Author's Accepted Manuscript

Bi-objective optimisation model for installation scheduling in offshore wind farms

Chandra Ade Irawan, Dylan Jones, Djamila Ouelhadj

PII: S0305-0548(15)00231-2
DOI: <http://dx.doi.org/10.1016/j.cor.2015.09.010>
Reference: CAOR3861

To appear in: *Computers and Operation Research*

Received date: 26 February 2015
Revised date: 10 July 2015
Accepted date: 30 September 2015

Cite this article as: Chandra Ade Irawan, Dylan Jones and Djamila Ouelhadj, Bi objective optimisation model for installation scheduling in offshore wind farms *Computers and Operation Research*, <http://dx.doi.org/10.1016/j.cor.2015.09.010>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting galley proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain

Bi-objective optimisation model for installation scheduling in offshore wind farms

Chandra Ade Irawan, Dylan Jones, Djamila Ouelhadj

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

Abstract A bi-objective optimisation using a compromise programming approach is proposed for installation scheduling of an offshore wind farm. As the installation cost and the completion period of the installation are important aspects in the construction of an offshore wind farm, the proposed method is used to deal with those conflicting objectives. We develop a mathematical model using integer linear programming (ILP) to determine the optimal installation schedule considering several constraints such as weather condition and the availability of vessels. We suggest two approaches to deal with the multi-objective installation scheduling problem, namely compromise programming with exact method and with metaheuristic techniques. In the exact method the problem is solved by CPLEX whereas in the metaheuristic approach we propose Variable Neighbourhood Search (VNS) and Simulated Annealing (SA). Moreover, greedy algorithms and a local search for solving the scheduling problem are introduced. Two generated datasets are used for testing our approaches. The computational experiments show that the proposed metaheuristic approaches produce interesting results as the optimal solution for some cases is obtained.

Key words: Variable Neighbourhood Search, Simulated Annealing, Multi-objectives, Compromise programming, Installation Scheduling, Offshore Wind Farm.

1. Introduction

Wind power is a promising electricity generation source as it is renewable and can hence contribute to the reduction of carbon gas emissions. As the power output from a wind turbine is a function of wind speed, a wind farm should be located in an area that has strong and steady wind. The number of global offshore wind farms is rapidly increasing annually as the average of wind speed at sea is superior compared to that of onshore. Additionally, siting

wind farms offshore can alleviate some of the land use and social concerns found in onshore wind farms. According to European Wind Energy Association (EWEA, 2014), in the European Union (EU), the cumulative installed capacity of offshore wind power increased significantly to 6,600 MW in year 2013 from 532 megawatts (MW) in year 2003.

However, CAPEX (capital expenditure) and OPEX (operating expenditure) costs of offshore wind turbine are much higher than those of onshore ones. At sea a wind turbine is more difficult to install and maintain so more resources and infrastructures are needed; therefore the cost spent on offshore one is much higher. The installation/construction phase of an offshore wind farm is very challenging as heavy equipment and costly vessels are required. Based on the Renewables Advisory Board (2010), the installation and commissioning phase makes up 26% of CAPEX cost of which vessel chartering costs contribute the biggest portion.

A wind turbine mainly consists of three components namely sub-structure (foundation and transition piece), cable, and top-structure (tower, nacelle, and blades). Sub- and top-structures are usually installed by a self-elevating vessel including a barge and a self-propelled installation vessel. In barge installation, a barge transports sub- or top-structures while the installation vessel positioned at site will conduct the installation. In self-propelled, an installation vessel will pick up sub- or top-structures at the staging area (port), and then return to site to do the installation. For cable (inner-array cable) installation, the most common methods involve the use of an ROV (Remotely Operated Vehicle) operated by either the main installation vessel or a specialized cable-laying vessel (Kaiser and Snyder, 2013).

The weather conditions (such as wind speed and wave height) and the vessel availability are the main factors that affect the performance of the installation process. The delay in installing wind turbines is mainly due to those factors and a one-day delay will cause a significant financial loss. For safety reasons, the installation must be conducted in the period when the required weather conditions are met. The sub-structure and cable can be installed with relatively stronger wind speeds, while the top-structure requires calmer weather. As good weather periods are limited, this leads to a massive stockpiling of material and resources in the port or on board of the vessel to exploit these periods (Scholz-Reiter *et al.*, 2010). Here, the schedule of the installation is very important for determining suitable target inventories for the whole supply chain.

When scheduling the installation of wind turbine, the planner usually seeks the best configuration to complete the project as soon as possible at minimum cost. However, it is not easy to achieve as the completion installation period/date and installation cost are conflicting objectives. For example in Northern Europe, to minimise the installation cost the installation of top-structures has to wait until spring or summer when the weather is relatively calm, otherwise in winter the installation time of a top-structure will take longer which results in the increase of the installation cost. In this paper, we are investigating the installation scheduling of offshore wind farm in the presence of two conflicting objectives namely total installation cost and total completion period. To the best of our knowledge, there is no paper in the literature studying such a problem.

The main contributions of this paper include: (i) a mathematical model of the installation scheduling problem, (ii) greedy algorithms and a local search for finding the best schedule that minimises total installation cost or total completion period, (iii) application and comparison of VNS and SA for solving the scheduling problems.

The paper is organized as follows: Section 2 presents a brief review of the past efforts at offshore wind installation mainly concentrating on the scheduling problem. Section 3 gives a description of our approach in developing a mathematical model of installation scheduling followed by an explanation of compromise programming method for solving the bi-objective scheduling problem. Our metaheuristic methods (VNS and SA) as well as the overall algorithms are presented in Section 4. Section 5 gives computational results using generated data. A summary of our findings and some avenues for future research are provided in the last section.

2. Past efforts at offshore wind installation

This section presents an overview of past efforts at offshore wind farm focussing on installation scheduling problem. We found four papers in the literature related to installation scheduling problem for offshore wind farm. Scholz-Reiter *et al.* (2010) introduced a mathematical model using mixed integer linear programming (MILP) to obtain the optimal installation schedule with the objective to reduce vessel operation times considering weather conditions. Their model is to schedule one vessel where the vessel can install both sub- and top-structures. The model also only runs for short planning horizon.

Scholz-Reiter *et al.* (2011) proposed a heuristic technique to overcome limitations of their previous model (Scholz-Reiter *et al.*, 2010). The heuristics approach is able to solve relatively large problems with longer time horizons, multiple vessels and a broader variety of weather conditions. Their computational experiments show that the proposed approach produces competitive results.

A simulation approach for determining the optimised configuration of a single-echelon inventory system for offshore installations of wind turbines was investigated by Lütjen and Karimi (2012). They also present a reactive scheduling heuristic based on the model in Scholz-Reiter *et al.* (2011). They found that it is feasible to determine optimised configurations of the logistic system.

A mathematical model dealing with the aggregated installation planning problem for medium planning horizon is introduced by Ait-Alla *et al.* (2013). Their model seeks the optimal aggregated schedule that minimises the total installation costs. The chartering costs and weather operation constraints for different vessel types are considered in the model.

Other interesting topic related to the scheduling problem of an offshore wind farm is offshore maintenance scheduling. Maintenance scheduling aims to produce a detailed schedule of maintenance tasks that have to be performed within a certain period considering the availability of several resources including vessels, spare parts, and crews. Besnard *et al.* (2009) investigated an opportunistic maintenance optimization model taking into account wind forecasts and corrective maintenance activities. Discrete event-based simulation models of maintenance scheduling were studied by Pérez, E. *et al.* (2010), Byon *et al.* (2011), and Pérez, E. *et al.* (2013).

Kovács *et al.* (2011) developed a mathematical model (MILP) to determine the best time for maintenance operations considering the availability of the resources and the performance of the wind turbine. Besnard *et al.* (2011) enhanced their earlier model (Besnard *et al.*, 2009) where uncertainty weather condition is taken into account so the problem becomes a stochastic optimisation problem. A formulation of mathematical model to optimise maintenance cost was introduced by Parikh (2012). The added value of a prognostic maintenance policy was quantified by Van Horenbeek *et al.* (2012).

Long- and short-terms scheduling models for wind power integrated systems were proposed by Wang *et al.* (2012) where the former model involves maintenance scheduling and energy allocation, while the latter finds hourly power output. Wu *et al.* (2012) studied the

maintenance scheduling model taking into account peak regulation pressure balance. Zhang *et al.* (2012) investigated an optimal preventive maintenance scheduling model for minimising the overall downtime energy losses taking into account weather conditions, crews, transportation, and tooling infrastructure. Maintenance scheduling of large-scale wind power considering peak shaving was studied by Zhang *et al.* (2012).

A simulation model for optimising maintenance schedule was implemented by Benmessaoud *et al.* (2013) which is used to analyse the influence of maintenance on the performance of a wind farm. A stochastic petri-net model for maintenance planning was proposed by Dos Santos *et al.* (2013) considering the availability of vessels, crews, and spare parts. Ge *et al.* (2013) studied a long-term scheduling method for wind-hydro-thermal power systems. Pan *et al.* (2013) proposed a long-term multi-objective optimisation dispatch and its evaluation in wind integrated power systems involving maintenance scheduling, unit commitment, and power output. An integrated planning and scheduling maintenance method was investigated by Pattison *et al.* (2013). Perez-Canton and Rubio-Romero (2013) put forward a model for the preventive maintenance scheduling of power plants including wind farms where the aim is to maximise the system reliability.

Stålhane *et al.* (2014) and Dai *et al.* (2015) investigated the problem of finding the optimal routes and schedules for a fleet of vessels that are to perform maintenance tasks at an offshore wind farm. Recently, a comprehensive review related to maintenance logistics in offshore wind energy can be found in Shafiee (2015).

3. Installation scheduling model for offshore wind farm

Offshore wind turbines can be installed using several scenarios. Figure 1(a) shows an approach to install the turbines which is considered in this paper. The components (cables, top- and sub-structures) are prepared at the port which is usually the nearest one to the wind farm site. The installation vessel picks up the components and transports them to the wind farm site. The vessel will also perform the installation which may take several days. The recent vessel can transport top-structure components (tower, nacelle, and blades) for more than six turbines. Once the installation process on the site has been completed, the vessel may return to the port again to pick up other components. Figure 1(b) illustrates the installation vessel, which is designed to transport and install more than four top-structures (turbines). In

Figure 1(c), an example of a layout of turbines at an offshore wind farm is illustrated whereas Figure 1(d) shows the main components of offshore wind turbine.

Figure 1. (a) Installation Approach; (b) Installation vessel (source: A2SEA); (c) An example of a layout of turbines (source: RWE); (d) components of offshore turbine

We propose an installation scheduling model of an offshore wind farm which involves two objectives, namely minimising total installation cost and minimising total completion period. The main aspects considered in the model are depicted in Figure 2. As the chartering vessel cost dominates the installation cost, the proposed model focuses on vessel scheduling taking account into the availability of the vessels and weather conditions. The following are the description of installation tasks, vessels, and weather conditions related to the installation process of offshore wind turbines.

■ Installation tasks

A wind turbine could be operated once all the required components have been installed which include sub-structure, cable, and top-structure. The installation of wind turbine is organised into a sequence of tasks. For example, installing cable can be performed if sub-

structure has been installed. Similarly, cable installation is a predecessor task of top-structure installation.

■ Vessel

A vessel might be used to perform several installation tasks. For example, an installation vessel can install both sub- and top-structure. In this model, a self-propelled installation vessel is considered for installing sub- and top-structures while for installing cable, we only take into account inner-array cable installation. The information regarding a vessel required is rent cost per period, the fixed cost of using a vessel, the number of components that can be installed per trip, loading time, transporting time, installing time, maximum periods that the vessel can be hired, and the availability per period.

■ Weather conditions

According to Scholz-Reiter *et al.* (2010), weather conditions are classified into three types namely good conditions (wind speed $< 6.5\text{m/s}$ and wave height $< 2.5\text{m}$), medium conditions (wind speed $< 12\text{m/s}$ and wave height $< 4.8\text{m}$), and bad conditions (wind speed $> 12\text{m/s}$ and wave height $> 4.8\text{m}$). In bad weather conditions, it is not possible to do some activities whereas in good conditions all activities can be performed. Loading, transporting, and installing sub-structure and cable can be done in medium condition.

Figure 2. The installation scheduling model

The following notations are used to describe the sets and parameters of the proposed installation scheduling model.

Sets and index

- V = set of installation vessels.
 v = index of installation vessel.
 J = set of installation tasks.
 j = index of installation task (sorted according to installation processes). In this study, $j=0$ for installing sub-structure, $=1$ for cable, and $=2$ for top-structure.
 V_j = set of vessels to perform task j , $V_j \subset V$.
 T = set of planning periods.
 t = index of period.

Parameters

- N = total number of turbines to be installed
 $\alpha_{v,j}$ = number of components that can be installed/transported (for sub- and top-structures) or number of turbines that can be connected (for cable) using vessel v for task j in one trip. For example, if $\alpha_{v,j} = 3$ then vessel v can install/transport 3 foundations and 3 transition pieces in one trip when $j=0$ or 3 towers, 3 nacelles, and 9 blades when $j=1$.
 ${}^r c_{v,j}$ = chartering cost of vessel v per period (day) to perform task j
 $\hat{c}_{v,j}$ = fixed cost of vessel v to perform task j
 ${}^l t_{v,j}$ = the time (in hours) required for vessel v to load $\alpha_{v,j}$ components of task j in port
 ${}^i t_{v,j}$ = the time (in hours) needed for vessel v to install $\alpha_{v,j}$ components of task j in site
 ${}^t t_{v,j}$ = the time (in hours) required for vessel v to transport components of task j from port to site.
 ${}^m t_{v,j}$ = the maximum periods (in days) allowed for vessel v to do task j in one trip
 $a_{v,t}$ = the availability of vessel v at period t ($=1$ if available, $=0$ otherwise)
 $\lambda_{v,j}$ = 1 if vessel v can be used to perform task j , $= 0$ otherwise

- w_t = the weather forecast at period t ($w_t = 0$ for good weather, $=1$ for medium, and $=2$ for bad)
- σ_j = the worst weather condition that task j can be performed. In our study, we set $\sigma_0 = \sigma_1 = 1$ and $\sigma_2 = 0$ meaning that installing sub-structure and cable can be done in medium conditions while top-structure must be installed in good weather. In addition, loading and transporting components can be conducted in medium conditions, whilst in bad weather no activities can be performed.

As the number of parameters involved in our method is relatively large, it is not easy to develop a mathematical model for this problem. Moreover, the unit of measurement used by a parameter is not necessarily the same with another. For example, weather conditions are forecast daily, while the processing time required to install components is in hours. We propose an approach to reduce the complexity of formulating the problem into a mathematical model. This can be done by introducing a new set called set of feasible slots/trips for vessel v to conduct task j ($S_{v,j}$). Here, the problem is to find the best slot configuration from the feasible slots that minimise total installation cost or total completion period. Therefore, the problem will be treated as a combinatorial optimisation problem. The next subsection will describe the procedure to generate feasible slots for a vessel to perform installation tasks.

3.1. The procedure for generating feasible slots

Generating feasible slots/trips can be done by considering loading time ($^l t_{v,j}$), transporting time ($^t t_{v,j}$), installation time ($^i t_{v,j}$), maximum hired periods per trip ($^m t_{v,j}$), availability of vessel ($a_{v,t}$), and weather conditions (w_t). The set of feasible slots ($S_{v,j}$) consists of several parameters as follows:

- $h_{v,j,s,t}$ = 1 if in slot s , vessel v is chartered at period t for performing task j ,
= 0 otherwise.
- $b_{v,j,s,t}$ = 1 if slot s of vessel v starts at period t to conduct task j ,
= 0 otherwise.
- $\beta_{v,j,s}$ = the starting period (integer) of slot s of vessel v to do task j .

$d_{v,j,s}$ = the hired duration (integer) of slot s of vessel v to perform task j .

Figure 3 shows a simple example of feasible slots of a vessel for installing a top-structure within 20 days where ${}^t t_{v,j} + {}^l t_{v,j} = 24$ (1 day), ${}^i t_{v,j} + {}^t t_{v,j} = 72$ (3 days), and ${}^m t_{v,j} = 5$ days. Weather conditions and availability of the vessel are also given in the figure.

Period	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Bad weather																				
Medium weather																				
Availability of vessel																				
Slot 1 (4 days)																				
Slot 2 (5 days)																				
Slot 3 (5 days)																				
Slot 4 (5 days)																				
Slot 5 (5 days)																				
Slot 6 (5 days)																				
Slot 7 (5 days)																				
Slot 8 (4 days)																				

Figure 3. An example of feasible slots of a vessel for installing top-structure

From Figure 3, it can be seen that the vessel will not be able to start at day 0 because of bad weather. Slot 1 starts at day 1 as loading and transportation are allowed in medium weather and it will finish at the end of day 4 as good weather will occur from day 2 to day 4. In slot 2, the vessel needs to be chartered for 5 days as top-structure cannot be installed in medium weather which happens in day 5. It means that idle time occurs on day 5 where the vessel has to wait for good weather. The vessel will not be able to start at day 4 because it will finish at the end of day 8 when the vessel is not available. Start time is not also possible at day 11 as this slot requires 6 days to finish (two-days idle) exceeding the maximum hired periods.

It is clear that this problem is a combinatorial optimisation problem where the best configuration of slots needs to be found taking into account some constraints. Figure 4 presents the procedure to generate feasible slots. The procedure of Figure 4 assumes that ${}^t t_{v,j} + {}^l t_{v,j} \geq 24$ for vessels installing top-structure and the period is in days (integer number). The procedure will populate parameters $h_{v,j,s,t}$, $b_{v,j,s,t}$, $\beta_{v,j,s}$, and $d_{v,j,s}$ which will be used for developing the formulation of mathematical model of the installation scheduling problem.

Figure 4. The procedure for generating feasible slots

3.2. Mathematical model

In this subsection, we present a mathematical model using integer linear programming (ILP) for installation scheduling of offshore wind farm. The decision variable is as follow:

$x_{v,j,s}$ = 1 if slot s of vessel v to perform task j is used in the optimal solution,
 = 0 otherwise

The followings are the objective functions that are considered in our model:

- Minimising total installation cost

The objective is to find the optimal schedule for vessels that minimises total installation cost (Z_c) which can be written as follows:

$$\text{Minimise } (Z_c = \sum_v \sum_j \sum_{s \in S_{v,j}} x_{v,j,s} \cdot (d_{v,j,s} \cdot c_{v,j} + \hat{c}_{v,j})) \quad (1)$$

- Minimising total completion period/date

The objective measures total completion period (Z_t) as sum of completion period of installation tasks for all wind turbines.

$$\text{Minimise } (Z_t = \sum_v \sum_j \sum_{s \in S_{v,j}} x_{v,j,s} \cdot (\beta_{v,j,s} + d_{v,j,s} - 1)) \quad (2)$$

Subject to following constraints:

$$\sum_j \sum_{s \in S_{v,j}} (x_{v,j,s} \cdot h_{v,j,s,t}) \leq 1, \forall v \in V, t \in T \quad (3)$$

$$\sum_{v \in V_j} \sum_{s \in S_{v,j}} x_{v,j,s} \cdot \alpha_{v,j} \geq N, \forall j \in J \quad (4)$$

$$\sum_v \sum_j \sum_{s \in S_{v,j}} (x_{v,j,s} \cdot b_{v,j,s,t}) \leq 1, \forall t \in T \quad (5)$$

$$\begin{aligned} \sum_{t=1}^{t=t'} \sum_{v \in V(j+1)} \sum_{s \in S_{v,(j+1)}} (x_{v,(j+1),s} \cdot b_{v,(j+1),s,t} \cdot \alpha_{v,(j+1)}) \leq \\ \sum_{t=1}^{t=t'} \sum_{v \in V_j} \sum_{s \in S_{v,j}} (x_{v,j,s} \cdot b_{v,j,s,t} \cdot \alpha_{v,j}), j = 1, \dots, (|J| - 1), \forall t' \in T \end{aligned} \quad (6)$$

Constraint 3 ensures that there are no overlapping slots allocated to a vessel. Constraint 4 makes sure that the sum of the built components is greater than or equal to the total number of wind turbines to be installed. The equality operator could be used in this constraint, however it will be more difficult to solve and might obtain a higher objective function value. Constraint 5 assures that only one vessel can do loading in the port at a period. Constraint 6 guarantees that in period t the sum of installed components of task $(j+1)$ by all vessels does not exceed the sum of installed components of task j .

3.3. Compromise programming method for solving bi-objective problem

Multi-objective problems can be approached by several methods such as goal programming, Pareto efficient set generation, and compromise programming. In this study, Compromise Programming (CP) is implemented to solve the installation scheduling problem of offshore wind farm in the presence of two objectives. This CP technique is shown to work well for bi-objective problems (Romero *et al.*, 1998) and does not require the information of goal target values needed by the goal programming method. Another advantage of this method is that it will reduce the amount of computation, especially when compared with Pareto efficient set generation methods.

CP was introduced by Zeleny (1973). According to Romero and Rehman (1989), this method aims to select a solution from the set of efficient solutions based on a reasonable assumption that any decision maker seeks a solution as close as possible to the ideal point. CP minimises a set of weighted, scaled distances between the ideal and efficient solutions (Jones, 2011). Gan *et al.* (1996) provide a brief explanation how CP works.

Romero *et al.* (1998) studied connections between the multi-criteria techniques of goal programming, compromise programming, and the reference point method. Compromise-based approach for road project selection in Madrid metropolitan area was investigated by Ballesterro *et al.* (2003). Metaheuristic CP for the solution of multiple-objective scheduling problems was introduced by Gagné *et al.* (2005). André *et al.* (2007) applied CP for macroeconomic policy making in a general equilibrium framework and they used Spanish economy as a study case. Ballesterro (2007) treated CP as the maximization of the decision maker's additive utility function. Fattahi and Fayyaz (2010) investigated a CP model to integrate urban water management considering satisfaction of the urban water consumers, the national benefits and social hazards as objectives. Amiri *et al.* (2011) proposed a model called nadir CP for optimization of multi-objective portfolio problem. Liberatore *et al.* (2014) implemented CP for optimising recovery operations and distribution of emergency goods. Kanellopoulos *et al.* (2015) proposed an approach for CP that can be used for scenario assessments.

In this method, a distance function is used to measure the closeness between a solution and the ideal point where a family of L_p metrics is usually utilised. The general formulation of a CP approach is expressed as follows:

$$\text{Min } L_p = \left(\sum_{i=1}^n \left| \hat{w}_i \frac{Z_i(x) - Z_i^*}{Z_{i*} - Z_i^*} \right|^p \right)^{\frac{1}{p}} \quad (7)$$

where

p indicates the distance measure with p in range $[1, \infty]$.

n is the number of objectives.

Z_i^* is the ideal solution of objective i .

Z_{i*} is the anti-ideal solution of objective i .

$Z_i(x)$ is the compromise solution that minimises L_p .

$\hat{w}_i$ is the weight/importance of objective i relative to the other objectives.

In this paper, the value of p is set to 1 and ∞ as this will allow the calculation of all intermediate compromise set points as the problem is bi-objective ($n = 2$). In the case $p = 1$, Equation (7) takes the following form:

$$\text{Min } L_1 = \text{Min } \sum_{i=1}^n \hat{w}_i \frac{|Z_i(x) - Z_i^*|}{|Z_{i*} - Z_i^*|} \quad (8)$$

whereas if $p = \infty$, the objective function (7) aims to minimise the maximum deviation (π) as follows:

$$\text{Min } L_\infty = \text{Min } \pi \quad (9)$$

$$\text{s.t. } \hat{w}_i \frac{|Z_i(x) - Z_i^*|}{|Z_{i*} - Z_i^*|} \leq \pi, \forall i = 1, \dots, n \quad (10)$$

Figure 5 presents the main steps of CP for solving the installation scheduling problem of an offshore wind farm incorporating an exact method. We refer to this procedure as “CP with exact method”. The procedure involves three stages. The first stage is to generate feasible slots of vessels to perform installation tasks presented in Subsection 3.1. The second stage is to obtain the ideal solution by optimising each objective (total installation cost and total completion date) separately subject to constraints 3 to 6. In this problem, we assume that an optimal slot configuration that minimises an objective is anti-ideal or nadir point for other objective. This is due to the fact that the objectives are in conflict with each other.

Stage 1

- Set arrays $h_{v,j,s,t} = \phi, b_{v,j,s,t} = \phi, \beta_{v,j,s} = \phi$, and $d_{v,j,s} = \phi$.
- Generate feasible slots of vessels to perform installation tasks using the procedure described in Subsection 3.1. As the result, the arrays $h_{v,j,s,t}, b_{v,j,s,t}, \beta_{v,j,s}$, and $d_{v,j,s}$ will be populated.

Stage 2

- Solve minimising installation total cost problem (Equation 1) subject to constraints 3 to 6. Let Z_c^* be the total cost obtained and Z_t^* the total completion period.
- Solve minimising total completion period problem (Equation 2) subject to constraints 3 to 6. Let Z_t^* denote the total completion period obtained and Z_c^* the total installation cost.

Stage 3

- Solve minimising L_1 problem subject to constraints 3 to 6 where

$$L_1 = \frac{w((\sum_v \sum_j \sum_{s \in S_{v,j}} x_{v,j,s} \cdot d_{v,j,s} \cdot c_{v,j}) - Z_c^*)}{Z_c^* - Z_c^*} + \frac{(1-w)((\sum_v \sum_j \sum_{s \in S_{v,j}} x_{v,j,s} \cdot (\beta_{v,j,s} + d_{v,j,s} - 1)) - Z_t^*)}{Z_t^* - Z_t^*} \quad (11)$$

and w is the weight (parameter) of the first objective (total installation cost). Let Z_c^1 denote the total cost obtained and Z_t^1 the total completion period.

- Solve minimising L_∞ problem where

$$L_\infty = \pi \quad (12)$$

subject to constraints 3 to 6 with additional constraints as follow:

$$\frac{w((\sum_v \sum_j \sum_{s \in S_{v,j}} x_{v,j,s} \cdot d_{v,j,s} \cdot c_{v,j}) - Z_c^*)}{Z_c^* - Z_c^*} \leq \pi \quad (13)$$

$$\frac{(1-w)((\sum_v \sum_j \sum_{s \in S_{v,j}} x_{v,j,s} \cdot (\beta_{v,j,s} + d_{v,j,s} - 1)) - Z_t^*)}{Z_t^* - Z_t^*} \leq \pi \quad (14)$$

Let Z_c^∞ be the total cost obtained and Z_t^∞ the total completion period.

- Compromise solutions are bounded by L_1 and L_∞ .

Figure 5. The procedure of CP with exact method for the scheduling problem

The final stage is to find the solutions that minimise L_1 and L_∞ as the compromise solutions are bounded by L_1 and L_∞ . Here, the ILPs will be solved using a linear programming optimiser, namely CPLEX.

Figure 6 shows a chart that describes compromise solutions for the installation scheduling problem of offshore wind farm. Points A and B are the ideal solutions for minimising total completion period and minimising total cost problems respectively. For the bi-objective problem, Point F is the ideal point whereas Point E is the anti-ideal or nadir

point. All compromise solutions are bounded by Points C and D. The decision maker (wind farm operator) will choose from within this solution set based on their individual preferences. A decision maker who prefers balance between the two objectives tends towards point C, whereas a decision maker more interested in efficiency tending towards point D.

Figure 6. Compromise solutions in the installation scheduling problem

4. Metaheuristic techniques for the bi-objective scheduling problem

When the number of periods and vessels included in the model are relatively large, it is hard to solve the ILP using exact method (CPLEX). Therefore, we propose a CP method incorporating Variable Neighbourhood Search (VNS) and Simulated Annealing (SA) to overcome this limitation which we refer to as “CP with VNS” and “CP with SA” respectively. The procedure of the method (CP with metaheuristics) is depicted in Figure 7 which consists of four stages.

The procedure in Stage 1 of Figure 7 is the same as the one in Stage 1 of Figure 5 where feasible slots of vessels to perform installation tasks are generated. In Stage 2, procedures based on a greedy algorithm are put forward to find the slot configuration that minimises total installation cost or total completion period. The solutions obtained (x_t and x_c) are then fed into next stages as the initial solution. The main steps of the greedy algorithms proposed are presented in Subsection 4.1.

Stage 1

This stage is the same as Stage 1 of Figure 5.

Stage 2

- a. Apply a multi-start method incorporating the greedy algorithm described in Figure 8 to find the solution that minimises total installation cost. Let X_c be the obtained slot configuration.
- b. Implement a multi-start method involving the greedy algorithm presented in Figure 8 to seek the slot configuration that minimises the total completion period. Let X_t be the obtained slot configuration.

Stage 3

- a. Apply VNS or SA to find the slot configuration that minimises the total installation cost with X_c as the initial solution. Let $\tilde{Z}_c$ denote the total cost obtained and Z_t the total completion period.
- b. Starting from X_t as the initial solution, implement VNS or SA to seek the solution that minimises the total completion period. Let $\tilde{Z}_t$ be the total completion period attained and Z_c the total cost.

Stage 4

- a. Implement VNS or SA to find the best solution that minimises L_1 with X_t as the initial solution. Let $\tilde{Z}_c^1$ denote the total installation cost obtained and $\tilde{Z}_t^1$ the total completion period.
- b. Apply VNS or SA to search the slot configuration that minimise L_∞ with the solution from previous stage (Stage 4a) as the initial solution. Let $\tilde{Z}_c^\infty$ be the total installation cost obtained and $\tilde{Z}_t^\infty$ the total completion period.

Note that in the L_1 and L_∞ formulations (Equations 11, 12, and 13), replace Z_c^* , Z_t^* , Z_t^* , and Z_{c^*} with $\tilde{Z}_c$, Z_t , $\tilde{Z}_t$, and Z_c respectively.

Figure 7. The procedure of CP with metaheuristics for the bi-objective scheduling problem

VNS or SA is implemented in Stage 3 and Stage 4 to improve the solutions obtained in Stage 2. Here, we develop VNS algorithm as well as a local search and SA algorithm for solving the installation scheduling problem. Stage 3 searches the ideal and anti-ideal solutions for each objective whereas Stage 4 finds compromise solutions represented by solutions that minimise L_1 and L_∞ respectively. In this method, the ideal solutions obtained in Stage 3 may not be the optimum solution.

4.1. Greedy Algorithms for solving the installation scheduling problem

This subsection presents the description of greedy algorithms for solving the installation scheduling of the offshore wind farm. The aim of this algorithm is to find a relatively good solution used as the initial solution for VNS and SA. Figure 8 describes our greedy algorithms used to find the slot configurations that minimise total installation cost and total completion period. As the algorithms involve a random number generator, a multi-start method is implemented. In other words, the greedy algorithm is executed for a certain number of iterations (η) and then the best solution is chosen.

The followings are notations of sets and parameters with their description used in our greedy algorithms:

- $O_{v,j}$ = the current solution comprising set of slots of vessel v to perform task j , $O_{v,j} \subset S_{v,j}$
- $v_{j,t}$ = the number of components of task j installed at period t
- μ_j = total components of task j that have been installed
- ρ_v = the candidate slot of vessel v that might be chosen for the solution
- τ_t = true, if a loading process in port is conducted at period t
= false, otherwise
- $Q_{v,j}$ = set of slots of vessel v to do task j sorted by installation cost per component,
($r c_{v,j} \cdot d_{v,j,s} / \alpha_{v,j}$), in increasing order

The greedy algorithms have a simple structure and need little computational effort so a multi-start method can be implemented. First, for each task, the feasible slots of each vessel are sorted by the installation cost per component (for minimising total installation cost problem) or by the earliest start time (for minimising total completion period problem) in increasing order. In other words, the best slot is put in the first place. Second, the algorithms start by scheduling the first installation task ($j=0$, sub-structure installation) until all components have been installed, then the second installation task, and so on. When scheduling an installation task, for each vessel, the algorithms determine a slot that could enter into the solution taking into account all defined constraints. A vessel with the best slot is then selected and its slot is put into the solution ($O_{v,j}$). The selection criteria are depended on the objective of the problem. This process continues until the number of components that needs to be installed is reached. Finally, the objective function values (Z_c and Z_t) are calculated.

- Set $O_{v,j} = \phi$, $x_{v,j,s} = false \ \forall v \in V, j \in J, s \in S_{v,j}$, $\mu_j = 0 \ \forall j \in J$, $\tau_t = false \ \forall t \in T$, and $v_{j,t} = 0 \ \forall j \in J, t \in T$.
- For minimising total cost problem: Populate array $Q_{v,j}$ based on $S_{v,j}$.
- For each task j in J do the followings:
 - Set $\rho_v = 0 \ \forall v \in V_j$ (start from the slot that has the smallest cost/earliest time).
 - While $\mu_j < N$ do the following:
 - For each vessel v in V_j do the following:
 - Let $\rho^* = \rho_v$
 - Update $\rho_v = s^*$, where:
 - For minimising total cost problem:
 - s^* is the slot with the smallest cost ($s^* \in Q_{v,j}$, $\rho^* \leq s^* \leq |Q_{v,j}|$, and $s^* \notin O_{v,j}$).
 - For minimising total installation time problem:
 - s^* is the slot with the earliest start time ($s^* \in S_{v,j}$, $\rho^* \leq s^* \leq |S_{v,j}|$, and $s^* \notin O_{v,j}$).
 - It must satisfy the following constraints (let $t' = \beta_{v,j,s^*}$):
 - $\tau_{t'} = false$, meaning that in period t' there is no vessel loading components in port.
 - Slot s^* of vessel v is not overlapping with other slots of this vessel that are used in the solution.
 - If $j > 0$, ensures that the constraint $\sum_{t=0}^{t'} v_{j-1,t} > \sum_{t=0}^{t'} v_{j,t}$ is met which means that the components of task $(j-1)$ must be installed first before installing components of task j .
 - End for v
 - Choose vessel v^* , where
 - For minimising total cost problem:
 - Slot ρ_{v^*} provides the smallest cost into the solution
 - $(v^* = Arg(Min_{v \in V_j} ((c_{v,j} \cdot d_{v,j,\rho_v}) / \alpha_{v,j})))$. Ties may be broken arbitrarily.
 - For minimising total installation time problem:
 - Slot ρ_{v^*} gives the earliest start time into the solution
 - $(v^* = Arg(Min_{v \in V_j} (\beta_{v,j,\rho_v})))$. Ties may be broken arbitrarily.
 - Let $t'' = \beta_{v^*,j,\rho_{v^*}}$, update the following variables: $O_{v^*,j} = O_{v^*,j} \cup \rho_{v^*}$, $x_{v^*,j,\rho_{v^*}} = true$, $\mu_j = \mu_j + \alpha_{v,j}$, $\tau_{t''} = true$, and $v_{j,t''} = \alpha_{v,j}$.
 - End While
 - End for j
 - Calculate the objective function values (Z_c or Z_t)

Figure 8. Greedy algorithm for minimising the total cost and the total time

4.2. VNS for solving the scheduling problems

Variable Neighbourhood Search (VNS) is first introduced by Brimberg and Mladenovic (1996) for solving continuous location-allocation problems. VNS was formally formulated by Hansen and Mladenovic (1997) who investigated the p -median problem. VNS is a metaheuristic technique that consists of local search and neighbourhood change. The local search seeks for local optimality, whereas the neighbourhood search aims to escape from the local optima. In the neighbourhood change, a larger neighbourhood is systematically used if an improvement is not found, otherwise it will revert back to the smaller one. In the VNS, the smallest neighbourhood is the one that is closest to the current solution whereas the largest one farthest from the current solution (Hansen and Mladenovic, 1997). For more detailed information on VNS, Hansen and Mladenovic (2001) and Hansen *et al.* (2010) provided an excellent explanation on variants and successful applications of VNS.

Basic Variable Neighbourhood Search (BVNS) presented in Figure 9 is used in our VNS algorithm.

Initialization

- Choose the set of neighbourhood structures N_k , for $k = 1, \dots, k_{\max}$.
- Generate an initial solution x and define a stopping condition.

Main Step

Repeat the following steps until the stopping condition is met:

(1) Set $k = 1$

(2) While $k \leq k_{\max}$ do the following steps

(a) Shaking

Generate a point x' at random from the k th neighbourhood of x ($x' \in N_k(x)$).

(b) Local Search

Apply the local search with x' as initial solution and let x'' be the obtained local minima.

(c) Move or not

If the local optima x'' is better than the incumbent x , update $x \leftarrow x''$ and continue the search with $N_1(k \leftarrow 1)$; otherwise set $k \leftarrow k + 1$.

Figure 9. Steps of the basic VNS (Hansen and Mladenovic, 2001)

Our VNS algorithm for solving installation scheduling problems is given in Figure 10. Initially, the solution obtained by greedy algorithms is used as the initial solution. Here, z could be Z_c , Z_t , L_1 , or L_∞ .

Initialization

Take the solution $(x_{v,j,s}, O_{v,j}, \mu_j, \tau_t, v_{j,t}, \text{ and } z)$ obtained by the greedy algorithm as an initial solution and copy into $x'_{v,j,s}, O'_{v,j}, \mu'_j, \tau'_t, v'_{j,t}, \text{ and } z'$ respectively.

Main Step

- (1) Set $k = 1$, $\hat{z} = z$ and $\partial = 0$
- (2) While $k \leq k_{\max}$ do the following steps

(a) Shaking

Do the following steps k times:

- Choose a slot from the current solution $(O'_{v,j})$ to be removed randomly. Firstly, pick randomly a task $(\hat{j})$ and then choose randomly a vessel $(\hat{v})$ in $V_{\hat{j}}$. Finally a slot $(\hat{s})$ in $O'_{\hat{v},\hat{j}}$ is selected randomly.
- If $\partial = 0$
Find a slot to be inserted (j^*, v^*, s^*) by using procedure “FindBestSlot($\hat{j}, \hat{v}, \hat{s}, j^*, v^*, s^*, \gamma$)” which is described in Figure 11. Let γ is the saving returned by the procedure.
Else
Pick a slot to be inserted (j^*, v^*, s^*) randomly satisfying the constraints and calculate the saving (γ) .
End If
- Set $x'_{\hat{v},\hat{j},\hat{s}} = false$, $x'_{v^*,j^*,s^*} = true$, and $z' = z' - \gamma$.
- Update $O'_{v,j}, \mu'_j, \tau'_t$, and $v'_{j,t}$ accordingly;

(b) Local Search

Apply the local search proposed with $x'_{v,j,s}, O'_{v,j}, \mu'_j, \tau'_t, v'_{j,t}$, and z' as input and output values (see Figure 12).

(c) Move or not

If $z' < z$ then

Set $x_{v,j,s} = x'_{v,j,s}, O_{v,j} = O'_{v,j}, \mu_j = \mu'_j, \tau_t = \tau'_t, v_{j,t} = v'_{j,t}, z = z'$, and $k = 1$.

Else

Set $x'_{v,j,s} = x_{v,j,s}, O'_{v,j} = O_{v,j}, \mu'_j = \mu_j, \tau'_t = \tau_t, v'_{j,t} = v_{j,t}, z' = z$, and $k = k + 1$.

End If

- (3) If $z < \hat{z}$ then $\partial = 0$ and $\hat{z} = z$

Else $\partial = \partial + 1$

- (4) If $\partial = c_{\max}$ then stop

Else go to Step (2)

Figure 10. Our VNS for solving the scheduling problems

In our method, the procedure of VNS is repeated until there is no improvement in $C_{\max}$ iterations. In the VNS, the neighbourhood search is conducted by ‘shaking’ the current solution. In our method, the shaking process is done by swapping a randomly chosen slot (in current solution, $O_{v,j}$) with either a slot obtained by the procedure in Figure 11 which we refer to as procedure “FindBestSlot” or a randomly slot chosen (to diversify the search even more). In the local search, the interchange heuristic is implemented with involving procedure “FindBestSlot” as well.

Procedure FindBestSlot ($\hat{j}, \hat{v}, \hat{s}, j^*, v^*, s^*, \gamma$)

- Set $j^* = \hat{j}$ and $\gamma = -\infty$.
- For each vessel v in V_{j^*} do the following:
 - For each slot s in S_{v,j^*} ($s \notin O'_{v,j}$) do the following:
 - Calculate the saving (γ') made by the swapping where:
 - $\gamma' = ({}^r c_{\hat{v}, \hat{j}} \cdot d_{\hat{v}, \hat{j}, \hat{s}}) - ({}^r c_{v, j^*} \cdot d_{v, j^*, s})$ for the minimising total cost problem
 - $\gamma' = (\beta_{\hat{v}, \hat{j}, \hat{s}} + d_{\hat{v}, \hat{j}, \hat{s}}) - (\beta_{v, j^*, s} + d_{v, j^*, s})$ for the minimising total completion period problem
 - $\gamma' = z - \tilde{L}_1$ and $\gamma' = z - \tilde{L}_\infty$ for the minimising L_1 and L_∞ problems respectively where $\tilde{L}_1$ and $\tilde{L}_\infty$ are the new objective function values after the swap occurs.
 - Check whether the swap satisfies all constraints:
 - $\sum_{t=0}^{t'} v_{j-1,t} > \sum_{t=0}^{t'} v_{j,t}$, $\forall t \in T, j = 1, \dots, (|J| - 1)$
 - $\tau_{\beta_{v, j^*, s}} = false$.
 - There is no overlapping slot for each vessel $v \in V_{j^*}$.
 - If all constraints are met then
 - If $\gamma' > \gamma$ then
 - Set $\gamma = \gamma', v^* = v$, and $s^* = s$.
 - End If
 - End If
 - End for s
 - End for v

Figure 11. The procedure “FindBestSlot”

The procedure “FindBestSlot” in Figure 11 aims to find slot s^* to be swapped with slot $\hat{s}$ where $s^* \notin O'_{v,j}$ and $\hat{s} \in O'_{v,j}$. In other words, slot $\hat{s}$ of vessel $\hat{v}$ (to perform task $\hat{j}$) is to

be removed from the current solution and will be replaced by slot s^* of vessel v^* (to do task j^*). In this procedure, $\hat{s}$, $\hat{v}$, and $\hat{j}$ are known. The swapping process is only allowed within the same task ($j^* = \hat{j}$), whereas v^* could be different to $\hat{v}$ (different vessel). The procedure seeks a slot in $S_{v,\hat{j}}$ that gives the highest saving when swapping occurs considering all defined constraints.

Our local search is given in Figure 12. It is based on the fast interchange heuristic introduced by Whitaker (1983) who studies large-scale clustering and median location problems. We adapted the algorithm so it will fit to the installation scheduling problem of offshore wind farm. The local procedure comprises three steps. The first step is an iteration phase where the algorithm searches the best slot ($\hat{s}$, $\hat{v}$, and $\hat{j}$) to be removed from the current solution ($\hat{s} \in O'_{v,\hat{j}}$) and the best slot (j', v', s') to be inserted into the solution. The second step is a termination phase where the local search will stop if there is no improvement. In the last step, variables affected by the swapping are updated. In our local search, we implement a best improvement strategy instead of a first improvement.

Procedure Local Search ($x'_{v,j,s}$, $O'_{v,j}$, μ'_j , τ'_t , $v'_{j,t}$, z')

Step 1

- Set $\theta = -\infty$ (θ is the best saving occurred from swapping).
- For each task j in J do the following:
 - For each vessel v in V_j do the following:
 - For each slot s in $O'_{v,j}$ do the following:
 - Remove slot s current solution ($O'_{v,j}$).
 - Find a slot to be inserted (j', v', s') and determine the saving (γ) by using procedure "FindBestSlot ($j, v, s, j', v', s', \gamma$)".
 - If $\gamma > \theta$ then set $\hat{j} = j, \hat{v} = v, \hat{s} = s, j^* = j', v^* = v', s^* = s'$, and $\theta = \gamma$

Step 2

If $\theta \leq 0$ then stop.

Step 3

- Set $x'_{\hat{v},\hat{j},\hat{s}} = false$, $x'_{v^*,j^*,s^*} = true$, and $z' = z' - \theta$.
- Update $O'_{v,j}$, μ'_j , τ'_t , $v'_{j,t}$, and z' accordingly;
- Go to Step 1

Figure 12. The local search for solving the scheduling problems

4.3. Simulated Annealing for solving the scheduling problem

Simulated annealing (SA) is a metaheuristic used to solve optimization problems. Kirkpatrick *et al.* (1983) first introduced SA to search for feasible solutions and converge to an optimal solution. The idea of behind SA comes from a method proposed by Metropolis *et al.* (1953) where the cooling of material in a heat bath is simulated. The method simulates the cooling process by gradually decreasing the temperature of the system until it has converged to a steady (freeze) state. This is a process called annealing. For more detailed information on SA, Nikolaev and Jacobson (2010) and Dowsland and Thompson (2012) explain the algorithm and application of SA. Our SA algorithm for solving installation scheduling problems is presented in Figure 13.

Initialization

Take the solution $(x_{v,j,s}, O_{v,j}, \mu_j, \tau_t, v_{j,t}, \text{ and } z)$ obtained by the greedy algorithm as an initial solution and copy into $x'_{v,j,s}, O'_{v,j}, \mu'_j, \tau'_t, v'_{j,t}, \text{ and } z'$ respectively.

Main Step

- (1) Define $\hat{T}$, $\hat{T}_{\min}$, L , and φ .
- (2) While $\hat{T} > \hat{T}_{\min}$ do the following steps:
 - (a) Set $l = 0$.
 - (b) While $l < L$ do the following steps:
 - Choose a slot from the current solution $(O'_{v,j})$ to be removed randomly. Select a task $(\hat{j})$ and a vessel $(\hat{v})$ in $V_{\hat{j}}$ randomly. Pick a slot $(\hat{s})$ in $O'_{\hat{v},\hat{j}}$ randomly as well.
 - Determine a slot to be inserted (j^*, v^*, s^*) into the solution by using procedure "FindBestSlot($\hat{j}, \hat{v}, \hat{s}, j^*, v^*, s^*, \gamma$)".
 - Calculate the saving (γ) and set $x'_{\hat{v},\hat{j},\hat{s}} = false$, $x'_{v^*,j^*,s^*} = true$, and $z' = z' - \gamma$.
 - Update $O'_{v,j}, \mu'_j, \tau'_t$, and $v'_{j,t}$ accordingly.
 - Calculate $\Delta = z' - z$
 - If $\Delta \leq 0$ then
 - Set $x_{v,j,s} = x'_{v,j,s}, O_{v,j} = O'_{v,j}, \mu_j = \mu'_j, \tau_t = \tau'_t, v_{j,t} = v'_{j,t}, \text{ and } z = z'$.
 - Else
 - Generate a random number (ℓ) from $U(0,1)$
 - If $\ell < \exp(-\Delta/\hat{T})$ then
 - Set $x_{v,j,s} = x'_{v,j,s}, O_{v,j} = O'_{v,j}, \mu_j = \mu'_j, \tau_t = \tau'_t, v_{j,t} = v'_{j,t}, z = z'$.
 - Else
 - Set $x'_{v,j,s} = x_{v,j,s}, O'_{v,j} = O_{v,j}, \mu'_j = \mu_j, \tau'_t = \tau_t, v'_{j,t} = v_{j,t}, z' = z$.
 - Set $l = l + 1$.
 - (c) Set $\hat{T} = \varphi \cdot \hat{T}$.

Figure 13. Our SA algorithm for solving the scheduling problems

Similar to our VNS approach, the initial solution used in SA is also obtained using greedy algorithms. In the procedure given in Figure 14, the parameters $\hat{T}$, $\hat{T}_{\min}$, L , and φ are defined first. $\hat{T}$ is temperature parameter dynamically adjusted using correction parameter φ (the rate of temperature decreasing). $\hat{T}_{\min}$ and L determine the termination condition of the search. In the search, the new solution is generated by swapping a randomly chosen slot (in current solution, $O_{v,j}$) with a slot obtained by implementing procedure “FindBestSlot” given in Figure 11. If there is an improvement ($\Delta \leq 0$), a new solution is automatically accepted. Otherwise it is accepted with probability by generating a random number (ℓ) in range $[0,1]$. The new solution is accepted if $\ell < \exp(-\Delta/\hat{T})$ which means that there is a chance to accept non-improvement (worse) solution in this method.

5. Computational study

We carried out extensive experiments to assess the performance of the proposed solution approaches. The code was written in C++ .Net 2012. The tests were run on a PC with an Intel Core i5 CPU @ 3.20GHz processor, 8 GB of RAM and under Windows 7.

Two sets of instances have been generated randomly and are referred to as datasets A and B. Datasets A and B consist of 5 and 10 vessels respectively ($|V|=5$ and 10) which represents the number of installation vessels usually involved in the installation process. The time period is measured in days and we set $|T|=365$ for all instances. The number of wind turbines (N) to be installed is set to 50 and 120 for dataset A and B respectively which represent typical medium and large wind farm sizes.

Table 1 presents an example of parameter values used in dataset A. These values are estimation values based on data from the literature. We randomly generate daily weather condition (good, medium, and bad weather) within 365 days. In the real problem, the weather forecast from meteorological (met) office can be used to support installation/operational activities in offshore wind farms. The availability of each vessel per period is also randomly generated.

Table 1. Dataset A ($|V| = 5$)

Vessel	$\lambda_{v,0}$	$\alpha_{v,0}$	$\hat{c}_{v,0}$	$r_{c_{v,0}}$	$l_{t_{v,0}}$	$t_{t_{v,0}}$	$i_{t_{v,0}}$	$m_{t_{v,0}}$
ves #1	0	0	0	0	0	0	0	0
ves #2	0	0	0	0	0	0	0	0
ves #3	1	6	100,000.00	140,000.00	10.00	1.50	110.00	10
ves #4	1	8	100,000.00	137,500.50	14.50	4.50	95.50	10
ves #5	1	8	100,000.00	157,500.50	13.50	3.50	85.50	10
Vessel	$\lambda_{v,1}$	$\alpha_{v,1}$	$\hat{c}_{v,1}$	$r_{c_{v,1}}$	$l_{t_{v,1}}$	$t_{t_{v,1}}$	$i_{t_{v,1}}$	$m_{t_{v,1}}$
ves #1	1	10	100,000.00	70,000.50	4.50	2.50	40.50	5
ves #2	1	12	100,000.00	72,000.50	7.50	3.50	55.50	6
ves #3	0	0	0	0	0	0	0	0
ves #4	0	0	0	0	0	0	0	0
ves #5	0	0	0	0	0	0	0	0
Vessel	$\lambda_{v,2}$	$\alpha_{v,2}$	$\hat{c}_{v,2}$	$r_{c_{v,2}}$	$l_{t_{v,2}}$	$t_{t_{v,2}}$	$i_{t_{v,2}}$	$m_{t_{v,2}}$
ves #1	0	0	0	0	0	0	0	0
ves #2	0	0	0	0	0	0	0	0
ves #3	1	6	100,000.00	140,000.00	16.00	1.50	130.50	15
ves #4	1	8	100,000.00	137,500.50	28.50	3.50	155.00	15
ves #5	1	8	100,000.00	157,500.50	24.50	2.50	135.00	15

In our computational study, we vary the value of w from 0.25, to 0.75 in increments of 0.25. Here, when $w = 0.25$ decision maker is more concerned to the total completion period rather than total installation cost, whilst when $w = 0.75$ the total cost desirable. In the compromise programming with the exact method, the ILPs are solved by using IBM ILOG CPLEX version 12.6 Concert Library. By solving the ILP using CPLEX, the optimal solution is attained. In the greedy algorithms, we set number of iterations (η) to 100 for both minimising total cost and minimising total time problems. In the “CP with VNS”, $c_{\max}$ and $k_{\max}$ are set to 2 and $5 \cdot |V|$ respectively whereas in the “CP with SA”, parameters $\hat{T}$, $\hat{T}_{\min}$, L , and φ are set to 1, 0.001, $4 \cdot |V|^2$, and 0.9. This parameter setting is based on our preliminary study. For the metaheuristic approaches (“CP with VNS” and “CP with SA”), the average results with their best one based on 5 runs on each instance are given.

Table 2 presents the summary of computational results of the exact method and our proposed method on dataset A and B for the minimising total installation cost and the

minimising total installation time. In the table, CPU time is measured in seconds. The table also provide Dev (in %) which is defined as:

$$Dev = 100 \cdot \left(\frac{Z_m - Z_e}{Z_m} \right), \text{ where } Z_m \text{ and } Z_e \text{ correspond to the } Z \text{ value obtained by metaheuristic}$$

techniques (VNS and SA) and the exact method respectively. The bold values in the tables indicate that the optimal solution is obtained.

Table 2. The summary of computational results

Dataset	Objective Function	Exact Method		VNS			SA		
				Best Result Dev (%)	Average Results		Best Result Dev (%)	Average Results	
		Z	CPU Time		Dev (%)	CPU Time		Dev (%)	CPU Time
V =5	Minimising Total Cost Problem	15,102,541.00	73.82	0	0	17.87	0	0	33.57
	Minimising Total Period Problem	867	66.84	0.81	0.81	20.07	1.04	1.18	41.34
V =10	Minimising Total Cost Problem	26,342,552.50	959.85	0	0	48.36	0	0	127.07
	Minimising Total Period Problem	3,212	1,224.70	6.57	7.48	336.57	10.93	14.51	228.25

According to the results shown in Table 2, the exact method requires more than thirteen times longer to solve dataset B compared to dataset A. In other words, when we increase the size of the problem, the computing time required for the exact method to solve the problem will increase exponentially. In general the metaheuristic methods run much faster than the exact method. For the minimising total installation cost problem, both VNS and SA are able to produce the optimal solution (for all runs) for both instances. For the minimising total installation completion period, VNS yields better deviation compared to SA. For this problem, based on the best result, VNS produces a deviation of 0.81% and 6.57% for dataset A and B respectively whereas SA provides 1.04% and 10.93%. Based on average results, VNS is also superior with respect to solution time when compared to SA. Here, it can be

observed that the minimising total installation cost problem is relatively easier to solve using our proposed metaheuristic methods than the minimising total completion period problem.

Tables 3 and 4 show the detail of computational results that present the compromise solutions for dataset A and B respectively. In the tables, Z_c and Z_t denote the total installation cost and the total completion period values respectively. CPU time is also measured in seconds. For the CP with metaheuristic approaches, the tables provide the compromise solutions from the best run.

Table 3. The compromise solutions for dataset A

Problem	Objective functions	CP with Exact Method	CP with VNS	CP with SA
Minimising Total Cost Problem	Z_c	15,102,541	15,102,541	15,102,541
	CPU Time	73.82	14.57	33.11
Minimising Total Period Problem	Z_t	867	874	877
	CPU Time	66.84	19.65	46.54
Minimising L_1 and L_∞ problems for $w = 0.25$	$Z_c (L_1)$	19,602,048	19,634,545	19,792,046
	$Z_t (L_1)$	913	920	922
	CPU Time (L_1)	68.35	28.32	35.54
	$Z_c (L_\infty)$	18,353,548	18,491,049	19,179,544
	$Z_t (L_\infty)$	1,019	1,032	1,040
	CPU Time (L_∞)	67.37	16.66	32.58
Minimising L_1 and L_∞ problems for $w = 0.5$	$Z_c (L_1)$	16,971,041	16,973,540	17,546,042
	$Z_t (L_1)$	1,191	1,204	1,138
	CPU Time (L_1)	67.67	32.82	32.20
	$Z_c (L_\infty)$	17,114,541	17,272,038	17,602,040
	$Z_t (L_\infty)$	1,174	1,179	1,183
	CPU Time (L_∞)	68.31	18.02	32.02
Minimising L_1 and L_∞ problems for $w = 0.75$	$Z_c (L_1)$	15,207,540	15,345,041	15,691,041
	$Z_t (L_1)$	1,577	1,560	1,494
	CPU Time (L_1)	67.12	40.49	32.28
	$Z_c (L_\infty)$	16,151,041	16,203,538	16,322,036
	$Z_t (L_\infty)$	1,344	1,343	1,371
	CPU Time (L_∞)	68.60	24.09	31.67
Average CPU Time		68.51	24.33	34.49

Table 4. The compromise solutions for dataset B

Problem	Objective functions	CP with Exact Method	CP with VNS	CP with SA
Minimising Total Cost Problem	Z_c	26,342,553	26,342,553	26,342,553
	CPU Time	959.85	47.16	127.03
Minimising Total Period Problem	Z_t	3,212	3,560	3,563
	CPU Time	1,224.70	164.99	254.03
Minimising L_1 and L_∞ problems for $w = 0.25$	$Z_c (L_1)$	33,143,553	35,768,047	35,401,547
	$Z_t (L_1)$	3,611	3,566	3,719
	CPU Time (L_1)	988.36	186.92	197.13
	$Z_c (L_\infty)$	31,376,055	32,970,547	33,448,046
	$Z_t (L_\infty)$	3,935	4,026	4,056
	CPU Time (L_∞)	1,254.08	219.29	197.12
Minimising L_1 and L_∞ problems for $w = 0.5$	$Z_c (L_1)$	28,075,055	31,395,547	31,414,547
	$Z_t (L_1)$	4,727	4,270	4,273
	CPU Time (L_1)	1,266.48	170.52	198.63
	$Z_c (L_\infty)$	29,075,055	30,581,547	30,999,547
	$Z_t (L_\infty)$	4,390	4,454	4,546
	CPU Time (L_∞)	1,007.00	175.70	188.30
Minimising L_1 and L_∞ problems for $w = 0.75$	$Z_c (L_1)$	26,572,553	28,066,546	28,323,046
	$Z_t (L_1)$	5,625	5,459	5,390
	CPU Time (L_1)	1,721.87	324.20	188.30
	$Z_c (L_\infty)$	27,666,054	28,753,046	28,970,547
	$Z_t (L_\infty)$	4,921	5,097	5,223
	CPU Time (L_∞)	1,475.67	162.44	187.68
Average CPU Time		1,237.25	181.40	192.28

The tables also show that the exact method requires more than eighteen times longer to solve dataset B compared to dataset A. Based on the results shown in the tables, in general the CP with metaheuristic methods runs much faster than the CP with exact method when solving the multi-objective compromise programming model. The computational time of the CP with VNS is slightly lower than the one of the CP with SA. The CP with VNS also produces relatively better solutions compared to the CP with SA as the solutions produced by

CP with VNS are not far from those by CP with exact method. The Dev (%) between the metaheuristic approaches and exact method for L_1 and L_∞ is not calculated as each approach uses different ideal and anti-ideal solutions.

The experiments show that the solutions obtained by either CP with exact method or with metaheuristic under L_1 and L_∞ are quite close to each other. As all the compromise solutions are bounded by L_1 and L_∞ , they are not much different from one another. Prior justification for selecting a solution on the compromise set bounded by L_1 and L_∞ is required. For example, in the case of dataset A and $w=0.5$ if the decision maker thinks that total completion period is more important, then the solution obtained by minimising L_∞ is a desirable option.

The results in the tables also reveal that when $w=0.25$, the solutions obtained by minimising L_1 and L_∞ are near to the solution attained by minimising total completion period problem. In contrast, when $w=0.75$, the solutions returned are near to the solution obtained by minimising total installation cost problem. It is due to the fact that w is the weight associated with the minimisation of total cost. In other words, if $w=1$ then the problem becomes the minimising total cost problem; conversely if $w=0$ then the minimising total completion period problem is considered.

6. Conclusion and suggestions

In this paper, we propose a multi-objective optimisation for installation scheduling of an offshore wind farm using the compromise programming method. A mathematical model of the scheduling problem is developed involving two objectives namely minimising total installation cost and minimising total completion period/date. Two approaches are proposed for solving the compromise programming problem. In the first approach a compromise programming with exact method (by using CPLEX) is put forward where the compromise optimal solutions could be obtained depending on the size of the problem. In the second approach the compromise programming with metaheuristic methods (VNS and SA) incorporating greedy algorithms are proposed to find a good solution in a reasonable computing time.

Two generated datasets are used for evaluating the performance of the proposed methods. The computational experiments show that the proposed approaches produce interesting results. The exact method is able to attain optimality for all datasets. The metaheuristic method performs well and runs much faster than the exact method. According to the computational results, the compromise programming with VNS outperforms the one with SA. The experiments also show that the solutions under L_1 and L_∞ are quite close to each other and therefore the solutions hence show a small amount of difference.

This study could be extended to investigate other related scheduling problems such as maintenance scheduling problem of offshore wind farm. Stochastic constraints such as weather condition uncertainty could also be considered in the model.

References

1. Ait-Alla, A., Quandt, M., Lütjen, M., Biba. (2013). Aggregate Installation Planning of Offshore Wind Farms. In Proceedings of the 7th International Conference on Communications and Information Technology (CIT'13) p. 130-135. Cambridge, US.
2. André, F.J, Cardenete, M.A., Romero, C. (2007). Using compromise programming for macroeconomic policymaking in a general equilibrium framework: theory and application to the Spanish economy. *Journal of the Operational Research Society*, 1-9.
3. Amiri, M., Ekhtiari, M., Yazdani, M. (2011). Nadir compromise programming: A model for optimization of multi-objective portfolio problem. *Expert Systems with Applications: An International Journal*, 38, 7222-7226.
4. Ballestero, E. (2007). Compromise programming: A utility-based linear-quadratic composite metric from the trade-off between achievement and balanced (non-corner) solutions. *European Journal of Operational Research*, 182, 1369–1382.
5. Ballestero, E., Antón, J.M., Bielza, C. (2003). Compromise-based approach to road project selection in Madrid metropolitan area. *Journal of the Operations Research Society of Japan*, 46, 99-122.
6. Benmessaoud, T., Mohammedi, K., Smaili, Y. (2013). Influence of maintenance on the performance of a wind farm. *Przegląd Elektrotechniczny*, 89, 174-178.
7. 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.

8. 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.
9. Brimberg, J., Mladenovic, N. (1996). A variable neighbourhood algorithm for solving the continuous location-allocation problem. *Studies of Locational Analysis*, 10, 1-12.
10. Byon, E., Pérez, E., Ding, Y., Ntamo, L. (2011). Simulation of wind farm operations and maintenance using discrete event system specification. *Simulation*, 87, 1093-1117.
11. Dai, L., Stålhane, M. Utne, I.B. (2015). Routing and scheduling of maintenance fleet for offshore wind farms. *Wind Engineering*, 39, 15–30.
12. Dos Santos, F.P., Teixeira, Â.P., Soares, C.G. (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.
13. Dowsland, K.A., Thompson, J. (2012). Simulated annealing. In Rozenberg, G., Back, T., Kok, J.N. (Eds.) Handbook of Natural Computing pp. 1623-1655. Springer Reference Springer-Verlag.
14. European Wind Energy Association (2014). Wind in power, 2013 European statistics. Available from: <http://www.ewea.org/statistics/>.
15. Fattahi, P., Fayyaz, S. (2010). A Compromise Programming Model to Integrated Urban Water Management. *Water Resources Management*, 24, 1211-1227.
16. Gagné, C., Gravel, M., Price, W.L. (2005). Using metaheuristic compromise programming for the solution of multiple-objective scheduling problems. *Journal of the Operational Research Society*, 56, 687-698.
17. Gan, J., Colletti, J.P., Kolison Jr, S.H. (1996). A Compromise Programming Approach to Integrated Natural Resource Management. In Society of American Foresters.
18. Ge, X., Zhang, L., Shu, J. (2013). A long-term optimal scheduling method for wind-hydro-thermal power systems. *Proceedings of the Chinese Society for Electrical Engineering*, 33, 136-144.
19. Hansen, P., Mladenovic, N. (1997). Variable neighbourhood search for the p -median. *Location Science*, 5, 207-225.
20. Hansen, P., Mladenovic, N. (2001). Variable neighbourhood search: Principles and applications. *European Journal of Operational Research*, 130, 449-467.
21. Hansen, P., Mladenovic, N., Perez, J. A. M. (2010). Variable neighbourhood search: methods and applications. *Annals of Operations Research*, 175, 367-407.

22. Jones, D.F. (2011). A practical weight sensitivity algorithm for goal and multiple objective programming. *European Journal of Operational Research*, 213, 238–245.
23. Kaiser, M.J., Snyder, B.F. (2013). Modeling offshore wind installation costs on the US Outer Continental Shelf. *Renewable Energy*, 50, 676-691.
24. Kanellopoulos, A., Gerdessen, J.C., Claassen, G.D.H. (2015). Compromise programming: Non-interactive calibration of utility-based metrics. *European Journal of Operational Research*, <http://dx.doi.org/10.1016/j.ejor.2015.01.031>.
25. Kirkpatrick, S., Gelatt, C., Vecchi, M. (1983). Optimization by Simulated Annealing. *Science*, 220, 671–680.
26. Kovács, A., Erdős, G., Viharos, Z.J., Monostori, L. (2011). A system for the detailed scheduling of wind farm maintenance. *CIRP Annals Manufacturing Technology*, 60, 497-501.
27. Liberatore, F., Ortuño, M.T., Tirado, G., Vitoriano, B., Scaparra, M.P. (2014). A hierarchical compromise model for the joint optimization of recovery operations and distribution of emergency goods in Humanitarian Logistics. *Computers & Operations Research*, 42, 3-13.
28. Lütjen, M., Karimi, H.R. (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) p. 502-508. Rhodes, Greek.
29. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E. (1953). Equation of State Calculation by Fast Computing Machines. *Journal of Chemical Physics*, 21, 1087-1091.
30. Nikolaev, A.G., Jacobson, S.H. (2010). Simulated Annealing. In Gendreau, G., Potvin, J. (Eds.), *Handbook of Metaheuristics* p. 1-39. Springer.
31. Pan, K., You, D.H., Wang, K. (2013). Long-term multi-objective optimization dispatch and its evaluation in wind integrated power systems. *Advanced Materials Research*, 732, 1033-1037.
32. Parikh, N.D. (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.
33. 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) p. 462-466. Rome, Italy.
34. Pérez, E., Ntamo, L., Ding, Y. (2013). Simulation of wind farm operations and maintenance. In Proceedings of the ASME Turbine Technical Conference and Exposition. San Antonio, US.

35. Perez-Canton, S., Rubio-Romero, J.C. (2013). A model for the preventive maintenance scheduling of power plants including wind farms. *Reliability Engineering & System Safety*, 119, 67-75.
36. Pérez, E., Ntamo, L., Byon, E., Ding, Y. (2010). A stochastic DEVS wind turbine component model for wind farm simulation. In Proceeding of the Spring Simulation Multi-conference (SpringSim). Florida, US.
37. Renewables Advisory Board (2010). Value breakdown for the offshore wind sector. Available from: <https://www.gov.uk/>.
38. Romero, C., Rehman, T. (1989). Multiple Criteria Analysis for Agricultural Decisions. Elsevier Science Publishers, Amsterdam, The Netherland.
39. Romero, C., Tamiz, M., Jones, D.F. (1998). Goal programming, compromise programming and reference point method formulations: linkages and utility interpretation. *Journal of the Operational Research Society*, 49, 986-991.
40. Scholz-Reiter, B., Heger, J., Lütjen, 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.
41. Scholz-Reiter, B., Karimi, H. R., Lütjen, M., Heger, J., Schweizer, A. (2011). Towards a Heuristic for Scheduling Offshore Installation Processes. In Maneesh, S., Rao, R.B.K.N., Liyanage, J.P. (Eds.), Proceedings of the 24th International Congress on Condition Monitoring (COMADEM) p. 999-1008. Stavanger, Norway.
42. Shafiee, M. (2015). Maintenance logistics organization for offshore wind energy: Current progress and future perspectives. *Renewable Energy*, 77, 182-193.
43. Stålhane, M., Hvattum, L.M., Skaar. (2014). Optimization of routing and scheduling of vessels to perform maintenance at offshore wind farms. Proceedings of the 12th Deep Sea Offshore Wind R&D Conference, EERA DeepWind'2015.
44. Van Horenbeek, A., Van Ostaeyen, J., Duflou, 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.
45. 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.
46. Whitaker, R.A. (1983). A fast algorithm for the greedy interchange for large-scale clustering and median location problems. *INFOR*, 21, 95-108
47. 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.

48. Zeleny, M. (1973). Compromise programming. In Cochrane, J.L., Zeleny, M. (eds). Multiple Criteria Decision Making. University of South Carolina, Press: Columbia, p. 262–301.
49. Zhang, J., Chowdhury, S., Zhang, J. (2012). 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.
50. Zhang, H., Yin, Y., Shen, H., He, J., Zhao, S. (2012). Generator maintenance scheduling of large-scale wind power integration considering peak shaving. *Automation of Electric Power Systems*, 36, 25-30.

- Development of a mathematical model of the installation scheduling problem
- Construction of greedy algorithms and a local search for finding the best schedule
- Application and comparison of VNS and SA for solving the scheduling problems.