A Matheuristic Algorithm for an Offshore Multi-period Location Routing Problem

Simen Tørseth

Thesis for Master of Science Degree at the University of Bergen

Supervised by Ahmad Hemmati

Department of Informatics
&
Geophysical Institute

26th June 2024

©Copyright Simen Tørseth
The material in this publication is protected by copyright law.

Year:	2024
Title:	A Matheuristic Algorithm for an Offshore Multi-period Location Routing Problem
Author:	Simen Tørseth

Acknowledgements

Completing this thesis marks the end of five years as a student in the beautiful city of Bergen and as a student at the Integrated Master's programme in Energy. Initially, I was unsure if this degree was the right direction; however, I now know that those doubts were unfounded. I have discovered a love for maths, computers and and a deep appreciation for a special group of people I was fortunate enough to meet. I am grateful to all the people I have met along the way who have made me the person I am today.

I would like to thank my supervisor, Ahmad Hemmati, for his support and guidance through this last year. I have learned a lot and hope to make use of the optimisation techniques taught in your courses some time in the near future.

I would to also like to thank my best friend, Vilde Alstad, for being an incredible person. The countless hours you have listened to me complain, cry, and being a diva about this thesis are too much for only one person to handle. Nevertheless, you still supported me while also finishing your own thesis. Additionally, I want to bring special attention to my partners in crime, Akilavan Sivakumaran, Kamilla Valla Hagen, and Håkon Blågestad. Not only did we hit it off early at the start of university, but we have stuck together ever since. I sincerely hope that we continue being a part of Akilavan's babes for the foreseeable future.

To my mum, dad and little sister, thank you for always believing in me and pushing me to be my best.

Finally, this work is part of the Ocean Charger project, mainly funded by the Norwegian Research Council (Project No. 340936).

Simen Tørseth
Bergen, 26.06.24

Abstract

With the recent increase in energy production generated by offshore wind farms, regular maintenance is important for wind farm owners to ensure that they remain operational. Currently, maintenance tasks are performed by service vessels driven by fossil fuels and the work they perform may take several days to complete. To minimise the environmental impact of ship operations, emission-free vessel operations are possible by connecting vessels to the power grid. To achieve this, it is important to strategically determine the locations for charging infrastructure. This thesis presents a location routing problem with a multi-period planning horizon to consider the routes of electric vessels while determining the locations of potential charging stations to offer strategic insights to wind farm owners. As a solution method, a matheuristic algorithm is proposed. The routing of vessels is determined with an adaptive large neighbourhood search (ALNS) metaheuristic which is passed to an integer program (IP) to find the optimal placement of charging stations based on the vessel routing variables. The algorithm is tested on newly generated instances and performs well for both small and large instance sizes. Furthermore, the matheuristic is compared to an exact approach to validate its performance. These findings indicate that the proposed approach is effective and offers wind farms owners a tool to strategically site the infrastructure of charging units.

Contents

Abbreviations

2E-FCMPLRPTW	two-echelon fleet composition mix periodic location-routing problem with time windows
ALNS	adaptive large neighbourhood search
ALNSM	ALNS-based matheuristic
CSOV	construction service operation vessel
ELRP-TWPR	electric location routing problem with time windows and partial recharging
EVRPTW	electric vehcile routing problem with time windows
FLP	facility location problem
GHG	green house gas
IP	integer program
LCOE	levelised cost of energy
LNS	large neighbourhood search
LRP	location routing problem
LRPIF	location routing problem with intra-route facilities
MFSMPLRPTW	maritime fleet sizing mix periodic location routing problem with time windows
MIP	mixed-integer program
MP	mathematical programming
OWT	offshore wind turbine
PLRP	periodic location routing problem
PVRP	periodic vehicle routing problem
RRT	record-to-record
SA	simulated annealing
SVP	supply vessel planning
TA	threshold accepting
VRP	vehicle routing problem

List of Figures

List of Tables

Chapter 1

Introduction

With the impending climate crisis, the European Green Deal (European Commission, 2019) aims for a total reduction in greenhouse gas emissions and a carbon-neutral society by 2050. In alignment with these objectives, renewable energy sources are contributing increasingly to the energy mix each year. The offshore wind industry, in particular, has experienced significant growth with a global installed capacity of 64.3 GW (Hutchinson & Zhao, 2023). Furthermore, the International Energy Agency (IEA) projects that, to achieve carbon neutrality by 2050, the global installed capacity of offshore wind must reach 80 GW by 2030, with an annual installation rate of 70 GW from 2031 to 2050 (International Energy Agency, 2021).

One well-established challenge in the development of offshore wind power is the substantial investment cost. However, the global levelised cost of energy (LCOE) for offshore wind experienced a 44 % reduction in the period from 2009 to 2019, making it able to compete with existing fossil fuel energy technologies (European Commission, 2020). The harsh weather and environmental conditions to which offshore wind installations are exposed to necessitate regular maintenance of systems, turbines, and components to ensure optimal energy production and smooth operation. Currently, maintenance is carried out using service vessels and helicopters powered by diesel, resulting in emissions from green house gases (GHGs) due to diesel combustion (Ren et al., 2021). Transitioning to electric service vessels could substantially reduce maintenance-related emissions, thereby greening the value chain of offshore wind farms. This decarbonisation effort also aligns with the climate goals established by the European Green Deal and the United Nations sustainability goals 7, 13 and 14 (United Nations Department of Economic and Social Affairs, 2023).

Electric service vessels represent a sustainable strategy for decarbonising the offshore wind industry. This initiative is an important part of the Ocean Charger project, led by VARD (VARD, 2023) in which the University of Bergen is a partner. The Ocean Charger initiative aims to develop, test, validate, and commercialise new technological solutions for energy transfer to battery-powered ships offshore over a three-year period (VARD, 2023). Batteries seem the most promising technology for zero or low-emission vessels, forming the foundation of the Ocean Charger project. The project seeks to ensure emission-free vessel operations without the need for additional energy sources through connecting vessels to the power grid in wind farms.

One of the University of Bergen's contributions to the Ocean Charger project is aimed at the planning phase of offshore wind farms, focusing on the optimisation of infrastructure used by vessels such as the placement of charging units.

On-site charging stations are essential for electric vessels since the journey between the port and the wind farm consumes most of the battery's charge. The restricted range of electric service vessels, determined by the battery capacity, underlines the need for a charging infrastructure Łebkowski (2020). This issue is amplified offshore, where access and energy sources are limited. Therefore, planning efficient routes for service vessels and determining ideal locations for charging stations are important for reducing overall costs and extending the operational range of the vessels. To the author's knowledge, no such planning problem have been researched in the literature. Consequently, this thesis seeks to address the following research question: *How can ideal locations for charging stations be determined, and efficient routes for service vessels be planned to minimise total costs?*

Chapter 2

Literature Review

In the location routing problem (LRP), the concepts of determining locations for arbitrary types of facilities (depots, hubs, factories, etc.) and the routing of vehicles servicing a set of customers are combined. The LRP consists of two heavily researched problems: the facility location problem and the vehicle routing problem. The facility location problem (FLP) involves selecting optimal locations for one or more facilities to minimise costs or maximise service efficiency. This issue is prevalent in various sectors such as manufacturing, distribution, and public services, and was initially studied by Weber & Friedrich (1929). The vehicle routing problem (VRP), on the other hand, concerns the optimal assignment of routes to vehicles to serve a specified set of customers. The primary goal is usually to minimise total route cost or distance while considering constraints like vehicle capacities, customer demands, and route durations. This problem was formally introduced by Dantzig & Ramser (1959), who described it in the context of a truck dispatching scenario. The interdependence of the strategic and tactical problems was highlighted fifty years ago (Maranzana, 1964; Von Boventer, 1961; Webb, 1968). According to Prodhon & Prins (2014), the first authors to directly address the LRP were Watson-Gandy & Dohrn (1973) through modelling decreasing sales with the distance to a depot with a non-linear profit function. The result of not combining the planning horizons of strategic and tactical problems when location and routing decisions are being made may be very suboptimal (Salhi & Rand, 1989; Salhi & Nagy, 1999).

The literature on problem variants of the LRP is extensive. Over the years, these problems have become increasingly complex, tackling a variety of scenarios and considering different physical characteristics. However, only a fraction of recent works have addressed the multi-period LRP, which is more commonly known as periodic location routing problem (PLRP)

(Mara et al., 2021). The drawback of only addressing one planning period was highlighted by Salhi & Nagy (1999) and will lead to a sub-optimal solution to the planning problem.

Prodhon (2008) introduced the PLRP, which extends the periodic vehicle routing problem (PVRP) to include strategic decisions. In short, the PVRP is a VRP with added frequency constraints where customers must be visited over multiple time periods (Francis et al., 2008). Formally, the PLRP involves determining facility locations, customer visiting patterns, customer-to-facility assignments, and vehicle routes over multiple periods to minimize total costs consisting of facility opening, vehicle fixed, and routing costs. Furthermore, customers may be to different facilities over the whole planning period (Prodhon & Prins, 2014; Drexl & Schneider, 2015).

Recent research involving the PLRP considering an offshore logistics context has been done by Amiri et al. (2018, 2019). Amiri et al. (2018) presents the maritime fleet sizing mix periodic location routing problem with time windows (MFSMPLRPTW). The MFSMPLRPTW is based on the supply vessel planning (SVP) problem which is combination of the fleet composition problem and the PVRP. The SVP problem involves determining the best fleet composition required to service a specific group of offshore installations from a single onshore supply depot, while minimising total costs and providing reliable service. Simultaneously, it includes planning the weekly routes and vessel schedules (Halvorsen-Weare et al., 2012). Amiri et al. (2018) extends the SVP problem to an LRP by considering a set of suppliers and determining the best locations of onshore bases through minimising costs.

Amiri et al. (2019) build upon their MFSMPLRPTW problem by proposing the two-echelon fleet composition mix periodic location-routing problem with time windows (2E-FCMPLRPTW). The 2E-FCMPLRPTW extends the MFSMPLRPTW by considering three dependent operation levels with heterogeneous fleets of road vehicles and supply vessels. The objective is to minimise total costs while determining the best number and type of road vehicles and supply vessels, the ideal locations of onshore base and the most efficient routes and voyages along with their schedules.

Although the MFSMPLRPTW and 2E-FCMPLRPTW present variants of LRPs in an offshore setting, these novel problems do not consider electric vehicles and focus solely on onshore location planning.

The electric location routing problem with time windows and partial recharging (ELRP-TWPR) is presented by Schiffer & Walther (2017). This is the first model that integrates a full range of recharging possibilities while simultaneously determining the locations of charging stations and planning routes for a fleet of electric vehicles. An exact solution is presented which minimises the total travelled distance. Due to the NP-hard nature of the problem, results are presented only for small instances of the electric vehcile routing problem with time windows (EVRPTW) as created by Schneider et al. (2014).

Schiffer & Walther (2018) introduce the location routing problem with intra-route facilities (LRPIF), formalising the concept of locating of facilities en-route, known as intra-route facilities. The LRPIF extends the ELRP-TWPR by allowing recharging and reloading rate of freight to be non-linear. An ALNS metaheuristic is proposed to solve large instances and results for the EVRPTW and ELRP-TWPR are shown.

Since the main focus of this thesis is the ideal locations for charging stations, it is required that facilities are the origins and destinations of routes. Therefore, the approach of LRPIF can not be applied directly to the offshore planning problem.

In conclusion, different models and applications for the PLRP and the electric LRP have been previously discussed in the literature. Electric vehicles in logistics fleets, city logistics, and offshore settings in the oil and gas industry are also addressed. However, LRPs set in offshore wind farms are not common. In the following chapter, a multi-period LRP in an offshore setting is presented to fill this research gap.

Chapter 3

Problem Description

This chapter introduces the multi-period LRP. In Section 3.1, the planning challenges faced by wind farm owners in offshore wind logistics is addressed. Subsequently, in Section 3.2, the planning problem is formalised with strategic and operational requirements, presented as a mixed-integer program (MIP).

3.1 Problem setting

In the offshore wind sector, efficient logistics are important for minimising the costs associated with transportation of personnel and equipment, as well as for maintenance and repair operations. The logistics involve complex planning and execution due to factors like weather conditions and the need for specialised vessels. Wind farm owners contract a fleet of vessels to perform maintenance and repairs on offshore wind turbines (OWTs), while adhering to additional operational constraints such as a maintenance schedule and weather windows. This setting is similar to city logistics and logistics networks, however the problem has some specific characteristics:

1. Offshore logistics operations are often planned well in advance, with a focus on scheduled maintenance and installation activities. Similar to logistics networks, these operations tend to follow repetitive patterns, particularly for maintenance schedules. The demand for personnel and equipment can be better estimated over the long term due to the planned nature of offshore wind farm operations and monitoring of system components.

2. The spatial distribution of offshore wind farms and the locations of

onshore bases and ports vary significantly. This distribution affects the routing and scheduling of vessels, similar to how customer locations impact city logistics. However, the distances involved in offshore logistics are typically greater which affect the number of turbines a vessel can service in a trip.

3. Offshore logistics operations are heavily impacted by the environment and weather conditions. Adverse weather conditions, particularly high wave heights and strong wind speeds, restrict the access of service vessels to OWTs and limit the transfer of personnel from vessels to the OWTs (Ren et al., 2021).

Additional operational complexity arises with the usage of electric service vessels. Due to the smaller energy density of batteries, electric service vessels have a limited range of travel compared to vessels propelled by fossil fuels. This is not helped by the large spatial distribution of offshore wind farms which further decrease the number of OWTs a service vessel can visit during a trip. To mitigate using combustion engines or emptying the battery when travelling to site, a strategic network of charging stations has to exist.

In this setting, a wind farm owner faces strategic and operational decisions:

1. At the strategic level, a wind farm owner needs to decide where to build charging stations given a set of potential locations. The placement of charging stations is inter-dependent on routes of vessels, thus placement greatly impacts daily operations, costs, and the feasibility of maintenance schedules. Variation in maintenance schedules due to failure of different components in OWTs may lead to greater costs if the charging network determination does not take into account a longer time horizon.

2. At the operational level, the wind farm owner determines the service routes of the vessel fleet. When using an electric fleet, the wind farm owner must account for shorter travel distances to meet time restrictions of the maintenance schedule.

The planning period is divided into several time windows, each representing a working day with a specific length. Once one time window ends, the next one begins immediately. It is assumed that during the transition between time windows, all vessels are refuelled, and other necessary resources, such as parts and components, are replenished. The maximum number of available time windows is determined by the planning horizon, which marks the end of the planning period. Additionally, not all time windows need to be utilised if the maintenance schedule can be completed in a shorter time frame.

Before proceeding with the formalisation of the problem in the next section, a few key assumptions are made,

- A homogeneous fleet of vessels is assumed, such that every vessel has the same characteristics, e.g. battery charge capacity, energy consumption rate and average velocity,
- The energy consumption rate and average velocity are assumed to be constant,
- A vessel can be used at most once during a time window and must wait until the next time window before it is available again,
- Every vessel is assumed to be fully equipped and has enough crew onboard to to perform any maintenance task,
- Every vessel must return to the charging station it originated from (Nagy & Salhi, 2007),
- It is assumed that both weather and sea are calm because high wave heights and strong winds restrict the access of service vessels to OWTs and hinder personnel transfer from the vessel to the OWT (Ren et al., 2021),
- All turbines must be visited at most once, in any order, during the planning period.

3.2 Mathematical model

The following section defines the MIP for the multi-period LRP. The notation used is defined in Table 3.1. Let G be a graph, where $G = (\mathcal{N}, \mathcal{A})$ with a set of all nodes $\mathcal{N} = \mathcal{I} \cup \mathcal{OD}$ and a set of arcs $\mathcal{A} = \{(i, j) : i, j \in \mathcal{N}, i \neq j\}$. Let $\mathcal{I}$ be a set of turbine nodes and $\mathcal{OD}$ be a set of origin nodes and destination nodes for the potential charging stations. The origins are labeled from 1 to ϕ, where ϕ is the total number of origin nodes, and the destinations are labeled from $\phi + 1$ to $\phi + \delta$, where δ is the total number of destination nodes. Let $\mathcal{K}$ be a set of vessels. For each arc (i, j) a distance D_{ij} between node i and node j is given. The transportation time associated with each arc (i, j) is calculated using average velocity v as $T_{ij} = D_{ij}v^{-1}$. The set $\mathcal{T}$ holds the time windows represented by $[\underline{T_p}, \overline{T_p}]$, where $p \in \mathcal{P} = \{1, 2, \ldots, \pi\}$ denotes the p-th time window. The planning horizon is given by $T_{\max} = \overline{T_\pi}$, where π is the number of time windows. The time it takes to service a turbine is given by the time demand at each node is given by W_i. For the potential charging station nodes, W_i is set to zero. For each vessel a battery capacity Q is considered. The energy used to travel from node i to node j is formulated as a linear relation between the distance D_{ij} and the energy consumption rate R. The binary decision variable x_{ijk} determines whether an arc (i, j) is used by vessel k. If the arc (i, j) is used, $x_{ijk} = 1$. Otherwise, if arc (i, j) is not used, $x_{ijk} = 0$. To determine whether a charging station is built at node j, the binary decision variable y_j is used. If a charging station is built at node j, $y_j = 1$, otherwise $y_j = 0$. The binary variable b_{kp} determines if vessel k uses time window p. If vessel k uses time window p, $b_{kp} = 1$. Otherwise, $b_{kp} = 0$. The time of arrival of a vessel k at node i is given by t_{ik}. Furthermore, the remaining battery capacity of a vessel k when arriving at node i is represented by e^a_{ik}. Conversely, the remaining battery capacity of a vessel k when departing node i is given by e^d_{ik}. Binary variables d_{ikp} and a_{jkp} limit vessel usage to one trip per time window. If origin i is departed by vessel k in time window $p \in P$, $d_{ikp} = 1$, otherwise $d_{ikp} = 0$. Binary a_{jkp} is analogously defined. Costs are divided into sailing costs C_{ij}, fixed costs for using a vessel F_k and fixed costs for building a charging station G_j.

Table 3.1: Notation used for the mathematical formulation of the multi-period LRP

Symbol	Description
$\mathcal{I}$	Set of turbine nodes
$\mathcal{OD}$	Set of origins $\{1, 2, \ldots, \phi\}$ and destinations $\{\phi + 1, \phi + 2, \ldots, \delta\}$ for the potential charging stations
$\mathcal{N}_o$	Set of turbine nodes plus origin nodes $o \in \{1, 2, \ldots, \phi\}$
$\mathcal{N}_d$	Set of turbine nodes plus destination nodes $d \in \{\phi+1, \phi+2, \ldots, \phi+\delta\}$
$\mathcal{N}$	Set of turbine nodes plus origin and destination nodes
$\mathcal{K}$	Set of all vessels
$\mathcal{T}$	Set of time window parameters $[\underline{T_p}, \overline{T_p}]$
$\mathcal{P}$	Set of time windows, $\{1, 2, \ldots, \pi\}$
G_j	Fixed costs for building charging station j
F_k	Fixed costs for using vessel k
C_{ij}	Cost of sailing from node i to node j
W_i	Demand at node i
T_{ij}	Transportation time of arc (i, j)
D_{ij}	Distance from node i to node j
Q	Charge capacity
R	Energy consumption per distance unit travelled of a vessel
$\overline{T_p}$	Upper bound of time window $p \in \mathcal{P}$
$\underline{T_p}$	Lower bound of time window $p \in \mathcal{P}$
$T_{\max}$	The planning horizon
π	Number of time windows
ϕ	Number of origin nodes
δ	Number of destination nodes
x_{ijk}	Binary: Arc (i, j) is travelled by vessel k
y_j	Binary: Charging station is sited at node j
z_k	Binary: If vessel k is used
b_{kp}	Binary: If vessel k uses time window $p \in \mathcal{P}$
t_{ik}	Arrival time at node i of vessel k
e^a_{ik}	Remaining battery capacity of vessel k when arriving at node i
e^d_{ik}	Remaining battery capacity of vessel k when departing from node i
d_{ikp}	Binary: If origin i is departed by vessel k in time window $p \in \mathcal{P}$
a_{jkp}	Binary: If destination j is departed by vessel k in time window $p \in \mathcal{P}$

Note: The table is subdivided as follows: sets/parameters/decision variables/auxiliary variables.

$$\underset{Z}{\text{minimize}} \quad Z = \sum_{j\in\mathcal{E}} G_j y_j + \sum_{k\in\mathcal{K}} F_k z_k + \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{N}_o, i\neq j} \sum_{j\in\mathcal{N}_d} C_{ij} x_{ijk} \tag{1}$$

subject to

$$\sum_{k\in\mathcal{K}} \sum_{j\in\mathcal{N}_d, i\neq j} x_{ijk} = 1, \qquad \forall i \in \mathcal{I}, \tag{2}$$

$$\sum_{i\in\mathcal{I}, i\neq j} x_{ijk} \leq 1, \qquad \forall j \in \{\phi+1, \phi+2, \dots, \phi+\delta\}, k \in \mathcal{K}, \tag{3}$$

$$\sum_{j\in\mathcal{N}_d, i\neq j} x_{ijk} = \sum_{j\in\mathcal{N}_o, i\neq j} x_{jik}, \qquad \forall k \in \mathcal{K}, \forall i \in \mathcal{I}, \tag{4}$$

$$\sum_{j\in\mathcal{I}} x_{j(i+\delta)k} = \sum_{j\in\mathcal{I}} x_{ijk}, \qquad \forall i \in \{1, 2, \dots, \phi\}, \forall k \in \mathcal{K}, i \neq j, \tag{5}$$

$$\sum_{i=1}^{\phi} \sum_{j\in\mathcal{I}} \sum_{k\in\mathcal{K}} x_{ijk} \leq |\mathcal{K}| \cdot \pi, \tag{6}$$

$$\sum_{i\in\mathcal{I}} \sum_{k\in\mathcal{K}} x_{ijk} \leq y_j |\mathcal{I}|, \qquad \forall j \in \{\phi+1, \phi+2, \dots, \phi+\delta\}, \tag{7}$$

$$\sum_{i\in\mathcal{N}_o} \sum_{j\in\mathcal{N}_d} x_{ijk} \leq z_k(|\mathcal{I}|+1), \qquad \forall k \in \mathcal{K}, i \neq j, \tag{8}$$

$$\sum_{p\in\mathcal{P}} b_{kp} = 1, \qquad \forall k \in \mathcal{K}, \tag{9}$$

$$\sum_{p\in\mathcal{P}} b_{kp} \underline{T_p} \leq t_{ik}, \qquad \forall i \in \mathcal{N}, \forall k \in \mathcal{K}, \tag{10}$$

$$\sum_{p\in\mathcal{P}} b_{kp} \overline{T_p} \geq t_{ik}, \qquad \forall i \in \mathcal{N}, \forall k \in \mathcal{K}, \tag{11}$$

$$d_{ikp} \geq x_{ijk} - (1 - b_{kp}) \qquad \forall i \in \{1, 2, \dots, \phi\}, \forall j \in \mathcal{I}, \forall k \in \mathcal{K}, \forall p \in \mathcal{P}, \tag{12}$$

$$a_{jkp} \geq x_{ijk} - (1 - b_{kp}) \qquad \forall i \in \mathcal{I}, \forall j \in \{\phi+1, \phi+2, \dots, \phi+\delta\}, \forall k \in \mathcal{K}, \forall p \in \mathcal{P}, \tag{13}$$

$$\sum_{i=1}^{\phi} d_{ikp} \leq b_{kp} \qquad \forall k \in \mathcal{K}, \forall p \in \mathcal{P}, \tag{14}$$

$$\sum_{j=\phi+1}^{\phi+\delta} a_{jkp} \leq b_{kp} \qquad \forall k \in \mathcal{K}, \forall p \in \mathcal{P}, \tag{15}$$

$$t_{jk} \geq t_{ik} + (W_i + T_{ij}) x_{ijk}, - T_{\max}(1 - x_{ijk}), \qquad \forall i \in \mathcal{N}_o, \forall j \in \mathcal{N}_d, \forall k \in \mathcal{K}, i \neq j, \tag{16}$$

$$
\begin{aligned}
& e^a_{jk} \leq e^d_{ik} - x_{ijk} D_{ij} R, && \\
& \qquad + Q(1 - x_{ijk}), && \forall i \in \mathcal{N}_o, \forall j \in \mathcal{N}_d, \forall k \in \mathcal{K}, i \neq j, && (17)\\
& e^d_{ik} = Q, && \forall i \in \{1, 2, \ldots, \phi\}, \forall k \in \mathcal{K}, && (18)\\
& e^d_{ik} = e^a_{ik}, && \forall i \in \mathcal{I}, \forall k \in \mathcal{K}, && (19)\\
& t_{ik} \geq 0, && \forall i \in \mathcal{N}, \forall k \in \mathcal{K}, && (20)\\
& e^a_{ik} \geq 0, && \forall i \in \mathcal{N}, \forall k \in \mathcal{K}, && (21)\\
& e^d_{ik} \geq 0, && \forall i \in \mathcal{N}, \forall k \in \mathcal{K}, && (22)\\
& x_{ijk} \in \{0, 1\}, && \forall i \in \mathcal{N}_o, \forall j \in \mathcal{N}_d, \forall k \in \mathcal{K}, && (23)\\
& y_j \in \{0, 1\}, && \forall j \in \{\phi + 1, \phi + 2, \ldots, \phi + \delta\}, && (24)\\
& b_{kp} \in \{0, 1\}, && \forall k \in \mathcal{K}, \forall p \in \mathcal{P}, && (25)\\
& d_{ikp} \in \{0, 1\}, && \forall i \in \{1, 2, \ldots, \phi\}, \forall k \in \mathcal{K}, \forall p \in \mathcal{P}, && (26)\\
& a_{jkp} \in \{0, 1\}, && \forall j \in \{\phi + 1, \phi + 2, \ldots, \phi + \delta\}, && \\
& && \forall k \in \mathcal{K}, \forall p \in \mathcal{P}. && (27)
\end{aligned}
$$

The objective function (1) minimises the total costs associated with construction of charging stations, operation of vessels and sailing costs. Single assignment is obtained by constraints (2) and ensures that each turbine is only visited once. Single assignment is relaxed for charging stations by constraints (3). Constraints (4) ensure that the vessel that leaves the turbine is the same vessel that visits the turbine. Trips are forced to start and end at the same charging station, enforced by constraints (5). Constraint (6) ensures the feasibility of time windows. Vessels are restricted to charge only at charging stations given by constraints (7). Constraints (8) tracks if vessel k is used. Time constraints are given by (9)-(11). While constraints (9) indicates that only one of the available time windows is used per vessel, constraints (10) and (11) enforces feasibility of time windows for all nodes. Vessel trip restrictions are determined in (12)-(15). Constraints (12) and (13) link origins and destinations of tours to time windows. There can be at most one departure and arrival in a time window given by constraints (14) and (15), for any given vessel. Arrival times are given by constraints (16). The remaining battery capacity of a vessel at arrival is determined by (17). Constraints (18) ensures that a vessel leaves a station with full battery capacity. While servicing a turbine no energy loss occurs, as guaranteed by constraints (19). Arrival times are defined for the positive values by constraints (20). Constraints (21) and (22) ensures that energy values are within the battery capacity range. Finally, binary variables are defined in constraint (23)-(27).

Chapter 4

Solution Approach

In this chapter, a solution approach is presented for solving instances of the multi-period LRP presented in Chapter 3. To solve the described problem, a matheuristic algorithm is used. In the algorithm, the advantages of a mathematical optimisation solver are utilised in combination with a metaheuristic framework to efficiently produce solutions.

In recent years, the notion of combining metaheuristics and mathematical programming (MP) has gained popularity (Puchinger & Raidl, 2005; Raidl, 2006). This hybridisation is called a 'matheuristic' and can be defined in two ways according to a 'master-slave' concept (Maniezzo et al., 2010). Maniezzo et al. (2010) states that, either 1) a metaheuristic is the 'master' and controls when to call the MP, or 2) the MP takes on the role as the 'master' and is in control of the metaheuristic algorithm.

In performing the repair step in the ALNS-framework the usage of a MIP solver is gaining attraction as these solvers consistently improve in performance with each new version released (Robert E. Bixby, 2012). In the presented algorithm, the first approach of the matheuristic framework is used, where the metaheuristic acts as the master, controlling when to call the MP. This choice leverages the strengths of the metaheuristic to guide the overall search process while capitalizing on the advancements in MIP solvers to enhance the repair steps.

The presented algorithm, referred to as ALNSM, is based on the algorithm proposed by Heidari & Hemmati (2023). The routing variables stated in Section 3.2 are managed by the metaheuristic procedure ALNS. Subsequently, the non-routing variables are solved by a MIP solver, while taking the routing variables into account.

The algorithm is outlined in Algorithm 1. First, an initial solution is constructed and assigned to s in line (2) and assigned to the best solution s_{best} in line (3). The search is initiated in line (5) and continues until $\eta^{\max}$

number of iterations is reached. The escape-algorithm is called every η^{esc} number of iterations, in line (7). A heuristic is selected and applied to the candidate solution s' in line (10) and (11). The candidate solution is passed to the MIP solver in line (12). In line (15), s_{best} is updated if the candidate solution s' is feasible and improves the best solution. The candidate solution is accepted in line (18) if it meets the acceptance criterion. After η^{seg} number of iterations, in line (23), the scores of each heuristic are updated. Finally, the best found solution s_{best} is returned in line (28) when the stop condition is met.

Algorithm 1 ALNS-based matheuristic

```
Input: All parameters, a of operators $\mathcal{H}$
 1: $InitialiseParameters()$
 2: $s \leftarrow InitialSolution()$ (Section 4.1)
 3: $s_{\text{best}} \leftarrow s$
 4: $\iota \leftarrow 0$
 5: while $\iota < \eta^{\max}$ and $\iota - \iota_{\text{imp}} < \eta^{\max}_{\text{noi}}$ do
 6:     if modulo$(\iota, \eta^{\text{esc}})$ then
 7:         $s' \leftarrow escape(s)$ (Section 4.4)
 8:     end if
 9:     $s' \leftarrow s$
10:     select $h \in \mathcal{H}$
11:     apply heuristic $h$ to $s'$
12:     pass $s'$ to MIP solver (Section 4.5) and determine station assignment
13:     if $s'$ is feasible then
14:         if $f(s') < f(s_{\text{best}})$ then
15:             $s_{\text{best}} \leftarrow s'$
16:             $\iota_{\text{imp}} \leftarrow \iota$
17:         end if
18:         if $accept(s', s)$ then
19:             $s \leftarrow s'$
20:         end if
21:         $updateScores()$
22:     end if
23:     if modulo$(\iota, \eta^{\text{seg}}) = 0$ then
24:         $updateWeights()$
25:     end if
26:     $\iota \leftarrow \iota + 1$
27: end while
28: return $s_{\text{best}}$
```

The construction of the initial solution is given in Section 4.1, followed by the explanation of the ALNS-framework in Section 4.2. The escape-algorithm is presented in Section 4.4 and the IP that determines the station assignment is described in Section 4.5.

4.1 Initial solution

It is widely accepted in the optimisation community that generating an initial solution is an important step, given that subsequent solutions depend on previous solutions and ultimately tracing back to the initial solution (Sarhani et al., 2023). However, biasing the initial solution may be dangerous and may lead to premature convergence (Sean Luke, 2013).

A construction heuristic is employed to create an initial solution. The initial solution is not allowed to be infeasible concerning time windows or battery charge level. Since the ALNSM does not allow infeasible solutions, an infeasible initial solution would never be improved and thus a good solution cannot be generated with the ALNSM. The construction heuristic works as follows:

1. Assign each turbine to its closest charging station with regards to distance.
2. For each charging station and its assigned turbines,
    - 2.1. Create a new route
    - 2.2. Add turbines to new route as long as it is feasible
    - 2.3. If the new route is not feasible, discard it, and repeat Step 2.1.

4.2 Adaptive large neighbourhood search algorithm

The main part of the ALNSM is based off of the work of Ropke & Pisinger (2006) who propose the adaptive large neighbourhood search framework. The structure of the ALNS-framework is shown in Algorithm 2. The ALNS is an innovative heuristic method used primarily for solving complex combinatorial optimization problems. Ropke & Pisinger (2006) extend the large neighbourhood search (LNS) proposed by Shaw (1998) by incorporating adaptive mechanisms to enhance the search process. The LNS operates by systematically breaking down and subsequently reconstructing the current solution through ‘destroy’ and ‘repair’ heuristics. The ‘destroy’ heuristic removes components of the existing solution, while the ‘repair’ heuristic rebuilds the solution to its full form, potentially improving upon the previous solution.

The ALNS uses the same principle as LNS of using ‘destroy’ and ‘repair’ methods to improve solutions. However, where a careful consideration of heuristics used in LNS is needed depending on the problem instance, the ALNS employs an adaptive mechanism to choose between multiple available destroy and repair heuristics. This adaptive mechanism is called ‘roulette wheel selection’. In roulette wheel selection each heuristic is assigned a weight such that it can be selected with a certain probability. If there are k heuristics with assigned weights ω_i, $i \in \{1, 2, \dots, k\}$, the probability of heuristic i being selected is given by

$$\frac{\omega_i}{\sum_{j=1}^{k} \omega_j}. \tag{4.1}$$

To avoid manually determining the weights of all heuristics, a dynamic adaptation method is used. The weights are adjusted based on their past performance, thus encouraging the exploration of more effective heuristics during the search. The search is divided into ‘segments’ to prevent it from getting stuck in a local optima and allows the algorithm to explore different parts of the solution space. Each heuristic is assigned a score to keep track of its performance during a ‘segment’ of the search. After each segment s, the weights $\omega_{i,s}$ of heuristics i are adjusted by

$$\omega_{i,s+1} = \omega_{i,s}\,(1-r) + r\frac{\sigma_i}{\theta_i}, \tag{4.2}$$

based on their contribution to improving the solution. The contribution is assessed with the score σ_i and the number of times θ_i the heuristic was used in segment s. The smoothing factor r controls the speed at which the weights are adjusted based on the heuristic’s effectiveness.

Whether to accept a new solution created by the ‘destroy’ and ‘repair’ heuristics is determined by the acceptance criterion. By accepting worse solutions temporarily, one may avoid local optima and encourage more exploration of the solution space. A simple acceptance criterion would be a ‘hill climber’ accept method proposed by Shaw (1998) in the original paper on LNS, where only better solutions than the global best are accepted. The most popular acceptance criteria comes from the popular metaheuristic framework simulated annealing (SA). The acceptance criteria in SA is based on the concept of annealing from metallurgy. Let σ' be a temporary solution, which is accepted with respect to the current solution σ with the probability $e^{\frac{-\Delta E}{T}}$ where $-\Delta E$ is the difference in objective value between the temporary solution and the current solution and T is the temperature. Initially, the temperature is set to a large non-zero value T_0 which is reduced progressively according to cooling schedule c, where $0 < c < 1$, during the search process.

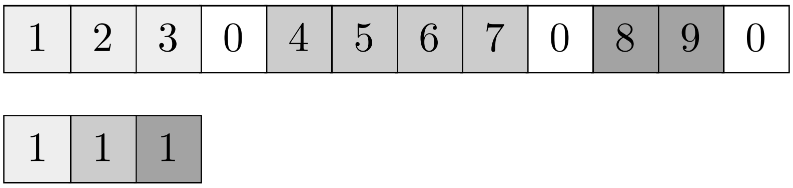
Santini et al. (2018) presents a large computational study which shows that the acceptance criteria SA, threshold accepting (TA) and record-to-record (RRT) perform best in the ALNS-framework compared to others. The acceptance criterion used in the ALNSM algorithm is explained in Section 4.3.

Algorithm 2 ALNS-Framework

```
Input: Initial solutions $s$
function ALNS
    $s_{\text{best}} \leftarrow s$
    repeat
        $s' \leftarrow s$
        remove $q$ items from $s'$
        reinsert removed items into $s'$
        if $(accept)(s', s)$ then
            $s \leftarrow s'$
        end if
        if $(f(s') < f(s_{\text{best}}))$ then
            $s_{\text{best}} \leftarrow s'$
        end if
        update parameters for selection
    until stop condition is met
    return $s_{\text{best}}$
end function
```

4.2.1 Solution representation

The solution representation used in the ALNSM is a two dimensional vector. The first row of this vector denotes the route for each tour, using zero as a delimiter between tours. It is read from left to right, with each non-zero number representing a visited turbine. The second row of the vector denotes the charging stations for these tours, defining both the start and end locations of each tour. An example solution is presented in



1	2	3	0	4	5	6	7	0	8	9	0

1	1	1

Figure 4.1: Solution representation of an example solution

Solution representation of an example solution used in the ALNSM. The top array holds the routes of the vessel where vessels are separated by the number zero. The bottom array tracks which charging station assigned to the start and end of each route in the top array. Each shade of gray belong to the same route.

Figure 4.1, where $\mathcal{I}$ is the set of turbines $\mathcal{I} \in \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ and $\mathcal{R}$ is the set of potential charging stations $\mathcal{R} \in \{1, 2, 3\}$.

This solution indicates that the first tour, originating from charging station 1, covers turbine nodes 1, 2, and 3. The second tour, starting at charging station 1, visits nodes 4 through 7, and the third tour, beginning again at station 1, services nodes 8 and 9. Charging station 2 and 3 are absent from the solution and thus remain unbuilt.

4.2.2 Operators

In this section, the operators utilized in the ALNSM are introduced. To determine the routing strategy for the fleet of vessels, four types of operators are implemented: Random-Remove, Cost-Random-Remove, Cluster-Random-Remove, and Greedy-Insert. Each operator focuses on the sequence of visits to turbine nodes with different methodologies involving both random and cost-based selection criteria. This enables exploration of the solution space in an efficient manner.

- Random-Remove: The Random-Remove operator removes $q \in [1, q_{\max}]$ nodes randomly from all tours in the solution.

- Cost-Random-Remove: The Cost-Random-Remove operator removes $q \in [1, q_{\max}]$ nodes from the solution according to the highest contributing cost. Nodes are removed with Shaw removal using removal costs of each node as the selection criteria (Shaw, 1998). This introduces diversity to the greedy nature of the operator and helps the ALNSM explore a wider solution space.

- Cluster-Random-Remove: The Cluster-Random-Remove operator removes $q \in [1, q_{\max}]$ nodes from the solution based on a similarity measure. First, it picks a node randomly and then iteratively removes q similar nodes using Shaw removal (1998).

- Greedy-Insert: The Greedy-Insert operator iteratively inserts removed nodes one by one into the solution. First, it finds where it is feasible to insert, and then inserts the node according to its least cost contributing position. If no feasible inserts are found, a new tour is opened and the node is inserted. The charging station from which this new tour originates is chosen randomly.

4.3 Acceptance Criterion

The ALNSM algorithm adopts the RRT acceptance criterion based on the findings of Santini et al. (2018), which suggest it is the preferred approach when implementing the ALNS-framework. Originally proposed by Dueck (1993), the goal is to accept a new solution if it deviates from the best found solution by a deviation D. That is, let s_{best} be the best found solution (the record), then a new solution s' is accepted if it is smaller than $s_{\text{best}} + D$. The definition of the deviation D is borrowed from Hemmati & Hvattum (2017) and is defined as

$$D = 0.2 \left(\frac{\eta^{\max} - \iota}{\eta^{\max}} \right) s_{\text{best}}, \tag{4.3}$$

where $\eta^{\max}$ is the maximum number of iterations performed by the ALNSM and ι is the current iteration number. The definition of D is linear which converges to zero at the end of the search: this approach simplifies tuning by reducing the number of parameters by one, while maintaining the quality of the solution (Santini et al., 2018).

4.4 Escape algorithm

To escape local optima an escape algorithm is used. In Algorithm 1, after η^{esc} iterations the escape algorithm is applied to the solution. In this algorithm, q nodes are selected to be removed from the solution where q is equal to q_{esc}. Then, the removed nodes are randomly inserted one by one at the first feasible position in the solution. If no feasible insertion is possible a new tour is created and assigned a station at random. The new solution is passed to the ALNSM regardless of its quality. The escape algorithm is outlined in Algorithm 3.

Algorithm 3 Escape algorithm

```
Input: solution $s$, $q$ nodes to remove
for $i = 1$ to $q$ do
    Remove a node at from random from $s$
end for
for each removed node do
    Insert iteratively at the first feasible position in $s$
    if no feasible insertion is possible then
        Create a new tour in $s$
        Pick a station as origin randomly
    end if
end for
return $s$
```

4.5 Station assignment model

The following section defines the IP used in the ALNSM to determine the optimal assignment of charging stations of a tour. Let $\mathcal{I}$ be a set of tours and $\mathcal{J}$ a set of potential charging stations. For each pair of tour i and charging station j a sailing cost TF_{ij} from charging station j to the first node in tour i and from the last node in tour i to charging station j is incurred. The binary decision variable x_{ij} determines whether tour i is assigned to charging station j. If tour i is assigned to charging station j, $x_{ij} = 1$. Otherwise, if tour i is not assigned to charging station j, $x_{ij} = 0$. To determine if charging station j is built, the binary decision variable y_j is used. If charging station j is built, $y_j = 1$, otherwise $y_j = 0$. The notation introduced is summarized in Table 4.1.

$$\underset{Z}{\text{minimize}} \quad Z = \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} TF_{ij} x_{ij} + \sum_{j \in \mathcal{J}} F_j y_j \tag{1}$$

subject to

$$\sum_{j \in \mathcal{J}} x_{ij} = 1, \qquad \forall i \in \mathcal{I}, \tag{2}$$

$$\sum_{i \in \mathcal{I}} x_{ij} \leq y_j |\mathcal{I}| \quad \forall j \in \mathcal{J}, \tag{3}$$

$$x_{ij} \in \{0,1\} \qquad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \tag{4}$$

$$y_j \in \{0,1\} \qquad \forall j \in \mathcal{J}. \tag{5}$$

Table 4.1: Notation used for the mathematical formulation of the station assignment model

Symbol	Description
$\mathcal{I}$	Set of tours
$\mathcal{J}$	Set of charging stations
TF_{ij}	To-and-from costs for tour i to charging station j
F_j	Fixed costs for building charging station j
x_{ij}	Binary: If tour i is assigned to charging station j
y_j	Binary: If charging station j is built

Note: The table is subdivided as follows: sets/parameters/decision variables.

The objective (1) minimises the total costs associated with to-and-from costs and building costs of charging stations. Constraints (2) ensures that a tour is assigned only one station. A station can only be built if it is assigned to a tour given by constraints (3). Binary variables are defined by constraint (4)-(5).

Chapter 5

Computational Results

In this chapter, computational results for the ALNSM are presented. Details regarding the computational environment and the definition of the algorithm parameters can be found in Section 5.1.To determine the performance of the algorithm, several instances have been designed in Section 5.2. To evaluate the quality of solutions produced by the ALNSM, smaller instances have been created which can be solved in a reasonable time using the commercial solver Gurobi. Furthermore, the efficiency and robustness of the ALNSM algorithm is tested in Section 5.3 on a set of large instances of the multi-period LRP.

5.1 Computational Environment and Parameter Setting

The numerical experiments regarding the mathematical formulation of the multi-period LRP are run on the Geophysical Insitute's computing system named Cyclone. Cyclone is a Dell PowerEdge R740 Server configuration with Intel Xeon Gold 6140M 2.3GHz and 1.5 TB RAM running a CentOS 7 linux operating system (Geophysical Institute, 2018). The model is implemented with Gurobi 11.0.0 in a Python 3.11.5 environment, with the solver configured to utilize up to 8 threads. The ALNSM is implemented as a single thread code in the same Python environment. Numerical experiments are conducted on a Macbook Pro with an Apple M1 chip and 16 GB RAM with macOS Ventura 13.5.1. The final parameter setting is used for all numerical experiments.

To identify good parameter settings for the ALNSM algorithm, the method detailed in Ropke & Pisinger (2006) is followed. First, an initial parameter setting is found through test runs on a diverse and sufficiently large subset of large problem instances. Second, each parameter is varied while the others are fixed. The heuristic is run 10 times for each parameter setting and instance, and the deviation of the average best solution from the best found solution for the instance using the best setting is calculated. The setting with the lowest overall average deviation, $\Delta\overline{\lambda}$, across all instances is selected as the optimal setting. The subset of instances used for parameter setting are c2-100-8-6-s, c6-100-8-10-s, c10-100-10-8-s, c14-100-8-6-l, r1-100-8-6-s, r5-100-8-10-s, r9-100-10-8-s, r13-100-8-6-l, rc3-100-8-8-s, rc7-100-10-6-s, rc13-100-8-6-l and rc17-100-8-10-l.

Table 5.1 shows the final setting for the parameters and the derivation process. The parameters that were selected are highlighted in bold. The ALNSM parameters in the table are as follows: the maximum number of iterations without an improvement of the global best solution $\eta_{\text{noi}}^{\max}$, the number of iterations before the escape-algorithm is called η^{esc}, the smoothing factor r in Eq. (4.2), the maximum number of turbines which be removed by an operator $q_{\max}$, the number of turbines to remove in the escape-algorithm q_{esc}, the scoring parameters $(\sigma_1, \sigma_2, \sigma_3)$ used in Eq. (4.2). For the parameter η^{esc}, the first and second setting resulted in a tie. The larger value was chosen to allow for more intensification between each call to the escape-algorithm. The ALNSM is stopped after 12500 iterations to achieve a balanced compromise between computational time and solution quality given the programming environment. The number of iterations in each segment η^{seg} is set to 100.

5.2 Generation of Instances

To the author's knowledge there does not exist any instances for the multi-period LRP that concerns the problem setting in Section 3.1. Therefore, two new sets of instances are presented in this section. The results for the new instances are presented in Section 5.3.

Two sets of instances are created: a set of 72 large instances with 100 turbines each, and a set of 72 smaller instances with number of turbines being 5, 10 and 15. The instances are classified by their geographical distribution and belong to one of the three groups: random (r), clustered (c), and mixed (rc). The locations of turbines for instances that belong to the Random group are randomly generated according to a uniform distribution. However, a minimum distance between each turbine is enforced to reflect the real-world spacing of turbines due the turbine rotor diameter.

The locations of turbines for instances in the Clustered group are

Table 5.1: Parameter setting for the ALNSM

Parameter	Setting 1	Setting 2	Setting 3
$\eta_{\text{noi}}^{\text{max}}$	7000	**9000**	10000
$\Delta\overline{\lambda}$	0.263	0.241	0.287
η^{esc}	700	**800**	900
$\Delta\overline{\lambda}$	0.260	0.260	0.274
r	**0.3**	0.4	0.6
$\Delta\overline{\lambda}$	0.266	0.268	0.288
$q_{\max}$	0.3	0.4	**0.5**
$\Delta\overline{\lambda}$	0.463	0.357	0.311
q_{esc}	$0.75\lvert\mathcal{I}\rvert$	$0.9\lvert\mathcal{I}\rvert$	$\mathbf{1}\lvert\mathcal{I}\rvert$
$\Delta\overline{\lambda}$	0.313	0.314	0.291
$(\sigma_1, \sigma_2, \sigma_3)$	(5, 2.5, 1)	**(6, 2, 0.5)**	(5, 3, 9)
$\Delta\overline{\lambda}$	0.305	0.292	0.303

Notes: The table presents the parameter settings alongside their corresponding overall average deviations, denoted as $\Delta\overline{\lambda}$ [%]. The selected settings for each parameter are highlighted in bold.

generated in the following way. The number of clusters is chosen according to $\lceil \sqrt{n}\, \rceil$ where n is the number of turbines of the current instance. Cluster centres are picked randomly according to a uniform distribution. The locations of turbines are generated according to a normal distribution and scaled around each cluster centre, while keeping a minimum distance to other turbines.

The locations of turbines for instances in the Mixed group utilities both approaches described above. The ratio of clustered turbine locations to random locations is set to 70 %. A minimum distance between turbine locations is enforced.

The time windows are divided into short and longer planning horizons. For small instances, i.e. 5, 10 and 15 turbines, the short and long planning horizon is set to 3 time windows and 7 time windows respectively. Larger instances have planning horizons of 5 time windows and 8 time windows.

Within each group, instances differ in geographical location, number of potential charging stations, number of vessels, and number of time windows.

The locations of the potential locations for charging stations are generated from a uniform distribution. However, to make the instances more realistic the locations are generated within the smallest bounding box that covers all turbines. Furthermore, each charging station must be a minimum distance away from other charging stations and turbines.

The fleet of vessels is assumed homogeneous and the technical specifications used to determine average velocity, battery capacity and consumption rate is based on the REM Energy CSOV (Rem Offshore AS, 2021). To determine the consumption rate the propeller law is used,

$$P \propto kV^c. \tag{5.1}$$

The propeller law relates the required power P of a vessel to its speed V, where c is set to at least 3 (MAN Energy Solutions, 2018). Because of the nonlinear relationship between speed and power, even a slight decrease in speed results in a substantial decrease in power. Assuming that k is constant between speeds we have that the ratio between P_V and $P_{V_{\max}}$ is,

$$\frac{P_V}{P_{V_{\max}}} = \left(\frac{V}{V_{\max}}\right)^3 \Rightarrow P_V = P_{V_{\max}} \left(\frac{V}{V_{\max}}\right)^3 \tag{5.2}$$

However, Berthelsen & Nielsen (2021) argues that c is closer to 2 at lower speeds in (5.1). This results in that (5.1) underestimates the required power because of a shallower slope. To make modelling easier the relationship in (5.1) is assumed. Equation (5.2) and Figure A.1 is used to calculate the consumption rate R of a vessel. For simplicity, a slow steaming average velocity of 5 knots is chosen which results in the consumption rate R being 0.0167 kWh/m (See Figure A.2).

The generated instances are available at https://github.com/simentorseth/multi_period_lrp_instances/.

5.3 Results for the Multi-period LRP

In this section, the results for the model formulated in Section 3.2 are presented. The results are divided into two parts based on instance size and solution approach. For small instances, both the Gurobi optimisation solver and the ALNSM heuristic are employed to evaluate their performance and accuracy. Furthermore, for large instances, only the matheuristic is used to obtain feasible solutions within a reasonable time frame due to the computational complexity of the planning problem. Additionally, two routes for small instances are inspected to gain insight into the routing and siting capabilities of the matheuristic.

Results for instance size 5, 10, and 15 are presented in Table 5.2, Table 5.3, and Table 5.4, respectively. Following Schiffer & Walther (2017), a time limit of 7200 seconds is imposed for solving smaller instances. Therefore, optimal solutions are not guaranteed for instances with a computational time of 7200 seconds. Figure 5.1 shows the average running time in seconds for each instance size (5, 10, and 15) using Gurobi. The computational difficulty of the planning problem is apparent due to the rapid increase in computational time for an increasing number of turbines. Furthermore, Gurobi failed to find a feasible solution for 10 instances shown in Table 5.4, further emphasising the complexity of the problem. For the solutions obtained using Gurobi, the objective value λ and computational time t are stated. λ is stated as '-' for instances where Gurobi failed to find a solution. For the solutions found by the ALNSM, the following metrics are provided: the best found objective value λ^b, the percentage gap Δ^b between λ and λ^b, the average objective value out of

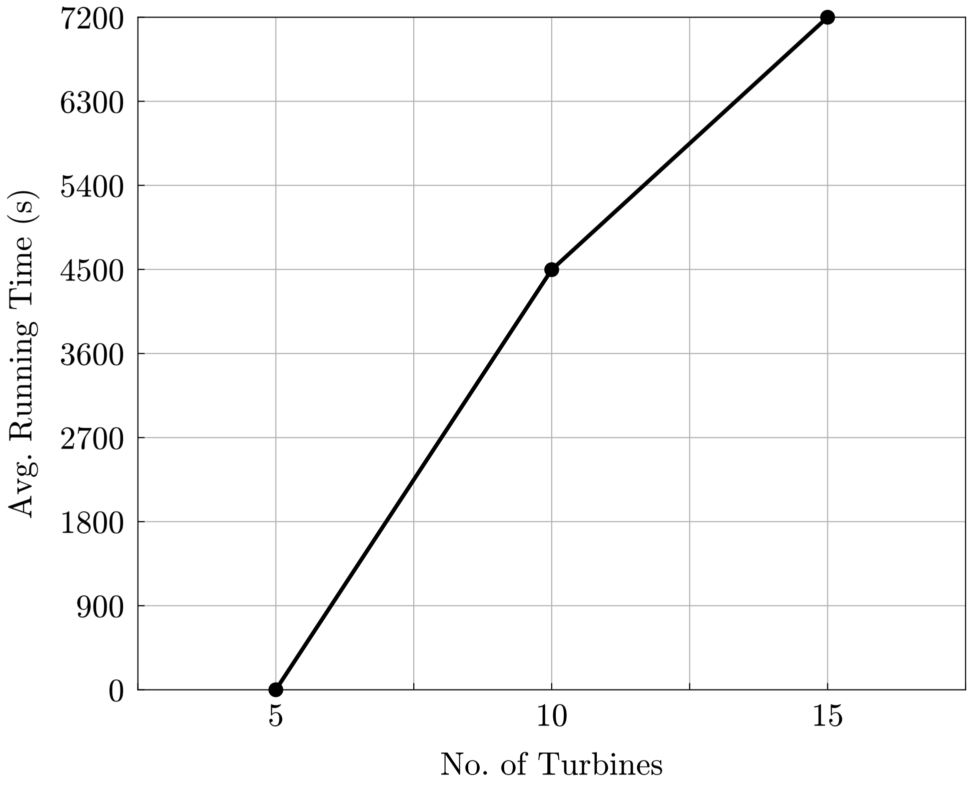


Figure 5.1: Average running time with respect to number of turbines

Average running time for each instance size solved using Gurobi. For each instance size (5, 10, and 15), 24 instances are solved, and the running time is averaged over these 24 instances.

10 runs λ^a, the percentage gap Δ^a between λ and λ^a, and the average running time t^a is stated. The gaps Δ^b and Δ^a are calculated by dividing the respective difference by λ.

Firstly, for instances with 5 turbines, the ALNSM nearly matches the performance of the mathematical model, with Δ^b at 0.27 % and Δ^a at 0.29 %. Given the small size of these instances, Gurobi quickly finds optimal solutions with an average computational time of 0.41 seconds. In comparison, the ALNSM generates feasible solutions for all instances, averaging a running time of 2.55 seconds. This difference is expected as the Gurobi engine is written in the C programming language (Gurobi Optimization LLC, 2024).

For instances with 10 turbines, there is a significant increase in computational time for solutions found using Gurobi. The ALNSM generates feasible solutions for all instances, with an average running time of 7.89 seconds. On the other hand, Gurobi's average running time is 4498 seconds, with some instances potentially not reaching optimal solutions due to the imposed time limits. Compared to instances with 5 turbines, the ALNSM performs slightly worse, with Δ^b at 1.29 % and Δ^a at 1.54 %. Despite this, the heuristic improves Gurobi's solution for instance c6-10-3-4-l, which is highlighted in bold.

Lastly, for instances with 15 turbines, Gurobi runs for 7200 seconds for all instances and fails to find feasible solutions for 10 instances. The ALNSM improves Gurobi's solutions for four instances and obtains feasible solutions for all instances, with an average computational time of 15.79 seconds. Across all instances, the gaps Δ^b and Δ^a are 0.67 % and 1.11 %, respectively.

Overall, the average performance of solutions provided by Gurobi and the ALNSM are detailed in Table 5.5. For small instances, the ALNSM performs well with a Δ^b of 0.75 % and a Δ^a of 0.96 % in a short running time. The algorithm's performance is thus validated, and results from large instances are used to evaluate the robustness of the matheuristic.

Results for large instances are presented in Table 5.6. The following metrics are provided: the best found objective value λ^b, the average objective value of 10 runs λ^a, the percentage gap between λ^b and λ^a, and the average running time t^a. To evaluate the robustness of the heuristic algorithm the gap Δ^a between the best found solution λ^b and λ^a is used. The ALNSM shows an average gap of 5.44 % over all 72 instances. Furthermore, over all instances the average of t^a is 269.02 seconds. This indicates that the matheuristic performs quite well, despite only having a small set of heuristics available.

Table 5.2: Results for instances with 5 turbines for the multi-period LRP

	Gurobi		ALNSM				
Instance	λ	t	λ^b	Δ^b	λ^a	Δ^a	t^a
c1-5-2-2-s	13456.73	0.30	13456.70	0.00	13456.70	0.00	0.90
c2-5-2-3-s	13996.50	0.45	13996.60	0.00	13996.60	0.00	0.93
c3-5-3-2-s	13706.09	0.38	13706.10	0.00	13706.10	0.00	3.93
c4-5-3-3-s	13479.58	0.61	13479.50	0.00	13479.50	0.00	3.94
c5-5-2-2-l	14474.22	0.37	14474.20	0.00	14474.20	0.00	0.93
c6-5-2-3-l	13891.05	0.48	13891.00	0.00	13903.15	0.09	0.98
c7-5-3-2-l	13831.62	0.38	13831.70	0.00	13831.70	0.00	3.94
c8-5-3-3-l	14535.65	1.05	14535.70	0.00	14535.70	0.00	2.70
r1-5-2-2-s	15190.50	0.17	15439.90	1.64	15439.90	1.64	0.97
r2-5-2-3-s	14684.85	0.44	14684.90	0.00	14684.90	0.00	0.93
r3-5-3-2-s	14158.97	0.28	14159.00	0.00	14159.00	0.00	4.42
r4-5-3-3-s	14873.41	0.38	15403.80	3.57	15403.80	3.57	4.38
r5-5-2-2-l	14481.44	0.30	14481.40	0.00	14481.40	0.00	0.92
r6-5-2-3-l	15359.46	0.34	15359.50	0.00	15359.50	0.00	1.10
r7-5-3-2-l	14867.44	0.32	14867.50	0.00	14867.50	0.00	3.58
r8-5-3-3-l	13625.42	0.63	13690.10	0.47	13690.10	0.47	4.04
rc1-5-2-2-s	13087.11	0.33	13087.00	0.00	13087.00	0.00	0.87
rc2-5-2-3-s	14034.51	0.37	14034.50	0.00	14034.50	0.00	0.91
rc3-5-3-2-s	14942.71	0.21	14942.70	0.00	14942.70	0.00	4.39
rc4-5-3-3-s	14179.03	0.50	14186.10	0.05	14253.42	0.52	4.85
rc5-5-2-2-l	15375.42	0.36	15375.50	0.00	15375.50	0.00	0.93
rc6-5-2-3-l	14233.50	0.36	14330.60	0.68	14330.60	0.68	0.94
rc7-5-3-2-l	15188.62	0.39	15188.60	0.00	15188.60	0.00	4.36
rc8-5-3-3-l	14544.75	0.52	14544.70	0.00	14544.70	0.00	5.27
Average		0.41		0.27		0.29	2.55

Notes: λ, objective value found by Gurobi; t (s), running time using Gurobi; λ^b, best objective value found by the ALNSM; Δ^b (%), gap between λ and λ^b; λ^a, average objective value of 10 runs; Δ^a (%), gap between λ and λ^a; t^a (s), average running time of 10 runs using the ALNSM.

Although the matheuristic generally handles most instances well, there are some cases where the algorithm shows no improvement. Instances with Δ^a values of 0.00 did not improve the initial solution, which occurs in only eight out of 72 instances. This may be due to an insufficient number of iterations in the search or a limited set of operators for the routing variables.

Table 5.3: Results for instances with 10 turbines for the multi-period LRP

	Gurobi		ALNSM				
Instance	λ	t	λ^b	Δ^b	λ^a	Δ^a	t^a
c1-10-3-3-s	16006.04	1401	16294.20	1.80	16294.20	1.80	8.01
c2-10-3-4-s	15133.27	7200	15213.30	0.53	15246.49	0.75	7.49
c3-10-4-3-s	15356.39	3823	15360.20	0.02	15360.20	0.02	8.57
c4-10-4-4-s	18345.63	1814	19098.60	4.10	19098.60	4.10	9.37
c5-10-3-3-l	14826.90	1222	14826.70	0.00	14826.70	0.00	5.92
c6-10-3-4-l	18266.14	7200	**18258.90**	**-0.04**	18258.90	**-0.04**	7.90
c7-10-4-3-l	14781.72	4008	14817.50	0.24	14817.50	0.24	8.55
c8-10-4-4-l	18143.33	7200	18156.00	0.07	18169.12	0.14	10.16
r1-10-3-3-s	18332.54	719.91	18661.50	1.79	18678.04	1.88	10.06
r2-10-3-4-s	17898.23	7200	17953.30	0.31	17953.30	0.31	7.25
r3-10-4-3-s	17214.96	132.24	18102.40	5.16	18102.40	5.16	9.15
r4-10-4-4-s	19649.82	7200	19649.60	0.00	19649.60	0.00	7.31
r5-10-3-3-l	18430.88	4096	18610.90	0.98	18706.69	1.50	8.41
r6-10-3-4-l	17701.57	2302	17701.30	0.00	17701.30	0.00	6.24
r7-10-4-3-l	16314.37	6922	16316.30	0.01	16316.30	0.01	8.39
r8-10-4-4-l	18144.98	7200	18163.20	0.10	18163.20	0.10	9.10
rc1-10-3-3-s	16254.02	2913	16262.30	0.05	16265.81	0.07	5.74
rc2-10-3-4-s	17529.87	5104	18757.40	7.00	18758.90	7.01	6.82
rc3-10-4-3-s	17464.57	1404	17464.60	0.00	17464.60	0.00	8.71
rc4-10-4-4-s	16600.28	2916	17959.10	8.19	18071.48	8.86	8.61
rc5-10-3-3-l	16647.10	6272	16647.00	0.00	16647.00	0.00	6.43
rc6-10-3-4-l	17956.94	7200	18085.50	0.72	18844.91	4.94	7.13
rc7-10-4-3-l	16288.88	5293	16288.70	0.00	16288.70	0.00	7.04
rc8-10-4-4-l	16789.91	7200	16790.00	0.00	16790.00	0.00	7.05
Average		4498		1.29		1.54	7.89

Notes: λ, objective value found by Gurobi; t (s), running time using Gurobi; λ^b, best objective value found by the ALNSM; Δ^b (%), gap between λ and λ^b; λ^a, average objective value of 10 runs; Δ^a (%), gap between λ and λ^a; t^a (s), average running time of 10 runs using the ALNSM.

Table 5.4: Results for instances with 15 turbines for the multi-period LRP

	Gurobi		ALNSM				
Instance	λ	t	λ^b	Δ^b	λ^a	Δ^a	t^a
c1-15-4-4-s	-	7200	18696.90	-	18718.67	-	11.85
c2-15-4-5-s	17514.90	7200	17524.40	0.05	17536.88	0.13	15.52
c3-15-5-4-s	16661.17	7200	**16026.40**	**-3.81**	16041.73	**-3.72**	16.15
c4-15-5-5-s	18569.10	7200	19315.00	4.02	19380.56	4.37	18.88
c5-15-4-4-l	18945.90	7200	18986.40	0.21	19012.68	0.35	15.17
c6-15-4-5-l	20215.54	7200	20353.20	0.68	20394.78	0.89	14.26
c7-15-5-4-l	16568.53	7200	**16438.00**	**-0.79**	16490.24	**-0.47**	17.28
c8-15-5-5-l	20311.79	7200	20406.90	0.47	20467.33	0.77	18.60
r1-15-4-4-s	-	7200	22822.10	-	22822.10	-	16.06
r2-15-4-5-s	-	7200	25397.50	-	25397.50	-	11.47
r3-15-5-4-s	-	7200	19916.20	-	19916.34	-	18.80
r4-15-5-5-s	21757.76	7200	21827.60	0.32	21879.14	0.56	19.32
r5-15-4-4-l	-	7200	21361.40	-	21485.44	-	15.80
r6-15-4-5-l	22142.08	7200	22316.00	0.79	22376.70	1.06	16.36
r7-15-5-4-l	-	7200	21216.30	-	21278.69	-	12.83
r8-15-5-5-l	20172.16	7200	20720.70	2.72	21360.43	5.89	17.39
rc1-15-4-4-s	-	7200	22033.00	-	22138.19	-	16.57
rc2-15-4-5-s	-	7200	22509.00	-	22509.00	-	10.26
rc3-15-5-4-s	-	7200	19200.20	-	19235.28	-	14.43
rc4-15-5-5-s	20248.24	7200	21777.20	7.55	21856.06	7.94	19.50
rc5-15-4-4-l	19654.07	7200	**19226.70**	**-2.17**	19258.75	**-2.01**	15.12
rc6-15-4-5-l	19787.43	7200	20100.00	1.58	20124.12	1.70	13.47
rc7-15-5-4-l	-	7200	19961.90	-	20221.16	-	16.88
rc8-15-5-5-l	20949.99	7200	**20474.30**	**-2.27**	20561.70	**-1.85**	16.98
Average		7200		0.67		1.11	15.79

Notes: λ, objective value found by Gurobi; t (s), running time using Gurobi; λ^b, best objective value found by the ALNSM; Δ^b (%), gap between λ and λ^b; λ^a, average objective value of 10 runs; Δ^a (%), gap between λ and λ^a; t^a (s), average running time of 10 runs using the ALNSM. Better solutions found by the ALNSM are highlighted in bold. λ is stated as '-' for instances where Gurobi failed to find a solution.

Table 5.5: Average results for small instances for the multi-period LRP

	Gurobi	ALNSM		
	t	Δ^b	Δ^a	t^a
Overall Average	3899.34	0.75	0.96	8.74

Notes: t (s), running time using Gurobi; Δ^b (%), gap between λ and λ^b; Δ^a (%), gap between λ and λ^a; t^a (s), average running time of 10 runs using the ALNSM.

Table 5.6: Results for instances with 100 turbines for the multi-period LRP

Instance	λ^b	λ^a	Δ^a	t^a	Instance	λ^b	λ^a	Δ^a	t^a
c1-100-8-6-s	48464.2	74719.23	54.17	217.74	r13-100-8-6-l	53191.1	54985.87	3.37	267.01
c2-100-8-6-s	88449.9	88449.9	0.00	190.81	r14-100-8-6-l	54099.7	55051.85	1.76	188.65
c3-100-8-8-s	51746.8	58615.6	13.27	230.97	r15-100-8-8-l	56578.9	57606.68	1.82	217.98
c4-100-8-8-s	47699.2	48294.45	1.25	235.78	r16-100-8-8-l	58566.5	59546.19	1.67	263.53
c5-100-8-10-s	49356.9	50226.75	1.76	197.46	r17-100-8-10-l	55446.6	56709.39	2.28	229.72
c6-100-8-10-s	51580.8	52727.02	2.22	168.85	r18-100-8-10-l	60141	62038.43	3.15	221.74
c7-100-10-6-s	43561.7	43881.93	0.74	287.19	r19-100-10-6-l	54733.2	55796.33	1.94	435.21
c8-100-10-6-s	105557	105557	0.00	250.52	r20-100-10-6-l	50892.1	52203.52	2.58	295.59
c9-100-10-8-s	49160.2	49790.17	1.28	293.53	r21-100-10-8-l	60152.9	61656.43	2.50	465.43
c10-100-10-8-s	51781.9	52550.22	1.48	372.63	r22-100-10-8-l	55915.4	56919.99	1.80	342.37
c11-100-10-10-s	49531.4	50467.45	1.89	310.11	r23-100-10-10-l	60724.8	69112.12	13.81	452.35
c12-100-10-10-s	46181.1	47026.44	1.83	285.07	r24-100-10-10-l	62357.9	63333.53	1.56	465.10
c13-100-8-6-l	46173.9	47819.25	3.56	219.95	rc1-100-8-6-s	45105.6	45749.21	1.43	198.44
c14-100-8-6-l	42164	42989.73	1.96	204.52	rc2-100-8-6-s	117682.2	117682.2	0.00	163.97
c15-100-8-8-l	50696	51603.84	1.79	209.76	rc3-100-8-8-s	58338.2	59437.27	1.88	253.10
c16-100-8-8-l	44661	45055.33	0.88	217.70	rc4-100-8-8-s	51974.5	52524.83	1.06	175.09
c17-100-8-10-l	47954.6	49540.32	3.31	205.38	rc5-100-8-10-s	57562	58986.78	2.48	256.49
c18-100-8-10-l	46862.3	47466.28	1.29	223.07	rc6-100-8-10-s	56407.1	57675.61	2.25	257.56
c19-100-10-6-l	36163.3	37040.85	2.43	258.51	rc7-100-10-6-s	118716	118716	0.00	389.92
c20-100-10-6-l	42779.1	43394.58	1.44	267.03	rc8-100-10-6-s	45766.9	109900.44	140.13	267.43
c21-100-10-8-l	54433.7	55497.58	1.95	365.81	rc9-100-10-8-s	50036.1	50753.43	1.43	312.69
c22-100-10-8-l	48906.3	49683.58	1.59	370.27	rc10-100-10-8-s	55066.4	55670.97	1.10	389.12
c23-100-10-10-l	44221.9	44999.14	1.76	262.49	rc11-100-10-10-s	59768.3	60739.3	1.62	432.71
c24-100-10-10-l	46351	47284.12	2.01	281.19	rc12-100-10-10-s	51611.5	52545	1.81	267.11
r1-100-8-6-s	116804.1	116804.1	0.00	190.75	rc13-100-8-6-l	50081.8	50842.5	1.52	148.01
r2-100-8-6-s	116101.6	116101.6	0.00	185.97	rc14-100-8-6-l	51291.5	51987.47	1.36	209.14
r3-100-8-8-s	54301.9	55585.57	2.36	180.46	rc15-100-8-8-l	52958.1	53651.88	1.31	165.72
r4-100-8-8-s	55070.6	80036.41	45.33	227.89	rc16-100-8-8-l	52293.4	52975.72	1.30	207.27
r5-100-8-10-s	56409.1	57693.17	2.28	172.68	rc17-100-8-10-l	55291.6	56288.75	1.80	210.92
r6-100-8-10-s	58183.8	60203.95	3.47	255.13	rc18-100-8-10-l	51823.5	52902.78	2.08	193.54
r7-100-10-6-s	135397.3	135397.3	0.00	275.81	rc19-100-10-6-l	50221.3	51280.78	2.11	305.69
r8-100-10-6-s	124826.8	124826.8	0.00	321.00	rc20-100-10-6-l	48994.8	49992.73	2.04	334.28
r9-100-10-8-s	54931.3	56063.08	2.06	311.99	rc21-100-10-8-l	54138.4	55050.53	1.68	284.89
r10-100-10-8-s	57021.8	65754.79	15.32	363.24	rc22-100-10-8-l	54743.5	55663.51	1.68	331.67
r11-100-10-10-s	56821.5	57489.63	1.18	320.79	rc23-100-10-10-l	52222.8	53048.58	1.58	273.05
r12-100-10-10-s	57774.3	59105.04	2.30	288.77	rc24-100-10-10-l	53131.4	54001.08	1.64	275.86
Average								5.44	269.02

Notes: λ^b, best objective value found by the ALNSM; λ^a, average objective value of 10 runs; Δ^a (%), gap between λ^b and λ^a; t^a (s), average running time of 10 runs using the ALNSM.

Figure 5.2 shows the running time of the ALNSM with respect to the problem complexity. The problem complexity is defined as the product of the number of turbines, the number of charging stations, the number of vessels and the number of time windows. The geographical distributions 'Clustered', 'Ramdom', 'Mixed' of the large instances are plotted. For each problem complexity and distribution, average running times are aggregated by average for each geographical distribution. The average running times show significant variability depending on the problem characteristics. However, there is only a slight upward trend in running times across all distributions. Most solutions fall within the range of 180 to 360 seconds, indicating that while the overall computational time performance varies, it remains relatively consistent.

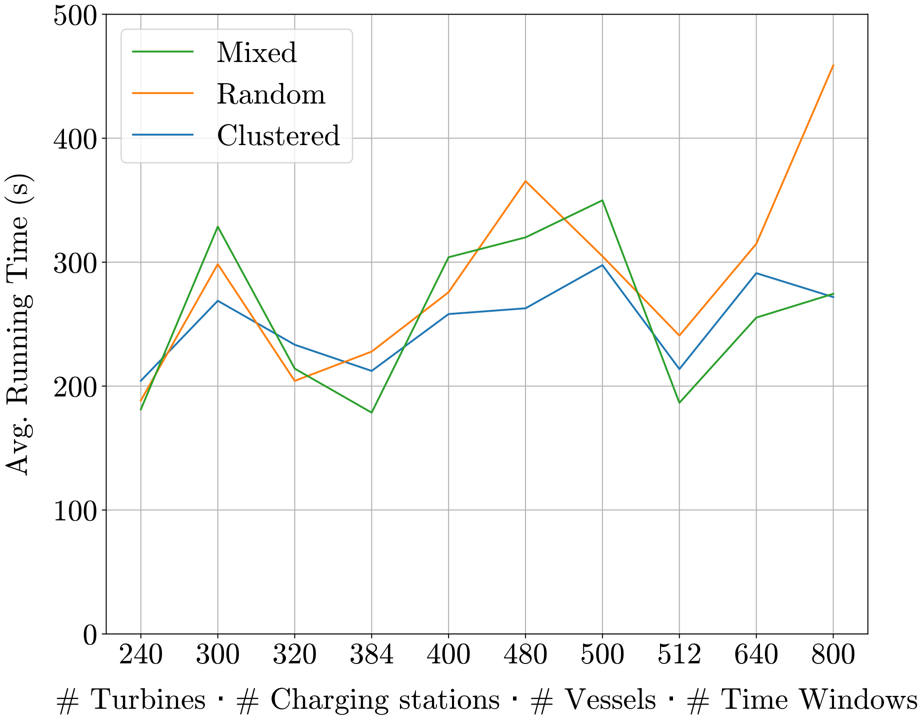


Figure 5.2: Average running time with respect to problem complexity

Average running times for the geographical distribution of large instances. For each problem complexity, the average running time is aggregated by average for each distribution 'Clustered', 'Ramdom', and 'Mixed', respectively.

Solution visualisations are provided for the instances c2-5-2-3-s and r3-10-4-3-s to examine the routing and siting capabilities of the ALNSM.

Figure 5.3 and Figure 5.4 present the optimal solution for instance c2-5-2-3-s found by using Gurobi and the ALNSM, respectively. For the instance r3-10-4-3-s, the solution found by Gurobi is shown Figure 5.5 and the solution generated by the ALNSM is displayed in Figure 5.6. Additional visualisations of solutions for remaining instance sizes and distributions are provided in Appendix B.

Since both solution approaches obtain an optimal solution for instance c2-5-2-3-s, the only difference lies in the routing direction of the vessels. In Figure 5.3, vessel routes are determined in a clockwise manner. Conversely, in Figure 5.4 routes are oriented anticlockwise. The routing direction of the vessels does not affect the solution, as both clockwise and anticlockwise routes cover the same set of nodes and have identical total distance and time requirements.

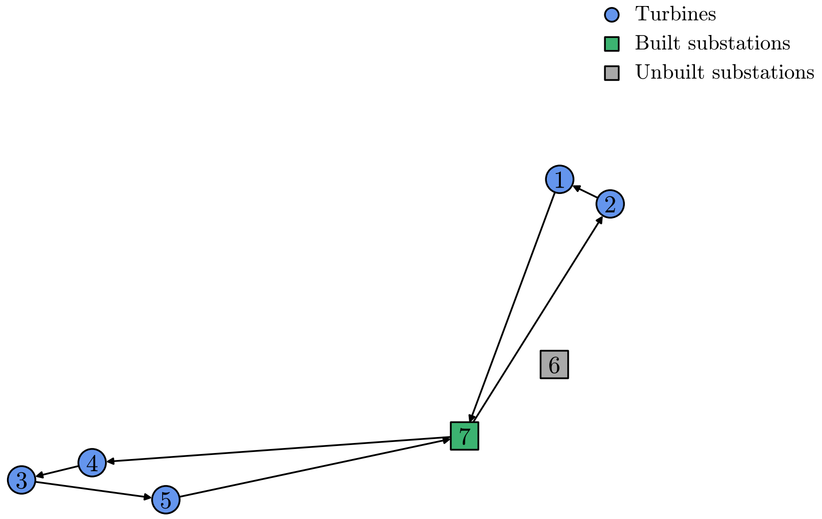


Figure 5.3: Solution for instance c2-5-2-3-s solved by Gurobi

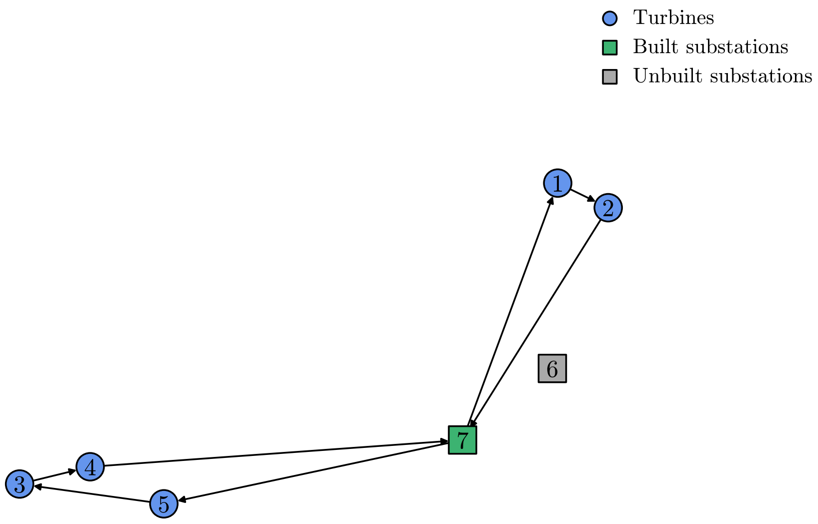


Figure 5.4: Solution for instance c2-5-2-3-s solved by the ALNSM

For instance r3-10-4-3-s, the ALNSM does not find the optimal solution and shows a 5.16 % gap between its best solution and the one found by Gurobi. This difference is clear to see in Figure 5.6, where turbine 4 is visited as a sub-tour and the trip visiting turbines 3, 2, and 5 is longer than the solution found by Gurobi. However, the ALNSM effectively handles sub-tour elimination in the other trips. Similar to instance c2-5-2-3-s, the routing directions differ in the solutions generated with Gurobi routing clockwise and the ALNSM routing anticlockwise.

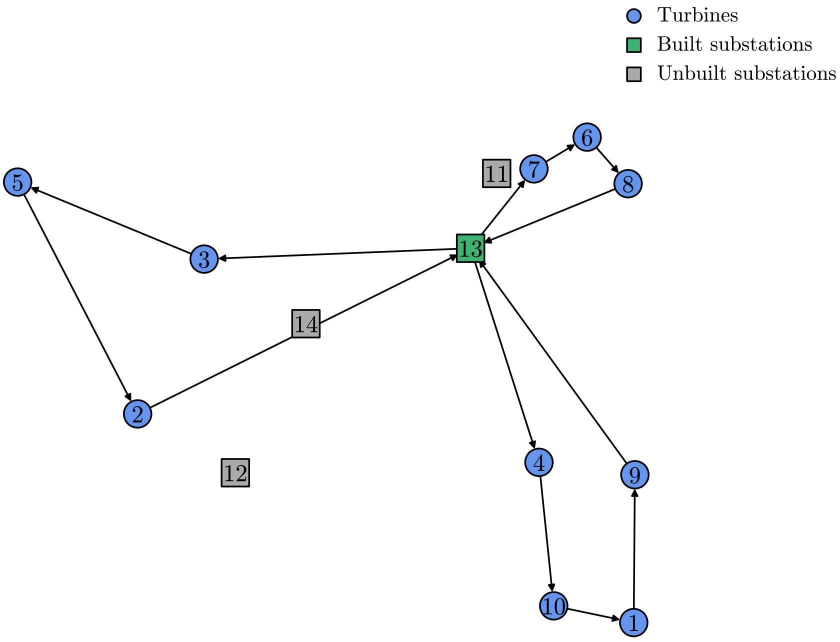


Figure 5.5: Solution for instance r3-10-4-3-s solved by Gurobi

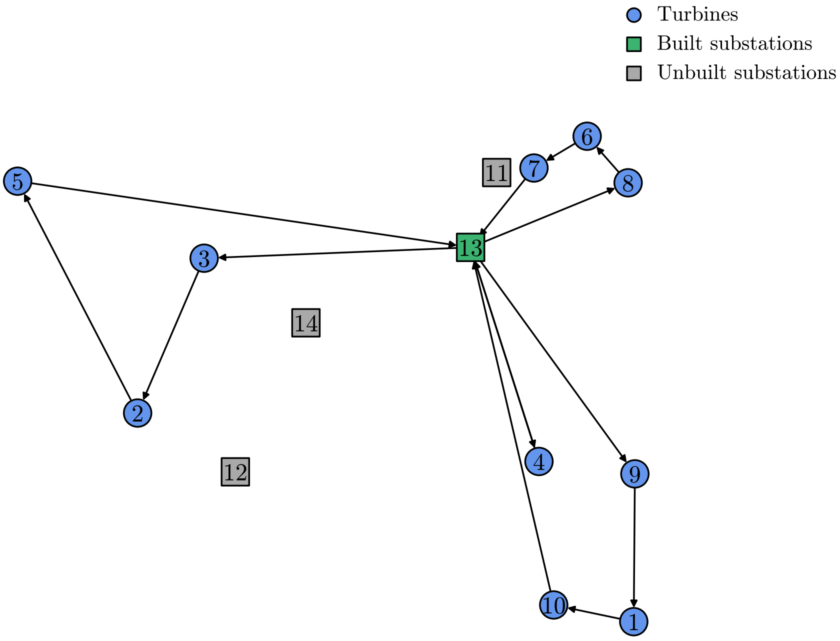


Figure 5.6: Solution for instance r3-10-4-3-s solved by the ALNSM

Chapter 6

Conclusion

In this thesis, a multi-period LRP in the offshore wind industry is proposed which considers the routing of electric maintenance vessels and the placement of offshore charging units. The aim is to minimise the total cost while meeting maintenance demands of OWTs. Offshore charging units allows a homogeneous fleet of maintenance vessels to be operational without returning to an onshore base to recharge. A mathematical formulation for the proposed planning problem is presented. Due to the computational difficulty of problem, a matheuristic algorithm is developed to efficiently solve larger instances. The presented heuristic uses common metaheuristic operators iteratively within an ALNS framework to determine the routing variables and uses a commercial solver to determine the placement of charging stations with respect to the resulting routing variables. To ensure that the matheuristic produces good solutions, a set of 72 smaller instances are generated. Additionally, 72 larger instances are generated to evaluate the robustness and stability of the proposed algorithm. These new instances cover three different geographical distributions such that the algorithm can handle various scenarios effectively. Despite a very limited set of available heuristics in the ALNS framework, the solution approach achieves good results in a short time span for both small and large instances. This indicates promising performance, and the algorithm could potentially find even better solutions with a more varied set of heuristics. Notably, the solution approach improved upon the solutions found by the commercial solver for small instances.

An analysis of the matheuristic's ability to effectively route vessels and site charging stations is performed. In most cases, the algorithm is able to eliminate sub-tours effectively and find good solutions for small instances.

In conclusion, this thesis contributes to the field of location routing by considering offshore logistics in offshore wind farms. A multi-period LRP is proposed and solved for 144 newly generated instances by a matheuristic algorithm which provide insights for siting charging infrastructure to offshore wind farm owners.

6.1 Future research

The limitations of the proposed ALNSM algorithm include the lack of implemented operators. Expanding the set of available heuristics would enhance the exploration of the solution space, leading to greater diversification and intensification during the search process. Both inter-route and intra-route specific heuristics could be considered. Moreover, taking advantage of the specific characteristics of the multi-period LRP problem is likely to improve the performance of the solution approach. By incorporating problem-specific operators, it may be possible to discover better solutions than those presented in this thesis.

A statistical analysis of the various components of the algorithm could be conducted to assess the significance of each element. This process would involve systematically removing individual components and observing their impact on overall performance.

Furthermore, evaluating the algorithm's performance on real-world instances would be valuable. Wind farm owners would gain managerial insights from real-world applications, potentially driving further development of electric maintenance vessels and charging station technology. Conducting case studies would provide practical insights into the effectiveness and robustness of the matheuristic, and into its performance under real-world conditions. Such studies would also help identify any potential limitations or areas for improvement that may not be evident in theoretical environments.

Bibliography

Amiri, M., Amin, S. H., & Tavakkoli-Moghaddam, R. (2019). A Lagrangean decomposition approach for a novel two-echelon node-based location-routing problem in an offshore oil and gas supply chain. *Transportation Research Part E: Logistics and Transportation Review*, *128*, 96–114. URL: https://linkinghub.elsevier.com/retrieve/pii/S1366554518313668. doi:10.1016/j.tre.2019.05.014.

Amiri, M., Sadjadi, S., Tavakkoli-Moghaddam, R., & Jabbarzadeh, A. (2018). Optimal fleet composition and mix periodic location-routing problem with time windows in an offshore oil and gas industry: A case study of National Iranian Oil Company. *Scientia Iranica*, *0*, 0–0. URL: http://scientiairanica.sharif.edu/article_20412.html. doi:10.24200/sci.2018.20412.

Berthelsen, F. H., & Nielsen, U. D. (2021). Prediction of ships' speed-power relationship at speed intervals below the design speed. *Transportation Research Part D: Transport and Environment*, *99*, 102996. URL: https://linkinghub.elsevier.com/retrieve/pii/S1361920921002947. doi:10.1016/j.trd.2021.102996.

Dantzig, G. B., & Ramser, J. H. (1959). The Truck Dispatching Problem. *Management Science*, *6*, 80–91. URL: https://pubsonline.informs.org/doi/10.1287/mnsc.6.1.80. doi:10.1287/mnsc.6.1.80.

Drexl, M., & Schneider, M. (2015). A survey of variants and extensions of the location-routing problem. *European Journal of Operational Research*, *241*, 283–308. URL: https://linkinghub.elsevier.com/retrieve/pii/S0377221714006651. doi:10.1016/j.ejor.2014.08.030.

Dueck, G. (1993). New Optimization Heuristics. *Journal of Computational Physics*, *104*, 86–92. URL: https://linkinghub.elsevier.com/retrieve/pii/S0021999183710107. doi:10.1006/jcph.1993.1010.

European Commission (2019). COMMUNICATION FROM THE COMMISSION TO THE EUROPEAN PARLIAMENT, THE EUROPEAN COUNCIL, THE COUNCIL, THE EUROPEAN ECONOMIC AND SOCIAL COMMITTEE AND THE COMMITTEE OF THE REGIONS: The European Green Deal. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52019DC0640.

European Commission (2020). COMMUNICATION FROM THE COMMISSION TO THE EUROPEAN PARLIAMENT, THE COUNCIL, THE EUROPEAN ECONOMIC AND SOCIAL COMMITTEE AND THE COMMITTEE OF THE REGIONS An EU Strategy to harness the potential of offshore renewable energy for a climate neutral future. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=COM:2020:741:FIN&qid=1605792629666.

Francis, P. M., Smilowitz, K. R., & Tzur, M. (2008). The Period Vehicle Routing Problem and its Extensions. In B. Golden, S. Raghavan, & E. Wasil (Eds.), *The Vehicle Routing Problem: Latest Advances and New Challenges* (pp. 73–102). Boston, MA: Springer US volume 43. URL: http://link.springer.com/10.1007/978-0-387-77778-8_4. doi:10.1007/978-0-387-77778-8_4 iSSN: 1387-666X Series Title: Operations Research/Computer Science Interfaces.

Geophysical Institute (2018). The new cyclone: A major update on the GFI computing infrastructure. URL: https://gfi.wiki.uib.no/index.php/The_GFI_computing_system.

Gurobi Optimization LLC (2024). Gurobi Optimizer Reference Manual. URL: https://www.gurobi.com/documentation/11.0/refman/index.html.

Halvorsen-Weare, E. E., Fagerholt, K., Nonås, L. M., & Asbjørnslett, B. E. (2012). Optimal fleet composition and periodic routing of offshore supply vessels. *European Journal of Operational Research*, *223*, 508–517. URL: https://linkinghub.elsevier.com/retrieve/pii/S0377221712004729. doi:10.1016/j.ejor.2012.06.017.

Heidari, N., & Hemmati, A. (2023). An ALNS-based matheuristic algorithm for a multi-product many-to-many maritime inventory routing problem. *Computational Management Science*, *20*, 44. URL: https://link.springer.com/10.1007/s10287-023-00478-8. doi:10.1007/s10287-023-00478-8.

Hemmati, A., & Hvattum, L. M. (2017). Evaluating the importance of randomization in adaptive large neighborhood search. *International Transactions in Operational Research*, *24*, 929–942. URL: https://onlinelibrary.wiley.com/doi/10.1111/itor.12273. doi:10.1111/itor.12273.

Hutchinson, M., & Zhao, F. (2023). Global Wind Report 2023. URL: https://gwec.net/wp-content/uploads/2023/04/GWEC-2023_interactive.pdf.

International Energy Agency (2021). *Net Zero by 2050 - A Roadmap for the Global Energy Sector*. Technical Report Internatio. URL: https://iea.blob.core.windows.net/assets/deebef5d-0c34-4539-9d0c-10b13d840027/NetZeroby2050-ARoadmapfortheGlobalEnergySector_CORR.pdf.

MAN Energy Solutions (2018). Basic principles of ship propulsion. URL: https://www.man-es.com/docs/default-source/document-sync-archive/basic-principles-of-ship-propulsion-eng.pdf?sfvrsn=48fc05b5_7.

Maniezzo, V., Stützle, T., & Voß, S. (Eds.) (2010). *Matheuristics: Hybridizing Metaheuristics and Mathematical Programming* volume 10 of *Annals of Information Systems*. Boston, MA: Springer US. URL: https://link.springer.com/10.1007/978-1-4419-1306-7. doi:10.1007/978-1-4419-1306-7.

Mara, S. T. W., Kuo, R., & Asih, A. M. S. (2021). Location-routing problem: a classification of recent research. *International Transactions in Operational Research*, *28*, 2941–2983. URL: https://onlinelibrary.wiley.com/doi/10.1111/itor.12950. doi:10.1111/itor.12950.

Maranzana, F. E. (1964). On the Location of Supply Points to Minimize Transport Costs. *Journal of the Operational Research Society*, *15*, 261–270. URL: https://www.tandfonline.com/doi/full/10.1057/jors.1964.47. doi:10.1057/jors.1964.47.

Nagy, G., & Salhi, S. (2007). Location-routing: Issues, models and methods. *European Journal of Operational Research*, *177*, 649–672. URL: https://linkinghub.elsevier.com/retrieve/pii/S0377221706002670. doi:10.1016/j.ejor.2006.04.004.

Prodhon, C. (2008). A Metaheuristic for the Periodic Location-Routing Problem. In J. Kalcsics, & S. Nickel (Eds.), *Operations Research Proceedings 2007* (pp. 159–164). Berlin, Heidelberg: Springer Berlin

Heidelberg volume 2007. URL: http://link.springer.com/10.1007/978-3-540-77903-2_25. doi:10.1007/978-3-540-77903-2_25 series Title: Operations Research Proceedings.

Prodhon, C., & Prins, C. (2014). A survey of recent research on location-routing problems. *European Journal of Operational Research*, *238*, 1–17. URL: https://linkinghub.elsevier.com/retrieve/pii/S0377221714000071. doi:10.1016/j.ejor.2014.01.005.

Puchinger, J., & Raidl, G. R. (2005). Combining Metaheuristics and Exact Algorithms in Combinatorial Optimization: A Survey and Classification. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, J. Mira, & J. R. Álvarez (Eds.), *Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach* (pp. 41–53). Berlin, Heidelberg: Springer Berlin Heidelberg volume 3562. URL: http://link.springer.com/10.1007/11499305_5. doi:10.1007/11499305_5 series Title: Lecture Notes in Computer Science.

Raidl, G. R. (2006). A Unified View on Hybrid Metaheuristics. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, F. Almeida, M. J. Blesa Aguilera, C. Blum, J. M. Moreno Vega, M. Pérez Pérez, A. Roli, & M. Sampels (Eds.), *Hybrid Metaheuristics* (pp. 1–12). Berlin, Heidelberg: Springer Berlin Heidelberg volume 4030. URL: http://link.springer.com/10.1007/11890584_1. doi:10.1007/11890584_1 series Title: Lecture Notes in Computer Science.

Rem Offshore AS (2021). Rem Energy - REM Offshore. URL: http://www.remoffshore.no/fleet/rem-energy/.

Ren, Z., Verma, A. S., Li, Y., Teuwen, J. J., & Jiang, Z. (2021). Offshore wind turbine operations and maintenance: A state-of-the-art review. *Renewable and Sustainable Energy Reviews*, *144*, 110886. URL: https://linkinghub.elsevier.com/retrieve/pii/S1364032121001805. doi:10.1016/j.rser.2021.110886.

Robert E. Bixby (2012). A brief history of linear and mixed-integer programming computation. *Documenta Mathematica*, *Extra Volume*

ISMP, 107–121. URL: https://www.emis.de/journals/DMJDMV/vol-ismp/25_bixby-robert.pdf.

Ropke, S., & Pisinger, D. (2006). An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. *Transportation Science*, *40*, 455–472. URL: https://pubsonline.informs.org/doi/10.1287/trsc.1050.0135. doi:10.1287/trsc.1050.0135.

Salhi, S., & Nagy, G. (1999). Consistency and Robustness in Location-Routing. *Studies in Locational Analysis*, (pp. 3–19).

Salhi, S., & Rand, G. K. (1989). The effect of ignoring routes when locating depots. *European Journal of Operational Research*, *39*, 150–156. URL: https://linkinghub.elsevier.com/retrieve/pii/0377221789901884. doi:10.1016/0377-2217(89)90188-4.

Santini, A., Ropke, S., & Hvattum, L. M. (2018). A comparison of acceptance criteria for the adaptive large neighbourhood search metaheuristic. *Journal of Heuristics*, *24*, 783–815. URL: http://link.springer.com/10.1007/s10732-018-9377-x. doi:10.1007/s10732-018-9377-x.

Sarhani, M., Voß, S., & Jovanovic, R. (2023). Initialization of metaheuristics: comprehensive review, critical analysis, and research directions. *International Transactions in Operational Research*, *30*, 3361–3397. URL: https://onlinelibrary.wiley.com/doi/10.1111/itor.13237. doi:10.1111/itor.13237.

Schiffer, M., & Walther, G. (2017). The electric location routing problem with time windows and partial recharging. *European Journal of Operational Research*, *260*, 995–1013. URL: https://linkinghub.elsevier.com/retrieve/pii/S0377221717300346. doi:10.1016/j.ejor.2017.01.011.

Schiffer, M., & Walther, G. (2018). An Adaptive Large Neighborhood Search for the Location-routing Problem with Intra-route Facilities. *Transportation Science*, *52*, 331–352. URL: https://pubsonline.informs.org/doi/10.1287/trsc.2017.0746. doi:10.1287/trsc.2017.0746.

Schneider, M., Stenger, A., & Goeke, D. (2014). The Electric Vehicle-Routing Problem with Time Windows and Recharging Stations. *Transportation Science*, *48*, 500–520. URL: https://pubsonline.informs.org/doi/10.1287/trsc.2013.0490. doi:10.1287/trsc.2013.0490.

Sean Luke (2013). *Essentials of Metaheuristics*. (2nd ed.). Lulu. URL: http://cs.gmu.edu/$\sim$sean/book/metaheuristics/.

Shaw, P. (1998). Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems. In G. Goos, J. Hartmanis, J. Van Leeuwen, M. Maher, & J.-F. Puget (Eds.), *Principles and Practice of Constraint Programming — CP98* (pp. 417–431). Berlin, Heidelberg: Springer Berlin Heidelberg volume 1520. URL: http://link.springer.com/10.1007/3-540-49481-2_30. doi:10.1007/3-540-49481-2_30 series Title: Lecture Notes in Computer Science.

United Nations Department of Economic and Social Affairs (2023). *Synergy Solutions for a World in Crisis: Tackling Climate and SDG Action Together: Report on Strengthening the Evidence Base - First Edition 2023*. United Nations. URL: https://www.un-ilibrary.org/content/books/9789213585238. doi:10.18356/9789213585238.

VARD (2023). Leading the way in the green maritime transition. URL: https://www.vard.com/articles/the-ocean-charger-project-has-officially-started.

Von Boventer, E. (1961). The Relationship Between Transportation Costs and Location Rent in Transportation Problems. *Journal of Regional Science*, *3*, 27–40. URL: https://onlinelibrary.wiley.com/doi/10.1111/j.1467-9787.1961.tb01276.x. doi:10.1111/j.1467-9787.1961.tb01276.x.

Watson-Gandy, C., & Dohrn, P. (1973). Depot location with van salesmen — A practical approach. *Omega*, *1*, 321–329. URL: https://linkinghub.elsevier.com/retrieve/pii/0305048373901084. doi:10.1016/0305-0483(73)90108-4.

Webb, M. H. J. (1968). Cost Functions in the Location of Depots for Multiple-Delivery Journeys. *Journal of the Operational Research Society*, *19*, 311–320. URL: https://www.tandfonline.com/doi/full/10.1057/jors.1968.74. doi:10.1057/jors.1968.74.

Weber, A., & Friedrich, C. J. (1929). *Alfred Weber's theory of the location of industires.*. University of Chicago Press.

Łebkowski, A. (2020). Analysis of the Use of Electric Drive Systems for Crew Transfer Vessels Servicing Offshore Wind Farms. *Energies*, *13*, 1466. URL: https://www.mdpi.com/1996-1073/13/6/1466. doi:10.3390/en13061466.

Appendix A

Power and Consumption Rate Estimation Curves

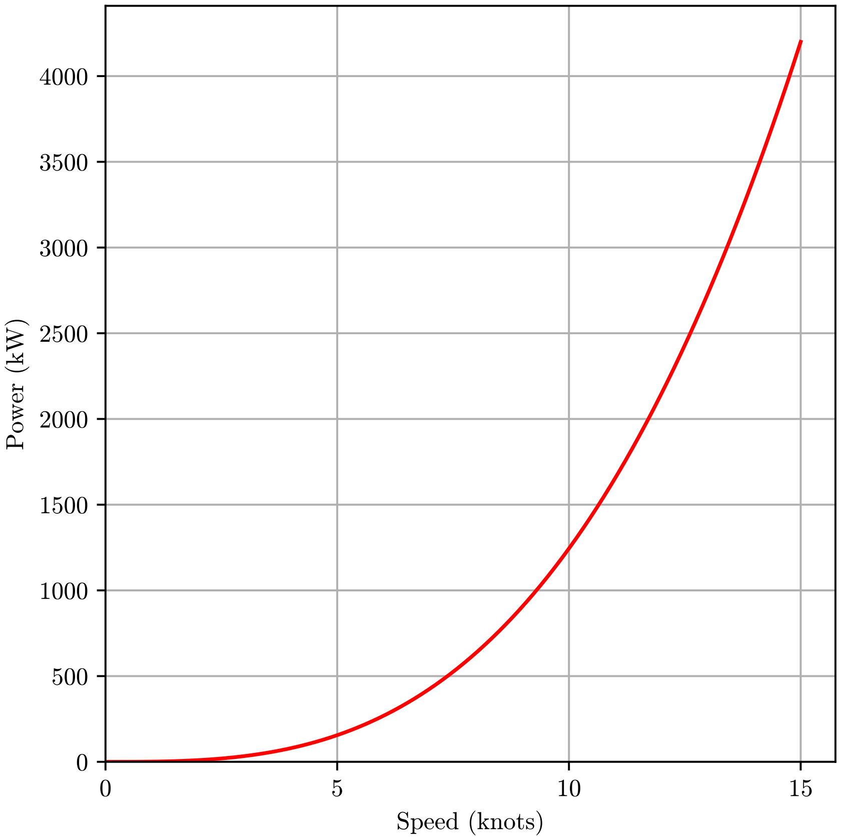


Figure A.1: Estimated power for the REM Energy CSOV

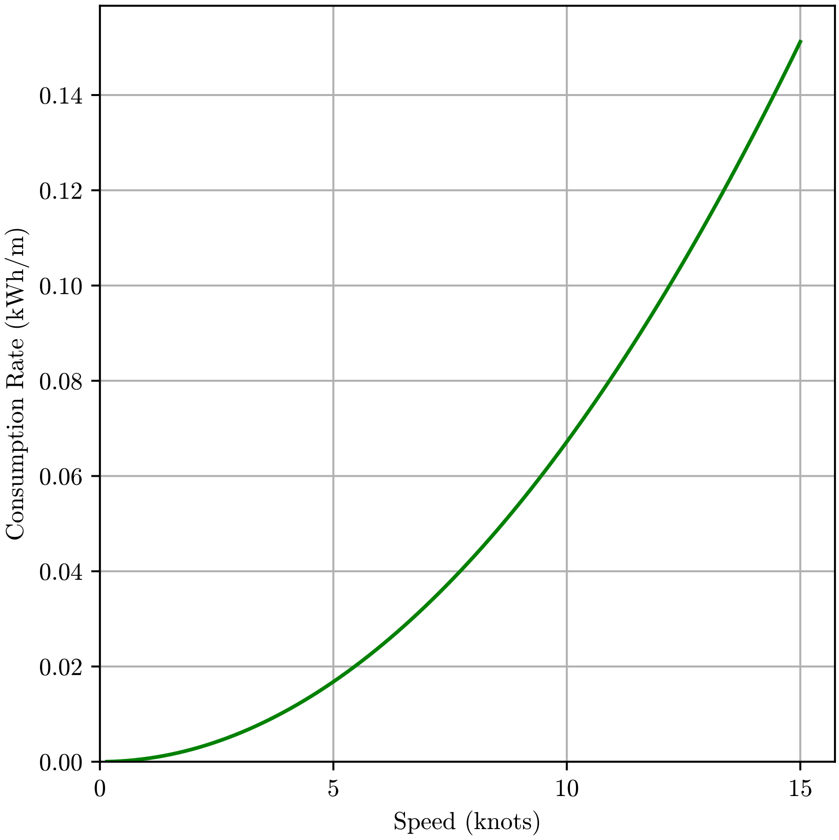


Figure A.2: Estimated consumption rate for the REM Energy CSOV

Appendix B

Route Visualisation

B.1 Additional instances with 5 Turbines

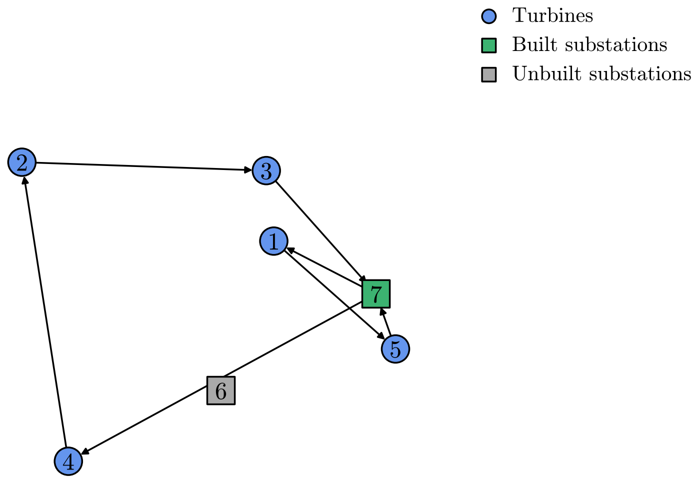


Figure B.1: Solution for instance r5-5-2-2-l solved by ALNSM

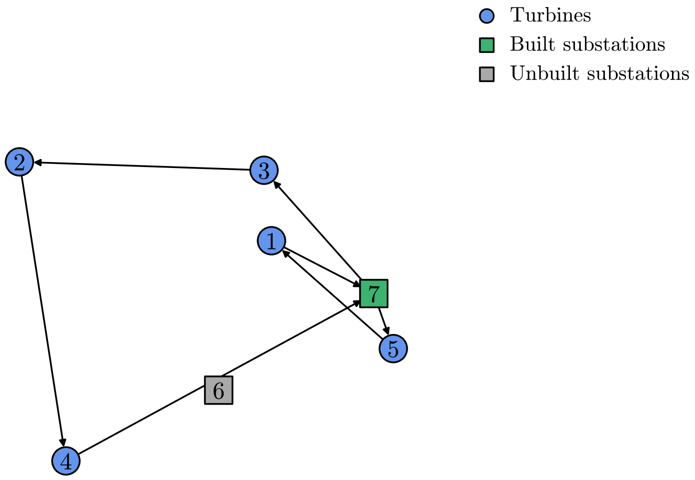


Figure B.2: Solution for instance r5-5-2-2-1 solved by Gurobi

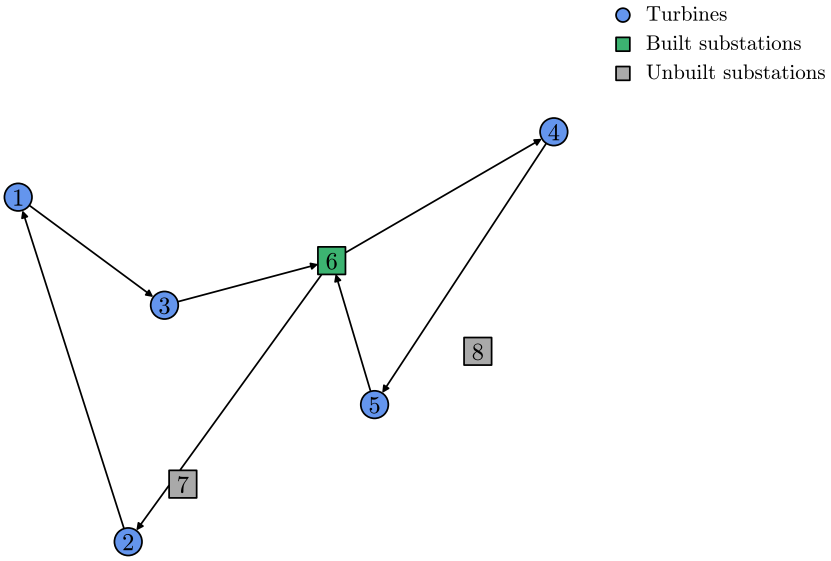


Figure B.3: Solution for instance rc7-5-3-2-1 solved by ALNSM

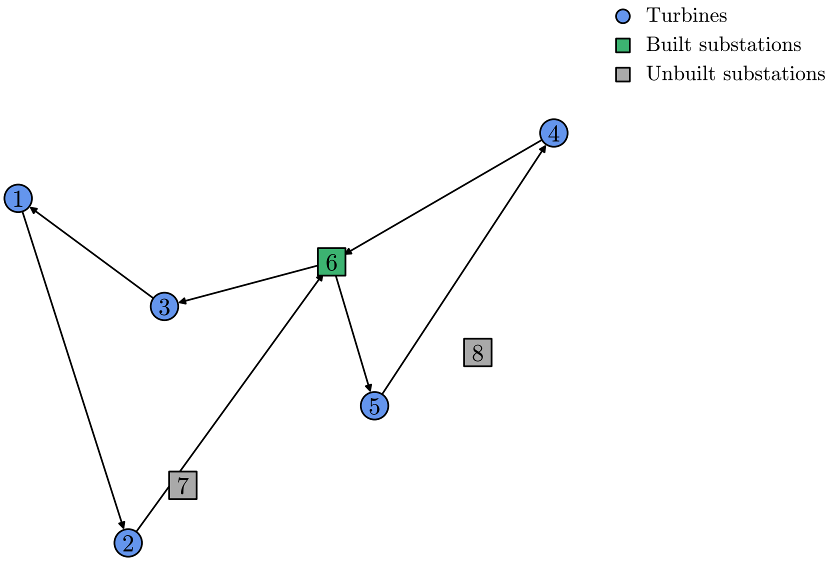


Figure B.4: Solution for instance rc7-5-3-2-l solved by Gurobi

B.2 Additional instances with 10 Turbines

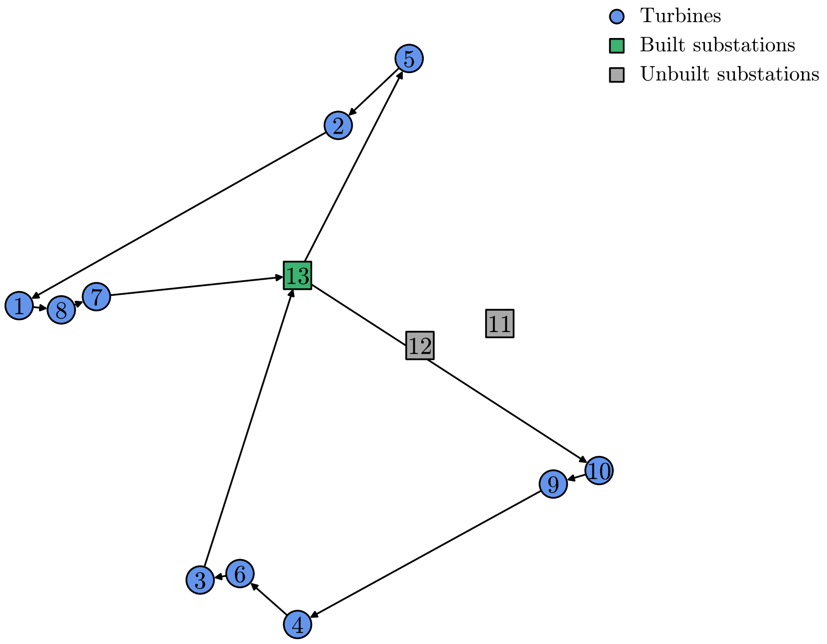


Figure B.5: Solution for instance c5-10-3-3-l solved by ALNSM

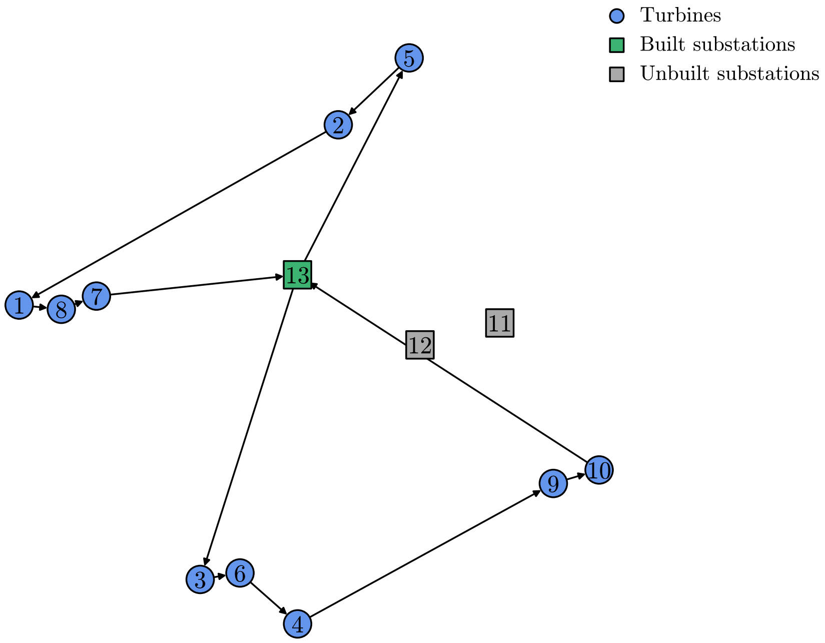


Figure B.6: Solution for instance c5-10-3-3-l solved by Gurobi

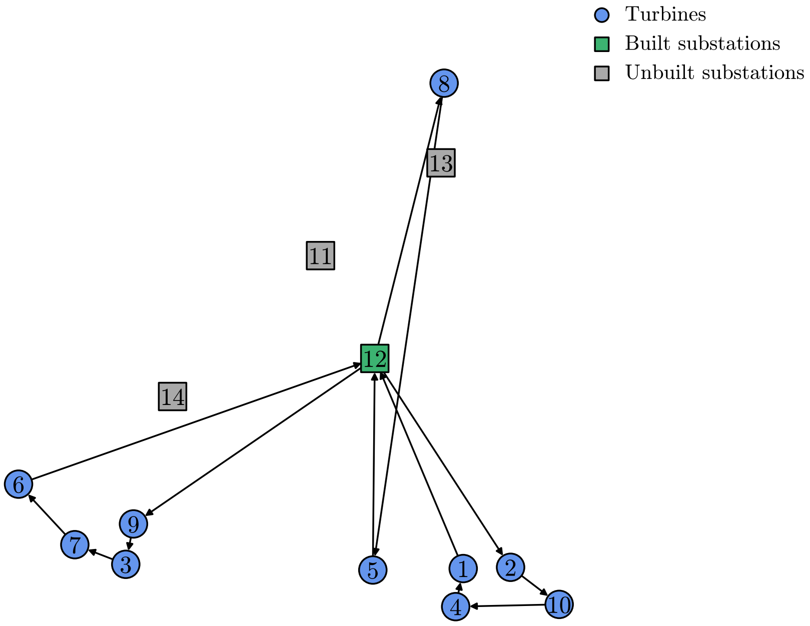


Figure B.7: Solution for instance rc3-10-4-3-s solved by ALNSM

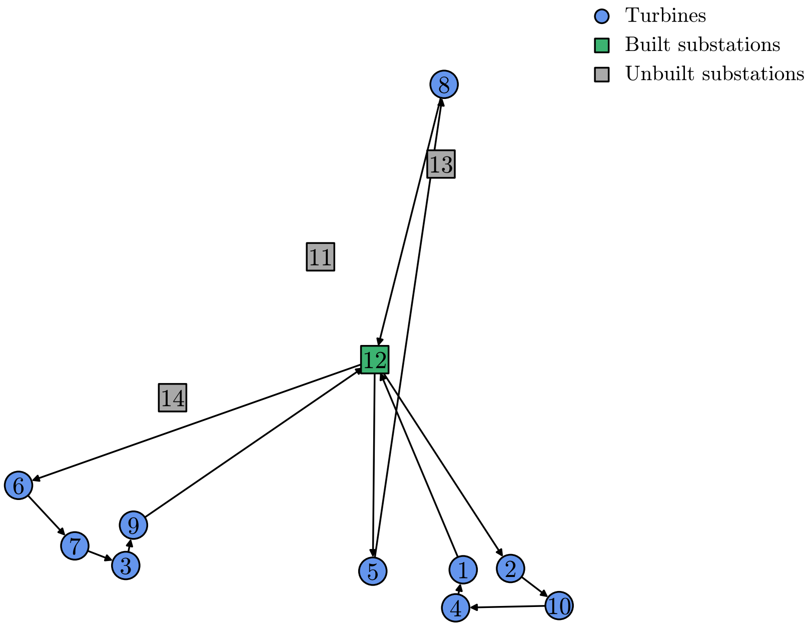


Figure B.8: Solution for instance rc3-10-4-3-s solved by Gurobi