations for the problem, i.e. an arc-flow and a path-flow formulation. For the path-flow formulation we have also proposed an efficient dynamic programming algorithm for generating the paths, which represent feasible vessel voyages. It was shown through a computational study on various realistic test instances provided by a major oil and gas company that the path-flow model was superior with respect to computational performance.

Even though the path-flow model presented in this paper can solve many instances to optimality, there is a need for methods that can provide high quality solution to even larger instances in a short amount of time. An interesting extension of this work would therefore be to look either at more advanced exact solution methods, such as branch-and-price, or to develop heuristic solution methods for the problem.

## References


1. Berbeglia, G., Cordeau, J.F., Gribkovskaia, I., Laporte, G.: Static pickup and delivery problems: a classification scheme and survey. TOP **15**(1), 1–31 (2007)
2. Brouer, B.D., Dirksen, J., Pisinger, D., Plum, C.E., Vaaben, B.: The vessel schedule recovery problem (VSRP) - a MIP model for handling disruptions in liner shipping. Eur. J. Oper. Res. **224**(2), 362–374 (2013)
3. Dell'Amico, M., Righini, G., Salani, M.: A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection. Transp. Sci. **40**(2), 235–247 (2006)
4. Fico.com: FICO Xpress Optimization Suite (2016). http://www.fico.com/en/products/fico-xpress-optimization-suite
5. Gribkovskaia, I., Laporte, G., Shlopak, A.: A tabu search heuristic for a routing problem arising in servicing of offshore oil and gas platforms. J. Oper. Res. Soc. **59**(11), 1449–1459 (2007)
6. Irnich, S., Desaulniers, G.: Shortest path problems with resource constraints. In: Desaulniers, G., Desrosiers, J., Solomon, M.M. (eds.) Column Generation. GERAD 25th Anniversary Series, pp. 33–65. Springer, New York (2005)
7. Kjeldsen, K.H.: Routing and scheduling in liner shipping. Ph.D. thesis, Aarhus University, p. 166 (2012)
8. Norwegian Petroleum Directorate: Petroleum figures (2015). http://www.npd.no/en/news/Production-figures/2015/December-2015/
9. Polat, O., Kalayci, C.B., Kulak, O., Günther, H.O.: A perturbation based variable neighborhood search heuristic for solving the vehicle routing problem with simultaneous pickup and delivery with time limit. Eur. J. Oper. Res. **242**(2), 369–382 (2015)
10. Shyshou, A., Gribkovskaia, I., Laporte, G., Fagerholt, K.: A large neighbourhood search heuristic for a periodic supply vessel planning problem arising in offshore oil and gas operations. INFOR: Inf. Syst. Oper. Res. **50**(4), 195–204 (2012)
11. Sopot, E., Gribkovskaia, I.: Routing of supply vessels to with deliveries and pickups of multiple commodities. Procedia Comput. Sci. **31**, 910–917 (2014)