

University of Groningen

Exact and heuristic methods for optimization in distributed logistics

Schrotenboer, Albert

DOI:
[10.33612/diss.112911958](https://doi.org/10.33612/diss.112911958)

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2020

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Schrotenboer, A. (2020). *Exact and heuristic methods for optimization in distributed logistics*. [Thesis fully internal (DIV), University of Groningen]. University of Groningen, SOM research school.
<https://doi.org/10.33612/diss.112911958>

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Chapter 3

Coordinating technician allocation and maintenance routing for offshore wind farms

Abstract. *A maintenance activity at offshore wind farms requires a combination of technicians with different skills. At an operational level, it is important to fully utilize and coordinate technicians in order to increase efficiency of the short-term maintenance planning. In this paper, we investigate sharing of technicians between wind farms over multiple periods, while determining per period vessel routes delivering and picking up technicians. The problem can be considered as a novel variant of the multi-period multi-commodity pick up and delivery problem. We develop an adaptive large neighborhood search heuristic which achieves high-quality, and often optimal, solutions on benchmark instances from the literature. The heuristic is used to explore the benefits of different sharing policies. By sharing technicians, both the flexibility of the daily planning is improved and the expected maintenance costs are reduced. In addition, the increased flexibility results in fewer vessel trips and increases the decision maker's ability to cope with extreme scenarios encountered in the short-term maintenance planning.*

This chapter is based on Schrottenboer et al. (2018a):
Schrottenboer AH, Uit het Broek MA, Jargalsaikhan B, Roodbergen KJ, 2018a *Coordinating technician allocation and maintenance routing for offshore wind farms. Computers & Operations Research* 98:185–197

3.1 Introduction

Resource sharing has the potential to improve the efficiency of the overall supply chain. To incorporate resource sharing into a firm's daily practice, supply chain integration is required on both a strategic and a tactical level of supply chain management. On an operational level, supply chain integration is likely to reduce the waste of valuable resources and to increase profitability of the business (Leuschner, Rogers, and Charvet 2013). Examples of such practice include inventory pooling (Berman, Krass, and Mahdi Tajbakhsh 2011, Zhong et al. 2017) and flexibly allocating workforce to multiple sites (Bhandari, Scheller-Wolf, and Harchol-Balter 2008). In this study, we analyse different policies to allocate the workforce in the context of maintenance routing and scheduling at offshore wind farms, and their consequences on an operational level. We, thereby, generalize existing routing and scheduling problems in this area (e.g., Dai, Stålhane, and Utne (2015), Stålhane, Hvattum, and Skaar (2015), Irawan et al. (2017)), and show that a flexible workforce policy is a natural way to improve the efficiency of the supply chain.

The total number of offshore wind farms has increased significantly in the last years, and will increase even more in the coming decades¹. This results in a situation in which maintenance providers are becoming responsible for maintaining multiple wind farms. It is clear that operating each wind farm in isolation may result in an incomplete utilization of the available resources, which is the main challenge of the short-term maintenance planning at offshore wind farms. A typical feature of this problem is the scarcity of technicians with different specialties that restricts the flexibility, and thereby the efficiency, of the short-term maintenance planning. A maintenance activity requires, besides required spare parts, a combination of technicians with different skills. Therefore, a coordinated approach in which the available technicians are jointly utilized between wind farms may increase the overall efficiency of the planning, for example, by reducing idle time of technicians (Shafiee 2015, Irawan et al. 2017). Such insights on an operational level offer opportunities to increase sustainability of the offshore wind sector, and increases competitiveness with traditional energy suppliers.

Usually, technicians are assigned on a long-term basis to one of the Operating and Maintenance (O&M) bases, from which the daily operations are coordinated. From this O&M base, the technicians are deployed to perform maintenance activities at one or multiple wind farms. In order to improve efficiency, the maintenance provider may reassign technicians between O&M bases in the short-term. For example, technicians may be assigned completely flexible on a daily basis or for multiple days to a single O&M

¹European Wind Energy Association. *Wind energy scenarios for 2030*, EWEA, 2015.

base depending on the geographical restrictions. Once the allocation of technicians is decided upon, the construction of a short-term maintenance planning itself presents several challenges. Generally, the time frame of a short-term planning is a week as the maintenance activities are heavily dependent on the weather conditions. Each maintenance activity requires an activity-specific set of technicians. Thus, technicians with a suitable skill set for an activity are chosen and transported to an offshore location to perform maintenance activities. After completion of the maintenance, the set of technicians may change to start other maintenance activities, i.e., technicians do not work in fixed teams throughout the period. Summarizing, a set of vessel routes needs to be determined for each period that describes which technicians are transferred to which offshore location, such that all maintenance activities are scheduled throughout the time horizon.

The allocation of technicians to different O&M bases classifies as tactical decision making. Embedding vehicle routing problems into optimization problems of a tactical or a strategic nature are practically relevant but challenging. Inherently, such problems are often intractable to solve exactly and, therefore, metaheuristics are often employed to provide insights on the problem. Ciancio et al. (2018) developed a heuristic solution approach for scheduling bus lines, and afterwards assigning crew to these lines. Technician allocation touches the field of workforce management, of which its influence in routing problems is studied by Smilowitz, Nowak, and Jiang (2013) as well. They study the efficiency of vehicle routing solutions in the presence of workforce management constraints, and show that properly including workforce management conditions into the optimization slightly increases overall costs. Many rich routing problems are found in health or home care applications. For example, Detti, Papalini, and de Lara (2017) have developed a hybrid heuristic combining variable neighborhood search and tabu search for a healthcare routing problem with many side constraints.

In this paper, we introduce the Technician Allocation and Routing Problem for offshore wind farms (TARP). Its goal is to jointly determine the daily allocation of differently skilled technicians to multiple O&M bases and the accompanying daily vessel routes used to perform the maintenance activities. Daily vessel routes should depart and return to their corresponding O&M bases every day, assuring the pickup and delivery of technicians and the transport of spare parts, to the offshore locations. By incorporating the technician allocation, the TARP is a generalization of the short-term maintenance planning and routing problem at offshore wind farms, see Dai, Stålhane, and Utne (2015), Stålhane, Hvattum, and Skaar (2015), Irawan et al. (2017). In the TARP, technicians are *flexibly* allocated on a daily basis; to analyze different practices at offshore wind farms, we introduce two variants of the TARP. The first

variant, called TARP-F, allocates technicians at the beginning of the time horizon and the allocation is *fixed* over time. The second variant, called TARP-G, assumes that the allocation of technicians to O&M bases is *given*, i.e., technicians are not shared between O&M bases. The TARP-G serves as a benchmark to obtain insights on the impact of technician sharing.

We develop a Two-Stage Adaptive Large Neighborhood Search (2-ALNS) heuristic to solve the TARP and its two variants. Our 2-ALNS achieves high-quality (and often optimal) solutions on a set of benchmark instances available in the literature for the TARP-G. Since we are the first to consider large-scale instances for the TARP and its two variants, a set of new benchmark instances is provided. Via simulation, we provide insights on the impact of technician sharing under the different technician allocation policies. To be specific, we varied parameters of the geographical layout of the wind farms, O&M bases, and the maintenance characteristics by performing a Monte-Carlo simulation which embeds the 2-ALNS for the estimation of the expected maintenance and routing costs. It is shown that reassigning technicians increases the robustness of the short-term planning, i.e., extreme scenarios are handled more efficiently. On top of reducing expected maintenance and routing costs, sharing technicians leads to fewer vessel routes on average.

In the remainder of this section, we discuss the literature related to the TARP. Note that TARP-G and TARP-F are special cases of the TARP and the statements regarding the TARP hold for its two variants as well, unless explicitly mentioned. We first consider the general pickup and delivery literature as the TARP can be modeled as a novel variant of a Pickup and Delivery Problem (PDP). Second, we discuss articles that study related problems on maintenance routing and scheduling at offshore wind farms. Finally, we review miscellaneous results related to our heuristic approach and computational analysis.

The TARP can be classified as a new variant of the multi-depot multi-period pickup and delivery problem with multiple commodities. Pickup and delivery problems (Dumas, Desrosiers, and Soumis 1991, Savelsbergh and Sol 1995) consider the fulfillment of transportation requests by developing cost-minimizing vehicle routes. Following the classifications by Berbeglia et al. (2007) and Parragh, Doerner, and Hartl (2008), the TARP combines a one-to-one with a many-to-many pickup and delivery structure. The former structure emerges between nodes representing the start and completion of a maintenance activity; a vessel delivering technicians to a maintenance location needs to pick up the technicians after completion. The latter structure, many-to-many pickup and delivery, is present between nodes representing different maintenance activities; after completion of a maintenance activity, technicians with different skills become

available to start the next activity.

Both pickup-and-delivery structures have been extensively studied. Sophisticated exact solution approaches have been proposed for the one-to-one PDP with time windows, e.g., see Ropke, Cordeau, and Laporte (2007), Ropke and Cordeau (2009), Baldacci, Bartolini, and Mingozzi (2011). An Adaptive Large Neighborhood Search (ALNS) is introduced by Ropke and Pisinger (2006), involving larger instances up to 500 nodes. Given the characteristics of our problem, ALNS is well suited for TARP as well. The many-to-many PDP is studied in several variants. We refer to Hernández-Pérez and Salazar-González (2004) and Hernández-Pérez, Rodríguez-Martín, and Salazar-González (2009) for the single vehicle case, and Hernández-Pérez and Salazar-González (2014) and Hernández-Pérez, Rodríguez-Martín, and Salazar-González (2016) for the multi-commodity case. One of the main areas of application is the so-called bike repositioning problem; a many-to-many pickup and delivery problem where one needs to find cost-minimizing routes for multiple vehicles that re-balance the inventory of bike rental systems (Dell et al. 2016, Bulhões et al. 2018).

Besides the transportation of technicians, spare parts need to be moved as well. This aspect shares characteristics with the fixed multi-compartment vehicle routing problem, in which multiple product types need to be stored separately in capacitated vehicles (see, e.g., Alinaghian and Shokouhi 2018). Instead of a regular vehicle routing structure, as encountered in multi-compartment vehicle routing problems, the TARP has a pickup and delivery structure. To the best of the authors' knowledge, such a structure is only examined in maintenance and routing problems at offshore wind farms.

Maintenance routing and scheduling at offshore wind farms, with a given set of technicians to a single O&M base, is first introduced by Dai, Stålhane, and Utne (2015): A mixed integer programming model is presented and solved with standard commercial solvers. An exact decomposition method for a single period presented by Stålhane, Hvattum, and Skaar (2015). Recently, an extension to multiple wind farms and multiple depots (TARP-G) is discussed by Irawan et al. (2017). Optimal solutions to instances of 24 maintenance activities were found by means of a set partition formulation. Their approach is not applicable for the TARP since the inclusion of technician allocation in the optimization problem increases the solution space exponentially. We, therefore, focus on a heuristic approach capable of finding high-quality solutions for large instances of the TARP.

We embed the 2-ALNS in a Monte-Carlo simulation for an in-depth analysis of the impact of technician sharing for different underlying networks. Such a simulation to obtain insights on complex routing problems are common yet crucial for the analysis.

For example, an analysis of periodic routing problems is given by Palhazi Cuervo, Vanovermeire, and Sørensen (2016). By a full-grid Monte-Carlo simulation on parameters describing the transportation network, they provided insights when it is beneficial for individual logistics carriers to cooperate. A similar simulation approach is taken by Post et al. (2018), incorporating routing heuristics to provide an in-depth analysis of an LNG transportation network.

The remainder of this paper is structured as follows. The complete problem description is given in Section 3.2. In Section 3.3, we introduce the 2-ALNS proposed to solve the TARP and its two variants. In Section 3.4, we show the performance of the 2-ALNS by solving benchmark instances from the literature, and secondly, we solve a new benchmark set consisting of computationally challenging instances. Managerial insights, including the circumstances in which technician sharing is beneficial, are provided in Section 3.5. We conclude and provide suggestions for future research in Section 6.6.

3.2 Problem description

In this section, we give a mathematical description of the Technician Allocation and Routing Problem (TARP). In the TARP, one needs to jointly determine the allocation of technicians to depots (i.e., O&M bases) and the accompanying short-term maintenance planning. After the mathematical description of the TARP, we present additional constraints that lead to the two variants TARP-F and TARP-G (with a fixed sharing policy and without sharing, respectively). At the end of the section, we restructure the travel costs and times to obtain a more efficient and simpler problem structure. This is especially useful for the computational performance of the 2-ALNS. A summary of the used notation can be found in Table 3.1.

3.2.1 The Technician Allocation and Routing Problem (TARP)

The TARP is defined on a graph $G = (\mathcal{N}, \mathcal{A})$, where the node set $\mathcal{N} = \mathcal{N}_d \cup \mathcal{N}_p \cup \mathcal{N}_0$ consists of delivery nodes $\mathcal{N}_d = \{1, \dots, n\}$, pickup nodes $\mathcal{N}_p = \{n+1, \dots, 2n\}$, and depots $\mathcal{N}_0 = \{2n+1, \dots, 2n+B\}$. Every delivery node $i \in \mathcal{N}_d$ has a corresponding pickup node $i+n \in \mathcal{N}_p$, i.e., every delivery and pickup pair represents a single geographical location as they indicate the start and end of a single maintenance activity. Every node $i \in \mathcal{N}_d$ and $(i+n \in \mathcal{N}_p)$ belongs to a wind farm $w_i \in \mathcal{W}$, where $\mathcal{W} = \{1, \dots, W\}$ is the set of wind farms. Let $\mathcal{T} = \{1, \dots, T\}$ be the finite time horizon in which the nodes may be visited and let $\mathcal{V}_b = \{1, \dots, V_b\}$ be the set of

Table 3.1: Summary of sets, parameters and decision variables for the TARP.

Sets	
$\mathcal{N}_d, \mathcal{N}_p, \mathcal{N}_0$	Set of delivery nodes (\mathcal{N}_d), pickup nodes (\mathcal{N}_p) and depots (\mathcal{N}_0).
\mathcal{W}	Set of wind W farms.
\mathcal{T}	Set of T periods.
\mathcal{L}	Set of L technician types.
\mathcal{V}	Set of all vessels.
\mathcal{V}_b	Set of vessels stationed at depot b .
\mathcal{R}_{vt}	Set of all feasible routes of vessel v in period t .
Parameters	
\widehat{q}_ℓ	Number of available technicians of type ℓ .
q_i^ℓ	Demand of technician type ℓ at node i .
s_i	The service time at node i .
d_i	Due date of maintenance activity i .
p_i	Penalty for exceeding due date of activity i
e_i	Equals 1 if a vessel needs to wait during maintenance activity i .
t_{ij}^v, c_{ij}^v	Travel time and travel costs between nodes i and j by vessel v .
$\widetilde{t}_{ij}^v, \widetilde{c}_{ij}^v$	Modified travel time and travel costs.
\widehat{c}_ℓ	Costs of using a technician of type ℓ .
$\overline{q}_v^1, \overline{q}_v^2$	The spare parts capacity (\overline{q}_v^1) and technician capacity (\overline{q}_v^2) of vessel v .
ω_{vt}	Maximum duration of vessel v in period t .
τ	Transfer time from a vessel to a node, and vice versa.
c_r	Costs of route r .
θ_r^i	Equals 1 if route r visits node i , 0 otherwise.
γ_r^ℓ	The number of technicians of type ℓ used in route r .
Decision variables	
$x_{bt\ell}$	Number of technicians of type ℓ at depot b in period t .
y_r	Equals 1 if route r is selected in a solution, and 0 otherwise.

heterogeneous vessels at depot $b \in \mathcal{N}_0$, so that the complete set of vessels is given by $\mathcal{V} = \cup_{b \in \mathcal{N}_0} \mathcal{V}_b$. Traversing arc $(i, j) \in \mathcal{A}$ with vessel $v \in \mathcal{V}$ takes t_{ij}^v time.

We consider a set of technicians' types $\mathcal{L} = \{1, \dots, L\}$. The demand for technician type $\ell \in \mathcal{L}$ at node i equals q_i^ℓ so that $q_i^\ell = -q_{i+n}^\ell$ for all $i \in \mathcal{N}_d$. Each maintenance activity has a service time $s_i \geq 0, i \in \mathcal{N}_d$, i.e., if node $i \in \mathcal{N}_d$ is visited at time $a_i \geq 0$, node $i + n$ cannot be left before time $a_i + s_i$. Every job is scheduled such that it is completed in a single period, and the vessel that delivers technicians to node i is responsible for picking them up at node $i + n$. In addition, a maintenance activity may require a vessel to wait during the performance of the maintenance activity. We denote this with the parameter e_i , equaling 1 if a vessel needs to wait, and 0 otherwise. For notational convenience, we let $e_i = 0$ for all $i \in \mathcal{N}_p$.

The total number of available technicians of type ℓ equals \hat{q}_ℓ . For every period $t \in \mathcal{T}$, one needs to determine $x_{bt\ell}$, the number of technicians of type ℓ designed to depot b in period t . Hence for every period t , $x_{bt\ell}$ technicians of type ℓ must be available at the start and end of period t . In addition, each vessel v has a limited capacity \bar{q}_v^1 to transport spare parts, a limited capacity \bar{q}_v^2 to transport technicians, and transferring technicians from a vessel to any node i takes τ time. The maximum time a vessel v can be utilized in period t equals ω_{vt} . This models restrictions in working hours due to weather conditions.

Each maintenance activity i has a due date $d_i \in \mathcal{T}$. If a maintenance activity is scheduled in some period $t' > d_i$, a penalty p_i per period exceedance is incurred. Using technicians to perform maintenance activities comes at a per period cost of \hat{c}_ℓ . Utilizing edge $(i, j) \in A$ with vessel $v \in \mathcal{V}$ comes at travel cost c_{ij}^v .

A solution to the TARP consists of an allocation $x_{bt\ell}$ of technicians and accompanying vessel routes, for every vessel $v \in \mathcal{V}_b$, depot $b \in \mathcal{N}_0$, period $t \in \mathcal{T}$, and all \hat{q}_ℓ technicians of type $\ell \in \mathcal{L}$. Let \mathcal{R}_{vt} denote the complete set of feasible vessel routes of vessel v in period t , satisfying the constraints described in the previous paragraphs. Let θ_i^r be equal to 1 if route $r \in \mathcal{R}_{vt}$ visits node $i \in \mathcal{N}_d$ and 0 otherwise. Similarly, let γ_ℓ^r equal the number of technicians of type ℓ used in route r . The costs of a route are given by c^r , consisting of travel costs c_{ij}^v for every edge (i, j) used in route r , and of technician costs \hat{c}_ℓ for all technicians of type ℓ deployed in r . Besides the continuous decision variable $x_{bt\ell}$, we let y_r be a binary decision variable equaling 1 if route $r \in \mathcal{R}_{vt}$ is part of the solution, and 0 otherwise.

The following set partitioning formulation models the TARP:

$$\min \sum_{t \in \mathcal{T}} \sum_{b \in \mathcal{N}_0} \sum_{v \in \mathcal{V}_b} \sum_{r \in \mathcal{R}_{vt}} y_r c^r \quad (3.1)$$

$$\text{s.t.} \quad \sum_{r \in \mathcal{R}_{vt}} y_r \leq 1 \quad \forall v \in \mathcal{V}_b, b \in \mathcal{N}_0, t \in \mathcal{T} \quad (3.2)$$

$$\sum_{b \in \mathcal{N}_0} \sum_{v \in \mathcal{V}_b} \sum_{t \in \mathcal{T}} \sum_{r \in \mathcal{R}_{vt}} y_r \theta_i^r = 1 \quad \forall i \in \mathcal{N}_d \quad (3.3)$$

$$\sum_{v \in \mathcal{V}_b} \sum_{r \in \mathcal{R}_{vt}} y_r \gamma_\ell^r \leq x_{bt\ell} \quad \forall \ell \in \mathcal{L}, b \in \mathcal{N}_0, t \in \mathcal{T} \quad (3.4)$$

$$\sum_{b \in \mathcal{N}_0} x_{bt\ell} \leq \hat{q}_\ell \quad \forall \ell \in \mathcal{L}, t \in \mathcal{T} \quad (3.5)$$

$$y_r \in \{0, 1\}, \quad x_{bt\ell} \geq 0 \quad \forall r \in \mathcal{R}_{vt}, v \in \mathcal{V}_b, b \in \mathcal{N}_0, t \in \mathcal{T} \quad (3.6)$$

The Objective (3.1) minimizes the costs of the chosen routes. Constraint (3.2) models

that each vessel can only perform a single trip per period, and Constraint (3.3) ensures that all nodes are visited exactly once throughout the time horizon. Constraint (3.4) limits the use of technicians of the selected vessel routes to the number of technicians being allocated, and Constraint (3.5) ensures that technician allocation obeys the availability of technicians.

3.2.2 The Technician Allocation and Routing Problem with Fixed Allocations (TARP-F)

In this variant, we study a scenario in which wind farms and depots are located such that changing depots in every period is not practical for the technicians. In the TARP-F, we adhere a fixed allocation policy: The technicians are assigned to depots and stay at their depots over the time horizon $[1, \dots, T]$. This variant arises when the wind farms and depots are located such that period changing of depots is not practical for the technicians. A set partitioning formulation for the TARP-F is obtained by extending formulation (3.1) - (3.6) with

$$x_{bt\ell} - x_{b\tilde{t}\ell} = 0 \quad \forall t, \tilde{t} \in \mathcal{T}, b \in \mathcal{N}_0, \ell \in \mathcal{L}. \quad (3.7)$$

3.2.3 The Technician Allocation and Routing Problem with Given Allocations (TARP-G)

To investigate the effects of efficiently coordinating technicians between the depots and wind farms, we consider the Technician Allocation and Routing Problem with Given Allocations (TARP-G). It is derived from the TARP by taking $x_{bt\ell}$ as given, i.e., it is not longer a decision variable but a parameter. The TARP-G is referred to as the Maintenance Routing and Scheduling Problem at offshore wind farms, and an exact solution approach based on a set partitioning formulation has been proposed by Irawan et al. (2017). However, an explicit enumeration of the route sets \mathcal{R}_{vt} is already complex and time consuming for a given technician allocation, and becomes intractable when the technician allocation is included as a decision.

3.2.4 Preprocessing: cost and travel restructuring

Rich routing problems are often characterized with fluctuating travel costs or travel times due to a variety of side constraints. By letting travel costs be period dependent, i.e., the costs of traversing arc (i, j) with vessel v in period t equals \tilde{c}_{ij}^{vt} , we can incorporate the due date penalties p_i into the cost function. The modified cost of

traversing edge (i, j) with vessel v in period t are defined as $\tilde{c}_{ij}^{vt} = c_{ij}^v + \max\{t - d_i, 0\} \cdot p_i$.

Recall that t_{ij}^v is the travel time of vessel v between nodes i and j . Both the transfer time and any required waiting time at location i during maintenance (if $e_i = 1$) are incorporated into the travel time function t_{ij}^v . Let $\tilde{t}_{ij}^v = t_{ij}^v + \tau + e_j s_j$, be the modified travel time function for all $i, j \in N$. To be precise, we add the transfer times to every incoming edge, and the service times to the incoming edges of the delivery nodes if needed. Then, for delivery nodes i with $e_i = 1$, the corresponding pickup can be performed without additional costs after the delivery, since they resemble the same location.

At first, it may seem that the restructuring invalidates well-studied characteristics of routing problems such as the triangle inequality. However, it does not introduce additional complexities for the TARP and its two variants, as the triangle inequality is already violated by the costs for using technicians. To be specific, as technicians can be reused from the pickup nodes at no additional expenses, it may be cheaper to visit a delivery location via a pickup location, instead of directly traveling to the delivery location.

3.3 Adaptive Large Neighborhood Search

We develop a Two-Stage Adaptive Large Neighborhood Search (2-ALNS) to solve the TARP and its two variants. We first give a brief overview of the 2-ALNS procedure. Afterwards, we provide the details of the 2-ALNS and the operators used therein. We conclude this section with some minor differences between solving the TARP-G instances on the one hand, and the TARP-F and TARP instances on the other hand.

Adaptive Large Neighborhood Search (ALNS), as first introduced by Ropke and Pisinger (2006), is a metaheuristic often used to solve complex routing problems, see, for example, Mancini (2016) and Ghilas, Demir, and Woensel (2016). ALNS is an iterative procedure relying on a set of destroy operators, removing parts of a solution, and a set of repair operators, inserting the removed parts into the solution. The successive appliance of destroy operators and a repair operator is called a *move*. A move consists of independently selected destroy and repair operators according to a continuously updated probability distribution. If a move appeared to be successful, the probability of selecting the corresponding operators will slightly increase. Thereby, effective operators are more likely to be chosen in future iterations of the ALNS. As often encountered in current metaheuristics, we embed the iterative destroying and repairing of the solution in a simulated annealing environment. A move is accepted with a probability that depends on the number of iterations and the difference in

Algorithm 3.1: ALNS Procedure

```

 $s_{\text{cur}}, s_{\text{best}}, s_{\text{glob}} \leftarrow \text{initialSolution}();$ 
 $n_1 \leftarrow 0$  and  $n_2 \leftarrow 0;$ 
while  $n_1 < N_{\text{iter}}^1$  do
   $s_{\text{cur}}, s_{\text{best}} \leftarrow s_{\text{glob}};$ 
  while  $n_2 < N_{\text{iter}}^2$  do
     $s_{\text{cur}} \leftarrow s_{\text{best}};$ 
     $\text{destroyRepair}(s_{\text{cur}});$ 
     $\text{updateProb}(s_{\text{cur}});$ 
    if  $\text{acceptSolution}(s_{\text{cur}}, s_{\text{best}}, n_2)$  then
      |  $s_{\text{best}} \leftarrow s_{\text{cur}};$ 
    end
    if  $c(s_{\text{cur}}) < c(s_{\text{glob}})$  then
      |  $s_{\text{glob}} \leftarrow s_{\text{cur}};$ 
    end
     $n_2 \leftarrow n_2 + 1;$ 
  end
   $n_1 \leftarrow n_1 + 1;$ 
   $\text{Shake}(s_{\text{best}}, n_1);$ 
end

```

objective value. To widen the search space, a shaking procedure is started when the above procedure terminates (after a fixed number of iterations). The shaking procedure is especially included to cope with the dependency between the allocation of technicians and the optimization of the routes. This relation between the allocation and routes increases the difficulty of escaping local optima. The general outline of the ALNS procedure is given in Algorithm 3.1.

The ALNS procedure as described above is embedded in a two-stage procedure. In the first stage, we start with ten initial solutions and perform, on each of the ten initial solutions, a single run of the ALNS without the shaking procedure. In the second stage, we select the four best solutions from the first stage procedure and improve those four solutions with the full ALNS procedure. This two-stage procedure is the complete 2-ALNS. In the following, a complete description of the ALNS (as part of the 2-ALNS) is presented.

3.3.1 Destroy operators

A destroy operator's goal is to remove potentially suboptimal parts of a solution. This can either be achieved by destroying random parts of the solution, or one could provide more guidance in destroying the solution. Throughout this section, we will use the term *job* to indicate a pickup and delivery node corresponding to the same maintenance activity.

We divide the destroy operators into two classes. The first class, *random destroy operators*, destroys the solution by randomly removing jobs while taking the actual solution structure into account. The second class, *guided destroy operators*, are more sophisticated destroy operators. In the following, we discuss both classes of destroy operators.

3.3.1.1 Random destroy operators

By exploiting the problem structure, four random destroy operators are constructed, called *randomJob*, *randomVessel*, *randomDepot*, and *randomPeriod*. The first one removes a random job, the second operator removes all jobs from a random vessel in a random period, the third one removes all jobs from a random depot in a random period, and the fourth operator removes all jobs in a random period.

3.3.1.2 Guided destroy operators

This class consists of six destroy operators. The first guided destroy operator is the *worstRemoval* operator (e.g., Ropke and Pisinger (2006)). It removes the job that results in the largest cost decrease, i.e., the most expensive job is removed from the solution. When multiple jobs are removed by this operator, it iteratively removes single jobs and recalculates the cost decreases after each removal.

The five other guided destroy operators are based on the Shaw removal heuristic (see, e.g., Ropke and Pisinger (2006)) which tries to identify jobs that have similar characteristics. The main difference with the classical use of the Shaw removal heuristic is that the similarity of jobs does not depend on the actual solution, which improves the computational efficiency of the ALNS. Namely, the similarity of the jobs is computed before starting the ALNS instead of being computed every iteration.

To remove two jobs simultaneously from the solution, we define $S(i, j)$ as the similarity score for two jobs i and j measuring the degree of similarity of jobs i and j . The higher the score, the higher the probability of being selected simultaneously. It is defined in the following way,

$$S(i, j) = \hat{\alpha} \left(1 - \frac{\text{dist}(i, j) \chi_{i,j}}{\max_{a,b \in \mathcal{N}} \text{dist}(a, b)} \right) + \hat{\beta} \left(1 - \frac{\sum_{\ell=1}^L |q_{i\ell} - q_{j\ell}|}{\sum_{\ell=1}^L \max\{q_{i\ell}, q_{j\ell}\}} \right) + \hat{\gamma} \left(1 - \frac{|s_i - s_j|}{\max\{s_i, s_j\}} \right) + \hat{\delta} \left(1 - \frac{|d_i - d_j|}{\max\{d_i, d_j\}} \right), \quad (3.8)$$

where $\text{dist}(i, j)$ equals the Euclidean distance between nodes i and j , and $\chi_{i,j}$ equals 1 if $w_i = w_j$ and 0 otherwise. The four terms of (3.8) measure for jobs i and j : The relative

Algorithm 3.2: Random Cheapest Feasible Insertion (RCFI)

Data: Solution s , repair operator op , set of nodes U
 shuffle(U);
foreach $v \in U$ **do**
 | insertBestLocation(s, v, op);
end

difference in distance, the relative difference in technician demand, the similarity in service times, and the difference in due dates. Let $P(i, j) = S(i, j) / \sum_{a \in \mathcal{N}} S(i, a)$ be the probability of selecting job j while job i is already selected. In other words, job i is randomly selected whereas job j is selected with probabilities $P(i, j)$.

We enhance the use of five destroy operators based on this similarity criteria. For each operator, different weights are attached to the individual parts of $S(i, j)$. The *shaw* operator has positive weight on every individual part, i.e., $\hat{\alpha}, \hat{\beta}, \hat{\gamma}, \hat{\delta} > 0$. The *shaw-distance*, *shaw-technician*, *shaw-service*, and *shaw-penalty* guided destroy operators only have a non-zero value for $\hat{\alpha}, \hat{\beta}, \hat{\gamma}$, and $\hat{\delta}$, respectively.

3.3.2 Repair operators

In total, six repair operators are defined and divided into two classes, namely, the regular *RCFI* operators and the guided *RCFI* operators. The repair operators are variants of the so-called Random Cheapest Feasible Insertion (RCFI). The general structure of RCFI can be found in Algorithm 3.2. It consists of two main procedures. First, the set of to be inserted jobs is randomly sequenced. Second, we insert the jobs one by one on their ‘best’ location, based on an objective defined by the different repair operators. A single call to RCFI is of order $\mathcal{O}(n^2)$.

Some repair operators evaluate a job’s insert location with so-called noise, characterized by the noise parameter $0 \leq \Omega \leq 1$. Noise is a simple, but effective, way to include randomness in the ALNS. The costs used to compare insertion locations is obtained by multiplying the actual insertion costs with a uniformly drawn random number between $1 - \Omega$ and $1 + \Omega$.

3.3.2.1 Regular *RCFI* operators

Regular RCFI operators are straightforward insertion operators based on the total costs of the repaired solution. The first regular RCFI operator, *RCFI-cost*, finds the cheapest location for the job to be inserted. As *RCFI-cost* considers the jobs in a random sequence, applying it more than once may give better results. The second operator, *RCFI-k-cost*, evaluates the *RCFI-cost* operator k times, and applies the one

that results in the lowest costs. Both operators do not include noise in the objective functions, i.e., $\Omega = 0$. The *RCFI-noise-cost* is similar to the *RCFI-cost* operator, but Ω is strictly positive for this operator.

3.3.2.2 Guided *RCFI* operators

Guided RCFI operators are tailored operators to steer the solution into a direction that uses more or fewer technicians or vessel trips. The main purpose is to quickly escape local optima during the search. The fourth repair operator, *RCFI-noise-tech*, only considers the use of technicians by assuming zero travel costs, and includes noise. The fifth repair operator, *RCFI-vessel*, favors insertion in unused vessels by ignoring the travel costs if a job is inserted in an empty vessel. For the rest, it is similar to *RCFI-noise-cost*. The last repair operator, *RCFI-alloc*, penalizes the use of technicians that cause an increase in the allocation of that particular technician type to a depot. To be precise, a penalty $\eta > 0$ per unit increase of the allocation is added to the costs of the particular insertion location. In addition, it does not consider travel costs.

3.3.3 Destroy and repair procedure

At the core of the ALNS lies the Destroy and Repair Procedure. Its general overview is presented in Algorithm 3.3. The repair and destroy operators deployed are the ones described in the previous sections. The Destroy and Repair Procedure consists of n_d times applying a randomly selected destroy operator, and afterwards repairing the solution by one of the repair operators. The number n_d is randomly drawn between \underline{n}_d and \bar{n}_d in each iteration.

Each of the n_d destroy operators is applied a random number of times. We call this the destroy intensity: The number of entities a destroy operator removes from the solution. For destroy operators that consider individual jobs, it equals the number

Algorithm 3.3: destroyRepair(s)

```

Node set  $U \leftarrow \emptyset$ ;
 $k \leftarrow 0$ ;
while  $k < N_{init}$  do
     $op \leftarrow \text{chooseRandomDestroy}()$ ;
     $v \leftarrow \text{destroy}(s, op)$ ;
     $U \leftarrow U \cup v$ ;
     $k \leftarrow k + 1$ ;
end
 $op \leftarrow \text{chooseRandomRepair}$ ;
repair( $s, op$ );

```

of removed jobs. For destroy operators that consider vessels, depots or periods, it denotes the number of those entities. The destroy intensity is a randomly drawn number between \underline{I}_i and \bar{I}_i , where i refers to the class the destroy belongs to, i.e., $i = 1$ refers to the random destroy operators and $i = 2$ to the guided destroy operators. The actual destroy intensity is determined each time an operator is applied.

Besides operator settings, ALNS adapts the probability of the operators being selected depending on a move being successful or not. To do so, we keep an integer value $p_j > 0$ for every operator j individually. Their initial values are set as part of the operator group they belong to. These initial values are given by p_i^{init} , where $i = 1$ and $i = 2$ are defined as above, $i = 3$ refers to the class of regular RCFI operators, and $i = 4$ to the class of guided RCFI operators. If a move is accepted, the values p_j of the operators involved in the move are increased by $\Delta > 0$. Thus, drawing a random number between 0 and $\sum_j p_j$ is sufficient to select an operator. Finally, if the shaking procedure is started, the operator values p_j are reset to the initial value p_i^{init} of their corresponding operator group i .

3.3.4 Initial solution

Due to the multiple constraints and aspects of the TARP and its two variants, we first generate a set of N_{init} random solutions. Each random solution is generated by calling the *RCFI-Cost* repair operator on an empty solution, i.e., all jobs are inserted in a random order on their best location into the solution. We select the solution with the lowest cost. Notice that in the complete 2-ALNS procedure, we start with 10 initial solutions, each being generated as described in the former.

3.3.5 Acceptance criterion

We use a classical simulated annealing criterion to determine the acceptance of a move. It is defined by a cooling parameter $c_{\text{sa}} > 0$ and a weight parameter $w_{\text{sa}} > 0$. Let u_{old} and u_{new} be the objective before and after the move, and let n_{iter} be the current iteration. If $u_{\text{new}} < u_{\text{old}}$, we always accept the move. If a move results in a higher objective value, we calculate the acceptance probability of the move as

$$\exp\left(\frac{-w_{\text{sa}}\hat{u}}{\hat{n}c_{\text{sa}}}\right),$$

where $\hat{u} = (u_{\text{new}} - u_{\text{old}})/u_{\text{old}}$, and $\hat{n} = 1 - n_{\text{iter}}/N_{\text{iter}}^2$. A higher weight parameter steers towards a more deterministic search, i.e., the probability of accepting a move that results in a higher objective value is lower. In addition, a higher cooling parameter

results in lower acceptance probabilities for the first few iterations.

3.3.6 Shaking procedure

When N_{iter}^2 iterations of applying the destroy and repair procedure have passed (see Algorithm 3.1), a shaking procedure is started to escape from a possible local optimum. It is a destroy and repair procedure as indicated by Algorithm 3.3, although two major changes are made. First, only destroy operators of the random destroy operators class are used. Second, only repair operators *RCFI-noise-vessel* and *RCFI-noise-alloc* are used in the shaking procedure, being chosen at random. In addition, its intensity depends on the number of passed shaking procedures n_{fail} that did not yield a new best solution: It is applied n_{fail} times, with a minimum of 1 and a maximum of $n_{\text{fail}}^{\text{max}}$ times. Preliminary experiments have shown that this works well.

3.3.7 Differences between TARP, TARP-F, and TARP-G

It is clear that the TARP and TARP-F have exponentially larger solution spaces than the TARP-G. The freedom of deviating from a given allocation of technicians causes the solution space to be rather unstructured. To avoid being trapped in local optima, we change the first stage of the two-stage ALNS when we are solving TARP(-F) instances: We perform the first stage of the ALNS two times for each of the 10 instances. First, we run it as if it was a TARP-G instance, thereby taking the given allocation of technicians randomly around the expected number of technicians. Then we take this solution as an initial solution for another round of the first stage, but now solving the instances as they are. The second stage is similar for each problem type.

3.4 Two-stage ALNS performance

We evaluate the performance of the 2-ALNS by solving benchmark instances from the literature, obtained from Irawan et al. (2017). In addition, we generate a new set of computationally challenging benchmark instances. The 2-ALNS is implemented in *C++* and the experiments are performed on an Intel Xeon E5 2680v3 CPU (2.5 GHz). We made use of four parallel running threads on a single core, a setup available on any common desktop machine nowadays. In the following, we first describe our benchmark instances in more detail. We follow with a description of the 2-ALNS' parameter tuning and conclude with its computational performance.

Table 3.2: The effect of the destroy operators when solving TARP instances. For each operator, we denoted the maximum average objective increase if that particular operator is excluded from the 2-ALNS

Operator	Description	max increase (%)
<i>randomJob</i>	removes random jobs	1.22
<i>randomVessel</i>	removes random vessels	0.68
<i>randomDepot</i>	removes random depots	1.47
<i>randomPeriod</i>	removes random periods	1.62
<i>worstRemoval</i>	removes jobs leading to largest objective decrease	1.17
<i>shaw</i>	removes jobs based on shaw criteria	0.75
<i>shaw-distance</i>	removes jobs based on distance	0.73
<i>shaw-technician</i>	removes jobs based on technician use	0.58
<i>shaw-service</i>	removes jobs based on service time similarity	1.07
<i>shaw-penalty</i>	removes jobs based on due date similarity	0.72

3.4.1 Benchmark instances

Two existing sets of benchmark instances (G1 and G2) are obtained from Irawan et al. (2017). Since their problem is equivalent to the TARP-G in our setting, we compared the performance of the 2-ALNS with the exact approach by Irawan et al. (2017). Both benchmark sets comprise of two depots and three wind farms, with two available vessels per depot. The number of technician types L equals three. We refer to their paper for a complete description of parameter settings. Benchmark set G1 consists of 10 instances with 24 maintenance activities over a planning horizon of 3 days, whereas set G2 consists of 10 instances with 36 maintenance activities over a planning horizon of 7 days.

In addition to benchmark sets G1 and G2, we created a new benchmark set H. The instances herein vary in size and in the length of the time horizon, see Table 3.4. The parameters regarding the vessels, depots, technicians, and the location of the turbines and wind farms are the same as of those in G1 and G2. The actual maintenance activities in each instance are obtained in the following way: For the first two wind farms, we randomly draw the number of maintenance activities between 15 and 35 percent of the total number of activities of that particular instance. The third wind farm takes the remaining number of maintenance activities. Due dates are uniformly drawn over the time horizon, service times are uniformly drawn between 2 and 6 hours, and spare part weights are uniformly drawn between 300kg and 900kg. The number of required technicians is randomly between 1 and 3 for each technician type. This results in benchmark instances similar to the instances of G1 and G2. However, as opposed to the instances of G1 and G2, the number of jobs per wind

farm may vary. Preliminary experiments showed that this increases the complexity of the instances, which is in line with our aim of to create computationally challenging instances reflecting practical scenarios in offshore wind maintenance.

3.4.2 Parameter tuning

An overview of the parameter tuning of the 2-ALNS is provided in Appendix A. It describes an extended grid search through the main parameters. The parameter values used for benchmarking the 2-ALNS are given in Appendix A, Table 3.A.1.

3.4.3 The impact of the destroy operators

We continue by studying the impact of the destroy operators on the performance of the 2-ALNS. For this purpose, we solve a randomly generated set of instances (constructed as in dataset H) with 40, 45, 50, 55 and 60 maintenance activities. For each destroy operator, two experiments are performed. First, all the instances are solved with the 2-ALNS in which only that particular destroy operator is included, and none of the others. Second, we solve the instances while excluding that particular destroy operator and including all other destroy operators. This led to 21 experiments, i.e., two experiments for each operator and the complete 2-ALNS for comparison. All other elements of the 2-ALNS (e.g., the repair operators) are as described in Section 3.3.

Running the complete 2-ALNS with a single destroy operator is on average 3.6% worse than including all the destroy operators. This is mainly due to the *worstRemoval* operator, which is on its own 21.9 % worse. All other destroy operators, when included in isolation, result in average cost increases up to 7%.

The results for excluding a particular operator are presented in Table 3.2. For each of the destroy operators it holds that when excluded, one or more instances showed a deterioration in solution quality of at least 0.58%. This is on average 0.99% for the random destroy operators and 0.84% for the guided destroy operators. Especially excluding the *randomDepot* operator seems to harm the solution quality, as it leads to a 1.46% decrease in the average solution quality for particular instances. Moreover, it must be noted that the number of operators is not a significant factor in the calculation times, as the total number of iterations is fixed and the operators are randomly selected. We therefore include all the destroy operators into the 2-ALNS.

Table 3.3: Performance of the two-stage ALNS for benchmark sets G1 and G2 from Irawan et al. (2017)

Instance	Irawan et al. (2017)		TARP-G (min)		TARP-G (avg)		TARP-F (min)		TARP-F (avg)		TARP (min)		TARP (avg)				
	obj.	t(s)	obj.	gap %	obj.	gap %	obj.	gap %	obj.	gap %	obj.	gap %	obj.	gap %			
G1-01	41839.74	244.69	41839.74	0.00	41839.74	0.00	417	40422.90	-3.39	40432.48	-3.36	4.06	40168.21	-4.00	40168.21	-4.00	8.11
G1-02	33425.30	542.18	33425.30	0.00	33425.30	0.00	3.61	33425.30	0.00	33425.30	0.00	3.72	33425.30	0.00	33425.30	0.00	6.86
G1-03	36501.28	317.69	36501.28	0.00	36502.35	0.00	3.47	36038.08	-1.27	36177.04	-0.89	3.54	36038.08	-1.27	36038.08	-1.27	6.60
G1-04	31316.73	2356.46	31316.31	0.00	31316.31	0.00	3.53	31316.31	0.00	31316.31	0.00	3.86	31316.31	0.00	31316.31	0.00	6.79
G1-05	31147.73	549.59	31147.73	0.00	31147.73	0.00	3.89	30961.79	-0.60	31047.65	-0.32	4.15	30961.79	-0.60	30961.79	-0.60	7.38
G1-06	36141.68	134.34	36141.68	0.00	36141.68	0.00	3.76	36141.68	0.00	36141.68	0.00	4.33	36043.37	-0.27	36043.37	-0.27	7.68
G1-07	29995.24	1779.76	29995.24	0.00	29995.24	0.00	3.99	29995.24	0.00	29995.24	0.00	4.43	29995.24	0.00	29995.24	0.00	7.58
G1-08	37242.48	534.70	37242.48	0.00	37242.48	0.00	4.16	37242.48	0.00	37242.48	0.00	4.36	37242.48	0.00	37242.48	0.00	7.91
G1-09	30779.64	3465.71	30779.64	0.00	30779.64	0.00	4.10	30779.64	0.00	30779.64	0.00	4.29	30779.64	0.00	30779.64	0.00	7.66
G1-10	33030.87	674.37	33030.87	0.00	33030.87	0.00	4.23	33030.87	0.00	33030.87	0.00	4.28	33030.87	0.00	33030.87	0.00	8.33
G2-01	50835.10	1775.46	50896.33	0.12	51117.82	0.56	6.15	50835.10	0.00	51009.10	0.34	6.92	50814.94	-0.04	51030.02	0.38	7.17
G2-02	49047.09	3369.74	49047.09	0.00	49055.38	0.02	6.17	49047.09	0.00	49078.06	0.06	6.99	49047.09	0.00	49056.31	0.02	7.12
G2-03	44026.05	3518.18	44049.88	0.05	44050.40	0.06	6.86	44026.05	0.00	44048.02	0.05	7.79	44026.05	0.00	44045.37	0.04	7.93
G2-04	54139.22	1505.83	54139.22	0.00	54139.22	0.00	6.55	53564.54	-1.06	53564.54	-1.06	7.10	52755.73	-2.56	52755.73	-2.56	7.36
G2-05	53733.12	3549.63	53733.12	0.00	53733.12	0.00	6.35	52751.86	-1.83	52751.86	-1.83	6.92	52725.82	-1.87	52725.82	-1.87	7.32
G2-06	49156.52	3456.92	49156.52	0.00	49230.68	0.15	6.53	49156.52	0.00	49299.51	0.29	7.38	49156.52	0.00	49190.27	0.07	7.61
G2-07	49315.18	3436.71	49315.18	0.00	49327.49	0.02	6.35	49315.18	0.00	49315.18	0.00	7.24	49315.18	0.00	49323.39	0.02	7.37
G2-08	51051.50	1422.22	51051.50	0.00	51055.03	0.01	6.26	51051.50	0.00	51060.71	0.02	7.30	51051.50	0.00	51070.91	0.04	7.14
G2-09	57920.34	907.24	57920.34	0.00	57927.26	0.01	6.67	57920.34	0.00	57928.91	0.01	7.21	57920.34	0.00	57932.11	0.02	7.62
G2-10	52668.30	2866.38	52668.30	0.00	52668.30	0.00	6.73	52367.14	-0.57	52446.52	-0.42	7.55	52264.39	-0.77	52326.04	-0.65	7.80
Avg.	42665.66	1820.39	42669.89	0.01	42686.31	0.04	5.18	42469.48	-0.44	42504.56	-0.36	5.67	42403.94	-0.57	42422.86	-0.53	7.47

Table 3.4: Performance of the 2-ALINS for Benchmark Set H. For four instances, no feasible solution is found with the TARP-F and TARP-G, as indicated with a '-'. The percentage transportation costs, personnel costs and penalty costs are denoted in the columns headed by d_{cost}

Instance	n	T	TARP				TARP-F				TARP-G					
			z_{min}	d_{cost}	z_{avg}	t(s)	z_{min}	% dif.	d_{cost}	z_{avg}	t(s)	z_{min}	% dif.	d_{cost}	z_{avg}	t(s)
H-01	20	5	41009.20	(32, 60, 8)	41009.20	6.61	41719.16	1.73	(33, 59, 8)	41719.16	3.86	44010.35	7.32	(32, 56, 12)	44010.35	3.87
H-02	20	5	40200.06	(30, 67, 4)	40200.06	8.81	41460.18	3.13	(32, 65, 3)	41460.18	4.79	42007.17	4.50	(30, 64, 6)	42007.17	3.35
H-03	20	5	46161.78	(30, 55, 14)	46161.78	5.72	46178.70	0.04	(28, 55, 16)	46178.7	3.57	47885.12	3.73	(30, 55, 14)	47885.12	3.39
H-04	25	5	50886.58	(30, 63, 6)	50886.58	7.41	50886.58	0.00	(30, 63, 6)	50886.58	4.6	54518.67	7.14	(29, 56, 14)	54518.67	4.39
H-05	25	5	64242.77	(27, 58, 15)	64242.77	6.53	68058.22	5.94	(23, 53, 24)	68062.35	3.98	70921.57	10.40	(25, 52, 23)	70921.57	3.76
H-06	25	5	44104.52	(33, 67, 0)	44104.52	7.12	44104.52	0.00	(33, 67, 0)	44574.35	4.26	45262.92	2.63	(31, 61, 7)	45262.92	4.21
H-07	30	6	61915.15	(28, 65, 8)	61942.24	9.08	64414.22	4.04	(30, 59, 11)	64439.29	5.08	65119.43	5.18	(31, 59, 11)	65119.43	5.50
H-08	30	6	55523.02	(28, 63, 9)	55523.02	8.88	55800.53	0.50	(29, 63, 8)	55803.13	5.50	56719.30	2.15	(29, 63, 8)	56719.30	5.54
H-09	30	6	69244.24	(28, 62, 11)	69244.24	8.64	73072.19	5.53	(26, 56, 18)	75481.43	4.43	76635.92	10.67	(27, 55, 18)	76635.92	4.36
H-10	35	6	86464.86	(25, 58, 17)	86492.27	19.36	-	-	-	-	-	-	-	-	-	-
H-11	35	6	62763.19	(31, 66, 3)	62782.29	10.54	63586.56	1.31	(31, 66, 3)	64913.34	6.04	66962.10	6.69	(32, 63, 5)	67012.53	6.11
H-12	35	6	66825.21	(29, 61, 10)	66904.33	9.64	66875.44	0.08	(29, 61, 10)	67313.31	5.46	67526.70	1.05	(30, 60, 10)	67526.70	5.47
H-13	40	7	80437.54	(29, 66, 6)	80518.61	27.14	78681.03	3.00	(29, 62, 9)	78681.03	6.04	79466.96	4.03	(29, 62, 9)	79584.81	6.21
H-14	40	7	76387.97	(29, 64, 7)	76421.25	10.25	78959.48	7.60	(29, 56, 15)	89866.18	6.99	90758.93	9.00	(27, 54, 19)	90805.34	7.21
H-15	40	7	83268.38	(28, 59, 13)	83556.70	12.20	89594.48	9.40	(29, 56, 15)	96972.31	7.17	99592.69	13.52	(26, 53, 21)	100118.49	7.06
H-16	45	8	87730.87	(28, 62, 10)	87987.42	13.37	95973.41	9.40	(29, 60, 11)	91611.49	15.74	91792.10	8.53	(30, 57, 13)	91863.09	7.42
H-17	45	8	84580.57	(30, 62, 8)	84891.96	20.14	90163.54	6.60	(29, 60, 11)	94486.24	14.41	98603.87	10.92	(24, 52, 24)	99024.75	9.23
H-18	45	8	88896.50	(25, 58, 17)	89014.79	22.74	93168.44	4.81	(27, 57, 17)	95488.37	8.49	98752.39	2.97	(31, 67, 2)	98791.57	8.72
H-19	50	9	83282.72	(31, 67, 2)	83736.41	19.33	85752.39	2.97	(31, 67, 2)	86163.71	8.49	85752.39	2.97	(31, 67, 2)	85791.57	8.72
H-20	50	9	90408.42	(30, 66, 5)	91681.52	16.18	93881.95	3.84	(28, 64, 7)	95178.30	8.56	96703.78	6.96	(28, 62, 10)	97112.26	8.93
H-21	50	9	86663.27	(31, 66, 3)	86947.44	17.70	90015.99	3.87	(30, 67, 3)	91917.10	10.43	91234.05	5.27	(33, 63, 4)	91497.11	8.59
H-22	55	10	108868.20	(28, 66, 6)	109478.67	49.02	-	-	-	-	-	-	-	-	-	-
H-23	55	10	94744.55	(29, 69, 2)	95217.99	17.63	98036.58	3.47	(31, 66, 3)	99442.43	9.16	98541.14	4.01	(32, 65, 3)	98861.88	9.23
H-24	55	10	124547.25	(23, 56, 21)	125697.84	35.28	130733.18	4.97	(25, 53, 22)	134571.77	23.9	137397.52	10.32	(24, 49, 27)	138366.20	13.29
H-25	60	10	117622.65	(27, 64, 9)	118752.65	67.36	-	-	-	-	-	-	-	-	-	-
H-26	60	10	101444.19	(29, 71, 0)	102188.35	24.28	103394.53	1.92	(33, 67, 0)	104629.78	11.41	103486.01	2.01	(33, 67, 0)	104526.42	12.39
H-27	60	10	102830.53	(33, 66, 1)	104249.67	22.00	108044.30	5.07	(33, 64, 3)	110170.64	10.07	108249.10	5.27	(32, 64, 4)	109361.19	10.31
AVG.			77816.82		78142.02	17.89	77112.87	3.47		78022.73	8.00	79093.38	6.27		67575.29	5.87

3.4.4 Computational results

We solve the TARP-G instances of benchmarks G1 and G2 and compare these with exact solutions of the literature. Results are presented in Table 3.3. We solve the benchmark instances as if they were TARP(-F) instances as well. The instances from benchmark H are solved as TARP-G, TARP-F, and TARP instances. The results are presented in Table 3.4.

The results of solving the instances from benchmarks G1 and G2 are presented in Table 3.3. The first two columns represent the objective (obj.) and computation time in seconds of the exact approach of Irawan et al. (2017). The next two columns provide the minimum solution found by the 2-ALNS over ten runs, solving the instances as the TARP-G, and the corresponding gap with the best-known solution. The average values corresponding to the ten runs are provided afterwards. The columns headed by ‘TARP-F (min)’ and ‘TARP (min)’ present the minimum objective value and the corresponding gap to the best-known solution when we solve the instances as the TARP-F and TARP, respectively. We included the average results over 10 runs of the TARP and TARP-F as well.

It is clear that the 2-ALNS provides high-quality, and often optimal, solutions to the instances of Irawan et al. (2017). Computation times are on average 5.25 seconds, a decrease of factor 347 compared with the exact approach, at the expense of an average optimality gap of only 0.04%. Although the objective values of the instances of benchmark set G2 are not proven to be optimal by Irawan et al. (2017), the 2-ALNS obtains similar results. Moreover, experiments have shown that by increasing the 2-ALNS’ run time, it is possible to close the gap to the best-known solutions. The results indicate that often the complete symmetric allocation of technicians is best for the instances of Irawan et al. (2017), as solving the instances as TARP and TARP-F only leads to slight improvements. We provided more insights regarding the differences between the TARP, TARP-F and TARP-G with the instances of benchmark set H, and in Section 5.

Table 3.4 presents the results of our benchmark set H. Every instance is solved by the 2-ALNS as a TARP, TARP-F, and a TARP-G instance, as the column headings indicate. For each problem type, we report the minimum objective, the average objective, and the average computation time over 10 runs. In addition, we provide the distribution of the costs of the minimum solution under ‘ d_{cost} ’. It indicates the percentage of the total costs belonging to transportation costs, personnel costs and penalty costs. The column ‘% dif.’ indicates the increase of the minimum objective when compared to the TARP. For four instances, no feasible solutions can be found if they are treated as TARP-G or TARP-F, as marked with a ‘-’.

The merits of the flexible allocation policy of the TARP can be observed, as the TARP-F and TARP-G give average objective increases of 3.47% and 6.27%, respectively. The impact of the fixed allocation policy (TARP-F) is instance dependent, and will be further investigated in the simulation study in Section 5. Finally, computation times of the TARP and TARP-F are somewhat larger than of the TARP-G, because the TARP and TARP-F have exponentially larger solutions spaces than the TARP-G.

3.5 Managerial insights

The 2-ALNS provides optimal or near-optimal solutions to the short-term maintenance planning problem, as shown in Section 3.4. In benchmark set H, it is observed that sharing technicians, according to both the flexible and fixed allocation rule, reduces the overall maintenance costs. Recall that in the TARP, technicians are completely shared between O&M bases, whereas in the TARP-F, technicians are assigned to O&M bases but they cannot change between O&M bases over time. In the TARP-G, we take the number of technicians at each O&M base equal to the expected number of needed technicians, i.e., we do not include the assignment as a decision in the TARP-G.

We study in more detail the impact of technician sharing. We are especially interested in the characteristics of the underlying network in which technician sharing is most promising. We do so by an extensive Monte-Carlo simulation, in which we vary both the instance characteristics (e.g., number of technicians, number of jobs) and layout characteristics (e.g., the distance between wind farms and depots). First, we describe the set-up of the Monte-Carlo simulation. Second, we provide managerial insights on technician sharing based on the outcomes of the simulation. In particular, we answer the following questions:

1. Does technician sharing contribute to sustainability by reducing the waste of valuable resources such as technicians and vessels?
2. What is the added value of a completely flexible allocation rule over a fixed allocation rule.
3. Which geographical settings offer opportunities for cost-savings by means of technician sharing?

3.5.1 Simulation set-up

We summarize the simulation set-up here, and refer to Appendix B for the specifics. Throughout all experiments, we consider three O&M bases that are α kilometer apart.

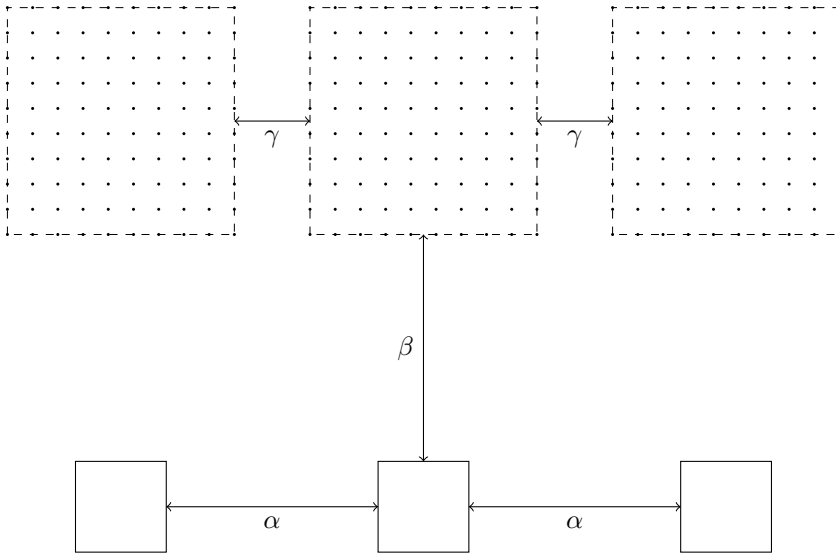


Figure 3.1: General layout used in the simulation study. The three squares at the bottom represent the O&M bases and the grids of dots represent the wind farms. The distances α , β , and γ are given parameters and are varied in the simulation study.

The distance between the O&M bases and the offshore wind farms equals β kilometer, and the distance between the wind farms is γ kilometer. We refer to α , β and γ as the *layout parameters*, see Figure 3.1.

The *instance parameters* of the simulation study are the number of available technicians \hat{q}_ℓ , the probability p of a turbine requiring maintenance, and the time horizon T . The fleet of vessels is kept fixed throughout all experiments, of which the specifics can be found in Appendix B. Other parameters, including service times and technician costs, are chosen in accordance with those in dataset H, as described in Section 3.4.1.

A single simulation experiment is defined by the instance and layout parameters described above. For each simulation experiment, we generate N_{sim} instances. Those are solved as the TARP, TARP-F and the TARP-G, where, in the latter one, we allocate technicians equally among the O&M bases as this coincides with the expected number of jobs serviced from the O&M bases. The 2-ALNS is deployed to approximate the expected cost of the short-term maintenance planning in each instance, under the instance and layout characteristics of that particular experiment. Each of the N_{sim} instances is generated by randomly selecting jobs for each wind farm, i.e., a turbine needs maintenance within the considered time horizon with probability p . As

Table 3.5: The mean and 90%, 95% and 99% quantiles of the performance measures. Given values are averages over all experiment runs for $T = 5$.

Cost Measure	Problem type	mean	90%	95%	99%
Total Costs	TARP-G	48427.56	66674.88	72284.24	76588.63
	TARP-F	47568.13	65788.28	70797.22	75178.79
	TARP	44879.77	61390.62	66309.56	70631.05
Penalty Costs	TARP-G	3649.70	7649.71	9040.22	10187.87
	TARP-F	3107.83	7020.74	7957.02	9191.37
	TARP	1466.98	3193.96	3800.00	4476.93
Number of trips	TARP-G	12.62	15.84	16.34	16.64
	TARP-F	12.24	15.46	16.08	16.40
	TARP	10.92	13.59	13.96	14.19
Mean time to maintenance	TARP-G	1.30	2.21	2.29	2.34
	TARP-F	1.16	2.15	2.27	2.31
	TARP	0.85	1.68	2.02	2.15

Table 3.6: Overview of the percentage feasible instances with $T = 5$ for different combinations of α and β .

	$\alpha, \beta \leq 30$	$\beta \leq 30, \alpha > 30$	$\beta > 30, \alpha \leq 30$	$\alpha, \beta > 30$
TARP-G	83.13	80.18	74.97	70.35
TARP-F	83.79	80.40	75.67	71.09
TARP	89.58	89.50	84.20	84.27

a result, the number of maintenance activities of the instances differs within a single experiment.

A complete overview of the instance and layout parameters can be found in Appendix B. We performed a complete grid search over the instance and layout characteristics space, obtaining cost estimations of the short-term maintenance planning for a total of 1500 scenarios. Common random numbers are used in every experiment to reduce the variance in our results. All experiments consist of $N_{\text{sim}} = 100$ instances, unless stated otherwise. Preliminary experiments have shown that this provides stable simulation outcomes.

3.5.2 Simulation outcomes

Four performance measures are used to assess the performance of the different technician allocation policies: The resulting total costs, the penalty costs, the number of needed visits to the wind farms (called trips), and the mean time to maintenance. First, we provide an in-depth analysis of the effect of technician sharing. Second, results on the relation between the layout parameters and the different sharing policies are presented.

3.5.2.1 Technician sharing analysis

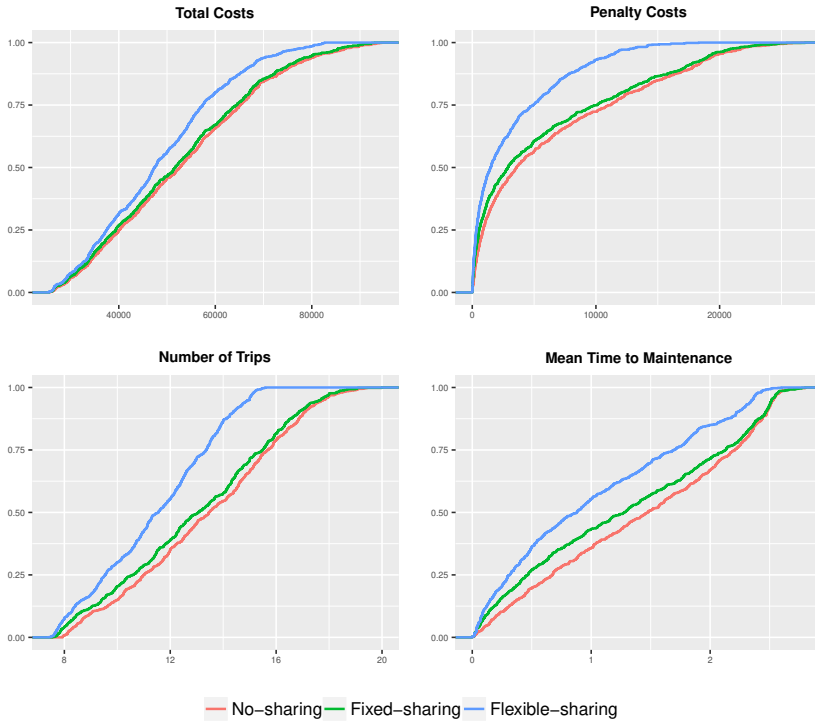


Figure 3.2: Empirical distribution functions of the Total Costs, Penalty Costs, Number of Trips and the Mean Time to Maintenance for all experiments with $T = 6$.

The mean and the 90%, 95%, and 99% quantiles of the four performance measures are summarized in Table 3.5. It shows the average outcomes of all experiments with $T = 5$. Instances that are infeasible by the TARP-G are excluded for the TARP-F and TARP as well. Analyzing the effect of technician sharing in general, it is seen that the number of trips decreases without increasing the mean time to maintenance. As a matter of fact, both the mean and the high quantiles of the mean time to maintenance are reduced if technicians are shared. The average cost increase of not allowing technicians to be shared equals 1.81% and 7.91% compared to the TARP-F and TARP, respectively. The main reason is the increase in penalty costs, which are 17.44% and 148.79% compared with the TARP-F and TARP, respectively. Summarizing, technician sharing leads to a decrease in the mean time to maintenance and the penalty costs, hence more maintenance activities are performed within their deadline. This allows the decision maker to cope more easily with future disturbances.

As mentioned before, not all the problem instances included in the experiments are feasible. Table 3.6 shows the percentage of instances that could be solved by each of the three problem variants, for different combinations of α and β . The TARP-F slightly increases feasibility of problem instances when $T = 5$, whereas the TARP shows a substantial increase of the feasibility of the instances at hand (e.g., from 70.35% to 84.27% for higher values of α and β). Thus, sharing of technicians is a natural way to cope with difficult short-term maintenance planning problems.

For the experiments with $T = 6$, empirical distribution functions of the four measures are given in Figure 3.2. With flexible sharing, 55% of the solutions use fewer than 12 trips while with a given allocation (no sharing) only 27% of the solutions use at most 12 trips. The maximum number of trips is reduced from 19 to 15. We conclude that flexibly sharing technicians effectively reduces the number of trips, while, at the same time, the mean time to maintenance is reduced.

Based on the simulation results presented so far, we can answer the first research question, posed at the beginning of Section 5, with the affirmative. Indeed, technician sharing contributes to sustainability as vessels and technicians are used more efficiently. We argue that sharing of technicians can be used as a tool by the operational manager to increase its ability to cope with extreme scenarios. In particular, flexibly allocating technicians clearly has its merits compared with the other allocation rules. For offshore wind operations that are relatively nearby, but operated in isolation, this can function as a simple but effective way to reduce costs.

3.5.2.2 Experiment layout impact

The Monte-Carlo simulation allows us to study the effect of the *layout parameters* on the efficiency of technician sharing. We present the results of the experiments with $T = 6$, as few instances are infeasible. We excluded the extreme scenarios with a minimum number of technicians and the highest job probability, as few comparisons could be made due to the high number of infeasible instances for the TARP-G.

In Figure 3.3, the effect of α (the distance between wind farms) on the performance of the different sharing policies is depicted. We compare the number of trips and the incurred penalty costs (as a proxy of the total costs) for two scenarios in which the wind farms are either relatively close, or relative far away, from the O&M bases. From Figure 3.3, one can observe that if O&M bases and wind farms are relatively far apart (higher values of α), the penalty costs can be reduced by allowing technician sharing. In addition, the decrease in the number of trips is slightly decreasing for higher values of α , meaning that the number of trips is most efficiently reduced when O&M bases are relatively close.

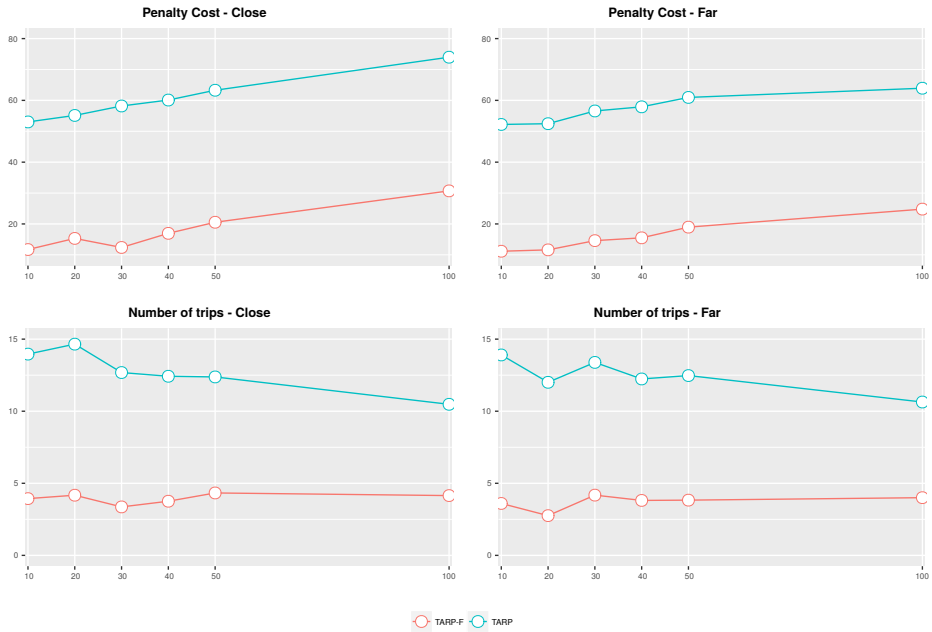


Figure 3.3: Percentage decrease in penalty costs and number of trips for the fixed (TARP-F) and the flexible (TARP) technician allocation policies for $\alpha \in \{10, 20, 30, 40, 50, 100\}$. The left two plots are for $\beta = 20$ and the right two plots are for $\beta = 40$

Reflecting on the second and third research question, we conclude that a flexible allocation policy increases robustness significantly compared to a fixed allocation policy. Additional experiments show that, for specific layouts, the fixed allocation performs significantly better than the no-sharing policy. This is especially true for wind farms operated in relative isolation, as we allow every wind farm to be served from every O&M Base.

3.6 Conclusions

In this paper, we introduced the Technician Allocation and Routing Problem for offshore wind farms (TARP). Its goal is to jointly optimize the daily allocation of technicians to O&M bases and the daily vessel routes transporting those technicians to offshore wind farms, in order to perform all maintenance activities throughout a given time horizon. We developed a Two-Stage Adaptive Large Neighborhood Search heuristic to solve the TARP and two variants. The first variant restricts the allocation of technicians to O&M bases to be constant throughout time, whereas the second

variant takes an allocation as given. We showed that the Two-Stage Adaptive Large Neighborhood Search provides high-quality, and often optimal, solutions on benchmark instances from the literature. In addition, we provided a new set of benchmark instances that are computationally challenging.

We embedded the Two-Stage Adaptive Large Neighborhood Search in a Monte-Carlo simulation to study the impact of the technician sharing in different practical scenarios of offshore wind maintenance service logistics. It is found that an increase in technician sharing reduces the number of vessel trips due to a more efficient use of the scarcely available technicians, which contributes to a more sustainable short-term maintenance planning. In addition, it is shown that the increased flexibility of technician sharing allows the short-term maintenance planner to cope with extreme scenarios more efficiently, leading to average cost-savings around 7%.

The opportunities for further research are numerous. To evaluate the effectiveness of the proposed solutions, one could impose stochastic travel times or stochastic weather conditions via a rolling horizon framework. Another possibility is the inclusion of strategic decisions regarding the composition of the fleet. Especially the decrease in vessel trips when technician are shared may contribute to significant cost-savings for such strategic decisions.

Appendices

3.A Parameter calibration

In this appendix, we present the details of the parameter calibration, and the resulting parameter values for benchmarking the 2-ALNS. The performance of the ALNS is crucial for the performance of the 2-ALNS, as it is the main ingredient of the 2-ALNS procedure. We therefore tuned the parameters of the ALNS by means of a grid search, as specified in Table 3.A.1.

The experiments are conducted on 90 instances with sizes varying between 20 and 60, similarly constructed as the instances in Benchmark Set H. As a measure of the ALNS' performance we take the average minimum objective value over 10 runs over all instances. The parameter values that gave the highest quality solutions are presented in Table 3.A.1. The column indicated by 'Benchmark' represents parameter settings used for benchmarking purposes in Section 4 (for all problem variants). We only considered parameter values that had an average run time of at most 20 seconds. For the simulation study in Section 5, we considered the best parameter values within a time limit of 150 seconds. They are given in the columns marked by 'TARP', 'TARP-F', and 'TARP-G'.

Table 3.A.1: Parameter calibration. Operator values (p_1, \dots, p_4) are taken equal to 1000. Shaw operator values $\hat{\alpha}, \hat{\beta}, \hat{\gamma}, \hat{\delta}$ are set equal to 5. The number of initial solution N_{init} is set equal to 1000

Parameter	Interpretation	Range	Benchmark	TARP-G	TARP-F	TARP
M_{iter}^1	Number of shaking procedure iterations	{20, 30, 50}	20	30	20	20
M_{iter}^2	Number of iterations Destroy and Repair procedure	{2000, 4000}	2000	4000	4000	4000
r_{max}	Maximum shaking intensity	{3, 5, 8}	5	8	5	8
c_{sa}	Cooling parameter of SA	{0.5, 1.0, 1.5}	1	0.5	1	0.5
u_{sa}	Weight parameter of SA	{50, 100, 150}	100	150	100	150
(\bar{n}_d, \bar{n}_d)	Number of destroy operators in a single destroy	{{(1,2), (1,3)}}	(1,2)	(1,3)	(1,3)	(1,3)
Δ	Change in operator probability after successful move	{1, 50}	1	1	50	50
(\bar{I}_1, \bar{I}_1)	Destroy operator class 1 intensities	{{(1,1), (1,2), (1,3)}}	(1,1)	(1,1)	(1,2)	(1,2)
(\bar{I}_2, \bar{I}_2)	Destroy operator class 2 intensities	{{(1,2), (1,3)}}	(1,3)	(1,3)	(1,3)	(1,3)

3.B Simulation study set-up

In this appendix, we present additional details to the simulation set-up summary as presented in Section 5. We made use of a fixed vehicle fleet as specified in Table 3.B.1. Every depot has two vessels: A relatively small though fast and cheap vessel, and a larger though slower and more expensive vessel. This is in line with practical scenarios of offshore wind as encountered in the literature (see, e.g., Stålhane, Hvattum, and Skaar (2015) and Irawan et al. (2017)).

In total 1500 scenarios are considered in the simulation study. Every scenario consists of a unique combination of instance and layout parameters, of which the considered values are denoted in Table 3.B.2.

Table 3.B.1: Overview of vessel characteristics in simulation. Distances are euclidean. Travel times are obtained by dividing travel distance with speed parameter. Travel costs are obtained by multiplying travel distance with the cost parameter.

Depot	Personnel cap.	Spare parts cap.	Speed	Cost
1	15	2000	45	225
1	25	25000	25	300
2	12	2000	45	225
2	20	25000	25	300
3	8	2000	45	225
3	16	25000	24	300

Table 3.B.2: Considered values of instance and layout parameters. For each combination of parameters 100 instances are solved as a TARP, TARP-G, and TARP-F.

Parameter	Interpretation	Considered Values
α	Distance between O&M Bases	{10, 20, 30, 40, 50, 100}
β	Distance between wind farms and O&M Bases	{10, 20, 30, 40, 50}
\hat{q}_ℓ	Number of technicians of each type	{9, 12, 15, 18, 21}
p	Probability of maintenance for each turbine	{0.08, 0.10, 0.12, 0.14, 0.16}
T	Length of time horizon	{5, 6}

