Available online at www.sciencedirect.com

ScienceDirect

Energy Procedia 80 (2015) 92 – 99

Energy Procedia

12th Deep Sea Offshore Wind R&D Conference, EERA DeepWind'2015

Optimization of routing and scheduling of vessels to perform maintenance at offshore wind farms

Magnus Stålhane[a,*], Lars Magnus Hvattum[a], Vidar Skaar[a]

[a]*Department of Industrial Economics and Technology Management, NTNU, Alfred Getz veg 3, NO-7491 Trondheim, Norway*

Abstract

This paper studies the problem of finding the optimal routes and schedules for a fleet of vessels that are to perform maintenance tasks at an offshore wind farm. To solve the problem two alternative models are presented: an arc-flow and a path-flow formulation. Both models are tested on instances of varying numbers of vessels and maintenance tasks. The arc-flow model is solved with commercial software using branch-and-bound. The path-flow model is solved heuristically by generating a subset of the possible routes and schedules, but produces close to optimal solutions using a lot less computing time than the exact arc-flow model.

Peer-review under responsibility of SINTEF Energi AS

Keywords: Operation and maintenance; Routing and scheduling; Offshore Wind; Vehicle routing

1. Introduction

The demand for energy in general and renewable energy in particular is growing, and one important source is wind energy. Onshore wind turbines have been used for decades, however, in recent years the wind industry has begun to look offshore to find locations that fulfill the requirements for space and wind conditions. As the more easily accessible sites close to shore are exhausted, the offshore wind industry is looking further offshore to build their wind farms. These locations may be good with regards to installation of wind turbines, and have good conditions for energy production, however, the logistics of performing operations and maintenance become increasingly complex. In addition, there is a lot of pressure on wind farm operators to reduce costs, for the energy production to become profitable. One of the largest cost components of an offshore wind farm is the maintenance operations, which may constitute as much as 25 % of the life-cycle costs [1].

To increase the efficiency of the maintenance operations, and thus reduce the cost of energy, we have developed a mathematical model to optimize the routes and schedules of a fleet of vessels performing maintenance operations at an offshore wind farm on a given day. The model can be categorized as a generalization of the well-studied pickup and delivery problem [2], but has some unique aspects that separates it from the existing body of scientific literature.

* Corresponding author. Tel.: +47 73593084.
E-mail address: magnus.stalhane@iot.ntnu.no

Peer-review under responsibility of SINTEF Energi AS
doi:10.1016/j.egypro.2015.11.411

The fleet of vessels is heterogeneous, and located at a depot at the beginning of the planning horizon. The goal is to create one route and schedule for each vessel, traveling from the base to a set of wind turbines where it will deliver and pick-up technicians and spare parts that are needed to perform maintenance tasks at each turbine. Each vessel has a limited capacity for technicians and spare parts. The wave and wind conditions on a given day will affect which vessels can perform which maintenance operations and may also affect when, and for how long, a vessel may be away from the depot on a given day.

There are many different types of maintenance activities that take place at a wind turbine. We divide them into two main categories: Corrective and preventive maintenance. The most important difference between them is how downtime costs are calculated. For corrective maintenance tasks, downtime costs are incurred from the start of the planning period and stop when the crew leaves the turbine after completing the maintenance task. On the other hand, for preventive maintenance tasks, downtime costs start running from the time the crew arrives at the turbine. Each activity requires a certain amount of technicians, equipment, and some activities require special vessels. In addition, some activities, for instance underwater inspections of the foundations using remotely operated underwater vehicles, require the vessel to stay at the turbine for the duration of the operation. Maintenance tasks may be postponed to the next planning period, which leads to a penalty cost.

Similar problems to the one studied in this paper is presented by [3] and [4]. In [3] a model for planning opportunistic maintenance is presented. The decision taken is whether maintenance tasks are to be done on a given day or not, but the actual routing of the vessels is not considered. In [4] a similar problem for routing maintenance vessels at an offshore wind farm is studied. The main differences are that we calculate the downtime cost on a more detailed level, and that we present an efficient solution method, in addition to an arc-flow model, of the problem. For more detailed information regarding related types of problems, models, and solution methods we refer the reader to [5].

The remainder of the paper is organized as follows. In Section 2 a detailed description of the problem is given together with the arc-flow and the path-flow model. The method of creating routes and schedules for the path-flow model is described in Section 3. Section 4 contains a computational study, presenting a set of test instances and results of both the arc-flow model and the path-flow model. Finally, Section 5 contains concluding remarks.

2. Problem formulation

The problem described in this paper is to find the optimal set of routes and schedules to perform n maintenance tasks at an offshore wind farm. Let $I = \{1, \ldots, n\}$ denote the set of maintenance tasks that can be performed during the planning period. Each maintenance task $i \in I$ is defined by the time it takes to complete the maintenance task (T_i^W), and the number of technicians (L_i^P) and the total weight of the spare parts (L_i^W) needed. Further, it takes T_i^L hours to transfer personnel and equipment from a vessel to the turbine, and we define a time T_i^{max} as the maximum time technicians can stay at the turbine before being picked up. Finally, each maintenance task i has a downtime cost K_i per hour the turbine is shut down, and a penalty cost S_i that is incurred if the task is not done at all during the planning horizon.

The set I can be divided into two disjoint sets I^C and I^P, representing corrective and preventive maintenance tasks, respectively. What distinguishes the two types of maintenance tasks, is how the downtime cost K_i is treated. For corrective maintenance tasks the downtime cost is accumulated from the beginning of the planning horizon and until the task is finished, while for preventive maintenance tasks it is only accumulated in the time which the technicians are at the turbine. To perform these maintenance tasks, the wind farm operator has a set of vessels V at his disposal. Each vessel v has a capacity for Q_v^P technicians onboard and can carry Q_i^E kg of equipment.

The problem is defined on a graph $G = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N} = \{0, \ldots, 2n + 1\}$ is a set of nodes, and $\mathcal{A}$ is the set of arcs connecting those nodes. With each maintenance task in I we associate two nodes: the delivery node i where L_i^P technicians disembark to perform the maintenance task, and the pickup node $n+i$ where the technicians are collected by a vessel after completing the task (for modelling purposes $L_{(n+i)}^P = -L_i^P$). In addition, the set of nodes contain the nodes 0 and $2n + 1$ which both represent the base where vessels begin and end their shifts.

We define $A_v \subseteq A$ as the subset of arcs that can be traversed by vessel v. In the set A_v we remove all arcs connected to a node that corresponds to a maintenance task that vessel v cannot perform. For all maintenance tasks i where the vessel needs to be present during the maintenance task, we remove all arcs with a tail in i, except $(i, n+i)$. For each arc $(i, j) \in A_v$ we let T_{ijv} and C_{ijv} denote the time and cost of traversing that arc, respectively. Finally,

V_i is the subset of vessels that can perform maintenance task i, I_v the set of maintenance tasks that can be performed by vessel v, and $[A_{iv}, B_{iv}]$ defines a time window within which it is possible for vessel v to start maintenance task i. These time windows are set either as the length of the weather window for the vessel on a given day, or as the length of a shift, whichever is shorter.

2.1. Arc-flow model

The objective is to find a route through the network from node 0 to node $2n+1$ for each vessel, such that the total cost of performing the maintenance tasks is as low as possible. To solve this problem we have formulated an arc-flow model containing three sets of variables. The variables x_{ijv} are equal to one if vessel v traverses arc (i,j) and zero otherwise, t_{iv} denote the time vessel v visits node i, and p_{iv} denote the number of technicians onboard vessel v when leaving node i. With this notation we can formulate the problem as follows:

$$\begin{aligned} min \sum_{v\in V}\sum_{i\in N_v}\sum_{j\in N_v} C_{ijv}x_{ijv} + \sum_{v\in V}\sum_{i\in I^C} K_i(t_{iv} + T_i^L \sum_{j\in N_v} x_{ijv}) \\ + \sum_{v\in V}\sum_{i\in I^P} K_i(t_{(n+i)v} - t_{iv} + T_{(n+i)}^L \sum_{j\in N_v} x_{ijv}) + \sum_{i\in I} S_i(1 - \sum_{v\in V}\sum_{j\in N_v} x_{jiv}) \end{aligned} \quad (1)$$

$$\sum_{v\in V_j}\sum_{i\in N_v} x_{ijv} \le 1, \qquad j \in I, \quad (2)$$

$$\sum_{(0,j)\in A_v} x_{0jv} = 1, \qquad v \in V, \quad (3)$$

$$\sum_{(i,j)\in A_v} x_{ijv} - \sum_{(j,i)\in A_v} x_{jiv} = 0, \qquad v \in V, j \in I_v, \quad (4)$$

$$\sum_{(i,2n+1)\in A_v} x_{i(2n+1)v} = 1, \qquad v \in V, \quad (5)$$

$$\sum_{i\in N_v} x_{ijv} - \sum_{i\in N_v} x_{i(n+j)v} = 0, \qquad v \in V, j \in I_v, \quad (6)$$

$$\sum_{i\in I_v^D}\sum_{j\in N_v} L_i^E x_{ijv} \le Q_v^E, \qquad v \in V, \quad (7)$$

$$p_{iv} \le Q_v^P, \qquad v \in V, i \in \{0, 2n+1\}, \quad (8)$$

$$0 \le p_{iv} \le Q_v^P - L_i^P, \qquad v \in V, i \in I, \quad (9)$$

$$L_i^P \le p_{(n+i)v} \le Q_v^P, \qquad v \in V, i \in I, \quad (10)$$

$$p_{iv} + L_j^P - p_{jv} \le (Q_v^P + L_j^P)(1 - x_{ijv}), \qquad v \in V, (i,j) \in A_v, \quad (11)$$

$$p_{iv} + L_j^P - p_{jv} \ge (L_j^P - Q_v^P)(1 - x_{ijv}), \qquad v \in V, (i,j) \in A_v, \quad (12)$$

$$A_{iv}(\sum_{(i,j)\in A_v} x_{ijv}) \le t_{iv} \le B_{iv}(\sum_{(i,j)\in A_v} x_{ijv}), \qquad v \in V, i \in I_v, \quad (13)$$

$$A_{dv} \le t_{(2n+1)v} \le B_{dv}, \qquad v \in V, \quad (14)$$

$$t_{iv} + T_i^L + T_{ijv} - t_{jv} \le M_{ijv}^t(1 - x_{ijv}), \qquad v \in V, (i,j) \in A_v, \quad (15)$$

$$(T_i^W + T_i^L)(\sum_{v\in V_i}\sum_{(i,j)\in A_v} x_{ijv}) \le \sum_{v\in V_i}(t_{(n+i)v} - t_{iv}) \le T_i^{max}, \qquad i \in I, \quad (16)$$

$$x_{ijv} \in \{0,1\}, \qquad v \in V, (i,j) \in A_v. \quad (17)$$

The objective function (1) minimizes the total cost of performing maintenance at the wind farm. The first term represents the transportation costs, while the second and third term represents downtime cost associated with corrective and preventive maintenance tasks, respectively. The fourth term is a penalty cost for not performing maintenance tasks in the current time period. Constraints (2) state that each maintenance task must be done at most once, while constraints (3)–(5) ensures that each vessels route is continuous from node 0 to node $2n + 1$ in the problem network. Further, constraints (6) force the pickup and delivery node of a given maintenance task to be visited by the same vessel, and constraints (7) limits the total weight of equipment and spare parts onboard a vessel to be below the vessel's capacity. The number of technicians onboard a vessel when leaving a given node is kept track of through constraints (8)–(12), while constraints (13)–(15) keep track of the time at which each node is visited. Finally, constraints (16) enforce that the delivery nodes are visited before the corresponding pickup node, and constraints (17) state that each x-variable has to be binary.

2.2. Path-flow model

The arc-flow model may be decomposed using the Dantzig-Wolfe decomposition method, into what is often reffered to as a path-flow model. Before we can present this model, some additional notation must be defined. Let R be the set of all feasible paths r through the network, and let R_v be the subset of paths compatible with vessel v. Further, let C_{vr} be the cost, including downtime cost, of sailing route r with vessel v, and let A_{ivr} be equal to one if vessel v performs maintenance task i on route r, and zero otherwise. Finally, we define the variables λ_{vr} which are equal to one if vessel v sails route r, and zero otherwise, while y_i is equal to one if maintenance task i is not performed within the planning horizon and zero otherwise. With this notation we may formulate the path-flow model as follows:

$$min \sum_{v \in V} \sum_{r \in R} C_{vr} \lambda_{vr} + \sum_{i \in I^D} S_i y_i, \tag{18}$$

$$\sum_{v \in V} \sum_{r \in R} A_{ivr} \lambda_{vr} + y_i = 1, \qquad i \in I, \tag{19}$$

$$\sum_{r \in R_v} \lambda_{vr} = 1, \qquad v \in V, \tag{20}$$

$$y_i \in \{0, 1\}, \qquad i \in I, \tag{21}$$

$$\lambda_{vr} \in \{0, 1\}, \qquad v \in V, r \in R. \tag{22}$$

The objective function (18) minimizes the sum of the sailing costs and the penalty costs associated with not performing maintenance. Constraints (19) state the all maintenance tasks have to be performed, or incurs a penalty cost if it is not, while constraints (20) ensure that each vessel sails exactly one route. Finally, constraints (21) and (22) state that all y and x-variables are binary.

3. Path generation

To generate the paths needed in the path-flow model, the labelling Algorithm 1 is used. U is the set of unprocessed labels, initially only containing the label $L(o(v))$ representing a path just visiting the ship's origin node, $o(v)$, and P is the set of processed labels, which initially is empty. While there are labels left in U, the $removeEarliest(U)$ function removes the label from U that has spent the least amount of time. This label is extended along all resource feasible arcs, creating new labels, L'. Each new label L' is then checked for dominance, and if it is not dominated by any other label, it is added to U and P. Any labels in P that are dominated by L' are removed from both P and U. Once there are no more unprocessed labels left in U we filter out the labels in P where the last node $\eta(L)$ is not equal to the depot end-node and return the remaining labels which may be converted into feasible non-dominated paths R. In the following we describe what information is stored in a label, what constitutes a feasible extension, and how labels are compared for dominance.

3.1. Labels

For each label we store the following data:

Algorithm 1 Path generator

Input: graph $G_v = (\mathcal{N}_v, \mathcal{A}_v)$
$U = \{L(o(v))\}, P = \emptyset$
while $U \neq \emptyset$ **do**
    $L = removeEarliest(U)$
    for each feasible extension of $L \rightarrow L'$ **do**
        if no label in P dominates L' **then**
            $P = P \cup \{L'\}$
            $U = U \cup \{L'\}$
            remove all labels in P and U that are dominated by L'
        end if
    end for
end while
for $L \in P$ **do**
    if $\eta(L) \neq d(v)$ **then**
        $P = P \setminus \{L\}$
    end if
end for
return P

1. η - the node of the label
2. ϕ - the predecessor label
3. t_i - the earliest possible start of service at the each visited node i
4. c - the accumulated cost
5. p - the number of personnel onboard
6. l - the weight of spare parts onboard
7. Δ - the set of nodes visited on the (partial) path

In the rest of this paper, the notation $c(L)$ is used to refer to accumulated cost of label L and similar notation is used for the rest of the data (e.g., $\eta(L), \phi(L), t_i(L), p(L), l(L)$, and $\Delta(L)$).

3.2. Label extension

When extending a label L along an arc ($\eta(L), j$), we create a new label L' at node j, and update the label data as follows:

$$\eta(L') = j \tag{23}$$

$$\phi(L') = L \tag{24}$$

$$t_i(L') = \begin{cases} t_i(L) & \text{if } i \neq j, \\ t_{\eta(L)} + T^L_{\eta(L)} + T_{\eta(L)jv}, & \text{if } i = j, j \in I^D \cup \{d\} \\ \max\{t_{\eta(L)} + T^L_{\eta(L)} + T_{\eta(L)jv}, t_{j-n}(L) + T^L_{j-n} + T^W_{j-n}\} & \text{if } i = j, j \in I^P \end{cases} \tag{25}$$

$$c(L') = c_{\eta(L)} + C_{ijv} + \begin{cases} K_j(t(L') + T^L_j) & \text{if } j \in I^D \cap I^{corr}, \\ +K_j(t(L') - t_{j-n} + T^L_j) & \text{if } j \in I^P, \\ 0, & \text{otherwise.} \end{cases} \tag{26}$$

$$p(L') = p(L) + L^P_j, \tag{27}$$

$$l(L') = l(L) + L^E_j, \tag{28}$$

$$\Delta(L') = \Delta(L) + \{j\}. \tag{29}$$

Equations (23) – (26) update the current node, the predecessor label, the earliest possible start of service, and the accumulated cost of the (parital) path represented by the label, respectively. Equation (27) and (28) update the set number of technicians and equipment onboard, while the set of nodes visited is updated by Equation (29).

The extended label L' is considered a feasible extension iff:

$$t(L') \leq B_{jv}, \tag{30}$$

$$0 \leq p(L') \leq Q_v^P, \tag{31}$$

$$l(L') \leq Q_v^E. \tag{32}$$

and one of the following hold:

$$0 < j \leq n \wedge j \notin \Delta(L) \tag{33}$$

$$n < j \leq 2n \wedge j \notin \Delta(L) \wedge (j - n) \in \Delta(L) \tag{34}$$

$$j = d(v) \wedge \forall i \in I(i \in \Delta(L) \rightarrow (i + n) \in \Delta(L)) \tag{35}$$

Equations (30) – (32) simply states that the time, number of technicians and the weight of the equipment must stay within their limits, while Equations (33) – (35) ensure that the generated path comply with the pairing and precedence constraints of the problem.

3.3. Dominance criterion

To avoid generating all possible routes through the network, a dominance criterion is used to remove paths that are unlikely to be a part of an optimal solution. We let label L_1 dominate L_2 if:

1. $\eta(L_1) = \eta(L_2)$
2. $\Delta(L_1) = \Delta(L_2)$
3. $t(L_1) \leq t(L_2)$
4. $c_j(L_1) + \sum_{i \in \mathcal{O}(L)} \max\{0, t_i(L_2) - t_i(L_1)\}K_i \leq c_j(L_2)$

The first three parts of the dominance criterion simply state that the two paths must have the same last node, have visited the same set of nodes, and that L_1 has not spent more time than L_2 up to that point. Together these three ensure that any feasible extension of L_2 will also be a feasible extension of L_1.

In many VRPs it would be sufficient to add that the accumulated costs must be lower ($c_j(L1) \leq c_j(L2)$). However, since downtime costs depend not only on the time when personnel leave the turbine, but also when they arrive, this is insufficient. Since downtime costs are added to c after completing assignments (when visiting pickup nodes), the paths represented by L_1 and L_2 may have differences in accumulated costs that have not yet been added to c. We will refer to assignments where the delivery node has been visited but not the corresponding pickup node as open assignments, $\mathcal{O}(L)$. To be able to dominate when there are open assignments, we add the difference in unrealized downtime costs to the accumulated costs in part four of the dominance criterion.

It is assumed in our labeling algorithm that visits to a node always happen as early as possible. This assumption can lead to incorrect dominance when preventive maintenance tasks are involved. When maintenance crew has to wait at the turbine after finishing a preventive assignment (personnel slack, $t_{i+n} - t_i > T_i^W + T_i^L$) and the vessel at some point between visiting nodes i and $i + n$ has been forced to wait for the completion of another assignment (vessel slack), there is potential for cutting costs by postponing the arrival at node i. This is illustrated in Figure 1. Let operation 1 be a corrective maintenance task, while operation 2 is preventive. The longest bar in the figure is the vessel's time line, while the two other bars are for personnel left at turbines. Shaded areas represent idle time for vessel or personnel. In this case the total cost of the route could be reduced by delaying the start of maintenance task 2, because we would then get rid of the personnel slack indicated at the end of that maintenance task.

The arc-flow model has the flexibility to delay the start of preventive assignments if it reduces costs. To apply the same kind of flexibility to the labeling algorithm would likely require solving a linear program for every completed path. This means that the path-flow model must be classified as a heuristic whenever preventive assignments are

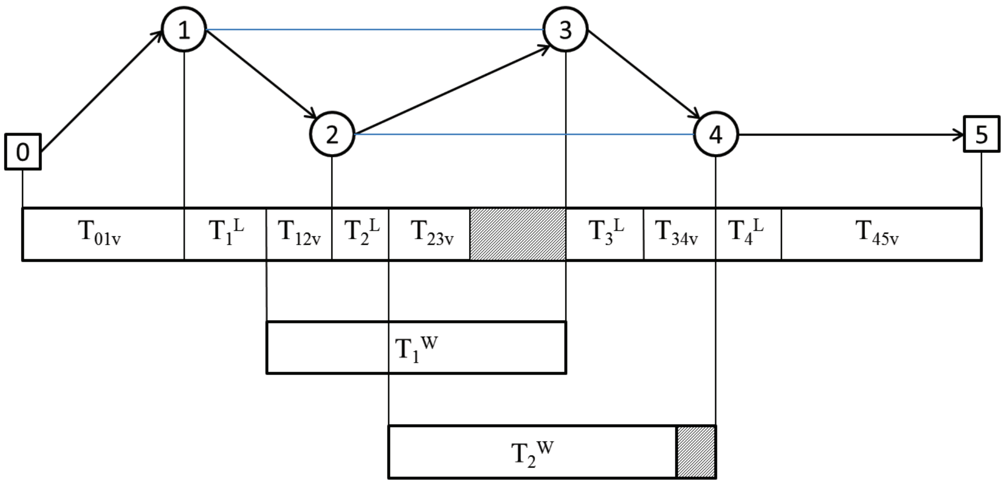

Fig. 1. Heuristic path with potential for savings

included. It may also be argued that having the technicians wait before disembarking at a turbine is unlikely to happen in practice. Especially since in a real life situation, the time it takes to perform a maintenance task will vary, and thus it will produce more robust solutions if the maintenance tasks are started as early as possible.

4. Computational study

The optimization models presented in Section 2 have been implemented in the Mosel programming language, and solved with FICO Xpress version 7.5. The tests have been carried out on a HP DL 160 G5 computer with an Intel Xeon QuadCore E5472 3,0 GHz processor, 16 GB RAM running in Linux. The instances and paths have been generated in MATLAB.

4.1. Instances

Instances are based on using a fleet of crew transfer vessels to serve a wind farm randomly placed between 60-80 km from an onshore depot. A workday lasts 12 hours, and the weather is assumed to be good enough that no restrictions are imposed on the time where maintenance activities can be started. Wind turbines are placed in a grid with 1 km between neighboring turbines. Five different corrective and two different preventive maintenance types are considered, with varying time and personnel requirements. Each instance generated contains between two and nine different maintenance tasks, to be handled by a fleet of between two and five vessels. Vessel and maintenance operation data are based on the work by [6].

4.2. Results

Table 1 presents results on 28 instances created as described above. The arc-flow model is solved to optimality within a time limit of 7200 seconds for all instances with up to seven tasks. All instances, except those with eight tasks, are solved to optimality within the time limit. As the table shows, the number of nodes in the branch-and-bound tree grows rapidly with an increasing number of tasks. For the path-flow model, with heuristically generated routes and schedules, the running times reported are almost entirely due to the path generation, as in all cases the resulting model is solved to optimality by Xpress in less than 0.1 seconds. Furthermore, the resulting solution is always within 1 % of the optimal solution found through the arc-flow model.

The path-flow heuristic has superior running times for all but the smallest instances. The differences increase with the number of vessels, since this parameter gives a linear increase in running times for the path-flow heuristic and an

Instance		arc-flow		path flow heuristic		
#V	#O	Nodes	Secs	Paths	Secs	Diff
2	**2**	15	0	10	0	0.0%
2	**3**	147	0	22	0	0.4%
2	**4**	1369	1	22	0	0.0%
2	**5**	14704	3	48	4	0.0%
2	**6**	51316	8	56	5	0.0%
2	**7**	784136	162	148	36	0.4%
2	**8**	22915783	> 7200	330	366	0.7%
3	**2**	37	0	15	0	0.0%
3	**3**	161	1	33	0	0.4%
3	**4**	1200	2	33	1	0.0%
3	**5**	26887	7	72	5	0.0%
3	**6**	155680	47	84	7	0.0%
3	**7**	2462032	980	222	54	0.4%
3	**8**	12285343	> 7200	495	549	0.3%
4	**2**	11	0	20	0	0.0%
4	**3**	267	1	44	0	0.4%
4	**4**	2805	2	44	1	0.0%
4	**5**	52592	15	96	7	0.0%
4	**6**	204928	88	112	9	0.0%
4	**7**	4222723	2369	296	73	0.4%
4	**8**	8269841	> 7200	660	733	0.8%
5	**2**	17	0	25	0	0.0%
5	**3**	427	1	55	0	0.4%
5	**4**	3246	2	55	2	0.0%
5	**5**	34661	12	120	9	0.0%
5	**6**	354880	189	140	11	0.0%
5	**7**	5656598	3674	370	91	0.4%
5	**8**	7016897	> 7200	825	915	0.5%

Table 1. Comparison of the arc-flow model and the path-flow model solved heuristically.

exponential increase in the arc-flow version. While the number of paths generated for the path-flow model are low throughout, the time taken to generate these paths increase quite fast with the number of maintenance tasks. Therefore, the largest instances that can be solved within 7200 seconds by the path-flow model are not much larger than what can be solved with the arc-flow model.

5. Concluding remarks

This paper studies the routing and scheduling of vessels that perform maintenance tasks at offshore wind farms. Two mathematical models are presented, one based on arc-flow and another based on path-flow. To solve the path-flow model an efficient heuristic labeling algorithm is used. Computational experiments show that the path-flow model together with the labeling algorithm solves the problem to near optimality at a significantly smaller computational effort than the arc-flow model.

References

[1] Snyder, B., Kaiser, M., 2009. Ecological and economic cost-benefit analysis of offshore wind energy. Renewable Energy 34 (6), 1567–1578.

[2] Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., Laporte, G., 2007. Static pickup and delivery problems: a classification scheme and survey. TOP 15, 1–31.

[3] Besnard, F., Fischer, K., Tjernberg, L., 2013. A model for the optimization of the maintenance support organization for offshore wind farms. Sustainable Energy, IEEE Transactions on 4 (2), 443–450.

[4] Dai, L., Stålhane, M., Utne, I. B., 2015. Routing and scheduling of maintenance fleet for offshore wind farms. Wind Engineering: The International Journal of Wind Power 39 (1), 15–30.

[5] Christiansen, M., Fagerholt, K., Nygreen, B., Ronen, D., 2013. Ship routing and scheduling in the new millennium. European Journal of Operational Research 228 (3), 467 – 483.

[6] Endrerud, O.-E., Sperstad, I. B., Hofmann, M., Martin, R., Dinwoodie, I., Feuchtwang, J., Nov., Dec. 2015. Operation and maintenance simulation model verification and validation, forthcoming in Wind Engineering : The International Journal of Wind Power.