Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Economics and Management
Dept. of Industrial Economics and Technology Management

Andreas Bakke Moan
Pål Ødeskaug

A Hybrid Genetic Approach to the Operational Supply Vessel Planning Problem with Speed Optimization

Reducing Costs and Emissions in the Upstream Supply Chain for Offshore Oil & Gas Production

Master's thesis in Industrial Economics and Technology Management

Supervisor: Kjetil Fagerholt

June 2020

NTNU
Norwegian University of
Science and Technology

Andreas Bakke Moan
Pål Ødeskaug

A Hybrid Genetic Approach to the Operational Supply Vessel Planning Problem with Speed Optimization

Reducing Costs and Emissions in the Upstream Supply Chain for Offshore Oil & Gas Production

Master's thesis in Industrial Economics and Technology Management
Supervisor: Kjetil Fagerholt
June 2020

Norwegian University of Science and Technology
Faculty of Economics and Management
Dept. of Industrial Economics and Technology Management

NTNU
Norwegian University of
Science and Technology

Preface

This master's thesis is written as a contribution to our Master of Science for the Department of Industrial Economics and Technology Management at the Norwegian University of Science and Technology. This thesis builds on our findings and the work done in our specialization project during the fall of 2019.

In this master's thesis, we address the issue of operational offshore supply vessel planning for Equinor. This project is done as a collaboration between SINTEF's LowEmission Research Centre, Equinor and the Norwegian University of Science and Technology to reduce costs and emissions from Equinor's upstream supply chain operations at the Norwegian Continental Shelf. We would like to thank our supervisor Professor Kjetil Fagerholt, our contacts Truls Flatberg, Lars Magne Nonaas and Elin Espeland Halvorsen-Weare at SINTEF and the team at Equinor for valuable guidance, interesting discussions and precise, constructive feedback. We also extend our gratitude to Professor Thibaut Vidal at the Pontifical Catholic University of Rio de Janeiro for guidance in the exploration of solution methods.

Trondheim, June 2020

Andreas Bakke Moan & Pål Ødeskaug

Abstract

This master's thesis addresses the operational aspect of supply vessel planning in offshore oil and gas logistics faced by Equinor, the leading energy company in Norway. In order to operate continuously, offshore installations regularly need supply deliveries from an onshore supply depot. These supplies are transported to the installations with platform supply vessels (PSVs). Currently, operations research-based support tools are used on a tactical level, whereas the operational planning is still performed by hand after taking external factors like weather forecasts into account. Obtaining cost-effective solutions by hand is cumbersome for problems of this size and Equinor has expressed the need for an operational decision-support tool.

This master's thesis considers the Operational Supply Vessel Planning Problem with Speed Optimization (OSVPPSO) which minimizes the costs related to the operations of PSVs in the original fleet and chartering of external PSVs for support. An exact mathematical formulation of the OSVPPSO along with a *Hybrid Genetic Search with Adaptive Diversity Control (HGSADC)* for quickly obtaining high-quality solutions are presented.

To introduce the operational aspect of the problem, weather forecasts are taken into account. For weekly plans where weather forecasts are not taken into account, poor weather may lead to disruptions and missed deliveries. Including weather forecasts enable planning of voyages and schedules accounting for weather-dependent operational restrictions for PSVs and installations. Weather-dependent speed optimization allows voyages and schedules to be tailored to the weather forecast for the upcoming days. If possible, fuel-efficient sailing speeds are desired. However, if the weather becomes worse with time, PSVs may increase their speed and perform deliveries in advance, and thus avoid postponed deliveries and expensive halts in production. Accounting for this, a solution to the OSVPPSO yields weather-adapted voyages and schedules for the PSVs departing the next day, where speed optimization is applied for efficient use of PSVs.

Smaller-sized problem instances are solved to optimality with the exact solution method within a time frame of one hour. Due to the complexity of the problem, medium and large sized instances cannot be solved to optimality using a commercial solver within reasonable time. However, the HGSADC provides environmentally friendly and cost-efficient solutions within this time frame. Results from the computational study show that a decision-support tool for the OSVPPSO can be valuable in supply vessel planning.

Sammendrag

Denne oppgaven adresserer det operasjonelle aspektet ved planlegging og bruk av forsyningsfartøy i Equinors offshore olje- og gasslogistikk. For at en offshore olje- og gassplattform skal kunne operere kontinuerlig, trenger den forsyninger fra et forsyningslager på land. Forsyningene fraktes fra lageret til plattformene med forsyningsfartøy, også kalt *platform supply vessel (PSV)*. På nåværende tidspunkt besitter Equinor beslutningsstøtteverktøy som brukes i planleggingen av repetitive ukentlige planer. Disse overordnede, repetitive ruteplanene må hver dag tilpasses operasjonelle faktorer som værforhold. Denne daglige planleggingen gjøres i dag for hånd, noe som både er tungvint og ineffektivt. Equinor har derfor uttrykt behov for et operasjonelt verktøy for beslutningsstøtte.

Masteroppgaven betrakter det *Operasjonelle Planleggingsproblemet for Forsyningsfartøy med Hastighetsoptimering (OSVPPSO)*. Problemet minimerer kostnader forbundet med bruk av fartøy i den originale flåten og eventuelle leiekostnader av ekstra forsyningsfartøy. Det presenteres en eksakt formulering av problemet, samt en metaheuristisk *hybrid-genetisk søkealgoritme med adaptiv mangfoldskontroll (HGSADC)* for å raskt oppnå gode og kostnadseffektive løsninger. For å introdusere det operasjonelle aspektet ved problemet, tas reelt værvarsel i betraktning i planleggingen.

I motsetning til ukentlige planer der dårlig vær fører til forstyrrelser i ukeplanen og leveranser som ikke kan utføres og må utsettes, gjør værbasert operasjonell planlegging det mulig å tilrettelegge ruter og timeplaner til perioder der plattformer kan motta forsyninger. Hastighetsoptimering med hensyn til vær tilrettelegger for at rutene og timeplanene blir tilpasset tidsvinduer hvor plattformene kan betjenes. I utgangspunktet ønsker man å holde langsom og drivstoffseffektiv seilingshastighet, men dersom værforholdene forverres med tiden kan PSVene øke seilingshastigheten og utføre leveranser før planen. På denne måten kan man unngå forsinkede forsyninger og kostbare avbrudd i produksjon fra utstyrsmangel. Med dette tatt i betraktning, vil løsninger på OSVPPSO gi værtilpassede ruter og tidsplaner for PSVene som reiser fra forsyningslageret den neste avreisedagen.

Den eksakte løsningsmetoden klarer å finne optimale løsninger på små probleminstanser innen en tidsramme på én time. Grunnet kompleksiteten av problemet, klarer den ikke å løse større instanser til optimalitet. Den metaheuristiske HGSADCen klarer å finne miljøvennlige og kostnadseffektive løsninger også på store instanser innen kort tid. Resultatene fra beregningsstudiene viser at et beslutningsstøtteverktøy som løser OSVPPSO kan være av stor verdi i operasjonell planlegging.

Contents

Appendices

List of Figures

List of Tables

List of Algorithms

Abbreviations

ALNS	-	Adaptive Large Neighbourhood Search
DAG	-	Directed Acyclic Graph
HGSADC	-	Hybrid Genetic Search with Adaptive Diversity Control
LNS	-	Large Neighbourhood Search
MDO	-	Monotonic Decomposition Algorithm
OSVPPSO	-	Operational Supply Vessel Planning Problem with Speed Optimization
PSV	-	Platform Supply Vessel
PSVPP	-	Periodic Supply Vessel Planning Problem
PVRP	-	Periodic Vehicle Routing Problem
RAP	-	Resource Allocation Problem
RAP-NC	-	Resource Allocation Problem with Nested Constraints
RSA	-	Recursive Smoothing Algorithm
SPP	-	Shortest Path Problem
SVPP	-	Supply Vessel Planning Problem
SWH	-	Significant Wave Height
TSRSPSO	-	Tramp Ship Routing and Scheduling Problem with Speed Optimization
UHGS	-	Unified Hybrid Genetic Search
VNS	-	Variable Neighbourhood Search
VRP	-	Vehicle Routing Problem
VRPPD	-	Vehicle Routing Problem with Pickups & Deliveries
WDSVSOP	-	Weather-Dependent Supply Vessel Speed Optimization Problem

Chapter 1

Introduction

In December 1969, the first evidence of oil and gas on the Norwegian Continental Shelf (NCS) was discovered. Since then, numerous oil and gas reservoirs have been explored and extracted. Since the first petroleum platform commenced production in the early 70s, the oil and gas industry has generated more than NOK 14,900 billion of gross national product in present value, according to the Norwegian Petroleum Directorate (2019b). The Norwegian oil and gas industry has thus accounted for major parts of Norwegian exports and has been an important factor for growth in the domestic economy. In 2017, the petroleum industry was solely accountable for exports worth of NOK 442 billion, almost half of total national exports. The oil and gas industry has played an important role in the Norwegian economy for many years. Today, the Norwegian Petroleum Directorate reports that there are still 53% of recoverable resources on the NCS yet to be extracted.

The oil and gas production on the NCS has, however, a significant environmental impact. According to Norwegian Petroleum Directorate (2019a), about one quarter of the domestic greenhouse gas emissions arise from oil and gas activities. This results in approximately 13 million tonnes of CO_2 equivalents in 2019. With a rising global awareness of climate change and the impact of greenhouse gasses, the petroleum industry is put in an awkward position. Norwegian environmental standards for the petroleum industry are very high compared to those of many other countries. This greatly motivates the research on more environmentally friendly ways to operate and extract oil and gas on the NCS.

In 1972, a Norwegian state owned oil and gas company named Statoil was established to perform petroleum operations on the Norwegian continental shelf. Since then, this company has been the leading operator on the NCS. The company has, for many years, been one of the worlds most carbon-efficient producers of oil and gas (Equinor, 2019). In 2018, as part of a rebranding strategy, its name was changed to Equinor, signalling that oil and gas was no longer the sole focus of the company.

Equinor's oil and gas operations are mainly carried out at offshore installations. These installations are mostly self-sufficient on energy and water, but needs other supplies in order to operate. Supplies include anything from food to industrial tools and equipment to perform certain operations. These supplies are sent out from onshore locations and are carried out to the offshore installations in containers by platform supply vessels (PSVs). These vessels are specifically designed for carrying supplies to offshore locations. Aside from gas turbines and torching, fuel consumed by the PSVs in the upstream supply chain accounts for one of the biggest carbon footprints in Equinor's operations with 350 thousand tonnes CO_2-equivalents yearly. Because of this, optimizing logistics has become a priority in order to enhance efficiency in the upstream supply chain at Equinor, both in terms of costs and carbon footprint.

Figure 1.1: Skandi Mongstad PSV servicing a platform at the Norne field

Through the past decades, Equinor has cut costs and emissions by improving their plans for PSV routing on a strategic and tactical level. As of today, Equinor uses planning support tools developed by the Norwegian University of Science and Technology (NTNU) and SINTEF. Plans are based on weekly averages of supply orders per installation and do not take into account that most weeks have variations within weather or unexpected changes in demand at installations. A plan serves as an optimal base schedule, or a master plan, that is followed until changes in demand or weather conditions force the planners to make changes. Not only have better master plans lead to a total reduction in distance travelled, but also a reduction in fleet size, yielding significant benefits in terms of cost and emission reductions. Equinor reports with this that since 2011, they have reduced the CO_2 emissions from its logistic operations by 26%. However, Equinor's ambition is to further improve their operations, and cut half of the total emissions from the supply chain by 2030. In collaboration with NTNU and the new LowEmission Research Centre within SINTEF, Equinor strives to cut even more emissions from these upstream supply chain activities.

Experience has shown that due to poor weather conditions, wrongly estimated order sizes and sudden urgent orders from the installations, voyages are often disrupted and deviate from the original plans. When this happens, Equinor relies on experienced planners to make the right choices, determining manually how to reroute the PSVs. Until now, the goal has been simple; To catch up with their supply delivery schedule so that the PSVs can return to the master plans as quickly as possible.

At this point, Equinor mainly handles disruptions in their master plans after they occur. Handling of these disruptions are both resource demanding and expensive. In order to return to their master plan after a bad weather period where installations cannot be serviced, planners must reroute the PSVs, and occasionally hire expensive spot vessel to eliminate the backlog. It is often easier to prevent backlog than dealing with them, and thus, we introduce a new approach to supply vessel planning.

Today, Equinor performs all operational planning by hand, every single day. In this planning process, experienced planners consider details for each of the vessels departing the next day, including which cargo to deliver to installations, the size of the deliveries, new urgent demand at installations, weather conditions and other factors. Based on this information, they often have to revise and alter voyages in the master plan. This process is both time-consuming, cumbersome and may lead to less cost-efficient plans. To improve decision-support in Equinor's upstream supply chain, we address the operational aspect of the supply vessel planning. Explicitly, we want to be able to harvest the benefits of planning with regards to the real-case weather forecast, allowing vessels to speed up or slow down to adjust for demand and weather restrictions. This means that the PSVs will no longer follow a weekly master schedule generated from statistical data and weekly averages. For each day, weather forecast and demand are taken into account and operational voyages for vessels departing the next departure day are planned thereafter. We believe that the value of operational information will add significant cost and emission reductions to the supply vessel planning by eliminating backlog and disruption management.

The aspect of operational planning where weather conditions are taken into account has, to our knowledge, not been introduced to the existing research concerning supply vessel planning. Expanding the set of decision-support tools to handle operational variations such as changing weather and unforeseen changes in supply demand can thus create value for Equinor. With better decision-support, Equinor can make even more emission- & cost-efficient decisions in their upstream supply chain.

The Operational Supply Vessel Planning Problem with Speed Optimization thus considers a set of supply vessels servicing a set of offshore installations from an onshore supply depot. Each installation has demand which consists of orders placed by the installation that needs to be delivered within a deadline. A typical solution to the problem includes voyages, schedules and sailing speed on all sailing legs for all supply vessels used. From the solution, it is clear which PSVs will depart from the supply depot the next day. For each PSV it is stated which installations should be visited before returning to the depot, when they should depart from each installation and which speed to keep for each sailing leg on their voyage. The planning period for each PSV typically lasts up to a few days before returning to the depot. The solution also determines if additional PSVs from the spot market are needed on short notice in order to visit all installations within the delivery deadlines. Since only short-term planning horizons are considered, long-term fleet sizing is not part of the operational problem.

In Chapter 2, relevant literature from existing studies is presented and discussed. The OSVPPSO is explained in detail in Chapter 3 and a mathematical model is provided in Chapter 4. Chapter 5 defines and describes the weather-dependent speed optimization problem which is solved as a subproblem in the metaheuristic solution method for the OSVPPSO. The metaheuristic solution method, the *Hybrid Genetic Search with Adaptive Diversity Control (HGSADC)*, is explained in Chapter 6. Chapter 7 elaborates on the problem instances and scenarios used. Further, the computational study can be found in Chapter 8, where a thorough analysis of the solution methods is provided. Chapter 9 concludes our remarks.

Chapter 2

Literature Review

In this chapter, relevant literature for the planning problem addressed in this master's thesis is presented. There exists a large scope of comprehensive literature within vessel planning. Thus, the literature studied is mainly collected from adjacent problems of vessel planning for the offshore petroleum industry. Several of the consecutive topics in this chapter is earlier addressed by Moan and Ødeskaug (2019), and parts of that literature review are used here. Section 2.1 covers earlier research on planning problems for supply vessels and other route planning problems. Section 2.2 presents how weather has been dealt with in earlier planning problems, while Section 2.3 presents research within speed optimization to reduce emissions and pollution. Section 2.4 elaborates on the use of heuristic solution methods to solve problems adjacent to the one addressed in this master's thesis. The terms "routes" and "voyages" are used interchangeably as routes are referred to as voyages in maritime literature.

2.1 Relevant Route Planning Problems

There exist several model formulations for the *Supply Vessel Planning Problem (SVPP)*, which address the tactical issue of identifying the cost-optimal fleet size, voyages and schedules for a set of vessels servicing a range of offshore installations from an onshore depot. One of the early papers addressing this problem was published by Fagerholt and Lindstad (2000). In this paper, cost-optimal schedules are obtained, and also the cost of having offshore installations closed for service at night is analyzed. The SVPP can, to a large degree, be compared to a Multi-Trip Vehicle Routing Problem, which allows each vehicle to perform several trips (i.e. depart from and arrive at the depot multiple times) during the planning horizon. A two-stage solution approach is used in this paper. In the first stage, feasible candidate schedules are generated for all vessels available. Further, in the second stage, an integer programming model decides which vessels to use and assigns weekly schedules to these vessels. Two-stage approaches have shown to be effective to solve various versions of the SVPP in several research papers, and has been heavily used through the last decade. In Fagerholt and Lindstad (2000)'s two-stage model, the second stage involves an integer programming model that solves a relaxed version of the complex SVPP, where some important real-life considerations like service capacity constraints for the supply depot, spread of departures, and minimum and maximum voyage duration are not included in the model. Halvorsen-Weare et al. (2012) also use a two-stage approach in their solution method.

In comparison to Fagerholt and Lindstad (2000)'s paper, Halvorsen-Weare et al. (2012)'s *voyage-based model* covers some of the constraints not included in Fagerholt and Lindstad (2000). Focusing on the periodic aspect of the problem has allowed the model to account for spread of departures during the planning period, which was not accounted for in Fagerholt and Lindstad (2000)'s methodology. This model has also been utilized as a decision support tool in the planning of Equinor's, formerly known as Statoil, offshore supply vessels.

The voyage-based solution approach has been heavily used in earlier literature to solve different versions of the SVPP. An *arc-flow model* is also a two-stage method presented by Halvorsen-Weare and Fagerholt (2017). In this paper, the arc-flow model presented is compared to the voyage-based model. The arc-flow model is proved to be outperformed by the voyage-based model on larger problem instances by the computational study. However, the arc-flow model provides a more detailed description of the problem compared to the voyage-based model.

Aas et al. (2007) suggests an approach to find optimal voyages for supply vessels as an extension of the *Single Vessel Routing Problem with Pickup and Deliveries (SVRPPD)* presented in Gribkovskaia et al. (2007). Their solution method to the SVRPPD is again a problem specific extension of the *Vehicle Routing Problem with Pickup and Deliveries (VRPPD)*. Further, Aas et al. (2007) builds on this model and extends the SVRPPD so that each customer is allowed to be visited more than once during the planning period. This paper suggests an arc-flow model to generate optimal voyages for the supply vessels. However, the relatable VRPPD is a NP-hard optimization problem, making large instances the main limitation of the model in Aas et al. (2007). The authors suggest that the SVRPPD with capacity restrictions can be solved using a tabu-search heuristic, especially for larger problems.

As the SVPP has some common features with the VRP-class, research on VRPs is often applicable for variants of the SVPP. Hashimoto et al. (2006) have an interesting approach to the *Vehicle Routing Problem with Time Windows (VRPTW)*, where they modify their solution method to include flexible time windows and travelling times. Normally for the VRPTW, the time window constraints are hard constraints, which must be fulfilled for a solution to be feasible. Hashimoto et al. (2006) suggest making the time windows constrains soft. These constraints may then be violated at a penalty cost.

2.2 Weather Handling in Routing Problems

Through the past couple of decades, different variants of the SVPP have been studied. Experience has shown that implementation of a plan is highly sensitive to weather conditions. Wave height, wave direction and wind speed can prevent vessels from carrying out planned voyages. It should be emphasized that wave height is the most influential factor on sailing and service times (Kisialiou et al., 2018b). Rough weather may also give speed limitations for vessels and slower installation service, resulting in voyages taking longer time than originally planned. An approach for robust solutions of the *Periodic Supply Vessel Planning Problem (PSVPP)*, which allows a PSV to sail more than one voyage in the planning horizon, that is less sensitive to uncertain and harsh weather conditions is presented by Kisialiou et al. (2018b). A voyage is considered robust if it is feasible for all weather conditions. This paper yields an analysis of the trade-off between cost and robustness. The aim of the paper is to generate weekly schedules, with the intention of being repeated over a longer time horizon to maintain stability and predictability in operations. To account for travel- and service time delay and avoid disruptions in the schedule, two types of slack are introduced. Intra-voyage slack, which is a time buffer for each installation visited during the voyage and inter-voyage slack, which is a time buffer between two consecutive voyages for the same vessel. The amount of intra- and inter-voyage slack assigned is controlled by a robustness parameter, where the robustness parameter is dependent on the expected weather condition. This solution method utilizes an adaptive large neighbourhood search (ALNS), inspired by the one developed by Kisialiou et al. (2018a), to construct robust schedules. A second heuristic search algorithm is applied to construct the schedule of highest robustness, by taking costs and fleet size into account. While Halvorsen-Weare and Fagerholt (2011) and Norlund et al. (2015) look at the impact of weather uncertainty for small- and medium-sized instances, Kisialiou et al. (2018b) study more realistic- and large-sized instances.

Weather handling is also important for speed optimization problems, as rough weather and wave height impacts a vessel's speed and fuel consumption as well as the time spent servicing the installations. Halvorsen-Weare and Fagerholt (2011) and Norlund and Gribkovskaia (2017) account for weather by dividing into four different categories of weather conditions that affects speed and installation service time, which will be addressed in the subsequent section.

It should be mentioned that earlier papers, to a large extent, create voyages for a short planning period that are meant to be repeated over a long period of time. Therefore, earlier solution models mostly handle weather with statistics over a longer period of time when optimizing voyages instead of taking in real time weather forecasts. Less research has been done on the operational short-term planning level, where the model uses real-time weather forecasts as input. The only papers, that to our knowledge, address operational supply vessel planning with regards to weather, are Albjerk et al. (2016) and Stålhane et al. (2019). However, instead of using predictions to foresee rough weather, they utilize disruption management to handle it after poor weather conditions have disrupted the plans. Their goal is to minimize the impact of the disruptions and get back to the tactical and predetermined master schedule as quickly and cost-efficiently as possible.

2.3 Speed Optimization

Supply vessel planning problems have mostly been studied from the perspective of cost minimization. Lately, it has been of larger interest to reduce emissions. Utilizing operations research can help supply chains plan more efficiently and hence reduce their environmental footprint.

Norlund and Gribkovskaia (2017) address the issue of reducing emissions in supply vessel planning and looks at how much fuel consumption can be saved by choosing the right speed on voyages in different weather conditions. By applying two different voyage speed optimization strategies on pregenerated voyages, a technique presented in an earlier paper by Norlund and Gribkovskaia (2013), they evaluate how much emissions can be reduced by utilizing the two different speed strategies. In the simplest algorithm, the algorithm checks if a vessel is prone to waiting on time windows to open on a sailing leg. If the vessel faces waiting time after arrival at the installation, it reduces the design speed to the leg distance divided by total sailing and waiting time or the lowest feasible speed. The second speed strategy is somewhat more comprehensive. It focuses on a full voyage, where each sailing leg in the voyage is considered with an initial design speed. For a voyage, it is observed for each sailing leg if and how much the time windows at each installation are violated. The sailing leg with the largest violation is then used to split the voyage into two separate sailing legs, recursively changing the optimal speed for the new split sailing legs. A similar algorithm is in the subsequent paragraph referred to as a *Recursive Smoothing Algorithm (RSA)*. After optimizing speed for every split leg on these pregenerated voyages, a discrete-event simulation model determines how different weather states in the planning horizon influence fuel consumption and voyage duration. Afterwards, a procedure to calculate the expected average fuel consumption is performed. The solution to the problem consists of schedules designed with speed optimized voyages for each vessel that are simulated under various weather conditions. Computational studies show that for a voyage with 3 installation visits and design speed of 12 knots, the effect of the first speed strategy yields a fuel reduction of 12%, while the second strategy yields a reduction of 25% compared to the fuel consumption with the design speed. With regards to dealing with weather, Norlund and Gribkovskaia (2017) takes into account that a specific speed in rough weather requires higher fuel consumption than in ideal weather and handles this with calculating fuel consumption as a function of wave height.

In Norstad et al. (2011), *Tramp Ship Routing and Scheduling with Speed Optimization (TSRSPSO)* is studied. They present an extension of the arc-flow formulation formulated by Christiansen et al. (2007), where speed optimization is taken into account by introducing speed variables to the model. Real-life instances of the TSRSPSO are too large to solve exactly within reasonable time, and the authors suggest using a multi-start local search heuristic. In the model, optimizing speed is considered a subproblem, which they suggest can be solved in two different ways. The first alternative is to take advantage of the arc-flow structure by discretizing the arrival times for nodes, creating a *Directed Acyclic Graph (DAG)*. The technique of discretizing arrival times in an arc-flow network, was originally introduced by Fagerholt (2001) and was also later used in speed optimization techniques in Fagerholt et al. (2010). Further, the resulting problem becomes a *Shortest Path Problem (SPP)* which can be solved for the Directed Acyclic Graph. The second way of solving it would be with a RSA, quite similar to the one described in the paragraph above, which changes sailing speeds on the sailing legs with the largest time window arrival violations. In terms of computational time, the recursive smoothing algorithm performs better than the SPP. The RSA was proposed in Norstad et al. (2011), but it is analyzed and proven to be exact in Hvattum et al. (2013). The authors also prove that its worst case running time equals $O(n^2)$.

Andersson et al. (2015) address a different approach of dealing with speed optimization in a RoRo-shipping routing problem. In this paper, an arc-flow model is presented in conjunction with a rolling horizon heuristic. A speed decision variable is included in the model to determine the weighing of speed alternatives for a ship along an arc. To calculate fuel consumption, they base their calculations on a non-linear, strictly increasing convex function, implying that fuel consumption increases quadratically with speed per distance. The authors of this paper has chosen to handle the convex non-linear fuel consumption function by approximating it with discrete values of speed, and hence, the approximation is given by the linear combination of neighbouring discrete speed values. This method makes it possible to omit use of special ordered sets of type 2 (SOS2). By applying a linear combination of two neighbouring points, the curve will always yield an equal or higher fuel consumption than the real convex consumption function. Thus, this approach will provide a small overestimate of the real fuel consumption.

In speed optimization problems, it is necessary to estimate fuel consumption for a ship at different speeds. In the objective function, different speed alternatives are evaluated to obtain the optimal solution based on their fuel consumption. For each unique ship, there is a corresponding unique fuel consumption function describing the relationship between speed and fuel consumption. However, it can be difficult to obtain the exact function for a specific PSV, and hence, it is not an unusual simplification to use fuel consumption functions from close-to-similar vessels. Most fuel consumption functions exhibits a cubic increase in fuel consumption per time as the speed increases. These consumption functions have been used in various studies within operations research. Psaraftis and Kontovas (2014) provide a detailed study of speed optimization with calculation of fuel consumption per time unit as a polynomial function of time and payload. However, the authors point out that cubic functions of speed yielding fuel consumption per time unit are reasonable approximations for ships of small size. Psaraftis and Kontovas (2014) emphasise that in terms of monetary costs, if variable chartering rates are high, optimal speed tends to increase, which is relevant for the issue of hiring PSVs from the spot market.

Vidal et al. (2019) address the interrelation in vessel speed optimization between the RSA by Norstad et al. (2011) and fuel consumption-dependent factors suggested in Psaraftis and Kontovas (2014). Vidal et al. (2019) take inspiration from the recursive smoothing algorithm by utilizing divide and conquer methods, but further states that the simplification of assuming a constant fuel consumption function is unrealistic. In Psaraftis and Kontovas (2014), it is stated that fuel consumption is dependent on continuous varying factors, such as sea condition, weather, current, water depth and ship load. In order to deal with variations in fuel consumption, Vidal et al. (2019) model the problem as a *Resource Allocation Problem with Nested Constraints (RAP-NC)* and with convex costs. Here, time is considered a resource that is allocated to each sailing leg. As speed optimization is used as a sub-procedure in many problems, and hence solved a vast number of times, a time-efficient algorithm that solves this RAP-NC is needed. For this, Vidal et al. (2019) suggests the *Monotonic Decomposition Algorithm (MDA)*, a recursive algorithm based on the divide and conquer principle for splitting voyages into smaller segments and using different fuel consumption functions.

In Lindstad et al. (2013), emissions are assessed by varying speed as a function of sea conditions and freight market. In their study, they provide the emission minimizing speed for a loaded dry bulk vessel, which varies on the direction of the waves.

2.4 Heuristic Approaches

Earlier research address different variants of the offshore PSV routing problem. Formulating these problems is often straight forward, however, solving them is difficult. The size of realistic problem instances yields a complexity which, for most computers, makes the problem too difficult to solve within reasonable time. Researchers have in response to this complexity, tried to get around the issue by implementing heuristics. These heuristics span from neighborhood searches to genetic algorithms and provide high-quality solutions quickly, even though optimality is difficult to prove.

Halvorsen-Weare et al. (2012) presents a voyage-based formulation for the Periodic Supply Vessel Planning Problem. Shyshou et al. (2012) base their study on this formulation and present a *Large Neighbourhood Search* (LNS) to solve larger problem instances. Computational results show that the voyage-based formulation performs slightly better and faster than the heuristic for instances up to 12 installations. When exceeding a size of 14 installations, enumerating all of the cheapest feasible voyages in the voyage-based formulation becomes time consuming enough for the heuristic to dominate. Computational results show that the neighbourhood search provides satisfying solutions for 31 installations within 15,000 seconds, and thus can solve realistic-sized problem instances.

In general, companies establish master schedules for their PSVs to maintain predictability in their operations. Due to unforeseen events like poor weather conditions and variations in demand, PSVs tend to deviate from the master schedule. It can often be resource demanding and expensive to handle these disruptions and return to the master schedule. Albjerk et al. (2016) address the problem of getting back to the master schedule. After disruption has occurred, the aim is to return to the master schedule before the next voyage is planned to start for each PSV. The problem is formulated as a Multi-Vehicle Pickup- and Delivery Problem, and both the arc-flow and the path-flow formulation presented are able to solve problem instances consisting of 5-8 installations, depending on the disruptions that occur. Stålhane et al. (2019) study the same issue, and suggests a variable neighbourhood search heuristic with perturbations for the disruption management problem. Comparison of the computational results show that the heuristic manages to find all optimal solutions where the optimal solution has been found by the exact solution method, only faster. The heuristic algorithm also manages to solve realistic-sized instances of 26 installations within reasonable time. It is worth noticing that both these disruption management problems are concerned with returning to a set master schedule, and are carried out after the disruption has occurred.

Routing problems for supply vessels often have similarities with the commonly known *Vehicle Routing Problem (VRP)*. Lately, evolutionary algorithms have been studied as a tool for solving different variants of the VRP. Vidal et al. (2012) have addressed several variants of the VRP and propose a metaheuristic population-based *hybrid genetic algorithm* framework for solving this class of VRPs. A hybrid genetic algorithm, also known as a *memetic algorithm*, is a special case of genetic algorithms, which stands out from general genetic algorithms by exploiting domain knowledge. This knowledge is acquired with tools like e.g. approximation algorithms, local search or other heuristics. The method developed by Vidal et al. (2012) is referred to as a *Hybrid Genetic Search with Adaptive Diversity Control (HGSADC)*, and has proven to outperform the state-of-the-art solution methods for the corresponding variants of the VRPs addressed. The HGSADC is an implementation of the *Unified Hybrid Genetic Search (UHGS)*, a generic solution framework for a broader set of VRPs, presented by Vidal et al. (2014). In the UHGS, procedures are general and needs to be tailored to best fit the problem addressed.

Later, the Hybrid Genetic Search with Adaptive Diversity Control for the *Periodic Vehicle Routing Problem (PVRP)* has been modified and adapted to solve the periodic supply vessel planning problem by Borthen et al. (2018). In addition to this method, they have added a method for reducing the fleet size. Hence, their solution approach comprises two components, whereas the first component is the HGSADC generating high-quality voyages and schedules for a fixed fleet, and the other component is fleet minimizing using the genetic search as a sub-procedure. In the fleet minimization step, the algorithm verifies that there exists a feasible solution for a given fleet size. If a solution exists, the algorithm iteratively reduces the fleet size by one PSV and restarts the procedure looking for a feasible solution. When the fleet size is reduced to a size where no feasible solution to the problem exists, the fleet size is increased by one vessel, to the fleet size where the algorithm knows that there exists feasible solutions using a minimal number of PSVs. Then, for this fleet size, the HGSADC is run to obtain high-quality solutions. This approach works because Borthen et al. (2018) state that chartering costs are much higher than sailing costs, and hence, reductions in sailing cost from increasing the size by one vessel will never exceed the increase in the corresponding chartering costs. This solution method have been tested on up to a size of 27 installations and 80 weekly services, and it significantly outperforms traditional two-stage approaches, yielding equal or better results faster. Computational results show that this method is both scalable and stable. Borthen et al. (2018) use this approach to solve the periodic supply vessel problem, and further extend the approach to solve a multiobjective problem in Borthen et al. (2019). The multiobjective approach provides good solutions with regards to costs and persistence. Persistence meaning that "a new plan contains few changes from the previous plan" (Borthen and Loennechen, 2016).

2.5 Summary

To this date, different solution approaches have been suggested in the voyage planning for offshore supply vessels. Some have proved to be efficient, and have been implemented as a planning-support tool for Equinor. Offshore supply vessel planning has mainly been studied from a tactical and strategic level. Various studies have looked at ways of dealing with weather disruptions and speed optimization, but none of these studies address it from the operational view presented in this master's thesis. In the existing literature covering planning problems for offshore supply vessels, no studies have, to our knowledge, used real-case weather forecasts as input to the solution methods, enabling the possibility of speeding up deliveries to installations in case of rough weather. Hence, this paper will introduce a new approach to supply vessel planning.

Chapter 3

Operational Supply Vessel Planning Problem with Speed Optimization

This chapter contains a thorough description of the problem addressed in this master's thesis. In Section 3.1, input to the problem is provided. Section 3.2 covers which decisions are to be made, and further, the objective and restrictions of the problem is presented in Section 3.3. In the last part of the chapter, Section 3.4 provides an illustrated example of the problem we are facing and a solution to the example problem. We will hereby refer to the problem addressed in this master's thesis as *The Operational Supply Vessel Planning Problem with Speed Optimization (OSVPPSO)*.

3.1 Problem Input

As input to the Operational Supply Vessel Planning Problem with Speed Optimization, a fixed set of long-term contracted PSVs is provided. However, at an arbitrary point in time, some vessels might not be available on the next departure day due to ongoing operations. Thus, a PSV is considered available as long as it has returned to the supply depot from its previous operations before vessel preparation starts the next day. Vessel preparation at the supply depot involves loading and stacking of cargo onto the departing PSVs. It must also be provided for each of the PSVs departing within when it has to be back at the supply depot, i.e. the maximum duration of a voyage for a PSV, also referred to as the *return time* for the specific PSV. Having the opportunity to specify when a vessels needs to be back at the supply depot provides the planners with flexibility and predictability in the planning. This way, planners can make sure they have a desired vessel back in the supply depot at a specific point of time in case it is needed for other operations. Also, the minimum and maximum limits on sailing speed for the PSVs are given, and for each vessel, its specific load capacity limit is provided.

A set of installations and a supply depot is provided with location and distance to the surrounding installations and supply depot. Each installation report their demand as orders to the supply depot, containing information about necessary replenishment and equipment, and within when the specific orders are needed, i.e the delivery deadline for orders. Multiple orders with different delivery deadlines can be placed by an installation. Which orders to bring on the PSVs departing the next departure day is up to the planners and must be provided as input to the problem. The orders not included on the PSVs departing the next day have to be shipped at some other day. A number of the installation's orders can also be aggregated and delivered on one visit, however, no delivery deadlines can be exceeded. Also, all orders with the same delivery deadline are aggregated into one installation visit. A subset of the installations experience limited opening hours, denoting the time interval during the day in which the installations can be serviced by a PSVs. Opening hours for when the supply depot can prepare a PSV for a new voyage is also given, and the preparation starts at the same specific point of time every day.

A voyage is defined as the ordered set of sailing legs a vessel will sail between departure from, and the return to the supply depot. This includes all sailing legs between the installations the vessel visits along that voyage. During a voyage, the PSV can perform four different activities, which are later illustrated in Figure 4.1. These activities include *preparation* of the vessel at the supply depot before departure, *sailing* between installations, *idling* at an installation while waiting for the installation to open or for the weather conditions to improve, and *servicing* of an installation, which includes delivering cargo. Some of these activities experience a larger fuel consumption per time than others. Therefore, each of the activities has separate and unique fuel consumption functions provided for each PSV to obtain a better estimate of the total fuel consumption on a voyage. The fuel consumption function for each activity also depends on various factors. The fuel consumed during the preparation activity only depends on the time, while the fuel consumption function for sailing, is more complex and also depends on the vessel´s sailing speed and the weather conditions in which it is sailing. Furthermore, the fuel consumption for idling and servicing also depends on the weather conditions. The weather conditions are provided by a weather forecast. The weather forecast presents the significant wave height, which is defined as the mean height of the highest third of the waves. Fuel consumption for all activities, except for the supply depot preparation, increases with the significant wave height. To obtain the correct fuel consumption, these functions thus need detailed information about the significant wave height from the weather forecast. The terms *weather* and *weather condition* will in this thesis refer to the significant wave height.

To evaluate the monetary cost of the fuel consumption, the price of fuel is provided. The time used to prepare PSVs for new voyages in the supply depot is given and equal for all PSVs. The service time at a given offshore installation, i.e. the time a PSV spends on loading and unloading, depends on the amount of cargo to be loaded and unloaded. It also varies with the weather conditions. Rough weather conditions, shown through higher significant wave height, might prolong the service times at the installations. Under very rough weather conditions, i.e. when the significant wave height goes above a certain limit, the service at an installation might even be postponed due to safety requirements.

3.2 Decisions To Be Made

A solution to the OSVPPSO will, for the next departure day, provide weather-adapted voyages and schedules for the vessels available for offshore operations, such that the delivery deadlines for all installations are met. Note that the voyage and schedule for a specific vessel's next voyage may be planned if and only if the vessel is available the next departure day. Also, note that for the available vessels, only the next voyage is considered as demand for the distant future, experience more uncertainty with regards to the actual demand at the time when it is needed. Additional PSVs from the spot market can support the original fleet in case it cannot meet all delivery deadlines.

For each sailing leg along a PSVs voyage, the sailing speed must be determined. Higher speed levels yields higher fuel consumption and therefore become more costly in terms of monetary value. Hence, the PSVs endeavour to sail as fuel-efficient as possible, still ensuring that all installations' supplies are received in time. Voyages and schedules may vary from day to day due to sudden urgent orders from installations, wrongly estimated order sizes and weather variations and thus, the PSVs do not follow a rigid repetitive schedule.

3.3 Objective & Restrictions

The overall objective is to minimize the total costs related to the variable operating cost of the current fleet as well as costs of chartering and operating additional PSVs from the spot market. The variable costs are evaluated from fuel consumption, which is multiplied with the price of fuel to obtain a monetary cost. As the time charter rate for hiring a PSV from the spot market is high, it is beneficial to avoid using vessels from the spot market. However, if needed, these PSVs can be hired for short-term use on short notice at a high cost.

Restrictions for the problem are explained in *italics*. On the departure day, a set of installations with supply demand is provided. To fulfil this demand, a PSV must carry the needed supplies from the supply depot to the installations and perform a service job to deliver the supplies. *All service jobs at the offshore installations must be performed by the PSVs*. To ensure that all installations receive their requested delivery in time, *delivery deadlines* must be met for all installations. Due to regulations, some of the installations are subject to *limited opening hours*, indicating when they be can serviced by a PSV. Different levels of significant wave height affects the fuel consumption for the PSVs. Generally this means that the fuel consumption increases with the significant wave height, and *when the significant wave height exceeds a certain limit, installations are closed for service* due to safety regulations. For the PSVs, *the minimum and maximum limits on sailing speed cannot be violated*. Also, the *maximum sailing speed for a PSV is dependent on the significant wave height*. Each PSV has an individual *load capacity limit* on the amount of cargo it can transport, allowing a heterogeneous fleet of PSVs. Each PSV is also subject to a *maximum voyage duration*. The maximum voyage duration can also be interpreted as a specific return time for a PSV, meaning that planners know at which time the PSV will be available at the supply depot again.

3.4 Illustrative Example of a Solution to the OSVPPSO

In this section, an example is provided to illustrate a feasible solution to the OSVPPSO. Let's say that a company operates on a large oil and gas field with 27 installations. In the following days, a subset of the installations will be serviced on the voyages sailed by the PSVs departing from the supply depot tomorrow. This subset includes the installations *SOD, SEN, CPR, OSC, HUL, WEP, WEL, STB, STA*, which have requested deliveries within a specific deadline set by the respective installation. The company running the installations have multiple PSVs at disposal, but only two of these PSVs, *PSV0* & *PSV1*, are available for departure tomorrow. Due to other offshore operations, the other PSVs are not available tomorrow. If, due to extraordinary circumstances, two PSVs are not sufficient to satisfy the installations' demand within the given deadlines, additional PSVs can be hired from the spot market at a higher price.

Weather conditions may have large impacts on how voyages should be planned. Thus, a weather forecast in terms of significant wave height per hour for the coming days is provided in Figure 3.1. According to this weather forecast, waves are steadily increasing throughout first two days, where the significant wave height is expected to rise above 4.5 meters, which is the limit where installations are not permitted to be serviced due to safety regulations. The no-service limit is shown as a red line in the weather forecast. As long as the significant wave height is above 4.5 meters, installations cannot receive deliveries by PSVs. The weather conditions improve and permits service again after 16 hours. An almost similar cycle is repeated at the beginning of Day 3.

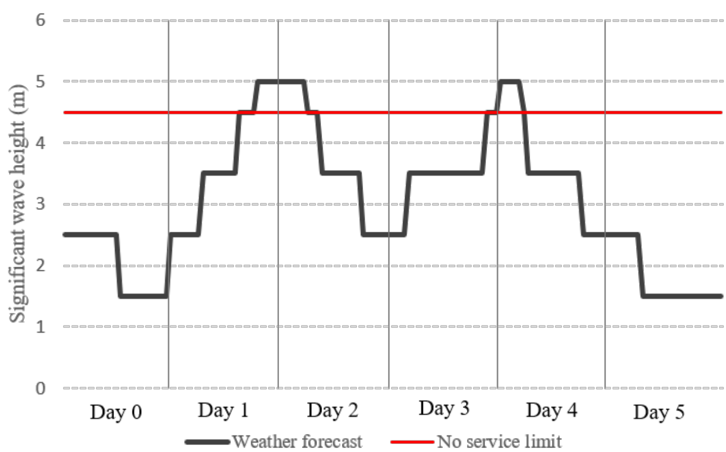


Figure 3.1: Weather forecast showing weather states for the coming days. The forecast starts Day 0 (i.e. the departure day) at 00:00

The weather forecast shows that due to the poor weather conditions coming in at the end of Day 1, it can be beneficial to visit as many installations as possible before the bad weather strikes. This is in order to avoid unfavourable idling at an installation while waiting until the significant wave height permits service of installations again. It is also because higher waves make it more difficult to service the installations, and hence the service becomes less time-efficient. In addition to this, higher waves also impact the fuel efficiency of the PSVs, resulting in higher fuel costs and more emissions released. Thus, it might be favourable to speed up the deliveries. Speeding up the deliveries can be done in two ways. Using additional vessels, i.e. hiring a PSV from the spot market, or speeding up the current fleet. As long as it is more cost-efficient, it might be better to increase the speed of the PSVs in the fleet rather than hiring additional PSVs.

Figure 3.2 shows a map of all installations in the field of operations. In this example, the installations *TRO, TRB, TRC* and *STA* experience limited opening hours from 7 am to 7 pm. All other installations are open 24 hours a day. The most cost-efficient voyages for *PSV0 & PSV1* are shown in the same figure. While *PSV0* visits installation *OSC, HUL, WEP, WEL, STB* and *STA*, *PSV1* visits installation *SOD, SEN* and *CPR*. Note that installation *STA* have limited opening hours.

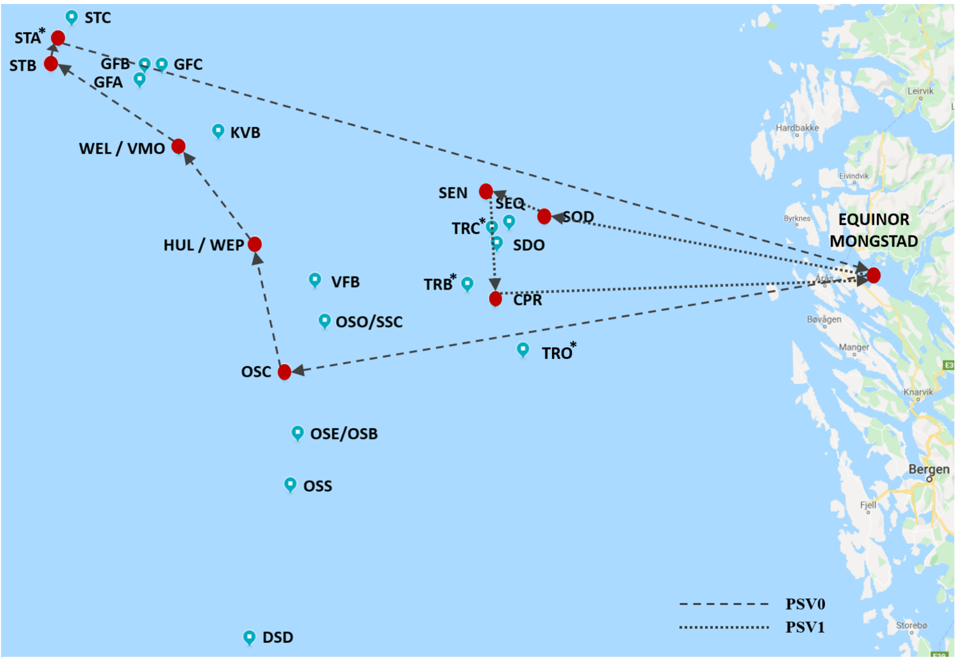


Figure 3.2: Two voyages sailed by *PSV0* and *PSV1*. All installations are marked with name. The installations visited are marked by a red dot. * denotes installations with limited opening hours

Figure 3.3 shows the schedule for the voyages departing tomorrow. The pattern bars indicate the start and finish times for all activities performed by a PSV, namely supply depot preparation, sailing, idling and service of an installation. Which installation that is visited and serviced is shown above the *servicing-bar* in the schedule. On top of the schedule, a speed profile is provided above the *sailing-bar*, showing the desired average speed for the sailing leg. The permitted speed sailed by a PSV ranges from 7 to 14 knots, which is the range shown in the speed profile.

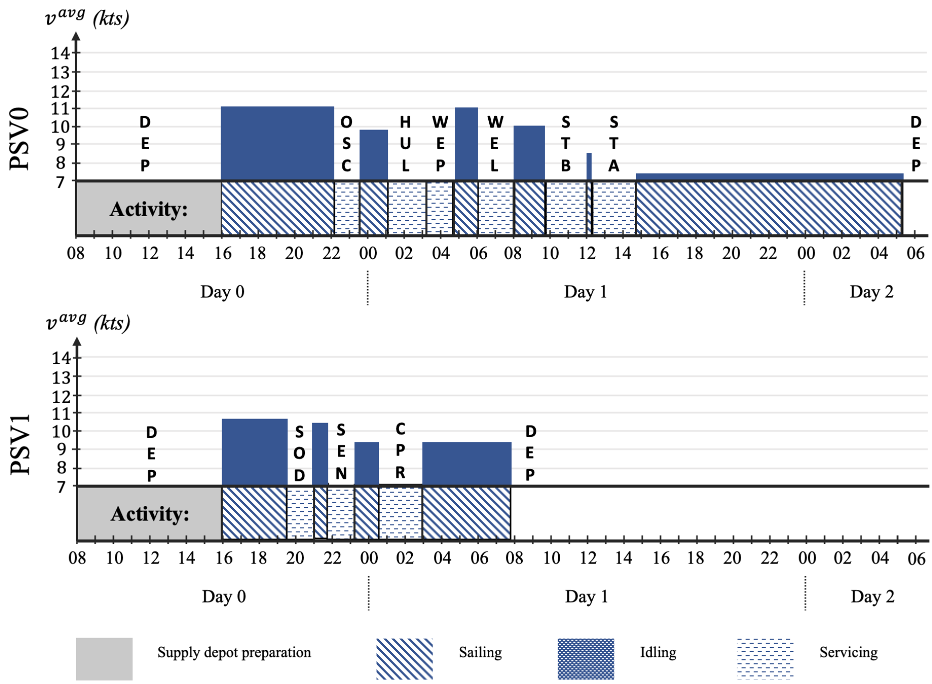


Figure 3.3: Weather-adapted schedules and speed profiles for *PSV0* and *PSV1*

The weather forecast in Figure 3.1, shows that there is a period of 16 hours between Day 1 and 2 where installations cannot be serviced. Hence, it is easy to comprehend the importance of visiting all installations before the significant wave height exceeds 4.5 meters. Thus, the PSVs must speed up to avoid several hours of idling. The most fuel-efficient sailing speed depends on the weather conditions, and lies in the interval between 7 and 9.5 knots. However, from the schedule presented in Figure 3.3, it is evident that the PSVs speed up to perform all deliveries in advance of the poor weather conditions. While returning to the base, time is not scarce, and the sailing speed is reduced significantly, reducing emissions and fuel costs. *PSV0* is then available again at Day 2 since it arrives before 8 am. *PSV1* sails somewhat faster and manages to return to the supply depot just before supply vessel preparation begins. Thus, it is available again for use Day 1. When returning to the depot, *PSV1* keeps a higher sailing speed than *PSV0*. This is because *PSV1* returns at a time where the weather conditions are nice. Thus, the fuel-efficient sailing speed in this weather state is equal to $9.5kts$. When *PSV0* returns to the supply depot, the weather conditions are poor, yielding a lower fuel-minimizing sailing speed at about $7kts$. Note that efficient planning ensures that the other installations are serviced on the voyage sailed by *PSV1* as long as installation *STA* is closed. Thus, unfavourable idling while waiting for the installation to open is avoided.

The available PSVs manage to satisfy demand within the deadlines given, and no additional spot vessels are needed. Making sure that the demand is satisfied this early also gives some protection, ensuring that the company is no longer dependent on the future uncertainty of the significant wave height rising above the servicing limit, causing disruptions in the supply chain. These disruptions might be easier to prevent than to fix. Thus, by thorough planning and bypassing service jobs scheduled during poor weather conditions, emissions and fuel costs can be significantly reduced.

3.5 Summary

In short, The Operational Supply Vessel Planning Problem with Speed Optimization is the problem of finding the cost-efficient voyages and weather-adapted speed optimized schedules for the PSVs servicing a set of offshore petroleum installations from an onshore supply depot. To deal with the real-case weather conditions, weather forecasts are used to identify time intervals where it is unfavourable to sail and service installations, and exploit the benefits of deciding sailing speed for each leg. This way, it is easier to speed up operations in case of a bad weather. If the current fleet cannot sustain the operations needed, additional PSVs can within short time be hired from the spot market at a higher cost to support the current fleet. The objective is to minimize the total costs of fuel consumption and hiring of external PSVs. Challenging restrictions such as openings hours, load capacity limits, delivery deadlines, weather regulations and others exists and must be taken into account. In this problem, it is aimed to generate new operational plans on a daily basis to account for operational factors like weather and urgent supply requests at installations.

Chapter 4

Mathematical Model

In this chapter, the mathematical model is presented. Section 4.1 elaborates on the modelling approach for the mathematical formulation. In Section 4.2, assumptions and notation for the model is described, followed by the mathematical formulation of the problem.

4.1 Modelling Approach

In this chapter, we present a time-discrete *arc-flow formulation* for The Operational Supply Vessel Planning Problem with Speed Optimization. An arc-flow model describes the problem as a network of nodes and arcs. Here, a node is an installation or the supply depot at a certain point of time. In most arc-flow models for supply vessel planning, an arc is usually defined such that one activity may happen per arc, namely vessel preparation, sailing to an installation or the supply depot, servicing an installation or idling. The problem is thus solved by finding the cheapest feasible combination of arcs through the network. As stated in Moan and Ødeskaug (2019), discretizing time allows arcs to be defined between two time points, and thus easily handle time-dependent constraints with arc pruning. This section elaborates on decisions made in the development of the formulation.

4.1.1 Handling the Non-Linearities Introduced by Weather and Speed Optimization

The OSVPPSO introduces a number of non-linearities which complicates the mathematical formulation. When optimizing for sailing speed with maritime vessels, it is important to take into account that when speed increases, the fuel consumption increases polynomially. This means that the main driver of costs in the OSVPPSO is a non-linear fuel consumption function. Some problems also arise because the sailing time of a given sailing leg is dependent on the sailing speed chosen, which thus complicates the steps in the arc-flow formulation. Further, the weather impact on fuel consumption is also dependent on how weather evolves over time and the sailing speed of the vessel. Depending on how the problem is formulated, the sailing time per sailing leg might also become a non-linear function of which distance is sailed and which speed is chosen.

To overcome these challenges, time is made discrete. Thus, any sailing leg between two installations starting at a given point in time can have many different arrival times dependent on the speed selected. Discretization of time thus yields a significant increase in the number of arcs used to model the entire solution space. The advantage is however that discrete time makes it possible to handle all the non-linearities outside the linear program. This is because sailing speeds become a function of the departure time and the chosen arrival time. Thus, fuel consumption can be calculated exactly as this modelling approach make it possible to plot the sailing speed per arc directly into the fuel consumption curve and adjust for the weather conditions per hour sailed.

Weather conditions, shown through significant wave height, affect several facets of the problem presented in this thesis. As the significant wave height increases, the fuel consumed by a PSV increases drastically. It also restricts the maximum permitted sailing speed for a PSV. In addition to this, the time spent on servicing an installation increases as the significant wave height increases, and when the significant wave height exceeds an upper safety regulation limit, it is prohibited to perform service. How weather affects these aspects of the problem is described below.

The way of dealing with weather in this thesis, is similar to the one introduced by Halvorsen-Weare and Fagerholt (2011). This technique has also been used by Norlund and Gribkovskaia (2017). It is assumed that weather conditions can be generalized as a weather state that directly adds resistance to the vessel without any regard for wave or wind directions. Norlund and Gribkovskaia (2017) show that there is a correlation between both service time and sailing speed and a generalised weather state defined by significant wave height. As this information is used to calculate fuel consumption for a given speed and weather state, we define two assumptions based on the findings of Halvorsen-Weare and Fagerholt (2011). First, it is assumed that the maximum limit on vessel speed is reduced by the same amount as Halvorsen-Weare and Fagerholt (2011) found for a given weather state. Second, it is assumed that when calculating the fuel consumption for a specific sailing speed and weather state, it can be calculated by adding a weather offset to the sailing speed and feed it into the fuel consumption function. This means that for the different weather states, fuel consumption is handled by calculating the consumption of the original speed the PSV sails plus an additional offset depending on how bad the weather is. In other words, if a PSV sails at speed v in rough weather, fuel consumption for speed $v + x$ calculated, where x is the offset added by the weather state.

Weather conditions are divided into four different states, depending on the significant wave height. Table 4.1 shows the impact on operations for each of the weather states defined.

Table 4.1: Impact by weather states, inspired by Halvorsen-Weare and Fagerholt (2011)

Weather state	Service possible	Increased service time	Reduced maximum speed limit	Increased fuel consumption (sailing)	Increased fuel consumption (idling & servicing)
0	✓	-	-	-	-
1	✓	✓	-	-	✓
2	✓	✓	✓	✓	✓
3	-	-	✓	✓	✓

Data for each weather state and the fuel consumption functions used in this thesis are provided in Section 7.2. Note that for simplicity, it is assumed that the weather conditions are homogeneous over the entire area of operations.

4.1.2 Reducing Variables in the Arc-Flow Model

An unmodified arc-flow model for the OSVPPSO could have four different types of arcs, i.e. one for each activity, as listed below:

- Arc for handling the supply depot preparation
- Arc for handling the sailing leg between two installations

- Arc for handling idling at an installation (i.e. waiting time)
- Arc for handling the service job performed at an offshore installation

Not only would this lead to the creation of many variables, but would also be likely to generate high complexity as these variables could be combined in many different ways to construct feasible solutions.

In order to reduce the number of variables, information from all these arcs are combined into one arc for each sailing leg. This way each arc holds information about everything that happens between each time a vessel departs from an installation. Thus, some assumptions are made: *1) A vessel will in any optimal solution commence a voyage immediately after preparation at the supply depot. 2) A vessel will only idle at an installation if it has to wait for better weather or for the installation to open in an optimal solution*. This implies that a vessel always will, upon arrival, commence the service job if possible and be ready to sail to the next installation as soon as the service job is finished. The reasoning behind this assumption is that it will always be more cost-efficient to idle at the supply depot than idling offshore, having to battle currents and weather. Thus, it is assumed that in any optimal solution, a vessel will try to minimize the time spent doing idling and servicing activities offshore as it will always be more cost-efficient to spend that time at the depot.

This means that any arc in the arc-flow model contains information about a vessel on a voyage departing from the supply depot or an installation at a point of time to the point of time where the vessel arrives at the supply depot or is ready to depart from the next installation. This means that each arc (also referred to as *sailing leg*), consists of *a potential vessel preparation time at the depot, a sailing time between the installations, a potential waiting time until the installation can be serviced and a service time*. Combining information about preparation time, sailing time, waiting time and service time into one arc type, has to our knowledge not been implemented in any earlier models.

A visualization of the arc definition is shown in Figure 4.1, where information about the preparation time, sailing time, waiting time and service time at installation *j* is embedded in the arc from node *(i,t)* to *(j, t')*. The figure shows a vessel departing from installation *i* at time *t*. The vessel sails to installation *j*, waits until the installation can be serviced and then performs the service job. Thus, at time *t'*, the vessel is finished servicing installation *j* and ready for a new sailing leg. Therefore, *t'* is referred to as the finishing time at installation *j*.

Note that supply depot preparation time, waiting time and service time are only potential

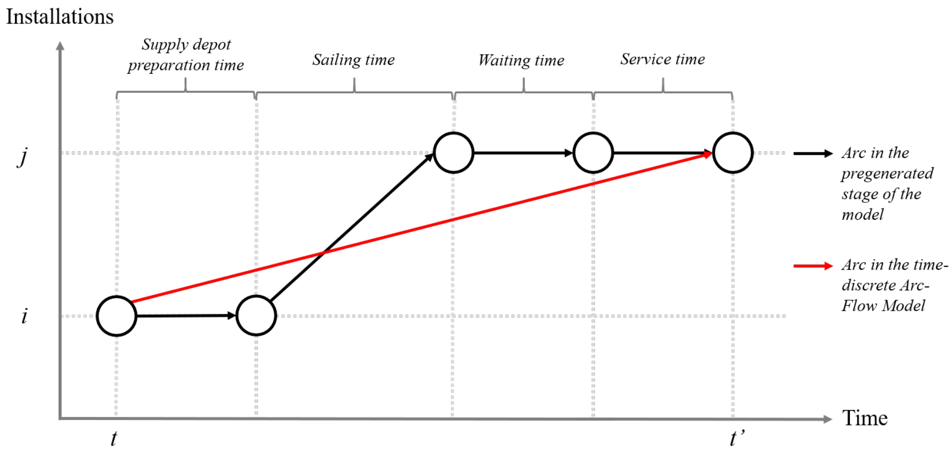


Figure 4.1: An arc in the time-discrete arc-flow model

parts of the arcs. This is because only sailing legs starting at the supply depot will have preparation time. Only vessels forced to wait by either weather or closed installations will have waiting time and sailing legs ending at the supply depot will not have any service time.

To make sure that only feasible arcs are included in the mathematical model, an *arc-generation procedure* is executed. This procedure is described in Section 4.1.3.

4.1.3 Arc-Generation Procedure

The *arc-generation procedure* is used to build a network for each vessel, consisting of all the feasible arcs the vessel might sail in a specific scenario. The network generated by the arc-generation procedure is then sent into the mathematical model for voyage and speed optimization. The arc-generation procedure handles a number of the restrictions faced by the OSVPPSO, so that the mathematical model is simplified.

The arc-flow model presented in Section 4.2 is subject to constraints on meeting the delivery deadline for each installation, Equation (4.4), and maximum voyage duration for specific vessels, Equation (4.6). These constraints are in practice taken care of in the arc-generation procedure, where arcs violating these constraints are not added to the network. However, they are included in the arc-flow formulation to give a more holistic understanding of the problem. These constraints play an import role when the OSVPPSO is solved using the genetic algorithm presented in Chapter 6.

Figure 4.2 illustrates how the arc-generation procedure and the arc-flow model interact.

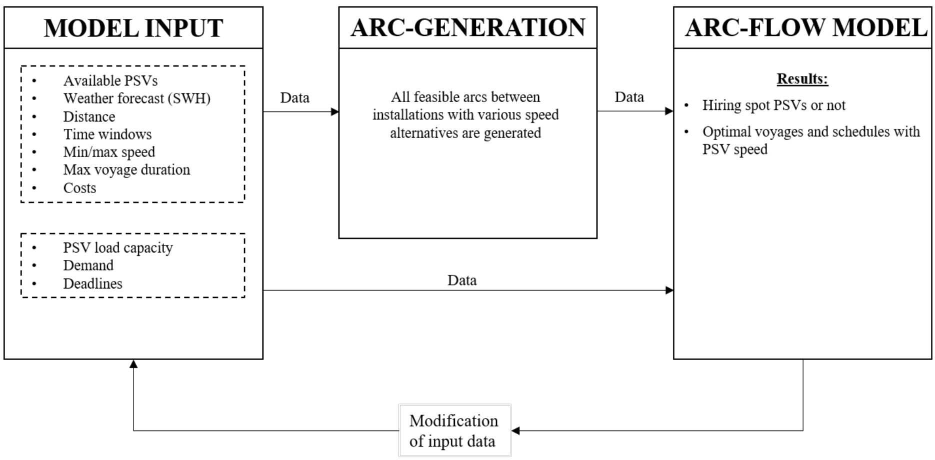


Figure 4.2: Schematic overview of solution method

Overview of the Arc-Generation Procedure

The arc-generation procedure is run for all vessels, both spot vessels and the fleet of available PSVs, so that every vessel has its own network of arcs. First, an algorithm decides which nodes the vessel might sail from. For each of those nodes, a second algorithm decides which nodes the vessel may sail to. When generating arcs, arcs are only generated from feasible departure times for each vessel and each installation. Further, arcs are only generated from an installation at the time points (i.e. a node) where another arc ends at that installation. To keep track of the set of feasible departure times for each installation and vessel, nodes are created for the time, installation and vessel at the end of each arc generated. Thus, if a node with time t, installation i and vessel v exists, generation of arcs from that node is allowed. The arc-generation procedure is described in Algorithm 1.

Algorithm 1 Arc-Generation Procedure

```
procedure GenerateAllArcs()
    for vessel v ∈ V do                                        ▷ all available vessels
        Nodes ← node(t^{startPrep}, depot, v)                  ▷ add depot node to the set of nodes
        for t ∈ [t^{StartPrep}, t_v^{MaxDur}] do               ▷ all discrete times for vessel v
            for installation i ∈ N do                          ▷ all installations and depot
                if node(t, i, v) ∈ Nodes then
                    BuildArcsFromNode(t, i, v)                 ▷ See Algorithm 2
                end if
            end for
        end for
    end for
end procedure
```

To know the span of theoretical feasible end times for arcs between two installations, two time points are calculated; the earliest theoretical end time and the latest theoretical end time. For every discrete point in time between these times, starting at the earliest theoretical end time, the arc-generation procedure attempts to build an arc for every discrete time point between the start time and that end time. If it is possible to create a feasible arc between those time points, it is created and added to the network. After each iteration time is incremented until all theoretical end times are considered. The pseudocode for this procedure is described in Algorithm 2.

Algorithm 2 Arc-Generation From Node

procedure $BuildArcsFromNode(t^{start}, i^{dep}, v))$
  for $i^{dep} \in AllInstallationsWithDepot$ **do**
    $t^{max} \leftarrow$ *latest theoretical end time for the arc*
    $t^{end} \leftarrow$ *earliest feasible end time*
    while $t^{end} \leq t^{max}$ **do**
      $t^{Service}, C^{Service} \leftarrow$ *CALCULATESERVICING*() ▷ Appendix A.1
      if there $distance/(t^{end} - t^{start}) > max\_speed$ given weather **then**
        increment t^{end} *and restart while loop*
      end if
      $t^{Arrival}, C^{Idling} \leftarrow$ *CALCULATEIDLING*() ▷ Appendix A.1
      $C^{Sailing} \leftarrow$ *CALCULATESAILING*() ▷ Appendix A.1
      $Network \leftarrow$ *Add* $arc(t^{start}, t^{end}, v, i^{dep}, i^{dest}, C^{Sail}, C^{Idle}, C^{Service})$
      $Nodes \leftarrow$ *Add* $node(t^{end}, i^{dest}, v)$
      increment t^{end}
    end while
  end for
end procedure

There are many ways one of these arcs might be infeasible. The destination installation might be closed for service by either deadline restrictions, closing times for the installation or poor weather conditions. These cases are handled in the *CALCULATESERVICING*() algorithm. There might also not be enough time for the vessel to sail to, and service the installation in time. This is handled in the *CALCULATEIDLING*() algorithm.

Illustrative Example

In Figure 4.3, the network of nodes and arcs for a given vessel is shown. Each arc has an assigned cost, but these are not shown in the network representation. To generate this network, three offshore installations are considered. Every discrete time point is a quarter of an hour apart. As with all network representations for our model, the supply depot is denoted *Installation 0*. Note that the different speed variations can easily be spotted for the sailing leg from the supply depot to *Installation 1* from the first discrete time point. Also, note the large gap between the nodes in the middle of the time horizon. This gap is a result of a period with poor weather conditions. During this period, no installation can be serviced.

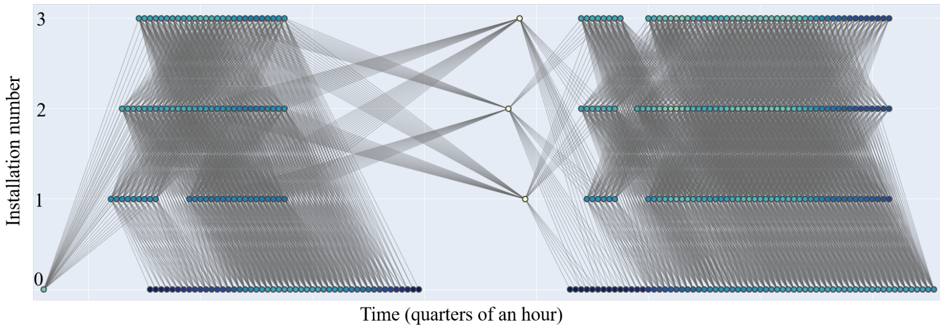


Figure 4.3: A fully generated network for an instance with 3 offshore installations

Figure 4.3 shows the network that is sent into the mathematical model for all available PSVs. The mathematical model finds the cheapest combination of arcs through the network for all available PSVs such that all constraints in the mathematical model are satisfied. The mathematical model is presented in the next section, Section 4.2.

4.2 Arc-Flow Model

4.2.1 Assumptions

- An optimal solution will never have a vessel wait before servicing an offshore installation unless it is restricted to do so by the installation's opening hours or poor weather conditions. Hence, it is assumed that for those vessels that arrive when the installation is open for service and the significant wave height is lower than the safety limit, service of the installation will commence immediately.
- For a PSV to be prepared for a voyage in time, the latest time of arrival at the supply depot must be before the supply depot opens that day.
- The capacity of the installations is not exceeded by any deliveries. Deliveries are frequent and provide small batches of supplies.
- Because only the voyages for the next departure is considered in this problem, the time horizon Until the vessels will return to the depot is short enough that the weather is assumed to be deterministic.

4.2.2 Notation

Sets & Indices

$\mathcal{N}$	-	Set of all onshore and offshore installations i in the problem instance. Including both the supply depot & all offshore installations to visit on the voyages starting the next departure day
$\mathcal{T}$	-	Set of discrete time points until the latest return time of any vessels departing the next departure day (e.g. quarters of an hour)
$\mathcal{T}^{D}_{ijv}$	-	Subset of $\mathcal{T}$ with all possible departure times for a vessel v sailing from installation i to installation j. Visualized in Figure 4.4
$\mathcal{T}^{SD}_{ijtv}$	-	Subset of $\mathcal{T}$ with all possible departure times for a vessel v sailing from installation i and finishing at installation j at time t. Visualized in Figure 4.5
$\mathcal{T}^{SF}_{itjv}$	-	Subset of $\mathcal{T}$ with all possible finishing times for a vessel v sailing from installation i at time t to installation j. Visualized in Figure 4.6
$\mathcal{G}_v$	-	The network containing all legal arcs, i.e. sailing legs, from installation i to j at all departure times t for a vessel v. Illegal arcs resulting from closing hours at installations or weather prohibiting installation service are not added to the network
$\mathcal{V}$	-	Set of all vessels available the next departure day, including vessels from the spot market

Note that all sets provided in the table above are zero indexed. This means that the first vessel is represented by the number 0. Also note that in $\mathcal{N}$, the supply depot is represented as 0, while installations are represented by non-zero numbers.

Figures 4.4 to 4.6 illustrate the time subsets defined above. The figures show nodes and arcs, illustrating which discrete time points are included in each set. In these figures, grey nodes are not relevant in the given set, and red nodes denote the nodes that reside at the time periods that should be included in the set.

Figure 4.4 illustrates the set of possible departure times, $t \in \mathcal{T}^{D}_{ijv}$, where a vessel v can commence the sailing leg from installation i to j. The red nodes indicate legal time points when vessel v may depart, i.e t_1, t_4 and t_5. If a vessel v departs from installation i to j

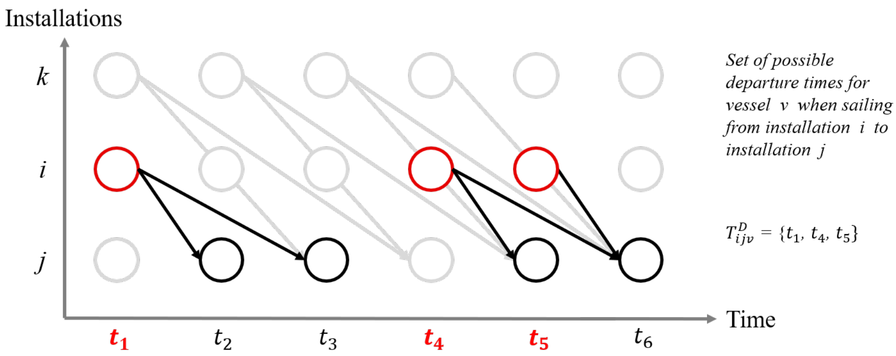


Figure 4.4: Visualization of $\mathcal{T}_{ijv}^{D}$

at a time point not included in the set, i.e. t_2, t_3 or t_5, the installation j will be closed at the arrival of vessel i, either due to closing hours, rough weather conditions or other time constraints. The vessel will thus face undesirable waiting time, and therefore, these arcs are not included in the set.

Figure 4.5 shows a set of time points containing more specific information than the set $\mathcal{T}_{ijv}^{D}$ presented in Figure 4.4 above. In this set, a specific finishing time, meaning the time point where the vessel is ready to depart from the next installation, is provided. The set $\mathcal{T}_{ijtv}^{SD}$ provides the possible time points a vessel v may depart from installation i in order to finish servicing installation j at the specified finishing time t, illustrated as red nodes in Figure 4.5. Figure 4.5 shows that in order to be able to finish service at installation j at time t_6 when sailing from installation i, vessel v must depart either at time t_4 or t_5. Note that the number of time points in this set is equal to the number of speed alternatives for the vessel on this sailing leg.

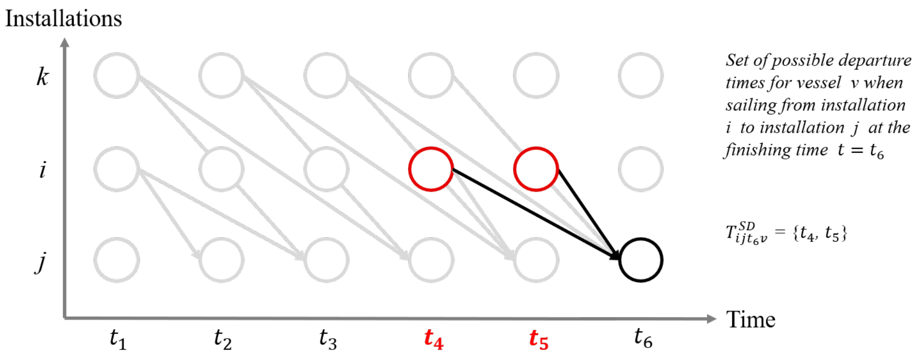


Figure 4.5: Visualization of $\mathcal{T}^{SD}_{ijtv}$

In contrast to Figure 4.5, which shows the possible departure time points a vessel v may depart from installation i in order to finish service at installation j at time t, Figure 4.6 provides possible finishing times. $\mathcal{T}^{SF}_{itjv}$ indicates, in red nodes, the possible time points a vessel v may finish the service at installation j when departing from installation i at time t. Thus, Figure 4.6 shows that a vessel v departing from installation i at time t_1 may finish servicing installation j at time t_2 or t_3. Note that the number of time points in this set is equal to the number of speed alternatives for the vessel on this sailing leg.

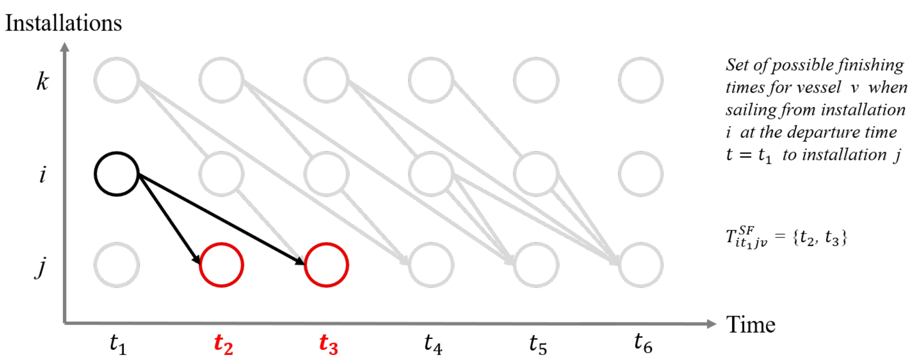


Figure 4.6: Visualization of $\mathcal{T}^{SF}_{itjv}$

Parameters

D_i	-	Demand at installation i, i.e. the number of standard unit containers to be delivered to installation i on a voyage departing the next day
T_i^D	-	Deadline for a delivery to installation i
$\overline{T}_v$	-	Maximum duration of a voyage sailed by vessel v, i.e. the number of discrete time points before vessel v must be docked at the supply depot after starting a voyage. This is equivalent to the number of discrete time points until the return time
Q_v	-	Cargo capacity of supply vessel v, i.e. maximum number of standard unit containers
$C^F_{itjt'v}$	-	Cost of fuel consumed when sailing from installation i at time t to installation j, finishing the service job at time t'
$C^V_{tt'v}$	-	Chartering cost of the vessel for the time period of the arc, when sailing with vessel v from time t to time t'. For a vessel hired on a long-term contract, this cost will be equal to zero, whilst a spot vessel will have a chartering cost

Decision Variables

$$x_{itjt'v} = \begin{cases} 1 & \text{if vessel } v \text{ travels along the arc where it starts at installation } i \text{ at time } t, \\ & \text{then sails to installation } j \text{, finishing service at installation } j \text{ at time } t' \\ 0 & \text{otherwise} \end{cases}$$

4.2.3 Mathematical Formulation

Objective

The objective function in Equation (4.1) minimizes the overall costs of vessels' fuel consumption and chartering vessels from the spot market for a given period.

$$min\ z = \sum_{v \in V} \sum_{((i,t),(j,t')) \in \mathcal{G}_v} (C^F_{itjt'v} + C^V_{tt'v}) x_{itjt'v} \tag{4.1}$$

Constraints

Constraints (4.2) ensure that voyages have a legal format. Voyages should start at the depot, only visit any specific installation once along the voyage and finish back at the depot. These constraints ensure *flow conservation* and make sure that an arc entering a node have a subsequent arc leaving the same node. This means that when a vessel finishes a service job at an installation at certain time, it is also required to depart from the same installation at that same point in time. These constraints are not applicable for the supply depot, which is the only location allowed to have only one outgoing (beginning of a voyage) or incoming arc (end of a voyage).

$$\sum_{j \in \mathcal{N}} \sum_{t' \in \mathcal{T}^{SD}_{jitv}} x_{jt'itv} - \sum_{j \in \mathcal{N}} \sum_{t' \in \mathcal{T}^{SF}_{itjv}} x_{itjt'v} = 0, \qquad i \in \mathcal{N} \backslash \{0\}, t \in \mathcal{T}, v \in \mathcal{V} \tag{4.2}$$

Constraints (4.3) - (4.5) handle installation visits, visit deadlines and vessel capacity. Constraints (4.3) ensure that *all installations are visited*, and thus receive their delivery. These constraints also ensure that an installation is only visited once. Constraints (4.4) make sure that *all deliveries are performed within the deadline* set for specific installations. Furthermore, Constraints (4.5) make sure that the total size of cargo delivered by a PSV on a voyage does not exceed the *PSVs load capacity*.

$$\sum_{v \in \mathcal{V}} \sum_{i \in \mathcal{N}} \sum_{t \in \mathcal{T}^{D}_{ijv}} \sum_{t' \in \mathcal{T}^{SF}_{itjv}} x_{itjt'v} = 1, \qquad j \in \mathcal{N} \setminus \{0\} \tag{4.3}$$

$$\sum_{i \in \mathcal{N}} \sum_{t \in \mathcal{T}^{D}_{ijv}} \sum_{t' \in \mathcal{T}^{SF}_{itjv}} t' x_{itjt'v} \leq T^{D}_{j}, \qquad j \in \mathcal{N} \setminus \{0\}, v \in \mathcal{V} \tag{4.4}$$

$$\sum_{((i,t),(j,t')) \in \mathcal{G}_v} D_j x_{itjt'v} \leq Q_v, \qquad v \in \mathcal{V} \tag{4.5}$$

Constraints (4.6) are in place to make sure that the duration of a voyage does not exceed a set maximum duration. In these constraints, a sailing leg is not allowed to finish after the maximum voyage duration has passed. Thus, these constraints make it easy to ensure a specific PSV v is available at a desired point of time in the future.

$$\sum_{i \in \mathcal{N}} \sum_{t \in \mathcal{T}^{D}_{ijv}} \sum_{t' \in \mathcal{T}^{SF}_{itjv}} t' x_{it0t'v} \leq \overline{T}_v, \qquad v \in \mathcal{V} \tag{4.6}$$

Constraints (4.7) ensure binary requirements for the decision variable, where an arc either is selected or not.

$$x_{itjt'v} = \{0, 1\}, \qquad ((i,t),(j,t')) \in \mathcal{G}_v, v \in \mathcal{V} \tag{4.7}$$

Again, note that in practice, the constraints on meeting delivery deadlines, Constraints (4.4), and maximum voyage duration, Constraints (4.6), are handled in the arc-generation procedure where arcs violating these constraints are not added to the network. However, they are kept in the arc-flow model formulation to give a better description of the problem at hand. These constraints are also referred to in the metaheuristic described in Chapter 6.

Chapter 5

The Weather-Dependent Supply Vessel Speed Optimization Problem

This chapter describes the *Weather-Dependent Supply Vessel Speed Optimization Problem*. Section 5.1 describes the mathematical formulation of the continuous-time problem, while Section 5.2 elaborates on how time can be discretized to reduce the complexity of the problem. Further, Section 5.3 suggests a solution method for the time-discrete problem.

The *Weather-Dependent Supply Vessel Speed Optimization Problem (WDSVSOP)* involves assigning weather-adapted sailing speeds to sailing legs along a predefined voyage sailed by a specific vessel and determine the duration of idling and servicing activities along each leg. In Chapter 6, a metaheuristic named the *Hybrid Genetic Search with Adaptive Diversity Control (HGSADC)* is proposed for the OSVPPSO. While the arc-flow formulation described in Chapter 4 decides a voyage and the sailing speed for each sailing leg on that voyage simultaneously, the HGSADC considers a voyage and further evaluates the voyage by the solution obtained from the WDSVSOP. Hence, the WDSVSOP is treated as a subproblem in the HGSADC.

As the weather, shown through significant wave height, changes with time, the fuel consumption for various operations changes accordingly. This, and all other impacts on operations imposed by weather conditions make the problem operational. These factors thus need to be accounted for when solving the WDSVSOP.

Note that the weather forecast is provided for discrete time intervals. While the arc-flow formulation in Chapter 4 may plan with discrete time intervals of e.g. a quarter of an hour, the weather forecast may provide significant wave height with a different time interval, e.g. every hour. Thus, the time discretization for the weather forecast is referred to as the *weather forecast discretization*.

5.1 Mathematical Formulation of The Weather-Dependent Supply Vessel Speed Optimization Problem

5.1.1 Sets

$\mathcal{L}$	-	Ordered set of sailing legs l in a voyage
$\mathcal{A}$	-	Set of activities, a, that may be performed by a vessel on any sailing leg, i.e. supply depot preparation, sailing, idling, and servicing. Shown in Table 5.1
$\mathcal{W}$	-	Set of weather states, w, as described in Table 4.1
$\mathcal{T}^{WF}$	-	Set of discrete time periods t for the weather forecast (e.g. hours). It is assumed that a solution exceeding the length of the weather forecast is of no interest. Spot vessels can be added in the master problem to obtain shorter voyages, eliminating this problem

Table 5.1: Set of activities, $\mathcal{A}$

Activity, a	Description
0	Supply depot preparation
1	Sailing
2	Idling (waiting)
3	Servicing an installation
4	Artificial activity denoting ”end of service,” i.e. that a vessel is ready to depart for the next sailing leg

Note that all sets are zero indexed. E.g. the first weather state is weather state 0, and the first activity, namely preparation is activity 0.

5.1.2 Parameters

I_{aw}^{FC}	-	Impact on fuel consumption on activity a from weather state w
I_{aw}^{T}	-	Impact on time needed to perform activity a in weather state w
I_{w}^{MS}	-	Impact on the maximum sailing speed limit for the vessel in weather state w
δ_{tw}^{WS}	-	Binary parameter equal to 1 if the weather state is equal to weather state w at time t, 0 otherwise
δ_{lt}^{SI}	-	Binary parameter equal to 1 if service on sailing leg l is infeasible during time period t, 0 otherwise. For this parameter service infeasibility may be a result of a closed installation or poor weather conditions
δ^{Spot}	-	Binary parameter equal to 1 if the vessel used is a spot vessel, 0 otherwise
$\overline{T}$	-	Latest time at which the vessel should be returned at the supply depot, i.e. the maximum voyage duration. This limit can be violated at a penalty cost per unit violated
T_{l}^{D}	-	Deadline for the delivery to the installation serviced on sailing leg l. This limit can be violated at a penalty cost per unit violated
$T_{l}^{Service}$	-	Time needed to perform the service job for the installation visit on sailing leg l under perfect weather conditions
C_{a}	-	Monetary cost of fuel per time unit for activity a
$C^{Charter}$	-	Monetary cost of chartering a spot vessel per time unit
$C^{Pen,\overline{T}}$	-	Penalty for violating the maximum duration per time unit
$C^{Pen,T^{D}}$	-	Penalty for violating a delivery deadline per time unit
D_{l}	-	Distance on sailing leg l
V^{Max}, V^{Min}	-	Maximum and minimum limits on vessel speed, respectively

5.1.3 Decision Variables

On each sailing leg $l \in \mathcal{L}$ it must be determined when activity a should start. Thus, the decision variable is defined as

τ_{la} - Continuous decision variable denoting at which time activity a starts on sailing leg l

5.1.4 Helping Variables

Both fuel consumption and constraints change with weather states. As the weather forecast is known for discrete time periods, it is necessary to know how much time an activity spends in each discrete time period of the weather forecast. The *time participation variable*, x_{lat}, is introduced to keep track of this. The continuous time participation variable denotes the proportion of the discrete time period starting at time t that is spent doing activity a on the sailing leg l. Thus,

$$x_{lat} = \begin{cases} 0 & \text{if } t < \lfloor \tau_{la} \rfloor \quad \text{or} \quad t \geq \lceil \tau_{l(a+1)} \rceil \\ t - \tau_{la} + 1 & \text{if } t = \lfloor \tau_{la} \rfloor \quad \& \quad t! = \lfloor \tau_{l(a+1)} \rfloor \\ \tau_{l(a+1)} - \tau_{la} & \text{if } \lfloor \tau_{la} \rfloor = \lfloor \tau_{l(a+1)} \rfloor \\ \tau_{l(a+1)} - t & \text{if } t = \lfloor \tau_{l(a+1)} \rfloor \quad \& \quad t! = \lfloor \tau_{la} \rfloor \\ 1 & \text{if } t < \lfloor \tau_{l(a+1)} \rfloor \quad \& \quad t \geq \lceil \tau_{la} \rceil, \end{cases} \tag{5.1}$$

$$l \in \mathcal{L} \setminus \{|\mathcal{L}|\}, a \in \mathcal{A} \setminus \{4\}, t \in \mathcal{T}^{WF}$$

Figure 5.1 illustrates how the time participation variable keeps track of the proportion of time spent in discrete time periods.

Further, because the maximum speed limit and consumption function of each vessel varies with the weather state, it is necessary to know how to distribute sailing speed to the different weather states. This again denotes a subproblem within the WDSVSOP, namely *how to distribute sailing speeds along different fuel consumption curves on a given sailing leg*. For practical purposes, it is assumed that it is preferred to have as little speed variance as possible over the sailing leg. Thus, minimal adjustments to sailing speeds in the discrete time intervals has to be done in order to stay within the maximum vessel speed limit for each weather state. This means that the optimization of speeds in the different weather

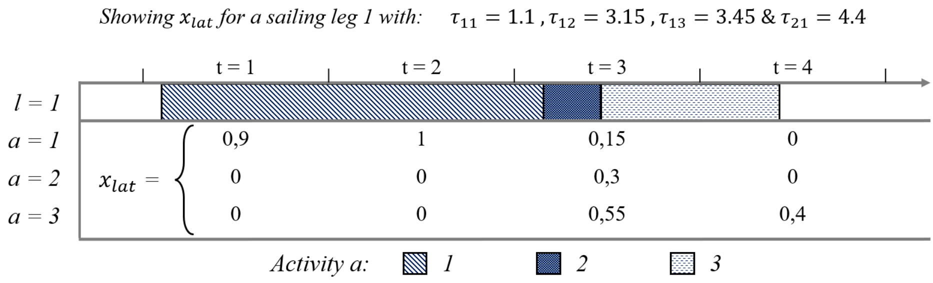


Figure 5.1: Visualization of x_{lat}

scenarios for a given sailing leg is in this formulation not considered. This is because it introduces unnecessary complexity, at a level of detail that will likely not give a large impact on objective values. The sailing speeds will however be adjusted to comply with maximum and minimum speed limits for the different weather states, To determine the sailing speed in these discrete time intervals, the portion of the sailing time that is being spent in the different weather scenarios must be known. Because the total duration of the sailing leg is known, this can be done by dividing the sum of the time participation variables multiplied with the binary weather state parameters δ_{tw}^{WS} on the total duration of the sailing leg. Thus, the variable y_{law} is introduced and denotes the portion of the activity a on sailing leg l spent in weather state *w*.

$$y_{law} = \sum_{t \in \mathcal{T}^{WF}} \frac{\delta_{tw}^{WS} x_{lat}}{\tau_{l(a+1)} - \tau_{la}}, \qquad l \in \mathcal{L}, a \in \{1, 3\} \tag{5.2}$$

The desired average speed, v_l^{*0} for sailing leg l may be calculated as

$$v_l^{*0} = \frac{D_l}{\tau_{l2} - \tau_{l1}}, \qquad l \in \mathcal{L} \tag{5.3}$$

In cases where the significant wave height increases over time, it might not be possible to sustain the average sailing speed through the whole voyage. In this case, the vessel has to sail at a higher speed when the weather allows. In Figure 5.2, an example of a sailing leg l with 4 discrete time periods is presented. In this example, the desired average speed is set to $v_l^{*0} = 12.5kts$. The vessel is exposed to different weather states along the voyage, and the example shows that the vessel in time period $t = 3$ and $t = 4$ is subject to a reduced maximum speed limit. Thus, in order to obtain the average desired speed $v_l^{*0} = 12.5kts$, the sailing speed at time $t = 1$ and $t = 2$ must to be above the average speed.

Figure 5.2 illustrates how the speed is adjusted in discrete time periods with different weather states on a sailing leg l to obtain the desired average sailing sailing speed

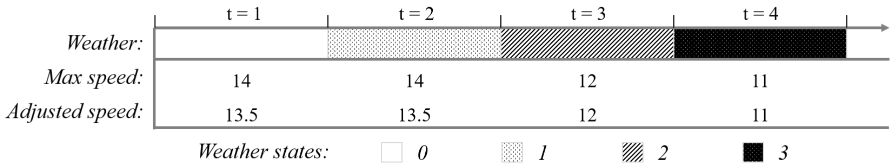


Figure 5.2: Example of how the weather-adjusted speed is increased to catch up on the time lost in *weather states 2* and *3*

In order to calculate how much the sailing speed in the non-restricted weather states must be increased with to obtain the desired average sailing speed, it is necessary to check if the desired average sailing speed is higher than the maximum vessel speed limit for the specific weather state, and then adjust it to the required sailing speed. These adjustments are performed in Equation (5.4) for the most restrictive weather state first, i.e *weather state 3*. Then, the new weather-adjusted sailing speed is used to do the same check and adjustment in Equation (5.5) in the next, less restrictive weather state, i.e. *weather state 2*.

$$v_l^{*1} = \begin{cases} v_l^{*0} + \frac{(v_l^{*0} - (V^{Max} - I_3^{MS}))y_{l13}}{1 - y_{l13}} & \text{if } v_l^{*0} > V^{Max} - I_3^{MS} \\ v_l^{*0} & \text{otherwise,} \end{cases} \qquad l \in \mathcal{L} \tag{5.4}$$

$$v_l^{*2} = \begin{cases} v_l^{*1} + \frac{(v_l^{*1} - (V^{Max} - I_2^{MS}))y_{l12}}{1 - (y_{l12} + y_{l13})} & \text{if } v_l^{*1} > V^{Max} - I_2^{MS} \\ v_l^{*1} & \text{otherwise,} \end{cases} \qquad l \in \mathcal{L} \tag{5.5}$$

After making adjustments to the sailing speed, the sailing speed v_{lt} for each sailing leg l, and discrete time period t in the weather forecast can be found with

$$v_{lt} = \begin{cases} V^{Max} - I_3^{MS} & \text{if } v_l^{*2} > V^{Max} - I_3^{MS} \\ V^{Max} - I_2^{MS} & \text{if } v_l^{*2} > V^{Max} - I_2^{MS} \\ v_l^{*2} & \text{otherwise,} \end{cases} \quad l \in \mathcal{L}, t \in \mathcal{T}^{WF} \tag{5.6}$$

In other words, each of the terms in Equation (5.6) denotes the speed a vessel should keep in each weather state at time t on the sailing leg l. The first, second and third term denotes the speed that should be used in *weather state 3, 2* and *0&1* on sailing leg l, respectively.

5.1.5 Objective

Since the decision variables τ_{la} have been converted to time participation variables x_{lat} for each activity $a \in \mathcal{A}$ on each sailing leg $l \in \mathcal{L}$ in a voyage, the objective function can be formulated as

$$\begin{aligned} min\ z = &\sum_{l \in \mathcal{L}} \left(f^{Prep}(l) + f^{Sail}(l) + f^{Idle}(l) + f^{Service}(l) \right) \\ &+ \sigma^{\overline{T}} \\ &+ \sigma^{T^D} \\ &+ C^{Charter} \delta^{Spot} (\tau_{|\mathcal{L}|4} - \tau_{00}) \end{aligned} \tag{5.7}$$

where the sub-objectives $f^{Prep}(l)$, $f^{Sail}(l)$, $f^{Idle}(l)$ and $f^{Service}(l)$, denote the cost of fuel consumed from preparation at the supply depot, sailing, idling and servicing on a sailing leg l, respectively. $\sigma^{\overline{T}}$ and σ^{T^D} denote the total penalty cost of violating the maximum voyage duration and the delivery deadline constraints. The last term represents the chartering cost if the vessel used is a spot vessel. Recall that fuel consumption for sailing in rough weather is found by calculating the fuel consumption of the actual sailing speed plus an offset, depending how high the significant wave height is. The sub-objectives are given by

$$f^{Prep}(l) = \sum_{t \in \mathcal{T}^{WF}} C_0 x_{l0t} \tag{5.8}$$

$$f^{Sail}(l) = \sum_{t \in \mathcal{T}^{WF}} g(v_{lt} + I_w^{MS} \delta_{tw}^{WS}) x_{l1t} \tag{5.9}$$

$$f^{Idle}(l) = \sum_{t \in \mathcal{T}^{WF}} I_{2w}^{T} \delta_{tw}^{WS} C_2 x_{l2t} \tag{5.10}$$

$$f^{Service}(l) = \sum_{t \in \mathcal{T}^{WF}} I_{3w}^{T} \delta_{tw}^{WS} C_3 x_{l3t} \tag{5.11}$$

$$\sigma^{\overline{T}} = C^{Pen,\overline{T}} max\{0, \tau_{|\mathcal{L}|4} - \overline{T}\} \tag{5.12}$$

$$\sigma^{T^D} = C^{Pen,T^D} \sum_{l \in \mathcal{L}} max\{0, \tau_{l3} - T_l^D\} \tag{5.13}$$

5.1.6 Problem Specific Cost Function

The fuel consumption function for sailing can be described by a quadratic function, where p_1, p_2, p_3 are the coefficients. How this fuel consumption function was determined is described in Section 7.2.2. The general form is given by

$$g(v) = p_1 v^2 + p_2 v + p_3, \tag{5.14}$$

where $g(v)$ gives the fuel consumption per time unit (e.g. $kg/hour$).

5.1.7 Constraints

Constraints (5.15) ensure that for each sailing leg, the activities occur in the right sequence, namely supply depot preparation, sailing, idling and servicing. If an activity is not present in a sailing leg, the time spent on this activity is set to zero, and the next activity present is immediately initiated. Further, Constraints (5.16) make sure that the end time for a sailing leg is identical as the start time for the next sailing leg.

$$\tau_{la} \leq \tau_{l(a+1)}, \qquad l \in \mathcal{L}, a \in \mathcal{A} \setminus \{4\} \tag{5.15}$$

$$\tau_{l4} = \tau_{(l+1)0}, \qquad l \in \mathcal{L} \setminus \{\mathcal{L}\} \tag{5.16}$$

Constraints (5.17) and (5.18) ensure that the highest sailing speed on the sailing leg in order to retain the desired average speed does not exceed the maximum and minimum speed limit for the vessel.

$$v_l^{*2} \leq V^{Max}, \qquad l \in \mathcal{L} \tag{5.17}$$

$$v_l^{*2} \geq V^{Min}, \qquad l \in \mathcal{L} \tag{5.18}$$

Constraints (5.19) make sure that service jobs are only performed during the time windows where it is possible to service an installation

$$\sum_{t \in \mathcal{T}^{WF}} \delta_{lt}^{SI} x_{l3t} = 0, \qquad l \in \mathcal{L} \tag{5.19}$$

Constraints (5.20) ensure that the length of the service job equals the time it takes to service the installation on sailing leg l adjusted for the weather impact on service time.

$$\tau_{(l+1)0} - \tau_{l3} = T_l^{Service} \sum_{w \in \mathcal{W}} I_{3w}^T y_{l3w}, \qquad l \in \mathcal{L} \tag{5.20}$$

Constraints (5.21) and (5.22) handle the continuous helping- and decision variables, respectively, and ensure non-negativity constraints and upper bounds.

$$0 \leq x_{lat} \leq 1, \qquad l \in \mathcal{L} \setminus \{|\mathcal{L}|\}, a \in \mathcal{A} \setminus \{4\}, t \in \mathcal{T}^{WF} \tag{5.21}$$

$$0 \leq \tau_{la} \leq |\mathcal{T}^{WF}|, \qquad l \in \mathcal{L}, a \in \mathcal{A} \tag{5.22}$$

5.2 The Weather-Dependent Supply Vessel Speed Optimization Problem as a Shortest Path Problem

The task of assigning sailing speeds to a specific sailing leg is no different than assigning time to the distance to be sailed on that leg. This is true because the distance for any sailing leg is provided and the sailing speed is calculated as distance divided by time. When regarding time as the main resource in the problem, the WDSVSOP may be formulated as a *resource allocation problem*, where time is to be allocated to activities in a sailing leg. However, there is a challenge with this formulation. The cost of a sailing leg is not only dependent on the resource allocated to that leg, but also strongly dependent on the absolute time allocated before that sailing leg. This is because weather states are dependent on absolute time, and not relative time for each sailing leg. Because of this property, along with other problem restrictions, e.g. limited opening hours, delivery deadlines, maximum voyage duration and others, the problem is not only non-convex globally, but also non-convex over each separate sailing leg. Thus, the recursive smoothing algorithm by Norstad et al. (2011) or other exact divide and conquer methods like the MDA presented by Vidal et al. (2019) will not be suitable to solve this problem.

If start time τ_{l0} is also discretized in the subproblem, both the start time, τ_{l0}, and end time, τ_{l4}, of an arc are known. Thus, the weather profile for the whole sailing leg l will also be known. When both the start and end time is known, it is also possible to calculate the departure time, τ_{l1}, arrival time, τ_{l2}, and the time at which service is commenced, τ_{l3}, for each sailing leg. The global problem is still non-convex. However, with this discretization, each sailing leg can be viewed as an arc similar as in the arc-flow model. This way, all feasible arcs may be generated, so that the problem becomes a shortest path problem. Note that the start time for preparation, τ_{l0}, is equal to the departure time, τ_{l1}, if the sailing leg l does not start at the supply depot.

Recall that in the Weather-Dependent Supply Vessel Speed Optimization Problem the order of installation visits in a voyage is provided. The network of arcs and the nodes of installations in the voyage at different times thus form a *Directed Acyclic Graph (DAG)*. This representation for solving maritime speed optimization problems was first introduced by Fagerholt (2001) and has later been used by Fagerholt et al. (2010).

Each arc has a cost associated. If the costs are seen as the distance between nodes, the problem is a shortest path problem. The arcs may then be structured as a *tree* where each level of the tree is a new sailing leg. Because of the simple tree structure of the network, the shortest path may easily be found using a breadth-first search. The tree structure of the DAG may be seen in Figure 5.3. Note that a voyage schedule is set when a path through the graph in Figure 5.3 is found.

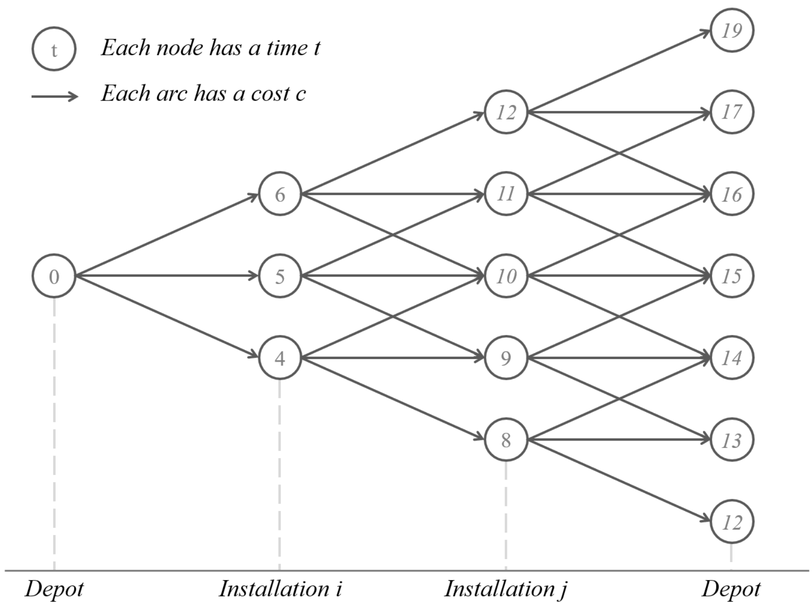


Figure 5.3: *Directed Acyclic Graph:* A network showing a PSV visiting two installations. Each arc has a total duration of 4, 5 or 6 discrete time periods on a leg. This means the PSV can choose between three speed alternatives on each leg

5.3 Using Tree Search to Solve the WDSVSOP

5.3.1 Tree-Specific Notation

A *tree* is a data structure built from nodes and edges. In standard tree theory, an arc between two nodes is defined as an *edge*. For the section describing the tree search, the term *edge* is used instead of *arc* to describe a sailing leg. Practically, the only difference between an arc and an edge is that an edge also holds tree-specific information.

The following sets are used:

$\mathcal{O}$	-	Set of nodes in the tree structure, where a node o is defined as an installation i at a time t
$\mathcal{O}_l$	-	Subset of $\mathcal{O}$, for all nodes where sailing leg $l \in \mathcal{L}$ starts
$\mathcal{E}$	-	Set of edges in the tree structure

Each node $o \in \mathcal{O}$ holds:

$\mathcal{E}_o^{Child}$	-	Set of edges going out of the node, defined as *child edges*
$\mathcal{O}_e^{Child}$	-	Set of child nodes o_e^{Child}, where $e \in \mathcal{E}_o^{Child}$
$e_o^{Parent,Best}$	-	Best parent edge of node o
C_o^{Best}	-	Best total penalized cost of a path to node o

Each edge $e \in \mathcal{E}$ holds:

o_e^{Parent}	-	Node at the start of the edge, defined as the *parent node*
o_e^{Child}	-	Node at the end of the edge, defined as the *child node*
C_e^{Fuel}	-	Fuel cost from all activities performed along the edge
$C_e^{PC,Deadline}$	-	Deadline penalty cost for the service job performed along this edge
$C_e^{PC,Duration}$	-	Duration penalty cost for edges ending at the supply depot after the return time of the vessel
$C_e^{Charter}$	-	Monetary cost of chartering the vessel for the time period. > 0 if the vessel used is a spot vessel, 0 otherwise

Note however that an edge only holds one parent- and child node. Thus, the child node of an edge also becomes the child node of the parent node of that edge. This holds both ways. Thus, the term *child node* is used for the child nodes of all of the nodes' child edges.

5.3.2 Node Expansion

When a node o is expanded, all child edges $e' \in \mathcal{E}_o^{Child}$ and child nodes $o' \in \mathcal{O}_{e'}^{Child}$ are evaluated. For each child node o', a temporary cost parameter $C_{o'}^{Current}$ is assigned the total penalized cost of travelling from node o to node o' through edge e'. This is calculated as

$$C_{o'}^{Current} = C_o^{Best} + C_{e'}^{Fuel} + C_{e'}^{PC,Deadline} + C_{e'}^{PC,Duration} + C_{e'}^{Charter} \tag{5.23}$$

This cost is then compared to the current best cost of the child node o', $C_{o'}^{Best}$. If $C_{o'}^{Current}$ is cheaper than the current best cost, $C_{o'}^{Best}$, both the best cost, $C_{o'}^{Best}$, and the best parent edge, $e_{o'}^{Parent,Best}$, is updated as

$$C_{o'}^{Best} = \begin{cases} C_{o'}^{Current} & \text{if } C_{o'}^{Current} < C_{o'}^{Best} \\ C_{o'}^{Best} & \text{otherwise} \end{cases} \tag{5.24}$$

$$e_{o'}^{Parent,Best} = \begin{cases} e' & \text{if } C_{o'}^{Current} < C_{o'}^{Best} \\ e_{o'}^{Parent,Best} & \text{otherwise} \end{cases} \tag{5.25}$$

5.3.3 Tree Search

Starting with the depot node, a breadth-first search can be used to find the shortest path through the network. The desired outcome of the tree search is that the leaf nodes, after all node expansions, will hold a cost, and the shortest path that leads to that cost. This is done by expanding each node that is visited in the breadth-first search. Because each node holds the information about the cheapest path to that node, it is easy to select the cheapest leaf node and trace the best solution backwards from that node. The procedure is described in Algorithm 3.

Algorithm 3 Tree search

for $l \in \mathcal{L} \setminus |\mathcal{L}|$ **do**
    for $o \in \mathcal{O}_l$ **do**
        $expand(o)$
    end for
end for
$o^{Best} \leftarrow$ first node o *in* $\mathcal{O}_{|\mathcal{L}|}$
for $o \in \mathcal{O}_{|\mathcal{L}|}$ **do**
    if $C_o^{Best} < C_{o^{Best}}^{Best}$ **then**
        $o^{Best} \leftarrow o$
    end if
end for

5.3.4 Solution

When the best node o^{Best} is known, the best objective value is the penalized cost of that node, $C_{o^{Best}}^{Best}$. To find the best heuristic schedule, it is easy follow the trace of $e_o^{Parent,Best}$ and o_e^{Parent} from the best leaf node o^{Best} to the start node.

Chapter 6

Hybrid Genetic Search with Adaptive Diversity Control for the OSVPPSO

In this chapter, a metaheuristic solution method for the *Operational Supply Vessel Planning Problem* is presented. The metaheuristic solution method used in this thesis is addressed as a *Hybrid Genetic Search Algorithm with Adaptive Diversity Control (HGSADC)*, a method first presented by Vidal et al. (2012) that quickly provides high quality solutions to specific variants of the well known *Vehicle Routing Problem*.

The intention of this metaheuristic is to ”combine the exploration breadth of population-based evolutionary search, the aggressive-improvement capabilities of neighborhood- based metaheuristics, and advanced population-diversity management schemes” (Vidal et al., 2012).

Borthen et al. (2018) addressed the Supply Vessel Planning Problem (SVPP) with a tailored version of Vidal et al. (2012)'s Hybrid Genetic Search with Adaptive Diversity Control. The metaheuristic for the SVPP was further extended to provide high quality solutions for the multi-objective SVPP in Borthen et al. (2019).

The HGSADC presented in this thesis for the Operational Supply Vessel Planning Problem with Speed Optimization partly draws on the one presented by Vidal et al. (2012) to solve VRPs and the one suggested by Borthen et al. (2018) to solve the single-objective SVPP. However, several problem-specific modifications have been added to efficiently address the problem at hand.

The HGSADC takes a set of available PSVs, a set of offshore installations with delivery deadline and weather forecast as input. Then, as output it provides a detailed voyage and schedule for each of the vessels departing from the onshore supply depot the next day. Note that the HGSADC is a non-deterministic metaheuristic, meaning it provides no guarantee for optimal solutions nor generating the same solution each run. In the following sections *HGSADC* refers to the metaheuristic developed in this thesis. It should also be emphasized that Borthen et al. (2018) do not take into account time windows-constrains, which increase the complexity of the problem significantly. Time window-constraints are considered in this thesis as a part of the operational aspect.

In Section 6.1, a general overview and a pseudocode of the Hybrid Genetic Search with Adaptive Diversity Control is presented. Further, in the subsequent sections, Section 6.2 - 6.7, a detailed explanation of each facet of the algorithm is provided.

6.1 Overview

The general structure of the HGSADC metaheuristic is shown in Algorithm 4, and is much similar to the one presented by Borthen et al. (2018). In order for the population-based algorithm to work, a construction heuristic creates multiple random initial individuals (i.e. solutions to the OSVPPSO including voyages and schedules), which further is placed in the initial population. At any time, the population $\mathcal{S}$ consists of two disjoint subpopulations $\mathcal{S}^{FEASIBLE}$ and $\mathcal{S}^{INFEASIBLE}$, where each of the subpopulation contains feasible and infeasible individuals, respectively. The infeasible population contains individuals violating one or more of the PSV capacity-, voyage duration- and delivery deadline constraints.

When the initial population is constructed, the algorithm keeps running until the best feasible individual has not improved for I^{NoImp} iterations, or the run time exceeds the time limit of T^{MAX}. In each iteration, i.e. one iteration of the *while-loop* in Algorithm 4, a new individual is formed and bred as an offspring of two carefully selected parents. It is further educated with probability p^{EDU}. Education includes systematic procedures looking for better neighbouring individuals using local search. If an individual is infeasible after the education procedure, regardless of whether it has been educated or not, it is *repaired* with a probability of p^{REP}. The repair procedure includes reeducating the infeasible individual with a higher penalty for violating the constraints. In case a large proportion of the individuals are infeasible, the penalty for violating specific constraints are adjusted dynamically in order to guide individuals into the feasible region.

As the number of individuals in a subpopulation reaches an upper bound, an elimination procedure referred to as *survivor selection*, is initiated to reduce the number of individuals in the respective subpopulation. After the elimination procedure, the μ best individuals in the subpopulation are retained for the next generation while the weaker individuals, i.e. poor solutions, are removed. A generation for a subpopulation is defined as the period between two elimination procedures for the same subpopulation. The minimum subpopulation size is given by μ and the generation size is given by λ. Thus, as new offsprings are created and bred in a subpopulation, reaching the upper bound subpopulation size of $\mu + \lambda$ individuals, the elimination process is initiated for that respective subpopulation. This means that the maximum number of individuals in the total population $\mathcal{S}$, is $2(\mu + \lambda)$, including both feasible and infeasible individuals.

A diversification procedure prevents the algorithm from getting caught in a local optimum, and is called every time the algorithm has bred I^{DIV} individuals without improving the best solution. An elimination process is initiated, reducing the population size and keeping the best individuals in the population. Further, new random individuals are created to further explore the feasible and infeasible regions of the solution space.

As the algorithm reaches I^{NoImp} iterations without improving the best feasible individual, or the run time exceeds the time limit of T^{MAX}, the best feasible individual is returned.

Algorithm 4 Hybrid Genetic Search with Adaptive Diversity Control

Create initial population using construction heuristic ▷ Section 6.3
while *Iterations without improvement* $< I^{NoImp}$ and $time < T^{MAX}$ **do**
  Select parent individuals s_1 and s_2 ▷ Section 6.5
  Generate offspring s_{new} from s_1 and s_2 (crossover) ▷ Section 6.5
  Educate offspring s_{new} with probability p^{EDU} ▷ Section 6.6
  if s_{new} is infeasible **then**
    Repair s_{new} with probability p^{REP} ▷ Section 6.6.4
  end if
  if s_{new} is still infeasible **then**
    Add s_{new} to the infeasible subpopulation
  else
    Add s_{new} to the feasible subpopulation
  end if
  if maximum subpopulation size $\mu + \lambda$ reached **then**
    Perform survivor selection ▷ Section 6.7.1
  end if
  Perform penalty parameters adjustment ▷ Section 6.7.2
  if the best individual has not been improved for I^{DIV} iterations **then**
    Diversify population ▷ Section 6.7.3
  end if
  Return *best feasible individual*
end while

6.2 Representation of individuals

This section explains what an individual is, which information it contains and how it is structured.

6.2.1 Representation of the Chromosome

In a genetic algorithm, an individual contains *chromosomes* holding information about the individual's characteristics, i.e. the voyages sailed by the vessels in the available fleet. Individuals can have multiple chromosomes, which are used in e.g. crossover and education operations.

In this genetic algorithm, each individual holds only one chromosome, in contrast to the algorithms proposed by Vidal et al. (2012) and Borthen et al. (2018) where each individual holds three different chromosomes. This is necessary as they address a periodic planning problem, where the departures of the various PSVs are spread throughout a planning horizon. Two of these three chromosomes address on which day a PSV should depart. Since the algorithm in this thesis only considers the next departure day, only one chromosome is needed. The chromosome held by an individual in our HGSADC is referred to as the *Vessel Tour Chromosome*.

The vessel tour chromosome keeps information about which of the available PSVs that are due to depart from the supply depot the next departure day and installations requiring deliveries. For each of the vessels departing, the chromosome will contain a corresponding list of installations. This works in the same manner as the *Giant Tour Chromosome* used by Vidal et al. (2012) and Borthen et al. (2018), by providing the list as the sequence in which the installations shall be visited. This sequence is referred to as the *voyage*, r_v for a PSV v. An example of the vessel tour chromosome is shown in Table 6.1 below. Note that the chromosome is zero-indexed with regards to PSVs. Also note that a PSV starts and returns to the supply depot, but the supply depot is not visualized in the chromosome.

Table 6.1: Vessel tour chromosome for individual s containing information about PSVs departing from the depot the next departure day, and which installations the corresponding PSV will visit

Vessel Tour Chromosome for individual s			
PSV, v	0	1	2
Voyage, r_v	$\{9, 6, 2, 7\}$	$\{5, 1, 8, 3, 4\}$	$\{\}$

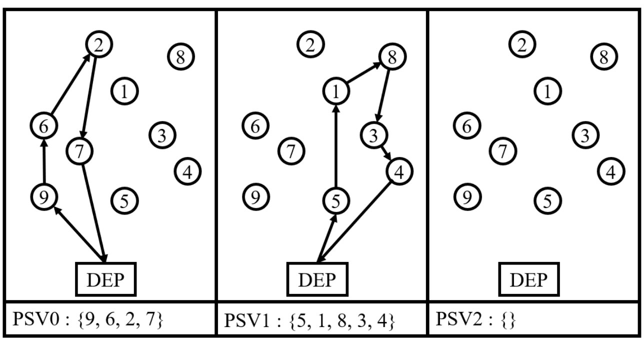


Figure 6.1: An illustration of the vessel tour chromosome in practice

From Table 6.1, it is shown three PSVs; $PSV0$, $PSV1$ and $PSV2$. Each of these vessels have a corresponding voyage r_v, which in practice is the sequence in which specific installations are to be visited. In this case, $PSV0$ loads and delivers cargo to the installation numbers 9, 6, 2 and 7, in that given sequence. $PSV1$ also has installations to visit in its corresponding voyage. However, $PSV2$ holds an empty list of installations to visit, and hence, no voyage will be sailed by this PSV. This means that $PSV0$ and $PSV1$ will depart the next departure day, but $PSV2$ will remain at the supply depot.

It is worth noticing that a subproblem is solved for each voyage in the HGSADC. This subproblem is the weather-dependent speed optimization problem presented in Chapter 5. When the speed optimization problem is solved as a shortest path problem for a given voyage, it is assigned with a cost and a schedule.

6.2.2 Infeasible Individuals

An individual is feasible as long as it does not violate any constraints. As earlier mentioned in Section 6.1, the population of individuals, $\mathcal{S}$, consists of one subpopulation with feasible individuals and one with infeasible individuals. The reason why infeasible individuals are considered in the genetic algorithm, is because optimal solutions often lie at the edge of feasibility. Thus, Vidal et al. (2014) argue that allowing infeasible individuals may enhance the performance of the search for better individuals.

The infeasible individuals are, however, only permitted to violate certain constraints, and they are penalized according to how much these constraints are violated. As with Vidal et al. (2012) and Borthen et al. (2018), the capacity and duration of a voyage are allowed to be violated. Also, not meeting the delivery deadline for an installation has been imposed as a soft constraints, meaning it can be violated and penalized according to how much it is violated.

6.3 Creating the Initial Population

Before the population-based part of the HGSADC can start evolving individuals and performing enhancements of generations, an initial population must be generated. This is done using a construction heuristic, a non-deterministic method to create random and nonsimilar individuals. The construction heuristic runs $K^{INIT}\mu$ iterations, generating and inserting a new individual into the initial population $\mathcal{S}$ in each iteration.

Let $\mathcal{N}^I$ be a subset of $\mathcal{N}$, which is presented in Section 4.2.2, containing all offshore installations to visit on the voyages departing the next day. In contrast to $\mathcal{N}$, $\mathcal{N}^I$ does not include the supply depot. Further, let $\mathcal{R}(s)$ be the set of voyages r_v assigned to a vessel v in individual s, i.e. the vessel tour chromosome of individual s. Note that the vessels $v \in \mathcal{R}(s)$ are the available PSVs in the set $\mathcal{V}$ from Section 4.2.2. Each time a new individual s is generated, all available PSVs are initiated with empty voyages. Each installation $i \in \mathcal{N}^I$ is assigned to a random available vessel v, and inserted at the last position in the voyage r_v sailed by this PSV. When all installations in $\mathcal{N}^I$ are assigned to a random vessel v, each voyage $r_v \in \mathcal{R}(s)$ shuffles its installation visit sequence, making the visit sequence in each voyage random.

As each installation $i \in \mathcal{N}^I$ has been assigned to a voyage, the individual s is educated, a procedure described in the Section 6.6. A feasibility check is subsequently performed to map whether or not individual s is feasible. If it is feasible, individual s will be inserted into the feasible subpopulation $\mathcal{S}^{FEASIBLE}$. If not, it is infeasible and undergoes a repair procedure with probability p^{REP}, which is explained in Section 6.6.4. In case the individual is still not feasible, it is inserted into the infeasible subpopulation $\mathcal{S}^{INFEASIBLE}$. When $K^{INIT}\mu$ iterations have been performed, the construction heuristic concludes and returns the initial population $\mathcal{S}$.

Algorithm 5 Construction heuristic

$individualsCreated \leftarrow 0$
while $individualsCreated < K^{INIT}\mu$ **do**
  STEP 1: CREATE INDIVIDUAL s:
  for $v \in \mathcal{R}(s)$ **do** ▷ Initiate empty voyages for all vessels
    $r_v \leftarrow \emptyset$
  end for
  for $i \in \mathcal{N}^I$ **do** ▷ Allocate installations to random vessels
    $v \leftarrow$ random vessel in $\mathcal{R}(s)$
    Add installation i to end of voyage r_v
  end for
  for $v \in \mathcal{R}(s)$ **do** ▷ Randomize installation visit sequence in each voyage
    Shuffle the installation visit sequence in voyage r_v
  end for
  Educate individual s

  STEP 2: INSERT INDIVIDUAL s TO SUBPOPULATION:
  if Individual s is infeasible **then**
    Repair s with probability p^{REP}
  end if
  if Individual s is still infeasible **then**
    Insert s into $\mathcal{S}^{INFEASIBLE}$
  else
    Insert s into $\mathcal{S}^{FEASIBLE}$
  end if
  $individualsCreated \leftarrow individualsCreated + 1$
end while

6.4 Evaluating Individuals

The overall fitness evaluation of an individual is a result of two contributions. The first is given by the monetary *cost contribution* from the individual, while the second is measured in terms of *diversity contribution*. An elaboration of these two terms follows.

6.4.1 Cost Evaluation

Let $\mathcal{R}(s)$ represent the chromosome consisting of a set of voyages in individual s, and let r_v denote the voyage performed by PSV v in $\in \mathcal{R}(s)$. A voyage is evaluated in terms of cost by its *Penalized cost*. The penalized cost of a voyage is found by solving the *Weather-Dependent Supply Vessel Speed Optimization Problem (WDSVSOP)* as a shortest path problem on a Directed Acyclic Graph. An explanation of WDSVSOP and the suggested solution method is presented in Chapter 5.

Penalized Cost

The penalized cost, ϕ_v, of a voyage r_v is defined as the monetary cost of operating that voyage and all penalties imposed if it is infeasible. Let c_v be the operational cost of operating voyage r_v in terms of fuel costs and potential spot vessel chartering costs.The penalized cost, ϕ_v, of a voyage r_v for vessel v is thus given by

$$\begin{aligned} \phi_v &= c_v \\ &+ \omega^Q max\{0, q_v - Q_v\} \\ &+ \omega^{\overline{T}} max\{0, t_v - \overline{T}_v\} \\ &+ \omega^D \textstyle\sum_{i \in r_v} max\{0, \tau_i - T_i^D\}, \end{aligned} \tag{6.1}$$

where ω^Q, $\omega^{\overline{T}}$ and ω^D represents the penalty parameters per unit violated for the capacity, duration and deadline constraints, respectively. Further, the variables q_v, t_v and τ_i signify the size of the cargo loaded onto a vessel v, overall duration for vessel v and the delivery finish time at installation i, respectively. Also, as in Section 4.2.2, Q_v is the upper load capacity of vessel v, $\overline{T}_v$ gives the maximum duration of a voyage v and T_i^D denotes the delivery deadline for installation $i \in r_v$. The last term in Equation (6.1) returns the accumulated overdue time points for all late deliveries in voyage r_v, if any exist.

Penalized Cost of an Individual

The penalized cost of an individual s amounts to the penalized cost of all voyages in individual s. Hence, the penalized cost of an individual is calculated as

$$\phi(s) = \sum_{v \in \mathcal{R}(s)} \phi_v \tag{6.2}$$

6.4.2 Diversity Evaluation

To avoid the HGSADC from getting caught in a local optimum, it is important to maintain a diverse population of individuals. The diversity contribution of individual s, $\Delta(s)$, is a measure used to ensure this, and is defined as the average distance to its n^{CLO} closest neighbours, which can be found in the set $\mathcal{N}^{CLO}$. As in Vidal et al. (2012) and Borthen et al. (2018), the diversity contribution of individual s_1 is calculated as

$$\Delta(s_1) = \frac{1}{n^{CLO}} \sum_{s_2 \in \mathcal{N}^{CLO}} \delta^H(s_1, s_2), \tag{6.3}$$

where $\delta^H(s_1, s_1)$ is the *normalized Hamming distance* from individual s_1 to s_2, which is based on the Hamming distance introduced by Hamming (1950). The normalized Hamming distance used in this algorithm slightly differs from the one used in Vidal et al. (2012) and Borthen et al. (2018). This method is used to capture how dissimilar the chromosomes of two individuals s_1 and s_2 are, and comprises two aspects; *vessel difference*, $\alpha(s_1, s_2)$, and *voyage difference*, $\beta(s_1, s_2)$. The first one being a count of how many installations that are visited with a different PSV in the chromosome of individual s_1 compared to the chromosome of s_2. The second being how many different sailing legs that are sailed in a PSV's voyage. The number of different sailing legs two individuals s_1 and s_2 can have equals sailing legs from the supply depot to an installation, sailing legs between installations and sailing legs from installations to the supply depot, which is given by $|\mathcal{N}^{\mathcal{I}}|+V^{DEP}$, where V^{DEP} denotes the number of vessels departing from the supply depot in individual s_1. The Hamming distance, i.e. the sum of vessel difference and voyage difference, is further normalized by dividing by two times the number of installations plus the number of vessels departing from the supply depot in the chromosome of individual s_1. Thus, the normalized Hamming distance obtains a value between 0 and 1. The value 0 signifies that the chromosome is completely similar and considered a clone, while the value 1 represents a completely dissimilar chromosome.

Let $v_i(s)$ denote the vessel number visiting installation i for individual s. The vessel difference between the chromosome of individual s_1 and s_2, $\alpha(s_1, s_2)$, is calculated as

$$\alpha(s_1, s_2) = \sum_{i \in \mathcal{R}(s_1)} \mathbf{1}(v_i(s_1) \neq v_i(s_2)), \tag{6.4}$$

where $\mathbf{1}(cond)$ equals 1 if condition *cond* is true, and 0 otherwise.

Now, let (i, j) denote the sailing leg from installation i to j. Then, the voyage difference between the chromosome of individual s_1 and s_2, $\beta(s_1, s_2)$, is given by

$$\beta(s_1, s_2) = \sum_{(i,j) \in \mathcal{R}(s_1)} \mathbf{1}(\neg \exists (i, j) \in \mathcal{R}(s_2)) \tag{6.5}$$

Note that the maximum value $\alpha(s_1, s_2)$ and $\beta(s_1, s_2)$ can take is $|\mathcal{N}^I|$ and $|\mathcal{N}^I| + V^{DEP}$, respectively. Further, the normalized Hamming distance is calculated as

$$\delta^H(s_1, s_2) = \frac{1}{2|\mathcal{N}^I| + V^{DEP}} (\alpha(s_1, s_2) + \beta(s_1, s_2)) \tag{6.6}$$

Example: Calculating the normalized Hamming distance

Example 6.8 below shows in detail how the normalized Hamming distance is calculated for the chromosomes of two individuals s_1 and s_2. The binary values within the brackets for the vessel difference- and voyage difference rows indicate if the conditions in Equation (6.4) and Equation (6.5) are true or false for each of the installations or sailing legs in individual s_1 compared to s_2. The value 1 signals *true* and a difference, while 0 indicates *false* and no difference. The sum of these binary values equals the vessel difference $\alpha(s_1, s_2)$ and voyage difference $\beta(s_1, s_2)$, as described in Equation (6.4) and Equation (6.5), respectively. Voyages in the vessel tour chromosome do not explicitly contain the supply depot in the beginning and the end, but note that sailing legs from and to the supply depot also counts. Thus, a voyage $r_v = \{9, 6, 2, 7\}$ consists of the following sailing legs: $(0, 9), (9, 6), (6, 2), (2, 7), (7, 0)$.

Let $\mathcal{N}^I$ contain all offshore installations to visit on voyages departing the next day. Then,

$$\mathcal{N}^I = \{1, 2, 3, 4, 5, 6, 7, 8, 9\} \tag{6.7}$$

Vessel Tour Chromosome for individuals s_1 and s_2			
PSV, $v \in \mathcal{V}$	0	1	2
Voyage, $r_v \in \mathcal{R}(s_1)$	{9, 6, 2, 7}	{5, 1, 8, 3, 4}	{}
Voyage, $r_v \in \mathcal{R}(s_2)$	{5, 6, 9, 4}	{1, 2, 7}	{8, 3}
Vessel Difference	{0, 0, 1, 1}	{1, 0, 1, 1, 1}	{}
Voyage Difference	{1, 1, 1, 0, 0}	{0, 1, 1, 0, 1, 0}	{}

In this example, installation 1, 6 and 9 are visited by the same PSV in both individuals, yielding a total vessel difference equal to the number of installations requiring visits less the number of installations visited by the same PSV, i.e. $9 - 3 = 6$. Also, five sailing legs are common for both individuals: $(2, 7), (7, 0), (0, 5), (8, 3)$. Hence, the total voyage difference equals the total number of sailing legs in individual s_1 less the number of equal sailing legs, i.e. $11 - 5 = 6$.

$$\begin{aligned} TotalVesselDifference &= 6 \\ TotalVoyageDifference &= 6 \\ |\mathcal{N}^I| &= 9 \\ V^{DEP} &= 2 \\ \delta^H(s_1, s_2) &= \tfrac{1}{2*9+2}(6+6) = 0.60 \end{aligned} \tag{6.8}$$

6.4.3 Biased Fitness

The *biased fitness* is an overall evaluation of an individual, where both penalized cost and diversity contribution is taken into account. Each individual is ranked in terms of penalized cost and diversity contribution. As in Vidal et al. (2012) and Borthen et al. (2018), let $Rank^C(s)$ be the *penalized cost rank*, and $Rank^D(s)$ be the *diversity contribution rank* of individual s. Further, let the individual with the lowest penalized cost, i.e. the best penalized cost, have $Rank^C(s) = 1$, and the one with the highest penalized cost have $Rank^C(s) = |\mathcal{S}^{FEASIBLE}|+|\mathcal{S}^{INFEASIBLE}|$. Also, let the individual with the best diversity contribution, i.e. closest to 1, have $Rank^D(s) = 1$, and the one with the worst diversity contribution have $Rank^D = |\mathcal{S}^{FEASIBLE}|+|\mathcal{S}^{INFEASIBLE}|$. Let n^{ELI} be the desired number of elite individuals to survive to the next generation. Then, the biased fitness for individual s is calculated as

$$BF(s) = Rank^C(s) + (1 - \frac{n^{ELI}}{|\mathcal{S}|})Rank^D(s) \tag{6.9}$$

6.5 Parent Selection and Crossover

Two existing individuals from the population $\mathcal{S}$ are selected and combined to create a new individual, known as an *offspring*. The process of creating an offspring as a combination of two parent individuals is known as *crossover* and is described in Algorithm 6. This procedure is in some ways similar to the one used by Borthen et al. (2018). However, for the OSVPPSO, constraints on the number of visits to each installation and spread of departures are not considered, yielding a somewhat less complex crossover procedure.

When not using the construction heuristic, the first step of generating a new individual s_{new} consists of selecting parents. Both parent individuals, s_1 and s_2, are selected one at a time with a technique referred to by Borthen et al. (2018) as a *binary tournament*. This technique involves picking two random individuals from the population $\mathcal{S}$ and selecting the one with the best biased fitness to be parent s_1, while discarding the other individual. The binary tournament is repeated to find the other parent s_2.

Recall that the vessels used in the chromosome are the vessels in the set of available PSVs, $\mathcal{V}$, presented in Section 4.2.2. Each vessel v in the vessel tour chromosome has a corresponding voyage r_v. The initial step, starting at Line 1 in Algorithm 6, is to determine which pieces of the vessel tour chromosome to inherit from which parent.

Each vessel is added to one of three disjoint sets; Λ_1, Λ_2 or Λ_{MIX}. These sets denote from which parent a vessel in the offspring should inherit characteristics. A vessel in Λ_1 and Λ_2 should inherit parts of the installation visit sequence from parent s_1 and s_2, respectively. A vessel in Λ_{MIX} will inherit characteristics from both parents. The number of vessels in each of these sets are randomly set from an uniform distribution. Let n_1 and n_2 be two random numbers between 0 and $|\mathcal{V}|$, where n_1 is the lowest of these. Then, Λ_1 contains n_1 randomly chosen vessels from the vessel tour chromosome, Λ_2 contains $n_2 - n_1$ randomly chosen vessels, and Λ_{MIX} contains the remaining $|\mathcal{V}| - n_2$ vessels.

The next step of the crossover procedure, starting at Line 6, involves data inheritance from parent individual s_1. For each of the vessel $v \in \Lambda_1$, the installation sequence of the voyage r_v sailed by vessel v is copied from parent s_1 to s_{new}. For a vessel $v \in \Lambda_{MIX}$, two randomly chosen splitting points, σ_v^1 and σ_v^2, denote indices in the installation sequence of the voyage r_v. Now, two cases are possible:

- *Case 1:* If σ_v^2 is larger than σ_v^1, a subsequence of the installation visit sequence from (and including) index σ_v^1 to (and including) index σ_v^2 of the voyage in the parent s_1 is copied to the offspring s_{new}.
- *Case 2:* If σ_v^1 is larger than σ_v^2, the entire installation visit sequence in voyage r_v, except for the subsequence from (but excluding) index σ_v^2 to (but excluding) index σ_v^1 of the voyage in the parent s_1 is copied to the offspring s_{new}.

The two possible cases are shown below in Figure 6.2.

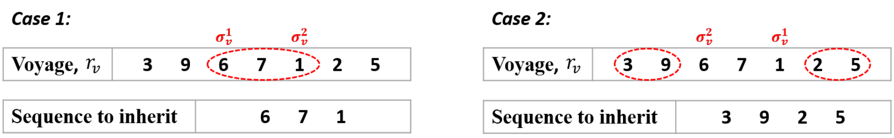


Figure 6.2: The two possible cases for σ_v^1 and σ_v^2

The next step, starting at Line 18, involves inheriting characteristics from parent individual s_2. For each vessel $v \in \Lambda_2$, the vessel v in s_{new} inherits the installation visit sequence from the voyage r_v in s_2. Vessels in the set Λ_{MIX} inherits installations from both parents, s_1 and s_2. Each vessel in Λ_{MIX} has already inherited a subsequence of installations from the voyage sailed by the respective PSV in parent s_1. When installations for the vessels in Λ_{MIX} are inherited from parent s_2, they are put after the subsequence inherited by s_1, in the same sequence as in the voyage sailed by the PSV in s_2. If they are already in the chromosome by inheritance from s_1, they will not be copied from s_2.

The last step of the algorithm, from Line 29, makes sure all installations from the parent individuals are passed on to the offspring individual. If there exist one or more installations that have not yet been assigned to the offspring s_{new}, it is added to s_{new}'s vessel tour chromosome in the voyage and at the position that gives the lowest marginal penalized cost of individual s_{new}, i.e. the lowest increase in total penalized cost of all voyages in the chromosome.

Algorithm 6 Crossover procedure

STEP 0: INHERITANCE RULE
Let n_1 and n_2 be the lowest and highest of two random integers between 0 and $|\mathcal{V}|$
Assign n_1 random vessels in $\mathcal{V}$ to the set Λ_1
Assign $n_2 - n_1$ of the remaining vessels in $\mathcal{V}$ randomly to the set Λ_2
Assign remaining vessels in $\mathcal{V}$ to the set Λ_{MIX}

STEP 1: INHERITANCE FROM PARENT s_1
for $v \in \Lambda_1$ **do**
    Copy the installation visit sequence of voyage r_v from s_1 to s_{new}
end for
for $v \in \Lambda_{MIX}$ **do**
    Create two random cutting points σ_v^1 and σ_v^2 in voyage r_v in s_1
    if $\sigma_v^2 > \sigma_v^1$ **then**
        Copy the subsequence of installations from σ_v^1 (including) to σ_v^2 (including) in voyage r_v from s_1 to s_{new}
    else
        Copy the voyage r_v in s_1 to s_{new}, but remove the subsequence of installations from σ_v^1 (excluding) to σ_v^2 (excluding) in voyage r_v from s_1 to s_{new}
    end if
end for

STEP 2: INHERITANCE FROM PARENT s_2
for $v \in \Lambda_1$ **do**
    Copy the installation visit sequence of voyage r_v from s_2 to s_{new}
end for
for $v \in \Lambda_{MIX}$ **do**
    for installation i in r_v in parent s_2 **do**
        if installation i is not already assigned to a voyage in the offspring s_{new} **then**
            add the installation i to the end of voyage r_v in s_{new}
        end if
    end for
end for

STEP 3: ASSIGN REMAINING INSTALLATIONS
while any installation i from any parent individual is not yet assigned to a voyage of a PSV in s_{new} **do**
    Find the voyage and the position to insert the installation yielding the lowest penalized cost of individual s_{new}
end while

6.6 Education of Individuals

Whenever an individual is created in the construction heuristic, or as an offspring of two parents, there may exist small alterations to the vessel tour chromosome enhancing the quality of the individuals significantly. The education procedure seeks to find these small changes and is performed in each of these situations. The education process consists of three procedures, which are performed in the following order:

1. Reducing the number of voyages
2. Intravoyage improvement
3. Intervoyage improvement

The first procedure attempts to reduce the total number of voyages sailed. The intravoyage improvement procedure looks for internal improvements within the installation visit sequence of one specific voyage. The intervoyage improvement on the other hand, looks for improvements across voyages sailed by different PSVs. These procedures will in the following sections will be further elaborated.

Through an individual's education, voyages are compared to each other several times. In order to be compared, they need to be evaluated in terms of a measure. The measure used is the penalized cost in Equation 6.1, which is obtained by solving the Weather-Dependent Supply Vessel Speed Optimization Problem presented in Section 5.3 for each voyage.

6.6.1 Voyage Reduction

This procedure aims to reduce the number of voyages sailed in an individual, and hence the number of departing PSVs. By reassigning the installations to visit on a voyage to the other voyages, the number of voyages for an individual can be reduced and decrease the overall costs significantly. As long as there is more than one voyage departing, this procedure is performed.

Some voyages are easier to eliminate than others. Often, short voyages where only a few installations are visited can have their installation visits reassigned to the remaining voyages. In case there are multiple shortest voyages, the one with the highest penalized cost is selected to be removed.

The installations visits that are to be reassigned to a remaining voyage are considered one by one. For each installation an evaluation is performed to identify into which voyage, and where within that voyage the installation should be inserted. The insertion that gives the best marginal penalized cost of the voyage, i.e. best change in penalized cost, is performed. When all installation visits are reassigned to the remaining voyages, a new evaluation of the individual is performed. If the penalized cost of the individual, i.e. all the voyages, is improved compared to the original individual, the voyage reduction procedure is undertaken. If not, the individual remains as before the procedure commenced. The procedure is shown in Algorithm 7.

Algorithm 7 Voyage reduction

if $departingVessels > 1$ **then**
  $penalizedCostBeforeReduction = getPenalizedCost(s)$ ▷ Section 6.4.1
  $s_{copy} \leftarrow s$ ▷ Make a copy of individual s

  STEP 1: FIND THE VOYAGE TO REMOVE FROM THE CHROMOSOME
  $r_{voyageToRemove} = findMostExpensiveShortestVoyage(\mathcal{R}(s_{copy}))$

  STEP 2: REASSIGN THE INSTALLATIONS IN THE TERMINATED VOYAGE TO THE REMAINING VOYAGES
  for installation $i \in r_{voyageToRemove}$ **do**
    $bestVessel = findCheapestVesselInsertion(i)$
    $bestPosition = findCheapestPositionToInsertInst(r_{bestVessel})$
    Insert installation i into voyage $r_{bestVessel}$ at position $bestPosition$
  end for
  Delete $r_{voyageToRemove}$ from $\mathcal{R}(s_{copy})$

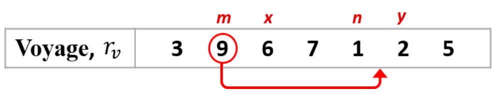
  STEP 3: EVALUATE AND PERFORM
  $penalizedCostAfterReduction = getPenalizedCost(s_{copy})$
  if $penalizedCostAfterReduction < penalizedCostBeforeReduction$ **then**
    Return individual s_{copy} ▷ Return new individual
  else
    Return individual s ▷ Return original individual
  end if
end if

6.6.2 Intravoyage Improvement

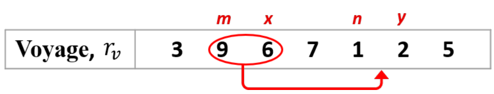
Voyage improvement is a procedure that is aimed to quickly look for simple, internal alterations within a voyage, resulting in a lower penalized cost of a voyage. These alterations includes reordering of the sequence in which installations are visited. After a change has been made to the voyage, the voyage will still contain the same installations as before, only with a different visit sequence. However, changes to a voyage will be undertaken if, and only if it enhances the penalized cost of the voyage.

Let the neighbourhood of an installation m in a voyage be defined as the set of all other installations to visit on that same voyage. The intention of the intravoyage improvement procedure is to evaluate each installation m and all of its neighbours in a random order. Let n be a neighbour of m. Then, let x and y be the successor of m and n, respectively. For each installation m, predefined moves can be undertaken and are evaluated in a random sequence. The first move resulting in a voyage with a better penalized cost, if any, is chosen and the next installation is evaluated in the same manner. These moves are similar to the ones used by Borthen et al. (2018). An explanation and visualisation of each move is shown below:

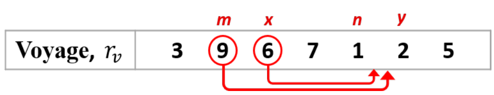
- *Move 1*: Remove m and place after n



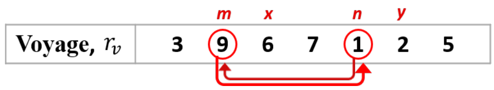
- *Move 2*: Remove m and x and place m and x after n



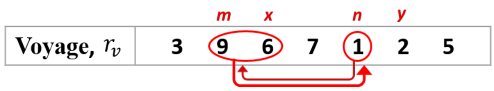
- *Move 3*: Remove m and x and place x and m after n



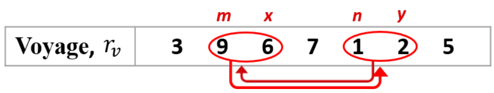
- *Move 4*: Swap the position of m and n



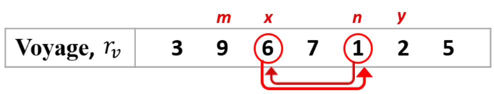
- *Move 5*: Swap the position of m and x with n



- *Move 6*: Swap the position of m and x with n and y



- *Move 7*: Swap the position of x and n



When all installations and their neighbours have been evaluated, the intravoyage improvement procedure is concluded. The intravoyage improvement procedure can be viewed in Algorithm 8.

Algorithm 8 Intravoyage Improvement

```
for voyage r_v ∈ R(s) do
    newVoyage ← r_v
    originalPenalizedCost ← getPenalizedCost(r_v)                ▷ Section 6.4.1
    if voyage r_v is not empty then
        N^UntreatedInsts ← r_v
        while |N^UntreatedInsts| > 1 do
            installation m = pickAndRemoveRandomInst(N^UntreatedInsts)
            N^Neighbours ← N^UntreatedInsts \ {m}
            while |N^Neighbours| > 0 do
                installation n = pickAndRemoveRandomInst(N^Neighbours)
                while allMovedExplored ≠ true do
                    newVoyage = doRandomMove(m, n)                ▷ Section 6.6.2
                    newPenalizedCost ← getPenalizedCost(newVoyage)
                    if newPenalizedCost < originalPenalizedCost then
                        Perform move on voyage r_v
                        Break
                    end if
                end while
                N^Neighbours ← N^Neighbours \ {n}
            end while
        end while
    end if
end for
```

$\mathcal{N}^{UntreatedInsts}$ - Set of installations that has not been checked for improvements yet

6.6.3 Intervoyage Improvement

Intervoyage improvement is a technique used to identify and perform simple alterations across voyages that yields a better penalized cost of the individual as a whole, i.e. the accumulated penalized cost of all voyages. The intervoyage improvement iterates through each and one of the installation visits in each of the voyages in the vessel tour chromosome. In each iteration the algorithm looks for the insertion within any voyage yielding the best penalized cost. Only one alteration is performed every time the procedure is run, and the move that gives the best penalized cost improvement, if any, is undertaken.

This is a procedure that theoretically can improve the solution significantly. As an installation visit can be moved from one voyage to another, an infeasible solution with regards to capacity, voyage duration and delivery date may become feasible and avoid any expensive penalties. It also makes sense to perform the intravoyage improvement on an individual that is already feasible, as intravoyage improvement cannot do anything with a voyage violating the capacity constraint.

Algorithm 9 Intervoyage Improvement

$bestPenalizedCost \leftarrow getPenalizedCost(s)$ ▷ Section 6.4.1
$s_{best} \leftarrow s$ ▷ Copy individual s
for voyage $r_v \in \mathcal{R}(s)$ **do**
    for installation visit $i \in r_v$ **do**
        for voyage $r_v^* \in \mathcal{R}(s) \setminus \{r_v\}$ **do**
            for position $k \in r_v^*$ **do**
                $s_{new} \leftarrow s$ ▷ Copy individual s
                insert installation i at position k in voyage r_v^* for individual s_{new}
                $newPenalizedCost \leftarrow getPenalizedCost(s_{new})$
                if $newPenalizedCost < bestPenalizedCost$ **then**
                    $bestPenalizedCost \leftarrow newPenalizedCost$
                    $s_{best} \leftarrow s_{new}$
                end if
            end for
        end for
    end for
end for
Return s_{best}

6.6.4 Repair

After the education procedure an individual will be either feasible or infeasible. If it is feasible, it is referred to as a *naturally feasible* individual. If it is infeasible, the individual may be exposed to a *repair* procedure.

The repair procedure is an attempt to make an infeasible individual feasible. It is performed with probability p^{REP} on infeasible individuals after education and before it is added to a subpopulation. The procedure involves increasing and thus multiplying the penalty parameters by a factor > 1 (e.g. 10) before educating the individual. If the individual is still infeasible, the original penalty parameters before the repair procedure are multiplied with a factor $>> 1$ (e.g. 100) and reeducated. The aim of the increased penalty parameters is to make it more costly for an individual to be infeasible, and thus provoke the individual into a feasible region of the solution space.

6.7 Population Management

This section describes the procedures that maintain sustainable populations of individuals throughout generations. The *survivor selection* procedure maintains the best individuals in every generation, ensuring intensification in the metaheuristic. Further, the *penalty parameter adjustment* is a dynamic mechanism ensuring a desired balance between feasible and infeasible individuals. Also, the *diversification* process ensures an extensive exploration of the search space. All these mechanisms affect the entire population, in contrast to the procedures above, which affect one individual at a time. These mechanisms take inspiration from the ones used in Borthen et al. (2018)'s population management.

6.7.1 Survivor Selection

The aim of the survivor selection procedure is to make sure the best fitted individuals survive and are passed on to the next generation, ensuring a population of high quality. This procedure is executed on a subpopulation as soon as the subpopulation size reaches $\mu + \lambda$ individuals. Then, all individuals in that subpopulation are sorted in terms of their biased fitness, removing the individuals with the worst biased fitness until there is a total μ individuals left in the respective subpopulation. As a result of this procedure, bad individuals and clones, i.e. individuals with a Hamming distance of zero, are removed.

6.7.2 Penalty Parameter Adjustment

The goal of the penalty parameter adjustment procedure is to generate a desired proportion of feasible individuals. The penalty parameters are adjusted dynamically as the number of individuals generated increase. The procedure considers the last κ individuals created.

The share of feasible individuals for each of the constraints decides whether or not the penalty parameter for that constraint should be adjusted up or down. Let ξ^Q, $\xi^{\overline{T}}$ and ξ^D denote the proportion of naturally feasible individuals among the last κ for the capacity, duration and deadline constraints, respectively. Then, let ξ^{REF} denote the target proportion for naturally feasible individuals. ζ^{UP} and ζ^{DOWN} are parameters providing a factor of how much a penalty parameter should be adjusted up and down, respectively. Algorithm 10 shows how the penalty parameters are changed dynamically every κ iteration.

Algorithm 10 Dynamic adjustment of penalty parameters

for $p = Q, D, Z$ **do**
  if $\xi^p \leq \xi^{REF} - 0.05$ **then**
    $\omega^p = \omega^p \zeta^{UP}$ ▷ Feasible share too low, increase penalty
  end if
  if $\xi^p \geq \xi^{REF} + 0.05$ **then**
    $\omega^p = \omega^p \zeta^{DOWN}$ ▷ Feasible share too high, decrease penalty
  end if
end for

6.7.3 Diversification

The aim of the *diversification* procedure is to prevent the algorithm from getting caught in a local optimum and rather expand the search into regions of the search space which has not been explored.

The diversification mechanism is executed if the best individual has not experienced any improvement for the last I^{DIV} iterations. Then, all individuals except for the best third of each subpopulation in terms of penalized cost are removed. Further, $K^{DIV}\mu$ new individuals are created and inserted into the correct subpopulation.

6.8 Summary

The Hybrid Genetic Search with Adaptive Diversity Control is an evolutionary population-based metaheuristic designed to find high-quality solutions to the OSVPPSO while at the same time exploring new regions of the search space. A special trait for this genetic algorithm is that every voyage in the algorithm is evaluated and assigned a cost through a speed optimization subproblem, where a shortest path problem is solved on a *Directed Acyclic Graph (DAG)* as described in Chapter 5.

Chapter 7

Problem Instances and Parameters

This chapter describes how weather affects the offshore operations and explains how a problem instance used to test the solution methods for the OSVPPSO is composed. Section 7.1 gives a detailed explanation of the problem case handed over by Equinor. Section 7.2 describes how weather states are defined, and also the impacts they impose on operations. Weather scenarios with different weather forecasts used in the computational study shown in Section 7.3. Further, a thorough description of installation data is provided in Section 7.4. An overview of the problem instances used in the computational study is listed in Section 7.5.

7.1 The Mongstad Case

Equinor has provided a set of real data which is used in the computational study. The data set contains information about all installations serviced from the Mongstad supply depot and the PSVs performing the supply runs. This case is referred to as the Mongstad Case.

Mongstad supply base delivers supplies to eight different oil & gas fields, with a total of 27 offshore installations in the Northern Sea. Four of these installations, namely TRO, TRB, TRC and STA, experience limited opening hours from 7 am to 7 pm. The rest of the

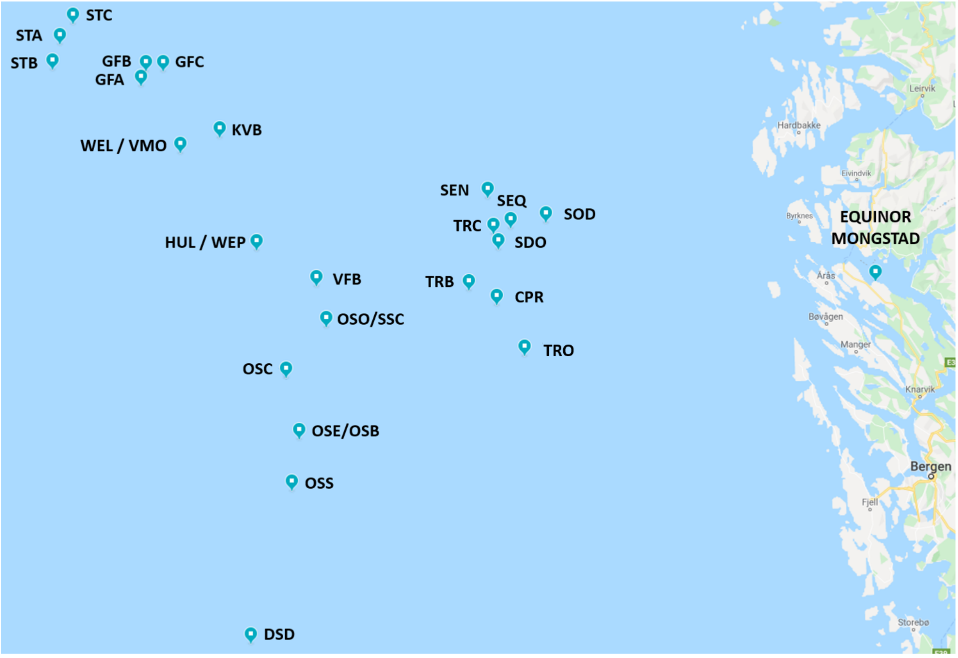


Figure 7.1: Map of the offshore installations in the Mongstad case

installations are open for service 24 hours a day. The time used to service an installation varies with the size of the delivery to the respective installation. It is assumed that ten units of cargo can be transferred from the PSV to the installation per hour.

From the Mongstad supply depot there are currently six PSVs hired on long-term contracts, with the possibility of hiring additional PSVs from the spot market at a price of USD 608 per hour (Seabrokers Group, 2018). The fuel price is set to USD 275.5 per metric tonnes.

7.2 Modelling Weather

Section 4.1.1 presents how the problem formulation deals with different weather conditions and how it affects the operations performed by PSVs. In this section, a definition of the four different weather states is provided along with fuel consumption functions.

7.2.1 Weather States

The four weather states presented in Section 4.1.1 describe which effects different weather conditions have on the operations performed by a PSV. Each of the states are defined by the significant wave height, and some states affect the PSV operations more than others. Higher significant wave heights may result in increased fuel consumed by a PSV, stricter maximum permitted vessel speed, increased service time, and in some cases no service of the installations at all. For which intervals of the significant wave height the different states are defined is shown in Table 7.1 along with the impact of each weather state.

Table 7.1: Weather state definitions inspired by Halvorsen-Weare and Fagerholt (2011)

Weather state	Significant wave height (m)	Offset & decrease in sailing speed (kts)	Increase in service time (%)	Increased fuel consumption for idling and servicing (%)
0	≤ 2.5	0	0	0
1	$(2.5, 3.5]$	0	20	20
2	$(3.5, 4.5]$	2	30	30
3	≥ 4.5	3	*Service Prohibited*	100

Recall that for the different weather states, fuel consumption for sailing is handled by calculating the consumption of the original sailing speed the PSV sails plus an additional offset depending on the weather state. In other words, if a PSV sails at speed v in rough weather, fuel consumption for speed $v + x$ is calculated, where x is the offset added by the weather state. These offsets are listed in the ”Offset & decrease in sailing speed (kts)”-column in Table 7.1.

7.2.2 Fuel Consumption

Each of the activities performed by a PSV, i.e. supply depot preparation, sailing, idling and servicing, experiences different fuel consumption. All activities, except for preparation, experience increased fuel consumption at higher significant wave heights. However, the fuel consumption for sailing is the only activity dependent on speed.

We make the assumption that the fuel consumption per time is a quadratic function of speed. The information available on fuel consumption per time for the PSVs is limited.

A few data points on fuel consumption for a specific sailing speed (including the fuel-minimizing speed) has been provided. These data points are plotted and used to generate a suitable quadratic function passing through all these data points. Thus, the quadratic function is assumed to provide a good estimate of the fuel consumption per time given the limited information available.

In this master's thesis, the quadratic function used to approximate fuel consumption per time (kg/hrs) as a function of the sailing speed v is given by

$$FC(v) = 11.111v^2 - 177.78v + 1011.1 \tag{7.1}$$

The resulting fuel consumption curves for sailing in each weather state are visualised in Figure 7.2 for the speed interval from 7 to 14 knots given in kg fuel consumed per hour.

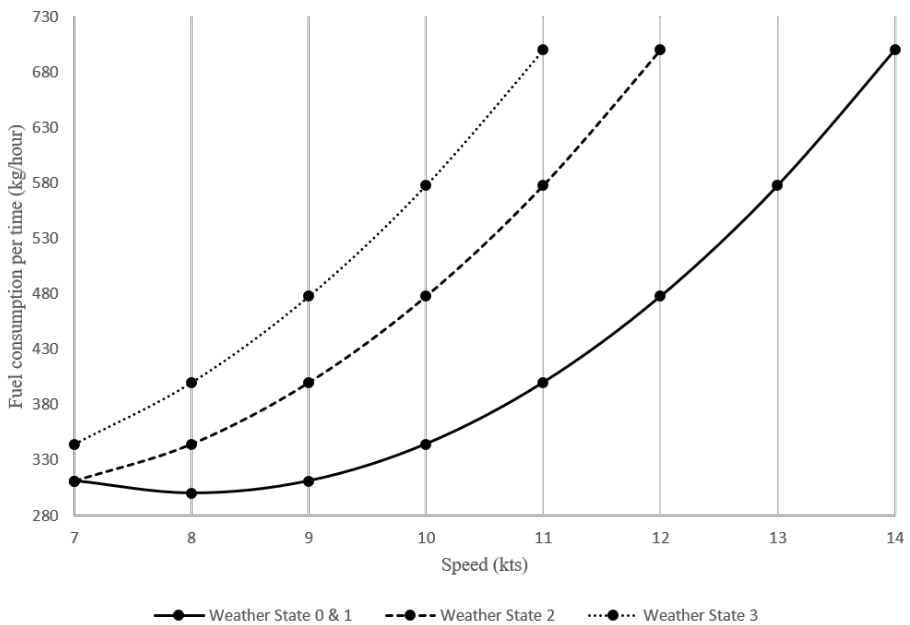


Figure 7.2: Weather impact on fuel consumption as a function of sailing speed in different weather states

The fuel consumption per time $(kg/hour)$ for the other activities is provided in Table 7.2.

Table 7.2: Fuel consumption rates for different activities performed by a PSV in *weather state 0*

Activity	**Fuel consumption per time** $(kg/hour)$
Supply depot preparation	45
Idling	120
Servicing	170

7.3 Weather Scenarios

Weather conditions have a significant impact on how voyages should be planned. Thus, it is important to test how the weather conditions affects the operational plans provided by the solution methods in this thesis. In order to test how weather conditions affects these solutions, three weather scenarios with different weather forecast are generated. A weather forecast shows the weather states for a scenario over time, where the weather states are presented in Section 4.1.1. This section provides a description of each weather scenario.

The weather scenarios are shown in Figure 7.3. The weather scenario notation summarize how the weather evolves over time. Thus, weather scenario *Bad* denotes that the weather conditions turns bad over time. In weather scenario *Good*, no installations are closed due to weather conditions, but some impact on operations still occur. Weather scenario *None* experience perfect weather, i.e. *weather state 0*, throughout the entire weather forecast and experience no weather impact on operations.

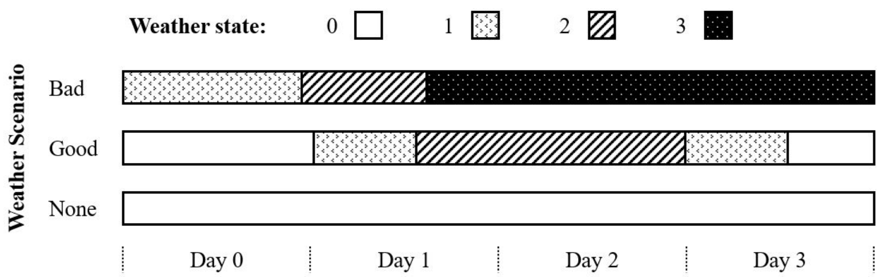


Figure 7.3: Weather forecast for weather states in the scenarios

The *Bad* weather scenario is generated with the sole objective of identifying how periods of bad weather is accounted for in the solutions. It is interesting to observe how vessels in the fleet increase the speed to perform the deliveries before the installations are closed for service.

The weather forecast time-discretization is set to one hour for all weather scenarios and problem instances used.

Some of these weather scenarios may be somewhat unrealistic. However, these weather scenarios are generated to analyze how the solution methods in this thesis accounts for weather in the operational voyage planning.

7.4 Installations Requiring Supply Delivery

The OSVPPSO is an operational problem where voyages are planned from day to day. Each day, the voyages departing the next departure day, e.g. tomorrow, are planned. The planners must know which installations that shall be visited on the voyages departing.

The number of installations that are to be visited varies. The number of installations with delivery requirements ranges from 3 to all 27 installations, where problem instances have been generated for all odd numbers in this range.

Each installation has an expected weakly demand and an expected number of visits per week. Therefore, there is also an expected size of each delivery to a specific installation. This expected size is referred to as a *baseline delivery size* for that respective installation. As this is an operational problem where the exact size of a delivery to an installation often is not known before the day of departure, variation of the delivery size is taken into account. Therefore, in each problem instance, most installations experience a demand equal to their baseline delivery size and some of the installations experience a somewhat smaller or larger demand than the baseline delivery size.

It is realistic that the urgency of supply requirements are considered. In each set, the delivery at an installation is needed within a deadline. The voyage starts on Day 0, while the deadlines vary from Day 1 to Day 4 for different installations.

7.5 Overview of Problem Instances

A general overview of the problem instances used in the computational study in Chapter 8 is presented in Table 7.3. Each instance provides information about the number of installations, the number of installations with limited opening hours and the number of available vessels in the fleet. All problem instances used in Chapter 8 have the option of chartering spot vessels. It is also denoted in the table which instances require more cargo delivered than the total accumulated cargo capacity of the available fleet. These instances are marked with ”Fleet size cargo capacity exceeded” which means that an additional vessel must be chartered from the spot market to not violate capacity constraints. This comment is based on tests conducted on the decent weather scenario, i.e. *weather scenario Good*.

Table 7.3: All 20 standard problem instances

Problem instance	**Number of installations**	**Vessels available**	**Comment**
0	$3^{(0)}$	1	
1	$5^{(1)}$	1	
2	$5^{(2)}$	1	
3	$7^{(1)}$	1	
4	$7^{(0)}$	1	Cargo capacity of available fleet exceeded
5	$9^{(0)}$	2	
6	$9^{(0)}$	2	
7	$11^{(1)}$	1	Cargo capacity of available fleet exceeded
8	$11^{(4)}$	2	
9	$13^{(1)}$	2	Cargo capacity of available fleet exceeded
10	$13^{(2)}$	2	Cargo capacity of available fleet exceeded
11	$15^{(3)}$	3	
12	$15^{(2)}$	3	
13	$17^{(2)}$	3	
14	$17^{(3)}$	3	
15	$19^{(4)}$	3	
16	$21^{(4)}$	3	
17	$23^{(4)}$	4	
18	$25^{(4)}$	4	Cargo capacity of available fleet exceeded
19	$27^{(4)}$	5	

$^{(x)}$ - x denotes the number of installations in the problem instance with limited opening hours

Each scenario has a specific number of available PSVs in addition to one extra spot vessel that can be hired if necessary. In general, each scenario contains enough available vessels to perform the deliveries as long as the weather remains decent. However, some scenarios, namely scenario number 4, 7, 9, 10 and 18, are generated with scarcity of available vessels, meaning that the spot vessel may be needed in a solution.

Chapter 8

Computational Study

In this chapter, the problem instances and weather scenarios discussed in Chapter 7 are used to test the arc-flow model formulated in Chapter 4 and the HGSADC presented in Chapter 6. Section 8.1 describes which parameters are tuned in the HGSADC as well as how the parameters are tuned. Section 8.2 provides an analysis of the computational results, describing how the solution methods scale and how the HGSADC adapts its solutions to different scenarios. Section 8.3 provides managerial insights and discusses the value of an operational solution.

Hardware and software used for implementation and testing of the model is described in Table 8.1.

Table 8.1: System environment for testing

Processor	2x Intel Xeon Gold 5115 CPU, 2.4 GHz, 10 Cores
Memory	96 GB RAM
Operating System	Linux - CentOS 7.8.2003
Programming language	Python 3.7.4
Solver	Gurobi 9.0.2

The maximum time limit is set to one hour for all problem instances.

8.1 Parameter Tuning for the HGSADC

This section describes how parameters in the HGSADC are tuned and provides the parameter values used for the computational study. The initial values used during and the final values after tuning are listed in Table 8.2.

8.1.1 Parameter Tuning Approach

In order for the Hybrid Genetic Search with Adaptive Diversity Control to behave efficiently, parameters must in the best way possible be tailored to fit the problem instances at hand. The tuning of parameters for the HGSADC is a complex process which acts like an optimization problem itself. Thus, the parameter tuning can be performed in many ways. However, a common way to tune parameters is by isolating one parameter at the time and test the parameter for different values, and further select the values giving the most desirable traits.

The tuning approach used in this thesis is based on the assumption that a parameter should not by default be tuned as an isolated parameter, as their impact on a solution might be dependent on the value of other parameters. Therefore, parameters that are believed to be inter-dependent are grouped and tuned together as a set of parameters. The parameters that are tuned together are shown in Table 8.3. Since the target proportion of feasible individuals has a large impact on the feasibility of the individuals generated, it is chosen to be tuned alone. Because the subproblem is the largest contributor to solution time, and the number of discrete time periods strongly correlates with the solution time of the subproblem, the number of discrete time periods per hour should be determined first. Further, the parameter groups are tuned in the order that is assumed to have the most effect on the HGSADC. This is the same order as shown in the Table 8.3. The target proportion of feasible individuals parameter is tuned after the Diversification parameter group.

Each parameter group is isolated and tuned. Each time a parameter group is tuned, untuned parameters are held fixed at their initial values, as shown in Table 8.2. For each parameter in the parameter group, a set of test values are used for the respective parameter. The initial values and the span of test values are inspired by the parameters used by Borthen et al. (2018) and Vidal et al. (2012). Tests are run five times for every combination of parameter values, i.e. *setting*, in the parameter group for four different problem instances.

Table 8.2: Initial and final parameter values

	Parameter	Initial Value	Final Value	Description
Population management	μ	25	25	Minimum subpopulation size
	λ	75	100	Generation size
	ξ^{REF}	0.5	0.4	Target proportion of feasible individuals
	K^{INIT}	4	4	Construction heuristic size multiplier
Diversification	K^{DIV}	4	4	Diversification size multiplier
	η^{ELI}	0.4	0.5	Proportion of elite individuals, $n^{ELI} = \eta^{ELI}\|S\|$
	η^{CLO}	0.2	0.3	Proportion of individuals considered for diversity evaluation, $n^{CLO} = \eta^{CLO}\mu$
	I^{DIV}	500	400	Iterations before diversification
Rates & Probabilities	p^{EDU}	0.5	0.75	Education rate, i.e. the probability of an individual undergoing education
	p^{REP}	0.5	0.25	Repair rate, i.e. the probability of an infeasible individual undergoing repair
Penalty	ω^{Q}	1000	500	Initial penalty per unit for violating the capacity limit
	$\omega^{\overline{T}}$	500	500	Initial penalty per unit for violating the maximum duration
	ω^{D}	200	250	Initial penalty per unit for violating the delivery deadline
	κ	100	100	Number of newest individuals considered for penalty adjustment
	ζ^{UP}	1.2	1.2	Factor to increase penalties with to reach target proportion of feasible individuals
	ζ^{DOWN}	0.85	0.85	Factor to decrease penalties with to reach target proportion of feasible individuals
Stopping criteria	I^{NoImp}	2000	5000	Iterations without improvement before termination
	T^{MAX}	3600	3 600	Maximum run time (seconds)
Discretization	θ	4	4	Number of discrete time points per hour

Table 8.3: Parameter tuning groups

Parameter group	Parameters		
Population Management	μ,	λ	
Diversification	η^{ELI},	η^{CLO},	I^{DIV}
Education, Repair & Termination	p^{EDU},	p^{REP},	I^{NoImp}
Penalty	ω^{Q},	$\omega^{\overline{T}}$,	ω^{D}

Problem instance 8, 9, 11 and 13 has been chosen for the parameter tuning. These problem instances are used because the number of installation visits and vessels departing daily from the supply depot in the instances are realistic, whereas some instances also contains a higher number than realistic, giving extra protection. Different installations are used in each problem instance. Problem instance 9 is also chosen because its solution includes the use of a spot vessel. The weather scenario remains the same (i.e. *Good* weather scenario) for all instances. Delivery deadlines are set between Day 2 - Day 4.

For each parameter setting, the average solution time and average objective value among all five runs of a problem instance are calculated. For each setting, it is given a percentage value measuring how much worse it is compared to the best solution time and objective value for the respective problem instance. This measure is referred to as the *relative solution time* and the *relative objective value* for the setting in the respective problem instance. Thus, there exist one relative solution time and one relative objective value for each setting in each of the four problem instances. The evaluate the different parameter value combinations, the relative solution time and relative objective value are averaged over all four problem instances. These average values can thus be sorted on the desired characteristics. These are the percentage values found in the parameter tuning tables in following subsections.

As a solution to the OSVPPSO is found quickly with the HGSADC, parameter groups are tuned with a lexicographic approach, where the average relative objective value is prioritized in order to find the best objective values. If multiple parameter value combinations have the same average relative objective value, they are further sorted on average relative solution time. Since solutions are found quickly, the best objective value is considered more important than solution time.

The most sensitive and important parameters impacting the objective values are tuned. Some parameters are however not tuned. This include the parameters which, based on the sensitivity analysis provided by Vidal et al. (2012), is assumed to have has less impact on the performance of the HGSADC. These parameters include K^{INIT}, K^{DIV}, κ, ζ^{UP},

ζ^{DOWN} and T^{MAX}.

Because of the operational nature of the problem, a maximum solution time of one hour is set. This is done so that the HGSADC may be used in day-to-day operations by operational planners, and limit the trouble of re-planning multiple times per day if unforeseen changes in weather conditions or orders occur. Thus, when tuning the parameters, the parameters yielding the best objective value within this time limit have been chosen.

For the parameter tuning as a whole, a total of 3220 tests have been conducted. The parameter groups are tuned in the order that is assumed to have the most effect on the HGSADC. Thus, the parameter values are carefully selected through deep analysis. In order to run multiple parameter tests simultaneously, the code was linearized so that each run only occupied one processing thread each. Thus, solution times reported in these results are higher than the ones reported in technical results. This does however not affect the results of the parameter tuning as the relative solution times are similar.

8.1.2 Tuning of Discrete Time Points Per Hour

The first decision to undertake is how finely the discrete time periods should be. A challenge is to find the right size of the time interval between the discrete points, in order to benefit from sailing speed alternatives in the arc-flow model. If distances between installations are small and the time interval between the discrete points in time is too large, sailing at maximum and minimum speed will result in arriving at the same discrete point of time. For the HGSADC, the time discretization, θ, indicates the precision in a voyage schedule for the start and end time of a sailing leg. With finer discretizations, the likelyhood than arc with a close-to optimal sailing speed is generated for each sailing leg is increased. Hence, there is a trade-off between time precision and complexity of the problem.

The HGSADC is tested with discretization values ranging from 1 to 10 time periods per hour, meaning discrete time points with the precision range from 1 hour to 6 minutes. A total of 50 tests are conducted in this tuning process.

A trade-off between increased solution time and increased objective value is also present. The graphs in Figure 8.1 show the change in solution time and the change in best objective value for different discretization values. From these plots, setting the discrete time periods per hour, $\theta = 4$, yields the most desirable benefits in the trade-off between solution time and improved objective value. Note that after four discrete points per hour, almost no benefit in terms of objective value is obtained by a finer discretization value. While this is the case for the change in objective value, the increase in solution time increases almost linearly with the number of time discretizations per hour. Graphs in the figure are based on the average values of the four test instances which are run five times each. These average values can be found in Appendix D.2.

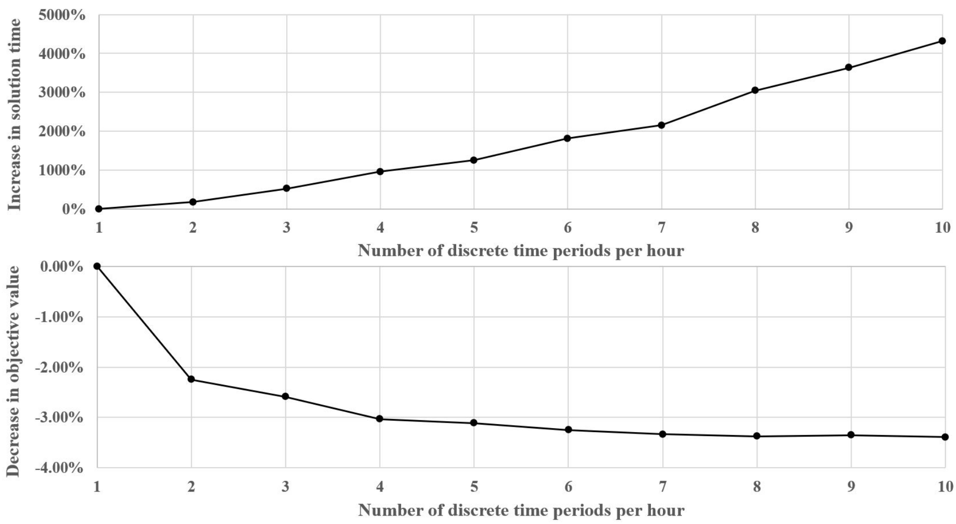


Figure 8.1: Change in solution time and objective value for the HGSADC for discretizations in the range 1 - 10 discrete time points per hour (i.e. discretizations from one hour to 6 minutes).

For most sailing legs, quarterly time intervals yield a sufficient precision, end such that close-to optimal speeds may be chosen for each sailing leg. In this case, four discrete time points per hour, $\theta = 4$, is chosen as the discretization value as it gives a good balance between solution time, precision and improved objective value.

8.1.3 Tuning of the Population Management Parameter Group

The parameter group *Population Management* involves tuning of the strongly dependent parameters that control the population in the HGSADC. This involves the minimum sub-population size, μ, and the generation size, λ. This is assumed to be the most important parameter group as it defines the boundaries for convergence and exploration of the search space in the problem. The values tested are shown in Table 8.4.

The minimum subpopulation size says something about how many individuals that are retained in the population during survival selection. This also says something about the span of local minimums in the search space the subpopulation may cover. Choosing a low value makes the algorithm converge faster as a higher concentration of the best solutions found are used to generate new individuals. High values allows for exploration of the search space.

The generation size says something about how much of the search space that is explored before survival selection is initiated. As seen in Table 8.5 the heuristic benefits from a large generation size. This makes sense as a smaller generation size might not allow for much exploration between generations, and thus force convergence towards a local minimum.

Table 8.4: Population Management group parameter values

Parameter	**Values**			
μ	15	25	35	50
λ	50	75	100	150

In this tuning process, a total of 320 tests are conducted. Further, the results of the five best parameter settings are shown in Table 8.5. Results show that the best parameter value combination is $\mu = 25$ and $\lambda = 100$.

Table 8.5: The five best results for Population Management tuning

Parameters		**Avg % from best**	
μ	λ	Time	Objective
25	**100**	**95%**	**0.0%**
50	50	131%	0.0%
35	150	144%	0.0%
50	150	178%	0.0%
25	150	125%	0.0%
...	...	...	...

Note that from the parameter combination ($\mu = 50$, $\lambda = 150$) to ($\mu = 25$, $\lambda = 150$), the solution time decrease. This is because the latter parameter value combination has a marginally higher objective value which is not captured by the two significant numbers in the table. Also note that in general, a 95% increase in solution time is a significant increase. Since the solutions are obtained quickly (i.e. in a few minutes only) by the HGSADC, time is not considered an issue, and thus the best objective value is preferred.

8.1.4 Tuning of the Diversification Parameter Group

The interaction of the parameters in the Diversification parameter group decides how much diversification is handled in the search of the HGSADC. It is important to have a diverse search, enabling exploration of the solution space, while at the same time focusing on finding local and global minimums. Table 8.6 shows the parameters that are tuned together, and shows the values tested.

Table 8.6: Diversification group parameter values

Parameter	Values			
η^{ELI}	0.3	0.4	0.5	0.7
η^{CLO}	0.1	0.2	0.3	
I^{DIV}	200	300	400	

The proportion of elite individuals, η^{ELI}, is used to calculate the biased fitness of an individual, where both penalized cost and diversity of an individual is considered (see Section 6.4.3). In the biased fitness calculation, η^{ELI} decides how much impact the diversity rank has on the biased fitness of an individual. A low value of η^{ELI} increases the importance of an individual's diversity in the biased fitness calculation. Four parameter values are tested for this parameter.

η^{CLO} denotes the proportion of individuals considered for diversity evaluation. High values for this parameter make the diversity contribution consider the average Hamming distance to a higher number of neighbouring individuals. However, the most interesting individuals to consider in the diversity evaluation are the closest ones. This parameter is tested for three values.

The diversification process is initiated if the best individual has not changed for I^{DIV} iterations. Finding an appropriate value for this parameter provides a fine balance for the algorithm between exploring local minimums as well as exploring the search space. Three different values are used in the test.

These parameters are assumed to affect each other and the overall diversification in the HGSADC. A total of 720 tests are conducted to tune this parameter group. The five best results for parameter value combinations are provided in Table 8.7.

Note that from the parameter combination ($\eta^{ELI} = 0.4$, $\eta^{CLO} = 0.3$, $I^{DIV} = 300$) to ($\eta^{ELI} = 0.4$, $\eta^{CLO} = 0.2$, $I^{DIV} = 400$), the solution time decrease. This is because the

Table 8.7: The five best results for the Diversification tuning

Parameters			**Avg % from best**	
η^{ELI}	η^{CLO}	I^{DIV}	Time	Objective
0.5	**0.3**	**400**	**48%**	**0.0%**
0.4	0.3	300	81%	0.0%
0.4	0.2	400	63%	0.0%
0.3	0.1	400	73%	0.0%
0.5	0.3	200	110%	0.0%
...	...	...	...	...

latter parameter value combination has a marginally higher objective value which is not captured by the two significant numbers in the table. The results show that two parameter settings are equally good for objective value, and thus the parameter setting resulting in the best solution time is chosen, which is $\eta^{ELI} = 0.5$, $\eta^{CLO} = 0.3$ and $I^{DIV} = 400$.

8.1.5 Tuning of the Target Proportion of Feasible Individuals Parameter

The target proportion of feasible individuals, ξ^{REF}, denotes the feasible proportion of the κ latest created individuals. Thus, for a high value of ξ^{REF}, having a high share of feasible individuals is more desired than having a high share of infeasible individuals. This parameter thus affects the HGSADC's ability to look for individuals at the edge of feasibility, where high-quality solutions often are located. A total of 100 runs are performed. The values tested along with the results are shown in Table 8.8.

Table 8.8: The five best results for the Target Proportion of Feasible Individuals tuning

Parameter	**Avg % from best**	
ξ^{REF}	Time	Objective
0.4	**72%**	**0.0%**
0.5	52%	0.1%
0.6	45%	0.2%
0.7	54%	0.2%
0.3	52%	0.6%

Results show that the most desirable value for $\xi^{REF} = 0.4$. This indicates that having a significant share of both feasible and infeasible solutions is beneficial. This emphasizes the statement that good solutions often lie at the edge of feasibility, and might thus be found in the process of repairing infeasible individuals.

8.1.6 Tuning of Education, Repair & Termination Parameter Group

The education rate, p^{EDU}, denotes the probability of an individual being educated when created with the construction heuristic or as an offspring of two parents. Education looks for better adjacent solutions and may improve the penalized cost of an individual significantly. Five different values are tested to span the range from no education to full education of an individual. The repair rate, p^{REP}, denotes the probability of an infeasible individual being repaired. Three values are tested for this parameter. I^{NoImp} indicates the termination criteria in terms of the number of iterations since the best feasible individual was improved. This parameter is tested for three values. The parameter values tested in this parameter group is shown in Table 8.9.

Table 8.9: Education, Repair & Termination group parameter values

Parameter	Values				
p^{EDU}	0	0.25	0.5	0.75	1
p^{REP}	0.25	0.5	0.75		
I^{NoImp}	2500	5000	7500		

Education and repair rates are assumed to be inter-dependent and largely contribute to the convergence of the algorithm. Because of this, different education and repair rates may require different values for I^{NoImp}. A total of 900 tests are conducted to tune this parameter group. The five best results are shown in Table 8.10.

Table 8.10: The five best results for the Education, Repair & Termination tuning

Parameters			Avg % from best	
p^{EDU}	p^{REP}	I^{NoImp}	Time	Objective
0.75	**0.25**	**5000**	**260%**	**0.0%**
0.0	0.75	5000	269%	0.0%
0.25	0.75	5000	315%	0.0%
0.25	0.75	7500	434%	0.0%
0.5	0.75	7500	494%	0.0%
...	...	...	...	...

For this case it is important to keep in mind that the best found solution found by the HGSADC is found for all parameter settings shown in Table 8.10. The same best solution was also found for the value $p^{EDU} = 1$, which is not listed in the table as it has higher solution time than the results listed. From the results provided, it requires a higher solution time to find the best solution when all individuals are educated. This may be caused by two factors: First, a realistic case for the OSVPPSO involves fewer installations than a realistic case addressed in the SVPP by Borthen et al. (2018), and thus less complexity in the voyage composition. Hence, exploration of the search space (due to a low education rate) is likely to find a specific high-quality solution faster than the systematic search using education. Second, in the crossover procedure, an offspring inherits parts of the chromosome from its parents. Remaining parts of the chromosome, if any, which are not inherited by the parents are inserted to the chromosome at the best possible (i.e. cheapest) positions. This ensures a certain inherent quality of an uneducated individual.

Also note the negative correlation between p^{EDU} and p^{REP}. When education is performed, infeasible individuals may become feasible. Thus, fewer individuals need to be repaired. However, if only a few individuals are educated, more individuals will be infeasible and thus need repair. The repair procedure may provide high-quality solutions as it looks for solutions on the edge of feasibility, which according to Vidal et al. (2012), often is where the best solutions are located.

It is an interesting observation that Borthen et al. (2018) use full education. In the paper by Vidal et al. (2012), the education rate spans from 0.7 to 0.86. The parameter values are set to $p^{EDU} = 0.75$, $p^{REP} = 0.25$ and $I^{NoImp} = 5000$. Thus, most individuals are educated and some infeasible individuals are repaired.

8.1.7 Tuning of the Penalty Parameter Group

Since the vessel capacity (ω^{Q}), maximum duration, ($\omega^{\overline{T}}$), and delivery deadline constraints, (ω^{T^D}), determine whether or not an individual is feasible or not, it makes sense to tune the penalty per unit for violating each of these together. However, since the penalties are changed dynamically throughout each run, the goal of tuning the penalty parameters is solely to obtain a good starting point for the algorithm. This way, the algorithm spends less time on obtaining the appropriate penalty parameter values.

Table 8.11: Penalty group parameter values

Parameter	Values		
ω^{Q}	100	500	1000
$\omega^{\overline{T}}$	100	250	500
ω^{D}	100	250	500

A total of 540 tests are conducted during this tuning process. The parameter values are listed in Table 8.11. Note that ω^{Q} is set higher than $\omega^{\overline{T}}$ and ω^{T^D} as the vessel capacity is considered to be a harder constraint than the time-constraints. The five best test results are shown in Table 8.12.

Table 8.12: The five best results for the Penalty tuning

Parameters			**Avg % from best**	
ω^Q	$\omega^{\overline{T}}$	ω^D	Time	Objective
500	**500**	**250**	**40%**	**0.0%**
500	100	100	43%	0.0%
500	100	250	44%	0.0%
500	100	500	44%	0.0%
500	250	500	46%	0.0%
...	...	...	...	...

The results for the different parameter settings clearly give the most indecisive results in this section. There is in general no pattern in these tuning results except that it is preferable to have a higher penalty for violating the capacity constraint. Thus, it may indicate that these parameters are not that sensitive to the performance of the HGSADC. This also makes sense because the parameter values change dynamically as the algorithm runs. The parameter values chosen are $\omega^Q = 500$, $\omega^{\overline{T}} = 500$ and $\omega^D = 250$.

8.1.8 Setting the Remaining Parameter Values

Some of the parameters are not tuned as they are not that sensitive to the objective value as the ones that are tuned. Vidal et al. (2012) and Borthen et al. (2018) argue that the parameters K^{INIT}, K^{DIV}, ζ^{UP}, ζ^{DOWN} are assumed to have little impact on the performance of the search. Therefore, K^{INIT}, K^{DIV}, ζ^{UP} and ζ^{DOWN} are given the same values as in these papers. The notation of the parameter κ does not exist in either of the papers, however, the same value is used for the same purpose. T^{MAX} is set to the same limit as for the arc-flow model, but it usually terminates within a few minutes.

8.1.9 Concluding Remarks On Parameter Tuning

Parameter tuning has been conducted on parameter groups rather than for each parameter isolated. This is assumed to enhance the interaction between the inter-dependent parameters. The parameters that are assumed to have the most impact on the search and the individuals have been tuned, while the parameters that yields little impact on the performance of the HGSADC is set to the same as Vidal et al. (2012) and Borthen et al. (2018). Generally, small differences from the parameter values used in these papers exist, which is expected as the problem is different. The parameters are tuned for realistic-sized problem instances for the *Good* weather scenario and should provide efficient performance of the HGSADC for close-to-similar instances. Most parameter groups, do not have test results showing a dominating parameter value combination, which may indicate that the HGSADC is stable for a large span of parameter values.

8.2 Computational Results

8.2.1 Comparing Solutions from the HGSADC with Solutions from a Commercial Solver

In the computational study, each problem instance described in Section 7.5 is tested with a commercial solver (Gurobi) using the arc-flow model, presented in Chapter 4, and the metaheuristic HGSADC, presented in Chapter 6. Both solution time and the objective value are presented for each problem instance in Table 8.13 and the objective values are illustrated graphically in Figure 8.2. Upper and lower bounds are provided for instances the commercial solver did not manage to solve to optimality. These tests are performed in the *Good* weather scenario.

Table 8.13: Comparison between HGSADC and commercial solver (Gurobi). The metaheuristic was run ten times for each problem instance to find average and minimum values.

	Commercial solver (Gurobi)			HGSADC			
#	**Time [s]**	**Upper bound**	**Optimality gap %**	**Avg. time [s]**	**Avg. obj. [$]**	**Min. obj. [$]**	**Diff.**
0	4.3	2388	0	10	2388	2388	0
1	5.1	2536	0	13	2536	2536	0
2	4.7	2064	0	11	2064	2064	0
3	577	3229	0	35	3229	3229	0
4	2792	11891	0	99	11891	11891	0
5	3600	4400	50.4	134	4223	4223	-177
6	3600	4528	42.5	129	4528	4528	-95
7	3600	15614	48.3	199	15173	15173	-441
8	3600	5136	39.3	167	5094	5094	-42
9	3600	15979	72.9	649	10910	10854	-5069
10	3600	15501	68.9	461	11415	11402	-4086
11	3600	8653	57.1	294	6582	6582	-2071
12	3600	6528	45.9	363	6468	6468	-60
13	3600	33153	89.2	498	7371	7370	-25782
14	3600	38034	90.2	485	6764	6764	-31270
15	3600	27346	85.6	688	7088	7086	-20258
16	3600	23871	80.5	1236	7611	7505	-16260
17	3600	NO SOL	NO SOL	1072	9345	9313	-
18	3600	NO SOL	NO SOL	2088	16488	16068	-
19	3600	NO SOL	NO SOL	1753	10715	10652	-
(0-16)	**2740**	**12997**	**45**	**321**	**6784**	**6774**	**-6212**

- Problem instance

Figure 8.2 shows that the commercial solver manages to solve each instance up to problem instance 4, which includes seven installation visits, to optimality within one hour. For larger instances, it finds worse upper bound solutions than the ones found by the HGSADC and it is not able to close the optimality gap. For the largest problem instances, no solutions are found by the commercial solver. The same test was also performed with a time limit of three hours, but no larger problem instance was solved.

Note that small problem instances, up to (and including) problem instance 2, are solved quicker to optimality by the commercial solver than with the HGSADC. Problem instance 2 includes five installation visits.

Problem instance 4, 7, 9, 10 and 18 experience a spike in the best obtained solutions. This

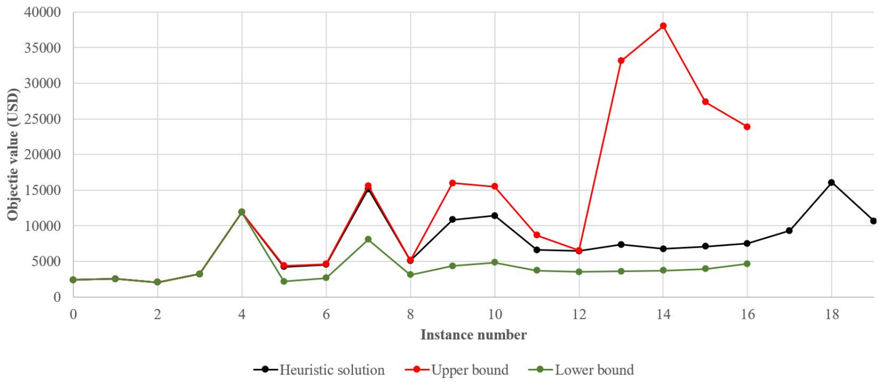


Figure 8.2: Comparison of bounds from the commercial solver and the objective found with the HGSADC

is due to use of a spot vessel in the solutions. These instances are generated such that a solution needs to include use of a spot vessel. From Figure 8.2, it is clear that the use of spot vessels has a major impact on cost.

The largest problem instance solved to optimality by the commercial solver, is problem instance 4, and the solver spent 1 906 seconds to find this solution. The same solution is found by the HGSADC already after 105 seconds, which shows that the HGSADC outperforms the commercial solver using the arc-flow model as the complexity increases.

8.2.2 Scaling with the Number of Installations for the HGSADC

Figure 8.3 shows how the solution time of the HGSADC increases with the number of installation visits in the instance. Solutions include use of a spot vessel for problem instances with 13 and 25 installation visits. Instances are run five times with the *Good* weather scenario and the average solution time is plotted.

The graph shows that the solution time grows with the number of installation visits. This is expected as the shortest path problem in the speed optimization problem is solved an increased number of times as the number of installation visits increase. The problem instances with 13 and 25 installations have solutions which include use of spot vessels. This may explain the observed spikes in solution time in the figure. An explanation of increased solution time when a spot vessel is used may be that the number of local minimums increases. This is because not all deliveries can be performed by the original fleet, and thus,

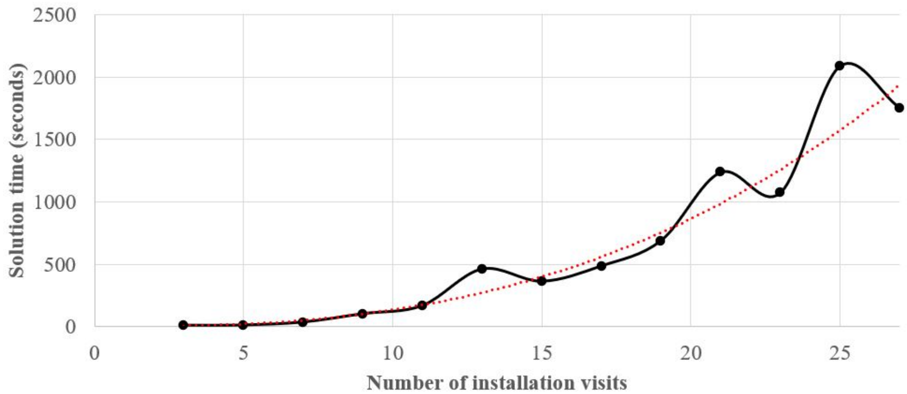


Figure 8.3: Solution time for the HGSADC compared with the number of installation visits

the decision regarding which deliveries should be performed with the spot vessel becomes important and has several good alternatives. With an increased number of local minimums, the solution time is expected to increase.

The red line in the figure shows the trendline for the increased solution time. To create the trendline in Figure 8.3, a power regression is performed. This regression shows that the estimated solution time for the HGSADC can be calculated by $t = 0.2813n^{2.681}$, where t is the solution time and n is the number of installation visits in the problem instance. All instances are solved within 30 minutes by the HGSADC, and the most realistic instances are solved within 10 minutes.

Observe that in Table 8.13 both the minimum objective value and the average objective value is provided for ten runs. For each problem instance up to problem instance 9, the minimum and average objective value are equal. As the number of installation visits increase, some variance between the minimum and average objective value occur, however, the variance remains small for each instance. Thus, it indicates stable performance of the HGSADC with regards to objective values as the number of installation visits increase, even for large problem instances.

A deeper analysis of the solution improvement for problem instance 17 is performed in Figure 8.4. The problem instance includes 23 installation visits and is solved for *Good* weather scenario. Note how easily the HGSADC finds improving solutions in the search already during the first seconds. After a while, the algorithm manages to get trapped in local minimums and stays there for a good amount of time. The diversification part of the

HGSADC ensures that the search space is explored, and just before the 5000th iteration, the algorithm manages to find the solution it terminates with.

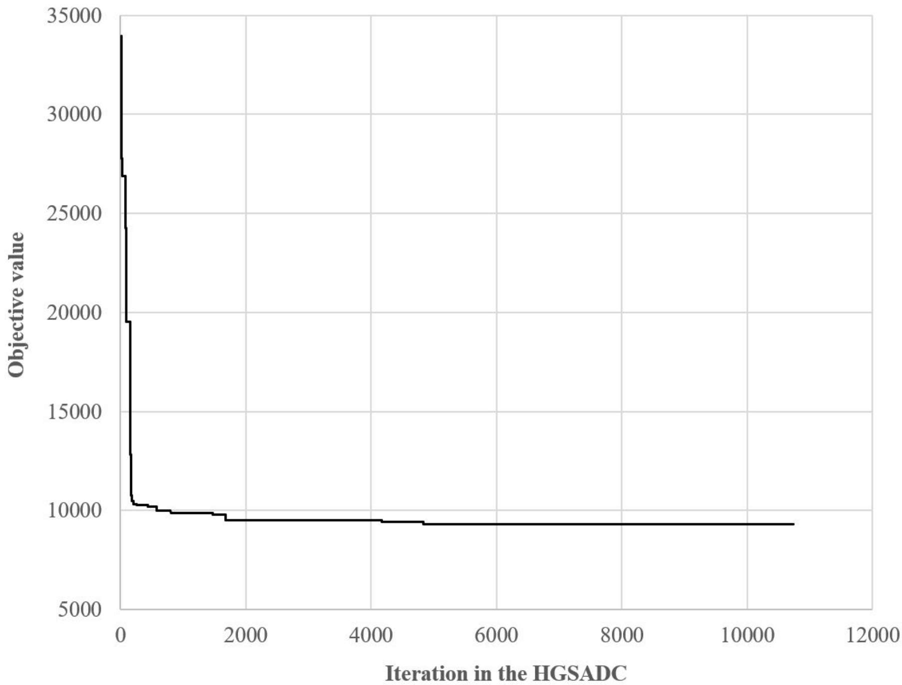


Figure 8.4: Progression of the objective value per iteration for one of the solutions for problem instance 17

8.2.3 Economical Impact of Operational Planning and Weather-Dependent Speed Optimization

To identify the economic impact operational planning and speed optimization may yield, seven problem instances have been solved with different optimization approaches. In this computational analysis, three disruption management approaches are presented and evaluated. Problem instances 11 to 17 have been selected for testing as they in perfect weather do not require any spot vessels in their solution. They are also selected as they are a combination of realistic- and large-sized instances in terms of the number of installation visits. The weather scenario used is the *Bad* weather scenario, where the weather conditions gradually becomes worse and reach *weather state 3* at the middle of Day 1. The weather conditions remain in *weather state 3* for several days.

The first approach is to disregard the tactical plan and re-optimize voyages and schedules operationally with weather-dependent speed optimization, which is the approach discussed in this thesis. To estimate the cost of this approach, the HGSADC has been used with the provided weather forecast as input. This approach is referred to in Table 8.14 as the *HGSADC*.

The next approach is to fully re-optimize the problem, but this time with a fixed design speed. Thus, the maximum and minimum speed limitations in the HGSADC was set to the economic and fuel-efficient sailing speed. Thus, the HGSADC algorithm could be solved to find the re-optimized solutions for this approach. This approach is referred to in Table 8.14 as *Design Speed*.

The last approach evaluated is assumed to be the worst case scenario. In this approach, the costs of not taking external factors into account when planning is considered. In this case, the approach is to follow the tactical schedule for all supply jobs that can be delivered in time. Then, for all the remaining deliveries, choose some disruption handling action to get the supplies delivered. In these estimations, it is assumed that the undelivered supplies must be placed on a spot vessel. To estimate these costs, the tactical schedule is evaluated in the *None* weather scenario using the HGSADC. Then, for all supply deliveries that cannot be performed in time, a penalty cost is added. The penalty cost added is an average estimate of the price per delivery performed by spot vessels and amounts to USD 3681. This is calculated by finding the average cost per delivery for problem instance 0-7 using only spot vessels. The estimations can be found in Appendix F. This approach is referred to as the *Limited Operational Planning* in Table 8.14.

Table 8.14: Comparing three different operational approaches for the *Bad* weather scenario

	HGSADC Solution		**Design Speed Solution**		**Limited Operational Planning**	
#	**Obj.val. [$]**	**s.d.**	**Obj.val. [$]**	**s.d.**	**Obj.val. [$]**	**s.d.**
11	7757	0	8168	0	25838	5
12	7585	0	7956	0	29315	6
13	8440	0	20604	2	19098	3
14	8278	0	19931	2	22087	6
15	8802	0	21695	3	22487	4
16	15991	1	24290	4	30255	6
17	10935	0	22705	3	24971	4
Avg.	9684	0.1	17907	2.0	24865	4.9

s.d. - number of deliveries that have to be performed with spot vessels
$ - monetary value in USD

From Table 8.14, clear indications show that for this weather scenario, there is significant value of weather-dependent speed optimization. The average objective value from the *HGSADC solutions* amounts in this scenario to USD 9684, whereas the *design speed solutions* almost double this value to an average of USD 17907. The cost savings from speed optimization amounts in this scenario to USD 8223 on average. Comparing with the *limited operational planning* solution, this difference rises even higher to USD 15180, clearly signaling the value of operational planning. This example indicates the value of weather-dependent speed optimization and shows that the HGSADC metaheuristic performs well in disruptive situations that may usually generate significant backlog in the upstream supply chain.

8.3 Managerial Insights

The following sections provide a further qualitative analysis of the supply vessel planning approach used in this thesis, the value it creates and how it may help the planners at Equinor in the operational planning. Section 8.3.1 discuss the environmental and cost-benefits of the solutions from the HGSADC, while Section 8.3.2 discuss how planners can use the HGSADC in the planning process. Lastly, in Section 8.3.3, master plans and the operational planning approach are discussed.

8.3.1 Economical and Environmental Value of Operational Planning

This thesis introduces a new insights to the field of supply vessel planning, where operational planning is the main focus. Weekly voyages and schedules are set by a tactical master plan. However, each day operational planning must be performed to adjust for weather conditions, miscalculations in cargo size of the deliveries and urgent unforeseen deliveries to installations. This type of planning is currently done manually by experienced planners in Equinor. However, addressing this problem by hand may be cumbersome and lead to inefficient solutions. It is stated by Equinor that weather conditions, miscalculation in cargo size of supply deliveries and sometimes urgent unforeseen orders made by installations are among the most challenging obstacles in the operational planning as they result in disruptions in the supply chain. This indicates that an operational planning tool can be of high value for Equinor.

Figure 8.5 shows two equal voyages with different schedules and speed profiles. The upper schedule represents a fixed weekly schedule for a voyage where the sailing speed of the PSV is fixed to $v^{avg} = 9.5kts$. The lower schedule shows how the metaheuristic presented in this thesis has generated a weather-adapted voyage with weather-adapted sailing speed for each sailing leg. The red area shows a time period where the significant wave height is above the safety limit that permits service of installations. Thus, installations cannot be serviced in this time interval.

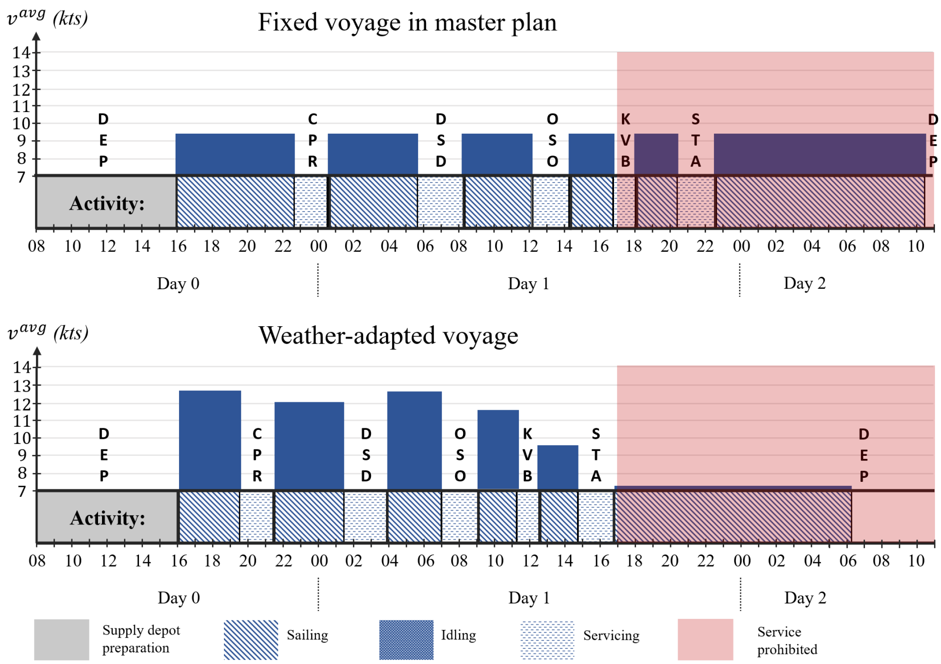


Figure 8.5: A fixed voyage in the master plan and a weather-adapted voyage. The red area indicates weather conditions where installations are not permitted to be serviced

Since the significant wave height exceeds the safety limit after Day 1 at 17:00, no installations can be serviced until it drops below the safety limit. If speed is not accounted for in the operational planning, the vessel will arrive at installation *KVB* at a point of time where it cannot service the installation. Installation *STA* will also not be serviced during this period. Thus, the vessel has two options, where both yields expensive and undesirable consequences. The first option involves the vessel to idle at the installation until the significant wave height drops below the safety limit. Then, a lot of fuel is wasted on battling currents, wind and tough sea conditions. Disruptions will also occur as the vessel will not be back at the supply depot at the planned return time. The second option is to return to the depot and add the deliveries for installation *KVB* and *STA* as backlog, resulting in possibly violated delivery deadlines. This would also yield disruptions in the master plan as these deliveries would have to be added to other fixed voyages or be delivered with a spot vessel when the weather conditions improve. The indirect cost of a backlog is difficult to estimate. However, it may be a reasonable estimate to use the cost of performing a backlog delivery using a spot vessel. In Section 8.2.3 this assumption is made. This example also indicates and emphasises the value of weather-dependent speed optimization which eliminates backlog. Note also that the chartering cost of a spot vessel is extremely high. With the chartering costs used in this thesis, which is provided by a company in this line of business, it is beneficial to run the spot vessel at the maximum speed to perform the delivery as quickly as possible in order to minimize the total costs of chartering and fuel consumption. This subject has also been addressed by Psaraftis and Kontovas (2014). When chartering supply vessels from the spot market, there is no incentive to operate environmentally friendly to reduce emissions. Thus, using the weather-dependent speed optimization greatly reduces the use of spot vessels yielding significant emission reductions in certain disruptive scenarios.

In the weather-adapted voyage, weather conditions are taken into account. Thus, the PSV increases its speed on the first sailing legs, such that it can finish all delivery jobs before the poor weather conditions strike. Not only is the solution cost- and emission-efficient in terms of fuel consumption, but it is also ensures that the potential costs of backlog are eliminated and indirect costs do not occur as a cause of shortage of supplies at the installations.

Speed optimization may have large impact on the fuel consumed on a voyage. As emissions are a result of fuel consumption, speed optimization makes the PSV operations more environmentally friendly in addition to making the usage of the PSVs cheaper in terms of monetary costs. In the fixed voyage in Figure 8.5, the sailing speed of the PSV is held constant at $v^{avg} = 9.5kts$. This is because it is considered the most fuel-efficient and economical sailing speed. However, this is only the case for the two first weather states and making a general assumption of perfect weather in the future would be naive. Figure 8.6 shows the fuel consumed per distance, (kg/nm), for every weather state. It presents the most fuel-efficient sailing speed (i.e. $v^{avg} = 9.5kts$) for the first two weather states with the fuel function used in this thesis. The fuel functions show that when the significant wave height increases, the economical and emission-reducing sailing speed is shifted to the left. This is taken into account by the *Weather-Dependent Supply Vessel Speed Optimization Problem (WDSVSOP)* which is solved as a subproblem in the HGSADC. To our knowledge, there exist no speed optimization algorithms that take this into account, although it can yield massive savings in the fuel consumed by a PSV on a voyage.

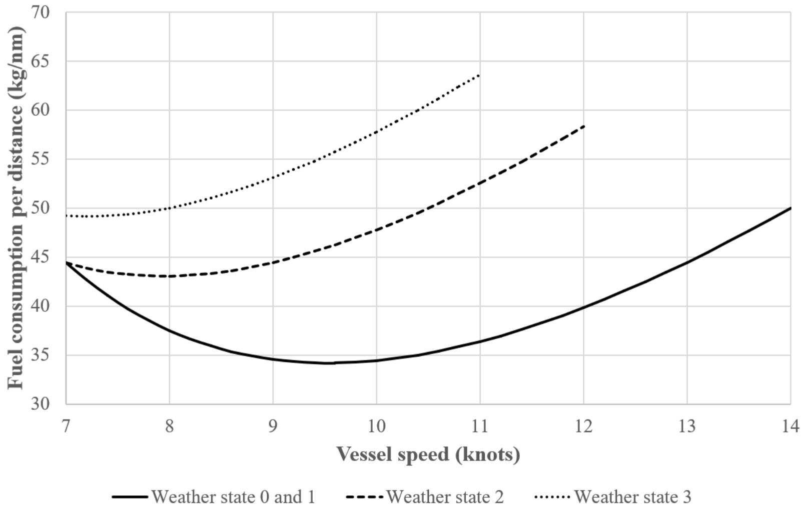


Figure 8.6: Fuel consumption per distance measured in kg/nm for a PSV in different weather states

From these weather-dependent fuel consumption curves, it can be observed that the fuel consumed per distance for the maximum speed for decent weather conditions almost equals the most fuel-efficient sailing speed for the worst weather conditions. This statement justifies speeding a PSV up almost to the maximum vessel speed in order to avoid the unnecessary time spent on a voyage in poor weather conditions. This can be viewed in Figure 8.5, where the vessel in the weather-adapted voyage sails at $v^{avg} = 13.8kts$ on the first sailing leg. Note that on this voyage, when the PSV returns to the depot, it has no hurry and sails at $v^{avg} = 7.2kts$, which according to the fuel consumption functions in Figure 8.6 is the most fuel-efficient speed for *weather state 2*.

The value of operational planning with weather-dependent speed optimization manifests itself in the computational study in Section 8.2.3, which indicates significant savings from taking weather forecast into account during planning. Computational results show that both emissions and costs related to disruptions and backlog are saved. This is one of the main challenges in Equinor's operational planning, and has the potential to become a significantly less challenging during planning in the future. For the scenario presented in Table 8.14, the HGSADC solutions required on average 0.1 deliveries to be performed with spot vessels, whereas the two other approaches evaluated required an average of 2.0 and 4.9 deliveries to be performed with spot vessels. Not only does this indicate the potential upstream supply costs savings, but also the risk taken by not optimizing operationally, as late supplies might cause halts in production at the offshore installations.

8.3.2 Assisting Planners in the Operational Planning Process

It is attractive to add flexibility to the planning process, and adding flexibility may mitigate the risk of disruptions. This operational planning tool have multiple traits that may assist an operational planner in Equinor to increase the flexibility in the planning.

One way to add flexibility is through the *return time*-restriction imposed on each specific PSV departing on a voyage. Some PSVs have specific equipment that is needed for specific missions. This way it is easy to ensure that the PSV is back at the supply depot when it is needed for special operations. Lets say that *PSV0* and *PSV1* are due to return the first and third day, respectively. Setting this requirement allows for flexibility in the way that other PSV1 may handle a higher number of deliveries than *PSV0*, which needs to be back at the supply depot earlier than *PSV1*.

Being able to handle a sudden and urgent incoming order from an installation also enhances the flexibility in the planning. This way urgent orders can be delivered to the installations on cost-efficient voyages, rather than just added to the beginning, or end, of a fixed voyage, yielding inefficient voyages. As shown in the computational study in Section 8.2.2, the solution time of the HGSADC on assumed to be realistic problem instances with nine installation visits are solved within minutes. Allowing voyages to be re-planned and adapted to urgent unforeseen incoming orders from installations short time before the vessel preparation commence, provides valuable flexibility for a company like Equinor.

It is also reported by Equinor that one of the most challenging obstacles to their operational planning is wrong estimates on the cargo size of deliveries. Just before the vessel preparation commence at the supply depot, equipment and supplies are assembled and packed in containers. When planning in advance, the planners have an estimate of the cargo size of the delivery to each installation. However, this estimation often turns out to be wrong after assembly and packing. If the estimated cargo size of a delivery is less than the actual cargo size, a vessel capacity problem may arise. A consequence may be that the PSV capacity is not sufficient to bring the cargo of all planned deliveries. Thus, voyages may have to be re-planned quickly after the real cargo sizes are known. As realistic-sized problem instance are solved within few minutes, the HGSADC may provide significant savings in emissions and costs in contrast to last-minute re-planning of voyages performed by hand.

As of now, the HGSADC is implemented as a planning tool such that the planners decide which installations that are to be visited on the voyages departing the next day. Often, when dealing with problems of this complexity, it is may be difficult to decide whether or not an installation should be visited on the voyages departing tomorrow or if they should be visited later. With the HGSADC, the planners may wonder how it would affect a voyage and a schedule to add or remove a delivery to an installation. As a problem instance is solved within few minutes by the HGSADC, this opportunity is provided, even for large problem instances. Thus, the HGSADC can be used by the planners as a decision-support tool deciding which installations that should be visited on the voyages departing the next day. It is also possible to use it as a tool to foresee how many installations that can be visited for a given weather forecast without having to charter an additional expensive spot vessel.

8.3.3 Operational Planning Versus Master Schedule

As of now, Equinor use fixed weekly voyages and schedules to service their offshore installations. Equinor argue that this is desired in order to have service at fixed weekly hours. Experience show that disruptions cause uncertainty to these predictable service hours and thus, installations end up being serviced in periods outside the weekly planned service hours. After these disruptions have occurred, large amounts of resources are used in order to return to the master schedule. Thus, it may be worth considering if it would be better to develop a more comprehensive operational planning tool. As part of this planning tool, the HGSADC could be included.

8.4 Summary

In this chapter, the solution methods suggested for the OSVPPSO have been evaluated. The commercial solver manages to solve small instances with up to seven installations to optimality within an hour. For the smallest problem instances, the exact solution method manages to find a solution marginally faster than the HGSADC. However, as the problem instances increase, the HGSADC experience stable performance and significantly outperforms the commercial solver using the arc-flow model. The HGSADC provides cost- and emission-efficient solutions for large problem instances within the time frame of one hour. Indications of cost and emission-savings resulting from operational planning have been discussed and estimated with calculations throughout the chapter. It is evident that many of the most challenging obstacles in the operational planning faced by Equinor may be greatly reduced with the right tools.

Chapter 9

Concluding Remarks and Future Research

This master's thesis has addressed the issue of planning voyages and schedules for platform supply vessels (PSVs) servicing a set of offshore oil and gas installations from an onshore supply depot. The problem addressed is denoted the *Operational Supply Vessel Planning Problem with Speed Optimization (OSVPPSO)*. The main focus of the thesis has been the operational aspect of the planning problem, where voyages and schedules also are tailored to account for the forecasted weather conditions in order to minimize costs.

This master's thesis is a collaboration project between Equinor, SINTEF, and NTNU. As of now, Equinor performs their daily operational supply vessel planning by hand. This is a difficult and time-consuming process that may lead to inefficient voyages and schedules. The planners at Equinor have described their main challenges faced in operational planning. The first includes disruptions to the master plans and costs associated with backlog caused by weather conditions. The second concern is the wrong estimations of cargo size for deliveries, yielding capacity problems for the supply vessels where not all deliveries can be brought on a planned voyage. The third challenge faced in operational planning is sudden, urgent supply orders from installations. Thus, Equinor has expressed the need for an operational decision-support tool. The main goal of the thesis has been to develop a decision-support tool that may assist the planners to overcome these challenges and create cost- and environmentally friendly voyages and schedules.

For decision-support, two solution methods for the OSVPPSO have been developed. One of these is an arc-flow formulation to be solved by a commercial MIP-solver. Using a commercial solver on the arc-flow formulation, only small problem instances up to seven installations may be solved to optimality within the time frame of 3600 seconds. The second solution method developed is a population-based metaheuristic, a *Hybrid Genetic Search with Adaptive Diversity Control (HGSADC)*. This metaheuristic solves the *Weather-Dependent Supply Vessel Speed Optimization Problem (WDSVSOP)* as a subproblem and finds high-quality solutions quickly. The WDSVSOP is the problem of creating a weather-adapted and speed optimized schedule for a given voyage, and is solved as a shortest path problem on a Directed Acyclic Graph (DAG) to evaluate and compare voyages in the HGSADC. The optimal solutions found with the exact solution method is also, in every case, found with the HGSADC. For problem instances with five or less installations, the commercial solver finds a solution faster than the HGSADC. However, for problem instances with more than five installations, the commercial solver is significantly outperformed by the HGSADC. The HGSADC quickly provides high-quality solutions to the largest problem instances with 27 installations within the time frame of one hour.

Through computational tests and discussions, the value of operational planning has been shown from the solutions provided by the HGSADC. It has been shown that significant savings can be obtained by tailoring voyages and schedules to the weather forecast. When the overall fuel costs and fuel consumption are reduced from thorough planning, emissions are also cut. This is important for Equinor in order to remain one of the most carbon-efficient producers within oil and gas. It has also been described in this thesis how weather-dependent operational planning mitigates the probability of disruptions and how the use of HGSADC can yield quick re-planning of voyages and schedules if last-minute changes occur. Thus, the HGSADC may contribute as a valuable decision-support tool.

A contribution to the field of research has also been added with the WDSVSOP, which includes speed optimization where fuel consumption functions and certain restrictions are weather-dependent.

In future research, the HGSADC may be adapted to operational planning of weather-adjusted voyages and schedules for a longer planning horizon. This can be implemented by expanding the chromosome to include departure days. This way, operational planners might be able to harness even greater benefits from operational planning with disruption management. The main challenge to overcome if the metaheuristic is extended to plan for more than one departure day at a time, is to keep the solution time to an acceptable level in order for the HGSADC to remain an operational decision-support tool. Another challenge when evaluating weather conditions further into the future, is that the weather states will become increasingly uncertain, such that some form of stochastic programming might become necessary.

Bibliography

Aas, B., Gribkovskaia, I., Halskau Sr, Ø., Shlopak, A., 2007. Routing of supply vessels to petroleum installations. International Journal of Physical Distribution & Logistics Management 37, 164–179.

Albjerk, N., Danielsen, T., Krey, S., Stålhane, M., Fagerholt, K., 2016. A vessel pickup and delivery problem from the disruption management in offshore supply vessel operations. International Conference on Computational Logistics , 50–64.

Andersson, H., Fagerholt, K., Hobbesland, K., 2015. Integrated maritime fleet deployment and speed optimization: Case study from roro shipping. Computers & Operations Research 55, 233–240.

Borthen, T., Loennechen, H., 2016. The Multi-objective Supply Vessel Planning Problem-A Hybrid Genetic Search Approach. Master's thesis. NTNU.

Borthen, T., Loennechen, H., Fagerholt, K., Wang, X., Vidal, T., 2019. Bi-objective offshore supply vessel planning with costs and persistence objectives. Computers & Operations Research 111, 285–296.

Borthen, T., Loennechen, H., Wang, X., Fagerholt, K., Vidal, T., 2018. A genetic search-based heuristic for a fleet size and periodic routing problem with application to offshore supply planning. EURO Journal on Transportation and Logistics 7, 121–150.

Christiansen, M., Fagerholt, K., Nygreen, B., Ronen, D., 2007. Maritime transportation. Handbooks in operations research and management science 14, 189–284.

Equinor, 2017. Skandi mongstad at the norne field (october 2013). URL: `https://www.equinor.com/en/news/`

awarding-contracts-five-ship-owners.html. [Online; accessed 15th October, 2019].

Equinor, 2019. Climate & sustainability. URL: https://www.equinor.com/no/how-and-why/sustainability.html.

Fagerholt, K., 2001. Ship scheduling with soft time windows: An optimisation based approach. European Journal of Operational Research 131, 559–571.

Fagerholt, K., Laporte, G., Norstad, I., 2010. Reducing fuel emissions by optimizing speed on shipping routes. Journal of the Operational Research Society 61, 523–529.

Fagerholt, K., Lindstad, H., 2000. Optimal policies for maintaining a supply service in the norwegian sea. Omega 28, 269–275.

Gribkovskaia, I., Halskau sr, Ø., Laporte, G., Vlček, M., 2007. General solutions to the single vehicle routing problem with pickups and deliveries. European Journal of Operational Research 180, 568–584.

Halvorsen-Weare, E.E., Fagerholt, K., 2011. Robust supply vessel planning, in: International Conference on Network Optimization, Springer. pp. 559–573.

Halvorsen-Weare, E.E., Fagerholt, K., 2017. Optimization in offshore supply vessel planning. Optimization and Engineering 18, 317–341.

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.

Hamming, R.W., 1950. Error detecting and error correcting codes. The Bell system technical journal 29, 147–160.

Hashimoto, H., Ibaraki, T., Imahori, S., Yagiura, M., 2006. The vehicle routing problem with flexible time windows and traveling times. Discrete Applied Mathematics 154, 2271–2290.

Hvattum, L.M., Norstad, I., Fagerholt, K., Laporte, G., 2013. Analysis of an exact algorithm for the vessel speed optimization problem. Networks 62, 132–135.

Kisialiou, Y., Gribkovskaia, I., Laporte, G., 2018a. The periodic supply vessel planning problem with flexible departure times and coupled vessels. Computers & Operations Research 94, 52–64.

Kisialiou, Y., Gribkovskaia, I., Laporte, G., 2018b. Robust supply vessel routing and scheduling. Transportation Research Part C: Emerging Technologies 90, 366–378.

Lindstad, H., Asbjørnslett, B.E., Jullumstrø, E., 2013. Assessment of profit, cost and emissions by varying speed as a function of sea conditions and freight market. Transportation Research Part D: Transport and Environment 19, 5–12.

Moan, A.B., Ødeskaug, P., 2019. Operational supply vessel planning problem with speed optimization .

Norlund, E.K., Gribkovskaia, I., 2013. Reducing emissions through speed optimization in supply vessel operations. Transportation Research Part D: Transport and Environment 23, 105–113.

Norlund, E.K., Gribkovskaia, I., 2017. Environmental performance of speed optimization strategies in offshore supply vessel planning under weather uncertainty. Transportation Research Part D: Transport and Environment 57, 10–22.

Norlund, E.K., Gribkovskaia, I., Laporte, G., 2015. Supply vessel planning under cost, environment and robustness considerations. Omega 57, 271–281.

Norstad, I., Fagerholt, K., Laporte, G., 2011. Tramp ship routing and scheduling with speed optimization. Transportation Research Part C: Emerging Technologies 19, 853–865.

Norwegian Petroleum Directorate, 2019a. Emissions to air. URL: https://www.norskpetroleum.no/en/environment-and-technology/emissions-to-air/.

Norwegian Petroleum Directorate, 2019b. Exports of oil and gas. URL: https://www.norskpetroleum.no/en/production-and-exports/exports-of-oil-and-gas/.

Psaraftis, H.N., Kontovas, C.A., 2014. Ship speed optimization: Concepts, models and combined speed-routing scenarios. Transportation Research Part C: Emerging Technologies 44, 52–69.

Seabrokers Group, 2018. Seabreeze. URL: https://www.seabrokers.no/wp-content/uploads/Seabreeze-September-2.pdf.

Shyshou, A., Gribkovskaia, I., Laporte, G., Fagerholt, K., 2012. A large neighbourhood search heuristic for a periodic supply vessel planning problem arising in offshore oil and gas operations. INFOR: Information Systems and Operational Research 50, 195–204.

Stålhane, M., Albjerk, N., Danielsen, T., Krey, S., Fagerholt, K., 2019. A variable neighbourhood search heuristic for disruption management in offshore oil and gas logistics. Journal of the Operational Research Society 70, 588–600.

Vidal, T., Crainic, T.G., Gendreau, M., Lahrichi, N., Rei, W., 2012. A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. Operations Research 60, 611–624.

Vidal, T., Crainic, T.G., Gendreau, M., Prins, C., 2014. A unified solution framework for multi-attribute vehicle routing problems. European Journal of Operational Research 234, 658–673.

Vidal, T., Gribel, D., Jaillet, P., 2019. Separable convex optimization with nested lower and upper constraints. INFORMS Journal on Optimization 1, 71–90.

Appendix A

Pseudocodes

Pseudocode for Arc-Generation Procedure

Algorithm 11 Arc-Generation Procedure

procedure $GenerateAllArcs()$
    for vessel $v \in \mathcal{V}$ **do** ▷ all available vessels
        $Nodes \leftarrow node(t^{startPrep}, depot, v)$ ▷ add depot node to the set of nodes
        for $t \in [t^{StartPrep}, t_v^{MaxDur}]$ **do** ▷ all discrete times for vessel v
            for installation $i \in \mathcal{N}$ **do** ▷ all installations and depot
                if $node(t, i, v) \in Nodes$ **then**
                    $BuildArcsFromNode(t, i, v)$ ▷ See Algorithm 1
                end if
            end for
        end for
    end for
end procedure

Pseudocode for Arc-Generation From Node

Algorithm 12 Arc-Generation From Node

```
procedure BuildArcsFromNode(t^start, i^dep, v))
    for i^dep ∈ AllInstallationsWithDepot do
        t^max ← latest theoretical end time for the arc
        t^end ← earliest feasible end time
        while t^end ≤ t^max do
            t^Service, C^Service ← CalculateServicing()          ▷ Appendix A.1
            if there distance/(t_end − t_start) > max_speed given weather then
                increment t^end and restart while loop
            end if
            t^Arrival, C^Idling ← CalculateIdling()              ▷ Appendix A.1
            C^Sailing ← CalculatelSailing()                      ▷ Appendix A.1
            Network ← Add arc(t^start, t^end, v, i^dep, i^dest, C^Sail, C^Idle, C^Service)
            Nodes ← Add node(t^end, i^dest, v)
            increment t^end
        end while
    end for
end procedure
```

Description of Supporting Algorithms in Arc-Generation

Table A.1: Description of how *CALCULATESERVICING*(), *CALCULATEIDLING*() and *CALCULATELSAILING*() works in order to calculate the correct fuel consumption and time points for a sailing leg.

CALCULATESERVICING()	-	Determines the discrete-time where the servicing ends by checking if servicing is feasible for the time period leading up to that point. If servicing is feasible for an end of servicing point in time, the starting point of the servicing time is calculated along with the servicing cost. Both servicing duration and consumption are adjusted for the weather states.
CALCULATEIDLING()	-	Checks whether the duration from the start of the arc is longer than the time it takes to sail the distance at the slowest possible speed. If so, the latest possible arrival time is determined from the minimum speed and distance, and for the remaining time before servicing starts, the idling consumption is calculated. The idling consumption is adjusted for the weather states.
CALCULATELSAILING()	-	First, the amount of time spent in each weather state for the whole duration of the sailing leg is calculated. Then the average sailing speed for the sailing leg is calculated given the distance, departure time, and arrival time. If this average speed is larger than the speed limit in some of the weather states, the speed sailed in the less restrictive weather states are increased to comply with max-speed restrictions. If the final, adjusted speed is above the max speed, the algorithm breaks, and the arc is not created. Otherwise, the sailing consumption is calculated by plotting the speed plus the weather offset for each weather state into the fuel consumption function and multiplying it by the time spent in that weather state.

Pseudocode for the HGSADC

Algorithm 13 Hybrid Genetic Search with Adaptive Diversity Control

Create initial population using construction heuristic ▷ Section 6.3
while *Iterations without improvement* $< I^{NoImp}$ and $time < T^{MAX}$ **do**
    Select parent individuals s_1 and s_2 ▷ Section 6.5
    Generate offspring s_{new} from s_1 and s_2 (crossover) ▷ Section 6.5
    Educate offspring s_{new} with probability p^{EDU} ▷ Section 6.6
    if s_{new} is infeasible **then**
        Repair s_{new} with probability p^{REP} ▷ Section 6.6.4
    end if
    if s_{new} is still infeasible **then**
        Add s_{new} to the infeasible subpopulation
    else
        Add s_{new} to the feasible subpopulation
    end if
    if maximum subpopulation size $\mu + \lambda$ reached **then**
        Perform survivor selection ▷ Section 6.7.1
    end if
    Adjust feasibility violation penalty parameters ▷ Section 6.7.2
    if the best individual has not been improved for I^{DIV} iterations **then**
        Diversify population ▷ Section 6.7.3
    end if
    Return *best feasible individual*
end while

Pseudocode for Creating The Initial Population

Algorithm 14 Construction heuristic

$individualsCreated \leftarrow 0$
while $individualsCreated < K^{INIT}\mu$ **do**
    STEP 1: CREATE INDIVIDUAL s:
    for $v \in \mathcal{R}(s)$ **do** ▷ Initiate empty voyages for all vessels
        $r_v \leftarrow \emptyset$
    end for
    for $i \in \mathcal{N}^I$ **do** ▷ Allocate installations to random vessels
        $v \leftarrow$ random vessel in $\mathcal{R}(s)$
        Add installation i to end of voyage r_v
    end for
    for $v \in \mathcal{R}(s)$ **do** ▷ Randomize installation visit sequence in each voyage
        Shuffle the installation visit sequence in voyage r_v
    end for
    Educate individual s

    STEP 2: INSERT INDIVIDUAL s TO SUBPOPULATION:
    if Individual s is infeasible **then**
        Repair s with probability p^{REP}
    end if
    if Individual s is still infeasible **then**
        Insert s into $\mathcal{S}^{INFEASIBLE}$
    else
        Insert s into $\mathcal{S}^{FEASIBLE}$
    end if
    $individualsCreated \leftarrow individualsCreated + 1$
end while

Pseudocode for the Crossover Procedure

Algorithm 15 Crossover procedure, Part 1

STEP 0: INHERITANCE RULE
Let n_1 and n_2 be the lowest and highest of two random integers between 0 and $|\mathcal{V}|$
Assign n_1 random vessels in $\mathcal{V}$ to the set Λ_1
Assign $n_2 - n_1$ of the remaining vessels in $\mathcal{V}$ randomly to the set Λ_2
Assign remaining vessels in $\mathcal{V}$ to the set Λ_{MIX}

STEP 1: INHERITANCE FROM PARENT s_1
for $v \in \Lambda_1$ **do**
  Copy the installation visit sequence of voyage r_v from s_1 to s_{new}
end for
for $v \in \Lambda_{MIX}$ **do**
  Create two random cutting points σ_v^1 and σ_v^2 in voyage r_v in s_1
  if $\sigma_v^2 > \sigma_v^1$ **then**
    Copy the subsequence of installations from σ_v^1 (including) to σ_v^2 (including) in voyage r_v from s_1 to s_{new}
  else
    Copy the voyage r_v in s_1 to s_{new}, but remove the subsequence of installations from σ_v^1 (excluding) to σ_v^2 (excluding) in voyage r_v from s_1 to s_{new}
  end if
end for

STEP 2: INHERITANCE FROM PARENT s_2
for $v \in \Lambda_1$ **do**
  Copy the installation visit sequence of voyage r_v from s_2 to s_{new}
end for
for $v \in \Lambda_{MIX}$ **do**
  for installation i in r_v in parent s_2 **do**
    if installation i is not already assigned to a voyage in the offspring s_{new} **then**
      add the installation i to the end of voyage r_v in s_{new}
    end if
  end for
end for
... ▷ **Continued on next page**

Algorithm 16 Crossover procedure, Part 2

```
...                                                        ▷ First part on previous page
STEP 3: ASSIGN REMAINING INSTALLATIONS
while any installation i from any parent individual is not yet assigned to a voyage of a
PSV in s_new do
    Find the voyage and the position to insert the installation yielding the lowest
    penalized cost of individual s_new
end while
```

Pseudocode for Voyage Reduction

Algorithm 17 Voyage reduction

```
if departingVessels > 1 then
    penalizedCostBeforeReduction = getPenalizedCost(s)          ▷ Section 6.4.1
    s_copy ← s                                                   ▷ Make a copy of individual s

    STEP 1: FIND THE VOYAGE TO REMOVE FROM THE CHROMOSOME
    r_voyageToRemove = findMostExpensiveShortestVoyage(ℛ(s_copy))

    STEP 2: REASSIGN THE INSTALLATIONS IN THE TERMINATED VOYAGE
            TO THE REMAINING VOYAGES
    for installation i ∈ r_voyageToRemove do
        bestVessel = findCheapestVesselInsertion(i)
        bestPosition = findCheapestPositionToInsertInst(r_bestVessel)
        Insert installation i into voyage r_bestVessel at position bestPosition
    end for
    Delete r_voyageToRemove from ℛ(s_copy)

    STEP 3: EVALUATE AND PERFORM
    penalizedCostAfterReduction = getPenalizedCost(s_copy)
    if penalizedCostAfterReduction < penalizedCostBeforeReduction then
        Return individual s_copy                                 ▷ Return new individual
    else
        Return individual s                                      ▷ Return original individual
    end if
end if
```

Pseudocode for Intravoyage Improvement

Algorithm 18 Intravoyage Improvement

```
for voyage $r_v \in \mathcal{R}(s)$ do
    $newVoyage \leftarrow r_v$
    $originalPenalizedCost \leftarrow getPenalizedCost(r_v)$          ▷ Section 6.4.1
    if voyage $r_v$ is not empty then
        $\mathcal{N}^{UntreatedInsts} \leftarrow r_v$
        while $|\mathcal{N}^{UntreatedInsts}| > 1$ do
            installation $m = pickAndRemoveRandomInst(\mathcal{N}^{UntreatedInsts})$
            $\mathcal{N}^{Neighbours} \leftarrow \mathcal{N}^{UntreatedInsts} \setminus \{m\}$
            while $|\mathcal{N}^{Neighbours}| > 0$ do
                installation $n = pickAndRemoveRandomInst(\mathcal{N}^{Neighbours})$
                while $allMovedExplored \neq$ true do
                    $newVoyage = doRandomMove(m, n)$          ▷ Section 6.6.2
                    $newPenalizedCost \leftarrow getPenalizedCost(newVoyage)$
                    if $newPenalizedCost < originalPenalizedCost$ then
                        Perform move on voyage $r_v$
                        Break
                    end if
                end while
                $\mathcal{N}^{Neighbours} \leftarrow \mathcal{N}^{Neighbours} \setminus \{n\}$
            end while
        end while
    end if
end for
```

$\mathcal{N}^{UntreatedInsts}$ - Set of installations that has not been checked for improvements yet

Pseudocode for Intervoyage Improvement

Algorithm 19 Intervoyage Improvement

$bestPenalizedCost \leftarrow getPenalizedCost(s)$ ▷ Section 6.4.1
$s_{best} \leftarrow s$ ▷ Copy individual s
for voyage $r_v \in \mathcal{R}(s)$ **do**
    for installation visit $i \in r_v$ **do**
        for voyage $r_v^* \in \mathcal{R}(s) \setminus \{r_v\}$ **do**
            for position $k \in r_v^*$ **do**
                $s_{new} \leftarrow s$ ▷ Copy individual s
                insert installation i at position k in voyage r_v^* for individual s_{new}
                $newPenalizedCost \leftarrow getPenalizedCost(s_{new})$
                if $newPenalizedCost < bestPenalizedCost$ **then**
                    $bestPenalizedCost \leftarrow newPenalizedCost$
                    $s_{best} \leftarrow s_{new}$
                end if
            end for
        end for
    end for
end for
Return s_{best}

Pseudocode for the Dynamic Adjustment of Penalty Parameters

Algorithm 20 Dynamic adjustment of penalty parameters

```
for p = Q, D, Z do
    if ξ^p ≤ ξ^REF − 0.05 then
        ω^p = ω^p ζ^UP                ▷ Feasible share too low, increase penalty
    end if
    if ξ^p ≥ ξ^REF + 0.05 then
        ω^p = ω^p ζ^DOWN              ▷ Feasible share too high, decrease penalty
    end if
end for
```

Appendix B

Arc-Flow Model for the OSVPPSO

Sets & Indices

$\mathcal{N}$	-	Set of all onshore and offshore installations i in the problem instance. Including both the supply depot & all offshore installations to visit on the voyages starting the next departure day
$\mathcal{T}$	-	Set of discrete time points until the latest return time of any vessels departing the next departure day (e.g. hours)
$\mathcal{T}^{D}_{ijv}$	-	Subset of $\mathcal{T}$ with all possible departure times for a vessel v sailing from installation i to installation j. Visualized in Figure 4.4
$\mathcal{T}^{SD}_{ijtv}$	-	Subset of $\mathcal{T}$ with all possible departure times for a vessel v sailing from installation i and finishing at installation j at time t. Visualized in Figure 4.5
$\mathcal{T}^{SF}_{itjv}$	-	Subset of $\mathcal{T}$ with all possible finishing times for a vessel v sailing from installation i at time t to installation j. Visualized in Figure 4.6
$\mathcal{G}_{v}$	-	The network containing all legal arcs, i.e. sailing legs, from installation i to j at all departure times t for a vessel v. Illegal arcs resulting from closing hours at installations or weather prohibiting installation service are removed from the network
$\mathcal{V}$	-	Set of all vessels available the next departure day, including vessels from the spot market

Parameters

D_i	-	Demand at installation i, i.e. the number of standard unit containers to be delivered to installation i on a voyage departing the next day
T_i^D	-	Deadline for a delivery to installation i
$\overline{T}_v$	-	Maximum duration of a voyage for vessel v, i.e. the number of discrete time points before vessel v must be docked at the supply depot after starting a voyage. This is equivalent to the number of discrete time points until the return time
Q_v	-	Cargo capacity of supply vessel v, i.e. maximum number of standard unit containers
$C^F_{itjt'v}$	-	Cost of fuel consumed when sailing from installation i at time t to installation j, finishing the service job at time t'
$C^V_{tt'v}$	-	Cost of hiring the vessel for the whole time period of the arc, when sailing with vessel v from time t to time t'. For vessels hired on a long-term contract, this cost will be equal to zero, whilst vessels hired on spot price will have a cost assigned in this parameter

Decision Variables

$$x_{itjt'v} = \begin{cases} 1 & \text{if vessel } v \text{ travels along the arc where it starts at installation } i \text{ at time } t, \\ & \text{then sails to installation } j\text{, finishing service at installation } j \text{ at time } t' \\ 0 & \text{otherwise} \end{cases}$$

Objective

$$min\ z = \sum_{v \in V} \sum_{((i,t),(j,t')) \in \mathcal{G}_v} (C^F_{itjt'v} + C^V_{tt'v}) x_{itjt'v} \tag{B.1}$$

Constraints

$$\sum_{j\in\mathcal{N}}\sum_{t'\in\mathcal{T}^{SD}_{jitv}} x_{jt'itv} - \sum_{j\in\mathcal{N}}\sum_{t'\in\mathcal{T}^{SF}_{itjv}} x_{itjt'v} = 0, \qquad i\in\mathcal{N}\backslash\{0\}, t\in\mathcal{T}, v\in\mathcal{V} \tag{B.2}$$

$$\sum_{v\in\mathcal{V}}\sum_{i\in\mathcal{N}}\sum_{t\in\mathcal{T}^{D}_{ijv}}\sum_{t'\in\mathcal{T}^{SF}_{itjv}} x_{itjt'v} = 1, \qquad j\in\mathcal{N}\setminus\{0\} \tag{B.3}$$

$$\sum_{i\in\mathcal{N}}\sum_{t\in\mathcal{T}^{D}_{ijv}}\sum_{t'\in\mathcal{T}^{SF}_{itjv}} t'x_{itjt'v} \le T^{D}_{j}, \qquad j\in\mathcal{N}\setminus\{0\}, v\in\mathcal{V} \tag{B.4}$$

$$\sum_{((i,t),(j,t'))\in\mathcal{G}_v} D_j x_{itjt'v} \le Q_v, \qquad v\in\mathcal{V} \tag{B.5}$$

$$\sum_{i\in\mathcal{N}}\sum_{t\in\mathcal{T}^{D}_{ijv}}\sum_{t'\in\mathcal{T}^{SF}_{itjv}} t'x_{it0t'v} \le \overline{T}_v, \qquad v\in\mathcal{V} \tag{B.6}$$

$$x_{itjt'v} = \{0,1\}, \qquad ((i,t),(j,t'))\in\mathcal{G}_v, v\in\mathcal{V} \tag{B.7}$$

Appendix C

Mathematical Formulation of the Weather-Dependent Supply Vessel Speed Optimization Problem

Sets

$\mathcal{L}$	-	The ordered set of sailing legs l in a voyage
$\mathcal{A}$	-	The set of activities, a, that may be performed by a vessel on any sailing leg, i.e. supply depot preparation, sailing, idling, and servicing. Shown in Table 5.1
$\mathcal{W}$	-	The set of weather states, w, as described in Table 4.1
$\mathcal{T}^{WF}$	-	Set of discrete time periods *t* for the weather forecast (e.g. hours). It is assumed that a solution with exceeding the length of the weather forecast is of no interest. Spot vessels can be added in the master problem to obtain shorter voyages, eliminating this problem

Table C.1: Description of the set of activities, $\mathcal{A}$

Activity, a	**Description**
0	Supply depot preparation
1	Sailing
2	Idling (waiting)
3	Servicing an installation
4	Artificial activity denoting ”end of service,” i.e. that a vessel is ready to depart for the next sailing leg

Parameters

I_{aw}^{FC}	-	Impact on fuel consumption on activity a from weather state w
I_{aw}^{T}	-	Impact on time needed to perform activity a in weather state w
I_{w}^{MS}	-	Impact on the maximum sailing speed limit for a vessel in weather state w
δ_{tw}^{WS}	-	Binary parameter equal to 1 when the weather state is equal to weather state w at time t, 0 otherwise
δ_{lt}^{SI}	-	Binary parameter equal to 1 if service on sailing leg l is infeasible during the time period t, 0 otherwise. For this parameter service infeasibility may be a result of a closed installation or poor weather conditions
δ^{Spot}	-	Binary parameter equal to 1 if the vessel used is a spot vessel, 0 otherwise
$\overline{T}$	-	The latest time at which the vessel should be returned at the supply depot, i.e. the maximum voyage duration. This limit can be violated at a penalty cost per unit violated
T_{l}^{D}	-	The deadline for the delivery to the installation serviced on sailing leg l. This limit can be violated at a penalty cost per unit violated
$T_{l}^{Service}$	-	The time needed to perform the service job for the installation visit on sailing leg l under perfect weather conditions
C_a	-	Monetary cost of fuel per time unit for activity a
$C^{Charter}$	-	Monetary cost of chartering a spot vessel per time unit
$C^{Pen,\overline{T}}$	-	Penalty for violating the maximum duration per time unit
C^{Pen,T^D}	-	Penalty for violating a delivery deadline per time unit
D_l	-	Distance on sailing leg l
V^{Max}, V^{Min}	-	Maximum and minimum limits on vessel speed, respectively

Decision Variables

τ_{la}	-	Continuous decision variable denoting at which time activity a starts on sailing leg l

Helping variables

$$x_{lat} = \begin{cases} 0 & \text{if } t < \lfloor \tau_{la} \rfloor \quad \text{or} \quad t \geq \lceil \tau_{l(a+1)} \rceil \\ t - \tau_{la} + 1 & \text{if } t = \lfloor \tau_{la} \rfloor \quad \& \quad t! = \lfloor \tau_{l(a+1)} \rfloor \\ \tau_{l(a+1)} - \tau_{la} & \text{if } \lfloor \tau_{la} \rfloor = \lfloor \tau_{l(a+1)} \rfloor \\ \tau_{l(a+1)} - t & \text{if } t = \lfloor \tau_{l(a+1)} \rfloor \quad \& \quad t! = \lfloor \tau_{la} \rfloor \\ 1 & \text{if } t < \lfloor \tau_{l(a+1)} \rfloor \quad \& \quad t \geq \lceil \tau_{la} \rceil, \end{cases} \tag{C.1}$$

$$l \in \mathcal{L} \setminus \{|\mathcal{L}|\}, a \in \mathcal{A} \setminus \{4\}, t \in \mathcal{T}^{WF}$$

$$y_{law} = \sum_{t \in \mathcal{T}^{WF}} \frac{\delta_{tw}^{WS} x_{lat}}{\tau_{l(a+1)} - \tau_{la}}, \qquad l \in \mathcal{L}, a \in (1,3) \tag{C.2}$$

$$v_l^{*0} = \frac{D_l}{\tau_{l2} - \tau_{l1}}, \qquad l \in \mathcal{L} \tag{C.3}$$

$$v_l^{*1} = \begin{cases} v_l^{*0} + \frac{(v_l^{*0} - (V^{Max} - I_3^{MS}))y_{l13}}{1 - y_{l13}} & \text{if } v_l^{*0} > V^{Max} - I_3^{MS} \\ v_l^{*0} & \text{otherwise,} \end{cases} \qquad l \in \mathcal{L} \tag{C.4}$$

$$v_l^{*2} = \begin{cases} v_l^{*1} + \frac{(v_l^{*1} - (V^{Max} - I_2^{MS}))y_{l12}}{1 - (y_{l12} + y_{l13})} & \text{if } v_l^{*1} > V^{Max} - I_2^{MS} \\ v_l^{*1} & \text{otherwise,} \end{cases} \qquad l \in \mathcal{L} \tag{C.5}$$

$$v_{lt} = \begin{cases} V^{Max} - I_3^{MS} & \text{if } v_l^{*2} > V^{Max} - I_3^{MS} \\ V^{Max} - I_2^{MS} & \text{if } v_l^{*2} > V^{Max} - I_2^{MS} \\ v_l^{*2} & \text{otherwise,} \end{cases} \quad l \in \mathcal{L}, t \in \mathcal{T}^{WF} \tag{C.6}$$

Objective

$$\begin{aligned} min\ z = & \sum_{l \in \mathcal{L}} \left(f^{Prep}(l) + f^{Sail}(l) + f^{Idle}(l) + f^{Service}(l) \right) \\ & + \sigma^{\overline{T}} \\ & + \sigma^{T^D} \\ & + C^{Charter} \delta^{Spot} (\tau_{|\mathcal{L}|4} - \tau_{00}) \end{aligned} \tag{C.7}$$

$$f^{Prep}(l) = \sum_{t \in \mathcal{T}^{WF}} C_0 x_{l0t} \tag{C.8}$$

$$f^{Sail}(l) = \sum_{t \in \mathcal{T}^{WF}} g(v_{lt} + I_w^{MS} \delta_{tw}^{WS}) x_{l1t} \tag{C.9}$$

$$f^{Idle}(l) = \sum_{t \in \mathcal{T}^{WF}} I_{2w}^T \delta_{tw}^{WS} C_2 x_{l2t} \tag{C.10}$$

$$f^{Service}(l) = \sum_{t \in \mathcal{T}^{WF}} I_{3w}^T \delta_{tw}^{WS} C_3 x_{l3t} \tag{C.11}$$

$$\sigma^{\overline{T}} = C^{Pen,\overline{T}} max\{0, \tau_{|\mathcal{L}|4} - \overline{T}\} \tag{C.12}$$

$$\sigma^{T^D} = C^{Pen,T^D} \sum_{l \in \mathcal{L}} max\{0, \tau_{l3} - T_l^D\} \tag{C.13}$$

Problem Specific Cost Function

$$g(v) = p_1 v^2 + p_2 v + p_3, \tag{C.14}$$

where p_1, p_2, p_3 are the coefficients.

Constraints

$$\tau_{la} \leq \tau_{l(a+1)}, \qquad l \in \mathcal{L}, a \in \mathcal{A} \setminus \{4\} \tag{C.15}$$

$$\tau_{l4} = \tau_{(l+1)0}, \qquad l \in \mathcal{L} \setminus \{\mathcal{L}\} \tag{C.16}$$

$$v_l^{*2} \leq V^{Max}, \qquad l \in \mathcal{L} \tag{C.17}$$

$$v_l^{*2} \geq V^{Min}, \qquad l \in \mathcal{L} \tag{C.18}$$

$$\sum_{t \in \mathcal{T}^{WF}} \delta_{lt}^{SI} x_{l3t} = 0, \qquad l \in \mathcal{L} \tag{C.19}$$

$$\tau_{(l+1)0} - \tau_{l3} = T_l^{Service} \sum_{w \in \mathcal{W}} I_{3w}^T y_{l3w}, \qquad l \in \mathcal{L} \tag{C.20}$$

$$0 \leq x_{lat} \leq 1, \qquad l \in \mathcal{L} \setminus \{|\mathcal{L}|\}, a \in \mathcal{A} \setminus \{4\}, t \in \mathcal{T}^{WF} \tag{C.21}$$

$$0 \leq \tau_{la} \leq |\mathcal{T}^{WF}|, \qquad l \in \mathcal{L}, a \in \mathcal{A} \tag{C.22}$$

Appendix D

Parameter Tuning

In this appendix the full list of averaged relative solution times and objective value is presented for each of the tuned parameter sets. Also, more decimals are shown for the averaged relative objective value so that the small differences not shown in Chapter 8 may be studied.

Tuning Parameters

Table D.1: Initial and final parameter values

	Parameter	Initial Value	Final Value	Description
Population management	μ	25	25	Minimum subpopulation size
	λ	75	100	Generation size
	ξ^{REF}	0.5	0.4	Target proportion of feasible individuals
	K^{INIT}	4	4	Construction heuristic size multiplier
Diversification	K^{DIV}	4	4	Diversification size multiplier
	η^{ELI}	0.4	0.5	Proportion of elite individuals, $n^{ELI} = \eta^{ELI}\|S\|$
	η^{CLO}	0.2	0.3	Proportion of individuals considered for diversity evaluation, $n^{CLO} = \eta^{CLO}\mu$
	I^{DIV}	500	400	Iterations before diversification
Rates & Probabilities	p^{EDU}	0.5	0.75	Education rate, i.e. the probability of an individual undergoing education
	p^{REP}	0.5	0.25	Repair rate, i.e. the probability of an infeasible individual undergoing repair
Penalty	ω^{Q}	1000	500	Initial penalty per unit for violating the capacity limit
	$\omega^{\overline{T}}$	500	500	Initial penalty per unit for violating the maximum duration
	ω^{D}	200	250	Initial penalty per unit for violating the delivery deadline
	κ	100	100	Number of newest individuals considered for penalty adjustment
	ζ^{UP}	1.2	1.2	Factor to increase penalties with to reach target proportion of feasible individuals
	ζ^{DOWN}	0.85	0.85	Factor to decrease penalties with to reach target proportion of feasible individuals
Stopping criteria	I^{NoImp}	2000	5000	Iterations without improvement before termination
	T^{MAX}	3600	3 600	Maximum run time (seconds)
Discretization	θ	4	4	Number of discrete time points per hour

Tuning of Discrete Time Points Per Hour

Table D.2: Change in solution time and objective value from different discretization resolutions

Discreti-zation	**Increase in Solution Time**	**Change to Objective Value**
1	0%	0.000%
2	179%	-2.247%
3	525%	-2.589%
4	962%	-3.034%
5	1259%	-3.110%
6	1818%	-3.251%
7	2154%	-3.332%
8	3045%	-3.375%
9	3634%	-3.353%
10	4322%	-3.393%

All Tuning Results for the Population Management Parameter Group

Table D.3: All results of the Population Management group

Parameters		**Avg % from best**	
μ	λ	Time	Objective
25	100	95%	0.000%
50	50	131%	0.000%
35	150	144%	0.000%
50	150	178%	0.000%
25	150	125%	0.003%
50	75	142%	0.004%
50	100	147%	0.006%
15	75	38%	0.008%
15	100	54%	0.018%
15	150	81%	0.018%
35	50	91%	0.018%
35	100	112%	0.018%
15	50	26%	0.020%
25	75	81%	0.066%
35	75	105%	0.078%
25	35	60%	0.104%

All Tuning Results for the Diversification Parameter Group

Table D.4: All results for the Diversification group

Parameters			**Avg % from best**	
η^{ELI}	η^{CLO}	I^{DIV}	Time	Objective
0.5	400	0.3	48.38%	0.000%
0.4	300	0.3	81.07%	0.000%
0.4	400	0.2	63.66%	0.000%
0.3	400	0.1	73.33%	0.000%
0.5	200	0.3	109.75%	0.000%
0.4	400	0.3	56.06%	0.002%
0.5	400	0.1	40.91%	0.003%
0.7	300	0.1	46.41%	0.003%
0.7	300	0.3	47.20%	0.003%
0.7	400	0.1	26.23%	0.003%
0.5	300	0.1	59.72%	0.003%
0.4	300	0.2	61.52%	0.003%
0.7	400	0.3	35.69%	0.005%
0.3	300	0.1	93.53%	0.009%
0.4	300	0.1	65.78%	0.013%
0.4	200	0.1	128.82%	0.014%
0.7	200	0.1	85.50%	0.018%
0.3	200	0.1	143.04%	0.019%
0.5	200	0.2	119.87%	0.019%
0.4	400	0.1	54.92%	0.021%
0.3	400	0.3	64.11%	0.021%
0.5	200	0.1	104.13%	0.022%
0.4	200	0.2	110.52%	0.023%
0.4	200	0.3	123.03%	0.028%
0.3	400	0.2	71.31%	0.033%
0.3	300	0.3	95.24%	0.034%
0.5	300	0.3	71.83%	0.036%
0.3	200	0.2	129.07%	0.045%
0.3	300	0.2	95.55%	0.045%
0.5	400	0.2	35.79%	0.069%
0.5	300	0.2	59.74%	0.072%
0.7	400	0.2	32.94%	0.081%
0.7	300	0.2	42.28%	0.133%
0.7	200	0.2	89.46%	0.138%
0.3	200	0.3	150.61%	0.187%
0.7	200	0.3	75.69%	0.220%

All Tuning Results for the Target Proportion of Feasible Individuals Parameter

Table D.5: All results for the Target Proportion of Feasible Individuals tuning

Parameter	**Avg % from best**	
ξ^{REF}	Time	Objective
0.4	71.58%	0.01%
0.5	52.49%	0.07%
0.6	45.31%	0.17%
0.7	53.53%	0.18%
0.3	51.59%	0.57%

All Tuning Results for the Education, Repair & Termination Parameter Group

Table D.6: Part 1: All results for the Education, Repair, Termination parameter group

Parameters			**Avg % from best**	
p^{EDU}	p^{REP}	I^{NoImp}	Time	Objective
0.75	0.25	5000	259.55%	0.000%
0	0.75	5000	269.49%	0.000%
0.25	0.75	5000	315.15%	0.000%
0.25	0.75	7500	433.86%	0.000%
0.75	0.75	7500	494.15%	0.000%
0.5	0.75	7500	496.42%	0.000%
1	0.5	5000	390.13%	0.002%
1	0.5	7500	555.16%	0.007%
1	0.75	7500	600.36%	0.008%
1	0.25	7500	525.88%	0.014%
0.25	0.5	7500	345.93%	0.033%
0.75	0.75	5000	363.64%	0.033%
0	0.75	7500	381.59%	0.033%
1	0.25	5000	405.46%	0.042%
0.75	0.5	5000	338.20%	0.046%
0.75	0.75	2500	250.52%	0.049%
0.75	0.25	7500	371.16%	0.057%
0.5	0.5	5000	296.10%	0.066%
0.5	0.25	2500	157.16%	0.099%
0.75	0.5	7500	444.01%	0.099%
1	0.75	5000	473.78%	0.099%
0.5	0.25	7500	340.86%	0.111%
0.75	0.5	2500	223.62%	0.143%
0.5	0.5	7500	368.75%	0.151%
0.5	0.75	5000	325.55%	0.151%
0.25	0.25	7500	718.15%	0.160%
0.5	0.75	2500	200.63%	0.161%
0.25	0.5	5000	257.30%	0.167%
...	...	...	...	...

Part 2: All results for the Education, Repair, Termination parameter group

Parameters			**Avg % from best**	
p^{EDU}	p^{REP}	I^{NoImp}	Time	Objective
...	...	...	...	...
1	0.75	2500	280.16%	0.170%
0	0.75	2500	151.79%	0.177%
0.5	0.25	5000	261.46%	0.212%
0.75	0.25	2500	177.41%	0.251%
0	0.5	5000	319.90%	0.253%
0.25	0.75	2500	193.80%	0.255%
1	0.25	2500	234.23%	0.256%
0	0.5	2500	111.23%	0.258%
0	0.5	7500	556.36%	0.315%
0.5	0.5	2500	177.07%	0.316%
0.25	0.25	2500	126.33%	0.330%
0.25	0.5	2500	132.71%	0.330%
0	0.25	5000	251.70%	0.393%
0.25	0.25	5000	440.79%	0.460%
0	0.25	2500	63.57%	0.493%
1	0.5	2500	267.13%	0.505%
0	0.25	7500	396.65%	0.661%

All Tuning Results for the Penalty Parameter Group

Table D.7: All results for the Penalty parameter group

Parameters			**Avg % from best**	
ω^Q	$\omega^{\overline{T}}$	ω^D	Time	Objective
500	500	250	40.23%	0.000%
500	100	100	43.04%	0.000%
500	100	250	44.29%	0.000%
500	100	500	44.40%	0.000%
500	250	500	46.38%	0.000%
100	500	250	48.96%	0.000%
500	500	500	49.52%	0.000%
100	100	100	50.12%	0.000%
100	500	100	50.90%	0.000%
500	500	100	57.47%	0.000%
1000	250	250	27.17%	0.003%
500	250	250	35.02%	0.003%
1000	500	250	36.21%	0.003%
100	250	500	49.46%	0.003%
100	250	250	53.55%	0.003%
500	250	100	50.49%	0.005%
100	100	500	55.70%	0.005%
100	100	250	51.26%	0.008%
100	250	100	48.72%	0.018%
100	500	500	48.24%	0.036%
1000	100	100	29.48%	0.064%
1000	500	500	39.01%	0.064%
1000	500	100	34.44%	0.064%
1000	250	500	39.56%	0.066%
1000	250	100	29.48%	0.136%
1000	100	250	39.79%	0.145%
1000	100	500	35.35%	0.191%

Appendix E

Computational Results

Technical Results

Table E.1: Comparison between HGSADC and commercial solver (Gurobi). The metaheuristic was run ten times for each problem instance to find average and minimum values.

	Commercial solver (Gurobi)			**HGSADC**			
#	**Time [s]**	**Upper bound**	**Optimality gap %**	**Avg. time [s]**	**Avg. obj. [$]**	**Min. obj. [$]**	**Diff.**
0	4.3	2388	0	10	2388	2388	0
1	5.1	2536	0	13	2536	2536	0
2	4.7	2064	0	11	2064	2064	0
3	577	3229	0	35	3229	3229	0
4	2792	11891	0	99	11891	11891	0
5	3600	4400	50.4	134	4223	4223	-177
6	3600	4528	42.5	129	4528	4528	-95
7	3600	15614	48.3	199	15173	15173	-441
8	3600	5136	39.3	167	5094	5094	-42
9	3600	15979	72.9	649	10910	10854	-5069
10	3600	15501	68.9	461	11415	11402	-4086
11	3600	8653	57.1	294	6582	6582	-2071
12	3600	6528	45.9	363	6468	6468	-60
13	3600	33153	89.2	498	7371	7370	-25782
14	3600	38034	90.2	485	6764	6764	-31270
15	3600	27346	85.6	688	7088	7086	-20258
16	3600	23871	80.5	1236	7611	7505	-16260
17	3600	NO SOL	NO SOL	1072	9345	9313	-
18	3600	NO SOL	NO SOL	2088	16488	16068	-
19	3600	NO SOL	NO SOL	1753	10715	10652	-
(0-16)	**2740**	**12997**	**45**	**321**	**6784**	**6774**	**-6212**

- Problem instance

Economical Evaluation of Speed Optimization and Operational Planning

Table E.2: Comparison between three different operational strategies for the *Bad* weather scenario.

	HGSADC Solution		**Design Speed Solution**		**Limited Operational Planning**	
#	**Obj.val. [$]**	**s.d.**	**Obj.val. [$]**	**s.d.**	**Obj.val. [$]**	**s.d.**
11	7757	0	8168	0	25838	5
12	7585	0	7956	0	29315	6
13	8440	0	20604	2	19098	3
14	8278	0	19931	2	22087	6
15	8802	0	21695	3	22487	4
16	15991	1	24290	4	30255	6
17	10935	0	22705	3	24971	4
Avg.	9684	0.1	17907	2.0	24865	4.9

s.d. - number of deliveries that have to be performed with spot vessels

Appendix F

Estimating the Value of Operational Planning and Weather-Dependent Speed Optimization

In Section 8.2.3, a penalized cost of performing installation visits with spot vessels was set. This was calculated by removing all available vessels from problem instance 0 - 7 and find the optimal solution using only spot vessels. Then, the average cost for all instances was divided by the average number of installation visits for all problem instances. The resulting cost of USD 3681 is an estimate of how much a single delivery with a spot vessel will cost on average. Instances 0-7 was used because the HGSADC in all known test runs have found the same best solution. Further, instances 0 - 7 have varying voyage lengths, such that the average cost per installation visits will give a reasonable picture of the real cost. The *Good* weather scenario was used in these tests.

Table F.1: Costs for problem instance 0 - 7 when only using spot vessels and the average costs per installation visit

Problem Instance	**Number of Installations Visits**	**Best Cost**	**Cost per Installation Visit**
0	3	16067	5356
1	5	17510	3502
2	5	16513	3303
3	7	22288	3184
4	7	30340	4334
5	9	30197	3355
6	9	31746	3527
7	11	31784	2889
Avg.	7	24556	3681

Appendix G

Installations in the Mongstad Case

Table G.1: Installations in the Mongstad Case

#	Installation
1	TRO (Troll A)
2	TRB (Troll B)
3	TRC (Troll C)
4	CPR (Cosl Promoter)
5	SEN (Songa Endurance)
6	SDO (Stena Don)
7	SEQ (Songa Equinox)
8	OSE (Oseberg A/D)
9	OSB (Oseberg B)
10	OSC (Oseberg C)
11	OSO (Oseberg Øst)
12	SSC (Safe Scandinavia)
13	OSS (Oseberg Sør)
14	DSD (Songa Delta)
15	KVB (Kvitebjørn)
16	VMO (Valemon)
17	WEL (West Eldara)
18	VFB (Veslefrikk B)
19	WEP (West Epsilon)
20	HUL (Huldra)
21	STA (Statfjord A)
22	STB (Statfjord B)
23	STC (Statfjord C)
24	GFA (Gullfaks A)
25	GFB (Gullfaks B)
26	GFC (Gullfaks C)
27	SOD (Songa Dee)

Appendix H

Code For Metaheuristic and Exact Solution Methods

Full code can be found at:

https://github.com/AndreasMoan/OSVPPSO

Intstructions are found in the *ReadMe*-file attached.

Andreas Bakke Moan & Pål Ødeskaug

A Hybrid Genetic Approach to the Operational Supply Vessel Planning Problem with Speed Optimization

NTNU
Norwegian University of
Science and Technology