

# An optimisation model for scheduling the decommissioning of an offshore wind farm

Chandra Ade Irawan · Graham Wall ·  
Dylan Jones

Received: date / Accepted: date

**Abstract** An optimisation model is proposed for scheduling the decommissioning of an offshore wind farm in order to minimise the total cost which is comprised of jack-up vessel, barge (transfer) vessel, inventory, processing and on-land transportation costs. This paper also presents a comprehensive review of the strategic issues relating to the decommissioning process and of scheduling models that have been applied to offshore wind farms. A mathematical model using integer linear programming (ILP) is developed to determine the optimal schedule considering several constraints such as the availability of vessels and planning delays. As the decommissioning problem is challenging to solve, a matheuristic approach based on the hybridization of a heuristic approach and an exact method is also proposed to find near optimal solutions for a test-set of problems. A set of computational experiments has been carried out to assess the proposed approach.

**Keywords** Scheduling · Offshore Wind Farm · Renewable Energy · Matheuristic

## 1 Introduction

The number of global offshore wind farms is increasing at a rapid rate. For example, the UK offshore wind industry has been gathering momentum over the past decade with the UK Government commissioning three-phases of suitable offshore sites. In addition, they have provided a number of financial incentives for potential developers such as feed-in-tariffs and renewable obligation certificates. As a result, the UK is the world leader in offshore wind energy with 1184 wind turbines connected (4.049GW). With a further 5.2GW in the planning system and

---

Chandra Ade Irawan  
Nottingham University Business School China, University of Nottingham Ningbo China,  
199 Taikang East Road, Ningbo 315100, China  
E-mail: chandra.irawan@nottingham.edu.cn

Graham Wall and Dylan Jones  
Department of Mathematics, Centre for Operational Research and Logistics, University of  
Portsmouth,  
Lion Gate Building, Lion Terrace, Portsmouth, PO1 3HF, UK

another 12.4GW under construction or having planning approval (some of which is expected to be completed by 2020), it is estimated that there will be 10GW operating by 2020 (Renewable UK, 2015). However, in order for the developers to get the required consent, a detailed decommissioning programme needs to be submitted as part of their installation plans. Decommissioning is the last part of the offshore wind farm life cycle which is estimated to be 20-25 years after installation. For some of the UK Round 1 sites (e.g. Blyth offshore wind farm commissioned in 2000), decommissioning decisions will need to be implemented between the 2022 and 2027. The major purpose of this stage is to re-power or clear the wind farm area. Even if a decision is taken to re-power the turbines, this will only postpone the need for the eventual dismantling and processing of the wind farm components, and the clearing of the site.

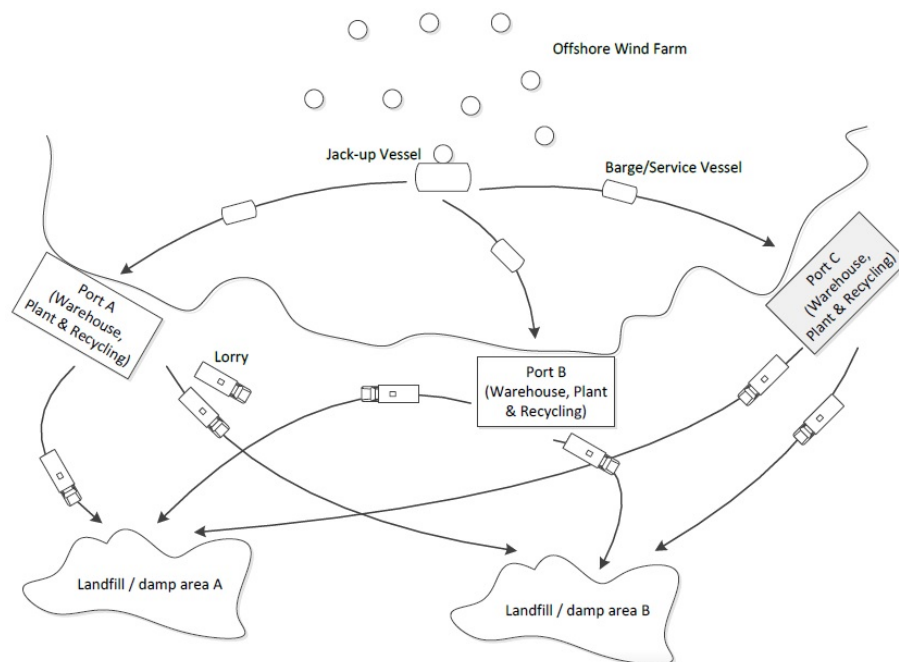
A wind turbine comprises three main components namely top-structure (tower, nacelle, and blades), array cable, and sub-structure (foundation and transition piece). To dismantle such components, heavy equipment such as jack-up and barge vessels are needed. A jack-up vessel is a mobile platform with a buoyant hull fitted with a jib crane and a number of moveable legs which can be raised or lowered to the sea bed. Based on the practical reality, these vessels are expensive to charter and are limited in their availability. A barge vessel is a flat-bottomed boat used for the transportation of heavy goods. In the model, barge vessels are always available. The availability of jack-up vessels and other factors causing the delay (such as weather conditions) influence the timing and performance of the dismantling activities. A one-day jack-up vessel idle time will cause a significant financial loss due the specialised nature, and hence high charter costs of the vessel involved. For safety reasons, the dismantling process must be performed in the period when a set of required weather conditions are met.

Figure 1 shows the scope of the decommissioning model considered in this paper. The components of the offshore wind turbines are dismantled using a jack-up vessel. These components are then transported to the suitable decommissioning ports using a barge vessel. In the ports, the components are stored and processed by a machine (i.e. crusher). The processed components are then transferred into recycle or dump/landfill areas. The decommissioning ports are usually near to the wind farm site and have a workshop area for preparing the components for recycling. In the proposed model, the recycling area is located in the decommissioning port.

In this paper, we are investigating the decommissioning scheduling of an offshore wind farm in order to minimise total decommissioning cost, which includes jack-up vessel, barge (transfer) vessel, inventory, processing and on-land transportation costs. The main contributions of this paper include:

- i. a novel mathematical model of the decommissioning offshore wind farm scheduling problem.
- ii. a novel matheuristic approach based on hybridization of a heuristic method and an exact method for solving the scheduling problems.
- iii. solutions for the West Gabbard (United Kingdom) wind farm site are proposed.

We propose a novel mathematical model and a novel matheuristic approach for the decommissioning problem in offshore wind farm as to the best of our knowledge, the model and its solution method have not been addressed in the



**Fig. 1** Scope of decommissioning model

literature. The remainder of this paper is organized as follows: Section 2 presents a review of the decommissioning of offshore wind farms along with the scheduling problem in offshore wind farms. Section 3 gives a description of the proposed model along with a mathematical model of the decommissioning scheduling of an offshore wind farm. Our proposed metaheuristic approach including the set of algorithms is presented in Section 4. Section 5 gives computational results using generated data based on the West Gabbard (UK) offshore wind farm site. A summary of our findings and some avenues for future research are provided in the last section.

## 2 Review on the decommissioning of offshore wind farms

This review covers the process and legislation, costs and managing waste and recycling. In addition, a review on scheduling problem in offshore wind farms is also given.

### 2.1 Process and legalisation

Although there is little experience of the process, the decommissioning of an offshore wind farm is often assumed to be a reversal of the installation process: removing the electrical infrastructure, the substation, removing the rotor, the nacelle, the tower, the transition piece, and (partially) removing the foundations. The main purpose of the offshore substation is to house the electrical high-voltage and

medium-voltage components for transforming power supplied by the wind turbines for export to the onshore grid via the underground electrical cables. In fact, there are a number of differences between the installation and decommissioning process. This includes the speed of the process, the planning time horizon, the engineering precision required, the inventory type handled, the management of waste and the necessity for on-going monitoring. In the installation process, there are two main objectives that need to be optimised, namely installation completion time and installation cost. The flow of components is usually from a single installation port to many points of the offshore turbine locations. While in the decommissioning process, the decommissioning cost is in the main concerned with the used components transported from the many turbine locations to several possible ports. Compared to decommissioning offshore oil and gas installations, or nuclear installations, wind farm decommissioning is less complex with far less toxic materials to deal with. While this is true in a general sense, there are still many complications to the process (Pearson, 2001). The following main offshore elements need to be dismantled: Offshore wind turbines, foundations and transition pieces and substations. It is envisaged that the foundation piles and the cabling below the sea bed (as well as the scour materials) are likely to be left in situ at least one metre under the sea bed due to the unacceptable environmental impact of removing them as well as the extreme cost.

Legalisation has evolved over recent years and become stricter, particularly in the area of mitigating the environmental impact of offshore installations by reducing dumping, pollution and waste. The decommissioning of many offshore wind farms is expected to be carried out in a legalisation framework with even more stringent environment regulations in the future. In order to comply with The Energy Act 2004 (DECC, 2011), the offshore wind farm developer is required to submit a decommissioning plan prior to construction. As well as having to cover areas such as scope and process, they are also required to provide details of exactly what will be decommissioned, the cost of decommissioning and an acceptable financial security to guarantee carrying out the work.

## 2.2 Costs

Given the limited experience of decommissioning offshore wind farms, estimating the costs involved can be very difficult. In addition, it is expected that there could be significant changes in technology and legislation by the time decommissioning is required which makes any estimate of the total costs even more difficult. If a decision to re-power the wind turbines is made after 25 years, full decommissioning may not take place for up to the lease period (which can be a maximum of 50 years). The cost of decommissioning is often based on the same variables as the installation phase, but also involving the costs related to cleaning and clearing the site as well as the final elimination process. Kaiser and Snyder (2012) estimate that the area encompassing an offshore wind farm could range 0.1-0.3 km<sup>2</sup> per MW and could contain a significant amount of debris that has accumulated over the previous 25 years and which needs to be cleared.

One such net estimate based on interviews with stakeholders is £40,000 per MW using a 240MW wind farm as a model (CCC, 2010). This figure includes the complete removal of the offshore wind turbines, including foundations and cables

1-2 metres below the seabed, but does not take into account the pre-dismantling surveys required, management of the waste and any post-monitoring that may be required. It also does not include any further surveys or remedial work that may be required in cases where the installation has not been fully removed and where there is a possibility of remaining materials (e.g. cables or foundations) becoming exposed over time, posing a risk to other users of the sea. From the industry's opinion, the dismantling cost is estimated at 3% of total cost of wind farm (Januario et al, 2007) or 2.5% (CCC, 2010) assuming an operating lifetime of 25 years. This followed on from an earlier estimate from Pearson (2001) who suggested a cost between £34,000 and £38,000 per MW. The decommissioning plan for Gwynt y Môr Welsh offshore wind farm adapted the CCC estimate which would have given a total cost of around £23M. However, they increased their estimate to £60M (almost triple the CCC estimate at £104,000/MW or £400,000/turbine). This was as a result of the "significant uncertainty that remains due to lack of experience and a 20 year plus time horizon before actual decommissioning during which market conditions, technology and environmental knowledge may change" (RWE, 2011). However, no detailed breakdown of these costs was given in the plan. More recently, the decommissioning of five 2MW offshore wind turbines in Sweden has an estimated cost of £80,000/MW (Vattenfall, 2016).

This large cost variation highlights the high uncertainty in the duration of carrying out the decommissioning process. Lack of experience in decommissioning offshore renewable installations increases the risk that developers are unable to provide a fair and accurate valuation of these costs. If a two-year period is allowed for the decommissioning process (as in the oil and gas industry), ample time is available to develop a cost-efficient strategy. This would schedule as much of the work in the summer months where there is likely to be less weather disruption and charter the required vessels and personnel at the cheapest rates. Developing new techniques may reduce removal costs over time in potential ways such as the use of specific decommissioning vessels, methods for toppling turbines and methods for removing copper from cables.

### 2.3 Managing waste and recycling

The environmental credentials of offshore wind energy are determined in part by the way in which components and materials are dealt with at the end of their life. The options available for offshore wind turbines can be summarised by the 5 R's - reuse, repair, recycle, retire or remanufacture (Bradley, S., 2013). Unlike in the oil and gas industry, the vast majority of the materials are inert making any disposal or recycling a much easier process. If re-manufacture or re-powering is not a solution for the wind turbine, then there is still a salvage value for many of the materials contained within the various components. Some materials, such as electrical cables or the steel from the platforms, can be sold, which implies a final income to cover part of the dismantling cost. As 90% of blades, the tower and the nacelle can be recycled, it has been suggested that this salvage income could offset a significant proportion of the decommissioning cost (Stecky-Efantis, 2013). However, the age and condition of many of the components of an offshore wind turbine at the time of decommissioning, scrapping or recycling may only be viable for certain components. The cost of decommissioning an offshore wind farm can be

reduced by implementing efficient plans, which can be generated by optimisation models. This paper presents a mathematical model along with its solution method to produce an optimal schedule for decommissioning activities starting from wind farm site until recycling or dump/land fill area.

## 2.4 Scheduling problems in offshore wind farms

Due to the dearth of dismantling scheduling research in an offshore wind farm, this section presents an overview of past efforts at offshore wind farms focussing on installation and maintenance scheduling problems.

### *Installation Scheduling Problem*

Scholz-Reiter et al (2010) proposed a mathematical model (MILP) to generate the optimal schedule for installation in order to reduce vessel operation times taking into account weather conditions. Based on their previous model (Scholz-Reiter et al, 2010), Scholz-Reiter et al (2011) investigated a heuristic approach to solve large installation scheduling problems with longer time horizons, multiple vessels and a broader variety of weather conditions. Lütjen and Karimi (2012) studied a simulation approach for determining the optimised configuration of a single-echelon inventory system for offshore installations of wind turbines. Ait-Alla et al (2013) proposed a mathematical model dealing with the aggregated installation planning problem in order to minimise the total installation costs for a medium planning horizon. In the model, the chartering costs and weather operation constraints for different vessel types are taken into account. Ursavas (2017) studied a benders decomposition approach for solving the offshore wind farm installation planning at the North Sea where a model that considers disruptions arising from weather conditions is proposed. Recently, Irawan et al (2017a) proposed a bi-objective optimisation model using a compromise programming approach. In their model, the installation cost and the completion period of the installation are considered as objective functions which are in conflict with each other.

### *Maintenance Scheduling Problem*

An opportunistic maintenance optimization model was proposed by Besnard et al (2009) considering wind forecasts and corrective maintenance activities. Pérez et al (2010), Byon et al (2011) and Pérez et al (2013) investigated discrete event-based simulation models of maintenance scheduling. A mathematical model (MILP) was built by Kovacs et al (2011) to find the best time for maintenance operations. Besnard et al (2011) proposed a model where uncertain weather conditions are considered. Parikh (2012) developed a formulation of mathematical model to optimise maintenance cost. Van Horenbeek et al (2012) quantified the added value of a prognostic maintenance policy. Wang et al (2012) introduced long- and short-term scheduling models for wind power integrated systems. The maintenance scheduling model considering peak regulation pressure balance was studied by Wu et al (2012). Zhang et al (2012b) proposed an optimal preventive maintenance scheduling model for minimising the overall downtime energy losses. Zhang et al (2012a) introduced maintenance scheduling of large-scale wind power considering peak shaving. Benmessaoud et al (2013) applied a simulation

model for optimising a maintenance schedule which can be used for analysing the influence of maintenance on the performance of a wind farm. Dos Santos et al (2013) developed a stochastic petri-net model for maintenance planning taking into account the availability of vessels, crews and spare parts. Pan et al (2013) introduced a long-term multi-objective optimisation model involving maintenance scheduling, unit commitment and power output. Pattison et al (2013) studied an integrated planning and scheduling maintenance method. Stålhane et al (2014) and Dai et al (2015) proposed a model to obtain the optimal routes and schedules for a fleet of vessels that are to perform maintenance tasks at an offshore wind farm. Camci (2015) investigated a method to schedule maintenance of geographically distributed assets (such as offshore wind farm) with failure probability predictions. Irawan et al (2017b) proposed an optimisation model and a solution method for maintenance routing and scheduling. This model determines the optimal schedule for maintaining the turbines and the optimal routes for the crew transfer vessels to visit the turbines along with the number of technicians required for each vessel. Sarker and Faiz (2017) developed a model for wind turbine installation and transportation in an offshore wind farm to minimise the total cost where the effects of wind farm and vessel parameters on cost are investigated. Recently, Barlow et al (2018) proposed a mixed-method framework to address current offshore wind farm installation projects. The framework benefits the complementary strengths of two decision-support methods namely discrete-event simulation and robust optimisation. The combined framework was applied to an installation case study located in the North Sea to illustrate the application of the combined approach.

Based on the papers cited above and to the best of our knowledge, a mathematical model for decommissioning problem in order to minimise the total cost has not been addressed in the literature. Therefore, in this paper, we propose the integrated model for the decommissioning of an offshore wind farm to obtain the optimal schedule of the jack-up vessel and the flows of components from wind farm site to recycling/dump areas. The proposed model also considers the on-land logistic system including inventory and transportation systems. As the decommissioning of an offshore wind farm can be considered as a reversal of the installation process, the proposed decommissioning model benefits from the installation model that available in the literature (i.e. Irawan et al (2017a)). However, with different process and scope, developing a mathematical model for the decommissioning problem in offshore wind farm is very challenging. In addition, as the nature of the decommissioning problem is quite unique, a matheuristic is also designed to solve the proposed problem. The proposed matheuristic is developed based on the relaxed problem principle and an interchange heuristic.

### 3 Decommissioning Model of an Offshore Wind Farm

This section consists of three subsections where the first one describes the decommissioning model of an offshore wind farm. The second subsection discusses the procedure for generating feasible slots/schedule for jack-up vessels to dismantle the turbines. A mathematical model using integer linear programming (ILP) is presented in the last subsection.

### 3.1 Description of the model

The objective function of the proposed model is to minimise the total cost of the decommissioning activities of an offshore wind farm. In contrast to its counterpart the installation problem, the completion time of the decommissioning process is not crucial. Therefore a cost minimisation model is developed. As array and export cables are recommended to be left in situ, the model considers two main components namely top- and sub-structures. The top-structure consists of three sub-components namely blades, nacelle and tower whereas the sub-structure comprises of the foundation and transition pieces. The substations are not taken into account as the number of substations in a wind farm is relatively small.

The dismantling of a wind turbine can be organised into a sequence of two tasks where removing the sub-structure is a successor task to the top-structure dismantling. A jack-up vessel is needed to dismantle these components. A jack-up vessel may be used to perform both dismantling tasks. For example, if only one jack-up vessel is used, that jack-up vessel will dismantle both sub- and top-structures. In this case, the jack-up vessel will remove the top-structures of a set of wind turbines. Once all top-structures of the wind turbines have been removed, the jack-up vessel will dismantle the sub-structures. In the case that more than one jack-up vessel is used, offshore wind turbines can be dismantled using several scenarios. For example, when two jack-up vessels are used, the first scenario is to assign one vessel to the dismantling top-structure task and another vessel to the sub-structure one. The other possible scenario is to divide the wind farm area into two areas where each vessel performs both tasks in its area. There are other possible scenarios to dismantle the turbines. The proposed model will optimise the best scenario to be implemented in dismantling a set of offshore wind turbines. The use of a jack-up vessel can be switched from one decommissioning task to another (e.g. sub-structure task to top-structure one). However, extra cost and time are required for doing this. Therefore, in this study the length of the rental period of a jack-up vessel to perform a decommissioning task must be greater than the minimum hire period. For example, the jack-up vessel must be hired for at least 7 consecutive days to do a decommissioning task ( $\beta_v = 7$ ). Each jack-up vessel has a different capacity to remove the turbines for each task. For example, a jack-up vessel can remove 4 top-structure turbines within one day ( $\alpha_{v,j} = 4$ ). Here, it is assumed that the value of is integer and greater or equal to one.

Once the components have been dismantled (by a jack-up vessel), the sub-components are transported to the port by using a barge/transfer vessel. The model only considers the transportation cost of this vessel and assumes that this vessel is always available when needed. The sub-components will also arrive to a port in the period when they are dismantled. In port, the unprocessed sub-components will be stored waiting to be processed (e.g. crushed) by a machine. Here, we assume that the unprocessed sub-components need at least one day to be stored. The processing cost and time for each sub-component is taken into account in the proposed model. It is assumed that a sub-component requires at least one day to be processed ( $\pi_{p,k} \geq 1$ ) and the value of  $\pi_{p,k}$  is integer. Note that the machine at a port has a capacity constraint to process the sub-components. For example the maximum amount of components to be processed by the machine is set to 5 units per period ( $\theta_{p,k} = 5$ ). After the sub-components have been processed, they will be stored and then transported to dump or recycle areas. The



model also takes into account the inventory cost for unprocessed and processed sub-components, the capacity of the processing machines, on-land (truck) transportation cost and the capacity of dump/recycle areas (in tonnages). Note that, as the recycling area is located in the port, there is no on-land transportation cost for the recyclable components. The tower and foundation can be considered as recyclable components whereas the blades and nacelle need to be transferred to a dump (landfill) area. We assume that trucks needed for transporting the components are always available in ports and one unit component that has been processed is transported by one or more trucks. The transportation time required to transfer components from ports to recycling/dump areas is not considered. Here, we only limit the amount of components that can be transferred from a port to recycling/dump areas in one period. In the model, the salvage value obtained from the recyclable components is not considered as the model presented treats the percentage of components that are salvageable as a constant and hence the salvage value will not affect the optimal scheduling decision made.

Figure 2 presents the main aspects considered in the model. The required information of the jack-up vessel is the rental cost per period, the fixed cost of using a jack-up vessel, the number of components (sub- and top-structures) that can be dismantled per period, minimum hired periods and the availability per period. A planning delay factor is also needed to determine whether the dismantling activity can be performed or not within a period. This delay can be based on one of several potential foreknown occurrences such as port conditions, the availability of non-vessel equipment, scheduled vacations or regulatory conditions causing staff shortages, company policy and environmental concerns such as bird migratory factors. The transportation cost of each sub-component from site to ports using a barge/service vessel is required where the cost is usually based on the distance along with the dimension and the weight of the sub-components. Here, it is assumed that a barge vessel is able to transport any type of turbine components with different dimensions.

In ports, the space for storing the sub-components (unprocessed or processed) is assumed to be unlimited as the area needed to store the components is relatively small compared to the port area. Therefore, the storage capacity is not considered in the model. Inventory (holding) cost to store a component per period is taken into account as it may not be the same for each port. As sub-components need to be processed before they are transferred to dump/recycling areas, the machines for processing these components are located in the port. Here, the machine has a limited capacity and the process may take several periods/days. Trucks are required to transport the processed components from port to dump area. Each port has a different maximum amount of components that can be transferred to dump/recycling areas per period. This model also takes into account the capacity constraints of dump and recycling areas (in tonnages). The following notations are used to describe the sets and parameters of the proposed scheduling model.

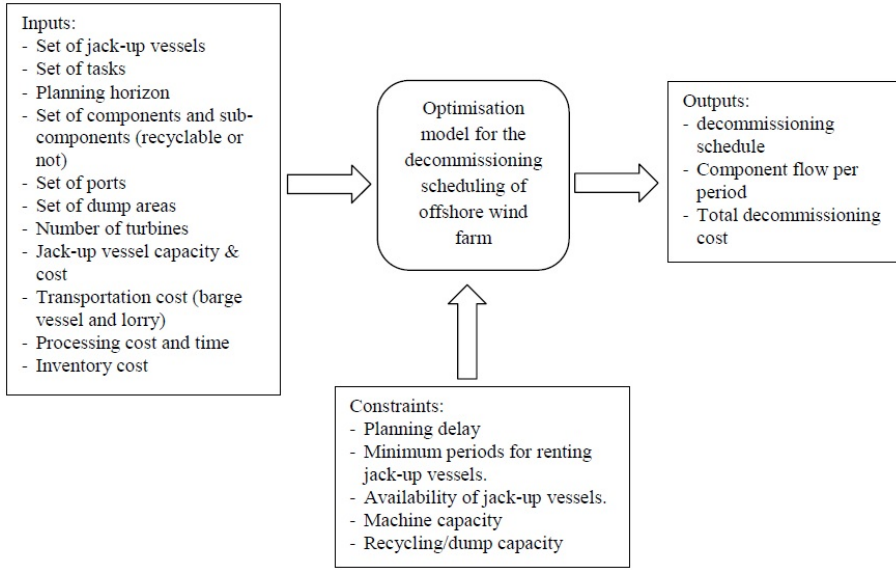
#### *Sets and indices*

$V$ : set of jack-up vessels indexed by  $v$ .

$J$ : set of decommissioning tasks indexed by  $j$  ( $j = 1$  for top-structure and  $j = 2$  for sub-structure).

$V_j$ : set of jack-up vessels to perform task  $j$ ,  $V_j \subset V$ .

$T$ : set of planning periods indexed by  $t$ .



**Fig. 2** The main aspects considered in the model

$P$ : set of sub-components/parts indexed by  $p$  (e.g. blades, rotor, nacelle, and etc.).

$P_j$ : set of sub-components included in task  $j$ ,  $P_j \subset P$ .

$P^r$ : set of recyclable sub-components,  $P^r \subset P$ .

$P^d$ : set of non-recyclable sub-components,  $P^d \subset P$ .

$K$ : set of ports (with warehouse, plant and recycling areas) indexed by  $k$ .

$D$ : set of dump/landfill areas indexed by  $d$ .

### Parameters

$n$ : total number of turbines to be removed.

$\alpha_{v,j}$ : the number of top/sub structures (task  $j \in J$ ) that can be removed by jack-up vessel  $v \in V_j$  in one period. This parameter is an integer value.

$f_{v,j}$ : the fixed cost per period of jack-up vessel  $v \in V_j$  to perform task  $j \in J$ .

$c_{v,j}$ : the variable cost per period of jack-up vessel  $v \in V_j$  to perform task  $j \in J$ .

$\rho_{v,t} = 1$  if vessel  $v \in V$  is available in period  $t \in T$ ,  $= 0$  otherwise

$\tau_{v,j} = 1$  if vessel  $v \in V_j \forall j \in J$ ,  $= 0$  otherwise

$\beta_v$ : the minimum hire periods of jack-up vessel  $v \in V$  to perform a decommissioning task.

$\lambda_{j,p} = 1$  if  $p \in P_j \forall j \in J$ ,  $= 0$  otherwise

$b_{p,k}$ : the cost of barge/transfer vessel to transport sub-component  $p \in P$  to port  $k \in K$ .

$\varphi_{p,k}$ : the inventory cost per period of unprocessed sub-component  $p \in P$  to port  $k \in K$ .

$\varsigma_{p,k}$ : the inventory cost per period of processed sub-component  $p \in P$  to port  $k \in K$ .

$\pi_{p,k}$ : the processing time (periods) to crush/process sub-component  $p \in P$  at port  $k \in K$ . This parameter is an integer value.

- $\theta_{p,k}$ : the number (capacity) of sub-components  $p \in P$  that can be processed by the machine at port  $k \in K$  per period.
- $m_{p,k}$ : the processing cost of sub-component  $p \in P$  to port  $k \in K$ .
- $\gamma_{p,k,d}$ : the transportation cost to transfer sub-component  $p \in P$  from port  $k \in K$  to dump area  $d \in D$ .
- $\psi_k$ : the maximum amount of components that can be transferred from port  $k \in K$  to recycling/dump areas in one period.
- $q_{t,j}$ : = 1 if there is no planning delay in period  $t \in T$  to perform task  $j \in J$ , = 0 otherwise.
- $w_p$ : the weight of sub-component  $p \in P$ .
- $\epsilon_k$ : the maximum capacity (tonnages) of recycle area at port  $k \in K$ .
- $\epsilon_d$ : the maximum capacity (tonnage) of dump/landfill area  $d \in D$ .

During the planning horizon (for example 90 or 120 days), each potential jack-up vessel has a set of possible slots to hire based on the minimum hire period of the jack-up vessel, the availability of the vessel and the planning delay. A slot consists of a set of periods when a jack-up vessel is hired. For example, a jack-up vessel can be hired to perform a decommissioning task from Period 1 to 7 termed as Slot 1. Another potential slot (Slot 2) for this jack-up vessel to be rented is from Period 1 to 8, and so on. Note that the jack-up vessel needs to be hired in consecutive periods. As the rental cost of the jack-up vessel is relatively expensive, the proposed model is developed mainly based on the jack-up vessel. Here, our strategy is to explore all feasible slots for all potential jack-up vessels to be hired. Then, the best slots need to be selected in order to minimise the total cost. The description of the procedure to generate feasible slots for jack-up vessels to perform dismantling tasks will be presented in the next subsection.

### 3.2 The procedure for generating feasible slots

Generating feasible slots can be done by considering total number of turbines to be removed ( $n$ ), the number of turbines that a jack-up vessel can perform in one period ( $\alpha_{v,j}$ ), the availability of the vessel ( $\rho_{v,t}$ ), the minimum hire periods for the vessel ( $\beta_v$  consecutive periods), and the planning delay ( $q_{t,j}$ ). The set of feasible slots ( $S_{v,j}$  indexed by  $s$ ) of jack-up vessel  $v$  to perform task  $j$  consists of several parameters as follows:

- $h_{v,j,s,t}$ : = 1 if period  $t$  is included in slot  $s$  of jack-up vessel  $v$  to perform task  $j$ , = 0 otherwise.
- $d_{v,j,s}$ : the duration (periods) of slot  $s$  of jack-up vessel  $v$  to perform task  $j$ .
- $u_{v,j,s}$ : the started period of slot  $s$  of jack-up vessel  $v$  to perform task  $j$ .
- $e_{v,j,s}$ : the fixed cost of jack-up vessel  $v$  to perform task  $j$  for slot  $s$ .
- $\eta_{v,j,s}$ : max. number of substructures  $j$  that can be removed by slot  $s$  of jack-up vessel  $v$ .
- $\varrho_{v,j} = |S_{v,j}|$ : the number of feasible slots of (jack-up) vessel  $v$  to perform task  $j$ .

Figure 3 shows a simple example of feasible slots of a jack-up vessel for performing a task within 30 periods (days) where  $n = 30$ ,  $\alpha_{v,j} = 3$  and  $\beta_v = 7$ . The planning delay per period and the unavailability of the vessel are also given in the figure. From Figure 3, it can be seen that a slot has at least 7 periods. The jack-up

vessel can be hired during the planning delay period, however the vessel will not do dismantling activities and so be idle. In case that the vessel is not available due to some reasons, the vessel will not be able to be hired. The maximum duration of slots starting from period 1 is 11 periods as hiring the vessel for 11 periods starting from period 1 will completely finish the task (1 idle period due to planning delay so the number of dismantled turbines is  $10 \cdot 3 = 30 = n$ ). As there is no storage capacity constraint in ports, the maximum duration of feasible slots is based on the capability of a jack-up vessel to dismantle  $n$  turbines. The vessel cannot be hired during periods 13-15 as the vessel is not available. Given that the vessel is also not available in period 21, it will also not be able to be rented in period 16 as the minimum hiring period is 7 (until period 22). The next feasible slots are the ones starting from period 22. For periods 4, 23, 24, and 29, the vessel can be rented but it will be idle due to the planning delay (light grey cell in Figure 3). However, the vessel should not be hired starting from these periods as the dismantling activities cannot be performed in these periods. Periods 25-30 as the starting hired period are not possible as well due to the minimum hired periods.

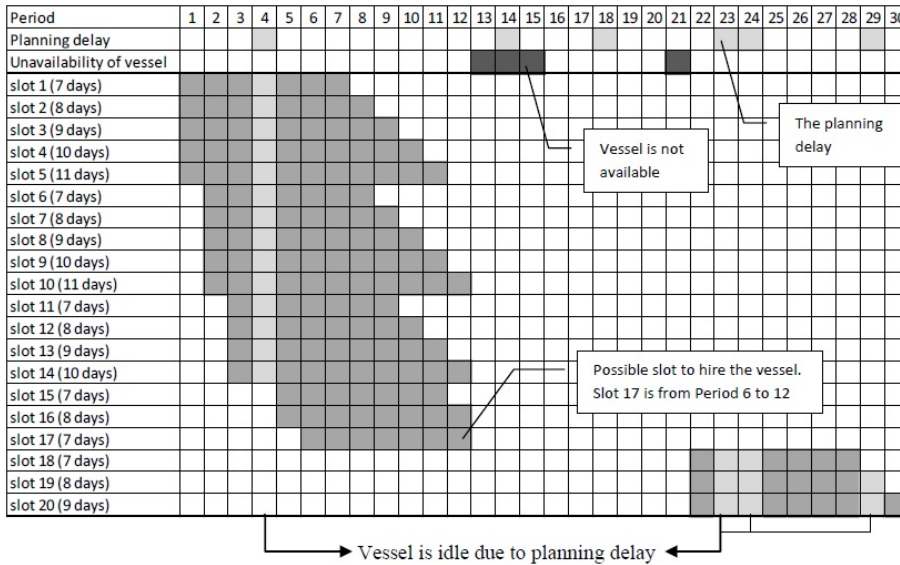


Fig. 3 An example of feasible slots of a jack-up vessel for performing a task

Figure 4 presents the procedure to generate feasible slots. The procedure will populate parameters  $h_{v,j,s,t}$ ,  $d_{v,j,s}$ ,  $e_{v,j,s}$ ,  $\eta_{v,j,s}$  and  $\varrho_{v,j} = |S_{v,j}|$  which will be used for developing the formulation of mathematical model of the decommissioning scheduling problem.

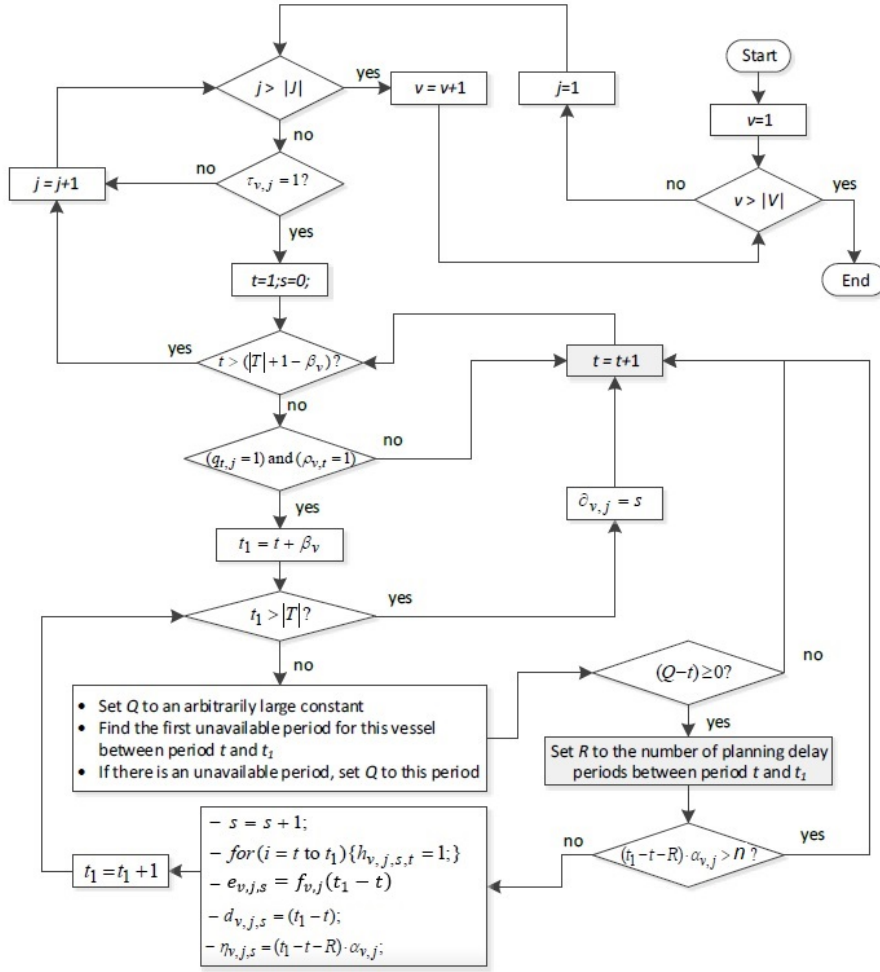


Fig. 4 The procedure for generating feasible slots

### 3.3 Mathematical model

In this subsection, we present a mathematical model using integer linear programming (ILP) for the problem of scheduling the decommissioning of an offshore wind farm.

#### Decision Variables

$$X_{v,j,s} = \begin{cases} 1 & \text{if task } j \text{ is performed in slot } s \text{ by vessel } j, \\ 0 & \text{otherwise} \end{cases}$$

$B_{p,k,t}$ : the number (integer) of sub-component  $p$  transported using barge/service vessels from the wind farm site to port  $k$  in period  $t$ .

$I_{p,k,t}$ : inventory level (integer) of unprocessed sub-component  $p$  at port  $k$  in period  $t$ .

$L_{p,k,t}$ : inventory level (integer) of processed sub-component  $p$  at port  $k$  in period  $t$ .

$M_{p,k,t}$ : amount (integer) of sub-component  $p$  processed at port  $k$  started at period  $t$ .

$Y_{p,k,t}$ : amount (integer) of processed recyclable sub-component  $p \in P^r$  transported to recycling area at port  $k$  in period  $t$ .

$U_{p,k,d,t}$ : amount (integer) of processed non-recyclable sub-component  $p \in P^d$  transported from port  $k$  to dump area  $d$  in period  $t$ .

#### Auxiliary Variable

$G_{j,t}$ : number (integer) of (top/sub) structures (task  $j$ ) dismantled in period  $t$ .

#### Objective functions

$$\text{Minimise Total cost } Z^c = Z^v + Z^b + Z^i + Z^m + Z^l \quad (1)$$

where

$$\text{Jack-up Vessel cost} = Z^v = \sum_{j \in J} \sum_{v \in V_j} \sum_{s \in S_{v,j}} (X_{v,j,s} \cdot (e_{v,j,s} + c_{v,j,s} \cdot d_{v,j,s})) \quad (2)$$

$$\text{Barge/service vessel cost} = Z^b = \sum_{p \in P} \sum_{k \in K} \sum_{t \in T} (B_{p,k,t} \cdot b_{p,k}) \quad (3)$$

$$\text{Inventory cost} = Z^i = \sum_{p \in P} \sum_{k \in K} \sum_{t \in T} (I_{p,k,t} \cdot \varphi_{p,k} + L_{p,k,t} \cdot \varsigma_{p,k}) \quad (4)$$

$$\text{Processing cost} = Z^m = \sum_{p \in P} \sum_{k \in K} \sum_{t \in T} (M_{p,k,t} \cdot m_{p,k}) \quad (5)$$

$$\text{On-land transportation cost} = Z^l = \sum_{p \in P^d} \sum_{k \in K} \sum_{d \in D} \sum_{t \in T} (U_{p,k,d,t} \cdot \gamma_{p,k,d}) \quad (6)$$

#### Constraints

Constraints (7) ensure that there are no overlapping slots allocated to a jack-up vessel.

$$\sum_{j \in J} \sum_{s \in S_{v,j}} (X_{v,j,s} \cdot h_{v,j,s,t}) \leq 1, \quad \forall v \in V, t \in T \quad (7)$$

Constraints (8) guarantee that in one period, the capacity of jack-up vessels for removing the components must be greater than the number of components transported by barge/service vessel to the ports.

$$\sum_{j \in J} \sum_{s \in S_{v,j}} (X_{v,j,s} \cdot h_{v,j,s,t} \cdot \alpha_{v,j} \cdot q_{t,j}) \geq G_{j,t}, \quad \forall j \in J, t \in T \quad (8)$$

Constraints (9) indicate that in the first period, any successor task (e.g. dismantling a sub-structure) cannot be performed.

$$G_{j,1} = 0, \quad \forall j = 2, \dots, |J| \quad (9)$$

Constraints (10) ensure that in period  $t$  the sum of removed components of job  $(j + 1)$  does not exceed the sum of removed components of job  $j$ .

$$\sum_{t=1}^{t'-1} G_{j,t} \geq \sum_{t=1}^{t'} G_{\hat{j},t}, \forall j = 1, \dots, (|J| - 1); \hat{j} = (j + 1), \dots, |J|; t' = 2, \dots, |T| \quad (10)$$

Constraints (11) make sure that all turbines (top/sub structures) are removed in the planning horizon.

$$\sum_{t \in T} G_{j,t} = n, \quad \forall j \in J \quad (11)$$

Constraints (12) ensure that the number of sub-components transported to ports must be the same as the number of (top/sub) structures.

$$\sum_{k \in K} B_{p,k,t} = G_{j,t}, \quad \forall j \in J, p \in P_j, t \in T \quad (12)$$

Constraints (13) state that the inventory of sub-component  $p$  in the first period is the same as the amount of sub-component  $p$  transported by barge/transfer vessel to the port.

$$I_{p,k,1} = B_{p,k,1}, \quad \forall p \in P, k \in K \quad (13)$$

Constraints (14) enforce the inventory level of sub-component  $p$  at port  $k$  during period  $t$ .

$$I_{p,k,t} = I_{p,k,t-1} + B_{p,k,t} - M_{p,k,t}, \quad \forall p \in P, k \in K, t = 2, \dots, |T| \quad (14)$$

Constraints (15) ensure that the number of sub-component  $p$  to be processed must be equal or less than the inventory level of this sub-component in the previous period.

$$I_{p,k,t-1} \geq M_{p,k,t}, \quad \forall p \in P, k \in K, t = 2, \dots, |T| \quad (15)$$

Constraints (16) describe that there are no processing activities in the first period.

$$M_{p,k,1} = 0, \quad \forall p \in P, k \in K \quad (16)$$

Constraints (17) ensure that the number of processed components cannot exceed the capacity of the machine.

$$\sum_{t=t'-\hat{t}_{p,k}}^{t'-1} M_{p,k,t} \leq \theta_{p,k}, \forall p \in P, k \in K, t' = (\hat{t}_{p,k} + 1), \dots, (|T| + 1) \quad (17)$$

Constraints (18) guarantee that there is no inventory of processed components in the period that is less than the processing time.

$$I'_{p,k,t} = 0, \quad \forall p \in P, k \in K, t = 1, \dots, \hat{t}_{p,k} \quad (18)$$

Constraints (19) state the inventory level of processed recyclable sub-component  $p$  at the port in period  $t$ .

$$L_{p,k,t} = L_{p,k,t-1} + M_{p,k,t-\pi_{p,k}} - Y_{p,k,t}, \forall p \in P^r, k \in K, t = (\pi_{p,k} + 1), \dots, |T| \quad (19)$$

Constraints (20) state the inventory level of processed non-recyclable sub-component  $p$  at the port in period  $t$ .

$$L_{p,k,t} = L_{p,k,t-1} + M_{p,k,t-\pi_{p,k}} - \sum_{d \in D} U_{p,k,d,t}, \forall p \in P^d, k \in K, t = (\pi_{p,k} + 1), \dots, |T| \quad (20)$$

Constraints (21) ensure that the number of recyclable sub-components transported to a port must be the same as the number transported to the recycling area from this port.

$$\sum_{t \in T} Y_{p,k,t} = \sum_{t \in T} B_{p,k,t}, \quad \forall p \in P^r, k \in K \quad (21)$$

Constraints (22) ensure that the number of non-recyclable sub-components transported to a port must be the same as the number transported to the dump area from this port.

$$\sum_{d \in D} \sum_{t \in T} U_{p,k,d,t} = \sum_{t \in T} B_{p,k,t}, \quad \forall p \in P^d, k \in K \quad (22)$$

Constraints (23) guarantee that all processed recyclable sub-components are transported to the recycling area.

$$\sum_{k \in K} \sum_{t \in T} Y_{p,k,t} = n, \quad \forall p \in P^r \quad (23)$$

Constraints (24) guarantee that all processed non-recyclable sub-components are transported to the dump/landfill area.

$$\sum_{k \in K} \sum_{d \in D} \sum_{t \in T} U_{p,k,d,t} = n, \quad \forall p \in P^d, k \in K \quad (24)$$

Constraints (25) ensure that in one period, the number of components transferred from port  $k$  to recycling/dump areas does not exceed the maximum level.

$$\sum_{p \in P^r} Y_{p,k,t} + \sum_{d \in D} \sum_{p \in P^d} U_{p,k,d,t} \leq \psi_k, \quad \forall k \in K, t \in T \quad (25)$$

Constraints (26) and (27) make sure that the recyclable and non-recyclable sub-components' movement occurs once the component has been processed respectively.

$$Y_{p,k,t} = 0, \quad \forall p \in P^r, k \in K, t = 1, \dots, \pi_{p,k} \quad (26)$$

$$U_{p,k,d,t} = 0, \quad \forall p \in P^d, k \in K, d \in D, t = 1, \dots, \pi_{p,k} \quad (27)$$

Constraints (28) and (29) indicate the capacity constraint for each recycling area and each dump/land fill area respectively.

$$\sum_{p \in P^r} \sum_{t \in T} (Y_{p,k,t} \cdot w_p) \leq \epsilon_k, \quad \forall k \in K \quad (28)$$

$$\sum_{k \in K} \sum_{p \in P^r} \sum_{t \in T} (U_{p,k,d,t} \cdot w_p) \leq \epsilon_d, \quad \forall d \in D \quad (29)$$



Constraints (30) indicate that variable  $X$  is binary

$$X_{v,j,s} \in \{0, 1\}, \quad \forall v \in V, j \in J, s \in S_{v,j} \quad (30)$$

Constraints (31) – (33) state the non-negativity and integer variable conditions.

$$G_{j,t} \geq 0, \text{ integer}, \quad \forall j \in J, t \in T \quad (31)$$

$$B_{p,k,t}, I_{p,k,t}, L_{p,k,t}, M_{p,k,t}, Y_{p,k,t} \geq 0, \text{ integer}, \quad \forall p \in P, k \in K, t \in T \quad (32)$$

$$U_{p,k,d,t} \geq 0, \text{ integer}, \quad \forall p \in P, k \in K, d \in D, t \in T \quad (33)$$

#### 4 Matheuristic for the decommissioning problem

Based on preliminary experiments, an exact method solution (using CPLEX) finds difficulties in solving the proposed ILP as CPLEX terminates before obtaining the optimal solution due to being out of memory. We have carried out experiments on some instances (by varying  $N$ ,  $|V|$ ,  $\beta_v$  and planning delays) without limitation but CPLEX terminates after running more than 10 hours due to being out of memory. We also noticed that there is no significant improvement on the %Gap produced until CPLEX terminated compared to the one within 3 hours. In other words, we were not able to obtain the optimal solution for this problem. It is also noted that the result (the upper bound value) produced by CPLEX within a certain time (3 hours) is not promising at this point. Transforming the ILP problem into the MILP problem by relaxing the integer variables has also been conducted. However, the results generated by solving the MILP problem are also not promising. The report for the computational experiments can be seen in Section 5. Therefore, we propose a matheuristic technique to overcome this limitation. Moreover, the sensitivity analysis by varying some parameters is possible to be conducted if solutions for the problem can be generated quickly.

Matheuristics are the hybridization of metaheuristics and exact procedures (Maniezzo et al, 2010). For example, Schwerdfeger et al (2018) and Garcia and Rabadi (2013) used this method successfully for solving scheduling problems. In our study, the proposed method for solving the scheduling problem of the decommissioning of an offshore wind farm benefits from the relaxed problem principle and an interchange heuristic to explore the solution space. The underlying principle of the relaxed problem is to transform some binary or integer decision variables to continuous in order to make the problem easier to solve. We also take benefit from the solution methods based on Variable Neighbourhood Search (VNS) and Simulated Annealing (SA) proposed by Irawan et al (2017a) to solve the installation scheduling problem in offshore wind farm. Both VNS and SA are designed for finding the optimal schedule (slots) for jack-up vessels to install the offshore wind turbines. Here, we propose a novel/different technique based on the principle of the relaxed problem and the application of local searches to fit with the nature of the decommissioning problem in offshore wind farm. The procedure of our proposed method is depicted in Algorithm 1 which consists of four stages. As the chartering vessel cost dominates the decommissioning cost, the proposed method focuses on finding the best slots to be used by a jack-up vessel. The other decision variables such as the flows of components and inventory levels are determined once based on the obtained slots configuration.

The first stage aims to obtain the promising slots by relaxing the proposed model (1-33) where the data type of decision variable  $X_{v,j,s}$  is transformed from binary to continuous with  $0 \leq X_{v,j,s} \leq 1$ . The problem is hence transformed to a Mixed Integer Linear Programme (MILP). The relaxed MILP scheduling problem is then solved using an exact method (via CPLEX) within  $\hat{t}_1$  seconds.  $\%Gap$  as a termination criterion for CPLEX to solve the relaxed MILP is also added which is set to  $\varepsilon_1\%$ . Let  $\tilde{F}_{v,j}$  be the set of slots of vessel  $v$  to perform task  $j$  where  $X_{v,j,s} > 0, \forall v \in V, j \in J, s \in S_{v,j}$  in the obtained solution. We consider the slots in  $\tilde{F}_{v,j}$  as promising slots. The larger value of  $X_{v,j,s}$  indicates the more promising slot in minimising the total cost. However, these slots in  $\tilde{F}_{v,j}$  cannot be treated as a solution of this problem as there may be overlapping slots for a jack-up vessel. To make sure that each selected jack-up vessel, with  $X_{v,j,s} > 0$ , has significant promising slots for performing a decommissioning task, Constraints (34) are added into the model:

$$\sum_{s \in S_{v,j}} X_{v,j,s} \geq 1, \quad \forall v \in V, j \in J \quad (34)$$

The second stage reduces the problem of Model (1)-(33) by replacing the slot set  $S_{v,j}$  with  $\tilde{F}_{v,j}, \forall v \in V, j \in J$  obtained from the previous stage. Moreover, to make the problem easier to solve, the model is also relaxed by converting the data type of all variables (except  $X_{v,j,s}$ ) from integer to continuous. As variable  $X_{v,j,s}$  is still a binary the problem becomes a reduced MILP. This reduced MILP is then solved by CPLEX with a stopping criterion of  $\hat{t}_2$  seconds or the  $\%Gap \leq \varepsilon_2\%$ . Let  $F_{v,j} (\forall v \in V, j \in J)$  be the set of slots with  $X_{v,j,s} = 1$  in the obtained solution. In set  $F_{v,j}$ , there is no overlapping slot for each selected jack-up vessel which is considered as an incumbent slots configuration (solution). This solution is then fed into the third stage as a relatively good initial solution.

Algorithm 1: The matheuristic for the decommissioning problem

*Initialization:*

- (a) Define the parameters  $\hat{t}_1, \varepsilon_1, \alpha, \hat{t}_2, \varepsilon_2, \beta$  and  $\hat{t}_3$ .

*Stage 1:*

- (a) Relax Model (1)-(33) by transforming the data type of decision variable  $X_{v,j,s}$  from binary to continuous variable with  $0 \leq X_{v,j,s} \leq 1$ . In addition, Equation (34) is added to make sure that each selected vessel has significant promising slots for performing a decommissioning task.
- (b) Solve the relaxed mixed integer scheduling problem using exact method (CPLEX) within  $\hat{t}_1$  seconds. Moreover, we add  $\%Gap$  as a termination criterion for CPLEX to solve the relaxed MILP which we set  $\%Gap$  to  $\varepsilon_1\%$ .
- (c) Let  $\tilde{F}_{v,j}$  be the set of slots where  $X_{v,j,s} > 0, \forall v \in V, j \in J, s \in S_{v,j}$ .

*Stage 2:*

- (a) Replace set  $S_{v,j}$  with  $\tilde{F}_{v,j}$  obtained from the previous stage in Model (1)-(33) so the size of the problem is reduced. In addition, relax the model by transforming the data type of all variables (except  $X_{v,j,s}$ ) from integer to continuous.
- (b) Solve the reduced MILP scheduling problem using CPLEX within  $\hat{t}_2$  seconds. CPLEX will also terminate when the gap reached  $\varepsilon_2\%$ .
- (c) Let array  $F_{v,j} (\forall v \in V, j \in J)$  be the obtained slots with  $X_{v,j,s} = 1$  and denote  $Z$  be its objective function value.

*Stage 3:*

- (a) Apply proposed local search 1 (given in Algorithm 2) with  $Z$  and  $F_{v,j}$  obtained from Stage 2 as inputs and outputs.
- (b) Implement proposed local search 2 (presented in Algorithm 3) with  $Z$  and  $F_{v,j}$  obtained from Stage 3a) as inputs and outputs.

*Stage 4:*

- (a) Replace set  $S_{v,j}$  with  $F_{v,j}$  obtained from the previous stage in Model (1)-(33).
- (b) Solve the reduced ILP scheduling problem using CPLEX within  $\hat{t}_3$  seconds with  $Z$  be its objective function value. In addition, the outputs obtained from this problem are  $X_{v,j,s}$ ,  $G_{j,t}$ ,  $B_{p,k,t}$ ,  $I_{p,k,t}$ ,  $L_{p,k,t}$ ,  $M_{p,k,t}$ ,  $Y_{p,k,t}$  and  $U_{p,k,d,t}$ .

In Stage 3, two local searches are proposed to improve the quality of the incumbent solution. In the first proposed local search, an iterative process is conducted where additional slots for each decommissioning task are pseudo randomly added into the incumbent solution ( $F_{v,j}$ ) forming a set of promising slots ( $A_{v,j}$ ). In the second proposed local search, an interchange heuristic is applied where the algorithm seeks a potential slot to replace one or more slots in the incumbent solution ( $F_{v,j}$ ). The detailed explanation of these proposed local searches is presented in the next subsection.

Stage 4 is the final stage where the ILP model (1)-(33) is reduced by replacing set  $S_{v,j}$  with  $F_{v,j}$  ( $\forall v \in V, j \in J$ ) obtained from the previous stage. The reduced ILP is solved by an exact method (CPLEX) within a time limit of  $\hat{t}_3$  seconds. Note that in this stage the model is not relaxed. However, the problem in this stage is relatively easy to solve as the size of the problem is relatively small comprising only the slots used in the incumbent solution. The outputs of this stage will be the final result of the proposed matheuristic which includes  $X_{v,j,s}$ ,  $G_{j,t}$ ,  $B_{p,k,t}$ ,  $I_{p,k,t}$ ,  $L_{p,k,t}$ ,  $M_{p,k,t}$ ,  $Y_{p,k,t}$  and  $U_{p,k,d,t}$ .

*Local Search 1*

In this subsection, the first proposed local search is described where the initial solution/slots configuration ( $F_{v,j}$ ) is obtained from Stage 2. Algorithm 2 presents the steps of Local Search 1 where in the first step, the earliest day (period) for conducting each decommissioning task ( $\tilde{t}_j, \forall j \in J$ ) is determined. Let  $\tilde{A}_{v,j}$  be a set of promising slots for vessel  $v \in V_j$  to perform task  $j \in J$ . Set  $\tilde{A}_{v,j}$  is initially populated by the incumbent solution ( $F_{v,j}$ ) as shown in Step 2.

*Algorithm 2: The proposed local search 1*

*Input:*  $Z$  and  $F_{v,j}$ .

*Initialization:* Define parameters  $\hat{t}_4$  and  $\varepsilon_3$ .

*Main Step:*

1. Find the earliest period to start performing task  $j$  ( $\tilde{t}_j, \forall j \in J$ ) in the incumbent solution ( $F_{v,j}$ ).
2. Set  $\tilde{A}_{v,j} = F_{v,j}$ .
3. For each  $j \in J$  do the following steps:

- 
- (a) Set  $\hat{n} = 0$ .
  - (b) While  $\hat{n} < n$  perform the followings:
    - i. Pick randomly vessel  $\hat{v}$  and its slot  $\hat{s}$  to perform task  $j$ .
    - ii. If  $(u_{\hat{v},j,\hat{s}} < \tilde{t}_j)$  go back to Step 3(b)i.
    - iii. Include slot  $\hat{s}$  of vessel  $\hat{v}$  in set  $\tilde{A}_{v,j}$ .
    - iv. Update  $\hat{n} = \hat{n} + \eta_{\hat{v},j,\hat{s}}$
    - v. If  $(u_{\hat{v},j,\hat{s}} < \tilde{t}_j)$ , set  $\tilde{t}_j = u_{\hat{v},j,\hat{s}}$ .
  4. Reduce the problem (Model 1-33) by replacing set  $S_{v,j}$  with  $\tilde{A}_{v,j}, \forall v \in V, j \in J$ .  
In addition, relax the model by transforming the data type of all variables (except  $X_{v,j,s}$ ) from integer to continuous (MILP).
  5. Solve the MILP using exact method (CPLEX) until the %Gap reached  $\varepsilon_3\%$ .  
Let  $\hat{F}_{v,j} (\forall v \in V, j \in J)$  be the obtained slots and  $\hat{z}$  be its objective function value.
  6. If  $\hat{z} < Z$ , update  $Z = \hat{z}$  and  $F_{v,j} = \hat{F}_{v,j}$ .
  7. If CPU time  $< \hat{t}_4$  go back to Step 1
  8. Return  $Z$  and  $F_{v,j}$ .
- 

In Step 3, randomly chosen slots of randomly chosen vessels are then added into the set of promising slots for each decommissioning task ( $\tilde{A}_{v,j}$ ). The number of selected slots to be added for each decommissioning task is limited. The additional slots for each task must be able to remove  $n$  structures/turbines. Moreover, when choosing randomly a slot (from a randomly selected jack-up vessel) for a sub-structure task, the chosen slot must start after the earliest period of dismantling the top-structure ( $u_{\hat{v},j,\hat{s}} < \tilde{t}_j$ ). Otherwise, this slot will not improve the solution as hiring a jack-up vessel for performing a sub-structure task with no top-structure has been removed is a waste.

Model (1)-(33) is then populated by the promising slots ( $\tilde{A}_{v,j}$ ) instead of all possible slots ( $S_{v,j}$ ). To make the problem easier to solve, we also relax the model by transforming the data type of all variables (except  $X_{v,j,s}$ ) from integer to continuous (to become a MILP). The MILP is then addressed by an exact method (CPLEX) until the %Gap reached  $\varepsilon_3\%$ . If the obtained solution is better than the incumbent solution then update the incumbent solution. This procedure is repeated until the computing time reached prescribed time ( $\hat{t}_4$ ). The best solution (slots configuration) attained from this local search is then fed into the second proposed local search which is described in the next subsection.

### Local Search 2

Local Search 2 is proposed based on the interchange heuristic where the main steps of this local search is presented in Algorithm 3. In the first step, the slots in the incumbent solution ( $F_{v,j}$ ) for each task are ordered in increasing number by the number of structures that can be removed by each slot ( $\eta_{v,j,s}$ ). These ordered slots are stored in array  $S'_j$ . An array  $V'_j$  is also constructed to give information which vessel is used to perform a decommissioning task for a slot in set  $S'_j$ . In other words, slot  $s' \in S'_j$  is performed by vessel  $v' \in V'_j$  with  $s' = v'$ .

Step 2 of Algorithm 3 seeks all possible slots to be swapped with slots in the incumbent solutions ( $F_{v,j}$ ) for each decommissioning task. The slots in  $F_{v,j}$  to

be swapped are represented by set  $\Omega(\Omega \subseteq S'_j)$  where the structure of set  $\Omega$  is presented in Figure 5. Steps 2a-2c are iterative processes based on the elements in set  $\Omega$ . The process of the algorithm uses a top-bottom procedure. In the first level, for each decommissioning task, a potential chosen slot  $s \in S_{v,j}(s \notin F_{v,j})$  is swapped with one slot in the incumbent solution where  $|\Omega| = 1$ . In the last level, this potential chosen slot replaces  $|S'_j|$  slots in the incumbent solution if it is feasible. It starts from the first level and then descends to the next level where the swap process in the higher level is feasible. In other words, it starts inserting slot  $s \in S_{v,j}(s \notin F_{v,j})$  into the incumbent solution ( $F'_{v,j}$ ) and removing a slot (element of set  $\Omega$ ) from  $F'_{v,j}$  (for example  $\Omega = \{s_1\}$ ), and if it is feasible (with no improvement) then the algorithm adds another slot  $\Omega = \{s_1, s_2\}$  to be swapped with slot  $s$ . If the solution  $F'_{v,j}$  is not feasible, set  $\Omega$  is populated by  $\{s_2\}$  instead of  $\{s_1, s_2, s_3\}$ . This process is repeated until all possible swapping of slots are explored or the computing time has reached  $\hat{t}_5$  seconds.

*Algorithm 3: The proposed local search 2*

*Input:*  $Z$  and  $F_{v,j}$ .

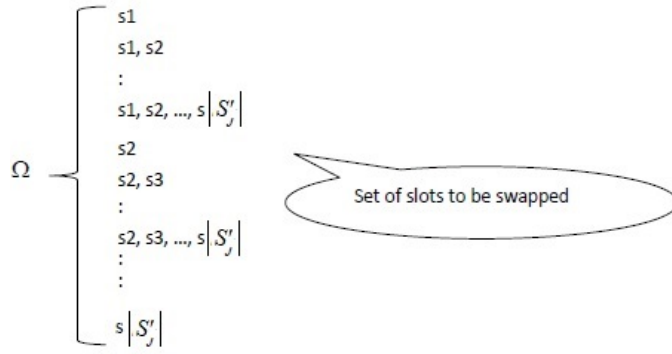
*Initialization:* Define parameter  $\hat{t}_5$ .

*Main Step:*

1. Let array  $S'_j(j \in J)$  as a set of slots in  $F_{v,j}$  ordered by the number of structures that can be removed by the slots (for task  $j$ ). Denote array  $V'_j(j \in J)$  be a set of vessels used for performing task  $j$  for slots in  $S'_j$ . In other words, slot  $s' \in S'_j$  is performed by vessel  $v' \in V'_j$  with  $s' = v'$ .
2. While CPU time  $< \hat{t}_5$  do the following:
  - For each vessel  $v \in V$  do the following:
    - For each task  $j \in J$  do the following:
      - For each slot  $s \in S_{v,j}(s \notin F_{v,j})$  do iteratively the following steps:
        - a. Set  $F'_{v,j} \leftarrow F_{v,j}$ .
        - b. Insert slot  $s$  into  $F'_{v,j}$  and remove a subset of slots ( $\Omega$ ) from  $F'_{v,j}$  where  $\Omega \subseteq S'_j$
        - c. If solution  $F'_{v,j}$  can remove  $N$  structures (turbines) and the cost of slot  $s$  less than total cost of slots in set  $\Omega$  and there is no overlapping slots for each vessel in set  $F'_{v,j}$  then do the following steps:
          - i. Solve model (1)-(33) with fixed slots ( $F'_{v,j}$ ) and continuous decision variables (linear programming)
          - ii. Let  $z'$  be its objective function value
          - iii. If the problem has a feasible solution and  $z' < Z$  then update  $Z = z'$  and  $F_{v,j} \leftarrow F'_{v,j}$  and go back to Step 1.
3. Return  $Z$  and  $F_{v,j}$ .

---

When slot  $s$  is swapped with slots (elements of  $\Omega$ ), the new solution ( $\hat{F}_{v,j}$ ) must be able to remove  $n$  substructures/turbines with no overlapping among the slots in the new solution. The cost of slot  $s$  must also be less than the total cost



**Fig. 5** A set of slots in incumbent solution ( $\Omega$ ) to be swapped

of slots ( $\Omega$ ) to be removed from the current solution. The feasibility of the new solution along with its objective function value is evaluated by solving Model (1)-(33) based on the new solution with some revisions. First, the slots in the new solution ( $\hat{F}_{v,j}$ ) must be selected (fixed) so variable  $X_{v,j,s}$  becomes a parameter. In other words,  $X_{v,j,s}$  is set to 1 if slot  $s$  exists in set ( $\hat{F}_{v,j}$ ), otherwise  $X_{v,j,s}$  is set to 0. Second, the data type of the other variables is transformed to continuous. Therefore, the model becomes a continuous linear programming problem which is relatively easy to solve by an exact method (CPLEX). In this local search, first improvement strategy is applied where the interchange between a potential slot and slots in the current slots is performed if improvement occurs where it will go back again to Step 1.

## 5 Computational study

Computational experiments have been carried out to assess the performance of the proposed solution method. The code was written in C++ .Net 2012 and used the IBM ILOG CPLEX version 12.6 Concert Library. The tests were run on a PC with an Intel Core i5 CPU @ 3.20GHz processor, 8.00 GB of RAM and under Windows 7. In the computational experiments, a dataset consisting of several instances is constructed based on the West Gabbard (UK) offshore wind farm site to assess our approach to solving the decommissioning model. This dataset is representative for the generation of offshore wind farms that is currently being developed. Three solution methods are used to solve the problem as follows:

### – Exact Method

IBM ILOG CPLEX version 12.6 was used to solve the proposed decommissioning model (Equations 1-33). The computational time (CPU) in solving the problem using the exact method (CPLEX) is limited to three hours as the problem is very hard to solve to optimality by the exact method. By limiting the computational time in CPLEX, the lower bound ( $LB$ ) and upper bound ( $UB$ ) can be attained.

### – 2-stage Exact Method

This solution method also uses CPLEX for solving the problem. This method consists of two stages as follows:

- i. Use CPLEX to solve the proposed decommissioning model (Equations 1-33) within a certain computational time (e.g. 3 or 24 hours). However, all integer decision variables except  $X_{v,j,s}$  are transformed to continuous. The problem is now reduced to the Mixed Integer Linear Programming (MILP) problem instead of the ILP problem. Therefore, the problem is relatively easier to solve.
- ii. The solution obtained from the previous stage may not be feasible as the number of (sub-) components, that need to be transported, has to be integer instead of continuous. Therefore, in this stage, Equations 1-33 is optimally solved using CPLEX by fixing decision variable  $X_{v,j,s}$  that has been obtained from previous stage. In other word,  $X_{v,j,s}$  becomes a parameter and other decision variables are treated as integer. Here, the problem become a relatively small ILP problem and the feasible solution can be obtained.

– **The proposed matheuristic**

In the matheuristic approach, the value of parameters are set as follows:  $\hat{t}_1 = \hat{t}_2 = \hat{t}_4 = \hat{t}_5 = 8 \cdot n^{2.25}$  seconds,  $\varepsilon_1 = 5\%$ ,  $\varepsilon_2 = 1\%$ ,  $\varepsilon_3 = 0.5\%$  and  $\hat{t}_3 = 50$  seconds. The computational times  $\hat{t}_3$  (the final stage) is set to a relatively small value as in this stage, the exact method (CPLEX) only deals with a relatively small problem. Those values are determined based on the preliminary study.

The performance of the solution methods will be measured by  $\%Gap$  between the  $Z$  value attained by the method and the lower bound ( $LB$ ) obtained from the exact method.  $\%Gap$  is calculated as follows:

$$Gap(\%) = \frac{Z_m - LB}{LB} \times 100 \quad (35)$$

where  $Z_m$  refers to the feasible solution cost obtained by either the exact method ( $UB$ ), 2-Stage exact method or the proposed matheuristic approach. The following subsections will describe the data used for the scheduling decommissioning model followed by the results of the proposed solution method.

### 5.1 Data

The West Gabbard wind farm site, located in the North Sea off the East coast of the UK, has been chosen as the test case for the models and methods developed in this paper. This is because it is a typical of the largest currently existing wind farms which will need decommissioning in 20-25 years' time. Figure 6 shows the location of wind farm site, candidate ports, and landfill areas included whereas Table 1 presents their coordinates.

A dataset is generated to evaluate the performance of the proposed solution methods. The dataset consists of several instances by varying some parameters used in the model. We first divided the instances based on the number of planning delays with either (i) no planning delay or (ii) two days planning delay occurs. The number of jack-up vessels ( $|V|$ ) considered to be hired for dismantling the offshore wind turbines is varied from 5 to 7. We set the minimum hire period for each jack vessel ( $\beta_v$ ) to 7 and 14 consecutive days. We also vary the number of turbines ( $n$ ) that need to be removed from 110 to 160 with an increment of 10. The turbine range represents the largest currently operational wind farms that are likely to



**Fig. 6** The map of the wind farm site, ports, and landfill areas (source: Google Maps)

**Table 1** The coordinates of wind farm, ports, and landfill areas

| Location                         | Latitude  | Longitude |
|----------------------------------|-----------|-----------|
| Wind Farm Site                   |           |           |
| West Gabbard site                | 51.98     | 2.08      |
| Location of Ports                |           |           |
| Oostende (Port 1)                | 51.2333   | 2.91667   |
| Able UK (Port 2)                 | 53.65     | -0.24     |
| Hull (Port 3)                    | 53.74     | -0.29     |
| Harwich (Port 4)                 | 51.95     | 1.28333   |
| Great Yarmouth (Port 5)          | 52.6167   | 1.73333   |
| Location of Landfill areas       |           |           |
| Gallymoor Landfill Site (Dump 1) | 53.848669 | -0.725156 |
| Winterton Landfill Site (Dump 2) | 53.642174 | -0.626461 |
| Immingham Landfill Site (Dump 3) | 53.61909  | -0.177185 |
| Dispit Ltd (Dump 4)              | 53.766963 | -0.462998 |
| Aldeby Landfill Site (Dump 5)    | 52.476521 | 1.629964  |
| Remo Milieubeheer (Dump 6)       | 51.057738 | 5.348292  |

need decommissioning in 20-25 years time. The time period is measured in days where in this study the number of periods required to dismantle the turbines is set to  $n$  days. A preliminary study shows that for  $n = 160$ , a feasible solution can be attained when  $|T| \geq 158$ . Therefore, it is assumed that at least one turbine can be removed within one day.



All ports have a recycling facility, meaning that the recycling area will be at the port. In the experiments, two tasks are considered, namely top- and sub-structure removal which are referred to as Task 1 and Task 2 respectively. The data relating to the vessels, the ports and dump/landfill area for the instance with no planning delay are given in Tables 2, 3 and 4 respectively. Tables 5 and 6 present the sub-component data for the instance with no planning delay which is randomly generated. There are four sub-components considered in this study; namely tower, nacelle, blade, and foundation. In Table 6, the sub-component data relating to ports such as processing time, capacity and cost along with inventory cost for each component is given.

**Table 2** The jack-up vessel data for the instance with no planning delay

| Vessel | $\alpha_{v,j}$ |        | $f_{v,j}$ (€) |        | $c_{v,j}$ (€) |         |
|--------|----------------|--------|---------------|--------|---------------|---------|
|        | Task 1         | Task 2 | Task 1        | Task 2 | Task 1        | Task 2  |
| ves 1  | 3              | 2      | 2,000         | 2,000  | 80,000        | 80,000  |
| ves 2  | 4              | 2      | 2,500         | 2,500  | 100,000       | 90,000  |
| ves 3  | 4              | 3      | 3,000         | 3,000  | 100,000       | 100,000 |
| ves 4  | 5              | 3      | 3,500         | 3,500  | 110,000       | 100,000 |
| ves 5  | 5              | 4      | 4,000         | 4,000  | 110,000       | 110,000 |
| ves 6  | 3              | 2      | 2,000         | 2,000  | 80,000        | 80,000  |
| ves 7  | 4              | 3      | 3,000         | 3,000  | 100,000       | 100,000 |

It is worth noting that the data structure for the dataset with two days planning delay is the same as the one with no planning delay presented in Tables 2-6. However, in dataset with two days planning delay, the cost data (jack-up vessels, ports, etc.) is slightly higher than the one with no planning delay. With a two-day planning delay, the total number of possible slots of jack-up vessels generated for the dataset with two days planning delay is slightly smaller than the one with no planning delay.

**Table 3** The parameter values relating to ports for the instance with no planning delay

|                           | Port 1 | Port 2 | Port 3 | Port 4 | Port 5 |
|---------------------------|--------|--------|--------|--------|--------|
| $\psi_k$                  | 9      | 8      | 7      | 6      | 5      |
| Distance to site (km)     | 104    | 268    | 276    | 55     | 71     |
| Recycling Capacity (tons) | 22,000 | 12,000 | 17,000 | 12,000 | 27,000 |

**Table 4** The dump/landfill capacity for the instance with no planning delay

|                           | Dump 1 | Dump 2 | Dump 3 | Dump 4 | Dump 5 | Dump 6 |
|---------------------------|--------|--------|--------|--------|--------|--------|
| Recycling Capacity (tons) | 5,000  | 6,000  | 4,500  | 6,000  | 4,000  | 3,000  |

**Table 5** Sub-component data for the instance with no planning delay

| Sub-component | Task | Recycle-able | Barge cost (€/km) | Weight (ton) |
|---------------|------|--------------|-------------------|--------------|
| Tower         | 1    | Yes          | 200               | 200          |
| Nacelle       | 1    | No           | 150               | 88           |
| Blade         | 1    | No           | 180               | 50           |
| Foundation    | 2    | Yes          | 300               | 300          |

**Table 6** Sub-component data relating to ports for the instance with no planning delay

|        | Tower       |                |               |                     |                       | Nacelle     |                |               |                     |                       |
|--------|-------------|----------------|---------------|---------------------|-----------------------|-------------|----------------|---------------|---------------------|-----------------------|
|        | $\pi_{p,k}$ | $\theta_{p,k}$ | $m_{p,k}$ (€) | $\varphi_{p,k}$ (€) | $\varsigma_{p,k}$ (€) | $\pi_{p,k}$ | $\theta_{p,k}$ | $m_{p,k}$ (€) | $\varphi_{p,k}$ (€) | $\varsigma_{p,k}$ (€) |
| Port 1 | 2           | 4              | 10,000        | 1,000               | 500                   | 1           | 3              | 8,000         | 750                 | 400                   |
| Port 2 | 3           | 2              | 10,000        | 1,100               | 500                   | 2           | 3              | 9,000         | 800                 | 500                   |
| Port 3 | 3           | 1              | 11,000        | 1,100               | 600                   | 2           | 1              | 9,000         | 800                 | 600                   |
| Port 4 | 4           | 1              | 11,000        | 1,200               | 700                   | 3           | 1              | 10,000        | 900                 | 600                   |
| Port 5 | 4           | 1              | 12,000        | 1,200               | 700                   | 3           | 1              | 10,000        | 900                 | 600                   |

|        | Tower       |                |               |                     |                       | Nacelle     |                |               |                     |                       |
|--------|-------------|----------------|---------------|---------------------|-----------------------|-------------|----------------|---------------|---------------------|-----------------------|
|        | $\pi_{p,k}$ | $\theta_{p,k}$ | $m_{p,k}$ (€) | $\varphi_{p,k}$ (€) | $\varsigma_{p,k}$ (€) | $\pi_{p,k}$ | $\theta_{p,k}$ | $m_{p,k}$ (€) | $\varphi_{p,k}$ (€) | $\varsigma_{p,k}$ (€) |
| Port 1 | 1           | 6              | 10,000        | 1,100               | 600                   | 3           | 5              | 15,000        | 1,000               | 500                   |
| Port 2 | 1           | 5              | 12,000        | 1,100               | 700                   | 3           | 4              | 16,000        | 1,100               | 600                   |
| Port 3 | 2           | 4              | 11,000        | 1,200               | 700                   | 4           | 3              | 15,000        | 1,300               | 700                   |
| Port 4 | 3           | 4              | 12,000        | 1,300               | 750                   | 5           | 2              | 16,000        | 1,500               | 700                   |
| Port 5 | 4           | 2              | 13,000        | 1,300               | 800                   | 5           | 1              | 17,000        | 1,700               | 750                   |

## 5.2 Results

Table 7 presents an example of the schedule of the jack-up vessels for the problem with  $n = 110$  when  $|V| = 6$ ,  $PD=0$ , and  $\beta_v = 7$  generated by the exact method, the 2-stage exact method and the proposed matheuristic. Those schedules are feasible solutions where all constraints are satisfied such as the sequence of task, the capacity of barge vessels, the capacity of processing equipment in ports, etc. For this case, the solution generated by the proposed matheuristic produces the lowest cost. In this solution, two vessels (V1 and V5) are used to remove the top structures. Those vessels are chartered for 32 days in total with the maximum top-structures that can be removed is the same as the requirement (110). For removing sub-structures, Vessel V5 is rented for 28 days which is able to dismantle 112 sub-structures. In this problem, there are 37,212 possible slots to select representing 37,212 binary variables. Stage 1 of the proposed matheuristic (Algorithm 1) reduces the problem by obtaining 48 promising slots only. In Stage 2 of Algorithm 1, an initial good solution is attained involving 9 slots. Each iteration of the proposed local searches (Stage 3 of Algorithm 1) deals with under 20 slots.

Tables 8, 9 and 10 show the computational results for  $|V| = 5, 6$  and 7 respectively where the objective function values and CPU time (in seconds) for the exact method, the 2-stage exact method and the proposed metaheuristic approach are given. Each table comprise 24 instances based on the number of planning delays (PD), the minimum hire periods ( $\beta_v$ ) and the number of turbines ( $n$ ) that need to be removed. Based on the tables, the exact method (CPLEX) was able to obtain upper and lower bounds ( $UB$  and  $LB$ ) for all instances within 3 hours. The 2-

**Table 7** The schedule of the jack-up vessels for  $n = 110$  with  $|V| = 6$ ,  $PD=0$ , and  $\beta_v = 7$ 

| Exact Method            |                   |           |          | 2-Stage Exact Method    |                   |           |          | Matheuristic            |                   |           |          |
|-------------------------|-------------------|-----------|----------|-------------------------|-------------------|-----------|----------|-------------------------|-------------------|-----------|----------|
| Vess.                   | Top/Sub Structure | Start Day | No. Days | Vess.                   | Top/Sub Structure | Start Day | No. Days | Vess.                   | Top/Sub Structure | Start Day | No. Days |
| V1                      | Top               | 33        | 7        | V1                      | Top               | 47        | 7        | V1                      | Top               | 25        | 7        |
| V1                      | Top               | 47        | 7        | V1                      | Top               | 78        | 7        | V1                      | Top               | 47        | 11       |
| V1                      | Top               | 69        | 7        | V4                      | Top               | 1         | 7        | V1                      | Top               | 76        | 7        |
| V1                      | Top               | 96        | 7        | V4                      | Top               | 18        | 7        | V5                      | Top               | 1         | 7        |
| V5                      | Sub               | 22        | 7        | V5                      | Sub               | 3         | 7        | V5                      | Sub               | 9         | 7        |
| V5                      | Sub               | 47        | 7        | V5                      | Sub               | 42        | 7        | V5                      | Sub               | 34        | 7        |
| V5                      | Sub               | 68        | 7        | V5                      | Sub               | 67        | 7        | V5                      | Sub               | 59        | 7        |
| V5                      | Sub               | 98        | 7        | V5                      | Sub               | 88        | 7        | V5                      | Sub               | 78        | 7        |
| V6                      | Top               | 1         | 7        | No. days for Top (112)* |                   |           | 28       | No. days for Top (110)* |                   |           | 32       |
| V6                      | Top               | 20        | 7        | No. days for Sub (112)* |                   |           | 28       | No. days for Sub (112)* |                   |           | 28       |
| No. days for Top (126)* |                   |           | 42       |                         |                   |           |          |                         |                   |           |          |
| No. days for Sub (112)* |                   |           | 28       |                         |                   |           |          |                         |                   |           |          |

\*: the maximum top/sub structures that can be removed by the vessels

stage exact method also generated feasible solutions for all instances. Here, in the first stage, the MILP problem is solved within 3 hours. Then, the ILP problem by fixing decision variable  $X_{v,j,s}$  is optimally solved to obtain the feasible solution. The table also presents %Gap obtained by the solution methods based on the formulation given in Equation (35). To evaluate the consistency of the proposed matheuristic method, in each instance the proposed method was executed 5 times, therefore the average result (%Gap<sub>avg</sub>) as well as the best one (%Gap<sub>best</sub>) are given. Boldface values in the table refer to the best solutions obtained.

Tables 8–10 reveal that the proposed matheuristic technique runs much faster than other methods while producing better %Gaps. Based on the best results, on average, the matheuristics method yields a gap of 7.9019%, 7.6221% and 7.5933% whereas based on the average results, it produces 8.2847%, 8.0033% and 7.8048% for the instances with  $|V| = 5, 6$  and  $7$  respectively. It can be argued that the results of the proposed matheuristic method are consistent as the value of the best results is not far from the value of the average ones. Those results are better than the ones produced by the exact method executed within 3 hours where CPLEX provides an average gap of 10.1746%, 9.9495% and 10.1412% for the instances with  $|V| = 5, 6$  and  $7$  respectively. It is worth noting that the matheuristic method produces better solutions for almost all instances. It can be highlighted that based on the %Gap obtained, instances with two days planning delay are easier to solve than instances with no planning delay as the average %Gap for instances with two days planning delay (at around 8.95%) is smaller than the one with no planning delay (at around 6.45%). It is also noted that a higher value of the minimum hire periods ( $\beta_v$ ) will make the problem more difficult to solve. It is also revealed that the %Gap does not change much with respect to the increase in  $n$  and  $|T|$ .

Table 8 The summary of computational results with  $|V| = 5$ 

| Ins     | PD | $\beta_v$ | $n =  T $ | Z Best Known | Exact Method (CPU = 3 hours) |               | 2-stage Exact Method (CPU = 3 hours) |               | Mathuristic          |                     |              |
|---------|----|-----------|-----------|--------------|------------------------------|---------------|--------------------------------------|---------------|----------------------|---------------------|--------------|
|         |    |           |           |              | L/B                          | %Gap          | %Gap MILP                            | %Gap ILP      | %Gap <sub>best</sub> | %Gap <sub>avg</sub> | CPU Time (s) |
| 1       |    |           | 110       | 24,449,530   | 22,510,880                   | 8.8654        | 7.9189                               | <b>7.9292</b> | 8.0550               | 8.3097              | 986          |
| 2       |    |           | 120       | 27,354,874   | 25,191,350                   | 10.6738       | 8.2682                               | 8.2858        | <b>7.9091</b>        | 7.9953              | 1,233        |
| 3       |    | 7         | 130       | 30,147,035   | 27,974,064                   | 9.8937        | 8.2455                               | 8.2617        | <b>7.2079</b>        | 7.5116              | 1,469        |
| 4       |    |           | 140       | 33,209,309   | 30,696,136                   | 10.8659       | 7.6471                               | 7.6568        | <b>7.5677</b>        | 7.6425              | 1,789        |
| 5       |    |           | 150       | 36,108,391   | 33,497,217                   | 9.8352        | 7.9455                               | 7.9578        | <b>7.2315</b>        | 7.3901              | 2,037        |
| 6       | 0  |           | 160       | 39,652,448   | 36,168,418                   | 12.5194       | 8.7463                               | <b>8.7864</b> | 11.0086              | 13.6340             | 1,654        |
| 7       |    |           | 110       | 25,213,633   | 22,599,256                   | 13.9957       | 10.4340                              | 10.4462       | <b>10.3689</b>       | 10.7127             | 707          |
| 8       |    |           | 120       | 28,023,902   | 25,246,919                   | 15.5663       | 10.2946                              | 10.3112       | <b>9.9093</b>        | 10.3221             | 1,069        |
| 9       |    |           | 130       | 31,053,225   | 27,995,457                   | 18.6859       | 10.9694                              | 10.9765       | <b>9.8469</b>        | 9.9437              | 1,311        |
| 10      |    | 14        | 140       | 34,115,187   | 30,756,847                   | 12.9210       | 10.3175                              | 10.3263       | <b>9.8441</b>        | 9.9673              | 1,603        |
| 11      |    |           | 150       | 37,322,356   | 33,456,356                   | 15.5844       | 10.6306                              | 10.6484       | <b>10.3584</b>       | 10.5184             | 1,877        |
| 12      |    |           | 160       | 40,379,527   | 36,231,363                   | 14.9747       | 11.0735                              | 11.0981       | <b>10.2729</b>       | 10.7633             | 1,843        |
| 13      |    |           | 110       | 23,393,771   | 21,825,329                   | <b>6.7045</b> | 7.3761                               | 7.3907        | 7.1333               | 7.4654              | 966          |
| 14      |    |           | 120       | 26,143,736   | 24,328,504                   | 8.5778        | 7.4017                               | 7.4475        | <b>6.9433</b>        | 7.2163              | 1,178        |
| 15      |    |           | 130       | 29,050,876   | 27,362,416                   | 7.2570        | 6.8986                               | 6.9393        | <b>5.8121</b>        | 5.8758              | 1,443        |
| 16      |    | 7         | 140       | 32,048,418   | 30,403,046                   | 5.3373        | 6.0604                               | 6.0671        | <b>5.1340</b>        | 5.3340              | 1,654        |
| 17      |    |           | 150       | 35,072,961   | 33,478,409                   | 5.2753        | 4.8384                               | 4.8657        | <b>4.5464</b>        | 4.7231              | 2,004        |
| 18      | 2  |           | 160       | 38,120,487   | 36,552,786                   | <b>4.1125</b> | 4.8597                               | 4.8659        | 4.4187               | 4.6472              | 2,225        |
| 19      |    |           | 110       | 24,466,292   | 21,919,697                   | 10.7527       | 10.5533                              | 10.4902       | <b>10.4086</b>       | 10.6544             | 789          |
| 20      |    |           | 120       | 26,942,019   | 24,331,370                   | 10.7422       | 11.0271                              | 11.0494       | <b>9.6899</b>        | 10.3584             | 690          |
| 21      |    |           | 130       | 29,628,374   | 27,401,830                   | 8.5167        | 9.6815                               | 9.7016        | <b>7.5149</b>        | 8.0351              | 1,245        |
| 22      |    | 14        | 140       | 32,532,643   | 30,415,986                   | 7.1639        | 8.8372                               | 8.8446        | <b>6.5063</b>        | 6.9634              | 1,418        |
| 23      |    |           | 150       | 35,674,744   | 33,482,381                   | 6.6919        | 7.1680                               | 7.1922        | <b>6.1454</b>        | 6.6048              | 1,927        |
| 24      |    |           | 160       | 38,824,764   | 36,568,167                   | 8.6783        | 5.8754                               | 5.8941        | <b>5.8123</b>        | 6.2450              | 1,923        |
| Average |    |           |           |              |                              | 10.1746       |                                      | 8.4764        | <b>7.9019</b>        | 8.2847              | 1,460        |

**Table 9** The summary of computational results with  $|V| = 6$ 

| Ins     | PD | $\beta_v$ | $n =  T $ | Z Best Known | Exact Method (CPU = 3 hours) |         | 2-stage Exact Method (CPU = 3 hours) |               | Mathuristic          |                     |              |
|---------|----|-----------|-----------|--------------|------------------------------|---------|--------------------------------------|---------------|----------------------|---------------------|--------------|
|         |    |           |           |              | L/B                          | %Gap    | %Gap MILP                            | %Gap ILP      | %Gap <sub>best</sub> | %Gap <sub>avg</sub> | CPU Time (s) |
| 1       |    |           | 110       | 24,321,019   | 22,508,026                   | 8.2070  | 7.8128                               | 7.8258        | <b>7.4544</b>        | 8.0925              | 957          |
| 2       |    |           | 120       | 27,357,023   | 25,209,348                   | 8.9264  | 8.0724                               | 8.0919        | <b>7.8505</b>        | 8.0768              | 1,240        |
| 3       |    |           | 130       | 30,277,705   | 27,976,213                   | 11.4240 | 8.3054                               | 8.3137        | <b>7.6013</b>        | 7.6605              | 1,448        |
| 4       |    | 7         | 140       | 33,214,537   | 30,713,026                   | 9.6746  | 7.5228                               | <b>7.5314</b> | 7.9185               | 8.0146              | 1,827        |
| 5       |    |           | 150       | 36,037,228   | 33,445,339                   | 9.6501  | 7.9128                               | 7.9214        | <b>7.1923</b>        | 7.2337              | 1,932        |
| 6       |    |           | 160       | 39,404,093   | 36,197,227                   | 10.2365 | 8.1195                               | <b>8.1384</b> | 9.1134               | 9.1134              | 1,591        |
| 7       | 0  |           | 110       | 25,215,190   | 22,567,325                   | 14.0650 | 11.1361                              | 11.1399       | <b>10.5011</b>       | 10.9970             | 872          |
| 8       |    |           | 120       | 28,027,519   | 25,258,082                   | 13.6743 | 10.9111                              | 10.9331       | <b>9.8811</b>        | 10.7566             | 865          |
| 9       |    |           | 130       | 31,049,225   | 27,937,752                   | 14.7903 | 11.1626                              | 11.1767       | <b>10.0211</b>       | 10.1569             | 1,108        |
| 10      |    | 14        | 140       | 33,948,687   | 30,716,873                   | 16.2116 | 11.4805                              | 11.4923       | <b>9.5197</b>        | 9.7829              | 1,652        |
| 11      |    |           | 150       | 37,015,290   | 33,477,363                   | 13.9446 | 11.2470                              | 11.2804       | <b>9.5580</b>        | 9.6171              | 1,932        |
| 12      |    |           | 160       | 40,167,986   | 36,215,425                   | 15.2265 | 11.6345                              | 11.6778       | <b>9.8401</b>        | 10.3324             | 1,970        |
| 13      |    |           | 110       | 23,289,111   | 21,830,229                   | 7.5854  | 6.9097                               | 6.9361        | <b>6.2642</b>        | 6.9940              | 959          |
| 14      |    |           | 120       | 26,058,711   | 24,328,242                   | 7.7702  | 6.6058                               | <b>6.6407</b> | 7.0005               | 7.1378              | 1,173        |
| 15      |    |           | 130       | 28,968,038   | 27,369,521                   | 5.7969  | 6.1821                               | 6.2165        | <b>5.5182</b>        | 5.7026              | 1,456        |
| 16      |    | 7         | 140       | 31,835,638   | 30,401,460                   | 5.5694  | 5.0712                               | 5.0864        | <b>4.5049</b>        | 4.5342              | 1,480        |
| 17      |    |           | 150       | 34,959,761   | 33,481,707                   | 4.6162  | 4.8395                               | 4.8502        | <b>4.2279</b>        | 4.4075              | 2,061        |
| 18      |    |           | 160       | 38,096,710   | 36,557,156                   | 4.5188  | 4.2158                               | 4.2228        | <b>4.0412</b>        | 4.2724              | 2,235        |
| 19      | 2  |           | 110       | 24,341,521   | 21,903,640                   | 10.3595 | 11.2236                              | 11.2374       | <b>10.0153</b>       | 10.3845             | 742          |
| 20      |    |           | 120       | 26,882,552   | 24,328,856                   | 11.7747 | 10.7238                              | 10.7425       | <b>9.4995</b>        | 9.6963              | 1,006        |
| 21      |    |           | 130       | 29,551,044   | 27,397,207                   | 11.8681 | 9.3769                               | 9.3955        | <b>7.2885</b>        | 7.3619              | 1,399        |
| 22      |    | 14        | 140       | 32,535,728   | 30,411,506                   | 7.8652  | 6.9809                               | 7.0032        | <b>6.5289</b>        | 6.9862              | 1,344        |
| 23      |    |           | 150       | 35,615,041   | 33,485,184                   | 6.3146  | 6.2325                               | 6.2495        | <b>5.9802</b>        | 6.1303              | 1,945        |
| 24      |    |           | 160       | 38,732,601   | 36,559,720                   | 8.7173  | 6.1180                               | 6.1276        | <b>5.6100</b>        | 5.8953              | 1,898        |
| Average |    |           |           |              |                              | 9.9495  |                                      | 8.3430        | <b>7.6221</b>        | 7.8891              | 1,462        |

Table 10 The summary of computational results with  $|V| = 7$ 

| Ins     | PD | $\beta_v$ | $n =  T $ | Exact Method<br>(CPU = 3 hours) |            | 2-stage Exact Method<br>(CPU = 3 hours) |         |                | Math heuristic |                      |                     |
|---------|----|-----------|-----------|---------------------------------|------------|---|---------|----------------|----------------|----------------------|---------------------|
|         |    |           |           | Z Best<br>Known                 | LB         | %Gap                                    | MILP    | %Gap           | ILP            | %Gap <sub>best</sub> | %Gap <sub>avg</sub> |
| 1       |    |           | 110       | 24,354,926                      | 22,486,021 | 8.4587                                  | 7.6617  | 7.6736         | <b>7.7132</b>  | 7.9500               | 832                 |
| 2       |    |           | 120       | 27,353,474                      | 25,213,230 | 10.0000                                 | 8.2854  | 8.3046         | <b>7.8244</b>  | 7.9032               | 1,196               |
| 3       |    | 7         | 130       | 30,399,571                      | 27,962,228 | 9.6326                                  | 8.1080  | 8.1197         | <b>8.0177</b>  | 8.1496               | 1,413               |
| 4       |    |           | 140       | 33,202,660                      | 30,669,979 | 9.7551                                  | 8.1573  | 8.1686         | <b>7.6279</b>  | 7.9194               | 1,667               |
| 5       |    |           | 150       | 36,023,223                      | 33,459,828 | 12.3032                                 | 7.9636  | 7.9710         | <b>7.1160</b>  | 7.3193               | 2,007               |
| 6       | 0  |           | 160       | 39,192,424                      | 36,201,643 | 10.9709                                 | 8.0875  | 8.1000         | <b>7.6310</b>  | 7.7877               | 2,251               |
| 7       |    |           | 110       | 25,235,004                      | 22,505,335 | 14.9827                                 | 10.8078 | <b>10.8170</b> | 11.1304        | 11.5442              | 851                 |
| 8       |    |           | 120       | 28,003,419                      | 25,218,222 | 12.2233                                 | 10.5105 | 10.5164        | <b>9.9459</b>  | 10.5976              | 1,059               |
| 9       |    |           | 130       | 30,928,717                      | 28,051,359 | 14.8395                                 | 11.0144 | 11.0356        | <b>9.3032</b>  | 9.8662               | 1,063               |
| 10      |    | 14        | 140       | 33,962,738                      | 30,739,327 | 14.2965                                 | 9.4796  | <b>9.4910</b>  | 10.0006        | 10.0333              | 1,644               |
| 11      |    |           | 150       | 36,989,132                      | 33,501,090 | 16.1361                                 | 10.4046 | 10.4170        | <b>9.4299</b>  | 9.5912               | 1,923               |
| 12      |    |           | 160       | 40,495,284                      | 36,222,028 | 15.9224                                 | 10.8183 | 10.8362        | <b>10.5525</b> | 10.8215              | 2,270               |
| 13      |    |           | 110       | 23,265,443                      | 21,825,595 | 8.4919                                  | 7.0620  | 7.0849         | <b>6.1888</b>  | 6.3458               | 955                 |
| 14      |    |           | 120       | 25,988,711                      | 24,327,951 | 6.4642                                  | 7.7091  | 7.7465         | <b>6.3903</b>  | 6.4760               | 1,166               |
| 15      |    |           | 130       | 29,012,514                      | 27,365,778 | 5.7072                                  | 6.0336  | 6.0743         | <b>5.6760</b>  | 5.7976               | 1,418               |
| 16      |    | 7         | 140       | 31,880,406                      | 30,397,997 | 5.4253                                  | 5.0923  | 5.1028         | <b>4.6499</b>  | 4.8199               | 1,671               |
| 17      |    |           | 150       | 34,983,233                      | 33,472,003 | 6.0128                                  | 4.9089  | 4.9251         | <b>4.3199</b>  | 4.4144               | 1,986               |
| 18      |    |           | 160       | 37,930,685                      | 36,557,853 | 8.3411                                  | 4.3982  | 4.4086         | <b>3.6193</b>  | 3.7756               | 2,323               |
| 19      |    |           | 110       | 24,369,149                      | 21,891,503 | <b>10.1671</b>                          | 11.2558 | 11.2827        | 10.1812        | 10.3641              | 678                 |
| 20      |    |           | 120       | 26,790,588                      | 24,327,830 | 11.6952                                 | 10.3261 | 10.9427        | <b>9.1926</b>  | 9.3608               | 860                 |
| 21      |    |           | 130       | 29,663,306                      | 27,399,951 | 9.1615                                  | 10.2209 | 10.2309        | <b>7.6301</b>  | 7.9579               | 1,397               |
| 22      |    | 14        | 140       | 32,535,131                      | 30,407,695 | 7.5938                                  | 8.3443  | 8.3532         | <b>6.5389</b>  | 6.6492               | 1,608               |
| 23      |    |           | 150       | 35,640,704                      | 33,483,066 | 6.8274                                  | 6.8120  | 6.8314         | <b>6.0539</b>  | 6.1513               | 1,933               |
| 24      |    |           | 160       | 38,691,188                      | 36,561,312 | 7.9791                                  | 6.0294  | 6.0409         | <b>5.5048</b>  | 5.7197               | 2,232               |
| Average |    |           |           |                                 |            | 10.1412                                 |         | 8.3531         | <b>7.5933</b>  | 7.8048               | 1,517               |

We also carried out additional computational experiments on several instances (i.e.  $|V| = 6$  with  $PD=0$  and  $\beta=14$ ;  $|V| = 7$  with  $PD=0$  and  $\beta=7$ ). The 2-stage exact method is executed much longer than in the previous experiments where in the first stage, the MILP problem is solved within 24 hours instead of 3 hours. For a fair comparison, the results of the 2-stage exact method are compared with the results of the proposed matheuristic method executed for a longer computational time. Moreover, a minor revision on the matheuristic is applied where in Stages 1, 2 and 3 of Algorithm 1, the relaxed MILP and ILP problems are optimally solved. The local searches are executed four times longer than in the previous ones. Unfortunately, the exact method can be run up to 24 hours due to memory issue. Table 11 presents the comparison of two methods when a problem is solved within the same computational time. In the table, the  $\%Gap$  is calculated based on the value of lower bound ( $LB$ ) obtained from the exact method run within 3 hours (Tables 8–10).

Similar to the previous results, Table 11 also reveals that the results of the proposed matheuristic technique yields a better average  $\%Gap$  than the 2-stage exact method. The matheuristic method produces an average  $\%Gap$  of 8.4584% for the best results whereas the 2-stage exact method yields 8.7004%. According to the table, the 2-stage exact method produces better solutions for the problem with  $|V| = 7$ ,  $PD=0$  and  $\beta=7$  when  $n = 110$  and  $140$ . In general, it can be concluded that the proposed matheuristic method performs better than the 2-stage exact method in term of the quality of solutions obtained and the computational time.

**Table 11** The results using the long computational time

| Instance                          | $N =  T $ | $Z$ Best<br>Known | 2-stage Exact Method<br>(CPU = 24 hours) |                | Matheuristic   |               | CPU<br>Time (s) |
|-----------------------------------|-----------|-------------------|--|----------------|----------------|---------------|-----------------|
|                                   |           |                   | $\%Gap$<br>MILP                          | $\%Gap$<br>ILP | $\%Gap_{best}$ | $\%Gap_{avg}$ |                 |
| $ V  = 6$<br>$PD=0$<br>$\beta=14$ | 110       | 25,146,763        | 10.6284                                  | 10.6439        | <b>10.2575</b> | 10.4487       | 2,891           |
|                                   | 120       | 27,935,302        | 9.6015                                   | 9.6057         | <b>9.5836</b>  | 10.2331       | 3,709           |
|                                   | 130       | 30,930,425        | 10.3226                                  | 10.3296        | <b>9.6755</b>  | 10.1136       | 4,705           |
|                                   | 140       | 33,911,538        | 9.4886                                   | 9.5004         | <b>9.4206</b>  | 9.4416        | 6,335           |
|                                   | 150       | 37,049,104        | 10.3243                                  | 10.3360        | <b>9.6406</b>  | 9.6884        | 7,177           |
| $ V  = 7$<br>$PD=0$<br>$\beta=7$  | 110       | 24,319,719        | 7.5287                                   | <b>7.5400</b>  | 7.8724         | 8.1040        | 3,472           |
|                                   | 120       | 27,117,654        | 7.8044                                   | 7.8256         | <b>7.0228</b>  | 7.3089        | 6,837           |
|                                   | 130       | 30,031,733        | 7.4701                                   | 7.4881         | <b>6.8911</b>  | 7.1699        | 6,953           |
|                                   | 140       | 32,855,777        | 6.6464                                   | <b>6.6527</b>  | 7.1565         | 7.2274        | 13,190          |
|                                   | 150       | 36,003,037        | 7.0730                                   | 7.0824         | <b>7.0639</b>  | 7.1490        | 26,481          |
| Average                           |           |                   |  | 8.7004         | <b>8.4584</b>  | 8.6885        | 8,175           |

## 6 Conclusions

This paper proposes an optimisation model for scheduling the decommissioning of an offshore wind farm. A mathematical model using ILP is introduced to determine the optimal schedule in order to minimise the total cost involving jack-up

vessel cost, barge vessel cost, inventory cost, processing cost and on-land transportation cost whilst considering several constraints such as the availability of vessels and the planning delay. A matheuristic approach is also proposed to find solutions as the exact method (using CPLEX) experienced difficulties to address relatively large problems. Two datasets based on a wind farm site located in North Sea (UK) are constructed for evaluating the performance of the proposed methods. The computational experiments show that the proposed approaches produce good results. The matheuristic method performs well and runs much faster than the exact method. The matheuristic approach obtains the feasible solution in a reasonable computational time. In addition, this paper also provides a comprehensive review on the decommissioning of an offshore wind farm relating to the process and legislation, costs and managing waste and recycling.

The decommissioning model presented in this paper is forward looking in that the dismantling of wind farms currently in operation or under construction is not yet scheduled a number of years. Nevertheless, models such as the one developed in this paper are essential in order to prepare for this future large scale operation. They also provide more accurate estimates of the costing of this final life cycle stage of a wind farm in the presence of an optimised logistics strategy. There are, of course, many possible technological enhancements and policy factors that could alter the data presented in this paper. However, the general modelling framework presented would still apply. The model could then be rerun with the new data resulting from the technological improvement in order to provide a new best estimate of the future cost of decommissioning. One important issue raised in the paper is that of a single or multiple repowering over the course of the wind farm site lease period. If and when to repower is a relevant operations research decision problem that deserves future consideration in the literature. It is now however the focus of this paper, and we note that a repowering strategy will only delay and not negate the need for our presented decommissioning model. In fact, the output of our model could be seen as one input to a repowering decision model. Furthermore, the output of our model could also be used interactively in wider offshore wind farm supply chain and life cycle models.

The correct handling of weather related uncertainties is key to optimal planning in all life cycle phases of an offshore wind farm. The weather cannot be considered as a "planned delay" in our model as the time-frame is too long to know the weather in any given slot accurately in advance. Therefore there are several strategies for incorporating the weather into the model. The first point is to schedule activities in the summer period when the average weather is better and hence the chance of disruption is minimised. The second is to use a retrospective analysis to see how robust our proposed schedules are against actual weather in past periods. The third is to use a form of Monte Carlo simulation to check how well the proposed schedules perform against specific weather conditions generated from long term averages. This latter method is in common with current industry practice. An optimisation based extension of the methodology in this paper to consider uncertain weather conditions via a technique such as **robust optimization, stochastic programming or multiple objective optimization** with different objectives representing different weather scenarios could be considered. This could be compared against current Monte Carlo simulation practice.

The model and solution methodology presented have been developed specifically for the case of the decommissioning of medium to large offshore wind farms,



but there is no reason why they could not be adapted to deal with the removal of other large offshore structures that consist of multiple components and are part recyclable. The model can be extended to consider not only the total cost but also the environmental impact. Therefore, it can be treated as a bi-objective problem instead of a single objective problem which is more difficult to solve. The application of a compromise programming (CP) can be used to deal with such problem. CP aims to select a solution from the set of efficient solutions based on a reasonable assumption that any decision maker finds a solution as close as possible to the ideal point.

**Acknowledgements** The research leading to these results has received funding from the European Union Seventh Framework Programme under the agreement SCP2-GA-2013-614020 (LEANWIND: Logistic Efficiencies And Naval architecture for Wind Installations with Novel Developments).

## References

- Ait-Alla A, Quandt M, Lütjen M (2013) Aggregate installation planning of offshore wind farms. In: Proceedings of the 7th International Conference on Communications and Information Technology (CIT&Ż13), Cambridge, pp 130–135
- Barlow E, ŨztĀijrk DT, Revie M, AkartunalĀs K, Day AH, Boulougouris E (2018) A mixed-method optimisation and simulation framework for supporting logistical decisions during offshore wind farm installations. *European Journal of Operational Research* 264(3):894 – 906
- Benmessaoud T, Mohammedi K, Smaili Y (2013) Influence of maintenance on the performance of a wind farm. *PrzełĀd Elektrotechniczny* 89:174–178
- Besnard F, Patriksson M, Strömberg A, Wojciechowski A, Bertling L (2009) An optimization framework for opportunistic maintenance of offshore wind power system. In: Proceedings of IEEE Bucharest Powertech Conference, Bucharest, Romania
- Besnard F, Patriksson M, Strömberg A, Wojciechowski A, Bertling L (2011) A stochastic model for opportunistic service maintenance planning of offshore wind farms. In: Proceedings of the IEEE Powertech Conference, Trondheim, Norway
- Bradley, S (2013) Offshore wind ĀĀ end of life opportunities. energy institute evening lecture. Tech. rep., London
- Byon E, PĀrez E, Ding Y, Ntaimo L (2011) Simulation of wind farm operations and maintenance using discrete event system specification. *SIMULATION* 87(12):1093–1117
- Camci F (2015) Maintenance scheduling of geographically distributed assets with prognostics information. *European Journal of Operational Research* 245(2):506 – 516
- CCC (2010) Offshore renewable energy installation decommissioning study: Final report. Tech. rep., Climate Change Capital Ltd.
- Dai L, StĀlhane M, Utne I (2015) Routing and scheduling of maintenance fleet for offshore wind farms. *Wind Engineering* 39:15ĀĀ30
- DECC (2011) Decommissioning of offshore renewable energy installations under the energy act 2004: Guidance notes for industry. Tech. rep., DECC

- Dos Santos F, Teixeira A, Soares C (2013) Maintenance planning of an offshore wind turbine using stochastic petri nets with predicates. In: Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering (OMAE), Nantes, France
- Garcia C, Rabadi G (2013) Exact and approximate methods for parallel multiple-area spatial scheduling with release times. *OR Spectrum* 35(3):639–657
- Irawan CA, Jones D, Ouelhadj D (2017a) Bi-objective optimisation model for installation scheduling in offshore wind farms. *Computers & Operations Research* 78:393 – 407
- Irawan CA, Ouelhadj D, Jones D, StÅelhane M, Sperstad IB (2017b) Optimisation of maintenance routing and scheduling for offshore wind farms. *European Journal of Operational Research* 256(1):76 – 89
- Januario C, S S, Bell M (2007) Offshore windfarm decommissioning: A proposal for guidelines to be included in the european maritime policy. In: Conference Proceedings EWEC, Milan
- Kaiser J, Snyder B (2012) Modelling the decommissioning cost of offshore wind development on the u.s. outer continental shelf mark. *Marine Policy* 36:153–164
- Kovacs A, Erdos G, Viharos Z, Monostori L (2011) A system for the detailed scheduling of wind farm maintenance. *CIRP Annals Manufacturing Technology* 60:497–501
- Lütjen M, Karimi H (2012) Approach of a port inventory control system for the offshore installation of wind turbines. In: Proceedings of the 22nd International Offshore and Polar Engineering Conference (ISOPE), Rhodes, Greek, pp 502–508
- Maniezzo V, Stutzle T, Vob S (2010) Matheuristics, hybridizing metaheuristics and mathematical programming. *Annals of information systems* 10
- Pan K, You D, Wang K (2013) Long-term multi-objective optimization dispatch and its evaluation in wind integrated power systems. *Advanced Materials Research* 732:1033–1037
- Parikh N (2012) Optimizing maintenance cost of wind farms by scheduling preventive maintenance and replacement of critical components using mathematical approach. Master thesis, Texas A&M University, US
- Pattison D, Xie W, Quail F (2013) The windy domain - a challenging real-world application of integrated planning and scheduling. In: Proceedings of the 23rd International Conference on Automated Planning and Scheduling (ICAPS), Rome, Italy, pp 462–466
- Pearson D (2001) Decommissioning wind turbines in the uk offshore zone. In: BWEA23: Turning Things Around – annual conference and exhibition, Brighton
- Peréz E, Ntaimo L, Byon E, Ding Y (2010) A stochastic devts wind turbine component model for wind farm simulation. In: Proceeding of the Spring Simulation Multi-conference (SpringSim), Florida, US
- Peréz E, Ntaimo L, Ding Y (2013) Simulation of wind farm operations and maintenance. In: Proceedings of the ASME Turbine Technical Conference and Exposition, San Antonio, US
- Renewable UK (2015) Offshore wind energy figures. Tech. rep., URL <http://www.renewableuk.com/page/UKWEDhome> [accessed on 20/10/17]
- RWE (2011) Decommissioning strategy: Gwynt y môr offshore wind farm ltd. Tech. rep., URL

- <https://www.innogy.com/web/cms/mediablob/en/3171308/data/3171562/1/rwe-innogy/rwe-innogy-uk/sites/wind-offshore/in-operation/gwynt-y-mr/English.pdf> [accessed on 30/10/17]
- Sarker BR, Faiz TI (2017) Minimizing transportation and installation costs for turbines in offshore wind farms. *Renewable Energy* 101:667 – 679
- Scholz-Reiter B, Heger J, Lutjen M, Schweizer A (2010) A milp for installation scheduling of offshore wind farms. *International Journal of Mathematical Models and Methods in Applied Sciences* 5:371–378
- Scholz-Reiter B, Karimi H, Lutjen M, Heger J, Schweizer A (2011) Towards a heuristic for scheduling offshore installation processes. In: *Proceedings of the 24th International Congress on Condition Monitoring (COMADEM) Stavanger, Norway.*, pp 999–1008
- Schwerdfeger S, Boysen N, Briskorn D (2018) Just-in-time logistics for far-distant suppliers: scheduling truck departures from an intermediate cross-docking terminal. *OR Spectrum* 40(1):1–21
- Stålhane M, Hvattum L, Skaar V (2014) Optimization of routing and scheduling of vessels to perform maintenance at offshore wind farms. In: *Proceedings of the 12th Deep Sea Offshore Wind R&D Conference, EERA DeepWind 2015*
- Stecky-Efantis A (2013) Evaluating ontario wind turbine decommissioning plans. Tech. rep., School of Urban and Regional Planning, Queen’s University, Kingston, Ontario, Canada
- Ursavas E (2017) A benders decomposition approach for solving the offshore wind farm installation planning at the north sea. *European Journal of Operational Research* 258(2):703 – 714
- Van Horenbeek A, Van Ostaeyen J, Dufflou J, Pintelon L (2012) Prognostic maintenance scheduling for offshore wind turbine farms. In: *Proceedings of the 4th Production & Operations Management World Conference, Amsterdam, the Netherlands*
- Vattenfall (2016) The first decommission of an offshore wind farm is now complete. Tech. rep., URL <https://corporate.vattenfall.com/press-and-media/press-releases/2016/the-first-decommission-in-the-world-of-an-offshore-wind-farm-is-now-complete/> [accessed on 30/10/17]
- Wang K, Zhang B, Wu X, Zhai J, Shao W, Duan Y (2012) Multi-time scales coordination scheduling of wind power integrated system. In: *Proceedings of the IEEE Innovative Smart Grid Technologies (ISGT)-Asia, Tianjin, China*
- Wu Y, Bing H, Lou S (2012) A unit maintenance scheduling model considering peak regulation pressure balance for power system containing wind farms. *Power System Technology* 36:94–100
- Zhang H, Yin Y, Shen H, He J, Zhao S (2012a) Generator maintenance scheduling of large-scale wind power integration considering peak shaving. *Automation of Electric Power Systems* 36:25–30
- Zhang J, Chowdhury S, Zhang J (2012b) Optimal preventive maintenance time windows for offshore wind farms subject to wake losses. In: *Proceedings of the 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Indianapolis, US*