ABSTRACT

As an important part of renewable energy resources, offshore wind energy has great potential compared to its onshore counterpart despite the vast developments in recent decades. However, due to the more complex environmental condition and physical restrictions the installation of an offshore wind farm is hard to plan and predict, which often results in delays. This paper focuses on the scheduling problem in the installation phase of an offshore wind farm. We propose an adaptive search strategy based on the Apriori property and information entropy. The purpose is to prune the search space effectively and intelligently to realize an agile and swift rescheduling according to the environmental changes. For the numerical experiments, we use the environmental data obtained from the German North Sea from the year 1958 to 2007 in hourly resolution.

1 INTRODUCTION

Renewable energy is, without doubt, the key to mitigating or even solving global warming, which is a severe environmental problem mainly resulting from the usage of dirty energy resources, such as fossil, coal, and fracking gas. As a crucial contributor to renewable energy resources, offshore wind energy (OWE) has drawn attention in the last 20 years. A recent report shows an exponential growth of OWE capacity worldwide from the year 1998 to the year 2020 (Lee and Zhao 2021). However, it also points out that the overall contribution of offshore wind energy to worldwide energy consumption is still at a low level. Furthermore, the offshore wind counts for only around 5% of the installed wind energy offshore and onshore combined.

Similar to other industrial projects, the OWT installation is encountering project delays. An effective way to overcome this issue is to improve the scheduling of the installation vessels under consideration of the dynamic weather conditions on the sea. In a recent study, Lerche et al. (2022) have analyzed the impact of different factors, such as financial, organizational, and environmental factors, on the project delay in OWF installation. Unsurprisingly, the weather is one the most significant factors that cause project delays regarding offshore operations. Besides, compared to the onshore wind farm installation, the offshore wind farm installation is more sensitive to the environment and has more physical restrictions. For example, the installation vessel cannot be jack-uped twice on the same construction site since the damaged sea bed cannot provide sufficient stability. This infers that once the installation process begins, it cannot be stopped even if the weather condition deteriorates during the installation. Such circumstance leads often directly to delays in the installation project. A closer look at the differences between onshore and offshore wind power is given by Bilgili et al. (2011) and a technical report regarding the state of the art of offshore wind turbine installation is given by Jiang (2021). Some other issues appearing in the offshore wind installation...
process are summarized in Rippel et al. (2019). It has been shown that computer-aided tools can help to schedule the installation projects efficiently and, thus, reduce project delays. Yet, studies have shown that the search space of such scheduling problems is positively correlated to factors such as desired time span, number of vessels, and desired workload. Improvements have been made regarding computational efficiency. For example, Peng et al. (2021a) have suggested using stochastic optimization methods such as simulated annealing. However, the nature of stochastic optimization methods leads to the fact that the global optima are not ensured. Another issue is that the overall optimal schedule is considered as a sequence of local optima. However, the strategy produces comparative predictions to the MILP approach (Rippel et al. 2020) despite the negligence of correlation. This work aims at improving the decision-support system proposed in Peng et al. (2021b) in the following manner: i) we first take advantage of the Apriori property to prune the search space and then we investigate the advantages and disadvantages of this approach; ii) we improve the pruning strategy by adding adaptivity.

The rest of the paper is organized as follows: section 2 gives a literature review of the state of the art regarding offshore installation and adaptivity. In section 3 the problem in the OWF installation is introduced and formulated as an optimization problem. Furthermore, we elaborate on the concept of search space pruning and propose an adaptive version using information entropy in section 4. The numerical results are presented in section 5 and the conclusion is drawn in section 6.

2 STATE-OF-THE-ART

In this section, we are going to review the related works in the literature. This will be separated into two parts. In the first part, we look at the state-of-art regarding OWF installation. In the second part, we focus on the application of adaptivity.

OWF Installation As stated in section 1 the best way to overcome the project delay caused by the weather is to consider it in the schedule of the OWF installation. In Chartron (2019), the authors have reviewed recent studies regarding offshore logistics. Essentially, mathematical approaches have been applied to model the offshore installation process to solve the planning and scheduling problems of the majority of the researchers. The Mixed Integer Linear Programming (MILP) model proposed by Scholz-Reiter et al. (2011) aims at scheduling offshore installation activities under the consideration of a single weather scenario, which has been extended by (Ait-Alla et al. 2013) with an aggregate planning strategy for minimizing the installation cost. To deal with the uncertainties in offshore weather, Herroelen and Leus (2005) have reviewed the fundamental approaches for scheduling that consider uncertainties, e.g., stochastic project scheduling, fuzzy project scheduling. Cardoso et al. (2013) have introduced uncertainty of products’ demand into their MILP model for the designing and planning of the general supply chains with reserve flows. A decomposition strategy proposed by Ursavas (2017) aims at improving the planning and scheduling to reduce the cost resulting from severe weather conditions.

Nonetheless, another bunch of recent works has focused on discrete-event simulations (DES) to investigate problems in the offshore wind industry. Vis and Ursavas (2016) proposed a decision-support tool based on DES to investigate the coherency between the logistical concepts and project performance. Muhabie et al. (2018) have investigated different assembly strategies used in the offshore installation by DES, including weather uncertainties, distances, vessel properties, and different assembly scenarios. It points out that offshore installation is strongly dependent on the season and geographical features of the construction site, and suggests combining different logistic strategies to achieve the best performance. Peng et al. (2020a) have proposed a TPN-based model to schedule the OWF installation process using a numerical search. Later on, the authors have presented a more general form of the TPN-based model using the Generalized Stochastic Petri Nets (GSPN) approach Peng et al. (2020b), which focused on the optimization of the system buffer, i.e., the base port. To improve the numerical efficiency of the search problem, Peng et al. (2021a) have shown the possibility of embedding a meta-heuristic algorithm into the DES model. In Peng et al. (2021b), the authors have presented a comprehensive study regarding the
scenario where multiple installation vessels are considered in the installation process and compared the quality of the schedules using DES simulation and mathematical programming method. Rippel et al. (2020) have shown the possibility of the complementary application of discrete-event simulation and mathematical optimization methods using a metamodel-based transformation framework. However, such methods are often associated with a large search space as the complexity of the problem increase. In order to increase the agility in the DES-based approaches, a strategy to reduce the search space is required.

Adaptivity The application of adaptivity is multifaceted in the literature. For example, adaptivity is one of the core values of the modern software development scheme SCRUM. Adaptivity in this context is to make appropriate adjustments to the development process according to the feedback obtained from daily scrums (Schwaber, Ken and Sutherland, Jeff ). However, within the scope of an algorithm adaptivity refers to the ability of the algorithm, that can adjust its behavior at runtime based on information available and/or a priori defined reward mechanism (Zaknich 2005). By adding adaptivity to the high-dimensional model representation (HDMR) technique, Ma and Zabaras (2010) have proposed an adaptive version of HDMR which detects the important dimensions automatically and constructs higher-order terms using only the important dimensions. Pang et al. (2016) focused on solving a non-smooth trajectory optimization problem, in which they proposed an adaptive method for mesh refinement, i.e., to detect the important location where a finer mesh is required. An adaptive approach was proposed by Li et al. (2019) to handle the uncertain log data in process mining. Ahmad et al. (2017) have given an overview on the application of adaptivity in human-robot interaction systems. Additionally, a bibliometric review on adaptivity was presented by Caya and Neto (2018). In numerical mathematics, there exist quite amount of work dealing with the numerical integration problem. Genz (1991) has proposed a globally adaptive algorithm for numerical multiple integration over an n-dimensional simplex. A dimension-adaptive quadrature method is proposed by Gerstner and Griebel (2003), which is based on sparse grid method. The fundamental idea is to select the integration points adaptively by evaluating the rewards obtained by considering the candidate integration points. Another field where adaptivity puts value is reconfigurable system. For example, Jung et al. (2006) have suggested an error threshold to the multiple model adaptive control scheme to guarantee its stability in closed-loop systems. Beck et al. (2014) addressed major drawbacks in reconfigurable systems, i.e., lack of transparency and lack of adaptivity in applications with different behaviors and characteristics. Thus, a binary translation mechanism, called Dynamic Instruction Merging, was proposed to handle these problems. Furthermore, Adaptivity has also been applied in solving flexible job shop problem. Cao et al. (2019) have proposed an A-HEFT scheduling algorithm, which extends the heterogeneous earliest finishing time (HEFT) algorithm by adding adaptivity, to handle the problem of scheduling stochastic jobs. Tang et al. (2015) suggested a platform-specific self-adaptive algorithm to reduce the make-span of tasks. In this work, a strategy is proposed for the offshore scheduling task, which aims at lowering the computational cost by adaptively reducing the search space with the help of predicted environmental information.

3 PROBLEM DESCRIPTION

A scheduling problem is a universal problem that exists in different domains. Different from the scheduling tasks in fields, such as CPU, and distributed computing, the scheduling of OWF installation is characterized by dynamic external weather conditions and physical restrictions. These obstacles result in immense project delays and are the main reason for the much lower construction efficiency compared to its onshore counterpart (Bilgili et al. 2011). In Section 3.1 we describe the fundamental knowledge of the offshore wind installation process, which is modeled with the TPN approach illustrated in Subsection 3.2. In Subsection 3.3 we formulate the scheduling task of OWF installation as an optimization problem. The notations used in this work are summarized in Table 1.
3.1 Offshore Wind Installation Process

Fundamentally, the offshore installation is a process that consists of five different operations: 1 Loading (L), 2 Sailing Forward and Backward (SF & SB), 3 Reposition (R), 4 Jack-Up and Jack-Down (JU & JD), and 5 Construction (C). The most important characteristic of offshore scheduling is that these base operations are restricted by external conditions, i.e., the wind speed, $v$, and wave height, $h$. The duration and weather limits of each operation are given in Table 2, which are typical values found in the literature for the conventional logistic concept (Peng et al. 2020b). For example, a construction will take 14 hours and during this operation the wind speed has to be less than 10 m/s. Different from some industrial job shop problems, where the jobs are modeled as Directed Acyclic Graph (DAG) (Birgin et al. 2015), an OWF installation job is a cycle of operations illustrated in Figure 1. In this work, we only consider the conventional installation concept. Some other installation concepts exist in the literature, such as the innovative ones (Oelker et al. 2018), but the conventional one is the most commonly used in the industry due to the reliability given by the system buffer, i.e., the base port (Rippel et al. 2019). However, offshore installation jobs have different length depending on the number of components loaded onto the agent, i.e., installation vessel in the context of OWF installation, which is the core of the complete offshore operations. As shown in Figure 2, at time $t_w$ an agent can be loaded with components up to the maximal capacity of the installation vessel. In this work, we consider that the maximal capacity of an installation vessel to be four OWTs. Based on the number of components loaded, there are different job variants. If the installation vessel is loaded with $k$ OWTs at time $t_w$, then the agent can choose jobs from the set $J = \{J_1, \cdots, J_K\}$, where the subscript simply indicates the number of wind turbines to be installed in the job. For example, the job $J_2$ means the agent will construct two wind turbines.

3.2 TPN Model

The Petri nets (PN) approach and its extensions, such as Timed Petri Nets (TPN) are widely used in modeling systems, which are characterized as being concurrent, asynchronous, distributed, parallel, non-deterministic, and/or stochastic (Murata 1989; Marsan et al. 2007). Originally, the PN approach was constructed without
Figure 1: OWF installation process with conventional logistic concept.

Figure 2: Possible job variants at time $t_N$.

The inclusion of the notion of time yields the PN extension, i.e., Timed Petri Nets (TPN). In difference from the conventional PN approach, transitions of a TPN model are assigned with time units, which define the time distances between the enabling and the firing of timed transitions. Same as to conventional PN model, a TPN model is still a bipartite graph with three different elements: places, timed transitions, and arcs, which are graphically illustrated as $\bigcirc$, $\square$, $\rightarrow$ respectively.

In this work, we distinguish the graphical representation of the temporal transition from the conventional one, $\square$, by adding diagonal lines to the graphical representation, $\bigstar$. Because the firing time function, $ft$, assigned to each temporal transition is evaluated using Discrete-Time Markov Chain (DTMC) approach proposed by Rippel et al. (2019). Whenever the transition is enabled, it asks the DTMC sub-procedure to calculate the firing time based on the actual information and historical data. Thus, the prediction of the operation duration is given by the firing time function $f_t(d_{op}, v_{max}, h_{max}, v_{data}, h_{data})$, which is a function of operation duration, operation limits, current time, and historical data. Here, we use the historical data measured on the German North Sea from 1958 to 2007 with an hourly resolution. The TPN representation of the OWF installation process is shown in Figure 3 (Peng et al. 2020a).

Table 2: Base operation duration and weather limitations.

<table>
<thead>
<tr>
<th>Operation</th>
<th>$d$ (h)</th>
<th>$v_{max}$ (m/s)</th>
<th>$h_{max}$ (m)</th>
<th>unit cost (€/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>12</td>
<td>-</td>
<td>-</td>
<td>1200</td>
</tr>
<tr>
<td>SF &amp; SB</td>
<td>4</td>
<td>21</td>
<td>2.5</td>
<td>2400</td>
</tr>
<tr>
<td>JU &amp; JD</td>
<td>2</td>
<td>14</td>
<td>1.8</td>
<td>1800</td>
</tr>
<tr>
<td>C</td>
<td>14</td>
<td>10</td>
<td>-</td>
<td>1800</td>
</tr>
<tr>
<td>R</td>
<td>2</td>
<td>14</td>
<td>2</td>
<td>1800</td>
</tr>
<tr>
<td>W</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>400</td>
</tr>
</tbody>
</table>

3.3 Optimization Problem

In this subsection, we summarize the information above and formulate the scheduling problem as an optimization problem. First of all, for each installation cycle, there are $|S|$ numbers of different options,
Peng and Szczerbicka

where \( S = \{s_1, ..., s_n, ..., s_N\} \) is the set of all possible schedules that can take place in an installation cycle. Second, each installation cycle can be postponed up to \( t_{\text{max}} \) hours. This yields a discrete search space with the size of \( |S| = |J| \times t_{\text{max}} \). Thus, the optimal schedule for an installation cycle is defined as \( s_{\text{opt}} = (J_{\text{opt}}, w_{\text{opt}}) \).

To find this optimum, an objective function is needed, which is used to measure the quality of an allowed set of inputs. The goal of this optimization problem is apparently to minimize the cost resulting from the installation. Thus, the following objective function is proposed:

$$
\min f(s) = C(J_s) + C(t_w).
$$

Function \( C() \) computes the cost of each operation based on the unit cost given in Table 2 and the cost of waiting time. The first term on the right-hand side in Equation (1) represents the regular cost of the Job \( J_k \), which is obtained by summing up the cost of each operation, i.e., \( J_k = \sum_{m=1}^{M} C(O_{k,m}) \). The second term can be seen as a penalty term. Otherwise, the system will try to push all the work into the future since doing nothing leads to zeros cost and, thus, is always the cheapest solution for the current scheduling step.

4 ADAPTIVE SEARCH ALGORITHM FOR OWF SCHEDULING

In this section, we first introduce the idea of using the Apriori property to prune the search space to mitigate the problem of overlooking the optimum. As an improvement to the strategy, we propose an adaptive threshold based on the maximum entropy principle in Subsection 4.2 to further reduce the cost resulting from the installation.

4.1 Search Space Pruning Using Apriori algorithm

The Apriori property has been widely used in solving frequent item set mining (FIM) problems. For a more detailed understanding of the subject, literature such as (Tan et al. 2019) is recommended. The Apriori property reveals the fact that if an itemset, \( X \), is frequent then all its subsets, \( Y \), are frequent against a threshold \( \tau \), i.e., if \( X \) is frequent then \( Y \subset X \) is frequent. Thus, it yields the inverse property, which infers if \( Y \) is not frequent, then all its supersets are not frequent. Indeed, the inverse property is the one that helps us to reduce the candidate itemsets set by pruning non-promising itemsets in the FIM task.

In this work, we use this concept in an analogical sense to prune the search space of a scheduling task. In Subsection 3.1 we have already introduced four different offshore installation jobs. Let \( k \) be the level of potential jobs, which also reveals their complexity, i.e., the higher the level the more complex the job and its duration will be; \( s_{\text{start}} \) be the starting time of a job. Besides, we assume that a job at level \( k \), \( I_k \), is included in a schedule at level \( k + 1 \), i.e., \( J_k \subset J_{k+1} \). In this case, the Apriori property implies that if it is good to start job \( J_{k+1} \) at time \( t_{\text{start}} \), then it is also good to start job \( J_k \) at time \( t_{\text{start}} \). Thus, the inverse property allows to prune the search space, since if a job \( J_k \) starting at \( t_{\text{start}} \) is bad, then all its supersets are potentially bad as well.

It is important to measure the quality of a job, \( J_k \), i.e., to figure out whether it is meaningful to investigate a job at a higher level, \( J_{k+1} \). For this purpose a predefined threshold, \( \tau \), is required. In the FIM task, the threshold, \( \tau \), is defined as a frequency or probability of occurrence. In scheduling tasks, we assume first \( \tau \) as a deviation in percentage to the optimal cost of a job. For example, the optimal cost of a job, \( J_k \), is \( C_{\text{opt}}(J_k) \). Then this job is considered to be good, if \( \frac{C_{\text{opt}}(J_k)}{C_{\text{real}}(J_k)} = \tilde{C} \geq \tau \).

The optimal cost, \( C_{\text{opt}}(J_k) \), is obtained by performing \( J_k \) under ideal conditions, i.e., every operation has been performed on time and no delay has occurred. The real cost, \( C_{\text{real}}(J_k) \), is determined by performing \( J_k \) under the current situation at time step \( t_w \), where delays caused by weather conditions and insufficient resources are taken into account. Apparently, the threshold \( \tau \) takes a value from 0 to 1, where 0 represents the worst case, where the operation never finishes, and 1 represents the best case, i.e., the reality is ideal.

This pruning strategy is done in two steps at each level. In the first step, the candidates at level \( L_k \) are investigated. The results of each candidate are examined with the predefined threshold, and those bad
solutions are filtered out. The remained candidates are delivered to level \( L_{k+1} \) for further investigation. Figure 4a gives an example of this two-step pruning strategy at level \( L_k \). Figure 4b shows an example of a pruned search space. The search space can be effectively reduced by using this strategy. However, there is a problem with the fixed threshold, i.e., the overlooking of the real optimum. In another word, we cannot always find the best schedule and there is no guarantee of the quality of the solution found. A numerical example is made to clarify this issue in Section 5. To overcome this problem we introduce an adaptive threshold.

### 4.2 Adaptive Threshold

It is hard to find a fixed threshold that is suitable to prune the search space as the complexity of the jobs and the external conditions are changing. To mitigate this problem, we introduce an adaptive threshold based on the concept of information entropy. As presented in Section 2 the concept of adaptivity has been used across many fields of study. Here, the notion of adaptivity refers to the ability of an algorithm, that it can adjust its behavior during the simulation based on information available and/or the Apriori-defined reward mechanism.

**Information Entropy**  
The entropy of a probability distribution can be interpreted as a measure of uncertainty, or lack of predictability, associated with a random variable drawn from a given distribution. For example, suppose we observe a sequence of symbols \( X_n \sim \mathcal{P} \) generated from distribution \( \mathcal{P} \). If \( \mathcal{P} \) has high entropy, it will be hard to predict the value of each observation \( X_i \). Hence we say that the dataset \( \mathcal{D} = \{X_1, \ldots, X_i, \ldots, X_I\} \) has high information content. By contrast, if \( p \) is a degenerate distribution with 0 entropy (the minimal value), then every \( X_i \) will be the same, so \( \mathcal{D} \) does not contain much information (Murphy 2022).

Let \( C_{k,w} = C(J_{k,t_w}) \) be the cost of the schedule on time \( t_w \) and at level \( L_k \) with \( w \in [0, t_{\text{max}}] \) and \( k \in [1, K] \). The probability of the cost of \( i \)-th schedule is given as \( p_i = \frac{N_i}{N} \), where \( N_i \) is the number of occurrences of \( C_{k,w} \) in the population, and \( N \) is the number of data points in total. The adaptive threshold here means that the \( q \)-th cost value is set as the threshold for the level \( L_k \). This means that the 1 \( \sim \) \( q-1 \)-th schedules will be eliminated and will not be considered in the level \( L_{k+1} \). In this way, we obtain the two following probability distributions:

\[
A : \frac{P_1}{P_A}, \frac{P_2}{P_A}, \ldots, \frac{P_{q-1}}{P_A} \quad \text{and} \quad B : \frac{P_q}{P_B}, \frac{P_{q+1}}{P_B}, \ldots, \frac{P_N}{P_B},
\]
where \( P_A = \sum_{i=1}^{q-1} p_i \) and \( P_B = 1 - P_A = \sum_{i=q}^{N} p_i \). The information entropy is given as follows (Mohri et al. 2018):

\[
\mathcal{H}_n = -\sum_{i=1}^{N} p_i \cdot \ln (p_i).
\]

Thus, the information entropy of the two groups \( A \) and \( B \) can be calculated as follows respectively:

\[
\mathcal{H}(A) = -\sum_{i=1}^{q-1} \frac{p_i}{P_A} \cdot \ln \left( \frac{p_i}{P_A} \right) = \ln (P_A) + \frac{\mathcal{H}_{q-1}}{P_A}
\]

\[
\mathcal{H}(B) = -\sum_{i=q}^{N} \frac{p_i}{P_B} \cdot \ln \left( \frac{p_i}{P_B} \right) = \ln (1 - P_A) + \frac{\mathcal{H}_N - \mathcal{H}_{q-1}}{1 - P_A}
\]

The total information entropy is simply the combination of the two parts:

\[
h(s) = \mathcal{H}(A) + \mathcal{H}(B)
\]

\[
= \ln (P_A) + \ln (1 - P_A) + \frac{\mathcal{H}_{q-1}}{P_A} + \frac{\mathcal{H}_N - \mathcal{H}_{q-1}}{1 - P_A}
\]

Our interest would be looking for the index \( q \) which maximizes the entropy according to the maximum entropy principle: \( q = \arg\max (h(q)) \). Thus, the cost value \( C_q \) is selected as the threshold for the current level \( L_k \), i.e., the \( C_q \) is different for each level based on the information obtained from the current level. In this way, the threshold for pruning is selected adaptively according to the knowledge obtained in the actual situation.

5 NUMERICAL EXPERIMENTS

Here, we consider the desired OWF with a size of 40 Offshore Wind Turbines (OWTs) located in the German North Sea. In this numerical study, we consider the application of two installation vessels. Theoretically, much more vessels can be dispatched to one OWF installation project, however, it has been shown that the application of more than two installation vessels in one installation project is not financially advisable (Peng et al. 2021b). Besides, we consider that an agent (i.e., an installation vessel) can be loaded with components for at most four OWTs. Furthermore, the base port has storage of components for 12 OWTs at the beginning of the simulation. As the system buffer, the base port can be stored with components for maximal of 32 OWTs. The bottom line of the base port storage is set as 8 OWTs, which means as long as the storage reaches this limit it will request for supply. The parameter setting of the TPN model for this scenario is summarized in Table 2.

5.1 Search Space Pruning

via Apriori algorithm In conclusion, the computational results show that it requires at least 6 installation cycles for each agent to finish the installation of 40 OWTs. For the convenience of graphical illustration, we take the results of the first installation cycle as an example. Figure 5 shows the search space of the first installation cycle with different pruning thresholds. Each dot in the figures represent a schedule run under the corresponding combination of job level and the postpone. It is clear that with the increasing value of the fixed threshold, \( \tau \), the search space is effectively pruned. For example, with \( \tau = 0.95 \) around 65% of the search space of agent 1 has been pruned in the first installation cycle. However, we also notice that with a higher value of \( \tau \) it is possible that the optimal solution, marked with a red triangle, is overlooked.

via adaptive threshold The last subfigures in Figure 5a and 5b show the pruned search space of agent 1 and agent 2 in the first installation cycle using adaptive threshold, \( \tau \), respectively. As we can see that the search space is effectively pruned, meanwhile it mitigates the problem of overlooking the optimal solutions. As we can observe that the optimal solution for agent 1 is found, however, the algorithm finds a slightly more expensive solution for agent 2.
5.2 Summary and Discussion

In this subsection, we mainly discuss 1) the findings of using different values of fixed threshold $\tau$ and the difference between the fixed one and the adaptive one $\tilde{\tau}$; 2) the problem of overlooking the optimum in the search space. For the discussion about the financial benefits of using the scheduling approach and the comparison between different approaches, we refer to Peng et al. (2021a) and Peng et al. (2021b).

Comparison between fixed $\tau$ and adaptive threshold $\tilde{\tau}$

As we can see in Figure 5, the higher-level jobs are significantly pruned as the value of the fixed threshold $\tau$ increases, i.e., the scheduler prefers lower-level jobs, where the dark blue dots and the orange dots represent one schedule in the search space. In another word, this results in schedules with more installation cycles. For example, the installation of the desired OWF can be finished within 6 installation cycles for each agent as we set $\tau = 0.35$, which does not prune the search space at all in this case; however it will take 10 installation cycles for each agent if we increase the fixed threshold $\tau$ to 0.95. This simply reveals that the higher the value of the fixed threshold $\tau$ the swifter the schedules, thus, more installation cycles will be needed.

Compared to fixed thresholds, the computational results show that the usage of an adaptive threshold, $\tilde{\tau}$, based on information entropy is meaningful. It adjusts itself according to the situation at the current step and effectively prunes the search space. The problem of overlooking the optimal solution still exists, for example, in Figure 5b agent 2 has missed the optimal solution marked as a red triangle while using an adaptive threshold. However, the algorithm is still able to find a solution that is close to the optimal solution. Figure 6 illustrates the computational costs and the sizes of search space using different thresholds relative to the case $\tau = 0.35$. As reference the computational cost of case $\tau = 0.35$ is about 7 min and the size of the search space is 4656 schedules. For example, the case where $\tau = 0.5$ has a computational cost and a search space about 80% of the reference value, i.e. 5.6min and 3725 schedules. It is easy to notice that the computational cost and search space size are positively correlated. In conclusion, we believe that the usage of the adaptive threshold is a reasonable tradeoff between computational time and the quality of results.

Overlooking of the optimum

In previous work, the problem of overlooking the optimum has been addressed while using the Simulated Annealing (SA) approach (Peng et al. 2021a). However, the reasons for the occurrence of this phenomenon are different. In the SA approach, the search space is investigated in a stochastic way, since there is always a probability of rejection for the neighborhood of a data point. Thus, it is possible, that the solution found via SA is not the optimum but somewhere near the optimum. This phenomenon also cannot be improved by changing the parameters, such as temperature, and the number of loops as long as the probability of rejection exists. To investigate the overlooking rate, we have simulated about 200 installation cycles starting from random dates for different threshold values. Figure 7 shows that the adaptive threshold, $\tilde{\tau}$, has captured 86% of the optima. There is a about 40% loss of optima with $\tau = 0.65$. The overlooking of the optimum observed in the proposed strategy is given by the dependency between the complexity levels. When the candidates are already eliminated in the lower levels, they will not be delivered as candidates to the higher levels. In the authors’ opinion of view, this could be solved by decoupling the complexity levels and by using back-propagation. Even though the proposed strategy provides satisfying results.

6 CONCLUSION

In this work, we have addressed one of the major problems in the offshore wind problem installation process, which is the delay resulting from the changeable offshore weather conditions. We have proposed an adaptive search space pruning strategy based on the Apriori algorithm and information entropy to improve the computational efficiency of the OWF scheduling task. The numerical results show that the proposed strategy makes a reasonable tradeoff between computational time and the quality of the solution. However, they also show that the phenomenon of overlooking the optimal solution is in this case unavoidable since there are still dependent effects on the pruning strategy. The reason for the occurrence of this phenomenon is discussed shortly compared to the SA approach. The opinion of the authors is that the problem in the
Figure 5: Search space of the first installation cycle constructing an OWF = 40 OWTs using two agents with fixed and adaptive pruning thresholds. Legends: a) Schedules calculated using $\tau$: (●); b) Schedules calculated using $\tilde{\tau}$: (●); c) Optimal Schedule: (★).

Figure 6: Computational time and search space using different thresholds relational to $\tau = 0.35$.

Figure 7: The rate on overlooking the optimal solution of different threshold values.

The proposed strategy is solvable by decoupling the complexity and back-propagation, which can be considered topics for future works. Another aspect that is worth further investigation is other adaptive strategies for pruning the search space.
ACKNOWLEDGMENTS

The authors are grateful for the financial support by the DFG (German Research Foundation) for the Project “OffshorePlan”, grant number LU 2049/1-1; SZ 51/33-1. We would also like to thank our colleagues, Mr. Daniel Rippel and Dr. Michael Lütjen, from BIBA for the valuable supports.

REFERENCES


