UNIVERSITÄT
BIELEFELD

Faculty of Business Administration
and Economics

Dissertation

Stochastic Optimization for Maritime Logistics Problems

M.Sc. Stefan Kuhlemann

Submitted in fulfillment of the requirements for the academic degree of
Doctor rerum politicarum (Dr. rer. pol.)
to the Faculty of Business Administration and Economics,
Bielefeld University

April 19, 2021

Examiners:
1. Prof. Dr. Kevin Tierney
2. Prof. Dr. Achim Koberstein

Gedruckt auf alterungsbeständigem Papier °° ISO 9706

Acknowledgments

This thesis marks one of the last steps of an exciting chapter of my life. I look back on interesting years at the university and I would like to take the chance to thank the people who have accompanied and supported me along the journey, especially those named below.

In the first place, I thank my supervisor Kevin Tierney for giving me the opportunity to do my PhD, for his valuable advise and his focused guidance. It was a great honor for me to work with and for him over the past years. I also thank my co-supervisor Achim Koberstein for the insights into the academic world and the collaboration during my time in research. Furthermore, I thank Michael Römer for being my third examiner.

My interest in operations research would not have developed to the same extent it did without Leena Suhl and the team at the Decision Support & Operations Research Lab. Thanks to Lars Beckmann, Daniela Guericke, Corinna Hallmann, Florian Isenberg, Artus Krohn-Grimberghe, Marius Merschformann, Daniel Müller, Kostja Siefen, Florian Stapel, Carina Uhde, Peter Volmich, Christoph Weskamp, Henry Wolf and Michaela Wissing I had an excellent introduction into the field of operations research and decision support. Because of them I had a great start when I began to work at the chair and the encouragement to do a PhD. Special thanks go to Corinna Hallmann and Michaela Wissing, with whom I shared an office and who became very good friends for me. They supported me all the time, in working and also private concerns. I will never forget my time at this chair with its unique working atmosphere.

With time came the transition to a new chair, where Guido Schryen gave me the opportunity to take on new challenges. With him came the new colleagues Sascha Burmeister, Philipp Speckenmeier, Martina Sperling and Miriam Stumpe. They were an important part for creating the feeling of being part of a new chair. I really appreciat the good cooperation and discussions on our research projects that we had from time to time. Especially Miriam Stumpe helped me to stay motivated during the last part of my PhD.

I also became part of the team of Kevin Tierney's new chair at Bielefeld University consisting of Veronika Aufdemkamp, André Hottung, Sebastian Risse, Dimitri Weiß and Daniel Wetzel. I thank them for their support and enjoy remembering the time we spent together at conferences.

I would also like to thank Jana Ksciuk for the cooperation over the past years. It was nice to have someone working on similar topics and good to have someone to exchange ideas with.

Finally, I would like to thank my parents for their everlasting support as well as my sister and my friends for supporting me in so many different ways over the past years.

Stefan Kuhlemann
Paderborn, April 2021

Contents

1 Introduction **1**

2 Background **5**
- 2.1 Maritime Logistics Problems . . . 5
  - 2.1.1 Liner Shipping . . . 5
  - 2.1.2 Tramp Shipping . . . 6
  - 2.1.3 Industrial Shipping . . . 6
  - 2.1.4 Uncertainties in Maritime Logistics Problems . . . 6
- 2.2 Solution Techniques . . . 9
  - 2.2.1 Mathematical Optimization Models . . . 10
  - 2.2.2 Genetic Algorithm . . . 13

3 Overview of Research Papers **17**
- 3.1 Paper 1: The Stochastic Liner Shipping Fleet Repositioning Problem with Uncertain Container Demands and Travel Times . . . 17
- 3.2 Paper 2: A Genetic Algorithm for Finding Realistic Sea Routes Considering the Weather . . . 17
- 3.3 Paper 3: Exploiting Counterfactuals for Scalable Stochastic Optimization 18

4 The Stochastic Liner Shipping Fleet Repositioning Problem with Uncertain Container Demands and Travel Times **19**
- 4.1 Introduction . . . 19
- 4.2 Related Work . . . 21
  - 4.2.1 LSFRP . . . 21
  - 4.2.2 Maritime models with demand and/or sailing time uncertainty . 22
- 4.3 Problem Description . . . 24
- 4.4 Mathematical Model . . . 26
  - 4.4.1 Graph structure . . . 26
  - 4.4.2 Node flow model . . . 27
  - 4.4.3 Stochastic modeling . . . 27
  - 4.4.4 Mathematical model . . . 28
- 4.5 Computational Results . . . 33
  - 4.5.1 Experimental setup . . . 33
  - 4.5.2 Scenario generation . . . 34
  - 4.5.3 Scenario reduction . . . 35
  - 4.5.4 Delay penalties . . . 37
  - 4.5.5 Experimental results for all repositioning instances . . . 37

4.5.6 Case study . . . 44
4.6 Conclusion . . . 47

5 A Genetic Algorithm for Finding Realistic Sea Routes Considering the Weather 49
5.1 Introduction . . . 49
5.2 Weather Routing of Ships . . . 50
5.2.1 Computation of the Fuel Consumption . . . 51
5.3 Related Work . . . 52
5.3.1 Exact Approaches . . . 52
5.3.2 Single Objective Heuristic Approaches . . . 53
5.3.3 Multiobjective Evolutionary Approaches . . . 53
5.3.4 Related Problems . . . 54
5.4 A Genetic Algorithm For Realistic Weather-dependent Routes . . . 55
5.4.1 Initial Route Generation . . . 57
5.4.2 Crossover Operators . . . 57
5.4.3 Mutation Operators . . . 59
5.4.4 Stochastic Optimization . . . 60
5.5 Computational Results . . . 61
5.5.1 Parameters for the GA . . . 61
5.5.2 Weather Data . . . 61
5.5.3 Experimental Results for Weather Data with Perfect Information 62
5.5.4 Pirate Zones and Travel Time Limitations . . . 64
5.5.5 Experimental Results for Stochastic Weather Data . . . 66
5.6 Conclusion . . . 71

6 Exploiting Counterfactuals for Scalable Stochastic Optimization 73
6.1 Introduction . . . 73
6.1.1 Stochastic Optimization . . . 73
6.1.2 Multi-stage Stochastic Optimization . . . 74
6.1.3 Simulation-based Optimization . . . 75
6.2 Technology Gaps . . . 75
6.3 Learning From Counterfactuals . . . 76
6.4 Stochastic Knapsack . . . 79
6.4.1 Stochastic Environment . . . 79
6.4.2 Winner Forecasting . . . 80
6.4.3 Numerical Results . . . 81
6.5 RCPSP with Uncertain Job Durations . . . 83
6.5.1 Stochastic Environment . . . 84
6.5.2 Winner Forecasting . . . 85
6.5.3 Numerical Results . . . 85

6.6 Stochastic Shortest Path Problem 86
    6.6.1 Stochastic Environment 87
    6.6.2 Winner Forecasting 87
    6.6.3 Numerical Results 87
6.7 Conclusion 88

7 Conclusion and Outlook **91**

Bibliography **95**

List of Figures

2.1 Maritime logistics problems including uncertainties 7

4.1 Repositioning route, initial service and goal service for Instance 4-58-125-0 assuming deterministic travel times 26
4.2 Comparison of routes of four vessels of deterministic and stochastic solutions for instance 4-58-125-0 assuming uncertain travel times in a time-space graph. 47

5.1 Visualization of the generation of an initial route from Perth, Australia to Brisbane, Australia in 12 steps. 58
5.2 Visualization of the crossover operator. 60
5.3 Route from Hamburg to Reykjavík and the SECA-zone 63
5.4 Route optimized with additional costs for security 66
5.5 Visualization of the solutions for the instances. 67
5.6 The four different routes from USNYC (New York, USA) to SRPBM (Paramaribo, Suriname). “PI”, “Stochastic”, “Stoch. NR” and “No WO”. 68

List of Tables

4.1 Summary of papers working on LSFRP related problems with uncertain input parameters. 23

4.2 Structure of the instances 34

4.3 Relative standard deviation of the average expected value out-of-sample and in-sample for instance 4-58-125-0 (15 test runs, $k = 50$) and deviation from the optimal solution (Dev.Opt.%) 36

4.4 Relative standard deviation of the average expected value out-of-sample and in-sample for instance 6-102-75-0 (15 test runs, $k = 50$) 36

4.5 Results for uncertain travel time. All values result from out-of-sample evaluations (in-sample size 50, out-of-sample size 1008). HN values are given relative to EEV values in percent, WS values are relative to HN values in percent. Below the OOSS expected values, (relative) CVaR90 values of the resulting distributions are given in parentheses. 41

4.6 Results for uncertain demand. All values result from out-of-sample evaluations (in-sample size 50, out-of-sample size 1000). HN values are given relative to EEV values in percent, WS values are relative to HN values in percent. Below the OOSS expected values, (relative) CVaR90 values of the resulting distributions are given in parentheses. 42

4.7 Results for simultaneous uncertainty of demands and travel times. All values result from out-of-sample evaluations (in-sample size 50, out-of-sample size 1000). HN values are given relative to EEV values in percent, WS values are relative to HN values in percent. Below the OOSS expected values, (relative) CVaR90 values of the resulting distributions are given in parentheses. 43

4.8 Uncertain travel time: Run times in seconds 44

4.9 Uncertain demand: Run times in seconds 45

4.10 Uncertain travel time/demand: Run times in seconds 46

5.1 Parameters for the GA 62

5.2 Comparison of routes computed considering weather to ones computed ignoring the weather regarding length in nautical miles duration in days and fuel consumption in tons, averages over five runs. 65

5.3 Comp. results for stochastic optimization with replanning along the route under perfect information (“PI”), stochastic optimization with replanning (“Stochastic”), stochastic optimization with no replanning (“Stoch. NR”), and without weather routing (“No WO”). Percentage gaps relate to perfect information and the standard deviation is given for the fuel. 70

6.1 SKP Results 82
6.2 RCPSP Results 86
6.3 SSPP Results 88

1 Introduction

In 2019, the world seaborne trade volume reached 11.08 billion tons, marking the maximum ever recorded (UNCTAD, 2020). The number of containers handled in ports worldwide reached 811.2 million TEUs[1] in 2019, which is an increase of 85% compared to 2000. For 2020, UNCTAD (2020) predicts a decline of international maritime trade by 4.1% with a subsequent recovery of 4.8% in 2021. Therefore, the trend of rising maritime trade volumes, excluding only events such as the world financial crisis 2008 and the global pandemic 2020, is expected to continue. To transport this huge amount of goods, the world fleet reached nearly 53,000 vessels summed over all vessel types in 2020, growing by 2.5% compared to 2019.

The growth of the maritime transportation shows the importance of the world seaborne trade. The transportation of goods with vessels has a much higher cost efficiency compared to other means of transport, making it a vital element of modern, cost sensitive delivery processes. The usage of standardized containers simplifies the handling of cargo in ports and on vessels and the combination with the huge capacity of container vessels makes the transportation of goods by vessels very efficient. Moreover, it is possible to transport nearly every commodity in containers nowadays. Overall, it is worthwhile to further improve the processes in the area of maritime transport due to the high trade volumes and the resulting savings potential.

There is already a significant amount of research in the domain of maritime transportation, summed up by for example Christiansen et al (2013) and more specifically for liner shipping by Brouer et al (2017) and Dulebenets et al (2019). A growing demand for the transport of goods with vessels makes the resulting problems for liner shipping carriers increasingly larger and therefore more difficult to solve. One factor that makes these problems even more complicated is the prevailing uncertainty of input data for the problems. These uncertainties include, for example, supplies, demands, sailing times of vessels, port times and fuel prices. Ignoring uncertainty and using deterministic data can lead to solutions causing extra costs or more serious consequences such as loss of containers or vessels. Therefore, it is necessary to include uncertainty into the optimization process to obtain realistic solutions. This dissertation presents the advantages of stochastic compared to deterministic approaches. Solutions of stochastic optimization are better adapted to reality than those from deterministic optimization. This makes approaches with uncertain input parameters also interesting for industry, where these approaches are rarely used so far. This leads to the following core question of this dissertation:

---

[1] twenty foot equivalent unit, unit of measure of a container

> *Does stochastic optimization provide solutions for maritime logistics problems that, when realized, result in higher expected profits than solutions found through deterministic optimization techniques?*

To answer this question, this dissertation focuses on the extension of selected existing maritime problems with stochastic components, on solution techniques for stochastic problems and on demonstrating advantages of stochastic optimization. This thesis applies stochastic optimization to specific problems, while also deriving general insights. In particular, the Liner Shipping Fleet Repositioning Problem (LSFRP) is considered, which has not been explored at all with stochastic influences. This problem considers the movement of vessels between services to adapt to changing conditions in the liner shipping network. Moreover, the weather routing problem is used to show how to deal with uncertain data in another problem. The importance of including uncertain data in decision making is pointed out and it is shown how to deal with data that is not certain until a particular point in time. The goals of this dissertation are:

1. Modeling of uncertain factors for the LSFRP.
2. Deriving of scenarios for the stochastic LSFRP from existing data.
3. Providing exact solutions for the LSFRP under the consideration of different uncertain input factors.
4. Development of a heuristic approach for the weather routing of vessels with stochastic weather data.

Furthermore, looking at practical problems and difficulties related to complexity and size of these problems arises the following secondary question:

> *Can a framework be developed that is capable of handling the computational complexity of realistic problems with an arbitrary number of decision stages and complex recourse decisions?*

This question leads to an additional goal to be achieved within the scope of this dissertation:

5. Introduction of a new method for stochastic optimization combining optimization and machine learning.

The remainder of this thesis is organized as follows. Chapter 2 introduces background information and fundamentals for this thesis. It gives an overview of maritime logistics problems, stochastic influences on these problems, about stochastic optimization in general and solution methods for stochastic maritime logistics problems applied within the reported doctoral project. In Chapter 3, the three research papers that are part of this dissertation are summarized. The first paper is included by Chapter 4 and is about a stochastic extension of the Liner Shipping Fleet repositioning Problem incorporating uncertain container demands and travel times. The paper in Chapter 5 presents a genetic algorithm for finding weather dependent routes for vessels. Chapter 6 covers a paper introducing a new method for stochastic optimization combining optimization and machine learning. Finally, Chapter 7 concludes this thesis and gives an outlook for possible future work.

2 Background

There is a wide range of maritime logistics problems that involve different types of uncertainty. This chapter gives an overview of these problems, explains key terms and points out the variety of stochastic components influencing the problems. Furthermore, this chapter introduces the fundamentals of mathematical optimization models and the metaheuristic genetic algorithm, which are applied to maritime logistics problems in Chapters 4 and 5.

2.1 Maritime Logistics Problems

The literature divides maritime logistics problems into three groups: liner shipping, tramp shipping and industrial shipping, with liner shipping constituting the sector with the highest transported volumes. In the following, the characteristics of these three different modes of operation in maritime transport are outlined. UNCTAD (2020) distinguishes the international maritime trade into tanker trade, main bulk and other dry cargo, which includes containerized trade.

2.1.1 Liner Shipping

In liner shipping, vessels have fixed schedules with cyclic routes, which are called services. Services are scheduled to visit ports in a fixed order in weekly or biweekly intervals at the same day to periodically provide customers with capacity to transport containers. A visit of a port is also referred to as a call, which has always the same arrival and departure times. To ensure regular visits, a service requires the number vessels equal to the number of weeks needed to complete it. Slots determine the time schedules for the calls of one vessel. Consequently, there are as many slots as vessels on one service. In summary, the service determines the day of the week and the time of a visit, while the slot determines the exact date of the visit.

The invention of the standard shipping container in 1956 created the basis for liner shipping. As Levinson (2016) states, containers are the center of a highly automated system that allows the transport of goods between two locations at very low cost and with great simplicity. The sizes of containers are given in twenty-foot equivalent units (TEUs) or the forty-foot equivalent units (FFEs), which refer to the length of the containers. Apart from special containers, the main containers used are dry and reefer containers. These types refer to standard containers that do not require special treatment and containers with refrigerated goods that require power supply, respectively. Containers can either be moved directly from their origin to their destination port or they are transshipped during their transport. Transshipment is the process of moving containers between services by unloading them from a vessel from one service and later loading them onto a vessel from another service. This allows carriers to offer delivery between ports that are not on the same service and is especially important to serve smaller ports that are only on a few or one service. Furthermore, carriers often use some ports as hubs in their hub-and-spoke networks for most of the transshipments. The smaller ports are then served by feeder services, while the hubs are located on geographically far-reaching services.

2.1.2 Tramp Shipping

In contrast to liner shipping, tramp shipping does not have fixed schedules and vessels sail where cargo is available to be transported. Thus, the vessels only sail on demand and not on a regular basis. This allows cargo to be transported at shorter notice, as there is no need to wait for the next fixed call of a port. However, there is often an amount of cargo for which the transport is contracted. The main goal of tramp shipping is the maximization of profit for delivered cargo reduced by operational costs.

2.1.3 Industrial Shipping

Industrial shipping falls between the other two types of shipping. Both the vessels and the cargo are owned by the same operator, who decides on the deployment of vessels adjusted to the supply of goods and aims to minimize the costs for the delivery. According to Christiansen et al (2013), both in tramp and industrial shipping mainly dry or liquid bulk cargoes are transported.

2.1.4 Uncertainties in Maritime Logistics Problems

All problems associated with liner, tramp and industrial shipping can be categorized into strategic, tactical and operational planning levels, respectively. Problems from all three levels of planning are subject to uncertainties regarding different input factors. The most important problems including uncertainties are listed in Figure 2.1.

Strategic planning has the most extensive consequences and includes decisions that are retained for many years. Maritime fleet renewal is one example of a strategic problem. This problem determines the change of size and composition of the fleet over a given time horizon. Therefore, the decisions to be taken in these problems primarily include the number of newly built, bought, sold and scraped ships. Another strategic problem is the Network Design Problem. For the case of liner shipping, Christiansen et al (2019) define that this problem is about designing a set of weekly services and assigning vessels with certain capacities to these services to flow available demands through the network complying to time constraints. Thus, the liner shipping network is used to ensure the transport of containers. Christiansen et al (2019) also present

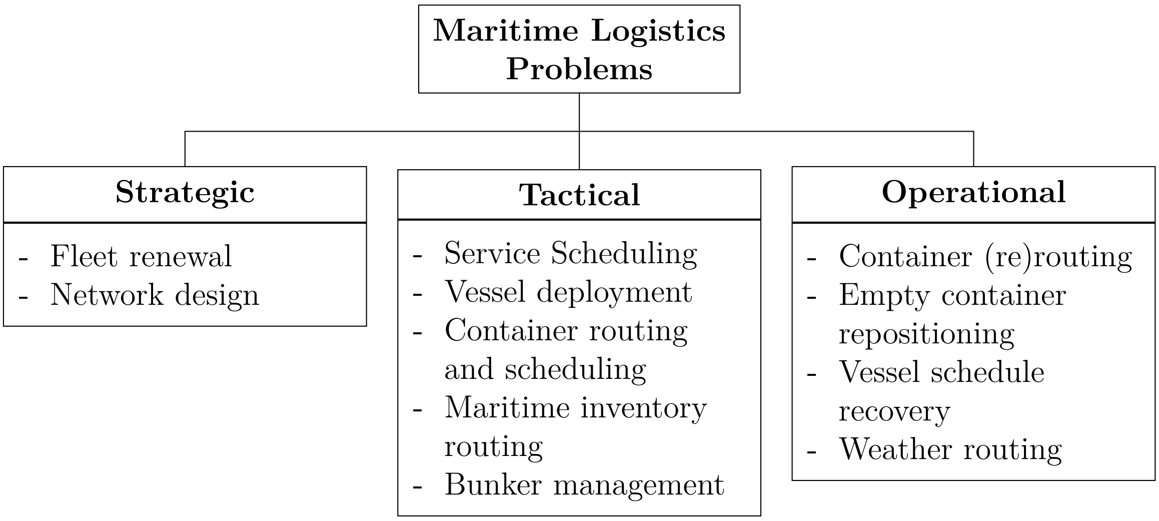


Figure 2.1: Maritime logistics problems including uncertainties

a comprehensive overview of existing models and solution approaches for the liner shipping network design problem.

Based on the networks designed at the strategic planning level, services are scheduled, vessels are deployed, the flow of containers is determined, and decisions are made on where and how much to bunker at the tactical planning level. These processes are optimized within the service scheduling problem, the vessel deployment problem, container routing and scheduling problems and the bunker management problem. Furthermore, the maritime inventory routing problem (MIRP) determines the routes for vessels to keep the inventory levels of ports within given limits. Brouer et al (2017) state that an integration of these processes into the network design would potentially result in higher quality networks but would lead to problems that are computationally not practical. Tactical planning is more short-term than strategic planning and includes decisions for one or several years.

Operational planning is done on a daily basis and includes decisions about actions regarding containers, vessel rescheduling and routing of vessels in response to changed conditions. As Meng et al (2014) state, containers are assumed to be homogeneous for a liner shipping company in container cargo booking. Therefore, just the number of containers to be transported between ports has to be determined. A related problem is the problem of empty container repositioning. The vessel schedule recovery problem (VSRP) is another operational problem, where actions are taken to compensate disruptions delaying vessels. The Liner Shipping Fleet Repositioning Problem (LSFRP), which is considered in more detail in Chapter 4, is closest related to the VSRP, but is a tactical problem. It deals with the process of moving vessels between services to adapt to changes in the network. The area of weather routing focuses only on the effects of weather on the routes of vessels based on bunker consumption and, more importantly, dangers for crew, ship and cargo.

The problems from all three planning levels are subject to several different uncertainties. The uncertainty that affects most of the problems regards demand. It has an effect on fleet renewal (e.g., Bakkehaug et al (2014); Arslan and Papageorgiou (2017)), scheduling of vessels (e.g., Soroush and Al-Yakoob (2018); Kisialiou et al (2019)), vessel deployment (e.g., Meng and Wang (2010, 2012); Wang et al (2012); Ng (2015)) and empty container repositioning (e.g., Tsang and Mak (2015); Wong et al (2015)). Zhen et al (2019) include uncertain container weight directly related to uncertain demand in their fleet deployment problem and Rodrigues et al (2019) state that uncertain production rates and uncertain demand are among others sources of uncertainty considered in the MIRP.

Demand uncertainty and related uncertainties are important to consider when solving maritime logistics problems, as they determine the only source of revenue for carriers and are therefore key factors for the profitability of shipments. These uncertainties are also major factors when deciding on the number and size of vessels to acquire. Thus, to accurately assess these factors, it is valuable to consider multiple future scenarios representing the uncertainty. For example, according to Bakkehaug et al (2014), taking into account only the average deterministic scenario can lead to decisions that are too specific for the expected future and therefore may lead to bad performance when realized. As uncertain demand has a large impact on a major part of maritime logistics problems, this dissertation deals with this uncertainty in more detail and extends the existing deterministic LSFRP with uncertain container demands to investigate the effect of this uncertainty on solutions.

Further frequently considered uncertainties in maritime logistics problems regard port and sailing times of vessels. For service scheduling, these uncertainties are for example included by Qi and Song (2012), Wang and Meng (2012a) and Wang and Meng (2012b) and for the MIRP by Agra et al (2015). However, there are also approaches for the MIRP that only consider the port time (e.g., dos Santos Diz et al (2019)) or only the sailing time (e.g., Christiansen and Nygreen (2005); Agra et al (2018)). Aydin et al (2017) include uncertain service times at ports into the bunkering problem and integrate speed optimization allowing delays to be compensated by higher speed.

Handling uncertain port and sailing times is challenging, as uncertainties can lead to cumulative delays that have to be compensated by higher buffer times at ports or increasing vessel speed between ports. Furthermore, the timeliness of vessels is an important aspect to keep customer satisfaction high and to avoid high costs charged for delays at ports. Since temporal uncertainties also have a large impact on solutions to many maritime logistics problems, uncertain sailing times are also included in the stochastic LSFRP presented in Chapter 4. Thus, this stochastic model integrates the combination of uncertain demands and travel times, which is close to a realistic application, where both uncertainties occur simultaneously.

Directly related to temporal uncertainties is the uncertainty of weather, as this can lead to delays, among other effects. Delays on connections and in ports caused by severe weather are for example included in the rerouting of containers (e.g., Xing and Zhong (2017)) and network design (e.g., Yang and Chen (2017)). Li et al (2016) mention uncertain weather as one factor for disruptions in the VSRP, which often occur considering the high number of vessels used globally. In bunker management, uncertain weather influences the bunker consumption and speed of vessels as for example presented by Du et al (2015).

Incorporating uncertain weather can result in less costly routes due to reduced bunker consumption and, more importantly, less dangerous routes for crew, ship and cargo. It allows generating more robust schedules for vessels and identifying meaningful buffer times that later lead to less additional costs caused by delays, which are a major challenge associated with uncertain port and sailing times caused by uncertain weather. Recent events reported, for example, by Hand (2020, 2021) and Wingrove (2021) show the importance of weather routing and the inclusion of uncertain weather data into the optimization to lower the risk for vessels. All these factors motivate a closer look at uncertain weather and its influence on vessel routes. This field is analyzed in more detail in Chapter 5, which also presents an approach for the generation of routes under the influence of uncertain weather.

Less frequently integrated and more special uncertainties include bunker prices (e.g., Meng et al (2015b); Wang et al (2018b); Gu et al (2019)) and uncertain bunker consumption (e.g., Sheng et al (2014); Ghosh et al (2015); De et al (2021)) for the bunker management problem. Integrating uncertainty of these factors can lead to more economical and therefore better solutions, since bunker costs make up the largest part of the operating costs of vessels. Other less common uncertainties primarily affecting the fleet renewal regard the prices for building, selling, buying, and chartering ships and operational costs. As these costs mainly determine the profit for carriers, it is of high interest to use stochastic data to get solutions that perform well in many future scenarios here as well.

2.2 Solution Techniques

To solve stochastic maritime logistics problems, exact algorithms and heuristics are used. According to Gendreau et al (2010), exact algorithms are able to guarantee that a solution found is optimal, while heuristic approaches lack this ability. However, exact solution methods often have problems finding solutions for big instances due to high computational complexity and long runtimes. Heuristics obtain at least good solutions for these problems. This makes heuristics especially interesting for real-world instances and stochastic optimization problems as these often have a high level of complexity.

Subsection 2.2.1 gives an overview of mathematical optimization models, which are used to mathematically represent optimization problems. These models can be solved with exact solution methods, which directly use the mathematical representation to find an optimal solution. These approaches have to search the whole or at least the part of the solution space that is guaranteed to contain the optimal solution, which can result in a long runtime. Heuristic approaches can solve the same problems as these exact solution methods, but they do not employ the same mathematical structure of the problems.

Metaheuristics are good approaches to find solutions for problems that are too big or complex to be solved to optimality in a reasonable amount of time. According to Gendreau et al (2010), metaheuristics are methods combining approaches for local improvements with a higher-level strategy to be able to overcome local optima. Metaheuristics are well suited to solve problems with stochastic elements as these make problems in general more complex. Stochastic optimization problems are in most cases difficult to solve because of the wide range of possible future scenarios, the huge amount of decisions and the high number of stages in practical problems. Subsection 2.2.2 presents the metaheuristic genetic algorithm, which is applied within this dissertation for the routing of vessels incorporating uncertain weather.

Metaheuristics use either a single solution or a set of promising solutions. When using a single solution, this solution is repeatedly changed during the runtime of the algorithm, while a set of solutions allows to continue the search for solutions at multiple positions in the solution space. Often the structure of the problem determines which approach is well suited to search the solution space and find good solutions. There are also approaches combining exact solution methods with metaheuristics to get optimal solutions in less time than with purely exact methods or to obtain better heuristic solutions. An overview of this field of algorithms is for example given by Puchinger and Raidl (2005).

2.2.1 Mathematical Optimization Models

Modeling a problem as a mathematical optimization model and solving it with a standard solver software is one way to get exact solutions for optimization problems. Mathematical optimization models use decision variables, constraints and an objective function to represent the important aspects of reality in a formal way like pointed out by Calafiore and El Ghaoui (2014). Decision variables are the determining values for the definition of a solution of an optimization problem. They represent all decisions that are important and can be taken. One combination of values for all decision variables defines a solution. In production planning, these would be, for example, the number of different goods produced; in a routing problem decision variables determine which routes to use. Constraints are used to define meaningful, also called feasible, solutions, by restricting the decision variables with equalities and inequalities. Therefore, the constraints limit the solution space. The objective function specifies how to evaluate solutions in order to assign a value to them. The objective function either minimizes or maximizes this value, depending on the purpose of the underlying problem. In the objective function, a part or all of the decision variables are multiplied by factors expressing their respective values as defined by the problem. The goal of solving a mathematical optimization model consisting of these three parts is to find values for the decision variables leading to the optimal value of the objective function. According to Calafiore and El Ghaoui (2014), there might be a set of solutions leading to the same, optimal objective value. This set can also be empty, meaning there are no feasible solutions and therefore no optimal one for a problem. Calafiore and El Ghaoui (2014) also state that types of problems exist where it is only of interest to find any feasible solution and an objective function is not provided. Birge and Louveaux (2011) provide the deterministic linear programming model

$$\begin{array}{lllll} \text{(DLPM)} & \min z = & c^T x & & \\ & \text{s.t.} & Ax & = & b \\ & & x & \geq & 0 \end{array}$$

in matrix notation. The vectors c and x have n entries, where n is the number of decision variables of the problem. The vector c represents the weights of the decision variables with respect to the problem, for example, costs. The objective function in the DLPM is minimized, but it is possible to transform a minimization problem into a maximization problem and vice versa but simply multiplying the objective function by -1. The number of constraints leads to m entries for b, limiting the feasible assignment of values to the variables. Matrix A contains the coefficients for the decision variables that specify the weights of the variables in the constraints and represent, for example, resource consumption.

The DLPM is an example for a linear programming model (LP) as the objective and all constraints are linear expression. As Calafiore and El Ghaoui (2014) state, the term ‘programming’ does not indicate computer code but is mainly used for historical reasons. Other types of models include integer programming models (IP), where the decision variables are only allowed to take integer values, and mixed integer programming models (MIP), containing a combination of integer and real valued decision variables. According to Conforti et al (2014), the inclusion of integer variables makes problems considerably more difficult to solve compared to LPs, as it makes problems NP-hard in general. Specific cases may be solvable in polynomial time when problems have a special structure.

Approaches that can be used to solve mathematical optimization models are for example the simplex method, interior-point methods, dynamic programming and branch and bound, which is used to solve IPs. State of the art solvers use these basic techniques to optimally solve problems. Vanderbei et al (2015) give an overview of exact solution approaches for mathematical optimization models and include all these examples.

Stochastic Optimization Models

The DLPM contains only deterministic parameters, which means c, A, b contain only deterministic values that are known when the decisions have to be taken. In many real-world problems, however, there is also uncertain information that is only available at a later point in time. This means that there are two groups of decisions: Decisions that have to be made directly and decisions that are made later, when the previously uncertain information is available. Since a model developed from such problems has two stages of decisions, it is called a two-stage optimization model. The decisions and associated variables are divided into first stage decisions and second stage decisions that depend on the first stage decisions, giving the corresponding variables also the name recourse variables as they react to the values of the first stage variables.

Birge and Louveaux (2011) present the following two-stage stochastic linear programming model with fixed recourse.

$$\begin{array}{llrcrcl}
\text{(SLPM)} & \min z = & c^T x & + & \mathrm{E}_\xi[\min q(\omega)^T y(\omega)] & & \\
& \text{s.t.} & Ax & & & = & b \\
& & T(\omega)x & + & Wy(\omega) & = & h(\omega) \\
& & & & x \geq 0, y(\omega) \geq 0 & &
\end{array}$$

This model is said to have fixed recourse, because W is not uncertain and does therefore not change for different future scenarios. It extends the DLPM with a second stage of decisions and associated parameters. The first stage decisions are represented by the vector x, the second stage decisions by vector y. The values for y can only be determined after the values for x are fixed and react to the stochastic data of the second stage $q(\omega)$, $h(\omega)$, $T(\omega)$. The random events that occur in the second stage are denoted by $\omega \in \Omega$. This means that for each event ω there are values for q, h, T depending on it. Like Birge and Louveaux (2011) state, combining all stochastic components leads to the vector $\xi^T(\omega)$ with all components influenced by the a single random event ω. When solving the model, a finite set of scenarios is used, each corresponding to a realization of ξ. The part in the objective function that is new in comparison to the DLPM minimizes the expected objective values of all subproblems, each referring to a scenario and the decisions made for it. The new constraints represent the limitations of the subproblems in the second stage.

The SLPM can be transformed to the deterministic equivalent

$$\begin{array}{llrclcll}
\text{(DE)} & \min z = & c^T x & + & \sum_{k=1}^{K} p_k q_k^T y_k & & & \\
& \text{s.t.} & Ax & & & = & b & \\
& & T_k x & + & W y_k & = & h_k, & k = 1, ..., K \\
& & & & x \geq 0, y_k \geq 0, & & & k = 1, ..., K
\end{array}$$

in the extensive form. The number of scenarios ω is specified by K and p_k are their respective probabilities of occurrence. In this form, it is also more apparent that there are separate variables and constraints for each scenario k. The individual objective values related to the scenarios are weighted by p_k in the complete objective function to obtain the expected value across all scenarios.

As this form only contains deterministic parameters, this model can be solved by state-of-the-art solvers, even if they do not support stochastic optimization. Furthermore, this form can be used to apply the L-shaped method, introduced by Van Slyke and Wets (1969) exploiting its block structure.

Two-stage optimization models can be extended to multi-stage optimization models to represent problems with more than one period where new decisions have to be made. The decisions in later stages then depend on all decisions from previous stages making such problems more complex. Practical problems include high numbers of decisions and stages combined with many possible future scenarios making the solution process challenging.

2.2.2 Genetic Algorithm

Like previously pointed out, stochastic optimization problems can easily become very complex and thus unsolvable with exact solution techniques. Then it is useful to apply metaheuristics to search for high-quality solutions. The concept of the metaheuristic genetic algorithm (GA) is introduced by Holland et al (1992). The approach works analogous to the process of evolving individuals known from nature. The basic concept of this metaheuristic is to encode the relevant features of solutions in a representation called chromosomes and to derive new solutions from the existing ones just as it is known from inheritance in nature. The approach uses a set of individuals called population analogous to nature.

The pseudocode for a general version of a genetic algorithm is given in Algorithm 1. In the first step of this algorithm, an initial population of individuals is generated. This requires an algorithm generating feasible solutions that differ from each other. The solutions for the initial population are nearly always randomly generated and are the basis for the generation of new solutions in the following steps.

Afterwards, an iterative process is started to generate new generations of solutions resulting in a high quality solution when the genetic algorithm terminates. Each iteration of generating a new generation consists of a crossover and mutation part, which are repeatedly performed, see lines 3 to 12 of Algorithm 1. In the first part, two individuals are selected and recombined to create a new solution corresponding to the crossover of individuals known from genetics, see lines 4 to 7. This means that a part of one solution is replaced by a part of the other. It is also possible to replace several parts of a solution and not only one contiguous part. It is possible to use a number of different crossover operators combining solutions in different ways, for example differing in the number of exchanged parts.

In the second part of one iteration, mutations occur with a given probability, see lines 8 to 10. These mutations represent changes within solutions that are not part of the two selected and combined parent individuals. This allows to preserve a certain level of diversity within population and to search parts of the solution space that cannot be searched by just combining existing solutions. Gendreau et al (2010) point out that mutations might be much more relevant than the crossover depending on the problem. It is also advisable to not only make random changes to solutions, but to also use domain and problem specific knowledge to change solutions, which makes mutation operators powerful instruments for finding good solutions.

The representation of the solutions has a high impact on possible crossover and mutation operators. If a solution can be encoded as a string of binary values, the crossover and mutation of individuals is rather simple. It might be more difficult to find operators in cases with other representations, but these make it also possible to apply problem-specific operators potentially leading to better solutions. Regardless of the representation, however, it is important to guarantee the feasibility of solutions, either by operators generating only feasible solutions or by rejecting infeasible solutions after the application of an operator.

After the crossover and mutation steps are completed, the fitness of the new solutions is evaluated meaning the calculation of their objective values. The process of crossovers and mutations is repeated until a sufficiently high number of new individuals is generated. The last part of one iteration for the creation of a new population within the algorithm is the selection of the individuals making up the new population for the next iteration, see line 11. There are many different strategies for the selection of individuals for the new population. The new population may only consist of new, combined solutions, but it is also possible to always preserve the best solutions across iterations. The genetic algorithm stops after a stopping criterion is met, otherwise it could run forever due to its stochastic search process. Examples for stopping criteria listed by Gendreau et al (2010) include reaching a limit on the number of fitness evaluations or time. Another option is to stop when the diversity of the population falls below a given threshold. This diversity can refer to the similarity of solutions considering at the elements of their representation, or simply to the fitness values of the individuals.

GAs can be applied to stochastic optimization problems by setting the fitness of individuals to the expected value of the objective values for the considered future scenarios. According to Bianchi et al (2009), GAs and other evolutionary algorithms are used for many problems with uncertain input parameters. These problems include, for example, labor scheduling, vehicle routing, warehouse scheduling and job-shop scheduling. Jin and Branke (2005) highlight evolutionary algorithms as well suited for problems with changing, stochastic parameters because they are based on evolution from nature, where individuals must also adapt to change. They also point out that altering old individuals and adapting them to information that becomes available later in the solution process improves the speed and the solution quality. Moreover, Bianchi

Algorithm 1 General structure of a genetic algorithm from Gendreau et al (2010)

```
Choose an initial population of chromosomes
while termination condition not satisfied do
    repeat
        if crossover condition satisfied then
            select parent chromosomes
            choose crossover parameters
            perform crossover
        if mutation condition satisfied then
            choose mutation points
            perform mutation
        evaluate fitness of offspring
    until sufficient offspring created
    select new population
```

et al (2009) note that evolutionary algorithms often outperform other metaheuristics when applied to problems with uncertainties.

3 Overview of Research Papers

This chapter gives an overview of all papers that are part of this dissertation. The papers address approaches for two maritime logistics problems and stochastic problems in general.

3.1 Paper 1: The Stochastic Liner Shipping Fleet Repositioning Problem with Uncertain Container Demands and Travel Times

This paper is authored by Stefan Kuhlemann, Jana Ksciuk, Kevin Tierney, Achim Koberstein. It was submitted to the EURO Journal on Transportation and Logistics in 2020.

The paper extends the existing Liner Shipping Fleet Repositioning Problem presented by Tierney (2015) with uncertain container demands and travel times of vessels. It introduces a two-stage stochastic optimization model determining the routes of vessels on the first stage and both the number transported containers and the recovery of delays on the second stage. The model also includes the risk measure conditional value-at-risk to analyze solutions with risk-neutral and risk-averse objectives. Furthermore, this paper presents a procedure to generate a representative scenario set out of a huge number of scenarios based on real world data. The computational results point out the advantages of using a stochastic optimization model regarding higher objective values and lower delays. The evaluation of the results includes the impact of uncertain demands and travel times separately and the impact of both combined.

3.2 Paper 2: A Genetic Algorithm for Finding Realistic Sea Routes Considering the Weather

This paper is written by Stefan Kuhlemann and Kevin Tierney. It was published in the Journal of Heuristics in 2020.

The paper introduces a heuristic approach for the maritime problem of weather routing including uncertain weather data. The genetic algorithm presented in this paper includes stochastic weather forecasts into the generation of routes for vessels. The approach supports freely chosen waypoints for the route, variable speed of the vessel and ensures the avoidance of sharp changes of the route. The paper presents an approach for the generation of initial routes necessary to start the genetic algorithm, problem specific crossover and mutation operators to change existing solutions within the genetic algorithm and a method to deal with stochastic weather. The results show a comparison of solutions generated using no weather data, perfect information about the weather, stochastic weather data and the possibility to recalculate the remaining route when new information about the weather becomes available. The findings show the impact of stochastic weather data on the routes of vessels and the importance of the inclusion of weather in the planning.

3.3 Paper 3: Exploiting Counterfactuals for Scalable Stochastic Optimization

The authors of this paper are Stefan Kuhlemann, Meinolf Sellmann and Kevin Tierney. It was published as a conference paper of the "International Conference on Principles and Practice of Constraint Programming" in 2019.

This paper introduces a new heuristic method for stochastic optimization. The proposed framework combines optimization and machine learning to overcome main drawbacks of existing stochastic optimization approaches. These are modeling complexity, the difficulty to generate meaningful scenarios and computational limitations caused by scaling problems to real-world size. The idea is to train a model with historic data and features of historic solutions that is then able to find the most promising solution among a number of near optimal solutions by pairwise comparisons. The use of existing solutions also allows to capture instance-dependent characteristics. The application of this method is shown for the stochastic versions of the Knapsack Problem, the Shortest Path Problem and the Resource Constrained Project Scheduling Problem to provide a proof of concept.

4 The Stochastic Liner Shipping Fleet Repositioning Problem with Uncertain Container Demands and Travel Times

Abstract

Liner shipping repositioning is the costly process of moving container ships between services in a liner shipping network to adjust the network to the changing demands of customers. Existing deterministic models for the liner shipping fleet repositioning problem (LSFRP) ignore the inherent uncertainty present in the input parameters. Assuming these parameters are deterministic could lead to extra costs when plans computed by the deterministic model are realized. We introduce an optimization model for the stochastic LSFRP that handles uncertainty regarding container demands and ship travel times. We extend existing LSFRP instances with uncertain parameters and use this new dataset to evaluate our model. We demonstrate the influence of uncertain demand and travel times on the resulting repositioning plans. Furthermore, we show that stochastic optimization generates solutions yielding up to ten times higher expected values and more robust solutions, measured against the CVaR90 objective, for decision-makers in the liner shipping industry compared to the application of deterministic optimization in the literature.

Keywords: Liner Shipping Fleet Repositioning Problem, Stochastic Optimization Model, Uncertain Demands, Uncertain Travel Times

4.1 Introduction

In 2018, the world seaborne trade volume exceeded 11 billion tons for the first time (United Nations Conference on Trade and Development, 2019), with an estimated 793.26 million twenty-foot equivalent units (TEUs)[1] of containerized trade handled in container ports worldwide. The number of containers shipped continues to rise each year, with over 80% more containers shipped in 2018 than in 2000. This trend is expected to continue, as cargo that was traditionally shipped in bulk, such as copper or bananas, are increasingly carried in containers (Rodrigue and Notteboom, 2015).

As the demand for transporting containers has grown, so has the difficulty of planning their transport. Liner shipping problems are particularly challenging due to the scale of the problems in terms of ships and containers, as well as due to their structure,

---

[1] A TEU represents a single twenty foot standardized shipping container.

which involves cyclical routes with no clear start or end port. The research field has traditionally identified several planning problems for the liner shipping industry (Meng et al, 2014). At the strategic level, liner carriers design their networks and their fleet, at the tactical level the fleet is deployed across the network, and at the operational level, containers are routed through the network. Since container ships are in constant operation, networks cannot be redesigned and rebuilt from scratch. In reality, the planning process is one of frequent improvements to the network, in which ships are redeployed and *repositioned* between services (Tierney et al, 2014).

The liner shipping fleet repositioning problem (LSFRP) involves minimizing the cost of moving a set of ships from their starting services to a new service in the network, while maximizing the revenue earned by transporting containers and moving empty containers on the repositioning ships (Tierney, 2015). Liner carriers solve repositioning problems whenever they adapt their networks, generally a few months in advance of making changes. The LSFRP considers a subset of the overall network spanning a couple of months and focuses on a single repositioning. This means the LSFRP avoids the planning complexity of flowing containers through the entire network. Thus, the goal of the LSFRP is to provide decision support to planners both to carry out redeployments, as well as to price what-if scenarios.

Many of the parameters of the LSFRP are subject to uncertainty. Current approaches for solving the LSFRP assume deterministic parameters regarding travel times or available container demands. However, it should not be surprising that demands fluctuate from week to week, and ship travel times are strongly influenced by the weather, tides, mechanical breakdowns, delays in port, etc. Including uncertainty in the LSFRP is critical for providing planners with solutions that work in practice.

Several techniques exist to handle uncertainty, such as robust optimization (Ben-Tal et al, 2009), chance constraints (Charnes and Cooper, 1959), and multi-stage stochastic programming. In stochastic programming, uncertain data is represented by random variables with known probability distributions. To ensure computational tractability, usually only discrete distributions are considered, consisting of a finite set of discrete scenarios with given probabilities of occurrence. In two-stage stochastic programming, which we use in this work, the uncertainty of a problem is revealed in stages. In the first stage, decisions are made not knowing what will happen in the next stage. These are called the *here-and-now* decisions. Decisions at the second stage are made with respect to the fixed first stage decisions once the uncertainty present at the second stage is revealed, allowing a plan to be modified to better handle uncertainty (Birge and Louveaux, 2011; Kall et al, 1994).

In this paper, we introduce a new two-stage stochastic model including uncertain container demands and ship travel times. In contrast to previous approaches, the number of containers transported and the speed of the vessel are scenario-dependent in our model and only the routing of the vessels is treated as a here-and-now decision, i.e., it is fixed in the first stage before the uncertainties become known to the decision maker. In the second stage, we allow ships to adjust the containers they carry or speed up to catch up after delays. Furthermore, we allow delay to propagate along vessel paths in the second stage. Finally, we generate realistic scenarios for demand and delay data to evaluate the model and to examine the influence of stochastic data on the repositioning problem.

The remainder of this paper is organized as follows. Section 4.2 presents the state of the art of the LSFRP and previous work on solving it and Section 4.3 describes our problem. Afterwards, we introduce the new stochastic optimization model based on the existing deterministic LSFRP in Section 4.4. Section 4.5 presents results for our instances. In Section 4.6, we conclude our work and give an outlook for future work.

4.2 Related Work

We provide an overview of related work in the literature, comparing and contrasting our contribution. First, we summarize work related to the LSFRP, then we discuss maritime optimization problems that consider types of uncertainty that are similar to what we consider in this work.

4.2.1 LSFRP

Tierney (2015) provides an overview of the research in the domain of the LSFRP, along with several different models of the LSFRP, one using an arc flow formulation, and one flowing over the nodes. Large models are solved with a simulated annealing approach. Pearce et al (2016) implement a column generation approach that generates vessel paths in its subproblem, and further use lazy constraints to reduce the model size. Tierney et al (2017) introduce a multi-objective approach considering sailing costs, cargo profit and equipment profit and extend a simulated annealing heuristic to provide solutions from a Pareto front. Finally, Wetzel and Tierney (2020) integrate the LSFRP with a fleet deployment problem (see, e.g., Perakis and Jaramillo (1991); Powell and Perakis (1997)) to allow repositioning to be considered when deciding which ships should deploy to which services.

More broadly, Brouer et al (2017), and Dulebenets et al (2019) provide overviews of optimization problems in the maritime domain, and we refer to these works for a review of the field. The most similar class of problem to the LSFRP is vessel schedule recovery, which is used to help ships return to their regular operations after a delay occurs (Brouer et al, 2013a; Li et al, 2015, 2016). While the vessel schedule recovery problem (VSRP) is also generally solved on a time space graph, the VSRP is an operational problem, whereas the LSFRP is tactical. Thus, despite similarities of the problems in terms of routing between services, the activities (picking up empty containers, etc.) available to the vessels are rather different.

Despite the liner shipping service structures in the LSFRP, it nonetheless bears some resemblance to tramp shipping problems. Tramp shipping is concerned with the transportation of bulk materials, and generally does not follow a fixed schedule the way liner shipping does (Christiansen et al, 2013). Tramp shipping problems maximize the profit for delivered cargo minus sailing costs and port fees, while lacking cyclical, fixed schedule structures (see, e.g., El Noshokaty (2018); Pache et al (2019)). Furthermore, demands in liner shipping problems have a fixed origin and destination, whereas tramp shipping problems can allow that vessels satisfy demands with any suitable cargo (see, e.g., Alvarez et al (2020)). Furthermore, the LSFRP does not allow split loads (see, e.g., Andersson et al (2011); Stålhane et al (2012)).

4.2.2 Maritime models with demand and/or sailing time uncertainty

We now survey the maritime literature for approaches dealing with demand and/or sailing time uncertainty on problems similar to ours. Table 4.1 gives an overview of related articles. The table groups references by the type of problem being solved, the type of uncertainty present, the cost/revenue components in the objective function, how the uncertainty is handled, and, finally, whether a two-staged model is used. We further highlight the most relevant works below.

We examine the approaches for the maritime inventory routing problem (MIRP) in more detail. The approaches of Agra et al (2015, 2016, 2018) use a two-stage stochastic optimization model, in which the routes of the vessels are fixed in the first stage, as is the case in our approach for the LSFRP. In the second stage, port visits are scheduled and the number of loaded/unloaded products specified on the first stage are adjusted to the scenarios. The uncertain sailing and port times primarily have an impact on inventory decisions and not on the handling of delays like in the LSFRP. Christiansen and Nygreen (2005) also only consider scheduling and routing for vessels and the management of inventory levels at ports. In contrast to our approach of handling delays with the inclusion of propagation, the approach of Zhang et al (2018) can only absorb delays by creating sufficiently large time windows for a visit at the destination port of an arc with a delay, by rerouting ships, or by purchasing missing demands on the spot market when it is impossible to meet time windows. In summary, while MIRPs have been considered with uncertainty, the LSFRP nonetheless poses unique challenges for stochastic modeling not present in the MIRP.

The literature on fleet planning models that include vessel routing only consider demand uncertainty. Meng et al (2012) and Meng et al (2015a) implement two-stage, stochastic optimization models, and the second stage handles demand the same way as we do in the LSFRP. However, these approaches do not include any temporal aspects connected to the sailing of vessels.

Several papers on routing and scheduling problems consider uncertain sailing times or demands, but not both in combination. Kepaptsoglou et al (2015) use chance constraints to limit travel times and also have no propagation of delays. Kisialiou et al (2019) allow visits (and associated demands) to be swapped between voyages of vessels. These binary second-stage decisions are possible as the authors solve the resulting optimization problem with a heuristic.

Problem type	**Reference**	**Uncertainty**	**Obj. components**	**Uncertainty handling**	**Two-stage?**
Maritime Inventory Routing	Christiansen and Nygreen (2005)	Sailing time	Transportation costs	Soft inventory constraints and associated artificial penalty costs	
	Agra et al (2015)	Port time, Sailing time	Sailing costs, port costs	Recourse: Schedule of loading and unloading operations Scenarios: Historical data and a given exponential distribution	✓
	Agra et al (2016)	Sailing time	Port costs, sailing costs, penalty for inventory deviation	Recourse: Schedule, inventory Scenarios: Travel times follow a three-parameter log-logistic probability distribution	✓
	Agra et al (2018)	Sailing time	Port costs, Sailing costs	Recourse: Schedule and stock levels Scenarios: Delays with discrete time steps	✓
	Zhang et al (2018)	Sailing time	Transportation costs, time window compliance	Recourse: Rerouting of vessels Scenarios: Random generation of disruptions Lagrangian heuristic	
Liner ship fleet planning	Meng et al (2012)	Demand	Profit for shipments	Recourse: Number of transported containers Scenarios: With different levels of standard deviation of demand	✓
	Meng et al (2015a)	Demand	Profit for shipments	Recourse: Number of transported containers Scenarios: Three container scenarios on each stage with demand increasing over time	✓ (multi stage)
Containership routing	Kepaptsoglou et al (2015)	Sailing time	Sailing costs	Chance constraints to limit travel time; no delay propagation Genetic algorithm solves for fixed confidence level	
Supply vessel routing and scheduling	Kisialiou et al (2019)	Demand	Expected number of vessels, fuel costs	Recourse: Relocation of visits, use charter vessels, hiring spot vessels Scenarios: Demand following various distributions Heuristic solves for a fixed reliability level	
Liner shipping single service design problem	Tierney et al (2019)	Arrival time	Vessel charter/deployment costs, sailing costs	Chance constraints for arrivals Travel times: log-logistic probability distribution functions for region-to-region pairs	
LSFRP	Our approach	Demand, Sailing time	Profit for shipments, port costs, sailing costs, CVaR	Recourse: Transported containers, change of speed of vessel, propagation of delays Scenarios: Demand based on real-world historical data; sailing time following three-parameter log-logistic probability distribution, weather dependent	✓

Table 4.1: Summary of papers working on LSFRP related problems with uncertain input parameters.

In the case of liner shipping single service design, Tierney et al (2019) focus only on travel time uncertainty, and handle it using chance constraints. The propagation of delays is implemented to evaluate the quality of solutions, but is not included in the models.

In summary, there is no approach combining both uncertain demands and sailing times as is necessary for solving the LSFRP. Furthermore, we also note that most models do not include the propagation of delays and the explicit modeling of recourse actions to avoid lateness.

4.3 Problem Description

We base our problem description on the notation and description in Tierney (2015). Liner shipping networks consist of many weekly or biweekly recurring, cyclical routes called *services*. Each service has one or more vessels sailing on it, with each vessel sailing in a particular week, or *slot*, of the service. For example, a service that has a total round trip time of three weeks requires three vessels so that every port called on the service can be visited on a weekly basis, and of course this notion can be easily extended to other service frequencies.

The LSFRP solves the problem of moving vessels from their current services to new ones to carry out changes in the network. Specifically, the LSFRP seeks to provide the vessels for a new service, called the *goal service*, that will start in the network. It does this by determining routes for vessels from their current services to the goal service. In this way, the network is updated in a cost minimal manner, while ensuring customer demands are transported even during the disruption to normal operations caused by repositioning. When a vessel reaches the goal service, this marks the end of its repositioning, and it is said to have *phased in* to a slot on the goal service. When a vessel leaves its current service, it is said to *phase out*, which marks the end of regular operations and the beginning of a repositioning.

The planning horizon of an LSFRP problem is defined by an earliest time at which the vessels under consideration are allowed to phase out and a latest time at which the goal service must begin. While small problems may only span a few weeks, large problems can require several months for all vessels to complete their repositionings. Since repositioning is performed to adjust the network and not (explicitly) to generate profits, the goal is to carry out the repositioning as cost-efficiently as possible. Therefore, certain cost-saving activities during repositioning are considered.

To earn revenue, either containers or equipment (empty containers) can be delivered. Container demands are defined by origin-destination (OD) pairs, whereas equipment originates at ports where it is in surplus and can be delivered to any port where there is a deficit. There are two types of containers and equipment: reefer (refrigerated) and dry (normal). The amount of containers available for each OD pair and container type fluctuates on a weekly basis. We consider the capacity of the vessel as defined in terms of reefer and dry containers. While this is not detailed enough to stow a vessel (Pacino and Jensen, 2012), it is sufficient to ensure at a tactical level that the capacity of the ship is respected. Furthermore, we assign OD pairs a maximum transit time, meaning that a demand can only be carried by a vessel if it is delivered within the given time limit.

Repositioning vessels can utilize services in the network that are neither the origin service for the vessel nor the goal service. This is called using a sail-on-service (SoS) opportunity, and involves replacing a vessel on the SoS service by the repositioning one. The repositioning vessel sails *in place* of the vessel previously on the service. The replaced vessel can be chartered out, laid up, or redeployed. The advantage of an SoS is that one vessel sails instead of essentially two vessels sailing in parallel, saving bunker fuel. Note that this is generally performed on the fronthaul of a service, i.e., the direction of a service earning the most revenue such as from China to the USA or Europe. The backhaul can then be left empty, as in many cases it is not profitable. Vessels join an SoS opportunity by either transshipping all containers from the on-service vessel, or by sailing in parallel with the on-service vessel for several ports and then transshipping any remaining containers to the repositioning vessel. Transshipments must respect *cabotage* restrictions, which are laws preventing foreign registered ships from carrying domestic cargo in many countries/regions around the world.

During repositioning, a vessel is allowed to add ports to a service that otherwise would not call that port (*induce* ports), or *omit* ports from its regularly scheduled route. Some restrictions apply to inducement, e.g., that the ports added should be in the same geographic region or trade zone.

The LSFRP considers the speed of the vessels during repositioning as a decision variable, as the amount of fuel consumed by container vessels varies with the vessel's speed roughly in a cubic relationship. For sailings between scheduled calls where the times on both sides of the sailing are fixed, we can (pre)compute the cost of the sailing according to the cubic function. Some ports do not have a fixed schedule, and these can also be visited by repositioning vessels. In this case, we choose the vessel speed during optimization according to a linearized fuel consumption function. Note, however, that in this work we focus on the version of the LSFRP in which the calls are scheduled, i.e., we do not need to optimize the vessel speeds directly in the model.

The time and cost of sailing between ports is influenced by external factors, such as the weather, port delays, and breakdowns. For example, if the ship is experiencing headwinds, it must exert more power to maintain its speed, and thus its scheduled travel time. Furthermore, the ship may have to divert on to a longer route at higher speed if there is extremely bad weather forecast for its original route.

Since repositioning plans are planned far in advance of their realization, the problem can be formulated in a two-stage framework. The first stage determines the routes, which cannot be easily changed during plan execution since berths must be reserved, and crews scheduled. The second stage prices the uncertainty related to demand fluctuations and travel delays. We note that a detailed view of delay handling, such as in Brouer et al (2013a), would be too computationally expensive to include in the second stage. We discuss the details of the stages in the next section.

Figure 4.1 shows an example repositioning route (blue) of a vessel for an instance from our dataset, which is described later in more detail. The solution is generated under the assumption of deterministic travel times. The repositioning vessel's starting

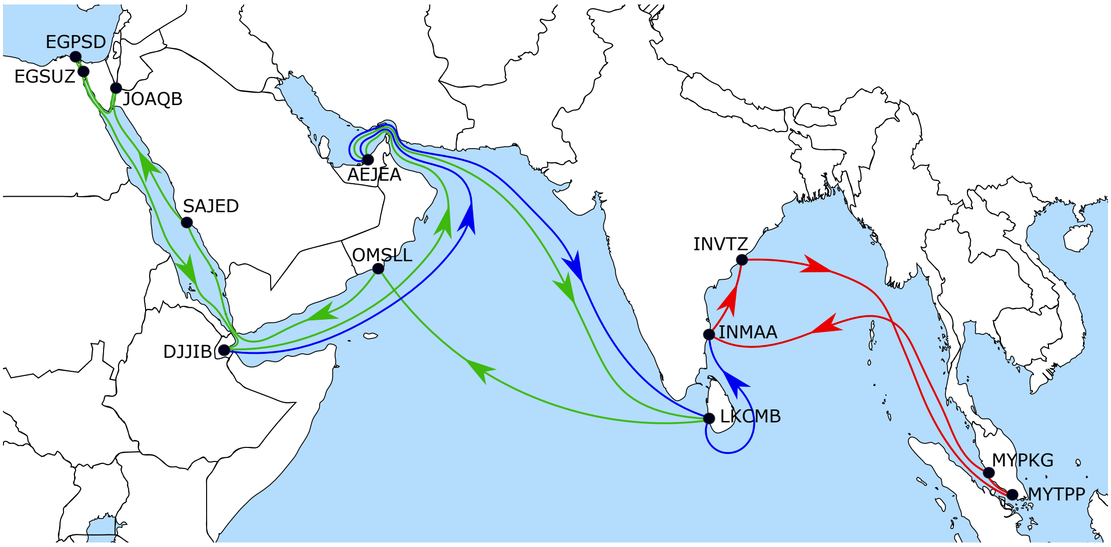


Figure 4.1: Repositioning route (blue), initial service (green) and goal service (red) for Instance 4-58-125-0 assuming deterministic travel times

service is the green service, and the earliest phase-out coincides with the port call at DJJIB. Subsequently, the vessel follows the route of the initial service (green) to AEJEA and LKCMB, where it phases out and joins the goal service (red) at INMAA.

4.4 Mathematical Model

We present a stochastic optimization model for the LSFRP under uncertainty, extending the notation and graph presented for the *inflexible* LSFRP in Tierney (2015). However, empty equipment is omitted because it does not have a major impact on the inflexible version of the LSFRP. We first provide an overview of the graph structure used to model the LSFRP, followed by an explanation of how we model demands using an arc flow alternative. We then describe how stochastic components are handled in the model, and finally provide the mathematical model.

4.4.1 Graph structure

The LSFRP is modeled using a graph in which a node represents a port and a time that the port is visited. We refer to the nodes as *visits*. Each vessel is associated with a node in the graph where it starts its journey. All vessels share a graph sink as the final destination along their route. Despite the fact that ships can basically sail anywhere across the sea, the graph is far from fully connected, as the arcs only connect activities that are allowed to be performed by the repositioning vessels. For example, a ship phasing out of a service in a the first slot cannot join the same phase-out service in a later slot. Moreover, the graph only contains arcs between visits when it is temporally feasible for a ship sailing at its maximum speed to make the voyage. Arcs connect visits on the same service or connect between services.

The graph is structured such that every possible path through the graph represents a feasible solution, as long as the paths of the vessels are disjoint, i.e., no two vessels visit the same node. For example, the SoS opportunities discussed in the previous section are completely embedded into the graph structure, including a basic model of cabotage restrictions. Furthermore, each slot of the goal service is represented by a sequence of connected nodes that then connects to the graph sink, meaning that the flow of the vessels through the graph ensures that every week of the goal service begins normal operations. As the mathematical formulation of the graph is rather long and detailed, we refer to Tierney et al (2014) for a formal description.

4.4.2 Node flow model

Traditionally, arc flow models are used to model the flow of demand in transportation problems. This means that the amount of transported goods is specified for each arc of the graph structure, in our case the number of containers of each demand on each arc. Arc flow models have $O(|V||M|)$ variables, where V is the set of nodes and M is the set of demands. As the size of the graph and number of demands increase, the number of variables can make these problems difficult to solve. In most transportation models the arc flow formulation is necessary to ensure that capacity constraints are respected for all vehicles transporting flow.

Tierney and Jensen (2013) develop an alternative flow model when all arcs have fixed durations that allows the flow of demands to be modeled using only $O(|S||M|)$ variables, where S is the set of vessels. This formulation is called a *node flow* model, since the capacity constraints are enforced on the nodes of the network instead of the arcs. The reason this is even possible is because of the disjoint paths of the vessels combined with the directed acyclic property of the graph. This means that when a vessel enters a node, we can compute which demands could potentially be on the ship through a reachability analysis in the graph and write a constraint over those demands for each ship accordingly. We refer to Tierney and Jensen (2013) for more details about the node flow model, but note that all necessary mathematical details for understanding this formulation are provided in our model.

4.4.3 Stochastic modeling

We present a two-stage stochastic model with a binary first stage and a continuous second stage. The first stage determines the routes of the vessels from their starting visit to the graph sink, meaning this portion of the model is unchanged from previous work on the LSFRP. In the second stage, we decide how many containers of the available demands are transported by the vessels and compute the costs of recovering from delays.

The modeling of delays poses a challenge, because (1) in real networks, delays propagate forwards in the network, (2) some delays may be unrecoverable, i.e., the delay is too large to still arrive on time, and (3) delays must be handled in the second stage, which should also only have continuous decision variables, as adding binary variables to the second stage would result in the model becoming extremely difficult to solve, as an integer second stage results in many NP-hard subproblems that must be solved (Conforti et al, 2014).

We extend the existing, deterministic node flow model by making the number of available and transported containers of a particular demand dependent on the scenarios. Furthermore, we model the scenario-dependent delays on arcs and the possibility to compensate delays by sailing faster causing higher costs. Delays that are not compensated are penalized in the objective function and propagated to the next arc the vessel is sailing on. We do not provide any recovery mechanisms as in, e.g., Brouer et al (2013a) or Li et al (2016), as these would require binary variables in the second stage.

We further extend the objective function computation of the LSFRP to analyze the risk associated with a certain repositioning plan. To do this, we use the conditional value-at-risk (CVaR) as a coherent risk measure (Rockafellar et al (2000); Artzner et al (1999)). An α-level CVaR is the mean value of the worst $\alpha\%$ of the scenarios. A high CVaR is especially important for risk-averse decision-makers to avoid repositioning plans with potentially high losses. We model the decision maker's risk aversion by deploying a mean-risk objective function, which can be controlled by a parameter $\lambda \in [0, 1]$. The case $\lambda = 1$ results in an optimization only of the expected profit, whereas $\lambda = 0$ leads to an exclusive optimization of the mean profit of the worst $\alpha\%$ of the scenarios (disregarding all other scenarios).

4.4.4 Mathematical model

Sets and Parameters

S	Set of ships.
V'	Set of visits except the graph sink.
A'	Set of arcs except the arcs connecting to the graph sink, i.e., $(i, j) \in A$, $i, j \in V'$.
A^{Sink}	Set of arcs connecting to the graph sink.
Q	Set of container types. $Q = \{dc, rf\}$.
M	Set of demand triplets of the form (o, d, q), where $o \in V', d \subseteq V'$ and $q \in Q$ are the origin visit, the set of destination visits and the cargo type, respectively.
M'	Set of single origin-destination pairs of demands, $M' := \cup_{(o,d,q)\in M} \cup_{d' \in d} (o, d', q)$. In other words, demands with multiple destinations are split into multiple demands, each with a single destination.

$M_i^{Vis'} \subseteq M$	Set of demands that can be feasibly transported through node $i \in V'$ based on whether the node is reachable from the demand origin and could visit at least one destination of the demand.
$M_{i,rf}^{Vis'}$	Set of reefer demands at node $i \in V'$.
Ω	Set of scenarios.
$P(\omega)$	Probability of occurrence of scenario $\omega \in \Omega$.
$u_s^q \in \mathbb{R}^+$	Capacity of vessel s for cargo type $q \in Q$.
M_i^{Orig}, $(M_i^{Dest}) \subseteq M$	Set of demands with origin (destination) visit $i \in V$.
$v_s \in V'$	Starting visit of ship $s \in S$.
$t_{si}^{Mv} \in \mathbb{R}$	Move time per TEU for vessel s at visit $i \in V'$.
$r^{(o,d,q)} \in \mathbb{R}^+$	Amount of revenue gained per TEU for the demand triplet.
$c_{sij}^{Sail} \in \mathbb{R}^+$	Fixed cost of vessel s utilizing arc $(i,j) \in A'$.
$c_{sij}^{VarSail} \in \mathbb{R}^+$	Additional hourly cost of vessel $s \in S$ sailing faster on arc $(i,j) \in A'$.
$c_i^{Mv} \in \mathbb{R}^+$	Cost of a TEU move at visit $i \in V'$.
$c_{si}^{Port} \in \mathbb{R}$	Port fee associated with vessel s at visit $i \in V'$.
$a^{(o,d,q)}(\omega) \in \mathbb{R}^+$	Amount of demand available for the demand triplet (o,d,q) in scenario ω.
$\mathit{In}(i) \subseteq V'$	Set of visits with an arc connecting to visit $i \in V$.
$\mathit{Out}(i) \subseteq V'$	Set of visits receiving an arc from $i \in V$.
$\tau \in V$	Graph sink, which is not an actual visit.
$l_{ij}(\omega)$	Delay caused by arc $(i,j) \in A$ in scenario $\omega \in \Omega$.
$d_{sij}^{Diff} \in \mathbb{R}^+$	Difference between the maximum and the minimum duration for vessel s to sail on arc $(i,j) \in A$.
p_i	Penalty factor for delay at visit $i \in V$.
α	Confidence level of the risk measure, $0 \leq \alpha \leq 1$.
λ	Profit weight in the objective function, $1-\lambda$ is the weight of the $CVaR$ objective.
$b(\omega)$	“Big M” constant (upper bound) for the delay in scenario ω.

Variables

$x_s^{(o,d,q)}(\omega) \in \mathbb{R}_0^+$	Amount of demand triplet $(o,d,q) \in M'$ carried on ship $s \in S$ in scenario $\omega \in \Omega$.
$y_{sij} \in \{0,1\}$	Indicates whether vessel s is sailing on arc $(i,j) \in A$.
$e_{ij}(\omega) \in \mathbb{R}_0^+$	Remaining delay on arc $(i,j) \in A$ after possible compensation of delay in scenario $\omega \in \Omega$.
$z_{sij}(\omega) \in \mathbb{R}_0^+$	Time saved by vessel s on arc $(i,j) \in A$ by sailing faster than the minimum possible speed in scenario $\omega \in \Omega$.
$\theta(\omega) \in \mathbb{R}$	Profit that is generated in scenario $\omega \in \Omega$.
$t \in \mathbb{R}$	Auxiliary variable for CVaR constraints.
$y^0(\omega) \in \mathbb{R}_0^+$	Auxiliary variables for CVaR constraints in scenario ω.
$CVaR \in \mathbb{R}$	Auxiliary variable representing the CVaR.

Objective and Constraints

$$\max \lambda \sum_{\omega\in\Omega} P(\omega)\theta(\omega) + (1-\lambda)CVaR \tag{4.1}$$

$$\text{s. t.}\ \ \theta(\omega) = \sum_{s\in S}\sum_{(o,d,q)\in M'} \left(r^{(o,d,q)} - c_o^{Mv} - c_d^{Mv}\right) x_s^{(o,d,q)}(\omega) \tag{4.2}$$

$$-\sum_{s\in S}\sum_{(i,j)\in A'} c_{sij}^{VarSail} z_{sij}(\omega) \tag{4.3}$$

$$-\sum_{(i,j)\in A'} p_j e_{ij}(\omega) \tag{4.4}$$

$$-\sum_{s\in S}\sum_{(i,j)\in A'} c_{sij}^{Sail} y_{sij} \tag{4.5}$$

$$-\sum_{j\in V'}\sum_{i\in In(j)}\sum_{s\in S} c_{sj}^{Port} y_{sij} \qquad \forall \omega \in \Omega \tag{4.6}$$

$$t - y^0(\omega) \leq \theta(\omega) \qquad \forall \omega \in \Omega \tag{4.7}$$

$$CVaR = t - \frac{1}{1-\alpha}\sum_{\omega\in\Omega} P(\omega) y^0(\omega) \tag{4.8}$$

$$\sum_{s\in S}\sum_{i\in In(j)} y_{sij} \leq 1 \qquad \forall j \in V' \tag{4.9}$$

$$\sum_{j\in Out(i)} y_{sij} = 1 \qquad \forall s \in S, i = v_s \tag{4.10}$$

$$\sum_{i\in In(\tau)}\sum_{s\in S} y_{si\tau} = |S| \tag{4.11}$$

$$\sum_{i\in In(j)} y_{sij} - \sum_{i\in Out(j)} y_{sji} = 0 \qquad \forall j \in \{V' \setminus \bigcup_{s\in S} v_s\}, s \in S \tag{4.12}$$

$$x_s^{(o,d,q)}(\omega) \leq a^{(o,d,q)}(\omega) \sum_{i \in Out(o)} y_{oi}^s \qquad \forall (o,d,q) \in M', s \in S, \omega \in \Omega \tag{4.13}$$

$$x_s^{(o,d,q)}(\omega) \leq a^{(o,d,q)}(\omega) \sum_{i \in In(d)} y_{id}^s \qquad \forall (o,d,q) \in M', s \in S, \omega \in \Omega \tag{4.14}$$

$$\sum_{(o,d,q) \in M_i^{Vis'}} x_s^{(o,d,q)}(\omega) \leq u_s^{dc} \qquad \forall s \in S, i \in V', \omega \in \Omega \tag{4.15}$$

$$\sum_{(o,d,q) \in M_{i,rf}^{Vis'}} x_s^{(o,d,q)}(\omega) \leq u_s^{rf} \qquad \forall s \in S, i \in V', \omega \in \Omega \tag{4.16}$$

$$\sum_{s \in S} \sum_{d \in d'} x_s^{(o,d',q)}(\omega) \leq a^{(o,d,q)}(\omega) \qquad \forall (o,d,q) \in M', \omega \in \Omega \tag{4.17}$$

$$\begin{aligned} & e_{ki}(\omega) + \sum_{s \in S} y_{sij}(l_{ij}(\omega)) \\ & - b(\omega) \left(2 - \sum_{s \in S} (y_{ski} + y_{sij}) \right) \\ & - \sum_{s \in S} z_{sij}(\omega) \leq e_{ij}(\omega) \qquad \forall (i,j) \in A, k \in In(i), \omega \in \Omega \end{aligned} \tag{4.18}$$

$$y_{sij} l_{ij}(\omega) - z_{sij}(\omega) \leq e_{ij}(\omega) \qquad \forall s \in S, \omega \in \Omega, i = v_s, (i,j) \in A \tag{4.19}$$

$$z_{sij}(\omega) \leq d_{sij}^{Diff} y_{sij} \qquad \forall (i,j) \in A, s \in S, \omega \in \Omega \tag{4.20}$$

$$e_{ij}(\omega) \leq b(\omega) \sum_{s \in S} y_{sij} \qquad \forall (i,j) \in A, \omega \in \Omega \tag{4.21}$$

The domains of the variables are as previously described. The objective function (4.1) is composed of two parts: the profits of the single scenarios weighted by their respective probabilities of occurrence, and the CVaR value. The expected profit and the CVaR value are weighted by λ and $1 - \lambda$, respectively, to express their importance in relation to each other. The profit for a single scenario consists of the following components. The profit gained for delivering cargo in Term (4.2) is computed for each scenario, ω, based on the revenue from delivering cargo minus the cost to load and unload the cargo. It is possible to deliver a fractional amount of cargo since every demand is an aggregation of containers between two visits. Therefore, a maximum of one container of a demand is fractional. Term (4.3) sums the additional costs for sailing faster than the previously planned minimum speed to recover from delays. Furthermore, Term (4.4) ensures that accumulated delays (i.e., delays that cannot be eliminated through speeding up and are propagated to the next used arc) are penalized depending on the visit and the number of containers moved there. The stage one sailing costs in Term (4.5) take into account the precomputed sailing costs for arcs between the visits. The precomputed sailing costs consist mainly of fuel costs for sailing at minimum speed to be on time, but can also include canal fees or chartering revenue using an SoS opportunity. Finally, port fees are deducted in (4.6) for all visits. Constraints (4.7) and (4.8) model the CVaR, with Constraint (4.8) calculating the second part of the objective function. Rockafellar and Uryasev (2002) show that the CVaR can be calculated by $\max\{t - \frac{1}{1-\alpha}\sum_{\omega \in \Omega} P(\omega)[t - \theta(\omega)]\}$. This formulation can be transformed to Constraints (4.7) and (4.8) to include the CVaR in an optimization model according to Fábián (2008).

The first stage of the model is represented by the constraints (4.9) to (4.12). Multiple vessels are prevented from calling the same visit in Constraints (4.9). The flow of each vessel from its source node to the graph sink is handled by Constraints (4.10), (4.11) and (4.12), with Constraint (4.11) ensuring that all vessels arrive at the sink. Constraints (4.10) guarantee that all vessels leave their starting port and Constraints (4.12) ensure that vessels leave entered ports. Note that with the exception of Constraint (4.11), these constraints form a node disjoint path problem for the vessels and form the entirety of the stage one decisions.

The rest of the constraints model the second stage. Constraints (4.13) and (4.14) allow a demand to be carried only if the same vessel visits both the origin and a destination of the demand, respectively. As the paths of the vessels are node disjoint, we do not need to explicitly restrict the demands to be carried only by a single vessel. In Constraints (4.15) and (4.16), we ensure that the capacity of the vessel is not exceeded at any visit for both dry and reefer containers, respectively. Constraints (4.17) ensure that the amount of demand carried for each single OD pair does not exceed the amount of containers that are actually available. This is important because demands with multiple destinations are split into demands with single destinations for this model formulation, but the sum of all of the split demands must not exceed the number of containers of the original demand.

Constraints (4.18) through (4.20) model the vessel's response to delay in the scenarios. The delay for each arc is set in Constraints (4.18), which is computed from three inputs. First, the delay on the incoming arc is propagated to the current arc. Then, we include additional delay (should any be present) on the current arc according to the scenario. Next, we ensure that this constraint is only binding if both arcs (k, i) and (i, j) are used by a ship, otherwise we get a large negative value on the left side of the inequality, allowing the delay on (i, j) to be set to 0. It is guaranteed that both arcs are traversed by the same vessel, as vessels are forced to leave visits they enter. Finally, we allow the ship to speed up to reduce its delay, resulting in the remaining delay for the current arc.

Constraints (4.19) set the delays for arcs originating at the starting visits of vessels by subtracting the time saved by speeding up from the delay caused by the arc. Constraints (4.20) prevent the speed up from exceeding the maximum speed of the vessel on the arc and limit the time that can be used to compensate delays, which depends on the difference of the maximum and minimum sailing duration on the arc. Constraints (4.21) are used to set the maximum possible delays on arcs. It is important to limit the delay on an arc without a vessel to zero to prevent the model from propagating delay to arcs that are not used by vessels. Without this restriction, delays would be propagated to arcs without vessels to save compensation costs, which is clearly nonsense.

4.5 Computational Results

We evaluate our stochastic model on a dataset of instances based on real-world data to see to what extent handling uncertainty in the model can lead to better repositioning plans. We show the impact of uncertain travel times and demands on repositioning plans and evaluate the influence of the CVaR risk measure. The central research question that we address is: to what extent are the solutions found with a stochastic model better than those found with a deterministic model? We start by giving an overview over the instances we examine and the process of obtaining scenarios for uncertain demands and travel times. Afterwards, we present the computational results for all instances, and analyze the results for one instance in particular to show in detail how solutions change when considering uncertain demand and travel time data individually and in combination.

4.5.1 Experimental setup

We use the “inflexible” instances from Tierney (2015) and extend them with stochastic scenarios, as described in the following subsection. Table 4.2 describes the properties of each instance, where each instance has a number of vessels $|S|$, nodes $|V|$ that are equivalent to the time windows to visit ports, arcs $|A|$, demands $|M|$, ports with equipment surpluses or demands $|E|$ and SoS opportunities $|SOS|$. We model the stochastic LSFRP in the modeling language MPL (Maximal Software, 2016) and solve it with Gurobi 9.0.2 (Gurobi Optimization, 2020) on an AMD Ryzen 9 3950X 16-Core 4.20GHz processor with 128 GB of RAM using all cores. Note that our focus is on the stochastic extensions of the LSFRP and not on runtimes, thus we solve the deterministic equivalents with Gurobi. Better runtimes can likely be achieved by applying more specialized methods such as a Benders-based decomposition.

We run experiments for three different setups considering uncertain demand data and uncertain delay data independently, as well as in combination. To examine the influence of stochastic travel times in isolation, the demands are set to the average demand scenario, whereas for the case of uncertain demands, the delays are set to zero, as is generally assumed in deterministic planning. As we deal with large-scale problems, only the smallest instances can be solved to optimality with all generated scenarios within an acceptable run time. Therefore, we reduce the number of scenarios to an in-sample set and use an out-of-sample evaluation to analyze the performance of the stochastic models. We discuss our scenario generation and reduction strategies in the following sections.

ID	$\|S\|$	$\|V\|$	$\|A\|$	$\|M\|$	$\|E\|$	$\|SOS\|$	Name in Tierney (2015)
3-36-28-0	3	36	150	28	0	1	repos1p
3-38-24-0	3	38	151	24	0	2	repos3p
3-42-20-0	3	42	185	20	0	3	repos4p
3-51-22-0	3	51	270	22	0	3	repos5p
3-54-46-0	3	54	196	46	0	4	repos7p
4-58-125-0	4	58	499	125	0	0	repos10p
4-74-145-0	4	74	603	145	0	2	repos12p
4-80-155-0	4	80	632	155	0	4	repos13p
4-80-155-24	4	80	632	155	24	4	repos14p
5-71-173-0	5	71	355	173	0	0	repos15p
5-106-320-0	5	106	420	320	0	5	repos16p
6-102-75-0	6	102	1,198	75	0	0	repos17p
6-135-87-0	6	135	1,439	87	0	9	repos18p
6-142-80-0	6	142	1,865	80	0	4	repos20p
6-142-80-13	6	142	1,865	80	13	4	repos21p
7-75-154-0	7	75	482	154	0	3	repos24p
7-77-156-0	7	77	496	156	0	0	repos25p
7-77-156-16	7	77	496	156	16	0	repos26p
7-79-188-0	7	79	571	188	0	0	repos27p
7-90-189-0	7	90	618	189	0	4	repos28p
7-90-189-19	7	90	618	189	19	4	repos29p
8-144-170-0	8	144	1,501	170	0	3	repos32p

Table 4.2: Structure of the instances

4.5.2 Scenario generation

We generate demand scenarios using historical demands from a liner shipping carrier covering a time horizon of 18 weeks. Given a problem instance, we generate a scenario by sampling an amount $a^{(o,d,q)}(\omega)$ from one of the 18 weeks available for each of the demand triplets in the instance. Overall, we generate 1000 demand scenarios for every problem instance. We assume that this number of scenarios gives a good representation of the underlying demand distribution while still enabling out-of-sample analysis within a reasonable amount of time for our problem instances.

The delays in a scenario are determined by combining weather data with travel time distributions between ports from Tierney et al (2019) to ensure delays are temporally and geographically consistent and, thus, realistic. We use a three-parameter, log-logistic distribution to sample the delays $l_{ij}(\omega)$ as it has been shown to be an adequate model for vessel lateness in maritime operations (Halvorsen-Weare et al, 2013; Agra et al, 2016; Rodrigues et al, 2019; Tierney et al, 2019). This distribution is not simply sampled for each arc independently, as doing so would lead in many cases to nonsense scenarios in which large delays could coexist at the same time and in the same region as no delays at all. Furthermore, the distribution is aggregated over time and we want scenarios for different points in time. Thus, we use weather information from Kuhlemann and

Tierney (2020) and the weather routing heuristic from that work to generate weather-optimized routes between ports for a discrete set of time points. Next, we sample from a distribution, using the parameters for various port-to-port connections from Tierney et al (2019) based on the impact of the weather on the route, the details of which follow.

Specifically, we investigate 18 different time points for the vessel to depart from a starting port, and therefore 18 different weather situations for each connection. We use the average weather on the route to determine the quantile of the distribution for the delay depending on the severity of the weather. The delays for all arcs for one delay scenario are sampled based on the same weather scenario, preventing us from generating independent delay values for arcs that are geographically close to each other. Each of the 18 weather situations is used to generate 56 scenarios by drawing 56 random delay values from the previously determined quantile of the distribution, leading to 1008 scenarios overall. This gives us a similar number of travel time scenarios to demand scenarios.

As computing routes for every single origin destination pair would be too expensive, we compute ten connections between random ports from the clusters already used to generate the delay distributions in Tierney et al (2019). When given a particular origin destination pair for which we need the delay, we consider the weather on the route with the length closest to the length of the original pair.

Regarding the analysis of simultaneous travel time and demand uncertainty, we assume that demands and delays are mutually independent and uncorrelated. Hence, we represent simultaneous travel time and demand uncertainty by constructing the Cartesian product of delay and demand scenarios, which leads to a base set of more than 1 million combined scenarios. In the following section, we explain how we reduce this set to get manageable out-of-sample and in-sample computations.

4.5.3 Scenario reduction

Stochastic optimization models are known to be extremely hard to solve with a large number of scenarios. In our case, we have up to a million scenarios in the combined demand and delay setting. It is essential that we reduce the number of scenarios input into the model so our instances can be solved in a reasonable amount of time. Therefore, we seek to find reduced sets of scenarios that approximate the initial distributions reasonably well. Since we are dealing with underlying multivariate distributions of very high dimensions, i.e., many hundreds of random delays and demands, we utilize the results of Löhndorf (2016). The author analyzes different clustering methods for scenario generation in the presence of high dimensionality and introduces a method called *Voronoi cell sampling (VCS)* as a novel scenario reduction technique.

We follow Löhndorf's approach and evaluate the performance of different clustering methods for the case of uncertain travel time. We reduce the original set of delay scenarios to 50 by drawing a random sample ("Random") and by clustering with the

Type of test	Random	k-means	k-means++	k-mediods	k-mediods++	VCS-Cent	VCS-Rand
OOSS %	0.21	0.16	0.20	0.21	0.16	0.20	0.0
(Dev.Opt.%)	(0.25)	(0.32)	(0.24)	(0.25)	(0.08)	(0.28)	(0.00)
ISS %	2.49	0.93	0.34	0.89	0.30	0.38	0.24
(Dev.Opt.%)	(-0.93)	(-1.04)	(-1.51)	(-0.36)	(-0.89)	(-0.01)	(-0.26)

Table 4.3: Relative standard deviation of the average expected value out-of-sample and in-sample for instance 4-58-125-0 (15 test runs, $k = 50$) and deviation from the optimal solution (Dev.Opt.%)

Type of test	Random	k-means	k-means++	k-mediods	k-mediods++	VCS-Cent	VCS-Rand
OOSS %	2.87	2.60	0.00	0.27	0.00	0.00	0.00
ISS %	3.62	2.43	1.10	2.13	1.06	0.83	0.75

Table 4.4: Relative standard deviation of the average expected value out-of-sample and in-sample for instance 6-102-75-0 (15 test runs, $k = 50$)

following algorithms: k-means, k-means++, k-medoids, k-medoids++ and VCS. VCS reduces the contraction of scenarios towards the mean by combining clustering and random sampling. More precisely, it draws one random realization from each partition generated with the k-means algorithm (VCS-Rand) in each of a certain number of iterations. Additionally, we test a modified version of VCS that draws the centroid from each partition (VCS-Cent).

We measure the quality of the generated subsets regarding our discrete and continuous multivariate distribution using the out-of-sample stability (OOSS) and in-sample stability (ISS) following the experimental approach of Kaut and Wallace (2003). As in this work, we use 15 test runs of each approach. The results are reported in Tables 4.3 and 4.4, in which we state the relative standard deviation σ of the average of the objective values *Avg* over all test runs for a medium and large sized instance, respectively.

The OOSS and ISS values are calculated by σ/Avg for the respective samples. In Table 4.3, we also provide the relative deviation from the mean of all test runs to the optimal solution *Opt*, based on all 1008 scenarios, calculated by $1 - (Avg/Opt)$. In Table 4.4 this value is not given, as the instance is too large to find the optimal solution in a reasonable amount of time.

The experimental results reveal that the clusters generated by *VCS-Rand* work best for our high-dimensional vectors. It shows stability, as the standard deviation is zero for the out-of-sample test runs and does not exceed 0.24% and 0.75% for the in-sample test runs for the two instances. These are the smallest values among all tested clustering methods. Therefore, we use *VCS-Rand* to create scenario subsets for all problem instances. Furthermore, the results support the idea that a subset size of 50 is appropriate to approximate the original set of scenarios for both medium as well as large instances.

We apply VCS to reduce the number of combined scenarios (1 million) to a size of 1000 scenarios, which is computationally manageable for out-of-sample evaluations. As delays and demand amounts are given in incomparable units, we scale the data to values between zero and one during scenario reduction. This set is then reduced to smaller sets of 50 scenarios for in-sample computations applying the same methodology again.

4.5.4 Delay penalties

The delay penalties, p_i, for arriving too late at a visit are calculated using a linear penalty function depending on the port times of vessels. Our goal is to discourage late arrivals while avoiding needing a detailed model of container flows missing transshipments or inland connections. As Li et al (2015) state, a longer port time usually implies a higher number of loaded and unloaded containers and therefore a higher impact for customers, leading to higher penalties. In two further approaches, Li et al (2015) use a step-wise function assuming different penalties for different ranges of delays and a combination of the linear, port time dependent and the step-wise, delay dependent approach. Li et al (2016) use a convex delay penalty function containing the square of the delay modeling the fact that higher delays cause a much higher reputation damage. The square of the delay is multiplied by a factor that depends on the expected port time on the basis of the same assumption Li et al (2015) make.

The delay of a vessel is calculated relative to the beginning of the planned time window at the visit and the maximum delay scenario is set to seven days for our calculations. When a vessel has a higher delay, the containers can be transported by the vessel in the next slot in the same cycle. We use a linear, port time dependent approach that allows us to take into account visit specific penalties for delays. To this end, we set the value of the penalty p_i equal to the expected number container moves at the corresponding port (port time in visit i times the historical average number of container moves per hour at the port) multiplied by a a constant factor of 100 representing the incurred cost. This value is based on the suggestion of Li et al (2015).

4.5.5 Experimental results for all repositioning instances

We now investigate the effectiveness of the stochastic model compared to solutions generated using average values for the uncertain parameters over our dataset of LSFRP instances. For all instances, we present out-of-sample test results that we obtain by fixing the optimal first stage solution of the stochastic model for the subset of 50 scenarios (in-sample computation) and its subsequent evaluation on the larger set of scenarios (1000 scenarios in case of demand uncertainty and 1008 scenarios in case of delay uncertainty). The upper bound for the delay of an arc $b(\omega)$ is set to the sum of all delays of all arcs of the instance for our computational tests, since this is the maximum delay value that can be encountered on an arc.

Tables 4.5, 4.6 and 4.7 show our main results by reporting the following measures for each problem instance *ID*. As reference values, we calculate the Wait-and-See value (WS^{oos}) as well as the expected value of the expected value solution (EEV^{oos}) with respect to the larger (out-of-sample) scenario sets. The WS value is the mean value of the optimal solution values of all the scenarios solved individually. To obtain the EEV, first, a deterministic model is solved using the average value over all scenarios of each uncertain parameter. Its first-stage solution (i.e., the routings of the vessels) is then evaluated (i.e. fixed and solved) for all scenarios individually. The EEV is the mean value of the resulting optimal solution values. For the stochastic model, we report Here-and-Now (HN) solution values $HN1.0^{oos}$, $HN0.5^{oos}$, $HN0.1^{oos}$ for three different values for the weight λ controlling the mean-risk profit function. Setting λ to 1.0 optimizes the expected value. With a decreasing λ, the focus shifts increasingly to the CVaR objective, which we set to use a 90% confidence level (CVaR90) to generate a robust solution.

Note that due to the out-of-sample evaluations, the classical inequality $EEV \leq HN \leq WS$ does not necessarily hold for the values we report. In Tables 4.5, 4.6 and 4.7, the HN values are given relative to the EEV in percent and the WS values relative to the HN values in percent, such that violations of this inequality can be easily detected. The fact that it is actually fulfilled for all instances in Tables 4.5, 4.6 and 4.7 emphasizes the high stability of the scenario reduction technique discussed in Section 4.5.3. We report the relative CVaR90 values of the resulting distributions in parentheses below the out-of-sample expected values. In our experiments, we limit the runtime for in-sample computations to 7200 seconds. Instances that could not be solved to optimality within this runtime are marked with an asterisk. For these instances, the MIP gap is given in square brackets.

Table 4.5 shows the computational results considering only delay uncertainty, illustrating a significant benefit of using the stochastic model formulation. Compared to the EEV solutions, the expected values are up to ten times higher, as seen for instances 6-135-87-0, 6-142-80-0 and 6-142-80-13. The superiority of the stochastic model results from the reduction of the average delay leading to less delay penalties. Reducing the average delay is possible by visiting fewer ports and by avoiding routes with a high likelihood of delay. We analyze this in detail for instance 4-58-125-0 in Section 4.5.6.

The CVaR90 of the $HN1.0^{oos}$ solution increases considerably for most instances compared to the EEV solution regarding the profit of the 10% worst scenarios. Further improvement is often possible by explicitly considering the CVaR in the objective function at the expense of expected profit. For instances 4-58-125-0, 4-80-155-0 and 4-80-155-24, we obtain the highest relative improvements of the expected CVaR90 of up to 692%, while the improvement of the expected profit reduces to 1.9%. All test instances with four vessels have similarities in the network structure resulting in the same routes for vessels, which leads to almost identical expected values for the EV model and the stochastic model. For the instances 7-75-154-0 and 7-77-156-16, we observe a significant improvement of the CVaR and a slight improvement of the expected profit at the same time. These seemingly contradictory results, which we do not see in pure in-sample results, can be explained by the out-of-sample evaluations. A first-stage solution with a reduced expected profit for the 50 in-sample scenarios can have an (typically only slightly) increased expected profit for the 1000 out-of-sample scenarios. In general, the limited effect of the explicit optimization of the CVaR is surprising. For many instances, the CVaR cannot be significantly improved without sacrificing a considerable fraction of the expected profit. This might indicate that the routing flexibility in repositioning problems is limited.

Regarding the results on stochastic demands presented in Table 4.6, we observe that the impact of uncertainty is not as high as for stochastic travel times. The maximum increase relative to the EEV amounts to 5.2% for instance 8-144-170-0. For 15 out of 22 instances, the stochastic model formulation does not provide a benefit. This can be explained by the structure of the instances, in that they lack the flexibility to use many different routes, despite the high variance of the demand scenarios. Especially the inflexibility of smaller instances makes it difficult to find alternative routes. Furthermore, sailing costs are relatively high and can hardly be compensated by additional revenue from carrying extra cargo. A cost analysis reveals that possible additional yields from transported cargo are often accompanied with extra sailing cost and port fees of almost the same height. We present a detailed analysis of this effect for instance 4-58-125-0 in Section 4.5.6. As repositioning is a costly process, even relative improvements between 0.4% and 5.2% imply high absolute cost savings. Beyond that, the increase in robustness generated by optimizing the CVaR90 is non-negligible for 10 instances. Relative improvements of up to 990% are possible as we can see for instances 7-90-189-0 and 7-90-189-19, however, at the cost of almost half the expected profit.

We report results simultaneously considering uncertainty of demands and travel times in Table 4.7. We observe that the stochastic model provides significant benefits compared to the deterministic model on average data (EEV). In the case of pure profit maximization ($\lambda = 1.0$), the expected profit of the stochastic model outperforms the EEV by a factor of up to ten (for instance 5-71-173-0). An analysis of the costs in the objective function shows that for all instances the increased profit is mainly attributable to reducing the delay penalty. As in the previous cases, increasing the weight of the CVaR on the objective function leads to more robust solutions in most cases, usually at the expense of a decreasing expected profit. In some cases, we can again observe the peculiar effect of an increasing expected profit or a decreasing CVaR. As discussed above, this phenomenon can be explained by the out-of-sample analysis. The effect vanishes for a pure in-sample analysis (which is clearly less informative for the decision maker). Another reason, e.g., for instance 8-144-170-0, is that the instance is not solved to optimality.

Furthermore, we observe a seemingly higher benefit of the stochastic model for combined uncertainties (Table 4.7) compared to pure delay or demand uncertainty (Tables 4.5 and 4.6). However, the relative improvements are not comparable as they relate to different EEV values. We analyze this effect in detail for instance 4-58-125-0 in Section 4.5.6.

Comparing the EEVs of instances in Tables 4.5, 4.6 and 4.7 reveals that there are discrepancies in the absolute objective values. Taking instance 3-36-28-0 as an example, the EEV is negative for the cases considering delays and both uncertainties combined, whereas it is positive for the case of uncertain demands. This can be explained by the artificial cost used to penalize delays. Separating the penalties from the objective function reveals that the EEV solutions for the individual consideration of delay, demand and the combination of the two are all similar. This can be exemplified for instance 3-36-28-0, where the average delay penalty of all out-of-sample scenarios is about 3.1 million when considering uncertain delay only, which almost coincides with the gap to the EEV in the case of pure demand uncertainty.

We conclude that the benefit of the stochastic model in the presence of of delay uncertainty is considerable, whereas it is significantly smaller in the presence of demand uncertainty. In a joint analysis of demand and delay uncertainty, expected profit values are dominated by delay penalties, so the benefits is at least as high as in the pure case of uncertain travel times.

Tables 4.8, 4.9 and 4.10 show the run times of the experiments in seconds. The nomenclature used for the tables is as follows:

- *ID*: name of the instance
- *PT1*: parsing time for the deterministic model (1 scenario)
- *ST1*: solution time for the deterministic model (1 scenario)
- *PT50*: parsing time of the stochastic model (50 scenarios) with the respective weight λ
- *ST50*: solution time of the stochastic model (50 scenarios) with the respective weight λ

Comparing the results in Tables 4.8 to 4.10, it can be observed that the type of uncertainty, while not affecting problem dimensions, has a considerable effect on the runtime. With few notable exceptions, the problem instances with uncertain travel times feature significantly higher runtimes than the instances with demand uncertainty. This is caused by the fact that delay uncertainty provides the model with many more variables, and thus options for sailing on the routes. The instances with combined uncertainty show similar runtimes as under travel time uncertainty. An exception from the rule is instance 4-80-155-24. Furthermore, runtime increases with an increasing weight on the CVaR optimization.

ID	EEV^{oos} ($CVaR90$)	WS^{oos}	$HN1.0^{oos}$ ($CVaR90$)	$HN0.5^{oos}$ ($CVaR90$)	$HN0.1^{oos}$ ($CVaR90$)
3-36-28-0	-2826660.3 (-15836734.6)	74.0	0.0 (0.0)	-41.2 (74.7)	-41.2 (74.7)
3-38-24-0	-5316886.2 (-18322448.1)	38.9	0.0 (0.0)	-16.4 (66.1)	-16.4 (66.1)
3-42-20-0	-2590942.8 (-2598248.6)	-0.0	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
3-51-22-0	-682405.4 (-689718.9)	-0.0	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
3-54-46-0	10423878.9 (10402427.6)	0.0	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
4-58-125-0	28995602.6 (3845545.3)	7.6	5.5 (463.3)	5.1 (691.7)	5.1 (691.7)
4-74-145-0	28995602.6 (3845545.3)	10.1	2.3 (439.3)	1.9 (667.7)	1.9 (667.7)
4-80-155-0	28995602.6 (3845545.3)	10.1	2.3 (439.3)	1.9 (667.7)	1.9 (667.7)
4-80-155-24	28995602.6 (3845545.3)	10.1	2.3 (439.3)	1.9 (667.7)	1.9 (667.7)
5-71-173-0	-37940536.3 (-191689947.6)	58.2	121.5 (101.5)	121.5 (101.5)	121.5 (101.5)
5-106-320-0	-44233350.6 (-213170269.7)	57.8	118.5 (101.3)	118.5 (101.3)	118.5 (101.3)
6-102-75-0	2247454.7 (-110016090.3)	42.6	862.1 (109.6)	818.1 (113.2)	818.1 (113.2)
6-135-87-0	1826831.2 (-99295865.2)	36.8	1188.7 (112.8)	1134.6 (116.4)	1134.6 (116.4)
6-142-80-0*	1859584.5 (-110403960.3)	45.0	1029.3 (110.0)	*998.8 [0.4] *(113.0)	985.9 (112.8)
6-142-80-13*	1859584.5 (-110403960.3)	45.0	1029.3 (110.0)	998.8 (113.0)	*998.8 [0.3] *(113.0)
7-75-154-0	17047198.2 (4279393.8)	8.3	0.0 (0.0)	0.9 (262.0)	0.9 (262.0)
7-77-156-0	17316583.6 (5694763.2)	6.8	0.0 (0.0)	1.0 (177.2)	1.0 (177.2)
7-77-156-16	17316583.6 (5694763.2)	5.7	1.0 (177.2)	1.0 (177.2)	1.0 (177.2)
7-79-188-0	5353558.6 (1681207.0)	4.8	0.0 (0.0)	-1.8 (5.4)	-1.8 (5.4)
7-90-189-0	4233462.6 (-1855035.1)	20.7	0.0 (0.0)	0.0 (0.0)	-53.6 (26.5)
7-90-189-19	4912673.3 (1910003.1)	4.1	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
8-144-170-0	36872670.8 (24549050.2)	2.8	1.4 (27.1)	-1.8 (43.3)	-1.8 (43.3)

Table 4.5: Results for uncertain travel time. All values result from out-of-sample evaluations (in-sample size 50, out-of-sample size 1008). HN values are given relative to EEV values in percent, WS values are relative to HN values in percent. Below the OOSS expected values, (relative) CVaR90 values of the resulting distributions are given in parentheses.

ID	EEV^{oos} (*CVaR90*)	WS^{oos}	$HN1.0^{oos}$ (*CVaR90*)	$HN0.5^{oos}$ (*CVaR90*)	$HN0.1^{oos}$ (*CVaR90*)
3-36-28-0	328518.5	35.7	0.0	0.0	-188.1
	(-2947736.6)		(0.0)	(0.0)	(-11.3)
3-38-24-0	-2155686.8	6.2	0.0	-15.2	-15.2
	(-5436741.8)		(0.0)	(-0.7)	(-0.7)
3-42-20-0	-2665354.1	0.0	0.0	0.0	0.0
	(-3815612.8)		(0.0)	(0.0)	(0.0)
3-51-22-0	-676546.3	0.0	0.0	0.0	0.0
	(-786842.2)		(0.0)	(0.0)	(0.0)
3-54-46-0	10417398.1	0.6	0.0	0.0	0.0
	(9882596.5)		(0.0)	(0.0)	(0.0)
4-58-125-0	30734015.0	10.3	0.4	-1.4	-1.4
	(20900168.9)		(8.3)	(14.2)	(14.2)
4-74-145-0	30734015.0	8.5	1.1	-1.4	-1.4
	(20900168.9)		(9.3)	(14.2)	(14.2)
4-80-155-0	30734015.0	9.1	1.1	-1.3	-1.3
	(20900168.9)		(9.3)	(14.3)	(14.3)
4-80-155-24	30734015.0	9.1	1.1	-1.3	-1.3
	(20900168.9)		(9.3)	(14.3)	(14.3)
5-71-173-0	18800734.0	7.0	0.0	0.0	-0.2
	(16112621.8)		(0.0)	(0.0)	(-0.1)
5-106-320-0	18800734.0	7.0	0.0	0.0	0.0
	(16112621.8)		(0.0)	(0.0)	(0.0)
6-102-75-0	31127184.2	2.4	1.2	1.2	1.2
	(21299672.9)		(3.2)	(3.2)	(3.2)
6-135-87-0	32661518.8	3.0	1.2	1.2	1.2
	(22747290.9)		(3.2)	(3.2)	(3.2)
6-142-80-0	29718467.5	4.0	0.0	0.0	0.0
	(19890008.1)		(0.0)	(0.0)	(0.0)
6-142-80-13*	29718467.5	4.4	-0.3	*-2.6 [0.1]	*-0.8 [0.1]
	(19890008.1)		(-0.5)	*(-3.8)	*(-1.1)
7-75-154-0	18016594.7	6.1	0.0	0.0	0.0
	(10623057.0)		(0.0)	(0.0)	(0.0)
7-77-156-0	18007800.9	5.9	0.0	0.0	0.0
	(10604380.3)		(0.0)	(0.0)	(0.0)
7-77-156-16	18007800.9	5.9	0.0	0.0	0.0
	(10604380.3)		(0.0)	(0.0)	(0.0)
7-79-188-0	5137584.8	6.1	0.0	-25.4	-25.4
	(1648099.5)		(0.0)	(42.9)	(42.9)
7-90-189-0	4450167.9	7.5	0.0	-44.8	-44.8
	(-109626.6)		(0.0)	(991.7)	(991.7)
7-90-189-19	4450167.9	7.5	0.0	-44.8	-44.8
	(-109626.6)		(0.0)	(991.7)	(991.7)
8-144-170-0	34917696.6	13.8	5.2	5.2	-6.6
	(15161819.5)		(10.4)	(10.4)	(21.6)

Table 4.6: Results for uncertain demand. All values result from out-of-sample evaluations (in-sample size 50, out-of-sample size 1000). HN values are given relative to EEV values in percent, WS values are relative to HN values in percent. Below the OOSS expected values, (relative) CVaR90 values of the resulting distributions are given in parentheses.

ID	EEV^{oos} ($CVaR90$)	WS^{oos}	$HN1.0^{oos}$ ($CVaR90$)	$HN0.5^{oos}$ ($CVaR90$)	$HN0.1^{oos}$ ($CVaR90$)
3-36-28-0	-4810377.3	63.7	0.0	22.4	17.0
	(-16628355.2)		(0.0)	(70.1)	(75.9)
3-38-24-0	-6406458.2	39.1	0.0	3.3	3.3
	(-17874289.8)		(0.0)	(65.2)	(65.2)
3-42-20-0	-2537624.9	0.7	0.0	0.0	0.0
	(-4447493.4)		(0.0)	(0.0)	(0.0)
3-51-22-0	-708650.9	0.0	0.0	0.0	0.0
	(-831075.0)		(0.0)	(0.0)	(0.0)
3-54-46-0	9799843.8	0.0	0.0	0.0	0.0
	(4247790.8)		(0.0)	(0.0)	(0.0)
4-58-125-0	18387494.6	29.4	15.7	14.9	14.9
	(-11418418.0)		(89.8)	(92.0)	(92.0)
4-74-145-0	17870844.3	32.4	16.7	16.7	16.7
	(-13636551.5)		(93.7)	(93.7)	(93.7)
4-80-155-0	18489845.4	32.1	13.5	13.1	13.1
	(-12766332.0)		(89.1)	(93.3)	(93.3)
4-80-155-24	17969668.4	34.6	12.6	12.4	12.4
	(-13249357.5)		(89.3)	(93.5)	(93.5)
5-71-173-0	569595.3	79.7	1155.1	692.0	180.2
	(-56392436.3)		(88.8)	(90.6)	(88.2)
5-106-320-0	-3102765.8	73.2	344.2	258.9	167.0
	(-76479863.6)		(95.4)	(95.1)	(92.2)
6-102-75-0	-20816421.3	63.3	158.9	158.5	158.5
	(-140333995.3)		(93.0)	(95.3)	(95.3)
6-135-87-0	-6166548.5	61.2	323.2	328.9	268.7
	(-110614508.7)		(90.7)	(91.4)	(97.5)
6-142-80-0	-22743274.5	64.2	150.1	148.3	133.1
	(-146428306.7)		(91.7)	(93.5)	(98.9)
6-142-80-13*	-25968172.1	65.2	142.5	*141.1 [0.2]	128.3
	(-149243906.4)		(91.5)	*(93.5)	(98.9)
7-75-154-0	15528137.2	10.2	0.4	0.2	0.2
	(-8851727.0)		(82.1)	(86.4)	(86.4)
7-77-156-0	14485456.0	11.6	0.0	1.3	1.3
	(-11202640.3)		(0.0)	(10.7)	(10.7)
7-77-156-16	15052239.5	9.3	0.8	0.8	0.8
	(-10026021.5)		(37.2)	(37.2)	(37.2)
7-79-188-0	3202879.7	34.2	0.0	0.0	-44.6
	(-7505652.6)		(0.0)	(0.0)	(7.6)
7-90-189-0	1339731.3	140.0	0.0	0.0	-93.6
	(-10417712.2)		(0.0)	(0.0)	(27.5)
7-90-189-19	2027156.8	64.8	-0.0	-0.0	0.0
	(-9423545.2)		(0.0)	(0.0)	(0.0)
8-144-170-0*	22426761.5	9.3	16.4	*14.2 [0.2]	14.6
	(-18476847.6)		(65.5)	*(88.9)	(87.8)

Table 4.7: Results for simultaneous uncertainty of demands and travel times. All values result from out-of-sample evaluations (in-sample size 50, out-of-sample size 1000). HN values are given relative to EEV values in percent, WS values are relative to HN values in percent. Below the OOSS expected values, (relative) CVaR90 values of the resulting distributions are given in parentheses.

ID	*PT1*	*ST1*	*PT50* ($w = 1.0$)	*ST50* ($w = 1.0$)	*PT50* ($w = 0.5$)	*ST50* ($w = 0.5$)	*PT50* ($w = 0.1$)	*ST50* ($w = 0.1$)
3-36-28-0	0.2	0.0	0.3	3.4	0.3	6.0	0.3	7.2
3-38-24-0	0.2	0.0	0.3	2.4	0.3	1.8	0.2	1.8
3-42-20-0	0.2	0.0	0.3	0.8	0.3	1.0	0.3	1.0
3-51-22-0	0.3	0.1	0.5	1.6	0.5	2.0	0.4	2.0
3-54-46-0	0.2	0.1	0.4	5.3	0.4	5.4	0.4	5.5
4-58-125-0	0.6	0.7	1.9	82.0	1.9	124.0	1.9	261.4
4-74-145-0	0.8	1.6	2.6	165.9	2.6	328.6	2.6	295.3
4-80-155-0	0.8	1.4	2.9	204.1	2.9	419.9	2.9	406.0
4-80-155-24	1.0	3.1	12.0	333.9	11.9	1053.4	11.9	1369.6
5-71-173-0	0.5	0.5	1.8	43.5	1.8	61.6	1.8	53.5
5-106-320-0	0.7	0.3	3.8	26.5	3.9	70.9	3.5	49.6
6-102-75-0	1.5	6.4	10.7	1776.0	11.2	2252.4	10.9	2237.8
6-135-87-0	1.7	3.3	13.9	2191.8	13.3	5169.0	13.5	1674.9
6-142-80-0	2.3	9.9	26.2	2369.5	20.6	7270.5	22.2	6519.8
6-142-80-13	2.4	14.0	35.8	1748.8	35.7	5703.3	36.9	7220.5
7-75-154-0	0.6	0.3	3.2	20.2	3.2	37.4	3.2	37.5
7-77-156-0	0.7	0.3	3.3	20.7	3.3	36.3	3.3	37.3
7-77-156-16	0.7	0.4	3.3	22.6	3.3	27.2	3.3	29.3
7-79-188-0	0.8	0.7	4.9	53.0	4.9	272.5	4.9	184.0
7-90-189-0	0.8	0.8	5.3	59.5	5.3	448.6	5.3	355.7
7-90-189-19	0.8	0.7	5.4	42.9	5.3	251.9	5.4	349.7
8-144-170-0	1.8	8.8	16.5	2516.2	16.5	5692.1	16.6	4528.8

Table 4.8: Uncertain travel time: Run times in seconds

4.5.6 Case study

In this section, we analyze the results for the instance 4-58-125-0 in detail to show how solutions change when considering different types of uncertainty. The instance contains four vessels and 58 nodes, and we choose it because it represents a small repositioning in a restricted geographic area that can be not only effectively visualized, but also solved to optimality in little time. Furthermore, the instance shows relative improvements both in terms of the expected profit and CVaR90 over a deterministic approach.

Delay uncertainty

Figure 4.2 depicts the repositioning routes generated by the deterministic and the stochastic model ($\lambda = 1.0$) for the four vessels and two services of instance 4-58-125-0. The routes are presented as time-space graphs, showing the time progress from the start of the repositioning on the x-axis in hours. The y-axis lists the ports. For vessels 1 (blue line) and 2 (red line), the deterministic route equals the stochastic route. For vessels 3 (green line) and 4 (turquoise line), different stochastic solutions are generated, as shown by dashed lines. Instead of sailing from LKCMB directly to INMAA, vessel 4 uses a later time slot at INMAA and takes a detour over MYPKG, which raises the probability of getting to INMAA on time. This leads to a relative improvement of the

ID	*PT1*	*ST1*	*PT50* $(w = 1.0)$	*ST50* $(w = 1.0)$	*PT50* $(w = 0.5)$	*ST50* $(w = 0.5)$	*PT50* $(w = 0.1)$	*ST50* $(w = 0.1)$
3-36-28-0	0.2	0.0	0.4	1.2	0.4	2.4	0.4	2.1
3-38-24-0	0.2	0.0	0.3	0.6	0.3	1.2	0.3	1.0
3-42-20-0	0.2	0.0	0.4	0.5	0.4	0.7	0.4	0.7
3-51-22-0	0.3	0.0	0.6	0.8	0.6	1.2	0.6	1.2
3-54-46-0	0.2	0.0	0.5	1.3	0.5	2.3	0.5	1.9
4-58-125-0	0.6	0.7	2.0	24.4	2.1	41.4	2.1	50.8
4-74-145-0	0.8	1.0	2.7	37.6	2.8	144.0	2.9	90.6
4-80-155-0	0.8	1.9	2.9	45.7	2.9	187.5	3.0	149.1
4-80-155-24	1.1	2.4	13.1	1312.4	12.6	1095.1	13.8	537.6
5-71-173-0	0.6	0.5	2.0	12.8	2.1	17.2	2.1	17.9
5-106-320-0	0.7	0.4	3.8	18.0	3.3	19.5	3.3	20.5
6-102-75-0	1.5	9.1	7.8	79.0	7.8	844.3	7.4	1005.5
6-135-87-0	1.7	3.4	9.3	80.9	9.3	1208.8	9.5	894.4
6-142-80-0	2.4	8.3	15.2	171.5	16.9	4043.6	15.1	3537.3
6-142-80-13	2.7	22.8	33.1	1305.7	44.6	7200.8	35.0	7200.5
7-75-154-0	0.6	0.3	3.1	10.6	3.0	16.4	3.0	14.2
7-77-156-0	0.7	0.3	3.1	11.2	3.1	15.9	3.1	16.0
7-77-156-16	0.6	0.3	3.1	11.3	3.2	23.0	3.1	32.0
7-79-188-0	0.8	0.7	4.4	27.6	4.5	50.1	4.5	61.4
7-90-189-0	0.8	0.7	4.9	24.5	4.9	56.0	4.9	47.9
7-90-189-19	0.8	0.8	4.9	35.6	4.9	79.6	4.9	64.1
8-144-170-0	1.9	8.4	11.1	241.1	11.0	5103.3	11.0	4168.6

Table 4.9: Uncertain demand: Run times in seconds

expected profit of 5.5% due to lower penalty costs. The arc from LKCMB to INMAA has an average delay of 5.5 hours, compared to average delays of -1.5 and 1.5 hours for the arcs LKCMB to MYPKG and MYPKG to INMAA, respectively.

Including the CVaR in the objective function leads to a more robust solution. The vessel sails on arcs with smaller average delays to have the least possible sailing and port costs. The average delay for the stochastic solution ($\lambda = 1.0$) is 4.8 hours, whereas in the stochastic solutions with $\lambda = 0.5$ and $\lambda = 0.1$ the average delay is reduced to zero. This also reduces the average penalty cost over all scenarios from about 1.1 million to zero. In contrast, the sailing cost and port fees increase by 0.6 and 0.02 million, respectively. The revenue gained from container transport decreases by 0.6 million for the CVaR optimization ($\lambda = 0.5$ and $\lambda = 0.1$). Those changes result in a more robust, but slightly less profitable solution. Table 4.5 shows that the expected CVaR90 raises by almost 230%, whereas the expected value of the stochastic model solution decreases by 0.4%.

Demand uncertainty

When comparing the deterministic and stochastic solutions considering only demand uncertainty, the routes for vessel 1 (blue) and vessel 2 (red) change. The first vessel has additional visits at the ports CNNGB and HKHKG in the stochastic solution. On average, an additional 547 TEUs are carried. The second vessel includes the port

ID	*PT1*	*ST1*	*PT50* ($w = 1.0$)	*ST50* ($w = 1.0$)	*PT50* ($w = 0.5$)	*ST50* ($w = 0.5$)	*PT50* ($w = 0.1$)	*ST50* ($w = 0.1$)
3-36-28-0	0.2	0.0	0.3	5.5	0.3	3.7	0.3	3.9
3-38-24-0	0.2	0.0	0.3	3.1	0.2	3.0	0.2	2.9
3-42-20-0	0.2	0.0	0.3	0.8	0.3	0.9	0.3	0.9
3-51-22-0	0.3	0.0	0.4	1.3	0.5	1.6	0.4	1.4
3-54-46-0	0.2	0.1	0.4	2.8	0.4	7.6	0.4	3.4
4-58-125-0	0.7	0.9	2.0	94.4	2.0	250.8	2.0	200.0
4-74-145-0	0.8	0.7	2.6	156.0	2.6	340.4	2.6	488.4
4-80-155-0	0.9	0.8	2.9	273.3	2.8	502.1	2.8	556.2
4-80-155-24	1.0	1.7	12.3	140.4	12.3	1250.7	11.9	1129.6
5-71-173-0	0.5	0.5	1.7	100.7	1.7	75.6	1.7	56.0
5-106-320-0	0.7	0.3	3.2	40.9	3.2	49.9	3.2	44.6
6-102-75-0	1.4	3.8	9.4	1519.8	9.4	1519.3	9.4	2578.2
6-135-87-0	1.7	3.0	11.5	1105.2	11.5	2171.0	11.5	2959.1
6-142-80-0	2.2	3.7	19.2	1586.7	19.2	3147.9	19.2	2486.6
6-142-80-13	2.5	7.3	38.7	4977.0	36.6	7225.4	36.9	5363.5
7-75-154-0	0.7	0.3	3.3	34.6	3.3	31.6	3.4	34.6
7-77-156-0	0.7	0.3	3.3	21.4	3.3	50.6	3.3	37.2
7-77-156-16	0.7	0.3	3.3	21.4	3.3	31.4	3.3	31.6
7-79-188-0	0.8	0.6	4.9	52.2	4.9	205.9	5.1	247.4
7-90-189-0	0.9	0.7	5.3	63.6	5.4	302.8	5.4	457.2
7-90-189-19	0.9	0.7	5.4	40.1	5.4	242.3	5.5	391.7
8-144-170-0	2.0	17.2	17.3	3207.7	16.8	7218.6	18.0	5247.4

Table 4.10: Uncertain travel time/demand: Run times in seconds

KRPUS in its route, which allows it to carry a demand originating in CNSHA and ending in KRPUS. In more than one third of the scenarios, there is a lucrative demand between CNSHA and KRPUS ports with an average of 380 TEUs. The overall expected value of the stochastic solution ($\lambda = 1.0$) increases by 0.4% compared to the EEV as shown in Table 4.6. Comparing the cost factors of the deterministic and stochastic solution, we observe that sailing cost and additional port fees raise by 0.1 million compared to an extra revenue of 0.3 million earned by transporting cargo. Furthermore, using the CVaR90 objective function results in a shorter repositioning route that saves 1.2 million in sailing cost and port fees. As a consequence, the revenue for transported cargo decreases by 1.8 million.

Combined travel time and demand uncertainty

For the simultaneous consideration of demand and travel time uncertainty, the relative improvement of the stochastic solution is most significant, although the generated routes are identical to the ones for the exclusive consideration of delay uncertainty. This results from the high influence of uncertain travel times. As delay penalty costs are high compared to the possible additional revenue gained by transporting extra cargo, the solution avoids including extra port and sailing costs and focuses on on-time arrivals. Table 4.7 shows a relative improvement of the stochastic solution ($\lambda = 1.0$) of 15.7% compared to the EEV solution. In contrast, a relative improvement of only 5.5%

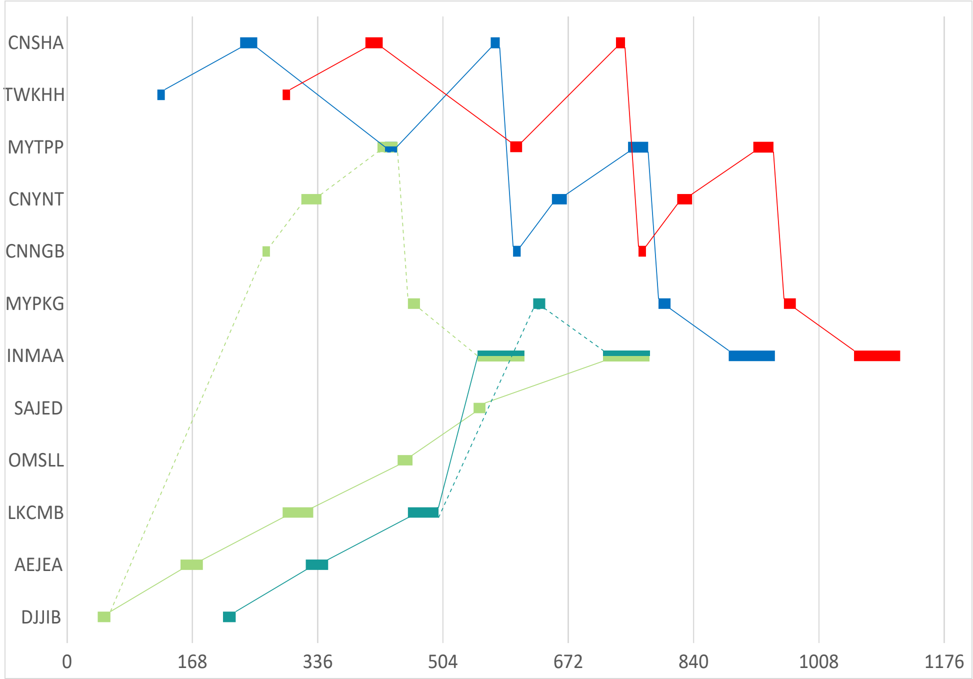


Figure 4.2: Comparison of routes of four vessels (blue, red, green, turquoise) of deterministic (solid line) and stochastic (dashed line) solutions for instance 4-58-125-0 assuming uncertain travel times in a time-space graph with visits as rectangles; the x axis is time in hours.

is achieved when considering just uncertain travel times. The average penalty cost of the stochastic solution ($\lambda = 1.0$) for the individual and combined approach differ by 0.5 million, compared to a difference of 1.8 million for the EV solution.

The solutions become more robust when focusing on CVaR90. For the stochastic model with $\lambda = 0.5$ and $\lambda = 0.1$, the average delay is lowered to zero, compared to 7 hours of average delay resulting from the stochastic model with pure optimization of the expected profit ($\lambda = 1.0$). This reduces the penalty cost to zero as well, which partly explains the relative improvement of the CVaR90 of 92.2% and the decrease of the expected value by 0.8% as shown in Table 4.7. Another reason is the detours taken by vessels 3 and 4 shown in Figure 4.2, where arcs with a lower chance of delays are included.

4.6 Conclusion

We presented a stochastic version of the LSFRP to make repositioning plans more robust against uncertain travel times and fluctuating demands. We used in- and out-of-sample evaluations to handle large-scale instances, and we conducted a qualitative analysis for our solution approach. Our instances and scenarios are based on real world data, and show that uncertain travel times and demands lead to repositioning plans differing from the ones based on deterministic data. We also showed that the stochastic model leads to expected values up to ten times better than those resulting from the EV model. Using the CVaR objective function, we further analyzed the behavior of the stochastic model, comparing risk-neutral and risk-averse objectives. Our results highlight that the operational delay can be significantly reduced when focusing on optimizing the CVaR compared to the deterministic approach. Additionally, the overall expected objective value can be increased, including higher profits from container transports when using stochastic optimization. The effect of using the stochastic model formulation strongly depends on the instance, and its added value is best apparent when uncertain travel times are considered.

For future work, we intend to include more types of uncertainty, such as delays in ports and port availability. Furthermore, we will focus on reducing the solution time for these large, stochastic models, so that real-world instances with many vessels can be solved to optimality.

Acknowledgments

This work is partially supported by Deutsche Forschungsgemeinschaft (DFG) grant 346183302. We thank the Paderborn Center for Parallel Computation (PC^2) for the use of the OCuLUS cluster.

5 A Genetic Algorithm for Finding Realistic Sea Routes Considering the Weather

Abstract

The weather has a major impact on the profitability, safety, and environmental sustainability of the routes sailed by seagoing vessels. The prevailing weather strongly influences the course of routes, affecting not only the safety of the crew, but also the fuel consumption and therefore the emissions of the vessel. Effective decision support is required to plan the route and the speed of the vessel considering the forecasted weather. We implement a genetic algorithm to minimize the fuel consumption of a vessel taking into account the two most important influences of weather on a ship: the wind and the waves. Our approach assists route planners in finding cost minimal routes that consider the weather, avoid specified areas, and meet arrival time constraints. Furthermore, it supports ship speed control to avoid areas with weather conditions that would result in high fuel costs or risk the safety of the vessel. The algorithm is evaluated for a variety of instances to show the impact of weather routing on the routes and the fuel and travel time savings that can be achieved with our approach. Including weather into the routing leads to a savings potential of over 10% of the fuel consumption. We show that ignoring the weather when constructing routes can lead to routes that cannot be sailed in practice. Furthermore, we evaluate our algorithm with stochastic weather data to show that it can provide high-quality routes under real conditions even with uncertain weather forecasts.
Keywords: Weather routing, Ship routing, Genetic algorithm, Uncertain weather

5.1 Introduction

Adverse weather conditions pose a significant danger to ships, their crews, passengers, and cargo and represents one of the main causes of delays in the shipping industry (Notteboom, 2006). Since over 90% of the world trade is carried by ships (Hoffmann and Sirimanne, 2017), considering the weather when finding safe and efficient routes for ships is of utmost importance. Furthermore, finding good routes for ships lowers fuel consumption, leading to lower CO_2 emissions. Finding routes that are not only safe and efficient, but also ensure that the ship arrives on time is a complex task that is difficult to do by hand, especially when considering the weather.

In this paper, we present an approach for finding routes between two points on the globe considering the current weather along the route. For this, we use a real-coded genetic algorithm (GA) with specialized mutation and crossover operators for the weather routing of ships. Our goal is to minimize the fuel consumption of the vessel while respecting a constraint on the latest arrival time at the vessel's destination. Our GA finds realistic, smooth routes that are not restricted to an arbitrary grid and avoid adverse weather conditions, such as strong storms, considering spatial and temporal aspects of the weather. Our approach contains the following novel components:

1. We provide a general routing algorithm that does not have preset values for the longitudes/latitudes of the waypoints along the path.
2. The speed of the vessel is variable, allowing for, e.g., slow-steaming.
3. The generated routes avoid sharp changes in direction to ensure they can be sailed by even large vessels.

We test our algorithm with instances based on many different geographies and weather conditions. Our results show that the minimization of fuel consumption leads to routes avoiding areas with adverse weather conditions, as weather and sea conditions largely effect the different factors travel time, safety and fuel consumption simultaneously. Due to the high bunker consumption of container vessels, even small percentage improvements lead to high cost savings. However, it is also possible to sail through areas with favorable weather conditions to increase the speed of the vessel. The weather data we use in our algorithm is provided by an industrial partner so that we have reliable, real-life information. Overall, the different instances show a wide range of possible savings potentials of up to 13.9% over ignoring the weather. Furthermore, our results also show that the consideration of weather conditions can have a high impact on the route depending on the weather intensity. Our algorithm is also faster than existing approaches, requiring only a single minute of CPU time even for large instances. For smaller instances, the runtime falls to under 30 seconds.

This paper is organized as follows. The weather routing problem is described in detail in Section 5.2. In Section 5.3, the main algorithms for weather-dependent routing of vessels are summarized. Our genetic algorithm is presented in Section 5.4. In Section 5.5, we show computational results for the algorithm for different instances. Section 5.6 summarizes the paper.

5.2 Weather Routing of Ships

Consider a decision maker who wants to determine a route for a vessel over the ocean between two ports. The route clearly must avoid landmasses, shallows, and any other undesirable area as specified by a planner (e.g., pirate zones, or Sulphur Emission Control Areas (SECA)). SECA zones are areas in which the emission of sulphur and sulphur oxides by ships is restricted by law. Planners may also specify a maximum time duration (due date) for the voyage.

We allow vessels to vary their speed along their voyage, thus parameters related to the fuel consumption and impact of weather on the speed of the vessel (draft, displaced volume, waterline length, block coefficient and design speed; the design speed is a speed of the vessel for which the fuel consumption is known.) must be determined. The vessel also has a minimum and a maximum speed between which the speed can be chosen throughout the route. Typically, the problem of weather routing is modeled as a minimum time problem or a minimum fuel consumption problem subject to some constraints (avoiding unfavorable weather, etc.). We solve the problem as a minimum fuel consumption problem, since in the variable speed setting, minimizing only the time leads to expensive solutions with high CO_2 outputs. In this work, we use a model of speed consumption that takes into account wind and waves. It is possible to use a more sophisticated model for the calculation of the fuel consumption, but we note that this has no impact on the algorithmic aspects of our approach.

5.2.1 Computation of the Fuel Consumption

Accurately representing the fuel consumption of vessels is critical for making realistic weather-dependent routes. We use the formula

$$F(v) = (v/v^*)^3 \cdot f^* \tag{5.1}$$

from Brouer et al (2013b) to compute the fuel consumption (subsequently referred to as the *bunker consumption*), where v is the speed of the vessel in knots, v^* is the design speed and f^* is the fuel consumption at the design speed. We split the route into n segments each with a fixed speed v_i. The bunker consumption of the whole route is then calculated by summing the consumption of each segments.

The weather has a strong influence on the *realized speed* of the vessel. We calculate a speed loss coefficient that accepts the parameters of the vessel and the current weather to determine the actual speed. This speed is determined by the direction and speed of the wind and the direction and height of the waves. We use the approach from Larsson and Simonsen (2014) to calculate the speed loss that takes into account both waves and wind simultaneously. The percentage of speed loss is calculated with the following formula (Kwon, 2008):

$$v_{loss} = \alpha_{corr} \cdot \mu_{red} \cdot \frac{\Delta v}{v} 100\%, \tag{5.2}$$

where v_{loss} is the speed loss in percent, α_{corr} is the correction factor for the block coefficient of the vessel and μ_{red} is the weather direction reduction factor. The block coefficient of a vessel is the ratio of the underwater volume of the ship to the volume of a rectangular block having the same length, breadth and depth. The factor α_{corr} depends on the block coefficient of the vessel and its loading condition. A full list of formulas for the calculation of α_{corr} for all different combinations of block coefficients and loading conditions can be found in Larsson and Simonsen (2014). The parameter μ_{red} changes with the intensity of the weather and the angle of the wind/waves with respect to the ship. The speed loss in head weather is given as $\frac{\Delta v}{v}100\%$. The speed loss is represented with Δv and the planned speed with v. The ratio is calculated by one of

$$\frac{\Delta v}{v}100\% = 0.5BN + \frac{BN^{6.5}}{2.7\nabla^{\frac{2}{3}}}, \tag{5.3}$$

$$\frac{\Delta v}{v}100\% = 0.7BN + \frac{BN^{6.5}}{2.7\nabla^{\frac{2}{3}}}, \text{ or} \tag{5.4}$$

$$\frac{\Delta v}{v}100\% = 0.5BN + \frac{BN^{6.5}}{22\nabla^{\frac{2}{3}}} \tag{5.5}$$

depending on the loading condition of the vessel. The parameter BN is the Beaufort number and ∇ is the displaced volume of the vessel. Equation 5.3 is used for vessels in laden condition and Equation 5.4 for vessels in ballast condition. We use Equation 5.5 for our experiments as this is the one for container ships and vessels in normal condition. The Beaufort number characterizes the strength of the waves and the wind.

5.3 Related Work

Existing approaches in the literature for ship weather routing can be divided into three groups: exact approaches, single objective heuristic approaches, and multiobjective evolutionary approaches. Exact approaches to solve the problem are especially found in early works, whereas recent approaches are mainly heuristics. The following overview is divided into the three groups of approaches and summarizes the most important ones. We further identify differing objective functions for the approaches. Touati and Jost (2012) partitions the objective functions found in the ship routing literature into three groups: economic, climate/sustainability and regional fairness/health. The most frequent objective functions are the minimization of fuel consumption and distance traveled. The optimization of fuel consumption is also often combined with the minimization of risk related to a route, resulting in a multiobjective optimization problem. In the last part of this section, we present some papers related to the problem of weather routing.

5.3.1 Exact Approaches

One of the first approaches for ship weather routing is the isochrone method (James, 1957). It minimizes the travel time of the ship and allows the manual construction of a route. This method is improved by Hagiwara and Spaans (1987), who make it suitable for a computer and include fuel consumption in addition to travel time in the objective function. Another approach for the problem is the calculus of variations method proposed by Haltiner et al (1962). It is an analytic approach to weather routing that determines the path and the engine power of the ship. There are also many approaches using dynamic programming such as De Wit (1990) and Motte and

Calvert (1990) or based on shortest path algorithms such as Montes (2005), Panigrahi et al (2012), Sen and Padhy (2015) and Mannarini et al (2016).

5.3.2 Single Objective Heuristic Approaches

Evolutionary approaches are well suited algorithms for solving path finding problems because they allow the inclusion of a wide range of constraints and objectives into problems that are hard to solve with exact algorithms. The optimization of only one objective makes these approaches additionally fast compared to the multiobjective ones. Walther et al (2018) propose a genetic algorithm (GA) that supports variable ship speed and minimizes fuel costs. They compare a graph algorithm for a ship weather routing problem with a GA. The exact details of the GA are not clear, however to the best of our knowledge, our GA differs in its domain specific heuristics and variable length encoding.

Wang et al (2018a) propose a real-coded GA with fixed longitudes for the waypoints that make up the solution representation. This approach therefore has a problem finding paths around vertical obstacles, meaning circumnavigating landmasses can be very difficult or even impossible if the path must traverse a high range of latitude in only a small range of longitude. They integrate further constraints in their optimization including a restriction of the sailing area, avoidance of land obstacles and shallow waters, an interval for the ship's speed and weather alarm zones that are caused by severe weather conditions and wave heights exceeding a certain value. Furthermore, they introduce a general mathematical model for the problem. Yuankui et al (2014) introduce a simulated annealing algorithm minimizing the travel time of the vessel.

5.3.3 Multiobjective Evolutionary Approaches

Multiobjective evolutionary approaches allow for the simultaneous optimization of several objectives. Evolutionary algorithms (EA) are the most common multiobjective solution procedures for weather routing, such as the work of Tsou (2010) and Azariadis (2017) who combine an EA with a modified A* algorithm. The A* algorithm is used to compute one-third of the initial population, and the remaining two-thirds are computed randomly. Tsou (2010) calculates routes avoiding obstacles that are given to the evolutionary algorithm as the first population. The EA then tries to find the optimal route incorporating the safety and economy of routes.

Li and Zhang (2017) propose a GA that minimizes the turning variation and fuel consumption of vessels while considering fixed orientations of the vessel at the start and end position for the vessel as their approach is used to find the optimal trajectory in close-range maneuvering. The real-coded GA presented by Maki et al (2011) supports variable ship speed by varying the propeller revolutions for each segment of the route. However, the longitudes of the route points are fixed and only the latitudes can be changed, severely limiting the possibilities of the algorithm. This leads to the same problems as the algorithm from Wang et al (2018a). Furthermore, the authors do not mention how to avoid obstacles (land masses, etc.) in the algorithm. Tsou and Cheng (2013) implement an ant colony algorithm that is combined with the crossover and mutation concept of GAs for finding routes while considering weather conditions. If ants pass the same waypoint, a crossover of both routes is conducted.

Krata and Szlapczynska (2012), Szłapczynska and Smierzchalski (2009), Szlapczynska (2013) and Szlapczynska (2015) use a multicriteria weather routing algorithm based on the concept of the Multi-objective Evolutionary Weather Routing Algorithm proposed by Szłapczyńska (2007) to solve the problem of finding routes taking into account changeable weather conditions. All these approaches focus on minimizing the passage time, fuel consumption and the risk factor of routes. The constraints in Szlapczynska (2015) include regions to avoid and the variables of the algorithm are the coordinates of the waypoints and the settings of the ship's engine. There is no information given about how the mutation and crossover in this algorithm are performed and therefore the quality of the algorithm cannot be evaluated.

Li et al (2017), Veneti et al (2015) and Veneti et al (2018) apply the NSGA II approach. Veneti et al (2018) compare it to an implementation of the SPEA (Zitzler and Thiele, 1999). Veneti et al (2015) define a non-linear integer programming problem and present a modified version of the NSGA II (Deb et al, 2002). The algorithm uses nodes in a grid to find an optimal path from the origin to the destination. The velocity of the ship is fixed, meaning adverse weather can only be avoided through route changes. Vettor and Guedes Soares (2016) perform their search only with the Strength Pareto Evolutionary Algorithm 2 from Zitzler et al (2001) to minimize the fuel consumption, time of arrival and risk related to rough weather. The optimization of the initial routes is done with a version of the grid-based Dijkstra's algorithm and the speed may differ between two waypoints. Within the evolutionary algorithm the mutation only changes one waypoint at once strongly restricting the changes within one iteration. To select the most favorable route the hyperplane strategy distance method also presented by Vettor and Guedes Soares (2016) is used. It is used to select a solution out of a Pareto set according to user preferences specifying the importance of the different objectives.

5.3.4 Related Problems

Several problems exist that are similar to the vessel weather routing problem. One of them is the routing of planes instead of vessels. The basis of the Flight Planning Problem (FPP) as presented by Knudsen et al (2018) is a directed graph with nodes representing waypoints at different altitudes. The arcs are associated with a resource consumption and costs. These costs depend on the fuel consumed so far and on the weather conditions that depend on the time when the arc is traversed. Therefore, in contrast to our approach, the problem is solved with the help of a graph structure. The variant of the problem presented by Knudsen et al (2017) assume a fixed altitude for the flight making the problem similar to ours. Normally, there are different flight levels and therefore a third dimension for each waypoint. Furthermore, variations in speed are often not considered in the FPP.

Shortest path problems over roads are also related to our problem (see Madkour et al (2017) for a survey). The key difference comes in the freedom of movement at sea as well as the varying speed of the ship. Although cars and trucks can vary their speed while driving, considering this in the planning phase is difficult since crowded roads and tight deadlines prevent speed variation in practice. Furthermore, the weather has much less impact on the quality of solutions and is therefore not included in approaches found in literature. One approach to solve these shortest path problems is the use of contraction hierarchies as presented by Geisberger et al (2008) and extended in Geisberger et al (2012). This approach is designed for road network specific structures where the visited points are directly linked, in contrast to the situation on oceans where the connections of the ports are flexible.

Another area of related problems is the path planning of Unmanned Aerial Vehicles (UAVs). Arantes et al (2016), for example, use a genetic algorithm (GA) to plan the paths of drones and mention no-fly zones and obstacles that are similar to the land masses in the weather routing problem. They include position uncertainty of the aircraft due to turbulence, which does not occur in the weather routing problem. The multi-population genetic algorithm is combined with a visibility graph maintaining all feasible paths for the drone, which would not be possible for the weather routing problem because of the much larger solution space. Hasircioglu et al (2008) also use a GA to plan paths offline for UAVs. The GA uses three different mutation operators that update, insert and delete control points visited by the UAV. Hence, there are no operators that are specific to the case of UAV path planning. In contrast to this, we are using customized operators involving the weather and the speed of the vessel when changing a solution to achieve better improvements than with general operators. Ragusa et al (2017) also investigate a GA for “micro aerial vehicles”. The algorithm is similar to the approach presented by this paper for finding routes for an initial population for our GA. However, for the problem of weather routing, intersections are infeasible, which is different than in the approach of Ragusa et al (2017). In their approach, intersections are allowed and the algorithm focuses on minimizing the degree of intersection with obstacles. Furthermore, the routes used by Ragusa et al (2017) have fewer waypoints than in the case of ship routing, because the routes are much shorter and need fewer waypoints.

5.4 A Genetic Algorithm For Realistic Weather-dependent Routes

We use a genetic algorithm (GA) to find weather-dependent routes for vessels. GAs provide an ideal framework for weather routing for a couple of reasons. First, GAs allow us to combine parts of routes in a natural way in the hopes of forming a high quality solution. Second, the population of a GA offers diversity that is important for avoiding local optima that can be induced by large storms or weather systems. Our GA uses a variable length, real-valued solution representation in which each individual is composed of waypoints, each characterized by a longitude, latitude and the planned speed of the vessel on the segment preceding the waypoint. The set A defines the arcs between the waypoints.

We fill the initial population with feasible routes that have random waypoints added to ensure diversity (Subsection 5.4.1). The crossover and mutation operators used within our algorithm are adapted to the weather routing problem and we use multiple operators, unlike in standard GA implementations (Subsections 5.4.2 and 5.4.3). The fitness function used for evaluating each individual of the population is the bunker fuel required to sail the route under the given weather conditions, and the computations for this are given in Section 5.2. We further add penalty costs to the fitness function for sailing through undesirable areas (e.g., areas with pirates) or not meeting temporal constraints. The GA terminates when the improvement between two iterations falls below a threshold value or after a given number of iterations. The objective function used to evaluate a solution is represented as

$$minimize \sum_{a \in A} (c^a \cdot F^{arc}(a)) + p^{pirate} \cdot y^{pirate} + p^{delay} \cdot t^{delay} \tag{5.6}$$

$$\text{where } F^{arc}(a) = \begin{cases} \infty & \text{, if the wave height on arc } a \text{ exceeds } h \\ \frac{F(v_a^{plan})}{24} \cdot \frac{d_a}{v_a^{real}} & \text{, otherwise.} \end{cases}$$

The set of all arcs used to construct the route from the start to the destination is denoted as A. The binary variable y^{pirate} is set to 1 if the route has at least one arc intersecting a pirate zone and the vessel sails slower than 18 knots. Furthermore, t^{delay} specifies the delay of the vessel in days if a time limit for the travel time exists. The penalty for additional safety costs is given by p^{pirate} and the parameter p^{delay} specifies the penalty for not meeting the arrival deadline, if one is specified. The function $F^{arc}(a)$ calculates the fuel consumption for a single arc between two way points of a complete route, where v_a^{plan} is the planned speed, d_a is the length and v_a^{real} specifies the real speed calculated with the loss factor v_a^{loss} for arc a. When the maximum acceptable wave height h is exceeded for an arc, the fuel consumption is set to ∞, because solutions containing such arcs are infeasible and must therefore be ignored. For feasible solutions, the fuel consumption is multiplied by the cost of fuel per ton c^a to determine its cost for the objective function.

We define two types of high cost zones: areas with increased risk of pirate attacks, and the SECA zone in the North Sea. We specify four different pirate zones based on public information about pirate encounters: the Caribbean Sea, the Gulf of Guinea, around Somalia/the Horn of Africa/the Gulf of Bengal, and the South China Sea. In these zones, it is necessary to sail at a speed of at least 18 knots as recommended by

The Baltic and International Maritime Council et al (2011) or to pay additional safety costs of 50,000 USD, as for example suggested by Wrede (2013). Therefore, p^{pirate} is set to this value for our calculations. In the SECA zone, we assume more expensive fuel is used and adjust the bunker consumption function of the vessel accordingly[1]. The cost factor c^a is set to 450 USD except in the SECA zone where it is 850 USD. The penalty p^{delay} for arriving too late is set to 25,000 USD and is calculated by multiplying a penalty of 100 USD per container per day, as suggested by Li et al (2015), with an assumed number of transported containers of 2500, which is the half of the capacity of the vessel we use. The parameter h is set to 9m corresponding to a Beaufort number of 10 for our calculations.

5.4.1 Initial Route Generation

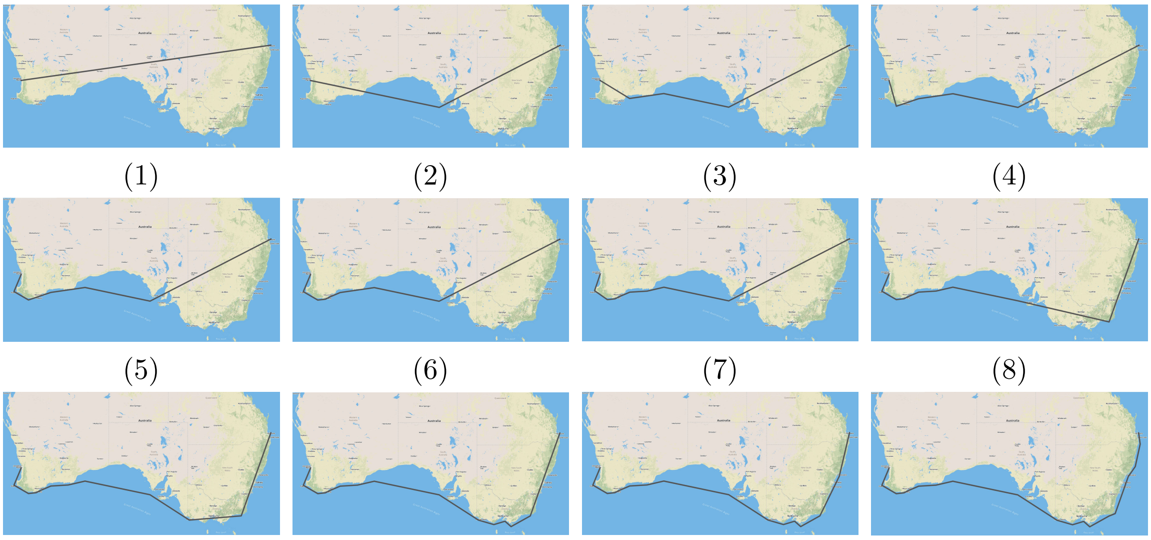
Our initial solution algorithm creates a route by first directly connecting the origin and destination and iteratively moving the midpoint of the line segments over land orthogonally into the water until no segments intersect with the land anymore. Note that a buffer zone could also be specified around land to ensure that ships travel further offshore. More specifically, a segment is divided into two parts by inserting and moving its midpoint. This is done until the distance between the start and end of a new segment falls below a set threshold, and therefore does not need be divided anymore. This process is visualized in Figure 5.1. The left half of the route is arbitrarily chosen to be moved into the water before the right half (note that the order does not matter), leading to the evolution of a route as shown. To generate individuals for the initial population, this algorithm is used to find a route that has to visit one random point between the starting and the target point. The pseudocode for this algorithm is given in Algorithm 2. The algorithm accepts a start node s, an end node e, a threshold t for the length of segments and a distance d for the movement of middle points. The algorithm returns the node sequence from the start node to the end node, with the end node missing, which therefore still needs to be added. We use this algorithm to generate some initial solutions and then use the crossover and mutation procedures to expand the initial population so that the first generation has the full population size. Start and end points of connections generated with this algorithm can be chosen arbitrarily, however, we only use ports for our computational experiments.

5.4.2 Crossover Operators

In this step of the GA, the routes of two random individuals from the current population are combined into a single, new route. The selection of the individuals is based on a roulette wheel selection with quadratic fitness scaling. We use two different crossover methods. The first combines the routes by choosing a point to join the routes somewhere

---

[1] With the adoption of scrubbers and low/no-sulpher engine technologies, the SECA zone may not be relevant to all ships. It can be easily ignored in the GA when this is the case.



(1) (2) (3) (4)

(5) (6) (7) (8)

(9) (10) (11) (12)

Figure 5.1: Visualization of the generation of an initial route from Perth, Australia to Brisbane, Australia in 12 steps.

near the middle of each route, preferably at a position in which both routes are near each other. This operator is similar to the one from Vettor and Guedes Soares (2016) who combine the first k waypoints of a route with the waypoints of a second route from $k+1$ to the end, but do not search for a waypoint in the middle of the route first. The second combines the routes using a random position from the first half of one route and a random position from the second half of the other route. Veneti et al (2015) and Veneti et al (2018) also use a crossover operator combining routes at random points, but it requires the same node to be contained in both parent routes to perform a recombination, meaning it is rarely applicable. Furthermore, for both of our operators we post-process the route to ensure that it is completely located in the water using the same procedure as for the initial routes. The result of one of our crossovers is visualized in Figure 5.2 by means of the website `geojson.io`, in which the two black routes are combined to obtain the red one. We use the great circle distance[2] for the length of all routes presented in this paper, but in the visualizations the way points are connected with straight lines for ease of visualization.

---

[2] The great circle distance specifies the shortest distance between two points on the surface of a sphere. The distance of two points on the earth is calculated by $d = 2r \cdot \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2-\phi_1}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\lambda_2-\lambda_1}{2}\right)}\right)$, where ϕ_1 and ϕ_2 are the latitudes, λ_1 and λ_2 are the longitudes of the two points and r is the radius of the earth.

Algorithm 2 Initial Route Generation

```
function IRG($s, e, t, d$)
    $N \leftarrow \emptyset$                                   ▷ Sequence of nodes along path
    if DISTANCE($s, e$) $> t$ then
        $m \leftarrow$ MIDPOINT($s, e$)
        $v \leftarrow$ vector orthogonal to the connection $s$ to $e$ with length 1
        $i \leftarrow 0$
        while $m$ is over land do
            for $j \in \{1, -1\}$ do
                $m' \leftarrow m + v \cdot i \cdot d \cdot j$
                if $m'$ is in water then $m \leftarrow m'$
            $i \leftarrow i + 1$
        $N \leftarrow N \cup$ IRG($s, m, t, d$) $\cup$ IRG($m, e, t, d$)
    else
        $N \leftarrow N \cup \{s\}$
    return $N$
```

5.4.3 Mutation Operators

The mutation operators are used to make different changes to random individuals of the population. These operators delete or move waypoints or change the speed at waypoints and allow the GA to search for new solutions that are not just combinations of existing individuals. We use several different operators to introduce a domain-specific heuristic for dealing with weather conditions and obtain realizable routes. The operator applied to an individual is selected uniformly at random from the list of nine operators.

Deleting a single point/deleting multiple points We first introduce two simple mutation operators that delete a single point or multiple points within a given interval, respectively. The now disconnected parts of the route are reconnected with the initial route algorithm to avoid any landmasses.

Moving a single point/moving multiple points Our second set of operators tries to move one point or an interval of a route by a random distance limited by a parameter in a random direction. Should any part of the route now intersect with land we use the same procedure as in the initial route algorithm to repair the route.

Moving a point with the maximum wind This operator takes the current weather on the route into account. In the first step the position along the route with the maximum influence of the wind on the vessel is determined. In the second step this waypoint is then moved in a randomly determined direction. Afterwards, the route points around the moved one are recalculated with the initial route algorithm to smoothen the route.

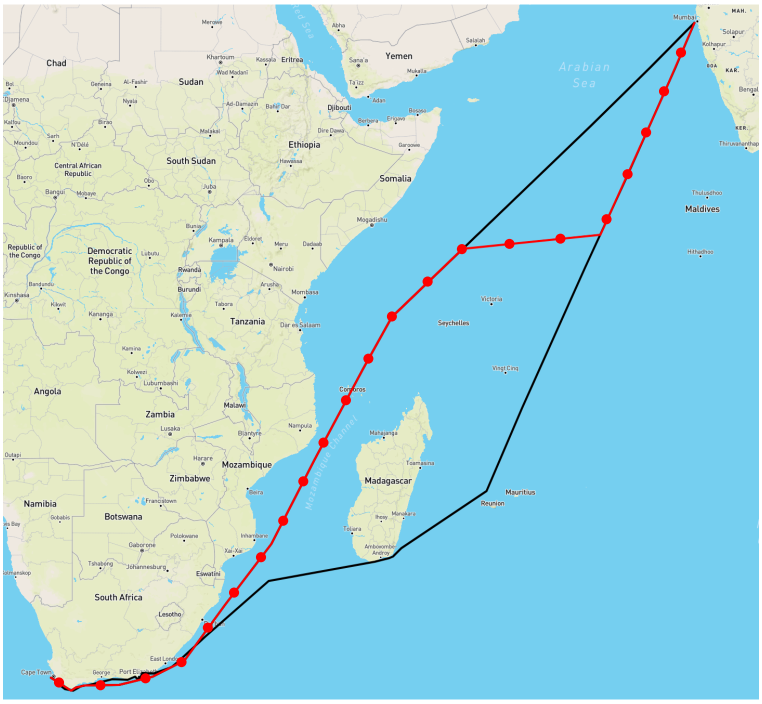


Figure 5.2: Visualization of the crossover operator.

Moving a point with the maximum angle/Moving multiple points with the maximum angle Routes with sharp angles are difficult to realize in practice, as the turning radius of large vessels is limited. This operator thus tries to remove points or multiple points with large angles from the route. We select the point with the maximum angle and remove a given percentage defined by a parameter of all nodes in each direction. Afterwards, the initial route algorithm is used to adjust the route should it end up over land. This process can be repeated for multiple points with large angles. This leads to the second operator of this group that moves the points around the several largest angles with one application.

Mutating the speed of the vessel Our final operator adjusts the speed of the vessel for a segment of the route. The new speed is randomly determined in the interval between 80% and 120% of the current speed in the segment.

5.4.4 Stochastic Optimization

We adjust the objective function to allow the GA to handle stochastic weather. Thus, instead of optimizing the fuel consumption for only a single weather scenario, we evaluate a route given multiple scenarios, giving us an expected fuel consumption. The multiple weather scenarios are forecasts for the true weather and are generated by modifying the historical weather data. The generation of scenarios is discussed in detail in Section 5.5.5. We note that this could be adjusted by risk averse users to use, e.g., conditional value at risk or other functions.

5.5 Computational Results

In this section, we present computational results for our approach tested on 15 instances that we have constructed. The scenarios model various times of departure, starting/ending locations and travel times. We present two different weather routing settings: one with perfect information about the weather data and one where we plan under uncertainty. The first setting shows the performance of our approach under perfect information. In the second case, we show how the algorithm performs in a more realistic situation. We are able to conclude from both settings that including weather is critical for generating realistic routes. We implement the GA in C# and run it on a computer with an Intel Core i7-7700K 4.2 GHz processor and 32GB of RAM. The computation time for the solutions for all instances is less than one minute. Every instance has been run 5 times and we report the average of these runs. Furthermore, we provide the standard deviation of the values for the fuel consumption of the five solutions. We cut the algorithm off at 60 CPU seconds or when the improvements within a single iteration become too small after a minimum number of iterations have been completed. We note that letting the GA run longer than 60 seconds can sometimes lead to slightly improved results on long routes.

5.5.1 Parameters for the GA

Our GA includes a number of parameters that influence the quality of the solutions found. Table 5.1 lists these parameters and the values used. The values have been determined using the GGA algorithm configurator (Ansótegui et al, 2009, 2015). We tune our GA with GGA for 5 days using 45 instances in our training set covering different departure times and locations. We use a Panamax vessel for all of our experiments, but it is also possible to run our algorithm with other vessel types having different vessel-specific parameters.

5.5.2 Weather Data

The weather data used in our experiments is provided by an industrial collaborator. It includes three months of weather information in a $0.5° \times 0.5°$ grid with a 3 hour interval for the time from August to October 2017. For each latitude, longitude, and timestamp, we are provided the wind direction and the wave height.

Parameter	Value
Population size	194
Mutations per iteration	529
Number of chromosomes used for crossover	43
Number of chromosomes used for mutation	34
Crossovers per iteration	40
Number of initial routes	31
Minimal number of iterations	130

Table 5.1: Parameters for the GA

5.5.3 Experimental Results for Weather Data with Perfect Information

We create 15 problem instances and try to include coverage of routes in a variety of locations around the world. Our instances also have a number of different weather conditions, ranging from “normal” inclement weather to Hurricane Irma in the instance from USNYC (New York, USA) to SRPBM (Paramaribo, Suriname). The computational results for all 15 instances using perfect information about the weather (one weather scenario) are listed in Table 5.2. The average of the five runs ignoring the weather is compared to the runs including the weather in terms of route length in nautical miles (nm), duration in days (d) and fuel consumption (FC) in tons (t) of fuel. The results of the tests with and without weather optimization (WO) can be found in the two rows for each instance and the difference of the values is given below each pair of values. The duration and fuel consumption were calculated both with and without the influence of weather to evaluate the quality of the routes leading to the two columns for each of the two key figures. The instances with more extreme weather (Beaufort scale 9 or higher) are marked with a star (*). The solutions for all instances are visualized in Figure 5.5. Weather-optimized routes are colored black and non-weather-optimized routes gray. The map is shaded based on the influence of the weather (dark red is worse weather, meaning stronger winds and higher waves) over the entire time period. The weather is visualized for the points in time when the ship following the weather-optimized route traverses the area or is near the visualized area.

In general, considering the weather leads to less costly routes with shorter travel times. In only one case, the instance from ZACPT (Cape Town, South Africa) to INBOM (Mumbai, India), including the weather leads to a route with less fuel consumption even when evaluating it without weather because the combination of the shorter route west of Madagascar and the strong wind and high waves along that path make it hard for the GA to move its path to the east of Madagascar when generating a route without considering the weather.

For some scenarios there is only a small change in the vessel routing when weather is considered, usually when regions with adverse weather can be easily avoided. However, there are also instances where the weather has a decisive influence on the routing.

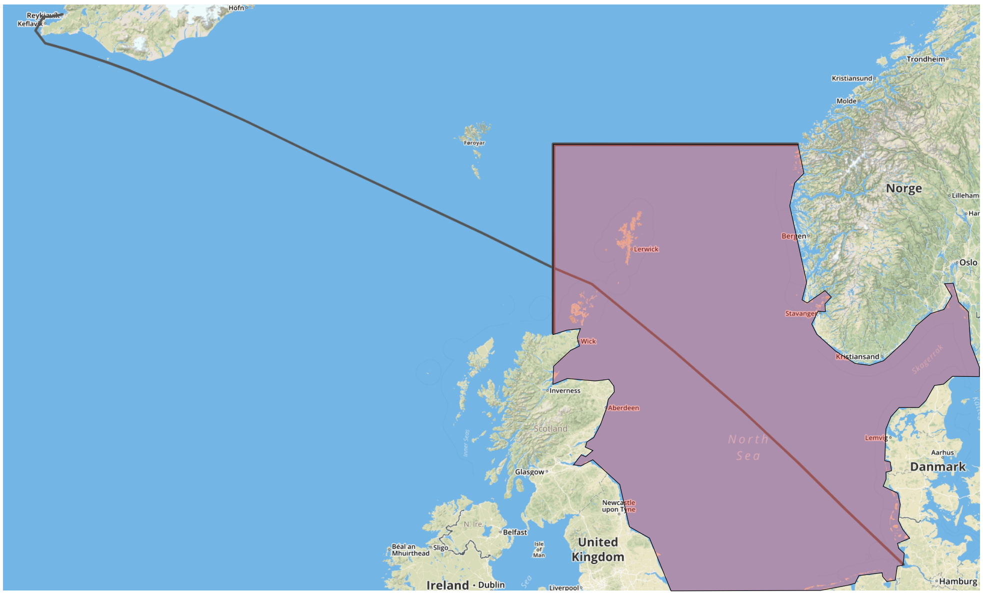


Figure 5.3: Route from Hamburg to Reykjavík and the SECA-zone

The instances from USNYC (New York, USA) to SRPBM (Paramaribo, Suriname) and from AUPER (Perth, Australia) to AUBNE (Brisbane, Australia) show that a path without the influence of weather would lead to routes that are not feasible in reality due to very strong wind and high waves, leading to an infinite value for the fuel consumption and the duration. On the route from New York to Paramaribo we have the strongest weather conditions of all instances (11 on the Beaufort scale). Therefore, the longer routes are necessary here to guarantee safety for the vessel and lead to huge improvements. The instances located in the Mediterranean region (ITTRS to EGALY, ESALG to EGALY, and NLRTM to FRMRS.) show that taking the weather into consideration also makes sense for short/medium distance routes. Our three instances in this region show reductions of the fuel consumption of up to about 3%. The SECA-zone in the North Sea does not have a high impact on the routing in the affected instances because it is not possible to leave it to save costs.

In the instance from DEHAM (Hamburg, Germany) to ISREY (Reykjavík, Iceland) it is not efficient to leave the SECA-zone on a route other than the generated one because the zone reaches far north. The route and the zone are visualized in Figure 5.3. One instance with a relatively small difference between the weather-optimized and the non-weather-optimized path is the connection from USDUT (Dutch Harbor, USA) to USLAX (Los Angeles, USA). Despite this small deviation in the routing, it leads to an improvement in travel time and fuel consumption of approximately 1.6% when comparing the weather-optimized route with the non-weather-optimized route. The weather-optimized route is visualized in Figure 5.5a in black.

The routes running from ZACPT (Cape Town, South Africa) to INBOM (Mumbai, India) show one of the biggest differences as the weather-optimized routes run east of Madagascar while routes ignoring the weather run west of it. This is the case because the western routes are shorter, but the weather to the west is worse than to the east. This is visualized in Figure 5.5d. This leads to an average improvement of fuel consumption by almost 7% considering the weather during optimization. The routes from USNYC (New York, USA) to DEHAM (Hamburg, Germany) show a similar observation with routes running north and south of the United Kingdom depending on the use of weather data. The weather-optimized routes run south of the United Kingdom although this path is longer because the weather conditions in the north are worse than in the south. Another important aspect for the routing in the Arabian Sea is the pirate zone that covers a large part of it. The weather-optimized routes, which already run east of Madagascar, take a right turn to bypass the zone as far as possible, while the routes running west of Madagascar run directly through it. This can be seen in Figure 5.4, which shows the pirate zone. The routes from AEDXB (Dubai, UAE) to AUPER (Perth, Australia) and the routes from MYTPP (Tanjung Pelepas, Malaysia) to OMSLL (Salalah, Oman) are not visibly affected by the zone because the pirate zone covers the complete coast and cannot be avoided. The greatest savings potential can be found for the last instance from JPTYO (Tokyo, Japan) to GUGUM (Guam), in which an approximately 4.3% longer distance leads to savings savings of fuel and travel time of nearly 14%.

Overall, the results for the instances using perfect information about the weather show that there is a huge savings potential when including weather conditions into the optimization of routes for vessels. The improvements range from about 1% to almost 14% excluding the instances for which infeasible routes were generated when not considering weather.

5.5.4 Pirate Zones and Travel Time Limitations

In addition to higher speeds in pirate zones, the objective function can be penalized to account for security services (e.g., armed mercenaries). Increasing the costs within pirate zones by 10%, as well as requiring a higher speed leads to routes similar to the one in Figure 5.4. This route has been generated without considering the influence of weather conditions to show the effect of the increased costs for a higher level of security. Our approach generates a solution that completely avoids the pirate zone marked in orange, and requires no modifications to our heuristics to identify such zones.

Another aspect that can be included into the routing problem is the addition of penalty costs for a delayed arrival at the destination port. Very high penalty costs can be used to force the algorithm to meet the deadline at any cost (if it is at all possible). Therefore, the course of the routes are changed for some instances and the speed on the routes must be increased in most cases.

			Evaluated Duration		Fuel consumption	
		Length	Ignoring Weather	Including Weather	Ignoring Weather	Including Weather
USDUT	Without WO	2374.54	8.25	8.90	128.39	138.49
to	With WO	2376.28	8.26	8.76	128.49	136.29
USLAX	Difference	0.1%	0.1%	-1.6%	0.1%	-1.6%
USNYC	Without WO	2295.97	7.98	∞	124.14	∞
to	With WO	2324.02	8.07	8.74	125.78	136.24
SRPBM*	Difference	1.2%	1.2%	-100%	1.3%	-100%
USNYC	Without WO	3404.30	11.83	14.21	184.07	221.03
to	With WO	3537.17	12.29	13.76	191.26	214.02
DEHAM*	Difference	3.9%	3.9%	-3.2%	3.9%	-3.2%
ZACPT	Without WO	4631.28	14.36	16.63	351.80	396.48
to	With WO	4690.69	14.91	16.86	332.74	368.87
INBOM	Difference	1.3%	3.9%	1.4%	-5.4%	-7.0%
AUPER	Without WO	2494.42	8.67	∞	134.87	∞
to	With WO	2563.17	8.91	9.63	138.59	149.89
AUBNE*	Difference	2.8%	2.8%	-100%	2.8%	-100%
AEDXB	Without WO	4877.69	15.92	17.90	324.28	359.06
to	With WO	4954.76	16.20	17.32	327.34	346.78
AUPER	Difference	1.6%	1.8%	-3.2%	1.0%	-3.4%
MYTPP	Without WO	3172.52	8.57	9.72	314.87	349.00
to	With WO	3188.60	8.60	9.48	316.92	345.24
OMSLL	Difference	0.5%	0.4%	-2.5%	0.7%	-1.1%
BRSSZ	Without WO	3407.19	11.84	13.25	184.23	206.09
to	With WO	3433.05	11.93	12.68	185.68	197.32
ZACPT	Difference	0.8%	0.8%	-4.3%	0.8%	-4.3%
ITTRS	Without WO	1183.40	4.11	4.46	63.99	69.33
to	With WO	1197.44	4.16	4.32	64.75	67.26
EGALY	Difference	1.2%	1.3%	-3.0%	1.2%	-3.0%
ESALG	Without WO	1793.84	6.23	6.49	96.99	100.97
to	With WO	1800.52	6.26	6.37	97.38	99.11
EGALY	Difference	0.4%	0.4%	-1.9%	0.4%	-1.8%
NLRTM	Without WO	2019.50	7.02	7.88	109.20	122.60
to	With WO	2046.40	7.11	7.77	110.65	120.87
FRMRS	Difference	1.3%	1.4%	-1.4%	1.3%	-1.4%
NOOSL	Without WO	555.25	1.93	2.16	30.02	33.55
to	With WO	565.52	1.96	2.06	30.59	32.10
GBFXT	Difference	1.9%	1.8%	-4.3%	1.9%	-4.3%
DEHAM	Without WO	302.63	1.05	1.32	16.36	20.47
to	With WO	314.74	1.09	1.22	17.02	19.00
GBFXT	Difference	4.0%	4.0%	-7.1%	4.0%	-7.2%
DEHAM	Without WO	1173.09	4.08	4.83	63.43	75.08
to	With WO	1176.24	4.09	4.73	63.60	73.62
ISREY*	Difference	0.3%	0.3%	-2.0%	0.3%	-2.0%
JPTYO	Without WO	1329.21	4.62	6.32	71.87	98.31
to	With WO	1386.67	4.82	5.44	74.98	84.60
GUGUM*	Difference	4.3%	4.3%	-14.0%	4.3%	-13.9%

Table 5.2: Comparison of routes computed considering weather to ones computed ignoring the weather regarding length in nautical miles duration in days and fuel consumption in tons, averages over five runs.

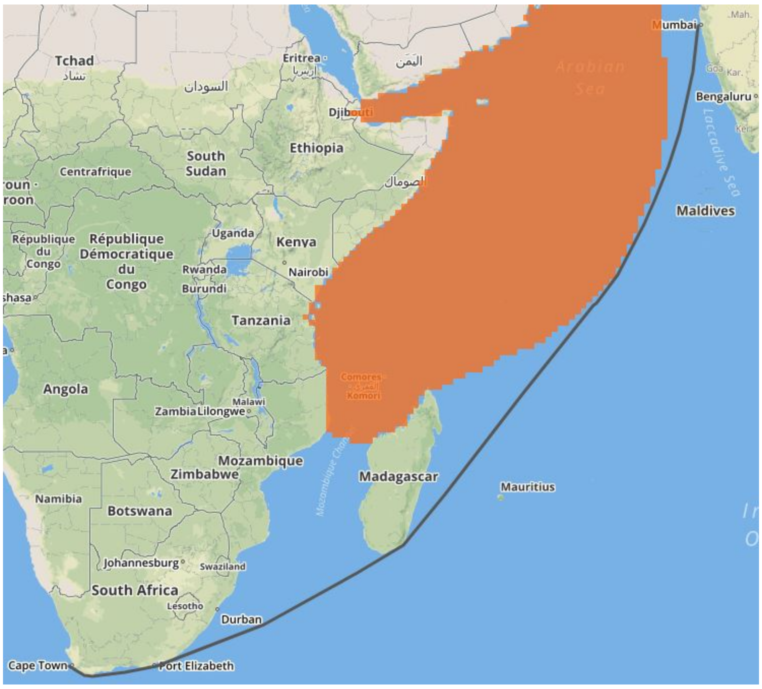
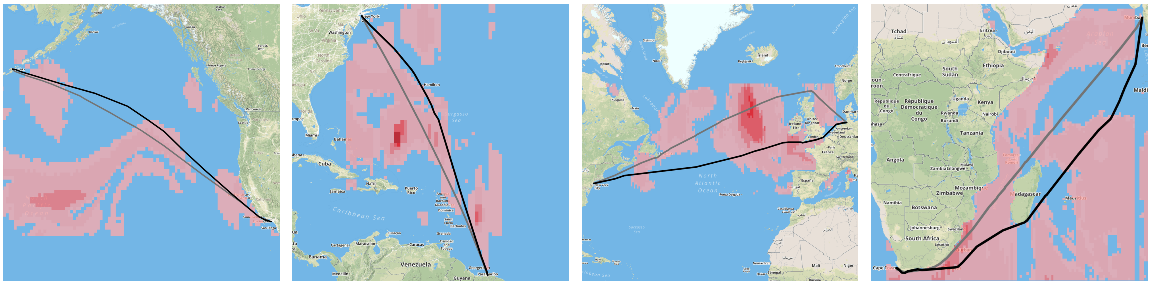


Figure 5.4: Route optimized with additional costs for security

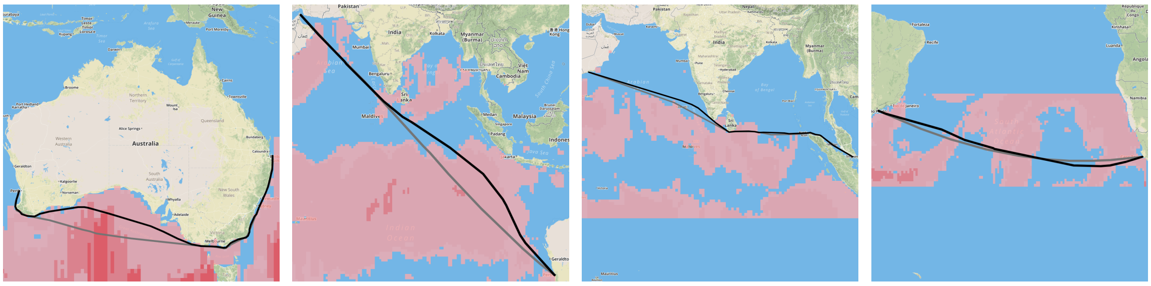
5.5.5 Experimental Results for Stochastic Weather Data

In reality, reliable weather forecasts are only available for the beginning of a route. Thus, we must plan a route under uncertainty and replan whenever a new weather forecast is available. Table 5.3 shows the results for the previously given instances using multiple stochastic weather scenarios instead of planning under certainty for a single scenario as shown previously. The route length, duration and fuel usage are computed using the "true" weather, averaging over five different executions of our approach. The route planning uses either perfect information ("PI"), the expected value over five scenarios with replanning ("Stochastic"), the expected value over five scenarios without replanning ("Stoch. NR"), or plans without considering the weather ("No WO"). Percentage differences to the plan with perfect information are given for each value. Furthermore, the standard deviation for the calculated values for the fuel consumption of the five different solutions is given. Figure 5.6 shows an example for the four different routes for the second instance from USNYC (New York, USA) to SRPBM (Paramaribo, Suriname). The weather is left out for this figure as it differs too much for the different solutions and would be misleading regarding the quality of the routes.

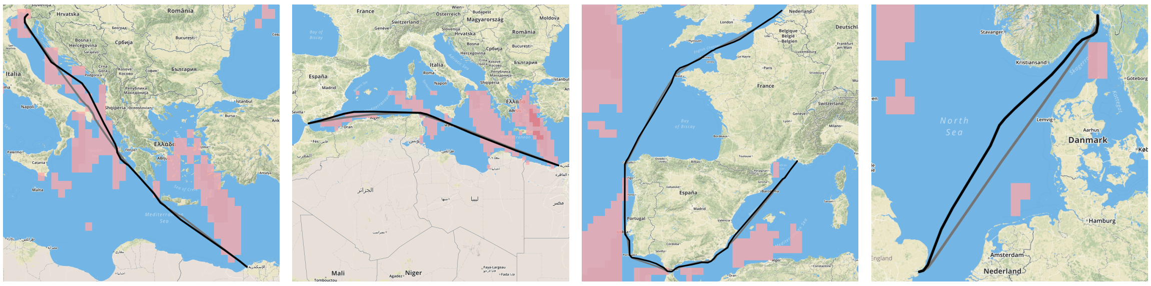
We perform stochastic routing by first generating five potential forecasts of the weather from the "true" weather. This is done by modifying the components (direction, wind speed, wave height) by a random factor between -50% and +50%. We plan a



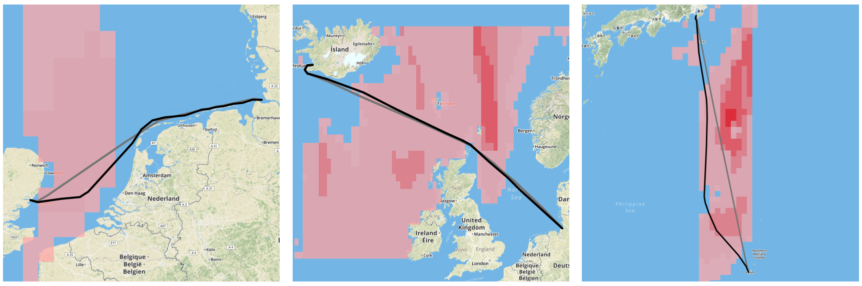
(a) USDUT to USLAX (b) USNYC to SRPBM (c) USNYC to DEHAM (d) ZACPT to INBOM



(e) AUPER to AUBNE (f) AEDXB to AUPER (g) MYTPP to OMSLL (h) BRSSZ to ZACPT



(i) ITTRS to EGALY (j) ESALG to EGALY (k) NLRTM to FRMRS (l) NOOSL to GBFXT



(m) DEHAM to GBFXT (n) DEHAM to ISREY (o) JPTYO to GUGUM

Figure 5.5: Visualization of the solutions for the instances.

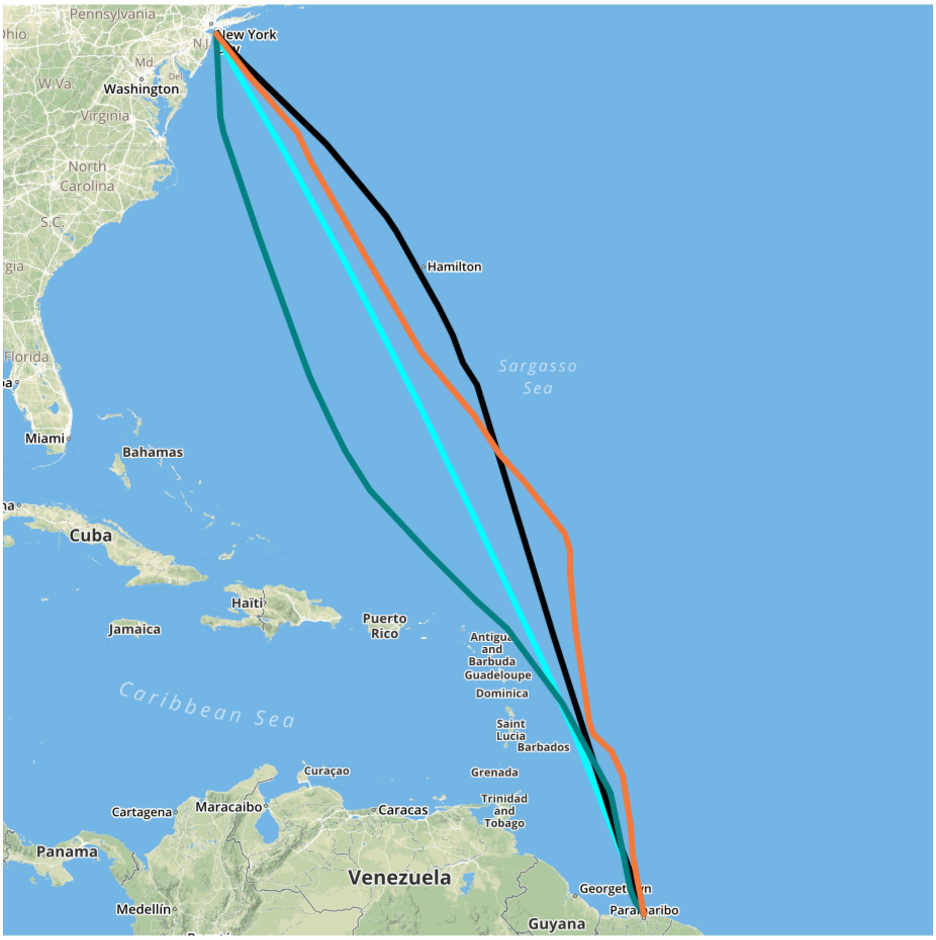


Figure 5.6: The four different routes from USNYC (New York, USA) to SRPBM (Paramaribo, Suriname). "PI" in black, "Stochastic" in orange, "Stoch. NR" in green and "No WO" in blue.

route from the starting location using these five forecasts and minimize the expected value over the scenarios. As the ship moves closer to a given location, the better the forecast becomes. Perfect information is provided for the exact location of the vessel and for all locations within seven days, a forecast is provided for other locations as an average of the "true" weather and randomized forecast weighted by the time needed to arrive at this point[3]. We therefore replan the route in regular intervals. These scenarios are then used to provide the algorithm with information about the possible weather on the route.

Overall, the results presented in Table 5.3 show that the use of stochastic weather data and replanning the route when new data becomes available leads to lower total

---

[3] We note that while we use real, historical weather data, we do not have detailed forecasts for different locations, hence we use the procedure described for stochastic data.

fuel consumption than only planning with stochastic data at the start of the route, or planning using no weather data at all. Our approach is generally only a few percent worse than the solution found planning with perfect information. Replanning the routes during the journey leads to shorter travel times for all instances. For two cases, not replanning the route even leads to infeasible routes (ZACPT to INBOM and AUPER to AUBNE) meaning replanning is essential in these cases. The fuel consumption is also improved and the second instance (USNYC to SRPBM) shows the biggest difference with a deviation of 21.2% (without replanning) compared to a deviation of 9.8% (with replanning). We observe that the best route found is often not much longer or shorter than the best route without replanning.

Comparing the results without the usage of weather data to the results with the usage of stochastic weather shows that there are two special cases. For the instance from ZACPT (Cape Town, South Africa) to INBOM (Mumbai, India), the route has a longer travel time caused by the fact that the fuel consumption is the optimization goal and a longer travel time leads to a lower fuel consumption in this case. The only case where the stochastic results are worse than the results generated without weather is the instance from ESALG (Algeciras, Spain) to EGALY (Alexandria, Egypt), which is due to unstable weather in front of the coast of Egypt.

Using stochastic data leads to longer travel times and higher fuel consumption than using perfect information, but not necessarily to longer routes. For 10 of the 15 instances the routes are longer when using stochastic scenarios instead of the perfect information about the weather. The travel times increase by 0.6% to 9.3% and the fuel consumption increases by 0.6% to 9.8%, respectively. Uncertainties regarding the weather conditions on the remaining route make it difficult for the algorithm to find a route that is good for all weather scenarios.

The best example of difficulty in planning the speed and the route of the ship to avoid strong weather with very high waves is the instance from USNYC (New York, USA) to SRPBM (Paramaribo, Suriname). For this instance, the largest deviation of the stochastic results from the results using perfect information is observed. The uncertain forecasts result in the route partly passing through unfavorable areas. Nevertheless, areas that are impassable due to strong waves and wind are avoided. Furthermore, the replanning of the route has a very high impact on the fuel consumption, showing the importance of the adaption to current weather data.

It is essential to adapt the route during the journey, as is exemplified by the instance from ZACPT (Cape Town, South Africa) to INBOM (Mumbai, India). Here, the route runs east of Madagascar because of the strong weather conditions west of it, which we also observed when using perfect information. The same observation can be made for the instance from AUPER (Perth, Australia) to AUBNE (Brisbane, Australia). However, there are many instances where the replanning does not have a large impact, we note that since replanning is computationally cheap, there is no reason not to do it. Usually, replanning has little impact when there are no large storms or extreme weather along the planned route.

		Length [nm]		Duration [d]		Fuel [t]		
		Avg.	%	Avg.	%	Avg.	%	σ
USDUT to USLAX	PI	2376		8.76		136.29		0.57
	Stochastic	2385	0.4	8.89	1.5	**138.33**	**1.5**	**1.02**
	Stoch. NR	2406	1.2	9.00	2.7	140.01	2.7	0.45
	No WO	2375	-0.1	8.90	1.6	138.49	1.6	0.71
USNYC to SRPBM	PI	2324		8.74		136.24		0.61
	Stochastic	2356	1.4	9.56	9.3	**149.53**	**9.8**	**0.85**
	Stoch. NR	2353	1.3	9.34	6.9	165.07	21.2	5.65
	No WO	2296	-1.2	∞	100	∞	100	0.00
USNYC to DEHAM	PI	3537		13.76		214.02		0.24
	Stochastic	3583	1.3	14.16	2.9	**220.35**	**3.0**	**1.00**
	Stoch. NR	3613	2.2	14.25	3.6	226.78	6.0	2.72
	No WO	3404	-3.8	14.21	3.3	221.03	3.3	2.35
ZACPT to INBOM	PI	4691		16.86		368.87		9.57
	Stochastic	4938	5.3	18.25	8.3	**381.36**	**3.4**	**8.05**
	Stoch. NR	4631	-1.3	15.50	-8.0	∞	100.0	∞
	No WO	4631	-1.3	16.63	-1.4	396.48	7.5	5.43
AUPER to AUBNE	PI	2563		9.63		149.89		0.26
	Stochastic	2524	-1.5	9.82	1.9	**152.71**	**1.9**	**1.17**
	Stoch. NR	2561	-0.1	9.19	-4.6	∞	100	∞
	No WO	2494	-2.7	∞	100	∞	100	0.00
AEDXB to AUPER	PI	4955		17.32		346.78		2.8
	Stochastic	4927	-0.6	17.84	3.0	**354.56**	**2.2**	**3.1**
	Stoch. NR	4888	-1.4	17.25	-0.4	365.56	5.4	2.1
	No WO	4878	-1.6	17.90	3.3	359.06	3.5	3.5
MYTPP to OMSLL	PI	3189		9.48		345.24		1.58
	Stochastic	3195	0.2	9.61	1.5	**348.29**	**0.9**	**1.60**
	Stoch. NR	3225	1.1	9.65	1.8	352.98	2.2	2.18
	No WO	3173	-0.5	9.72	2.6	349.00	1.1	2.70
BRSSZ to ZACPT	PI	3433		12.68		197.32		1.08
	Stochastic	3427	-0.2	13.09	3.3	**203.74**	**3.3**	**3.95**
	Stoch. NR	3428	-0.1	13.14	3.7	204.45	3.6	0.37
	No WO	3407	-0.8	13.25	4.5	206.09	4.4	1.03
ITTRS to EGALY	PI	1197		4.32		67.26		0.11
	Stochastic	1195	-0.2	4.46	3.1	**69.32**	**3.1**	**0.13**
	Stoch. NR	1194	-0.3	4.46	3.1	69.38	3.2	0.21
	No WO	1183	-1.2	4.46	3.1	69.33	3.1	0.37
ESALG to EGALY	PI	1801		6.37		99.11		0.45
	Stochastic	1806	0.3	6.83	7.2	106.23	7.2	0.19
	Stoch. NR	1803	0.1	6.85	7.5	106.50	7.5	0.20
	No WO	1794	-0.4	6.49	1.9	**100.97**	**1.9**	**0.08**

Table 5.3: Comp. results for stochastic optimization with replanning along the route under perfect information (“PI”), stochastic optimization with replanning (“Stochastic”), stochastic optimization with no replanning (“Stoch. NR”), and without weather routing (“No WO”). Percentage gaps relate to perfect information and the standard deviation is given for the fuel.

		Length [nm]		Duration [d]		Fuel [t]		
		Avg.	%	Avg.	%	Avg.	%	σ
NLRTM to FRMRS	PI	2046		7.77		120.87		0.24
	Stochastic	2047	0.0	7.81	0.6	**121.55**	**0.6**	**0.22**
	Stoch. NR	2055	0.4	7.84	0.9	121.94	0.9	0.08
	No WO	2019	-1.3	7.88	1.4	122.60	1.4	0.29
NOOSL to GBFXT	PI	566		2.06		32.10		0.14
	Stochastic	565	-0.2	2.07	0.5	**32.72**	**1.9**	**0.55**
	Stoch. NR	564	-0.3	2.11	2.2	32.75	2.0	0.65
	No WO	555	-1.8	2.16	4.5	33.55	4.5	0.12
DEHAM to GBFXT	PI	315		1.22		19.00		0.10
	Stochastic	318	1.1	1.23	0.7	**19.18**	**0.9**	**0.36**
	Stoch. NR	323	2.8	1.24	1.5	19.37	1.9	0.43
	No WO	303	-3.8	1.32	7.7	20.47	7.7	0.20
DEHAM to ISREY	PI	1176		4.73		73.62		0.11
	Stochastic	1178	0.2	4.78	1.1	**74.39**	**1.0**	**0.55**
	Stoch. NR	1176	0.0	4.78	1.1	74.45	1.1	0.18
	No WO	1173	-0.3	4.83	2.0	75.08	2.0	0.21
JPTYO to GUGUM	PI	1387		5.44		84.60		0.22
	Stochastic	1380	-0.4	5.74	5.5	**89.91**	**6.3**	**3.19**
	Stoch. NR	1409	1.6	5.82	7.0	90.54	7.0	0.41
	No WO	1329	-4.1	6.32	16.2	98.31	16.2	0.77

Table 5.3: continued

It can be concluded that our approach provides high quality solutions when provided stochastic data and can effectively replan the route in the face of adverse weather conditions. We are further able to show that the routes we find are not much worse than those generated with perfect information, meaning our algorithm could be used in a real system.

5.6 Conclusion

In this paper, we presented a GA for the weather-dependent optimization of routes for vessels. We introduced an algorithm for generating initial routes as a useful supplement for the GA and listed a variety of domain-specific mutation operators for the GA to find good routes adapted to the present weather. The combination of this algorithm and the GA is highly effective at finding weather-dependent routes. Overall, the solutions for the different instances indicate that there is a need for the consideration of weather when generating routes for vessels. Weather-optimized routes lead to lower costs, and in some cases these routes are the only feasible routes. The experimental results showed that the proposed GA is able to find high quality solutions in a short amount of time. We have shown that the algorithm can handle stochastic weather data and generate reasonable routes that are not much worse than routes generated with perfect information. Furthermore, it makes sense to recalculate the route when new weather data becomes available. For future work, we plan to include more external factors, for example tides. The function for the bunker consumption will have to be extended then as well to include the new factors.

Acknowledgements

This work is partially supported by Deutsche Forschungsgemeinschaft (DFG) Grant 346183302. We would like to thank Lukas Arendt, Tino Engelbrecht, and Stephan Volkmann for their contributions to a student project leading to this work. Furthermore, we thank the Paderborn Center for Parallel Computation (PC^2) for the use of the OCuLUS cluster.

6 Exploiting Counterfactuals for Scalable Stochastic Optimization

Abstract

We propose a new framework for decision making under uncertainty to overcome the main drawbacks of current technology: modeling complexity, scenario generation, and scaling limitations. We consider three NP-hard optimization problems: the Stochastic Knapsack Problem (SKP), the Stochastic Shortest Path Problem (SSPP), and the Resource Constrained Project Scheduling Problem (RCPSP) with uncertain job durations, all with recourse. We illustrate how an integration of constraint optimization and machine learning technology can overcome the main practical shortcomings of the current state of the art.

6.1 Introduction

Optimization relies on data. To solve a knapsack problem, we need to know the profits and weights of the items, as well as the knapsack's capacity. To solve a shortest path or travelling salesperson problem, we need to know the lengths of the links in the network. To solve a revenue optimization problem, we need to know demand and how prices affect demand. In practice, we often lack perfect knowledge of the situation we ultimately needed to plan for. Profits, transition times, price sensitivity, and demands frequently have to be estimated.

One simple and still widely used approach is to optimize for point estimates of the data: We estimate demand, profits, transition times, etc, and optimize for the resulting optimization problem. The problem with using only one set of estimates, even if they represented the maximum likelihood scenario, is that the probability of exactly this scenario taking place is close to zero, and performance of the solution that is optimal for this one scenario may decline steeply across a range of scenarios that, together, would have a reasonable probability mass. In other words, a solution that is sub-optimal for all scenarios but works with good performance for a large number of potential futures will lead to much better *expected* performance than the solution that is provably optimal for the maximum likelihood scenario yet abysmal otherwise.

6.1.1 Stochastic Optimization

The brittleness of solutions obtained by optimizing for one, point-estimated scenario only is well-studied in the field of stochastic optimization (SO). The objective of SO is to provide a solution that optimizes the expected returns over all possible futures.

This led to the idea of *two-stage stochastic optimization*: In the first stage, we need to take certain decisions based on uncertain data. After taking these decisions, the uncertainties are revealed and we can take the remaining decisions based on certain data. This allows us first to make up for certain inconsistencies our initial decisions might have created (note that the constraints are also based on estimates) and thus exercise certain *recourse actions* to regain feasibility, and second to optimize the second-stage decisions that can wait to be taken until we know the real data. An overview of two-stage stochastic integer programming problems can be found in Birge and Louveaux (2011), and Van Slyke and Wets (1969) present a method to solve two-stage problems using the special form of these problems.

One crucial step in stochastic optimization is the generation of a representative set of potential futures (*scenarios*). Many methods exist to generate scenarios, and Kaut and Wallace (2007) points out that quality scenario generation is critical to the success using SO. Hochreiter and Pflug (2007) recommend that a number of different data sources should be used for scenario generation.

Obviously, solving SO problems to optimality gets harder the more scenarios are considered. Sample average approximation Kleywegt et al (2002) has been developed to generate a small random sample of scenarios and approximate the expected value function. This technique has been applied to a variety of problems (see, e.g., Long et al (2012), Schütz et al (2009), Verweij et al (2003)) and can help the method scale a bit better. However, the fundamental problem remains that SO relies on a representative set of scenarios to be considered, and that it must make optimal first and second stage decisions for every scenario under consideration.

6.1.2 Multi-stage Stochastic Optimization

One practical aspect that we also need to take into account is that the execution of a planned solution is frequently disrupted by outside events: equipment or crew assumed to be available may suddenly go out of service, requiring adjustment of a plan during operations. Consequently, the plan may need to be adjusted multiple times.

This leads to the idea of *multi-stage stochastic optimization.* In multi-stage SO, uncertainty is revealed in multiple consecutive steps, and more decisions need to be taken at each stage. In these problems, random variables in later stages depend on the decisions taken in the earlier stages. Models and solutions to these problems are therefore structured in the form of a tree Dupačová et al (2000), with independent decisions at the root node, and dependencies between decisions modeled with parent-child relationships in the tree.

Due to their richer modeling power, these types of SO models are especially relevant for real-world decision making, but unfortunately explode in complexity very quickly, even when employing advanced decomposition techniques like presented by Birge (1985) who extend the work of Van Slyke and Wets (1969). Furthermore, the problem of scenario generation is even more daunting in this more realistic setting, as conditional scenarios need to be generated, and often the data needed for this purpose may not be available. An overview of scenario generation methods for multi-stage stochastic programs is provided in Dupačová et al (2000).

6.1.3 Simulation-based Optimization

Modeling dynamic recourse and managing a meaningful number of scenarios in stochastic optimization is often cumbersome. An alternative is to employ a *simulator* that can evaluate a given plan on a number of scenarios, whereby the algorithm to generate recourse actions is built into the simulation. The recourse policy employed by a real-world organization may involve solving nested optimization problems on the go, as SO assumes, but oftentimes the real-world operational constraints may not allow for a full-fledged optimization, for example because the data needed is not readily available, or because re-optimization would be too time-consuming. A simulator can easily reflect the real recourse actions that would be taken, which are usually locally optimal only, or maybe just best-effort heuristics.

Simulation-based optimization is thus an alternative to stochastic optimization April et al (2003); Fu et al (2005). In this setting, a simulation is constructed to provide a stochastic evaluation of a provided solution. The search for good solutions can then be conducted by employing a meta-heuristic procedure. For example, Glover et al (1999) employs tabu search for this purpose.

An alternative to using a general local search heuristic is to apply bandit theory and to conduct a search based on Bayesian optimization Pelikan et al (1999). In this method, the search space is traversed in a statistically principled way which balances exploitation and exploration by considering new solutions for simulation next which combine high expected performance with high uncertainty of this performance.

No matter which search method is employed, to compute an objective function value for an instance, we need to expose it to certain futures. In SO these were called scenarios, in simulation-based optimization the "scenario generation" is hidden in the simulator. However, both methods rely on an adequate representation of potential futures *of the world as it currently presents itself.*

6.2 Technology Gaps

In practice, many organizations do not take the uncertainty in their forecasts into account when devising their operational plans. In fact, this observation even holds for those organizations that would stand to benefit the most, because events disrupting their plans frequently ruin all operational success. Airlines are one prototypical example. Over decades, the airline industry has spent billions of dollars on optimization technology to improve their operational planning (e.g., in crew planning Luo et al (2015)). There is certainly no lack of affinity to optimization technology, nor a lack of understanding that their current optimized plans are very brittle. The question is, why then is decision-making under uncertainty not employed?

We believe there are three main factors that prevent current decision making under uncertainty technology from being applied in practice:

- Complexity of modeling the base problem
- Inability to generate meaningful future scenarios for the current situation
- Computational limitations preventing the scale-up to real-world numbers of primary and recourse decisions

Take the example of an airline again. Flights may be delayed due to weather or traffic. Gates may be occupied and have to be changed. Crew may be out of service because of sickness or because they are delayed and past their maximum allowed service time. Equipment may not be available because of technical issues or because other issues in the network prevented the plane from being at the airport where it was planned to be.

Modeling the operation of an airline is extremely complex to begin with, which is why airlines break down the original problem into network design, revenue management, fleet assignment, crew pairing, tail assignment, and crew scheduling problems. There are literally millions of decision variables to consider. Secondly, there are frequently no models available for assessing the probabilities of disruptions with any meaningful accuracy. This is especially true for the joint distributions of disruptions which are frequently correlated. And finally, the number of recourse decisions taken during operation is staggering: Airlines literally run their recovery solvers every minute to adjust their plans to ever new, thankfully usually minor, disruptions.

Stochastic optimization is not applicable, because computation times are prohibitively long, and the number of recourse decisions far exceeds efficient modeling capabilities. However, simulation-based optimization cannot handle the millions of decision variables or the complex constraints that govern whether solutions are even feasible.

This analysis is the starting point for our research. In the following, we propose a framework for decision making under uncertainty that overcomes the limitations of existing technology. In a nutshell, we propose a paradigm shift away from trying to anticipate the future and towards discovering structures in the solutions that correlate with historically good performance. In doing so, we trade dual bounds (i.e., a guarantee of the relative quality of the solution provided by SO) for scalability and easier modeling.

6.3 Learning From Counterfactuals

A key limitation of stochastic optimization is the need to model every decision. Not only does this put an enormous burden on the modeler and the optimization, it often also falsely assumes that we were able to optimize recourse decisions during operation. Another problem that both simulation-based optimization and stochastic optimization share is that they need to generate meaningful scenarios how the execution of a solution may unfold in the future. Finally, both methods scale to a hundred, maybe a few thousand decision variables, before computation times become impractically long.

Our proposal is to combine both what disruptions are likely, as well as how well the initial solution is adjusted during execution, into one data-driven forecast.

Consider a model that, given two solutions to an optimization problem can provide a classification as to which of the two solutions will fare better when they are executed. Consider solving this problem in two stages. In the first stage, we solve the problem as if there were no uncertainty using the expected costs in the objective function, and generate multiple near-optimal solutions. The goal of the second stage is to determine which of the near-optimal solutions will likely lead to better results when executed. We train a model that compares these solutions on a pairwise basis and choose the solution that wins the most times against the other solutions. This general method alleviates many of the problems existing approaches encounter:

- The first-stage problem is as easy to model as the optimization problem without uncertainties.
- There is no need to generate future scenarios for the current data at hand.
- There is no complexity blow-up, no matter how many recourse actions are needed.

All of the complexity is off-loaded into the second stage model. The crucial question is, of course: How can we obtain a model that, given two solutions, can predict which one will fare better in operations?

Thesis: We can learn such a model from historical data.

We argue that all that is needed to learn such a model is to keep track of our estimates over time, and what eventually happened. Consider, e.g., the Stochastic Shortest Path Problem (SSPP), a problem that Cao et al (2015) argue is particularly difficult when there are no assumptions about the uncertain travel times. We can track how the arc transit time estimates for the entire network have evolved over time, and what they ended up being. Or, for the Stochastic Knapsack Problem (SKP), we can examine what our weight and profit estimates were before each decision for historical solutions and what they turned out to be in reality. That is to say: Historical data often enables us to compare *multiple* historical solutions, even though only one of them was actually executed in reality, whereas all others are essentially *counterfactuals*.

Please note a subtle but very important difference; the historical data is enough to compare two potential solutions for the optimization problem *as it presented itself in the past*. It would, however, not enable us to simulate two solutions for a new instance of our underlying optimization problem. Take the SSPP as an example. We may have a historic example where we needed to go from some node s to node t. We know how our estimated arc transition times evolved over time and the resulting values on all arcs in the network. With this, we can compare two paths P_1 and P_2 that connect s and t.

Now imagine we currently need a solution to go from node s to node t again. Our initial estimates are of course completely different from those in the historic example.

Consequently, we cannot just simulate two paths Q_1 and Q_2 in the old scenario and assume that their relative performance would remain the same under the current conditions. In fact, if this were the case, we should forget our estimates altogether and just always go from s to t using the exact same route all the time.

However, if we could capture *estimate-dependent characteristics*, or *features*, of pairs of paths P_1 and P_2, and associate these characteristics with the *relative* performance of these paths, then, by repeating this exercise many times, we just might be able to *learn* to tell which of any given pairs of paths will probably execute better – albeit with no guarantees.

Through this framework, we have now decomposed the problem of making primary decisions based on uncertain forecasts and assumptions regarding estimate distributions and recourse policies into two tasks: We first need to model the primary optimization problem. Second, we need to use historical data to build a supervised set of examples of pairs of solutions, recording which one fared, or would have fared, better. Crucially, we need to devise a set of features to characterize the solutions *in the context of the problem instance they were generated for.*

We formalize our framework as follows. We are given a deterministic optimization problem P with decision variables $\mathbf{x}$. Let $f(\mathbf{x})$ be the objective function of the deterministic problem, and $f'(\mathbf{x}, \omega)$ be the objective function when the decisions are evaluated under scenario ω.

1. Training set generation: We first generate n solutions $\mathbf{x}_{i1}, \dots, \mathbf{x}_{in}$ to the problem instance i in a set of training instances I, where all uncertain parameters take their expected value. The choice of such solutions is up to the user of this framework, but we recommend high quality solutions with some diversity. We associate a label $y_{ij} = \sum_{\omega \in \Omega_i} f'(\mathbf{x}_{ij}, \omega)/|\Omega_i|$ with each solution of each instance for a set of counterfactual scenarios Ω_i that are derived from the true scenario that unfolded for the historic problem instance i. Finally, we compute problem dependent feature vectors $\mathbf{u}_{ij} \in \mathbb{R}^f$ describing each solution j of instance i.

2. Learning a classifier: Next, we train a binary classifier M that, given two solutions j, k to a problem instance i, forecasts which of the two solutions will likely perform better when executed. The training input for this cost-sensitive learning task are triples $(u'_{ijk}, y_{ij}, y_{ik})$, where $u'_{ijk} \in \mathbb{R}^{3f}$ consists of a concatenation of feature vectors u_{ij}, u_{ik}, and $u_{ij} - u_{ik}$. We use the technique from Malitsky et al (2013) for this purpose.

3. Deployment: Given a problem instance i, we generate n solutions using the deterministic optimization model with expected values for uncertain parameters. We then compute the features u'_{ijk} for each pair of solutions j, k. Then, we query the model M for all such pairs and choose the solution that "wins" the most times.

In the following, we will exercise the above steps for three optimization problems: the SKP, the SSPP, and the Resource Constrained Project Scheduling Problem (RCPSP), each with recourse. The objective of this study is to investigate whether we can effectively learn which solution for a problem instance will perform better.

6.4 Stochastic Knapsack

The SKP is the stochastic variant of the well-known optimization problem: Given $n \in \mathbb{N}$ items $\{1, \ldots, n\}$ with profits $p_1, \ldots, p_n \in \mathbb{N}$, expected weights $w_1, \ldots, w_n \in \mathbb{N}$, and a capacity $C \in \mathbb{N}$, the objective is to find a subset of items $I \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in I'} w'_i \leq C$ and $P = \sum_{i \in I'} p_i$ is maximized, where $w'_1, \ldots, w'_n$ are the actual weights incurred, and I' is the set of items we ultimately include in our knapsack.

6.4.1 Stochastic Environment

To complete the setup of our problem, we need to determine how the weights w'_i are derived from the expected weights w_i, and how I' derives from I during operations. This is precisely the task of determining the distributions of stochastic data, and the incorporation of recourse policies that we aim to avoid estimating and modeling when solving the stochastic variant of the underlying optimization problem. However, for the sake of experimentation, we obviously need to fix the stochastic environment.

We will assume that items have to be decided for inclusion or exclusion in sequence 1 to n.[1] That is, we first decide if we want to insert item 1 in the knapsack. If not, we can directly move on to the next item. If yes, then we add the actual weight w'_1 to our knapsack, the remaining capacity is reduced accordingly, and the profit p_1 is achieved. We consider all items in sequence. At stage i, we sample w'_i from a Pareto distribution with mean w_i (note: the nature of this distribution is not known to the optimization approach). In our variant of the problem, should the new item overload the knapsack, the item is automatically not inserted and we proceed as if we had never decided to include the item. However, if the item fits into the remaining capacity, we have to take it, even if the actual weight of the item is much larger than we had anticipated.

In terms of recourse, whenever during the sequential consideration of items our remaining capacity deviates from the anticipated capacity at that point in the sequence by more than a given percentage threshold p, we are allowed to reconsider our original plan and change the tail of our plan. However, if we include an item that was originally not planned to be included, we incur a profit penalty b (late buy penalty). Similarly, we incur a penalty s for items we do not include in our knapsack that we had originally committed to include (restocking fee). Finally, we cannot change the original plan for the next r items (minimum reaction time).

The recourse policy is to re-optimize the rest of the knapsack based on the profits adjusted for penalties, the originally estimated weights, and the remaining items and capacity. The selection of the next r items is fixed.

To support our introductory claim that existing technology is not feasible even for such a simple practical setting, we invite the reader to try to model this problem as an

---

[1] This is in contrast to some theoretical results on the SKP that assume we can decide in what order we wish to consider the items Dean et al (2008). We consider having this freedom less realistic.

n-stage stochastic optimization problem or as a simulation-based optimization problem with n variables and an uncertain side constraint.

6.4.2 Winner Forecasting

In stage 1 of our approach, we consider the original knapsack problem with the given capacity, profits, and estimated weights. We solve the problem to optimality using dynamic programming and generate a desired number of solutions that are either optimal or as close to optimal as possible.

In stage 2, we need to characterize each solution with respect to the given problem instance. Before we list the features, we introduce for this purpose, we define a number of quantities we can compute for any sequences of numbers.

For monotonically increasing (or decreasing) sequences, we define the following quantities (leading to $3q+2$ quantities for q quantiles considered):

- The mean, and the mean of the second moment.
- The median and the median of the second moment.
- For a desired number q of quantiles over the range of the sequence, the percentage of numbers in the sequence before each quantile is first reached, depending on whether the sequence is increasing or decreasing (including the last quantile).
- For a desired number q of quantiles, the value of the sequence at each quantile of items in the sequence, and the corresponding values in the second moment (excluding the last quantile).

For general sequences, we define the following 8 quantities:

- The mean and the mean of the second moment.
- The median and the median of the second moment.
- The minimum and maximum, and the corresponding values of the second moment.

Now, to characterize a given solution to a knapsack instance, we consider the following five monotone sequences, and the six general sequences thereafter:

M 1: For each item i in the sequence, the total profits achieved so far, as a percentage of the maximum achievable profit (here and in the following per the given solution).

M 2: For each item i in the sequence, the remaining capacity as a percent of the total capacity.

M 3: For each item i in the sequence, the linear programming upper bound for the remaining items and the remaining capacity, as percentage of the optimum profit.

M 4: For each item i in the sequence, the linear programming upper bound for the remaining items and the total original capacity C, as percentage of the optimum profit achievable.

M 5: For each item i in the sequence, we compute the number d_i of items since the last item that was included in the solution. We aggregate and normalize these numbers by setting $D_i = \sum_{k \leq i} d_i/n$ and considering the monotone sequence $(D_i)_i$.

G 1-3: For each item selected in the given solution, its profit (as percentage of maximum profit), weight (as percentage of total capacity), and efficiency (the ratio of profit over weight).

G 4-6: The same three values as above, but over the items not selected in the solution.

We consider 5 quantiles, therefore the above yields $5(3 * 5 + 2) + 6 * 8 = 133$ features. We add two more by also computing the total efficiency of the solution, defined by the ratio of total profit divided by total capacity, and finally the LP/IP gap as percentage of maximum achievable profit. In total, for each solution we thus obtain 135 features. For a given pair of solutions, we concatenate the features of each solution, as well as the difference of the features of the two solutions. Our machine learning approach thus has access to $3 * 135 = 405$ features to decide which of the two solutions given is likely to perform better than the other.

To complete the data-driven part of our approach, we choose binary cost-sensitive classification to rank the solutions, in particular, the cost-sensitive hierarchical clustering approach from Malitsky et al (2013). We use this technique in all following test cases.

6.4.3 Numerical Results

We generate knapsack instances with 1,000 items and (expected) weights drawn between 1 and 100 uniformly at random. The capacity is set to 10% of the total expected weights of all 1,000 items. Weakly correlated knapsack instances are generated by choosing the profit of item i with weight w_i in the interval $[w_i - 3, w_i + 3]$. Strongly correlated instances are generated by setting the profits to $w_i + 5$. Furthermore, almost strongly correlated instances are generated by choosing the profits in $[w_i + 4, w_i + 6]$ uniformly at random.

We build a simulation environment where the weight of an item i is drawn from a random variable following a Pareto distribution with mean w_i and minimum value $0.95 w_i$. Note that the Pareto distribution is heavy-tailed: With the given parameters, there is only about a 20% chance of seeing a value larger than the mean, but a 1.5% chance to see a value of at least 1.5 times the mean, and about a 0.3% chance of encountering values of twice the mean or more.

We set the recourse threshold $p = 5\%$, the restocking fee and late buying fee $s = b = 10$, and the minimum reaction time $r = 5$. Whenever the remaining capacity in the knapsack deviates by more than $p = 5\%$, we solve a new knapsack problem (with adjusted profits to reflect the respective restocking and late buying fees) to determine our recourse action for the remaining items beyond the minimum reaction threshold.

Using this environment, we generate 100 instances of each knapsack type (weakly, strongly, and almost strongly correlated). To build our *test benchmarks*, we solve each knapsack to optimality using dynamic programming and choose ten near-optimal solutions. We then run each of these solutions through our simulation environment twenty times, so that each solution is exposed to the exact same twenty simulations. We then record the average performance for each near-optimal solution over the twenty simulations to grade them. In practice, there would only be one reality the selected solution would be exposed to, of course. We run each test solution through twenty potential futures to lower the possibility that we are just lucky with the scenario we encountered.

The task for our data-driven solution selector is to pick a solution from the set of ten that exhibits very good performance in the simulated environment. To train this assessor, we generate training data as follows: For each knapsack type, we generate 500 instances. For each instance, we generate twenty near-optimal solutions. Moreover, for each of these instances, we generate one, and only one, vector of weights for each item. Note that, in practice, we would equally have access to our originally expected weights w_i, and the actual weights w'_i.

Next, we need to counter-factually assess the performance of each solution. To lower the variance in these labels, we proceed as follows: First, we build twenty derived scenarios from each real scenario, by choosing weights $w''_i \in [w'_i - \alpha, w'_i + \alpha]$, where $\alpha = \frac{|w_i - w'_i|}{2}$, uniformly at random. That is, we derive scenarios from the historical examples without any assumptions regarding, or knowledge of, any distributions. We merely consider the *actually encountered* deviations from our original estimates and derive scenarios by varying these deviations a little. Please note that these changes do not affect the *direction* of the deviations: A weight that was under-estimated, remains under-estimated in each derived scenario, and each weight that was over-estimated remains over-estimated.

Type	**Max**	**Mean**	**Min**	**ML**	**GC**
Weakly Correlated					
500-100 10-10 10%-10%	15.29	10.67	5.79	9.10	32
500-100 5-5 10%-10%	14.30	9.93	5.47	8.95	22
500-100 10-10 20%-20%	9.75	6.69	3.82	5.77	32
100-100 10-10 10%-10%	15.29	10.67	5.79	9.39	26
500-100 5-10 10%-10%	15.29	10.67	5.79	9.38	26
Strongly Correlated					
500-100 10-10 10%-10%	15.56	10.84	6.03	8.88	41
500-100 5-5 10%-10%	14.46	10.03	5.54	8.81	27
500-100 10-10 20%-20%	11.73	8.18	4.58	6.50	47
100-100 10-10 10%-10%	15.56	10.84	6.03	9.31	32
500-100 5-10 10%-10%	15.56	10.84	6.03	8.83	42
Almost Strongly Correlated					
500-100 10-10 10%-10%	15.63	11.12	5.86	9.42	32
500-100 5-5 10%-10%	14.21	10.03	5.44	8.93	24
500-100 10-10 20%-20%	11.82	8.31	4.43	7.31	26
100-100 10-10 10%-10%	15.63	11.12	5.86	9.68	27
500-100 5-10 10%-10%	15.63	11.12	5.86	9.59	29
Heterogeneous Mix					
500-100 10-10 10%-10%	17.28	11.34	5.31	9.32	34
500-100 5-5 10%-10%	15.73	10.35	4.83	8.83	27
500-100 10-10 20%-20%	11.97	7.86	3.85	6.53	33
100-100 10-10 10%-10%	17.28	11.34	5.31	10.0	22
500-100 5-10 10%-10%	17.28	11.34	5.31	9.51	30

Table 6.1: SKP Results

Finally, we execute each of our twenty near-optimal solutions under each derived scenario (including the recourse actions we would have taken) and label each with the average performance observed. Note that all that is needed to conduct this counterfactual assessment of additional solutions is the knowledge about our original estimates and the real item weights that were encountered.

Test results on all three classes of knapsacks are shown in Table 6.1. In the first column we denote the parameters of the experiment: The number of training scenarios vs number of test scenarios (usually 500-100), late-buying and restocking fees on train vs test (usually 10-10 or 5-5), and the knapsack capacity on train vs test (usually 10% or 20% of the weight of all items for both).

Next, we show the average of the worst of the ten solutions we generated for each test instance, the expected performance, and the performance if we chose the best of the ten solutions generated. Note that the latter is the maximum gain we can hope to achieve by selecting among the ten solutions generated. The numbers represent percentages above an imaginary best solution (since the ten we generated may obviously not include the optimum under uncertainty), which we set at three standard deviations below the average of the ten solutions, and whose performance itself we measure as percent above the best omniscient solution. In absolute terms, the numbers presented are thus percentages over percentages over the true profits.

Finally, we show the performance of counterfactual selection (ML), as well as the percent gap closed (GC) between the average performance and the best performance that is achievable by selecting among those selected ten solutions for each instance.

Overall, we close between 22% and 47% of the gap between the average performance and the best solution available to us. That means that our forecasting models are certainly not optimal, but nevertheless effective at choosing solutions which are expected to perform better than the average near-optimal solution. This holds for varying knapsack types as well as different capacities and recourse penalties.

To assess how critical the amount of historical scenarios is, we lowered the training set to only 100 scenarios. On all knapsack types, this leads to a reduction in effectiveness, but the approach still works: We close 26%, 32% and 27% for the three knapsack types using only 20% of scenarios.

Encouragingly, we see that counterfactual forecasting can also be reasonably effective when the historical scenarios used were gathered under a different regime. For example, assume that, historically, the late-buy and restocking fees were 5, but now they are 10. Please note that what should be done when operational parameters change is to re-run the historical scenarios under the new penalties and to generate a new counterfactual training set this way. For experimental purposes only, we did not do that here so we can assess how robust our forecasting models are under varying parameters. Under [500-100 5-10 10%-10%] we see that we achieve 26%, 42% and 29% gap closed for weakly, strongly, and almost strongly correlated knapsacks, respectively.

Finally, we generated a benchmark which consists, in equal parts, of weakly, strongly, and almost strongly correlated knapsack instances, both for training and for testing. As the table shows, the counterfactuals-based predictive models work for heterogeneous mixes of different knapsack types as well.

Overall, we conclude that, for the SKP, we can learn an effective, though suboptimal, data-driven model to predict which near-optimal solution has greater chances of performing well in an uncertain future.

6.5 RCPSP with Uncertain Job Durations

The RCPSP with uncertain job durations involves the scheduling of a set of jobs J given a set of resources R and a set of time periods T. Each job j consumes u_{jr} units of resource r in each time period the job is running. Each resource has a maximum capacity k_r that may be consumed in each time period. A precedence graph $P = J \times J$

specifies an order in which jobs are executed, i.e., for $(i, j) \in P$ job i must be completed before j can start. In the deterministic case, each job j has a fixed duration d_j. We consider a version of the problem where the job duration is uncertain, and assume that, if a job takes longer than planned, it continues to consume u_{jr} resources in each additional time period. This version of the problem corresponds closely with real-world RCPSPs, such as construction or software projects in which delays are common, and resource consumption of jobs continues even if they take longer than expected.

The (deterministic) RCPSP can be modeled with the following constraint program Berthold et al (2010), in which the start time of each job is given by S_j:

$$\min \max_{j \in J} S_j + d_j \tag{6.1}$$

$$\text{subject to } S_i + d_i \leq S_j \qquad \forall (i, j) \in P \tag{6.2}$$

$$\texttt{cumulative}(\mathbf{S}, \mathbf{d}, \mathbf{u}_{\cdot r}, k_r) \qquad \forall r \in R \tag{6.3}$$

6.5.1 Stochastic Environment

We sample the job durations d'_j from a Pareto distribution with the expected value d_j. The simulation starts at time period 0 and iterates through each time period until the maximum time is reached or all jobs have been executed. For some time period t', all jobs ending in that period ($t' = S_j + d'_j$) are ended and the resources they are consuming freed. We then start jobs that have a start time of the current time period, if their precedence constraints are satisfied and their resource consumption requirements can be met. If job j with $S_j = t'$ cannot start in t', $S_j \leftarrow S_j + 1$, i.e., we delay its start by one time period.

If significant delays occur, it may be appropriate to do recourse planning and find new start times for the remaining jobs based on the current forecast. The recourse planning involves simply fixing the start times of jobs that are finished, or running and updating the job durations with either the real duration for finished/running jobs or the current forecast for scheduled jobs. This deterministic problem can then be solved by any RCPSP algorithm, based on the CP model above.

We forecast job durations by assuming that, when a job i that must precede j is finished (i.e., $(i, j) \in P$), we know more about the duration of j than we did before i finished. We construct a graph with the same nodes and arcs as in the precedence graph, and assign the true duration d'_i to every arc (i, j). Let a_{ij} be the shortest path between all pairs of jobs on the newly constructed graph with Dijkstra's algorithm. We then compute the forecast as $f_{ij} := \text{round}(d_j + (1 - a_{ij}/\max_{k \in J}\{a_{ik}\})(d'_j - d_j))$, such that f_{ij} is the forecast for job j when job i is finished, assuming j is reachable from i in P. While simulating, when job i finishes we check f_{ij}, and if it is closer to the true duration of j, we update our expected duration.

6.5.2 Winner Forecasting

We propose the following groups of features to describe solutions to the RCPSP with uncertain job durations.

1. The expected makespan of the solution divided by the maximum time
2. Let $B_{ij} := S_j - S_i$ for all $(i, j) \in P$ be the buffer between jobs with precedence relations. We compute the mean, median, standard deviation, skew, 25% quantile, and 75% quantile of the values in B.
3. Let $B^T_{ij} := B_{ij}(S_j/|T|)$. We compute the mean and skew of B^T.
4. We execute the solution with the expected durations and compute the percentage residual resource usage $\hat{k}_{tr}$ for each resource at each time period, and aggregate this into $\hat{k}_t := \sum_{r\in R} \hat{k}_{tr}/|R|$. We compute the mean, standard deviation, 25% quantile and 75% quantile over all values of $\hat{k}_t$.
5. Let m^1 and m^2 be the number of jobs *directly* affected (we do not examine network effects here) due to insufficient available resources if a job j starts 1 or 2 time periods later than planned, respectively.
6. Let a *delay chain* be a path in the precedence graph which forms a sequence of jobs that are separated with buffer less than the 25% quantile of B. We compute the maximum length delay chain and divide it by the total number of jobs.

6.5.3 Numerical Results

We test our approach on the well-known instances from the PSPLIB Kolisch and Sprecher (1997). We use the j30 and j60 categories, which have 30 and 60 jobs, respectively, and split each into 320 training instances and 160 testing instances. The maximum job duration in these categories is 10 time units, so we add two more instance categories containing randomly generated job durations with a maximum of 50 and 100 time units, respectively.

We solve the constraint programming model in (6.1) through (6.3) with Google OR Tools CP-SAT solver version 7.0 Google (2019). We first generate the optimal expected values solution. We note that, in the RCPSP, shifting the buffer of a few jobs results in a “new” solution, but this is not desirable for our approach, as the realized performance will be nearly the same. Therefore, to generate $k - 1$ solutions in addition to the optimal solution, we begin an iterative process. After a solution S' is found, we append the following constraints to require that a given percentage of the jobs have a different order than the previously found solution:

$$o_{ij} = 1 \Leftrightarrow S_i \circ S_j \qquad \forall S'_i \star S'_j,\ (\circ, \star) \in \{(>,<),(<,>),(\neq,=)\} \tag{6.4}$$

$$\sum_{i,j\in J, i<j} o_{ij} \geq h \tag{6.5}$$

where the decision variable $o_{ij} \in \{0, 1\}$ for $i, j \in J, i < j$ is 1 iff jobs i and j have

Class	Max Dur.	Train	Test	Max	Mean	Min	ML	GC
j30	10	317	157	4.48	3.27	2.67	3.44	-28
	50	275	138	3.84	3.08	2.59	3.01	15
	100	260	135	3.70	3.05	2.63	3.02	7
j60	10	239	122	4.06	3.29	2.53	3.18	15
	50	225	109	3.48	2.88	2.27	2.70	30
	100	226	113	3.38	2.82	2.28	2.70	23

Table 6.2: RCPSP Results

a different order than in the previous solution. We require the number of job order changes in (6.5) to be greater than a threshold h, which we set to 5% of the unique job pairs ($|J||J-1|$).

As for the SKP and SSPP, we test our approach on 20 simulations per instance, simulating training and testing instances the same way as the previous two problems, with training simulations being derived from only one real simulation without knowledge of the actual distributions of job delays, and test simulations running 20 real scenarios for proper evaluation. Table 6.2 shows the results in the same format as the SKP and SSPP, with the addition of columns indicating the number of training and testing instances.

We are able to achieve modest gains over using the expected value solution, except in the case of j30 with a maximum expected duration of 10. On this instance set, the learning algorithm failed to find a good way of identifying superior solutions. This may be due to our features, which focus closely on buffer, and this may not be sufficient when the durations are low.

On the j60 instances, we are able to close between 15% and 30% of the gap to the best available solution. Even though the absolute gain may seem small, as with many optimizations under uncertainty problems, real-world RCPSP problems can involve expensive resources (specialized digging equipment, etc.), and even small absolute improvements often translate into significant cost savings, as well as time savings for the overall plan. Therefore, even though our method is heuristic in nature, it can be of high value in practice.

6.6 Stochastic Shortest Path Problem

In the SSPP, we are given a graph $G = (V, A)$ of nodes V and arcs A. Every arc $(i, j) \in A$ has an uncertain cost with an expected value of c_{ij}. The objective is to find a minimal cost path through the graph between a source node s and destination node t. The SSPP can model problems such as the routing of ships under the influence of weather, or routing a vehicle through a road network considering traffic delays.

6.6.1 Stochastic Environment

We base the stochastic environment for the SSPP on the one described for the SKP with a few problem-specific modifications. Given a solution to an SSPP instance, we first sample the realized costs c'_{ij} for each arc from a Pareto distribution with mean c_{ij} and the minimum at 90% of c_{ij}. We then begin executing the path given to us as one of the ten solutions, using c'_{ij} for each realized arc. If the accumulated delay exceeds 10% of the expected costs, we allow recourse planning every 5 nodes.

In the recourse planning, we adjust our forecast based on the current node. The assumption is that arcs close to this node have a more accurate forecast than those far away, since we would traverse these arcs in the nearer future. To assemble our forecast, let a_{ij} be the number of arcs between nodes i and j. Then, let the forecast cost for (i, j) be $f_{ij} := \text{round}(c_{ij} + (1 - a_{ij}/\Delta\})(c'_{ij} - c_{ij}))$ if $a_{ij} < 5$, and c_{ij} otherwise. We set Δ to 7 to keep the forecasts from becoming too accurate when we get close, but keep them inaccurate when we are far away.

6.6.2 Winner Forecasting

We introduce the following features to characterize an SSPP solution. For each feature set, we compute the minimum, maximum, mean, standard deviation, skew and kurtosis of the array of values. For features using arc costs, we divide the costs by the average arc cost of the graph, and for features using node degrees, we use the average node degree of the entire graph.

1. Array of arc costs on the path
2. Array of arc costs over the set of arcs leaving nodes of the path going to nodes not on the path
3. Array of arc costs over the set of arcs leaving nodes that are connected to the path by a single node (excluding any arcs to nodes directly connected to the path or nodes on the path)
4. Array of node degrees in the path
5. Array of node degrees of nodes that are connected to nodes on the path
6. Array of node degrees over the set of nodes that are connected to the path by a single node (excluding any nodes directly connected to the path)

As in the case of the SKP, we concatenate the features for two given solutions with the difference between the features of both solutions, which are then used by the machine learning approach to determine the most promising solution.

6.6.3 Numerical Results

We build a dataset of SSPP instances consisting of graphs based on one of three graph types: G_{nm} Erdös and Rényi (1959), “bottleneck”, and Watts-Strogatz small-world graphs. The bottleneck instances consist of five G_{nm} graphs of equal size connected sequentially with 5 links between each graph. We create 300 instances of each graph

Graph Type	**Nodes**	**Max**	**Mean**	**Min**	**ML**	**GC**
G_{nm}	50k	207.38	91.68	43.00	68.63	47
Bottleneck	50k	165.55	97.06	54.27	71.80	59
Watts-Strogatz	25k	175.88	76.95	33.91	50.79	61
Mixed	50k/25k	182.81	92.98	43.50	61.17	64

Table 6.3: SSPP Results

type and size and select random source and sink nodes for the path, splitting the instances into 200 train and 100 test. The expected arc costs c_{ij} are drawn uniformly random between 1 and 100. We further ensure that all graphs have no isolated components by adding arcs between such components and the rest of the graph.

For each SSPP graph, we generate the ten shortest paths for a given graph between s and t using Yen's algorithm Yen (1970). We simulate using the same scenario structure as in the SKP. The "true" arc costs c'_{ij} are drawn from a Pareto distribution with an expected value c_{ij}, shifted so that the minimum is at $0.9c_{ij}$. Training instances are evaluated on 20 scenarios that are all variations of a single scenario (using the exact same scenario variation as in the SKP), and test instances are evaluated on 20 scenarios generated independently of each other. Every 5 nodes we check if the accumulated delay is more than 10% of the expected cost, and if it is, we run a recourse algorithm that tries to replan the shortest path to t from the current node.

Table 6.3 shows the results of our computational experiments. We compute the gap to the optimal shortest path considering c'_{ij} and average it over 20 scenarios as described above. Note that, for many graph instances, paths that were near optimal for the point-estimated scenario may perform much worse than the shortest path had we known the true arc distances beforehand. This leads to relatively high values in our table, but is really more a reflection of the inherent cost of uncertainty in this particular problem than the absolute performance of the particular algorithm used to optimize under the uncertainty. Looking closer at our data, we find that the expected path lengths of the solutions is usually about the same, with the bottleneck graphs exhibiting slightly higher variance than for the other graph types.

Despite the simplicity of our features (we just measure arc costs and node degrees), we are able to close the gap by around 50% in all graph types and 64% for the mixed setting. This provides further support that we can learn from historical data which solution features are favorable for later execution under stochastic disruption.

6.7 Conclusion

We have introduced a new methodology for modeling and heuristically solving stochastic optimization problems. The key idea is to move away from trying to accurately forecast the uncertainty in the problem instance at hand. Instead, we propose to use logs of historical estimates and the realities that followed for comparing various counterfactual solutions. Our thesis is that we can devise features that capture instance-dependent characteristics of the solutions that allow us to predict which solution from a solution pool will likely perform well for a new problem instance at hand.

The objective of this paper was to provide a proof of concept. We considered three stochastic optimization problems that would each be extremely hard to model and solve with existing approaches, even heuristically. For all three problems, we were able to quickly devise sets of features that were effective enough to choose solutions that were superior to picking an average solution from our pool of optimal (with respect to the underlying point-estimated optimization problem) or near-optimal solutions.

Note that we did not spend any time to optimize hyper-parameters of our learning approaches, or to engineer more effective features. Providing a general set of features and pairing it with off-the-shelf machine learning methods was enough to tackle each of the three optimization problems. We believe that the experimental results provided strongly support our thesis that we can learn from data which solutions will exhibit superior performance in an uncertain future. However, this is of course not to say that, in practice, one should not conduct feature engineering and hyper-parameter optimization to achieve even better results.

The ability to tackle complex stochastic optimization problems with thousands of recourse stages comes at a cost, though. The framework presented gives no guarantees regarding the quality of the solutions achieved, and dual bounds are not provided. Therefore, whenever traditional stochastic optimization is applicable and full online-reoptimization is feasible during real-world operations, we would recommend this approach. The framework introduced here is meant for situations when the traditional methods break down.

In the future, we intend to investigate if the models trained on historic counterfactuals can be mined to infer constraints to guide the search for less brittle solutions directly: solutions that are not only near-optimal for the “fair weather” data, but also have high probability of performing well under stochastic disruption. In this sense, the new framework opens the door for a comprehensive new research agenda for stochastic constraint optimization.

Acknowledgements This work is partially supported by Deutsche Forschungsgemeinschaft (DFG) grant 346183302. We thank the Paderborn Center for Parallel Computation (PC^2) for the use of the OCuLUS cluster.

7 Conclusion and Outlook

This dissertation considered various aspects of stochastic optimization of maritime logistics problems. Chapters 1 and 2 motivated for the importance of stochastic optimization of maritime logistics problems and outlined the most relevant uncertainties. Chapter 3 gave an overview of the research papers examining three different problems related to stochastic optimization that are part of this dissertation, and which were then presented in the following chapters.

Chapter 4 introduced a stochastic version of the LSFRP including uncertain container demands and travel times. The solutions of this model showed the possible reduction of delays and increase of profits compared to the deterministic version of the problem. Chapter 5 presented a heuristic approach for the problem of weather routing of vessels, where stochastic weather influences the speed and consequently the fuel and time consumption of a route. It also emphasized the importance of including the influence of uncertain weather on routes of vessels, as the resulting routes are more cost efficient and safer to sail when applied under realistic conditions. A new methodology for modeling and heuristically solving stochastic optimization problems was introduced in Chapter 6. It was shown that the proposed combination of optimization and machine learning is able to overcome drawbacks of traditional stochastic optimization and identify promising solutions for unknown problem instances.

The core question of this dissertation was to find out if stochastic optimization provides solutions for maritime logistics problems that, when realized, result in higher expected profits than solutions found through deterministic optimization techniques. This question can be affirmed when looking at the results for the goals defined in Chapter 1. Each reported stochastic optimization approach for achieving the goals showed a better solution quality when compared to deterministic solutions.

The first goal of this thesis was the *modeling of uncertain factors for the Liner Shipping Fleet Repositioning Problem.* This goal was achieved by modeling the LSFRP with uncertain container demands and travel times. The approach showed how to include a combination of these uncertainties into an optimization model and is an example helping to apply this extension to other maritime problems. Furthermore, it showed the complexity of the model caused by only two uncertain factors, which raises awareness for the complexity of real-world problems faced by industry that may involve even more uncertainties. This also shows the importance of the development of sophisticated solution techniques capable of dealing with such complex problems.

Solving the stochastic optimization model and analyzing the solutions led to the achievement of the goal of *providing exact solutions for the LSFRP under the consideration of different uncertain input factors.* The results showed the positive impact of stochastic optimization on solutions regarding both costs saved due to lower delays and higher profits earned by fulfilling demands. Furthermore, it was identified that uncertain travel times have a higher impact on solutions than uncertain demands. It also became clear that the application of optimization is necessary to obtain good solutions for problems with uncertain input parameters and that it is no longer sufficient to use traditional tools, which are still partly in use in industry today.

Chapter 4 also presents an approach for *deriving of scenarios for the stochastic LSFRP from existing data*. First, historical data was used to generate scenarios. Afterwards, the number of scenarios was reduced to representative scenario sets to keep the models solvable in a reasonable amount of time. This process showed the complexity and importance of generating representative scenario sets that and can be used to obtain solutions that perform well when applied.

The usage of traditional stochastic optimization and the related inclusion of scenarios poses some relevant problems. It is essential that the selected scenarios are as representative as possible. Furthermore, large-scale problems, and therefore especially real-world problems, have a high number of decisions and a huge amount of stages, and are therefore difficult or even impossible to solve in a reasonable amount of time. This led to the *introduction of a new method for stochastic optimization combining optimization and machine learning* in Chapter 6. This framework overcomes the problems of scenario generation and solving huge problems by learning to identify promising solutions by training a machine learning model with historical data and related possible solutions. This approach shows the high potential of combining traditional optimization techniques with machine learning and provides new opportunities for solving stochastic optimization problems. Its novelty makes it interesting for further research and the short runtime for finding promising solutions makes it attractive for the application in industry. Furthermore, this approach confirms the secondary question of this dissertation as it proves the possibility to develop a framework that is capable of handling the computational complexity of realistic problems with an arbitrary number of decision stages and complex recourse decisions.

The *development of a heuristic approach for the weather routing of vessels with stochastic weather data* was realized by implementing a genetic algorithm for the problem of vessel routing considering uncertain weather. Chapter 5 first showed the importance of the inclusion of weather data into the generation of routes for vessels. Afterwards, it was shown how the approach deals with stochastic weather data. It was pointed out how routes change under the influence of uncertain weather data and how to adjust routes with information that becomes available throughout the journey.

As the three research papers stated, there are still possibilities for extensions and future work. The stochastic model for the LSFRP can be extended with additional uncertain input parameters, for example related to times in ports or the availability of ports. Furthermore, the solution time can be improved with new solution techniques to solve problems based on large instances or containing more scenarios to optimality.

The genetic algorithm for the generation of weather-dependent routes for vessels can be extended by including more external factors, as for example tides. Furthermore, it is possible to use a more sophisticated function for the calculation of the bunker consumption of the used vessel.

The models trained within the presented scalable stochastic optimization framework could be used to infer constraints to guide the search for near optimal solution that have a high probability to perform well in different stochastic settings, but this still needs to be examined. Furthermore, the proposed approach can be applied to further problems to test its performance on other optimization problems. The method has a high potential to be helpful for many real-world problems that have been difficult or impossible to solve so far due to a high model complexity. Moreover, the approach could be used to solve the stochastic LSFRP to overcome the difficulties related to the high number of scenarios.

Overall, this dissertation has shown the importance of including uncertainty into optimization problems in order to obtain solutions that perform well in real-world situations. The exact and heuristic approaches presented provide examples of how to deal with uncertainties and offer possibilities for application to other problems, which can be examined in further research. Especially, the scalable stochastic optimization approach provides an interesting framework that should be further explored. It provides a new approach to problems that were previously difficult to solve, especially for industry, which has access to a large amount of historical data needed to apply this approach. Stochastic optimization, in particular in the field of maritime logistics problems, still offers a great potential for research in order to obtain solutions increasingly better adapted to reality.

Bibliography

Agra A, Christiansen M, Delgado A, Hvattum LM (2015) A Maritime Inventory Routing Problem with Stochastic Sailing and Port Times. Computers & Operations Research 61:18–30

Agra A, Christiansen M, Hvattum LM, Rodrigues F (2016) A MIP based local search heuristic for a stochastic maritime inventory routing problem. In: International Conference on Computational Logistics, Springer, pp 18–34

Agra A, Christiansen M, Hvattum LM, Rodrigues F (2018) Robust Optimization for a Maritime Inventory Routing Problem. Transportation Science 52(3):509–525

Alvarez A, Cordeau JF, Jans R, Munari P, Morabito R (2020) Inventory Routing Under Stochastic Supply and Demand. Omega 102304

Andersson H, Christiansen M, Fagerholt K (2011) The Maritime Pickup and Delivery Problem with Time Windows and Split Loads. INFOR: Information Systems and Operational Research 49(2):79–91

Ansótegui C, Sellmann M, Tierney K (2009) A Gender-Based Genetic Algorithm for the Automatic Configuration of Algorithms. In: International Conference on Principles and Practice of Constraint Programming, Springer, pp 142–157

Ansótegui C, Malitsky Y, Samulowitz H, Sellmann M, Tierney K (2015) Model-Based Genetic Algorithms for Algorithm Configuration. In: 24th International Joint Conference on Artificial Intelligence, pp 733–739

April J, Glover F, Kelly JP, Laguna M (2003) Practical introduction to simulation optimization. In: Proceedings of the 35th conference on Winter Simulation: Driving Innovation, Winter Simulation Conference, pp 71–78

Arantes MdS, Arantes JdS, Toledo CFM, Williams BC (2016) A Hybrid Multi-Population Genetic Algorithm for UAV Path Planning. In: Proceedings of the Genetic and Evolutionary Computation Conference 2016, ACM, pp 853–860

Arslan AN, Papageorgiou DJ (2017) Bulk ship fleet renewal and deployment under uncertainty: A multi-stage stochastic programming approach. Transportation Research Part E: Logistics and Transportation Review 97:69–96

Artzner P, Delbaen F, Eber JM, Heath D (1999) Coherent measures of risk. Mathematical finance 9(3):203–228

Aydin N, Lee H, Mansouri SA (2017) Speed optimization and bunkering in liner shipping in the presence of uncertain service times and time windows at ports. European Journal of Operational Research 259(1):143–154

Azariadis P (2017) On using density maps for the calculation of ship routes. Evolving Systems 8(2):135–145

Bakkehaug R, Eidem ES, Fagerholt K, Hvattum LM (2014) A stochastic programming formulation for strategic fleet renewal in shipping. Transportation Research Part E: Logistics and Transportation Review 72:60–76

Ben-Tal A, El Ghaoui L, Nemirovski A (2009) Robust Optimization, vol 28. Princeton University Press

Berthold T, Heinz S, Lübbecke ME, Möhring RH, Schulz J (2010) A Constraint Integer Programming Approach for Resource-Constrained Project Scheduling. In: Int. Conf. on Integration of AI and OR Techniques in Constraint Programming, Springer, pp 313–317

Bianchi L, Dorigo M, Gambardella LM, Gutjahr WJ (2009) A survey on metaheuristics for stochastic combinatorial optimization. Natural Computing 8(2):239–287

Birge JR (1985) Decomposition and Partitioning Methods for Multistage Stochastic Linear Programs. Operations research 33(5):989–1007

Birge JR, Louveaux F (2011) Introduction to Stochastic Programming. Springer Science & Business Media

Brouer B, Dirksen J, Pisinger D, Plum C, Vaaben B (2013a) The Vessel Schedule Recovery Problem (VSRP) – A MIP model for handling disruptions in liner shipping. European Journal of Operational Research 224(2):362–374

Brouer BD, Alvarez JF, Plum CE, Pisinger D, Sigurd MM (2013b) A Base Integer Programming Model and Benchmark Suite for Liner-Shipping Network Design. Transportation Science 48(2):281–312

Brouer BD, Karsten CV, Pisinger D (2017) Optimization in liner shipping. 4OR 15(1):1–35

Calafiore GC, El Ghaoui L (2014) Optimization Models. Cambridge University Press

Cao Z, Guo H, Zhang J, Niyato D, Fastenrath U (2015) Finding the Shortest Path in Stochastic Vehicle Routing: A Cardinality Minimization Approach. IEEE Transactions on Intelligent Transportation Systems 17(6):1688–1702

Charnes A, Cooper WW (1959) Chance-Constrained Programming. Management science 6(1):73–79

Christiansen M, Nygreen B (2005) Robust Inventory Ship Routing by Column Generation. In: Column generation, Springer, pp 197–224

Christiansen M, Fagerholt K, Nygreen B, Ronen D (2013) Ship routing and scheduling in the new millennium. European Journal of Operational Research 228(3):467–483

Christiansen M, Hellsten E, Pisinger D, Sacramento D, Vilhelmsen C (2019) Liner shipping network design. European Journal of Operational Research

Conforti M, Cornuéjols G, Zambelli G, et al (2014) Integer Programming, vol 271. Springer

De A, Choudhary A, Turkay M, Tiwari MK (2021) Bunkering policies for a fuel bunker management problem for liner shipping networks. European Journal of Operational Research 289(3):927–939

De Wit C (1990) Proposal for Low Cost Ocean Weather Routeing. The Journal of Navigation 43(3):428–439

Dean BC, Goemans MX, Vondrák J (2008) Approximating the Stochastic Knapsack Problem: The Benefit of Adaptivity. Math Oper Res 33(4):945–964

Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE transactions on evolutionary computation 6(2):182–197

Du Y, Meng Q, Wang Y (2015) Budgeting Fuel Consumption of Container Ship over Round-Trip Voyage Through Robust Optimization. Transportation Research Record 2477(1):68–75

Dulebenets MA, Pasha J, Abioye OF, Kavoosi M (2019) Vessel scheduling in liner shipping: a critical literature review and future research needs. Flexible Services and Manufacturing Journal pp 1–64

Dupačová J, Consigli G, Wallace SW (2000) Scenarios for Multistage Stochastic Programs. Annals of operations research 100(1-4):25–53

El Noshokaty S (2018) Tramp Shipping Optimization: A Critical Review. Global Journal of Management and Business Research 18(1):1–13

Erdös P, Rényi A (1959) On random graphs, I. Publicationes Mathematicae (Debrecen) 6:290–297

Fábián CI (2008) Handling CVaR objectives and constraints in two-stage stochastic models. European Journal of Operational Research 191(3):888–911

Fu MC, Glover FW, April J (2005) Simulation optimization: a review, new developments, and applications. In: Proceedings of the Winter Simulation Conf., 2005, IEEE, pp 13–pp

Geisberger R, Sanders P, Schultes D, Delling D (2008) Contraction Hierarchies: Faster and Simpler Hierarchical Routing in Road Networks. In: International Workshop on Experimental and Efficient Algorithms, Springer, pp 319–333

Geisberger R, Sanders P, Schultes D, Vetter C (2012) Exact Routing in Large Road Networks Using Contraction Hierarchies. Transportation Science 46(3):388–404

Gendreau M, Potvin JY, et al (2010) Handbook of Metaheuristics, vol 2. Springer

Ghosh S, Lee LH, Ng SH (2015) Bunkering decisions for a shipping liner in an uncertain environment with service contract. European Journal of Operational Research 244(3):792–802

Glover F, Kelly J, Laguna M (1999) New advances for wedding optimization and simulation. In: Winter Simulation Conference 1999 Proceedings., IEEE, vol 1, pp 255–260

Google (2019) Google OR-Tools. URL `developers.google.com/optimization/`

Gu Y, Wallace SW, Wang X (2019) Integrated maritime fuel management with stochastic fuel prices and new emission regulations. Journal of the Operational Research Society 70(5):707–725

Gurobi Optimization (2020) Gurobi optimizer reference manual. `https://www.gurobi.com/wp-content/plugins/hd_documentations/documentation/9.0/refman.pdf`

Hagiwara H, Spaans J (1987) Practical Weather Routing of Sail-assisted Motor Vessels. The Journal of Navigation 40(1):96–119

Haltiner G, Hamilton H, ’Arnason G (1962) Minimal-Time Ship Routing. Journal of Applied Meteorology 1(1):1–7

Halvorsen-Weare EE, Fagerholt K, Rönnqvist M (2013) Vessel routing and scheduling under uncertainty in the liquefied natural gas business. Computers & Industrial Engineering 64(1):290–301

Hand M (2020) Fireworks, batteries and liquid ethanol among cargoes lost from ONE Apus. URL `https://www.seatrade-maritime.com/casualty/fireworks-batteries-and-liquid-ethanol-among-cargoes-lost-one-apus`, as of 19.04.2021

Hand M (2021) Maersk Eindhoven waiting to berth in Yokohama following container losses. URL `https://www.seatrade-maritime.com/casualty/maersk-eindhoven-waiting-berth-yokohama-following-container-losses`, as of 19.04.2021

Hasircioglu I, Topcuoglu HR, Ermis M (2008) 3-D Path Planning for the Navigation of Unmanned Aerial Vehicles by Using Evolutionary Algorithms. In: Proceedings of the 10th annual conference on Genetic and evolutionary computation, pp 1499–1506

Hochreiter R, Pflug GC (2007) Financial scenario generation for stochastic multi-stage decision processes as facility location problems. Annals of OR 152(1):257–272

Hoffmann J, Sirimanne SN (2017) Review of maritime transport 2017. United Nations Publication

Holland JH, et al (1992) Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press

James RW (1957) Application of wave forecasts to marine navigation. US Navy Eydrographic Office, Washington DC SP-1

Jin Y, Branke J (2005) Evolutionary Optimization in Uncertain Environments-A Survey. IEEE Transactions on evolutionary computation 9(3):303–317

Kall P, Wallace SW, Kall P (1994) Stochastic Programming. Springer

Kaut M, Wallace S (2003) Evaluation of Scenario-Generation Methods for Stochastic Programming. Pacific Journal of Optimization 3

Kaut M, Wallace SW (2007) Evaluation of Scenario-Generation Methods for Stochastic Programming. Pacific Journal of Optimization 3(2):257–271

Kepaptsoglou K, Fountas G, Karlaftis MG (2015) Weather impact on containership routing in closed seas: A chance-constraint optimization approach. Transportation Research Part C: Emerging Technologies 55:139–155

Kisialiou Y, Gribkovskaia I, Laporte G (2019) Supply vessel routing and scheduling under uncertain demand. Transportation Research Part C: Emerging Technologies 104:305–316

Kleywegt AJ, Shapiro A, Homem-de Mello T (2002) The sample average approximation method for stochastic discrete optimization. SIAM Journal on Optimization 12(2):479–502

Knudsen AN, Chiarandini M, Larsen KS (2017) Constraint Handling in Flight Planning. In: International Conference on Principles and Practice of Constraint Programming, Springer, pp 354–369

Knudsen AN, Chiarandini M, Larsen KS (2018) Heuristic Variants of A* Search for 3D Flight Planning. In: International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research, Springer, pp 361–376

Kolisch R, Sprecher A (1997) PSPLIB–a project scheduling problem library. European Journal of Operational Research 96(1):205–216

Krata P, Szlapczynska J (2012) Weather Hazard Avoidance in Modeling Safety of Motor-Driven Ship for Multicriteria Weather Routing. Methods and Algorithms in Navigation Marine navigation and safety of sea transportation, Weintrit and Neuman (ed), Gdynia Maritime University 6(1)

Kuhlemann S, Tierney K (2020) A genetic algorithm for finding realistic sea routes considering the weather. Journal of Heuristics 26(6):801–825

Kwon Y (2008) Speed loss due to added resistance in wind and waves. Naval Architect pp 14–16

Larsson E, Simonsen MH (2014) DIRECT Weather Routing. Master's thesis, Chalmers University of Technology, Gothenburg

Levinson M (2016) The Box: How the Shipping Container Made the World Smaller and the World Economy Bigger. Princeton University Press

Li C, Qi X, Lee CY (2015) Disruption Recovery for a Vessel in Liner Shipping. Transportation Science 49(4):900–921

Li C, Qi X, Song D (2016) Real-time schedule recovery in liner shipping service with regular uncertainties and disruption events. Transportation Research Part B: Methodological 93:762–788

Li G, Zhang H (2017) A Bézier Curve Based Ship Trajectory Optimization for Close-Range Maritime Operations. In: ASME 2017 36th International Conference on Ocean, Offshore and Arctic Engineering, American Society of Mechanical Engineers, paper V07BT06A026

Li X, Wang H, Wu Q (2017) Multi-objective Optimization in Ship Weather Routing. In: Constructive Nonsmooth Analysis and Related Topics (dedicated to the memory of VF Demyanov)(CNSA), 2017, IEEE, pp 1–4

Löhndorf N (2016) An empirical analysis of scenario generation methods for stochastic optimization. European Journal of Operational Research 255(1):121–132

Long Y, Lee LH, Chew EP (2012) The sample average approximation method for empty container repositioning with uncertainties. European Journal of Operational Research 222(1):65–75

Luo X, Dashora Y, Shaw T (2015) Airline crew augmentation: Decades of improvements from sabre. INFORMS Journal on Applied Analytics 45(5):409–424

Madkour A, Aref WG, Rehman FU, Rahman MA, Basalamah S (2017) A Survey of Shortest-Path Algorithms. arXiv:1705.02044

Maki A, Akimoto Y, Nagata Y, Kobayashi S, Kobayashi E, Shiotani S, Ohsawa T, Umeda N (2011) A new weather-routing system that accounts for ship stability based on a real-coded genetic algorithm. Journal of marine science and technology 16(3):311

Malitsky Y, Sabharwal A, Samulowitz H, Sellmann M (2013) Algorithm Portfolios Based on Cost-Sensitive Hierarchical Clustering. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI), Beijing, China, 2013, pp 608–614

Mannarini G, Pinardi N, Coppini G, Oddo P, Iafrati A (2016) VISIR-I: small vessels–least-time nautical routes using wave forecasts. Geoscientific Model Development 9(4):1597–1625

Maximal Software (2016) MPL manual. `http://www.maximalsoftware.com/mplman/`

Meng Q, Wang S (2012) Liner ship fleet deployment with week-dependent container shipment demand. European Journal of Operational Research 222(2):241–252

Meng Q, Wang T (2010) A chance constrained programming model for short-term liner ship fleet planning problems. Marit Pol Mgmt 37(4):329–346

Meng Q, Wang T, Wang S (2012) Short-term liner ship fleet planning with container transshipment and uncertain container shipment demand. European Journal of Operational Research 223(1):96–105

Meng Q, Wang S, Andersson H, Thun K (2014) Containership Routing and Scheduling in Liner Shipping: Overview and Future Research Directions. Transportation Science 48(2):265–280

Meng Q, Wang T, Wang S (2015a) Multi-period liner ship fleet planning with dependent uncertain container shipment demand. Maritime Policy & Management 42(1):43–67

Meng Q, Wang Y, Du Y (2015b) Bunker Procurement Planning for Container Liner Shipping Companies: Multistage Stochastic Programming Approach. Transportation Research Record 2479(1):60–68

Montes AA (2005) Network shortest path application for optimum track ship routing. Master's thesis, Monterey, California. Naval Postgraduate School

Motte R, Calvert S (1990) On The Selection of Discrete Grid Systems for On-Board Micro-based Weather Routeing. The Journal of Navigation 43(1):104–117

Ng M (2015) Container vessel fleet deployment for liner shipping with stochastic dependencies in shipping demand. Transportation Research Part B: Methodological 74:79–87

Notteboom TE (2006) The Time Factor in Liner Shipping Services. Maritime Economics & Logistics 8(1):19–39

Pache H, Kastner M, Jahn C (2019) Current State and Trends in Tramp Ship Routing and Scheduling. In: Digital Transformation in Maritime and City Logistics: Smart Solutions for Logistics. Proceedings of the Hamburg International Conference of Logistics (HICL), Vol. 28, Berlin: epubli GmbH, pp 369–394

Pacino D, Jensen RM (2012) Fast Generation of Container Vessel Stowage Plans. PhD thesis, IT University of Copenhagen

Panigrahi J, Padhy C, Sen D, Swain J, Larsen O (2012) Optimal ship tracking on a navigation route between two ports: a hydrodynamics approach. Journal of marine science and technology 17(1):59–67

Pearce RH, Tyler A, Forbes M (2016) Column Generation and Lazy constraints for solving the Liner Ship Fleet Repositioning Problem with cargo flows. arXiv preprint arXiv:160302384

Pelikan M, Goldberg DE, Cantú-Paz E (1999) BOA: The Bayesian Optimization Algorithm. In: Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 1, Morgan Kaufmann Publishers Inc., pp 525–532

Perakis A, Jaramillo D (1991) Fleet deployment optimization for liner shipping Part 1. Background, problem formulation and solution approaches. Maritime Policy and Management 18(3):183–200

Powell B, Perakis A (1997) Fleet Deployment Optimization for Liner Shipping: An Integer Programming Model. Maritime Policy and Management 24(2):183–192

Puchinger J, Raidl GR (2005) Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification. In: International work-conference on the interplay between natural and artificial computation, Springer, pp 41–53

Qi X, Song DP (2012) Minimizing fuel emissions by optimizing vessel schedules in liner shipping with uncertain port times. Transportation Research Part E: Logistics and Transportation Review 48(4):863–880

Ragusa VR, Mathias HD, Kazakova VA, Wu AS (2017) Enhanced Genetic Path Planning for Autonomous Flight. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp 1208–1215

Rockafellar RT, Uryasev S (2002) Conditional Value-at-Risk for General Loss Distributions. Journal of banking & finance 26(7):1443–1471

Rockafellar RT, Uryasev S, et al (2000) Optimization of Conditional Value-at-Risk. Journal of risk 2:21–42

Rodrigue JP, Notteboom T (2015) Looking inside the box: evidence from the containerization of commodities and the cold chain. Maritime Policy & Management 42(3):207–227

Rodrigues F, Agra A, Christiansen M, Hvattum LM, Requejo C (2019) Comparing techniques for modelling uncertainty in a maritime inventory routing problem. European Journal of Operational Research 277(3):831–845

dos Santos Diz GS, Hamacher S, Oliveira F (2019) A robust optimization model for the maritime inventory routing problem. Flexible Services and Manufacturing Journal 31(3):675–701

Schütz P, Tomasgard A, Ahmed S (2009) Supply chain design under uncertainty using sample average approximation and dual decomposition. European Journal of Operational Research 199(2):409–419

Sen D, Padhy CP (2015) An approach for development of a ship routing algorithm for application in the North Indian Ocean region. Applied Ocean Research 50:173–191

Sheng X, Lee LH, Chew EP (2014) Dynamic determination of vessel speed and selection of bunkering ports for liner shipping under stochastic environment. OR spectrum 36(2):455–480

Soroush H, Al-Yakoob S (2018) A maritime scheduling transportation-inventory problem with normally distributed demands and fully loaded/unloaded vessels. Applied Mathematical Modelling 53:540–566

Stålhane M, Andersson H, Christiansen M, Cordeau J, Desaulniers G (2012) A branch-price-and-cut method for a ship routing and scheduling problem with split loads. Computers & Operations Research

Szłapczyńska J (2007) Multiobjective Approach to Weather Routing. TransNav, International Journal on Marine Navigation and Safety of Sea Transportation 1(3):273–278

Szlapczynska J (2013) Multicriteria Evolutionary Weather Routing Algorithm in Practice. TransNav: International Journal on Marine Navigation and Safety of Sea Transportation 7(1):61–65

Szlapczynska J (2015) Multi-objective Weather Routing with Customised Criteria and Constraints. The Journal of Navigation 68(2):338–354

Szłapczynska J, Smierzchalski R (2009) Multicriteria optimisation in weather routing. Marine Navigation and Safety of Sea Transportation 3:423

The Baltic and International Maritime Council, et al (2011) Best Management Practices for Protection Against Somalia Based Piracy. Edinburgh: Witherby 4

Tierney K (2015) Optimizing Liner Shipping Fleet Repositioning Plans. Springer

Tierney K, Jensen R (2013) A Node Flow Model for the Inflexible Visitation Liner Shipping Fleet Repositioning Problem with Cargo Flows. In: Pacino D, Voß S, Jensen R (eds) Computational Logistics, LNCS, vol 8197, Springer Berlin Heidelberg, pp 18–34

Tierney K, Áskelsdóttir B, Jensen R, Pisinger D (2014) Solving the Liner Shipping Fleet Repositioning Problem with Cargo Flows. Transportation Science 49(3):652–674

Tierney K, Handali J, Grimme C, Trautmann H (2017) Multi-Objective Optimization for Liner Shipping Fleet Repositioning. In: International Conference on Evolutionary Multi-Criterion Optimization, Springer, pp 622–638

Tierney K, Ehmke JF, Campbell AM, Müller D (2019) Liner Shipping Single Service Design Problem with Arrival Time Service Levels. Flexible Services and Manufacturing Journal pp 1–33

Touati N, Jost V (2012) On green routing and scheduling problem. arXiv:1203.1604

Tsang HT, Mak HY (2015) Robust Optimization Approach to Empty Container Repositioning in Liner Shipping. In: Handbook of ocean container transport logistics, Springer, pp 209–229

Tsou MC (2010) Integration of a Geographic Information System and Evolutionary Computation for Automatic Routing in Coastal Navigation. The Journal of Navigation 63(2):323–341

Tsou MC, Cheng HC (2013) An Ant Colony Algorithm for efficient ship routing. Polish Maritime Research 20(3):28–38

United Nations Conference on Trade and Development (2019) Review of Maritime Transport 2019. United Nations Publications

United Nations Conference on Trade and Development (2020) Review of Maritime Transport 2020. United Nations Publications

Van Slyke RM, Wets R (1969) L-Shaped Linear Programs with Applications to Optimal Control and Stochastic Programming. SIAM Journal on Applied Mathematics 17(4):638–663

Vanderbei RJ, et al (2015) Linear Programming, vol 3. Springer

Veneti A, Konstantopoulos C, Pantziou G (2015) An evolutionary approach to multi-objective ship weather routing. In: Information, Intelligence, Systems and Applications (IISA), 2015 6th International Conference on, IEEE, pp 1–6

Veneti A, Konstantopoulos C, Pantziou G (2018) Evolutionary Computation for the Ship Routing Problem. In: Modeling, Computing and Data Handling Methodologies for Maritime Transportation, Intelligent Systems Reference Library, Springer, Cham, pp 95–115

Verweij B, Ahmed S, Kleywegt AJ, Nemhauser G, Shapiro A (2003) The Sample Average Approximation Method Applied to Stochastic Routing Problems: A Computational Study. Computational Optimization and Applications 24(2-3):289–333

Vettor R, Guedes Soares C (2016) Development of a ship weather routing system. Ocean Engineering 123(Supplement C):1–14

Walther L, Shetty S, Rizvanolli A, Jahn C (2018) Comparing Two Optimization Approaches for Ship Weather Routing. In: Operations Research Proceedings 2016, Operations Research Proceedings, Springer, pp 337–342

Wang HB, Li XG, Li PF, Veremey EI, Sotnikova MV (2018a) Application of Real-Coded Genetic Algorithm in Ship Weather Routing. The Journal of Navigation 71:1–22

Wang S, Meng Q (2012a) Liner ship route schedule design with sea contingency time and port time uncertainty. Transportation Research Part B: Methodological 46(5):615–633

Wang S, Meng Q (2012b) Robust schedule design for liner shipping services. Transportation Research Part E: Logistics and Transportation Review 48(6):1093–1106

Wang T, Meng Q, Wang S (2012) Robust Optimization Model for Liner Ship Fleet Planning with Container Transshipment and Uncertain Demand. Transportation research record 2273(1):18–28

Wang Y, Meng Q, Kuang H (2018b) Jointly optimizing ship sailing speed and bunker purchase in liner shipping with distribution-free stochastic bunker prices. Transportation Research Part C: Emerging Technologies 89:35–52

Wetzel D, Tierney K (2020) Integrating fleet deployment into liner shipping vessel repositioning. Transportation Research Part E: Logistics and Transportation Review 143:102,101

Wingrove M (2021) Weather routeing reduces container loss risk. URL `https://www.rivieramm.com/news-content-hub/weather-routeing-reduces-container-loss-risk-62668`, as of 19.04.2021

Wong EY, Tai AH, Raman M (2015) A maritime container repositioning yield-based optimization model with uncertain upsurge demand. Transportation Research Part E: Logistics and Transportation Review 82:147–161

Wrede I (2013) Hohe Kosten für Reeder durch Piraterie. URL `https://p.dw.com/p/18XHu`, as of 19.04.2021

Xing J, Zhong M (2017) A reactive container rerouting model for container flow recovery in a hub-and-spoke liner shipping network. Maritime Policy & Management 44(6):744–760

Yang Z, Chen D (2017) Robust optimisation of liner shipping network on Yangtze River with considering weather influences. International Journal of Shipping and Transport Logistics 9(5):626–639

Yen JY (1970) An algorithm for finding shortest routes from all source nodes to a given destination in general networks. Quarterly of Applied Mathematics 27(4):526–530

Yuankui L, Yingjun Z, Feixiang Z (2014) Minimal Time Route for Wind-Assisted Ships. Marine Technology Society Journal 48(3):115–124

Zhang C, Nemhauser G, Sokol J, Cheon MS, Keha A (2018) Flexible Solutions to Maritime Inventory Routing Problems with Delivery Time Windows. Computers & Operations Research 89:153–162

Zhen L, Hu Y, Wang S, Laporte G, Wu Y (2019) Fleet deployment and demand fulfillment for container shipping liners. Transportation Research Part B: Methodological 120:15–32

Zitzler E, Thiele L (1999) Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. IEEE transactions on Evolutionary Computation 3(4):257–271

Zitzler E, Laumanns M, Thiele L (2001) SPEA2: Improving the Strength Pareto Evolutionary Algorithm. TIK-report 103

CURRICULUM VITAE

PERSONAL DATA

Address: Pohlweg 74, 33098 Paderborn, Germany
Email: stefan.kuhlemann@uni-bielefeld.de

EDUCATION

2017 - 2021	**Ph.D., Bielefeld University**, Bielefeld, Germany Doctorate in Business Information Systems (Dr. rer. pol.) Topic: "Stochastic Optimization for Maritime Logistic Problems" Advisors: Prof. Kevin Tierney, Prof. Achim Koberstein
2015 - 2017	**M.Sc., Paderborn University**, Paderborn, Germany Degree: Master of Science Business Information Systems Advisor: Prof. Kevin Tierney Final grade: 1.2 (1.0 is the best, 4.0 is the worst)
2012 - 2015	**B.Sc., Paderborn University**, Paderborn, Germany Degree: Bachelor of Science Business Information Systems Final grade: 1.6 (1.0 is the best, 4.0 is the worst)

ACADEMIC EMPLOYMENT

2018 - 2021	**Research assistant** Bielefeld University, Bielefeld, Germany Chair of Decision and Operation Technologies Faculty of Business Administration and Economics
2018 - 2021	**Research assistant** Paderborn University, Paderborn, Germany Lehrstuhl für Management Information Systems and Operations Research Faculty of Business Administration and Economics
2017 - 2018	**Research assistant** Paderborn University, Paderborn, Germany Chair of Decision Support & Operations Research Lab Faculty of Business Administration and Economics

2017 - 2018	**Research assistant** Europa-Universität Viadrina, Frankfurt (Oder), Germany Lehrstuhl für Betriebswirtschaftslehre, insbesondere Business Informatics & Operations Research Faculty of Business Administration and Economics
2015 - 2017	**Student assistant** Paderborn University, Paderborn, Germany Chair of Decision Support & Operations Research Lab Faculty of Business Administration and Economics

Publications

2021	Kuhlemann, S., Ksciuk, J., Tierney, K., Koberstein, A., The Stochastic Liner Shipping Fleet Repositioning Problem with Uncertain Container Demands and Travel Times. EURO Journal on Transportation and Logistics, 100052.
2020	Kuhlemann S., Tierney K., A Genetic Algorithm for Finding Realistic Sea Routes Considering the Weather, Journal of Heuristics, 26(6), pp. 801-825.
2019	Kuhlemann, S., Sellmann, M., Tierney, K., Exploiting Counterfactuals for Scalable Stochastic Optimization, International Conference on Principles and Practice of Constraint Programming, Springer, Cham, pp. 690-708.
2018	Hallmann C., Kuhlemann S., Model Generator for Water Distribution Systems (2018), Operations Research Proceedings 2017, Springer Cham, pp. 245-251.

Presentations

September 2019	Operations Research 2019, Dresden, Germany A Genetic Algorithm for Finding Realistic Sea Routes Considering the Weather
June 2019	30th European Conference on Operational Research, Dublin, Ireland A Genetic Algorithm for Finding Realistic Sea Routes Considering the Weather
October 2018	International Conference on Computational Logistics 2018, Salerno, Italy A Genetic Algorithm for Finding Realistic Routes for Vessels under the Consideration of Weather Conditions

July 2018	29th European Conference on Operational Research, Valencia, Spain The Stochastic Liner Shipping Fleet Repositioning Problem with Uncertain Container Demands
December 2017	LDE Winter School on Optimization in Ports and Shipping Networks, Rotterdam, Netherlands Examining the Influence of Stochastic Optimization on Liner Shipping Fleet Repositioning Problems with Uncertain Demands
September 2017	Operations Research 2017, Berlin, Germany The Stochastic Liner Shipping Fleet Repositioning Problem with Uncertain Container Demands