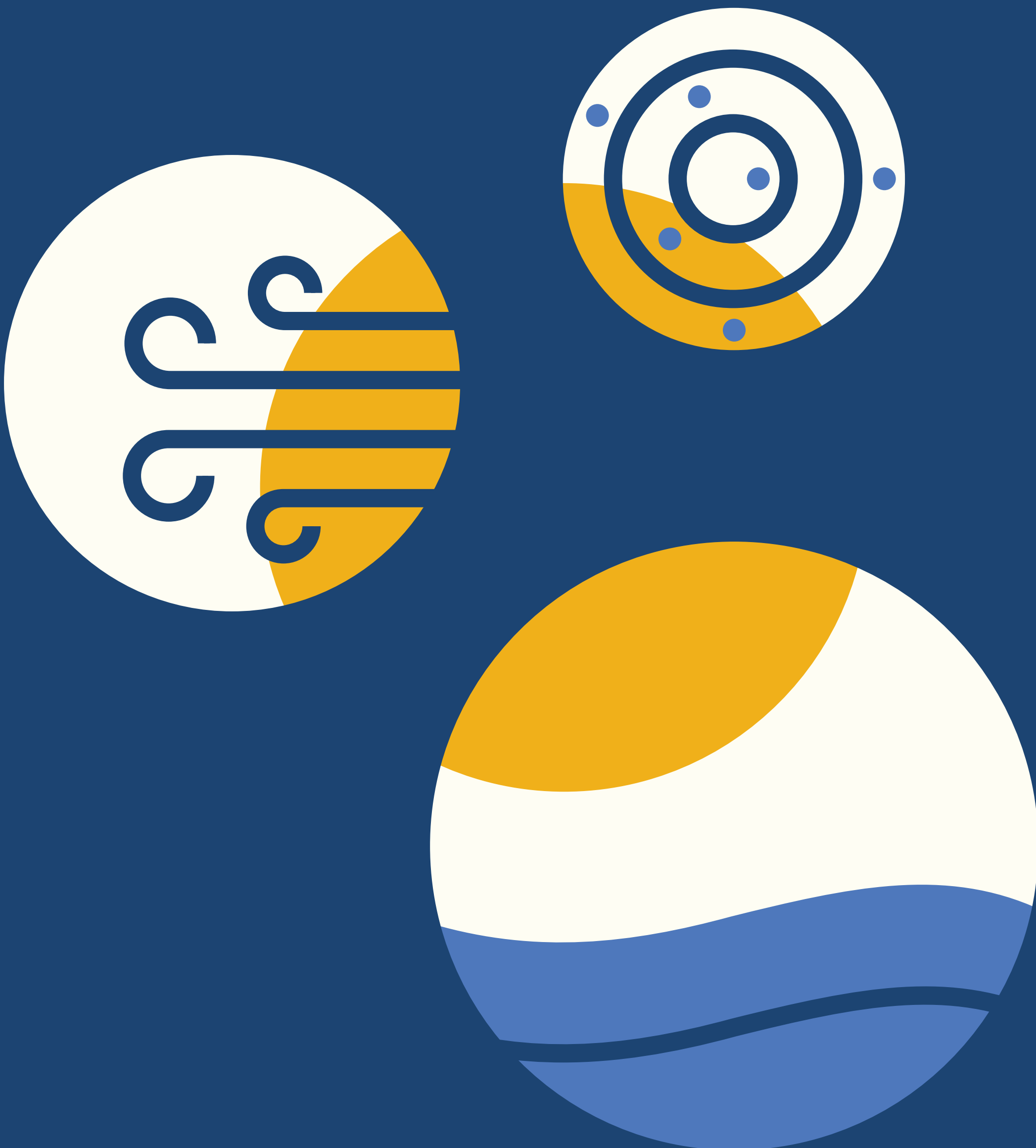
Mette Wagenvoort

Mathematical Models for Planning in Military and Humanitarian Logistics



Mathematical Models for Planning in Military and Humanitarian Logistics

Mathematical Models for Planning in Military and Humanitarian Logistics

Wiskundige modellen voor het plannen van militaire en humanitaire logistiek

Thesis

to obtain the degree of Doctor from the
Erasmus University Rotterdam
by command of the
Rector Magnificus

Prof.dr.ir. A.J. Schuit

and in accordance with the decision of the Doctorate Board.

The public defence shall be held on

Thursday 19 June 2025 at 10:30 hrs

by

Mette Wagenvoort
born in Goes, the Netherlands.

Erasmus University Rotterdam

Doctoral committee

Promotor: Prof.dr. D. Huisman

Other members: Prof.dr. M. Sevaux
Dr.ir. J.T. van Essen
Dr. W. van den Heuvel

Co-promotors: Dr. P.C. Bouman
Dr. M. van Ee

Erasmus Research Institute of Management - ERIM
The joint research institute of the Rotterdam School of Management (RSM)
and the Erasmus School of Economics (ESE) at the Erasmus University Rotterdam
Internet: www.erim.eur.nl

ERIM PhD Series in Research in Management, 601
ERIM reference number: EPS-2025-601-LIS
ISBN 978-90-5892-735-4

Cover design: van Kira | www.vanKira.nl/phd
Print: OBT bv, www.obt.eu

This publication (cover and interior) is printed in certified FSC® paper PlanoSuperior.

This research was made possible by TNO in collaboration with Erasmus University Rotterdam (EUR) and the Netherlands Defence Academy (NLDA).

MIX
Papier | Ondersteunt verantwoord bosbeheer
FSC® C022586
FSC www.fsc.org

TNO innovation for life

Acknowledgements

In September 2015, I started studying at Erasmus University Rotterdam. The completion of my PhD thus marks the end of a ten-year period at Erasmus University Rotterdam. During this time, I have met and worked with many people whom I would like to thank.

First, I want to thank my supervisors for giving me the opportunity to pursue a PhD. Dennis, due to the large number of supervisors, we did not have many discussion about research itself as you took on a higher-level supervision role. I appreciate the time you took to read and give feedback on some of my papers as well as on my final dissertation. Paul, I want to thank you for being willing to take over the supervision task as my EUR daily supervisor. I am also grateful for your time whenever I spontaneously popped into your office for a short question, which sometimes turned out to have a not so short answer. Kerry, thank you for taking over the supervision role from TNO during the first year of my PhD. I appreciate your useful comments on the papers, which greatly improved the writing and taught me some of my English pitfalls. Martijn, I really enjoyed our time discussing research and working out new ideas on the whiteboard in your office and during the breaks of the EURO conference in Copenhagen. These discussions were what I imagined doing research would be like.

Second, I would like to thank the inner committee, Marc Sevaux, Theresia van Essen, and Wilco van den Heuvel, for taking the time to read my dissertation, provide feedback, and attend my defence. Thank you to Herman Monsuur and Rommert Dekker for completing the doctoral committee.

My PhD was part of a collaboration between EUR, TNO and NLDA. While I spent most of my time working at the EUR (or at home during lockdowns), I also spent part of my time at TNO and NLDA. I want to thank everyone at the department of Modeling of Tactical Operations, and formerly the department of Military Operations, at TNO. I appreciate the time Rudi Gouweleeuw and Roy Benda spent talking about the resupply operation and providing feedback in our meetings.

I enjoyed the time spent at the NLDA and want to thank everyone for being so welcoming. Lander, I am happy we met during your internship at NLDA.

During my PhD, I went on a three-month research visit to INSEAD, Fontainebleau, where I worked with Iman Parsa and Luk Van Wassenhove. I want to thank them for this opportunity and their time. I learned a lot from working with you, and am still learning as we complete the paper. I also want to thank Abed, Bengisu, Boas, Clara, Eugenio, Francesco, Henrique, Lin, Priscilla, and Xabier for making me feel welcome in Fontainebleau and inviting me to join for foosball and El Classico.

My sincere thanks go to all colleagues at the Econometric Institute. After starting my PhD mostly at home due to lockdowns, it was nice to experience such a nice atmosphere at the office once those restrictions were lifted. Special thanks go to the (former) OR colleagues, Albert, Ece, Emö, Evelot, Hakan, Jehum, Noa, Olga, Remy, Rommert, Ruben, Sharon, Twan, and Wilco, for the engaging discussions about research, teaching, or non-work-related matters, as well as for all the feedback and advice I received. Christiaan, thank you for taking the time to give me feedback on the first lecture I taught for the course Simulation, the Bachelor Thesis supervision sessions, and for writing such a praising lecture observation for me. I also want to thank Evelien, Manne, and Marianne for their administrative and moral support.

Many thanks go to Shadi and Krzysztof. Shadi, we met in 2018 when I took the course Simulation. I remember you asked me to become a TA for you before I had even taken the exam. This demonstrates the trust you have put in my over the years. I enjoyed working with you, both on Simulation and on research when I was your research assistant. Krzysztof, you were initially supposed to be my daily supervisor, but your departure from the department meant this was not possible. Nevertheless, you took the time to continue working on the paper, especially helping me improve my writing. I want to thank both of you for all the time you have spent giving me feedback and advice, and I am grateful for the friendship that came out of this.

A special word of thanks goes to my fellow PhDs. I enjoyed the time we spent together at the office, during conferences, at PhD Game Night, as Econometric Winstitute at various pub quizzes, during the OR PhD dinners, and at the PhD vrijmibo's. Thank you, Aisha, Bart, Daniël, Danny, Khue, Lisanne, Marten, Mathijs, Maurice, Max, Pedro, Rick, Roby, Rolf, Rowan, Sam, Stan, Terri, Tobias, Utku, and Ymro.

I am thankful for all the friendships I have made and for the nice distractions from my PhD. Nienke, Lisa, and Tessa, I am thankful for the almost ten years we have been friends, and I hope to add many more quotes to my Dino's notes. Shuo, you were my default buddy for all assignments during our studies, and somehow I never had a hard time handing things over to you. Thank you for the fun times we had both during and after our studies. Myrthe and Sean, I want to thank you for all the fun dinners, times in Het Park, and the interesting conservations we had. Lara, I don't think I laughed with anyone as much as I did with you. Bart, David, Justin, Kjeld, and Thomas, I do not even remember how this group arose, but I am happy to be part of it and spend time with you. Bart, I am happy that, after spending our bachelor, master and PhD together, I can now finish my PhD with you as one of my paranimfs. Floor, Jeanne, Lea, Marit, and Marlou, although we were a bit *dispersed* throughout the country after secondary school, I am happy that we kept our friendship alive. Even though we sometimes did not see each other for a while, our friendship did not change. Marit, this summer we will have known each other for twenty years, and I am happy to have you by my side as one of my paranimfs.

Next, I want to thank my family. Afra, I always looked up to you and wanted to be like you when I was younger, which resulted in, amongst others, a really bad horse riding attempt. I realise having a younger sister that wanted to follow in your footsteps and whom people called a mini-version of you must have been quite annoying. I am happy that we get along so well now. Thijs, as my younger brother – I stopped calling you little brother quite some years ago – I always felt like I had to take care of you. It turns out you are perfectly capable of taking care of yourself. I enjoy spending time with you and you always make me laugh with your stories. Mom and dad, I want to thank you for supporting me in all the choices I have made, giving me advice when I need it, cheering me on and celebrating with me when I succeed at something, but also reassuring me it is okay to not be good at everything.

Riley, thank you for listening to me when I talk too much, making me brunch when I go for a run, laughing at my bad jokes (sometimes), wiping away my tears when I am sad, and supporting me in all choices I make.

Mette Wagenvoort
Rotterdam, March 2025

Table of contents

Chapter 1

Introduction

In the mid-19th century, postal rates in England were calculated based on the distance and the number of pages that were sent (National Postal Museum, n.d.). Moreover, mail could be sent with insufficient postage. Due to the high postal rates, demand for postal services was low and most mail was sent postage due, making the postal system expensive to operate. Charles Babbage analysed the efficiency of postal operations and showed that, given sufficiently high demand for postal services, the distance a letter travelled was negligible on the total costs of the postal operations (Sutherland, 2021). Babbage proposed a standardised postal rate based solely on the letter's weight. This recommendation led to the introduction of the 'Universal Penny Post' in 1840 in which the price was set to one penny for letters weighing up to half an ounce. This significantly increased the demand for postal services and made the postal system profitable.

While the term was not used yet, the mathematical analysis on the performance of the postal system by Babbage is an example of Operations Research (OR), in which mathematical models are used for decision-making. The field of OR was formally developed in World War II, when Patrick Blackett and his team of scientists – Blackett's Circus – studied military operations (Assad and Gass, 2011). They analysed, amongst others, the effect of the size of a merchant convoy on the probability of sinkage (Falconer, 1976). They found that the probability that a convoy was sighted did not depend strongly on the size of the convoy. Furthermore, they found that the probability of breaking through the protection of the escort vessels, depended only on the ratio of the number of escort vessels and the perimeter of the convoy that had to be protected, and that this perimeter grows less than linearly in the number of ships in the convoy. Therefore, while it was believed that smaller convoys were safer, Blackett suggested using fewer, but larger convoys.

Due to the recognition of the value of OR techniques in World War II, research on military operations continued. Harris and Ross, an American researcher and retired general, respectively, studied the railway network of the Soviet Union (Harris and Ross, 1955). The first documented study of this network dates back to 1930 when Tolstoı published an article on minimising the total distance required to transport resources using this railway network. A secret report released at the end of the 20th century revealed that Harris and Ross studied the railway network as a maximum flow problem (Schrijver, 2002). In this problem, the aim is to send as much flow through a network as possible, *i.e.*, transport as much cargo as possible, while adhering to capacity constraints. However, according to Ford and Fulkerson, the study's actual aim was to find the minimum cut, *i.e.*, the minimum number of train tracks that should be destroyed to prevent transport by the Soviet Union to Europe. Harris and Ross were the first to formulate the maximum flow problem which was used by Ford and Fulkerson to prove the well-known max-flow min-cut theorem (Ford and Fulkerson, 1956). This theorem states that the maximum flow in a network, *i.e.*, the largest amount of flow that can be sent from a source to a destination in a network without exceeding the capacities on the edges, equals the minimum cut, *i.e.*, the smallest total capacity of connections that, if removed, would disconnect the source from the destination (Ford and Fulkerson, 1956).

Military applications had an impact in the emergence of OR and remain an area of research as new problems arise due to, for example, changes in the equipment, technology, or strategies used by the military. After the success of OR techniques in military applications, their value was also recognised in other fields as a way to solve operational problems. OR techniques can be used to, for example, ensure your package is delivered on time, the supermarket is restocked with the right quantities, and all work shifts at a hospital are covered while meeting all labour agreements related to resting hours between shifts. While the problems in the private sector often optimise cost or profit, problems in military logistics consider effectiveness – achieving the desired results, readiness – ensuring deployment capability and capacity, and survivability – maintaining operational continuity. In addition to these different priorities, military problems often include scarce resources and unique constraints that do not exist in other applications.

A more recently developed field of OR is humanitarian logistics, which considers the preparedness and response to disasters. While OR techniques have been applied to problems in the private sector for many decades, the application of these techniques to humanitarian problems is relatively new (Van Wassenhove, 2006). Compared to problems in the private sector, humanitarian logistics faces challenges such as scarce resources and uncertainty about if and when a disaster will take place, the extent of the damage, and where help is needed. Furthermore, equity concerns are important in humanitarian logistics and therefore have to be considered in the decision-making process (Breugem et al., 2024).

Military and humanitarian logistics thus both consider effectively and efficiently using limited resources, possibly in unpredictable environments. They often involve unique constraints such as coordination or equity, that are not present in the private sector. As a result, the problems faced in military and humanitarian logistics differ significantly from those in the private sector. Solution approaches developed for (related) problems in the private sector can thus not be (directly) applied to problems in the military and humanitarian sector.

In this thesis, we consider various military and humanitarian optimisation problems. The remainder of this chapter is structured as follows. In Sections 1.1, 1.2, and 1.3, we introduce the three different optimisation problems considered in this thesis. In Section 1.4, we describe the set-up of the remainder of this thesis. The contributions and research statement are given in Sections 1.5 and 1.6, respectively.

1.1 The Ship-to-Shore Problem

In coastal areas, access from the sea may be the fastest or only viable way to deliver essential resources – personnel, vehicles, and supplies – for military or humanitarian operations on land. The navy plays a vital role by transporting these resources by large amphibious warfare ships. These large ships are not always able to reach the shore themselves. The resources require transport to the shore using smaller ships and helicopters, called *connectors*. Figure 1.1 illustrates this process, in which connectors make trips between a large amphibious warfare ship and the shore to deliver the resources. It is essential to deliver all resources to the shore as soon as possible to enable their use. Scheduling the transportation from the large amphibious warfare ship(s) to the shore is known as the *Ship-to-Shore Problem*.

Constructing a schedule for the Ship-to-Shore Problem is complex as the schedule has to adhere to various requirements. For example, the connectors have both space and weight capacities. We thus have to ensure that we assign sets of resources that

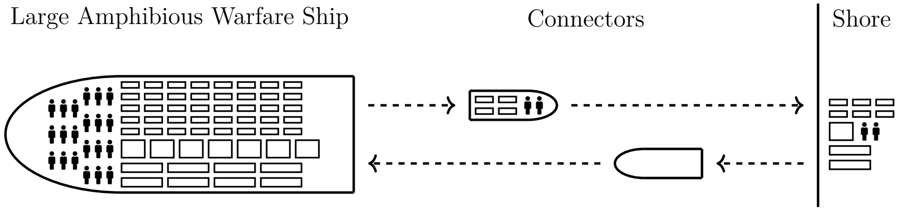


Figure 1.1: Illustration of the Ship-to-Shore Problem in which resources have to be transported from the large amphibious warfare ship on the left to the shore on the right. These resources are transported using small ships, called *connectors*.

can fit on the connector. The amphibious warfare ships have a limited number of loading spots and therefore a maximum number of connectors can be loaded at the same time. In our schedule, we can thus not schedule more connectors to be loaded at the same time, than the maximum number of available loading spots. Furthermore, not all connectors can be loaded from the same spot. A helicopter can for example only be loaded on the deck on top of the large amphibious warfare ship, from which small ships cannot be loaded.

In addition to requirements regarding the capacity of the connectors and large amphibious warfare ships, there can be requirements regarding the coordination of the deliveries. Resources can have different priority levels where resources with a lower priority can only be delivered once all resources with a higher priority have been delivered. Furthermore, resources can be complementary to each other and therefore should be delivered shortly after each other. For example, personnel and their vehicles should not be delivered too far apart.

A schedule for the Ship-to-Shore Problem should thus meet various requirements to be feasible in practice. Additionally, we aim to enable the operation on land to start as soon as possible. We are therefore interested in finding a feasible schedule that minimises the time it takes to transport all resources.

Based on such a schedule, preparations will be made. The resources on the large amphibious warfare ship are placed in a certain order and therefore block resources that are scheduled to be transported later. Also, personnel is potentially assigned other tasks until their scheduled departure time. This limits the ability to change the timing and order in which resources are transported. Therefore, delays in the trips of the connectors and/or the (un)loading of the connectors, can propagate through the schedule and result in large delays.

Our research focuses on both the construction of a schedule for the Ship-to-Shore Problem and the evaluation of their robustness against delays.

1.2 The Generalised Capacitated Resupply Problem

During military operations, it is important that units are supplied on time with essential commodities, such as food, fuel and medical supplies. Traditionally, resupply operations have focussed on supporting large, centralised units using motorised vehicles. Advancements in drone technology are opening up new possibilities for resupply operations, particularly for delivering smaller quantities of supplies. At the same time, developments in military strategies cause a shift toward smaller, more dispersed units. Dispersed operations introduce new logistical challenges since numerous small units should be resupplied, instead of resupplying a few centralised locations. A relevant question is whether there is sufficient logistical capacity to sustain these dispersed operations.

Besides military applications, this problem can also occur in humanitarian settings. For example, in the aftermath of a natural disaster, emergency shelters have to be resupplied with essential commodities such as food, water, and medical supplies, that arrive at a central depot.

We are interested in finding the minimum number of vehicles that are required to ensure none of the locations run out of stock. We consider locations that each have their own capacity and demand that depletes their stock. Initially, all locations are at full capacity. They are resupplied from a central depot by round-trips, *i.e.*, the vehicles used to resupply the locations go directly to the location and then back to the central depot. When a location is resupplied, its stock increases with the vehicle payload up to its capacity. The time required to resupply a location from the depot can differ and we assume the stock is delivered to the location at the end of the resupply time. The problem of finding a feasible resupply schedule, is called the *Generalised Capacitated Resupply Problem*.

Figure 1.2 shows a small example to illustrate this problem. Here, four locations, A, B, C, and D, should be resupplied by vehicles from a central depot. For each location, the capacity is given by the number of blocks in the adjacent bar, and the resupply time is given on the corresponding arc. All locations have a demand rate of 1, equal to one block in the adjacent bar. We see, for example, that location C has a very low capacity, while location B has a much higher capacity. Locations A and C should be resupplied in the next time period, as they will otherwise experience a stock-out. Figure 1.3 shows a feasible resupply schedule for the instance of Figure 1.2 with a vehicle payload of 4, *i.e.*, when a vehicle visits a location, the stock level

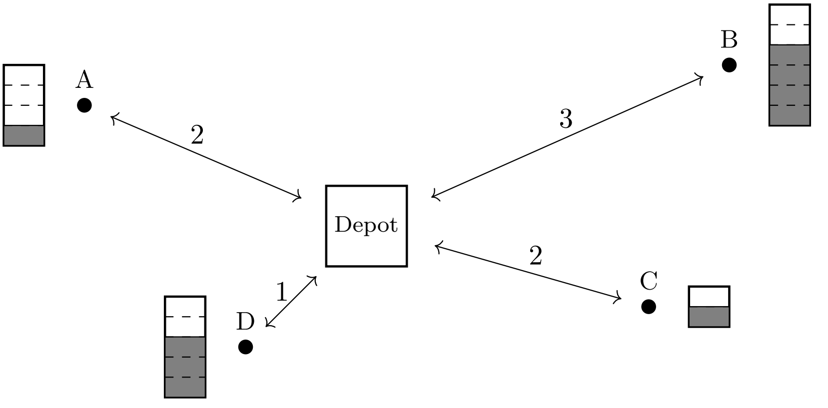


Figure 1.2: Illustration of an instance of the Generalised Capacitated Resupply Problem. Here, all locations – A, B, C and D – have a demand rate of 1. The capacity of a location is denoted by the number of blocks in the bar next to it. Each block corresponds to the demand for one period. The grey area shows the current stock level. The resupply times are given on the corresponding arc.

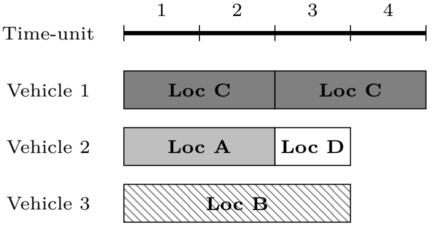


Figure 1.3: Example of a feasible schedule for the instance of the resupply problem in Figure 1.2 with a payload of 4. This schedule is repetitive, *i.e.*, after time period 4, the schedule is executed again. The stock levels in Figure 1.2 correspond to the stock level at the end of time period 5.

increases by 4, or up to its capacity. This schedule has a length of 4 time periods and it is thus repeated after 4 periods. Here, locations B and C are assigned a dedicated vehicle, while vehicle 2 is used to resupply both location A and D.

Our research focuses on developing simple policies that can be used to construct a feasible resupply schedule. We provide approximation guarantees on the number of required vehicles for these resupply schedules, *i.e.*, a worst-case guarantee on the quality of the resulting schedule.

1.3 The Value of Mobile Labs in Disease Surveillance

Disease surveillance, *i.e.*, the collection and testing of samples to identify and monitor the spread of pathogens, is crucial for reducing the impact of infectious diseases. Good surveillance enables rapid implementation of preventive measures, such as social distancing for air transmitted diseases and mosquito nets for malaria, thereby reducing the spread of the disease. However, low- and middle-income countries (LMICs) often face significant gaps in their surveillance infrastructure. These gaps can be addressed through investments in local health facilities or by deploying mobile laboratories (labs). Mobile labs, vans equipped with high-quality diagnostic equipment that allow for on-site testing, can travel between regions, thereby periodically improving the surveillance in multiple regions. While mobile labs appear to be a promising alternative to investments in the local health facilities, they are expensive and the budget in LMICs is limited. Therefore, we are interested in when and how much value mobile labs can bring by analysing when investments should be made in mobile labs and when in local health facilities themselves.

We consider a budget allocation problem in which we have to allocate a limited budget across multiple regions to improve surveillance from the perspective of a health ministry. In each region, the budget can be used to either invest in improving local health facilities, or in deploying mobile labs. Here, we aim to reduce the expected detection time, *i.e.*, the time at which a threshold for the number of positive tests is reached. Figure 1.4 illustrates the problem, in which the budget is allocated to four regions. The regions can differ in the quality of the local health facilities, represented by the difference in the size of the local health facilities, and in size/the population, represented by the figures. The budget should be allocated to the region, either to improve the local health facilities, or to deploy mobile labs. In this example, two of the regions receive investment to improve their local health facilities, while the other two regions are assigned mobile lab visits.

We analyse the problem both analytically and numerically. We are interested in the value of mobile labs, *i.e.*, how much additional reduction in detection time can be obtained by using mobile labs instead of considering investments in local health facilities. In public health resource allocation, equity is import. Inequitable allocations may lack the necessary support from regional stakeholders, which prevents them from being implemented. Therefore, we are interested in potential equity concerns that can arise in the optimal budget allocation and in the price of fairness, *i.e.*,

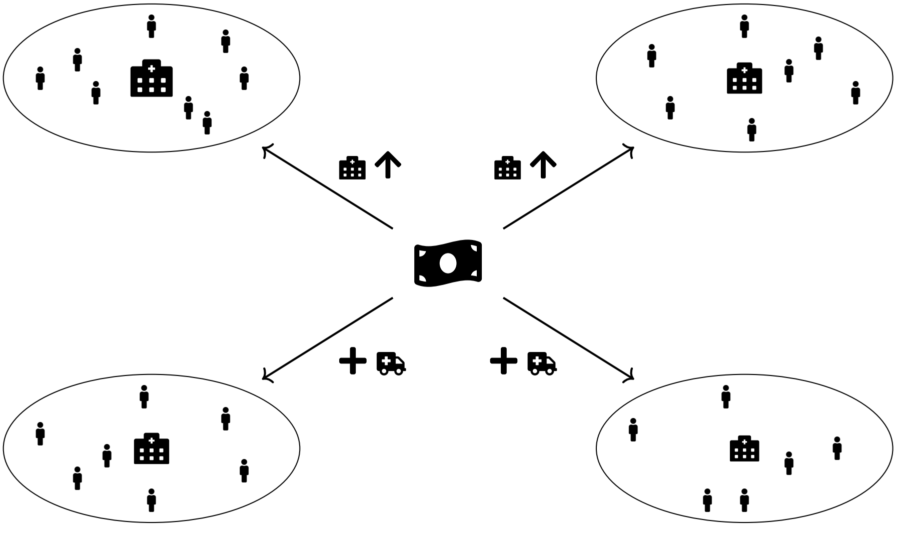


Figure 1.4: Illustration of the budget allocation problem in which a limited budget has to be distributed across multiple regions to improve surveillance through either investments in the local health facilities or the deployment of mobile laboratories.

the difference in the reduction in detection time between the optimal and equitable solutions.

1.4 Thesis Outline

This thesis is structured into six chapters. Figure 1.5 gives a schematic overview of the problems considered in Chapters 2-5. Chapters 2 and 3 consider the Ship-to-Shore Problem described in Section 1.1. In Chapter 2, we present an exact and a heuristic approach to solve the problem, *i.e.*, to find a schedule. In Chapter 3, we simulate the execution of the problem to analyse the robustness of the schedule as well as the effect of allowing connectors to depart a limited amount of time ahead of the scheduled time. Although these chapters can be read independently, we recommend reading these in order. In Chapter 4, we consider the resupply problem described in Section 1.2. We present policies for different versions of the problem to find feasible resupply schedules and give guarantees on the quality of the schedule. In Chapter 5, we consider the budget allocation problem introduced in Section 1.3. We end with a conclusion in Chapter 6.

In the remainder of this section, we describe Chapters 2 - 5 in more detail.

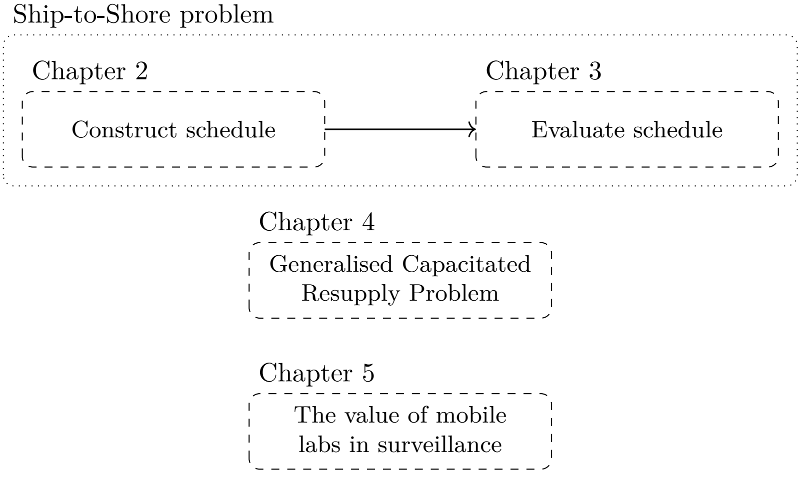


Figure 1.5: Schematic overview of Chapters 2-5 in this thesis.

Chapter 2: Wagenvoort, M., Bouman, P.C., van Ee, M., Lamballais Tessensohn, T., and Postek, K. (2025). "Exact and Heuristic Approaches for the Ship-to-Shore Problem". *European Journal of Operational Research, 320*(1), 115-131.

In this chapter, we consider the construction of schedules for the Ship-to-Shore Problem described in Section 1.1. In the Ship-to-Shore Problem, trips with their accompanying resources are determined while minimising the makespan, such that the operation on land can start as soon as possible. Limited (un)loading capacities, heterogeneous connector characteristics and constraints posed by priority of the resources and grouping of complementary resources (resource sets) all require that the connector trips are carefully coordinated. Despite the criticality of this coordination, existing literature does not consider resource sets and has only developed heuristics. We provide a formulation that incorporates resource sets and develop (i) an exact branch-and-price algorithm and (ii) a tailored greedy heuristic that can provide upper bounds using discretised time periods. We find that 84% of our 98 practical instances terminate within an hour. On average, these instances are solved in 80 seconds. Our greedy heuristic can find optimal solutions in two-thirds of these instances, mostly for instances that are very constrained in terms of the delivery order of resources. When improvements are found by the branch-and-price algorithm, the average gap with the makespan of the greedy solution is 40% and, in most cases, these improvements are obtained within three minutes. For the 20 artificial instances, the greedy heuristic has consistent performance on the different types of instances. For these artificial instances, improvements of on average 35% are found in reasonable time.

Chapter 3: Wagenvoort, M., Bouman, P., van Ee, M., and Malone, K.M. "Evaluating Ship-to-Shore Schedules using Simulation". *Journal of Defense, Modeling and Simulation (accepted for publication).*

In this chapter, we evaluate schedules for the Ship-to-Shore Problem described in Section 1.1 that are created assuming deterministic parameters regarding the speed and (un)loading time of the connectors. These schedules might therefore not be robust to delays. We developed a simulation model to analyse the effect of uncertainty in these parameters on the execution of a schedule. We analyse (i) whether these discrete time periods are able to capture the delays, (ii) the effect of using more conservative parameters when constructing a schedule, and (iii) the effect of being less rigid in the execution, *i.e.*, when being allowed to depart a limited amount of time ahead of schedule. We find that significant delays occur and that using more conservative parameters for the (un)loading time can have a positive significant effect on the duration of the operation. Being less rigid can also have a positive significant effect on the duration, however, being less rigid comes at the cost of violating constraints regarding the grouped delivery of complementary resources.

Chapter 4: Wagenvoort, M., Bouman, P., van Ee, M., and Malone, K.M. "Policies for the Generalised Capacitated Resupply Problem". *Under review.*

In this chapter, we study the Generalised Capacitated Resupply Problem (GCRP) described in Section 1.2, of which an earlier version is published in the ATMOS 2023 conference proceedings (Wagenvoort et al., 2023). In this problem, locations with a given capacity and demand rate should be resupplied by vehicles such that they do not run out of stock and the number of vehicles is minimised. Compared to related problems, we consider the scenario where the payload of the vehicles may not suffice to bring the stock level back to full capacity. In addition to the GCRP, we consider three variants of the problem in which the capacities and demand rates and/or the resupply times are homogeneous. We prove that the problem is NP-hard, even on one vehicle and with homogeneous capacities, and present policies to solve the different variants of the problem and provide corresponding approximation guarantees.

Chapter 5: Wagenvoort, M., Parsa, I., and Van Wassenhove, L. "Outbreak prevention in low- and middle-income countries: Investing in local health facilities or in mobile laboratories?". *In preparation for journal submission.*

In this chapter, we analyse the value of mobile laboratories (labs) in surveillance, as described in Section 1.3. Mobile labs are vans with high-quality diagnostic equipment that can perform fast on-site testing at various locations. We consider the perspective of a health ministry of a low- or middle-income country that has a limited budget available for improving surveillance of infectious diseases in multiple regions. For each region, the budget can be used to deploy mobile labs, or to invest in improving the surveillance by local health facilities. We formulate a budget allocation problem, where the objective is to optimally allocate the budget across the regions. We analyse the problem both analytically and numerically for both the case with identical regions and non-identical regions that differ in the current quality of their local health facilities. We analyse the value of mobile labs and potential equity concerns that can arise. We find that mobile labs can add significant value when the budget is tight and identify potential equity concerns. Even when regions are identical, inequity can arise. Furthermore, inverse inequity can arise where regions with currently higher quality local health facilities face longer detection times. However, when we consider equitable budget allocations, we find that mobile labs can still add significant value.

1.5 Contributions

In this section, we present the contributions of Chapters 2-5 separately.

Chapter 2: Wagenvoort, M., Bouman, P.C., van Ee, M., Lamballais Tessensohn, T., and Postek, K. (2025). “Exact and Heuristic Approaches for the Ship-to-Shore Problem”. *European Journal of Operational Research, 320*(1), 115-131.

We propose a formulation for the Ship-to-Shore Problem that enables coordination between complementary resources and show that the problem is NP-hard. We develop two solution approaches, one exact and one heuristic method, to solve the problem. We evaluate the performance of the solution approaches using both real-world and artificial instances. We analyse under which circumstances the heuristic works well and under which circumstances the exact solution approach finds significant improvements.

Chapter 3: Wagenvoort, M., Bouman, P., van Ee, M., and Malone, K.M. “Evaluating Ship-to-Shore Schedules using Simulation”. *Journal of Defense, Modeling and Simulation (accepted for publication).*

We develop a simulation model that can be used to evaluate the performance of schedules for the Ship-to-Shore Problem. The model accounts for uncertainties in the connector speed, (un)loading times, and changes in the weather conditions that impact the travel times. We analyse how well schedules constructed using discrete time periods, as in Chapter 2, perform in practice, and analyse the effect of using more conservative parameters. In practice, there is some rigidity in the execution of the schedule. Due to preparations made in advance, connectors are not able to depart ahead of schedule. We analyse the effect of this rigidity in the execution of the schedule and find that while it has a positive effect on the time required to transport all resources, it can result in violating requirements regarding the coordination between complementary resources.

Chapter 4: Wagenvoort, M., Bouman, P., van Ee, M., and Malone, K.M. "Policies for the Generalised Capacitated Resupply Problem". *In preparation for journal submission.*

We formally introduce the Generalised Capacitated Resupply Problem (GCRP) and variants of this problem, which differ based on whether locations have homogeneous or heterogeneous parameters. We present complexity results for the GCRP and two of its variants and present corresponding inapproximability results, *i.e.*, we present a lower bound on the approximation guarantee for polynomial time approximation algorithms. We describe policies and corresponding approximation guarantees for the GCRP and its variants.

Chapter 5: Wagenvoort, M., Parsa, I., and Van Wassenhove, L. "Outbreak prevention in low- and middle-income countries: Investing in local health facilities or in mobile laboratories?". *In preparation for journal submission.*

We analyse the value of mobile labs in improving surveillance, taking into account the alternative option of investments in the existing local health facilities. We incorporate an equity perspective. Not considering potential equity concerns can result in solutions that fail the required support of stakeholders and is often overlooked in research.

1.6 Research Statement

This research was made possible by TNO in collaboration with Erasmus University Rotterdam (EUR) and the Netherlands Defence Academy (NLDA). The author of this thesis is the main contributor to all chapters in this thesis, which includes model/solution method development, implementation, and writing. Chapter 2 is written in collaboration with Paul Bouman (EUR), Martijn van Ee (NLDA), Tim Lamballais Tessensohn (TNO), and Krzysztof Postek (independent researcher). Chapter 3 and 4 are written in collaboration with Paul Bouman (EUR), Martijn van Ee (NLDA), and Kerry Malone (TNO). Chapter 5 is written in collaboration with Iman Parsa (Stockholm School of Economics) and Luk Van Wassenhove (INSEAD).

Chapter 2

Exact and Heuristic Approaches for the Ship-to-Shore Problem

This chapter is based on Wagenvoort et al. (2025a).

2.1 Introduction

After a natural disaster such as a hurricane, the navy can provide aid by bringing supplies, helping to clear roads, and evacuating victims. In case of coastal areas, the navy provides support by transporting supplies from ships to the shore using smaller ships and helicopters, called *connectors*. For example, the US military delivers supplies in relief missions through a floating dock from which smaller and lighter vessels make deliveries to the pier (Debusmann, 2024). This has to be done efficiently for the help on land to start as soon as possible. As the planning of such an operation may depend on various situational parameters, such as weather conditions, the planning of such an operation has to be done fast. In addition to humanitarian purposes, this problem can be encountered in other military operations such as assault, withdrawal, raid, or support of other operations (Maritime Warfare Centre, 2019).

Planning such an operation is known as the Ship-to-Shore Problem, which is a type of transportation problem. In the Ship-to-Shore Problem, connectors pick-up resources, such as personnel and vehicles, from ships, and deliver them to the shore. Hence, it can be seen as a pick-up and delivery vehicle routing problem (PDVRP) (Zachariadis et al., 2016). However, large naval operations typically require the coordinated delivery of various types of resources, which we identified based on various interviews with experts on military operations at the Defence, Safety & Security unit of the Netherlands Organisation for Applied Scientific Research (TNO), a Dutch national research institute. To ensure this coordinated delivery, additional decisions on how to load the heterogeneous resources on the connectors are needed. Compared to the traditional PDVRP, the Ship-to-Shore Problem focuses on including the various types of coordination required within large scale operations, while reducing the movement of connectors to round-trips between the various loading and unloading spots at the ships and the shore.

One way in which coordination between the deliveries of the resources is imposed is through priority levels. Resources with a lower priority can only be delivered after items with a higher priority have arrived on the shore. This helps ensure that the command-and-control structure remains clear and ensures there are clear phases in the execution of the plan. Between these phases, resources should not be mixed, as it is imposed that preparatory measures should be completed before expensive equipment can be safely deployed. Strict priority orderings can also exist in other related problems. For example, consider the installation of a wind farm where resources should be brought from the shore to the sea. Here it is preferable to deliver expensive components only after the foundation is completed. Another way coordination is imposed is by requiring groups of resources to be delivered at the same time or immediately after each other. Namely, units may have trained with particular vehicles and other units. They can be temporarily separated while being transported, but should be able to work together on the shore. These types of coordination do not exist in the PDVRP. Despite their importance in practical operations, these coordinating constraints have not been considered in prior research on the Ship-to-Shore Problem (Christafore Jr., 2017; Danielson, 2018; Strickland, 2018).

The aim of the Ship-to-Shore Problem to minimise the duration of the operation, such that all resources are on land as soon as possible. In the operation, connectors are loaded at a ship, called the sea base (SB), after which the resources are transported to a landing area (LA) to be unloaded such that the connector can return to a (possibly different) SB for its next trip. Note that only one SB and one LA is visited in each trip for safety reasons. The SBs and LAs accommodate different types of connectors. For example, helicopters can only be loaded at a landing platform on the deck of the ship, while surface connectors cannot use this landing platform. Connectors also have different dimensions, fuel capacities, fuel consumption rates, speeds and weight capacities. Furthermore, the speed of a connector depends on its design speed, whether it is loaded or not, and the state of the sea, *i.e.*, the wind and the waves.

The Ship-to-Shore Problem can include various operational constraints that require coordination between the schedules of the different connectors. Our problem formulation has three such constraints. Firstly, there is limited (un)loading capacity, as there is a limited number of (un)loading spots, putting a constraint on the number of connectors that can be (un)loaded at the same time. Secondly, we consider a heterogeneous fleet of connectors with varying speeds, fuel capacity and consumption, and dimensions, affecting the set of resources that can be transported simultaneously. Thirdly, we require coordination between the delivery of the resources in the two ways described before. Namely, we consider priority levels and groups of resources, called *resource sets*, that should be delivered together. The time period in which the resources from a resource set are delivered is called a *delivery wave* for that resource set. There is no ordering imposed between the different waves with the same priority and these delivery waves can (partially) overlap. Note that this implies that the minimum number of connectors should be such that each resource set can be delivered using each connector at most once.

An input to this problem are the ways in which connectors can be loaded during a trip. In the ship-to-shore application, there are very specific practical constrains that are very challenging to incorporate in the model. We therefore focus on optimising the transportation schedule and take the ways connectors can be loaded as input.

The main contributions of this chapter are as follows. First, we develop a formulation for the Ship-to-Shore Problem that allows for coordination between the resources, and we prove the Ship-to-Shore Problem to be NP-hard. We then develop two solution methods: An exact branch-and-price algorithm in which heuristic pricing is used in combination with an exact pricing method, and a tailored greedy heuristic that can also be used as an upper bound in the formulation and algorithm. A branch-and-price algorithm is used as preliminary results showed that the integrated problem did not work well. We conduct computational experiments with instances from practice to show that the branch-and-price algorithm is able to solve the majority of these instances within an hour. Finally, we investigate under which circumstances which method is preferred. We observe that the greedy algorithm performs particularly well on instances where resource set constraints exist, but no priorities are defined. We furthermore observe that the exact method has the highest potential to improve solutions when resource set constraints are not present. We additionally use 20 artificial instances to compare with our practical instances. These show that for general instances there is no difference in the performance of the greedy heuristic. The artificial instances confirm the ability of the branch-and-price algorithm to find improvements compared to the solution of the greedy heuristic fast.

The remainder of the chapter is organised as follows. In Section 2.2, we formally introduce the Ship-to-Shore Problem and the time-space network which we use to solve this problem. Section 2.3 gives an overview of the related literature and how the problem differs from a PDVRP. Our mathematical model and proof of NP-hardness are given in Section 2.4. Our branch-and-price algorithm is provided in Section 2.5 and our greedy heuristic is described in Section 2.6. In Section 2.7, we describe the experimental set-up and analyse the performance of our exact algorithm and heuristic. We end with a conclusion in Section 2.8.

2.2 Problem Definition

In this section, we formally define the problem and its notation. An overview of the notation used in this section can also be found in Appendix 2.A.

In the Ship-to-Shore Problem we have a set of sea bases Σ and a set of landing areas Λ that each have a set of (un)loading locations, $\mathcal{P}_i$ for $i \in \Sigma$ and $\mathcal{D}_i$ for

$i \in \Lambda$, respectively. The set of loading locations and unloading locations can then be defined as $\mathcal{P} = \cup_{i \in \Sigma} \mathcal{P}_i$ and $\mathcal{D} = \cup_{i \in \Lambda} \mathcal{D}_i$, respectively. We denote the set of connectors, *i.e.*, smaller ships and helicopters used for transporting the supplies, as $\mathcal{C}$. Let $\mathcal{M}$ be the set of resource types for which the dimensions are known, and n_m the demand for resource type $m \in \mathcal{M}$. Here, we aggregate resources with the same origin, destination, priority level, and resource set, if assigned any. It is not possible to aggregate resources that are not identical in the origin, destination, priority, and/or resource set, as, for example, personnel should stick together with their unit and it is therefore specified what their origin is. Personnel with different priority levels, origins, and/or destinations, are thus considered different resource types. We define the meaning of priority level and resource set later in this section.

For each connector $c \in \mathcal{C}$, $\mathcal{L}^c$ denotes the set of feasible ways connectors can be loaded, called *loadings*. Here $l \in \mathcal{L}^c$ is a vector of length $|\mathcal{M}|$ representing the number of resources of each type $m \in \mathcal{M}$ that can be transported together, *i.e.*, they can feasibly fit together on connector c, and have the same priority level, origin and destination. This implies that exactly one loading needs to be selected for each trip made by a connector from an SB to an LA. Travel times between the locations are denoted by t_{ijc} for $i, j \in \mathcal{P} \cup \mathcal{D}$ and $c \in \mathcal{C}$. When $i \in \mathcal{P}$ and $j \in \mathcal{D}$, t_{ijc} is determined using the speed of connector $c \in \mathcal{C}$ while it is loaded, and when $i \in \mathcal{D}$ and $j \in \mathcal{P}$, using the speed while it is empty. The (un)loading time for a connector $c \in \mathcal{C}$ at location $i \in \mathcal{P} \cup \mathcal{D}$ is denoted by t'_{ic}. For each connector $c \in \mathcal{C}$ we denote the fuel capacity, fuel consumption rate, and refuelling rate by Q_c, h_c, and g_c, respectively.

Due to the (un)loading capacities, at most one connector can be located at each $i \in \mathcal{P} \cup \mathcal{D}$ at all times. Furthermore, the fuel level of a connector should be non-negative at all times. All resources have a priority level and our convention shall be that the lower the number, the higher the priority. Hence, all priority $\pi \in \{1, \ldots, \Pi - 1\}$ resource types should be delivered at their destination, before the unloading of priority $\pi + 1$ resource types starts, where Π is the number of priority levels. Furthermore, we have a collection of resource sets $s \in \mathcal{S}$, where $\mathcal{S}$ is a partition of (a subset of) $\mathcal{M}$, *i.e.* not all resources are necessarily part of a resource set. The delivery of resource types within a resource set should start within ϵ time of each other. Namely, if unloading of a connector with resources from resource set s starts at time t and takes t' time, then unloading the next connector containing resources from resource set s should start at time $t + t' + \epsilon$ at the latest. We call these requirements the resource set constraints. If a resource type is not assigned to any resource set, there is no requirement to link the delivery of these resources to the delivery of other resource types, but only to those of that same type.

A solution to this problem defines for each connector $c \in \mathcal{C}$, a sequence of trips from a loading location $i \in \mathcal{P}$ to an unloading location $j \in \mathcal{D}$ where for each trip a loading is assigned, such that the makespan is minimised and all constraints are met.

2.2.1 The Time-Space Network

To solve the problem, we make use of a time-space network. In the time-space network we work with discrete time periods. This time discretisation results in loss of exactness unless the time period length is set to the greatest common divisor of all (un)loading times and travel times (Boland et al., 2019). However, when the time period length decreases, the number of required time periods increases and so does the size of the network. Hence, there is a trade-off between the size of the time-space network and solution quality. In practice, the distances between the SBs and LAs are quite large. Thus, the travel times are relatively long compared to the (un)loading time. Therefore, the length of a time period is chosen such that (un)loading can occur in one time period (Amrouss et al., 2017). In the remainder of this section, we explain how the time-space network is constructed.

Due to (un)loading capacities, there is a maximum number of connectors that can be (un)loaded at the same location simultaneously. Furthermore, as we are considering a heterogeneous fleet of connectors that can reach different loading spots, the type of connectors (un)loading at a location has to be taken into account. Moreover, it is possible that the same connector visits the same location multiple times. Thus, we have to model the movement of connectors over time and allow them to visit the same location multiple times. To model this problem, we therefore opt for the usage of a time-space network inspired by the approach of Chardaire et al. (2005). To construct a time-space network for an instance, an upper bound on the number of required time periods is required. To this extend, we will use the upper bound obtained using the greedy heuristic described in Section 2.6 denoted by T.

Let $G = (\mathcal{N}, \mathcal{A})$ be the time-space network with $\mathcal{N}$ the set of nodes and $\mathcal{A}$ the set of arcs. The nodes in the time-space network are a combination of a location and a time period and the arcs denote a transition through time and space.

There are four types of nodes: starting, ending, (un)loading, and waiting nodes. The starting and ending nodes denote the starting and ending locations of the connectors which are denoted by τ_c and τ'_c for connector $c \in \mathcal{C}$, respectively. Hence, the set of starting and ending nodes is defined as $\tau = \{(\tau_c, 1) \mid c \in \mathcal{C}\}$ and $\tau' = \{(\tau'_c, T) \mid c \in \mathcal{C}\}$, respectively. (Un)loading nodes correspond to (un)loading locations at the SBs and LAs and are defined as $\mathcal{P}' = \{(i,t) \mid i \in \mathcal{P}, t = 1,\dots,T\}$ and $\mathcal{D}' = \{(i,t) \mid i \in \mathcal{D}, t = 1,\dots,T\}$ for loading and unloading activities, respectively. The interpretation of node (i,t) is that the connector is (un)loaded at location i during time period t. Waiting nodes are added to model that a connector waits for a free (un)loading spot or travels below maximum speed. The set of waiting locations is denoted by $\mathcal{W} = \{(i,c) \mid i \in \Sigma \cup \Lambda, c \in \mathcal{C}\}$ and the corresponding nodes by $\mathcal{W}' = \{(i,t) \mid i \in \mathcal{W}, t = 1,\dots,T\}$. The total set of nodes can then be defined as $\mathcal{N} = \mathcal{P}' \cup \mathcal{D}' \cup \mathcal{W}' \cup \tau \cup \tau'$.

The set of arcs $\mathcal{A}$ consists of four types of arcs: starting, ending, travelling, and waiting arcs. The starting and ending arcs denote the departure and return of a connector from its starting or ending location and are defined as $\{(i,j) \mid i \in \tau, j \in \mathcal{P}'\}$ and $\{(i,j) \mid i \in \mathcal{D}', j \in \tau'\}$, respectively. Furthermore, we include the option of not using a connector by including an arc from node $(\tau_c, 1)$ to (τ_c', T). Travelling arcs correspond to a feasible transition in both time and space between a loading location i and unloading location j. A feasible transition in space implies that i and j can both be accessed by one of the available connectors. A transition is feasible in time when the time difference of node i and j is at least equal to the travel time t_{ijc} for some connector $c \in \mathcal{C}$. It is possible that a connector does not directly proceed to the next location due to the priority, resource set, and capacity constraints, but either waits for a spot to be free or travels below maximum speed. For a travel time of t_{ijc} between loading location i and unloading location j, or vice versa, by connector $c \in \mathcal{C}$, let Δ be the equivalent number of time periods. Then, arcs would have to be added from time period t to $t+\Delta+1, t+\Delta+2, \dots, T$. To avoid adding all possible transitions between the different locations, we use the previously defined waiting nodes. With the use of these waiting nodes, we only have to add the shortest arc between locations. Namely, from each loading node an arc is added to an unloading node or waiting node at an LA when feasible in time and space, and from each unloading node an arc is added to a loading node or waiting node at an SB when feasible in time and space. Finally, waiting arcs are added between waiting nodes to allow for a connector to stay at a waiting location, namely the arcs $\{(j,t),(j,t+1) \mid j \in \mathcal{W}, t = 1,\dots,T-1\}$. Hence the number of arcs in the network is linear in the number of SBs, LAs, connectors, and time periods.

Example 2.1. *An example of a time-space network for one connector with one SB containing one dock and one LA containing one beach is given in Figure 2.1. In this example, the connector needs two time periods to travel from the SB to the LA and one time period to travel from the LA to the SB. The dashed and thick lines represent*

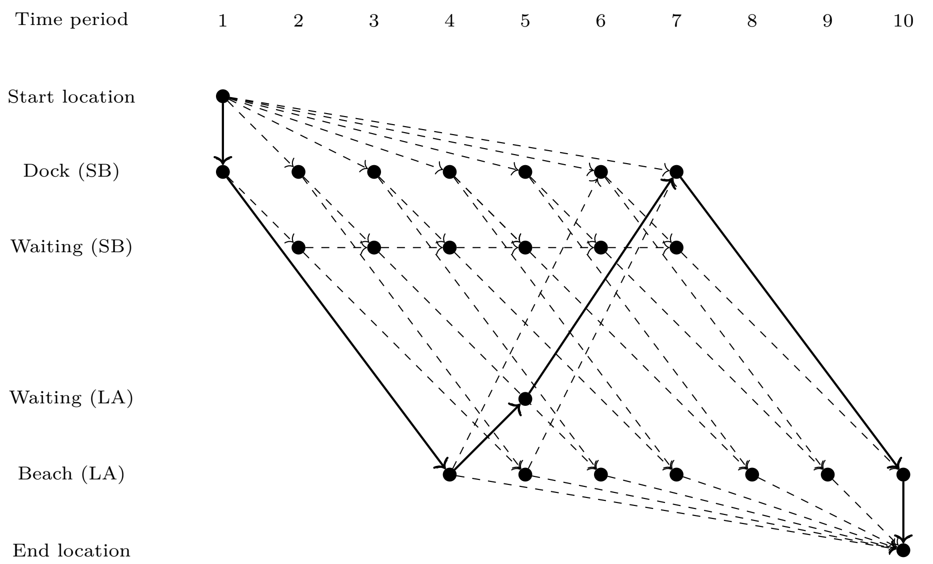


Figure 2.1: Example of a time-space network for one connector with one SB containing one dock and one LA containing one beach. Here all possible arcs are given (the bold and dashed lines combined) for a connector that takes two time periods to travel from the SB to the LA and one time period to travel from the LA to the SB. The bold lines are an example of a path through the network.

all arcs and the thick lines correspond to one feasible path through the network. We see that the connector is loaded with resources at the SB in time period 1, travels to the beach during time periods 2 and 3 to arrive for unloading in time period 4. After visiting the beach, the connector does not immediately proceed to the SB, but waits for one time period before proceeding. After loading at the SB in time period 7, the connector ends its last trip after unloading in time period 10.

Finally, we have to define the deliveries of resources from resource sets in terms of the time-space network. For a resource set, it is imposed that in a delivery wave, deliveries of its resources take place within ϵ time of each other. In the time-space network, we therefore impose that resources from a resource set should be delivered in consecutive time periods.

The above described time-space network can be used to model the movement of connectors through both time and space. A solution for the Ship-to-Shore Problem consists of, for each connector, a path through the time-space network and for each trip from an SB to an LA, the loading that should be transported. Which loadings can be used on a specific arc while satisfying the priority and resource set constraints, will depend on the loadings that were assigned in the trips preceding that arc. Therefore, simply duplicating all arcs from an SB to an LA node for each loading would result in some paths through the time-space network being infeasible with respect to these constraints. We describe how loadings are considered in Section 2.5.2.

2.3 Literature Review

We first focus on the literature related to the Ship-to-Shore Problem. Thereafter, we link it to some other practical problems with similar characteristics and we describe how the Ship-to-Shore Problem can be interpreted as a special case of the pick-up and delivery vehicle routing problem.

2.3.1 The Ship-to-Shore Problem

The Ship-to-Shore Problem has been addressed in some research. Villena (2019) solves the Ship-to-Shore Problem using an Integer Linear Programming model to minimise the makespan. The author considers the priority levels of the connectors, however, the author does not consider (un)loading capacities and the fact that some resources are complementary and should belong to the same wave. Fuel capacities are imposed by defining a maximum distance that the connectors can travel and thus, refuelling is not allowed.

Christafore Jr. (2017), Danielson (2018) and Strickland (2018) generate schedules for the Ship-to-Shore Problem using a three-phase approach. In the first phase, a quickest flow problem is considered. In this problem, the aim is to maximise the total demand satisfied given a fixed number of time periods. If the actual demand is higher than the satisfied demand, the number of time periods is increased. When the number of time periods is large enough for all demand to be satisfied, the corresponding number of connector trips per connector type can be determined. In this phase, capacity constraints are disregarded and hence no direct schedule can be constructed from these trips. In the second phase, trips are assigned to connectors in an assignment problem using a heuristic. Finally, in the last phase, a schedule is constructed from the output of the assignment problem such that the makespan is minimised and (un)loading capacities are satisfied. Christafore Jr. (2017) and Strickland (2018) only consider the transport of fuel, while Danielson (2018) extends this framework to multiple commodities.

In the three-phase approach from Christafore Jr. (2017), Danielson (2018) and Strickland (2018), (un)loading capacities are disregarded when the set of trips to be executed are determined in the first phase. Therefore, this set of trips could be suboptimal compared to the set of trips that result from an integrated approach. Furthermore, resource set constraints are hard to impose as the set of trips selected in the first phase could result in an infeasible solution.

2.3.2 Related Problems

There exist problems that have a similar structure to the Ship-to-Shore Problem. First, two specific examples are given. Second, its relation to the pick-up and delivery vehicle routing problem is explained.

An example of a problem with similar characteristics to the Ship-to-Shore Problem is an evacuation where people are located at one or multiple locations and have to be brought to one or multiple shelter locations. Some locations might have to be evacuated first because they are more in danger and hence have a higher priority. The question is then what trips the vehicles, *e.g.* buses, have to make to evacuate the area as quickly as possible. Kulshrestha et al. (2014) aim to minimise the evacuation time by assigning buses to pick-up location, while incorporating uncertainty in the demand. They do not consider priorities, resource sets, fuel capacities, and (un)loading capacity at the locations and assume that buses only travel to a fixed pick-up point. Zhao et al. (2020) use a heuristic to determine the allocation of buses to trips with the aim of minimising both the in-bus travel time and the waiting time of the evacuees. They incorporate time-windows in which each location should be visited, which can be seen as a strict type of priority constraints. They do not consider resource sets, and fuel and (un)loading capacities.

Another example of a problem with similar characteristics is the installation of a wind farm. Components of the wind mills and the underlying wind farm infrastructure have to be transported from the shore to the sea using vessels. Priorities arise since certain components are required at the start, while others are only required later on. Resource set constraints can be interpreted as the delivery of components that have to be used together in the next step in the installation process. Vessels are rented and hence to minimise the costs, the installation of the wind farm has to be completed as fast as possible. Ursavas (2017) uses a Benders decomposition approach to determine the time at which a particular vessel should start a certain building process and what loading is selected in each tour. Weather predictions are included as the weather has a big influence on the time that certain steps in the building process require.

Another related problem is the pick-up from and delivery to offshore platforms.

Here, vessels have to be scheduled to visit offshore oil and gas platforms to execute deliveries and pick-ups, *e.g.* the delivery of equipment and collection of waste. The vessels have capacity constraints, but also capacity constraints for (un)loading at the offshore platforms can exists. This problem is studied by Gribkovskaia et al. (2008) and Cuesta et al. (2017) and solved using a tabu search and adaptive large neighbourhood search, respectively.

More generally, the Ship-to-Shore Problem can be linked to a pick-up and delivery vehicle routing problem (PDVRP), where items should be collected at a pick-up location and then delivered to its delivery location. However, the location that is visited determines the resource that is transported, while in the Ship-to-Shore Problem it has to be determined which resources are picked-up each time a connector visits the location. Therefore, unless the general PDVRP is extended to a split delivery problem, each location is visited exactly once as opposed to the Ship-to-Shore Problem in which trips are made back and forth between a limited set of locations (Nowak, 2005). In general, this problem does not contain priorities, resource sets, refuelling constraints, and (un)loading capacities at the locations. For each of these aspects, we briefly explain the difference between related aspects of the PDVRP and the Ship-to-Shore Problem.

Time windows, which are a common extension of the basic PDVRP (Ticha et al., 2017), can be seen as a type of priority ordering between the resources. However, the difference is that there is no strict ordering, but only a partial ordering when using time windows. Furthermore, the time windows impose the constraint that the pick-up and delivery occur during a certain time window, which is not necessarily the case in the Ship-to-Shore Problem.

Resource set constraints could be interpreted as requiring all items with the same pick-up and delivery pair to be delivered at the same time, while allowing for split deliveries. Synchronised visits in a vehicle routing problem (VRP) have been considered to allow for driver transitioning or executing tasks to be performed by more vehicles, but differ from the Ship-to-Shore Problem where visits should be synchronised in a delivery wave that can contain multiple consecutive time periods (Bredstrom and Rönnqvist, 2007; Drexl, 2012; Liu et al., 2019).

Fuel is sometimes considered in transportation problems with the aim of minimising these costs (Xiao et al., 2012) or the emissions (Behnke et al., 2021). Refuelling options have been studied in relation to the VRP since the introduction of alternative fuel-powered vehicles that have a more limited driving range and more limited and costly refuelling options compared to traditional vehicles. We can model the fuel level in the Ship-to-Shore Problem in a similar way as is done in the electrical VRP. Desaulniers et al. (2002), Hiermann et al. (2016), Schneider et al. (2014) assume vehicles remain at a charging location until the batteries are full. However, in the Ship-to-Shore Problem, refuelling can occur simultaneously with loading a connector, and can be terminated before the connector is fully refuelled, which requires an additional constraint.

The (un)loading capacities are essential in the Ship-to-Shore Problem, as disregarding them can lead to infeasible schedules where the capacities are exceeded. A limited number of (un)loading spots is realistic for routing problems, as loading capacity might be limited at a depot. However, these are usually disregarded in the basic vehicle routing problem. In practice, depots have a limited number of loading bays at which vehicles can be loaded, resulting in the vehicle routing problem with docking constraints (Rieck and Zimmermann, 2010).

2.4 Mathematical Model and Complexity

In this section, we discuss the computational complexity of the Ship-to-Shore Problem defined in Section 2.2. In Section 2.4.1, the computational complexity of the Ship-to-Shore Problem and two special cases are determined to be NP-hard. Thereafter, we provide a mathematical model in Section 2.4.2 to solve the problem using the time-space network approach defined in Section 2.2.1.

2.4.1 Computational Complexity

To prove that the Ship-to-Shore Problem defined in Section 2.2 is NP-hard, we consider the decision version of the problem. In this problem, the question is whether all resources can be transported in at most N periods while adhering to all previously mentioned constraints.

First, we consider the special case in Theorem 2.1. In this case we have one sea base, one landing area and one connector.

Theorem 2.1. *The decision version of the Ship-to-Shore Problem with a single sea base, landing area, connector and priority level; with zero (un)loading times, fuel consumption rates and resource sets; and with unit travel times and multiple resource types, is strongly NP-complete.*

Proof. This special case is in NP as it can be checked in polynomial time whether demand is satisfied and the makespan is at most N.

We will use a reduction from Set Cover, which is known to be strongly NP-complete (Karp, 1972), to prove that this special case is NP-complete. In Set Cover, we are given a set of elements $E = \{e_1, \ldots, e_n\}$ and some subsets of those elements $S_1, \ldots, S_m$, where each $S_j \subseteq E$. We say that an element is covered if at least one subset containing the element is chosen. The question is whether we can cover all elements by choosing at most K subsets.

Given an instance of Set Cover, create a resource type for each element, and set its demand equal to 1. Furthermore, create a loading for every subset, where the loading contains the resource types corresponding to the elements from the subset. Finally, set $N = 2K - 1$, since the makespan is the time required to deliver the resources and does not include the required for the connectors to return to the sea base in their last trip.

If the instance of Set Cover is a yes-instance, we can use the loadings corresponding to the chosen subsets. Therefore, the created instance of the ship-to-shore instance is also a yes-instance. Similarly, if the ship-to-shore instance is a yes-instance, it naturally follows that the instance of Set Cover is also a yes-instance. ■

Second, consider the special case in Theorem 2.2. In this case we have one sea base. The resource types that have to be transported only differ in the destination to which they have to be transported, *e.g.*, only fuel has to be transported from the sea base to different locations. Connectors can have various capacities of transporting this resource, *e.g.*, one connector can transport 1000 gallons of fuel at a time, while another connector can transport 2000 gallons. There is one priority level, *i.e.*, the delivery of the resource at one location is not prioritised over other locations, and there are no resource sets, *i.e.*, it is not required to deliver the demand at one location at the same time or directly after one another. This problem can thus be interpreted as the delivery of fuel to petrol stations from a central depot. This special case differs from the special case in Theorem 2.1 as it contains multiple destinations and connectors, but resources that only differ in their destination.

Theorem 2.2. *The decision version of the Ship-to-Shore Problem with one sea base, resource types that only differ in their destination, multiple connectors, (un)loading times equal to zero, unit travel times between the sea base and the landing areas, one priority level, no resource sets, and a fuel consumption rate equal to zero, is strongly NP-complete.*

Proof. This special case is in NP as it can be checked in polynomial time whether demand is satisfied and the makespan is at most N.

We will use a reduction from 3-Partition, which is known to be strongly NP-complete (Garey and Johnson, 1979), to prove that this special case is NP-complete. In 3-Partition, we are given $3m$ numbers, $a_1, \ldots, a_{3m}$. Each number a_i satisfies $B/4 < a_i < B/2$, where $B = \frac{1}{m}\sum_{i=1}^{3m} a_i$. The question is whether we can partition the numbers in subsets of size 3 with equal sum. More formally, do there exist sets $S_1, \ldots, S_m \subseteq \{1, \ldots, 3m\}$ with $|S_j| = 3$ for all j, $S_j \cap S_{j'} = \emptyset$ for all j, j', and $\sum_{i \in S_j} a_i = B$ for all j?

Given an instance of 3-Partition, create m landing areas, each with a demand of B. Hence, there are m resource types with demand B that only differ in the destination. Then, create $3m$ connectors, where connector i can ship a_i units of a resource per trip. Finally, set $N = 1$.

Suppose that the instance of 3-Partition is a yes-instance. If we send the connectors corresponding to the numbers in S_j to landing area j, all demand is satisfied within one time unit. Therefore, the ship-to-shore instance is also a yes-instance. Conversely, suppose that the ship-to-shore instance is a yes-instance. This means that all demand is satisfied within one time unit. Therefore, the connectors visiting landing area j have a total capacity of B. Furthermore, because $B/4 < a_i < B/2$, we know that each landing area is visited by exactly 3 connectors. Hence, the instance of 3-Partition is also a yes-instance. ■

A similar reduction from Partition (Karp, 1972) shows that the problem is (weakly) NP-hard for two landing areas.

Corollary 2.1 follows from Theorem 2.1 and Theorem 2.2, as the Ship-to-Shore Problem is a generalisation of the special cases in these theorems. Thus, the Ship-to-Shore Problem is NP-hard.

Corollary 2.1. *The Ship-to-Shore Problem is strongly NP-hard.*

2.4.2 Integer Linear Programming Formulation

In this section we introduce a linear programming formulation to solve the Ship-to-Shore Problem. To solve the problem, we need to find a route for each connector, which consists of a path through the time-space network, such that a loading to be transported is assigned for each trip from an SB to an LA. The Ship-to-Shore Problem is then equivalent to assigning exactly one route to each connector such that the makespan is minimised, all resources are transported, and all constraints regarding (un)loading capacities, resource set constraints, and priority levels are met. First, some notation is introduced in Table 2.1.

Table 2.1: List of sets, variables and parameters for the Integer Linear Programming formulation.

Set	Explanation
$\mathcal{C}$	Set of connectors
$\mathcal{P}'$	Set of loading nodes
$\mathcal{D}'$	Set of unloading nodes
$\mathcal{M}$	Set of resource types
$\mathcal{N}$	Set of nodes
$\mathcal{R}$	Set of routes
$\mathcal{S}$	Set of resource sets
Variables	**Explanation**
x_{rc}	A binary variable equal to 1 if route $r \in \mathcal{R}$ is assigned to connector $c \in \mathcal{C}$, 0 otherwise
$y_{\pi t}$	A binary variable equal to 1 if priority $\pi \in \{1, \dots, \Pi\}$ resources are delivered in time period $t \in \{1, \dots, T\}$
T_π^{start}	Time period at which the first resource with priority $\pi \in \{1, \dots, \Pi\}$ is unloaded
T_π^{end}	Time period at which the last resource with priority $\pi \in \{1, \dots, \Pi\}$ is unloaded
v_{st}	A binary variable equal to 1 if resources from set $s \in \mathcal{S}$ are delivered in time period $t \in \{1, \dots, T\}$
w_{st}^{start}	A binary variable equal to 1 if a wave of unloading resources from set $s \in \mathcal{S}$ starts in time period $t \in \{1, \dots, T\}$
w_{st}^{end}	A binary variable equal to 1 if a wave of unloading resources from set $s \in \mathcal{S}$ ends in time period $t \in \{1, \dots, T\}$
t_{span}	The makespan, *i.e.* the duration of the operation
Parameters	**Explanation**
d_r	The last delivery period of route $r \in \mathcal{R}$
n_m	The number of resources of type $m \in \mathcal{M}$ that have to be transported
n_{rm}	The number of resources of type $m \in \mathcal{M}$ that are transported in route $r \in \mathcal{R}$
T	The number of time periods
Π	The number of priority levels

Furthermore, we introduce the following notation. Let $\mathcal{R}(c)$ for $c \in \mathcal{C}$ be the set of routes for connector c and $\mathcal{R}(c, m)$ for $c \in \mathcal{C}$ and $m \in \mathcal{M}$ be the set with routes for connector c that delivers at least one resource of type m. For $c \in \mathcal{C}$ and $i \in \mathcal{N}$, we define $\mathcal{R}(c, i)$ as the set of routes for connector c that visit node i. Finally, $\mathcal{R}(c, \{\pi, t\})$ for $c \in \mathcal{C}$, $\pi \in \{1, \dots, \Pi\}$ and $t \in \{1, \dots, T\}$ denotes the set of routes for connector c that deliver priority π resources in time period t and $\mathcal{R}(c, \{s, t\})$ for $c \in \mathcal{C}$, $s \in \mathcal{S}$ and $t \in \{1, \dots, T\}$ is the set of routes for connector c that deliver resources from resource set s in time period t.

Now, an Integer Linear Programming (ILP) formulation can be defined as follows:

$$\min\ t_{span} \tag{2.1}$$

$$\text{s.t.} \sum_{c \in \mathcal{C}} \sum_{r \in \mathcal{R}(c,m)} n_{rm} x_{rc} \geq n_m \qquad m \in \mathcal{M} \tag{2.2}$$

$$\sum_{r \in \mathcal{R}(c)} x_{rc} = 1 \qquad c \in \mathcal{C} \quad (2.3)$$

$$\sum_{c \in \mathcal{C}} \sum_{r \in \mathcal{R}(c,i)} x_{rc} \leq 1 \qquad i \in \mathcal{N} \quad (2.4)$$

$$\sum_{r \in \mathcal{R}(c,\{\pi,t\})} x_{rc} \leq y_{\pi t} \qquad c \in \mathcal{C}, t \in \{1,\dots,T\}, \pi \in \{1,\dots,\Pi\} \quad (2.5)$$

$$\sum_{\pi=1}^{\Pi} y_{\pi t} \leq 1 \qquad t \in \{1,\dots,T\} \quad (2.6)$$

$$T_{\pi}^{start} \leq t y_{\pi t} + T(1 - y_{\pi t}) \qquad t \in \{1,\dots,T\}, \pi \in \{1,\dots,\Pi\} \quad (2.7)$$

$$T_{\pi}^{end} \geq t y_{\pi t} \qquad t \in \{1,\dots,T\}, \pi \in \{1,\dots,\Pi\} \quad (2.8)$$

$$T_{\pi+1}^{start} \geq T_{\pi}^{end} \qquad \pi \in \{1,\dots,\Pi - 1\} \quad (2.9)$$

$$\sum_{c \in \mathcal{C}} \sum_{r \in \mathcal{R}(c,\{s,t\})} x_{rc} \leq |\mathcal{D}'| v_{st} \qquad s \in \mathcal{S}, t \in \{1,\dots,T\} \quad (2.10)$$

$$v_{st} \leq \sum_{c \in \mathcal{C}} \sum_{r \in \mathcal{R}(c,\{s,t\})} x_{rc} \qquad s \in \mathcal{S}, t \in \{1,\dots,T\} \quad (2.11)$$

$$v_{st} - v_{s,t-1} = w_{st}^{start} - w_{s,t-1}^{end} \qquad s \in \mathcal{S}, t \in \{2,\dots,T\} \quad (2.12)$$

$$w_{st}^{start} + w_{s,t-1}^{end} \leq 1 \qquad s \in \mathcal{S}, t \in \{2,\dots,T\} \quad (2.13)$$

$$\sum_{t=1}^{T} w_{st}^{start} \leq 1 \qquad s \in \mathcal{S} \quad (2.14)$$

$$t_{span} \geq \sum_{r \in \mathcal{R}(c)} d_r x_{rc} \qquad c \in \mathcal{C} \quad (2.15)$$

$$x_{rc} \in \mathbb{B} \qquad c \in \mathcal{C}, r \in \mathcal{R}(c) \quad (2.16)$$

$$y_{\pi t} \in \mathbb{B} \qquad t \in \{1,\dots,T\}, \pi \in \{1,\dots,\Pi\} \quad (2.17)$$

$$T_{\pi}^{start}, T_{\pi}^{end} \in \mathbb{N}^{+} \qquad \pi \in \{1,\dots,\Pi\} \quad (2.18)$$

$$v_{st} \in \mathbb{B} \qquad s \in \mathcal{S}, t \in \{1,\dots,T\} \quad (2.19)$$

$$w_{st}^{start}, w_{st}^{end} \in \mathbb{B} \qquad s \in \mathcal{S}, t \in \{1,\dots,T\} \quad (2.20)$$

$$t_{span} \in \mathbb{N}^{+}. \quad (2.21)$$

The objective (2.1) is to minimise the makespan, *i.e.* the duration of the operation. This is set by constraints (2.15). Constraints (2.2) ensure that all resources are transported from a sea base to a landing area. Constraints (2.3) assign a route to each connector. Note that a route can also be a route directly from the start location of a connector to the end location of that connector, *i.e.* not all connectors have to transport resources. To ensure the (un)loading capacities are respected, constraints (2.4) are imposed. Constraints (2.5) force the decision variable $y_{\pi t}$ equal to 1 if a route is selected that transports priority $\pi \in \{1, \ldots, \Pi\}$ resources in time period $t \in \{1, \ldots, T\}$. Constraints (2.6) ensure that it is not possible to unload resource types from different priorities in the same time period.

Constraints (2.7) and (2.8) set the start and end time of unloading resources from a certain priority while constraints (2.9) ensure the ordering in the priorities. The term $T(1 - y_{\pi t})$in constraint (2.7) ensures that no bound is imposed on T_{π}^{start} when $y_{\pi t}$ equals 0, *i.e.* when no priority π resource types are delivered in time period t. We impose a strict priority ordering, where priority levels cannot be mixed even if they correspond to different locations. However, a more lenient interpretation where priority levels are destination specific is possible in our model. Namely, we can define variables T_{π}^{start} and T_{π}^{end} for each destination separately and add constraints (2.9) for each separate location.

All resources in a resource set $s \in \mathcal{S}$ have to be delivered in the same time period or in consecutive time periods, *i.e.* in the same wave. Constraints (2.10), (2.11), (2.12) and (2.13) ensure that w_{st}^{start} and w_{st}^{end} denote whether a delivery wave of resource set s starts or ends in time period $t \in \{1, \ldots, T\}$, respectively. Constraints (2.10) and (2.11) jointly set v_{st} equal to 1 if there is at least one route selected that delivers at least one resource from resource set s in time period t. Here $|\mathcal{D}'|$ is used as there can be at most $|\mathcal{D}'|$ connectors delivering resources in a single time period. Constraints (2.12) and (2.13) jointly set w_{st}^{start} and w_{st}^{end} equal to 1 if period t is part of the delivery wave for resource set s and if $t - 1$ is not or $t + 1$ is not, respectively. An example of how these constraints work is given in Example 2.2. Constraint (2.14) ensures that there is only one wave for each resource set.

Example 2.2. *To illustrate, Figures 2.2 and 2.3 present the nodes of a beach (LA) in a time-space network. The arcs in these figures correspond to the connectors trips to this beach that deliver resources from resource set $s \in \mathcal{S}$. In Figure 2.2, v_{s2}, v_{s4}, w_{s2}^{start}, w_{s2}^{end}, and w_{s4}^{start} are forced to one and all other variables v_{st} and w_{st}^{end} are forced to zero. This solution violates constraint* (2.14) *and hence is infeasible. In Figure 2.3, v_{s2}, v_{s3}, w_{s2}^{start} and w_{s3}^{end} are forced to be equal to 1 and all other v_{st} and w_{st}^{end} are forced to be equal to 0. This satisfies constraint* (2.14) *and thus forms a feasible wave for resource set s.*

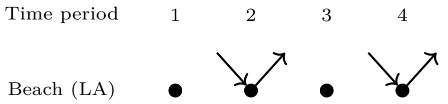


Figure 2.2: Wave example 1 (infeasible)

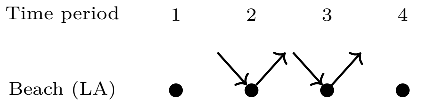


Figure 2.3: Wave example 2 (feasible)

2.5 Branch-and-Price Algorithm

In the ILP formulation defined in Section 2.4.2, we consider routes which are paths through the time-space network where each trip from an SB to an LA has an assigned loading. To avoid enumerating an exponential number of routes, we resort to column generation within a branch-and-price framework (Barnhart et al., 1998). We solve a restricted master problem (RMP) containing a subset of all feasible routes and use dual information of the LP-relaxation of this RMP to find improving routes via a Pricing Problem (PP), until no improving routes exist. We solve the PP using a fast heuristic labelling algorithm which is backed up by an exact MIP formulation to ensure we find an improving column if one exist. As the optimal solution of the LP-relaxation of the RMP may be fractional, branch-and-price is used to ensure we find an exact integer solution.

The RMP and PP used in the column generation are explained in Sections 2.5.1 and 2.5.2. Thereafter, the branching strategies will be explained in Section 2.5.3. We conclude with some ideas to strengthen the formulation in Section 2.5.4.

2.5.1 Restricted Master Problem

The aim of the RMP is to assign a route to each connector such that the makespan is minimised and all constraints are met. However, contrary to the mathematical model in Section 2.4.2, we only consider a subset of the routes, namely $\mathcal{R}' \subseteq \mathcal{R}$. The set of routes is extended using the PP within the branch-and-price framework.

To ensure that a feasible solution can be found for any subset of routes, variables $p_m \in \mathbb{R}^+$ for $m \in \mathcal{M}$ and $p_c \in \mathbb{R}^+$ for $c \in \mathcal{C}$ are introduced and added to the left hand side of constraints (2.2) and (2.3), respectively. Assigning positive values to these variables is penalised in the objective such that all resources are transported by the routes in an optimal solution, if possible. To ensure that the penalty is high enough to avoid using them if possible, their costs are set to the upper bound on the number of time periods in the time-space network.

2.5.2 Pricing Problem

After solving the LP-relaxation of the RMP, the dual variables can be used to find new routes that will improve the solution, if any remain. Thus, the objective of the PP is to find routes with negative reduced costs. These routes can then be added to the set of available routes in the RMP, such that it can be solved again with a larger set of routes. From the RMP defined in Section 2.5.1, the reduced costs for a route r by connector $c \in \mathcal{C}$ can be found as:

$$
\begin{aligned}
RC(x_{rc}) = &- \sum_{m \in \mathcal{M}(r)} n_{rm}\lambda_m^{(2.2)} + \lambda_c^{(2.3)} + \sum_{i \in \mathcal{N}(r) \setminus \{\tau, \tau'\}} \lambda_i^{(2.4)} \\
&+ \sum_{\pi=1}^{\Pi} \sum_{t \in \mathcal{T}(\pi, r)} \lambda_{ct\pi}^{(2.5)} + \sum_{s \in \mathcal{S}} \sum_{t \in \mathcal{T}(s,r)} \left(\lambda_{s,t}^{(2.10)} - \lambda_{s,t}^{(2.11)}\right) + d_r\lambda_c^{(2.15)} \qquad (2.22)
\end{aligned}
$$

where $\lambda_m^{(2.2)}$, $\lambda_c^{(2.3)}$, $\lambda_i^{(2.4)}$, $\lambda_{ct\pi}^{(2.5)}$, $\lambda_{st}^{(2.10)}$ and $\lambda_c^{(2.15)}$ are the dual values of the LP-relaxation of the RMP and $\mathcal{T}(i)$ corresponds to the set of time periods in which resource types with characteristic i are delivered. Here it holds that $\lambda_m^{(2.2)}$, $\lambda_i^{(2.4)}$, $\lambda_{ct\pi}^{(2.5)}$, $\lambda_{st}^{(2.10)}$, $\lambda_c^{(2.15)} \in \mathbb{R}^+$, while $\lambda_c^{(2.3)} \in \mathbb{R}$ as these correspond to equality constraints.

The PP can be solved using a mixed-integer programming model using the formulation provided in Appendix 2.B. Alternatively, it can be solved as a shortest path problem with resource constraints (SPPRC) on the time-space network using an appropriate labelling algorithm (Irnich and Desaulniers, 2005). Since solving the PP in an exact manner as an SPPRC is very time-consuming when all resources/constraints are considered, we opt for a hybrid approach. First, we use a heuristic pricing method, where we limit the number of labels we store in each node. Second, once this heuristic procedure fails at finding routes with reduced costs below some threshold, we switch to exact pricing using the mixed-integer programming model until the node is solved to optimality.

In the labelling algorithm, we define a label as $L(C, f, \pi, S)$, where C denotes the reduced costs, f denotes the current fuel level, π denotes the current priority level, and $S \subseteq \mathcal{S}$ denotes the set of resource sets of priority level π of which at least one resource has been delivered. Labels are created by extending paths through the time-space network and appropriately updating the reduced costs based on the dual variables, the fuel level, the current priority level and the current set of resource sets covered. Whenever a path is extended from a node at an SB to a node at an LA, a label is constructed for each possible loading. Namely, if the current priority level is

π, it is only possible to assign loadings with priority level $\pi' \geq \pi$. Furthermore, due to the delivery waves, if resources of a resource set in $S \subseteq \mathcal{S}$ are already delivered in a route, it is not possible to assign a loading that contains resources of a resource set in S as this route cannot be used. Label $L_1(C_1, f_1, \pi_1, S_1)$ then strictly dominates label $L_2(C_2, f_2, \pi_2, S_2)$ if (i) $C_1 \leq C_2$, (ii) $f_1 \geq f_2$, (iii) $\pi_1 = \pi_2$, (iv) $S_1 \subseteq S_2$, and (v) at least one of these conditions is strict.

We use this labelling algorithm heuristically by limiting the number of labels at each node (Desaulniers et al., 2002). Here we use a different parameter for the sink and the bound on the number of labels at the other nodes in the network. Furthermore, to limit the number of routes that are added to the RMP in each iteration, a similarity score is used to select routes that are disjoint in the nodes that it visits and/or resources that it delivers (Breugem, 2020). Namely, we sort the candidate routes in ascending order of reduced costs and select the first route. Then, for each following candidate route x, we compute its similarity score with all routes y that are already selected. The route x is selected if none of the similarity scores exceeds 0.5, *i.e.* each route is at least 50% dissimilar to the other routes. We define the similarity score of routes x and y as an overlap coefficient known as the Szymkiewicz-Simpson coefficient which is based on both the nodes visited and resources delivered (Simpson, 1947; Szymkiewicz, 1934):

$$\frac{1}{2}\frac{|\mathcal{N}(x) \cap \mathcal{N}(y)|}{\min\{|\mathcal{N}(x)|, |\mathcal{N}(y)|\}} + \frac{1}{2}\frac{\sum_{m\in\mathcal{M}} \min\{n_{xm}, n_{ym}\}}{\min\left\{\sum_{m\in\mathcal{M}} n_{xm}, \sum_{m\in\mathcal{M}} n_{ym}\right\}}, \tag{2.23}$$

for $\mathcal{N}(x)$ the set of nodes in route x, and n_{xm} the number of resources of type m delivered in route x. We perform this step for each connector separately and do not compare routes of different connectors as the demand of a resource can be fulfilled by different connectors.

2.5.3 Branching Strategies

To complete the branch-and-price algorithm, branching strategies have to be defined that will exclude the fractional solution we obtain at a branching node, but do not exclude any feasible integer solutions in that branch.

Let a_{ij} be a binary variable representing the usage of an arc from i to j and z_{li} a binary variable representing the delivery of loading l to location i. We then apply branching on the arcs a_{ij} and the deliveries z_{li} used. We need to branch on both variables as solely branching on one of them does not guaranteed an integer solution: branching on the arcs only can mean that a connector is assigned to a set of routes that use the same arcs, but executes different deliveries in different routes; branching on the deliveries only can mean that deliveries are split by the different connectors. We choose to first branch on the arcs until a solution with integer a_{ij}'s is found, whereafter we branch on the deliveries.

When branching on an arc (i, j), one of the children nodes is not allowed to use this arc whence all routes using (i, j) are removed from the corresponding RMP and (i, j) is removed from the PPs. In the other child node, the usage of arc (i, j) is imposed by which, all routes that use arcs (i, j') for $j' \neq j$ or (i', j) for $i' \neq i$ are removed, and these arcs are not used in the corresponding PPs.

Branching on a delivery implies branching on a combination of a loading $l \in \mathcal{L}$ and a delivery node $i \in \mathcal{D}'$. When a combination (l, i) is forbidden, all routes that deliver loading l to node i are removed from the RMP and the delivery of l at i is forbidden in the PPs. When a combination (l, i) is obligatory, all routes that deliver loading l' for $l' \neq l$ to node i are removed from the RMP and the delivery of l' to i is forbidden in the PPs.

For imposing an arc or a delivery, the arc/loading removal operations above are not sufficient yet as they affect only variables a_{ij}/z_{li}, and do not enforce the fact that in the RMP, only x_{rc} decisions have to be allowed that respect the a_{ij}/z_{li} impositions. For this reason, in the corresponding child nodes, we require constraints that would not be needed, *e.g.* in the classical vehicle routing problem where each node should be visited. These constraints ensure that among the available routes, at least one is selected that respects the given arc/delivery imposition:

$$\sum_{c \in \mathcal{C}} \sum_{r \in \mathcal{R}'(c,\{i,j\})} x_{rc} \geq 1 \qquad (i, j) \in \mathcal{A}^o \tag{2.24}$$

$$\sum_{c \in \mathcal{C}} \sum_{r \in \mathcal{R}'(c,\{l,i\})} x_{rc} \geq 1, \qquad (l, i) \in \mathcal{L}^o \tag{2.25}$$

where $\mathcal{A}^o$ is the set of obligatory arcs and $\mathcal{L}^o$ the set of obligatory loading and delivery pairs.

Adding constraints (2.24) and (2.25) to the restricted master problem changes the objective function of the pricing problem. Hence, the following terms are added to the objective in (2.27):

$$-\sum_{(i,j) \in \mathcal{A}^o} \lambda_{ij}^{(2.24)} a_{ij} - \sum_{(l,i) \in \mathcal{L}^o} \lambda_{li}^{(2.25)} z_{li}, \tag{2.26}$$

where $\lambda_{ij}^{(2.24)}$ and $\lambda_{li}^{(2.25)}$ are the dual variables of constraints (2.24) and (2.25), respectively.

There are multiple ways to select which arc or delivery to branch on. Here, the least, *i.e.* closest to 0 or 1, and most, *i.e.* closest to 0.5, fractional arc or delivery strategy will be used. Ties are broken arbitrarily.

As with arc or loading selection, different methods exist to determine which node in the branching tree to branch on next. Here, after preliminary experimentation, the depth first approach is used. Again, ties are broken arbitrarily.

2.5.4 Strengthening the Formulation

The objective for the Ship-to-Shore Problem is to minimise the makespan. The makespan is determined by constraints (2.15) in the RMP. However, in the branch-and-price algorithm, decision variables can be fractional and hence constraints (2.15) bound the makespan by the weighted average of the last delivery period of the assigned routes. Minimising this weighted average gives incentives to combine long routes in which many resources are transported with very short routes to reduce the bound on the makespan. This negatively affects the quality of the solution at a node in the branching tree. Therefore, with the aim of strengthening the formulation, bounds on both the length of routes and on the makespan will be used.

First, we can explicitly bound the lengths of routes. Short routes cannot be prohibited because it is possible for a connector to have a short route in an optimal solution. However, routes longer than the current upper bound (makespan of the initial greedy solution or the best integer solution so far) can be prohibited as we know these will not be chosen in an optimal integer solution.

Second, we can find a lower bound on the optimal makespan by relaxing the constraints on the (un)loading capacities and resource sets. This relaxation can be solved as a job-shop scheduling problem in which resource-transporting trips to the shore are assigned to the connectors and can easily be solved in a pre-processing phase. Because this lower bound can be higher than the optimal value at the root node of the branching tree, it can improve the estimates of the optimality gaps of the integer solutions from the branch-and-price algorithm. Secondly, we can add this bound explicitly in the RMP as a constraint on the makespan, which will accelerate the solution time per node.

2.6 Greedy Heuristic

The greedy heuristic aims to mimic the current scheduling procedure and will also serve as an upper bound of the makespan for the purpose of constructing the time-space network. Routes for the connectors are extended by iteratively adding new trips, *i.e.* a visit to a loading location to pick up a specific set of resources, followed by a visit to an unloading location to deliver these resources. A forbidden list is used to denote trips that are not allowed to be added as they have previously resulted in infeasible solutions.

In the algorithm, we need to use some measure to determine which trip should be added. Since we are interested in minimising the makespan, this measure should include the change in the makespan when the trip is executed. When solely the change in the makespan is used, faster connectors are preferred over slower, but larger, connectors. We therefore want to compensate a larger change in the makespan with the quantity of resources that are transported. As there are both people and large vehicles that can be transported, using the number of resources that are transported is not an appropriate measure. This would namely imply that large resources, which can only be transported by a certain connectors, are left until the smaller resources are transported. Therefore, we have chosen to use the surface area measured in square meters, a in Step 2, that is transported as a measure of the quantity of resources that are transported.

In the greedy heuristic, the following steps are executed:

STEP 0: Let $\pi = 1$ be the current priority level and let $T = 0$ be the last delivery period.

STEP 1: For every connector determine the first possible delivery period t for a loading with priority π resources that is not forbidden. Here both the capacity constraints at the different locations and the priority constraints are taken into account. Furthermore, we keep track of the fuel level of the connector, hence if necessary, the connector remains one or more additional time periods at the SB for refuelling purposes.

STEP 2: For each loading with priority level π, determine the total area in m^2 of resources that have not been delivered yet, let this area be a.

STEP 3: Select the connector-loading pair with minimum $\frac{t-T}{a}$, *i.e.* the minimum ratio of the change in the makespan divided by the area of resources that are transported. Here, ties are broken arbitrarily. Add this trip to the current schedule and set $T = t$.

STEP 4: While there is an incomplete resource set, *i.e.* a trip was added in which part of the resources from a resource set are delivered, the completion of the delivery of this resource set is prioritised before considering the delivery of resources that are not part of this resource set. This is done by repeating versions of Steps 1-3 until all resources in the resource sets are completed. In Step 1, besides the capacity and priority constraints, the resource sets now also have to be considered, *i.e.* only consecutive time periods to the current delivery of the incomplete resource set are considered. In Step 2, only the area of the resources that belong to the incomplete resource set are considered. If no feasible pair exists in Step 3, *i.e.* the resource set constraint cannot be met, the last trip that was added to the schedule is removed and marked as forbidden.

STEP 5: If all priority π resources are delivered, $\pi = \pi + 1$.

STEP 6: If all resources are delivered, return T, else go to Step 1.

2.7 Computational Experiments

In this section we present the results of computational experiments on both artificial instances and instances constructed using data from the Royal Netherlands Navy. In these experiments, we first analyse the effect of using the bounds as explained in Section 2.5.4 and the heuristic pricing method as explained in Section 2.5.2. Thereafter, we analyse the performance of the branch-and-price algorithm and the greedy heuristic for different types of instances.

In Section 2.7.1 we describe the data and instance construction, and in Section 2.7.2 the corresponding results are presented.

2.7.1 Experimental Design

We consider both artificial instances and instances constructed using data from the Royal Netherlands Navy. Each instance consists of a demand set containing the resources that should be delivered and the corresponding SBs and LAs, and a supply set in terms of the available connectors and (un)loading capacities at the SBs and LAs. Furthermore, an instance is characterised by the operational constraints that are considered. We define the following naming scheme for instances: d-s-o, where d denotes the demand set, s the supply set, and o the set of operational constraints that are considered. We first discuss each of these three aspects of an instance and then give an overview of the constructed instances for the instances constructed with data from the Royal Netherlands Navy. Thereafter, we describe the artificial instances.

Table 2.2: Overview of the demand sets containing the identifier, number of SBs, number of LAs, the distance between the SBs and LAs in nautical miles, the number of priority levels, the number of resource types, and the number of resource sets for each demand set.

Demand Set	#SBs	#LAs	Distance SB - LA (nm)	# Priority Levels	# Resource Types	# Resource Sets
1A	2	2	15	1	10	1
1B	2	2	15	2	15	2
1C	2	2	15	2	18	2
2A	2	3	15	1	10	1
2B	2	3	15	2	15	2
2C	2	3	15	2	20	2
3A	2	1	15	2	10	2
3B	2	1	15	2	18	3
3C	2	1	15	2	25	3

Royal Netherlands Navy Instances

In this section we describe the instances constructed with data from the Royal Netherlands Navy. The demand at the shore consists of a set of resource types that have to be transported. For each resource type, the origin and destination is defined. Furthermore, it is denoted what the priority number of the resource type is and to what resource set it belongs to, if any.

To test the performance of the greedy heuristic for different sized instances, the demand sets have different sizes, where it holds that iA $\subset$ iB $\subset$ iC, for i $\in \{1, 2, 3\}$. We thus define nine demand sets d, of which the descriptive statistics are shown in Table 2.2. This includes the number of SBs, LAs, priority levels, resource sets, and the distance between the SBs and LAs. The table shows that demand sets 1A and 2A contain one priority level, *i.e.* all resources have the same priority level.

The supply consists of the number of available connectors and the (un)loading capacity at both the SBs and LAs. There are three types of loadings spots: docks, davits, and landing platforms, and there is one type of unloading spot: landing zones. The different types of (un)loading spots can be used by different connector types. Namely, landing platforms can only be used by helicopters, while docks and davits can only be used by surface connectors. Furthermore, not all surface connectors can access a davit. This can only be used by the smaller surface connectors. An overview of the supply sets s is given in Table 2.3.

The operational constraints consist of, amongst others, constraints regarding the order of the delivery of the resources. As we are interested in the performance of our branch-and-price algorithm and our greedy heuristic under different circumstances, we vary these operational constraints. These constraints are the priority constraints and resource set constraints. For instances without a priority ordering, this implies

Table 2.3: Overview of the supply sets containing the identifier, the number of connectors, the number of different connector types, and the number of each type of (un)loading spot available at each SB/LA for each supply set.

Supply Set	# Connectors	# Connector Types	# Docks per SB	# Davits per SB	# Landing Platforms per SB	# Landing Zones per LA
1	4	2	1	2	1	2
2	6	2	1	2	1	2
3	12	2	2	4	2	2
4	16	4	2	4	2	2
5	8	3	2	2	2	2

that all resources have the same priority level. When either or both of these coordinating constraints are absent from an instance, there are fewer constraints. This affects the number of choices in each step of the greedy heuristic and the number of variables and constraints in the branch-and-price algorithm.

We define four options for o. Let N denote the case in which neither the priority nor the resource set constraints are considered. Let P denote the case in which the priority constraints are added to case N, and let W denote the case in which the resource set constraints are added to case N. The full and default model we consider contains both constraints, denoted by F.

Using the demand sets, supply sets, and the set of operational constraints, we can construct the instances. Large demand sets with low supply in terms of the number of available connectors can result in feasibility issues due to the resource set constraints. Namely, it should be possible to deliver all resources of the same resource set at the same time or shortly after each other. Since travel times between the SBs and LAs are large, it is not possible to assign a connector to multiple trips transporting resources of the same resource set, as this would lead to a violation of the resource set constraint. Therefore, for each resource set in a combination of a demand set and a supply set, we verify whether it is possible to assign loadings to the connectors such that all resources of this set are transported and each connector is used at most once. If this is not possible, the instance is not feasible when resource set constraints are imposed and hence we only consider the instances with options P and N. On the other hand, small demand sets in combination with a large supply set, are not realistic. Hence, not all combinations of demand and supply sets are considered.

Taking both the feasibility as well as the realism of combinations of demand sets and supply sets into account results in the instances shown in Table 2.4. Here ‘All’ implies that all four subsets of the operational constraints are considered. The instances used for only options P and N imply that the other combinations of demand set and supply set are infeasible in terms of the resource set constraints, *e.g.* for demand set 3C with supply set 5. Demand sets 1A and 2A have a single priority level, hence there is no difference between incorporating the priority constraints or not. Therefore, for these demand sets only options W and N are considered. This results in 96 instances.

Artificial Instances

In this section, we describe the artificially constructed instances. The instances and solutions can be obtained from Wagenvoort (2023). The artificial instances are constructed from a set of connectors and a set of available resources. From these sets, five combinations of a demand and supply set are generated. We consider all four configurations of the operational constraints, resulting in 20 instances.

We consider four different types of connectors and seven different types of resources. For the connector types, we consider one type of helicopter and three types of surface connectors: large, medium, and small. For each connector type, given in Table 2.5, we specify the (un)loading time, capacity, fuel related parameters, speed when loaded and empty, and at which locations it can be (un)loaded. For each of the resources, given in Table 2.6, we specify the size of the resource and for each connector type whether it can be placed on this connector type. Note that for simplicity we have defined the capacity of a connector and the size of a resource in terms of one dimension only. We consider a loading feasible for a connector type when it does not contain any resources that are not compatible with the connector type and the sum of the sizes of the resources in the loading does not exceed the capacity of the connector.

To construct the artificial instances, we randomly select a set of connectors, where we always include at least one connector of type 'Large' to ensure feasibility. We then randomly select a set of resources, for which we randomly generate a priority

Table 2.4: Overview of the instances where for each demand set and supply set combination that is used, the corresponding set of operational constraints is denoted. In total, this results in 96 instances.

Supply	Demand Set								
Set	1A	1B	1C	2A	2B	2C	3A	3B	3C
1	W, N	All		W, N	P, N		All	P, N	
2	W, N	All		W, N	All		All	All	
3			All			All			All
4		P, N	All		All	All		P, N	All
5	W, N	P, N	All	W, N	All	All	All	All	P, N

level, resource set, and quantity. Here we select the smaller resources with a higher probability. Furthermore, we generate for each resource an origin and destination. We let there be either one or two SBs and one or two LAs reachable by surface connectors and at most one LA reachable by helicopters. The distance between the SBs and LAs is set to 15 nm for each instance.

We construct five demand sets Di and supply sets Si for $i \in \{1, \dots, 5\}$. The contents of these five demand and supply sets can be found in Appendix 2.C. These can be used to obtain all 20 instances, namely we consider instances $Di - Si - o$ for $i \in \{1, \dots, 5\}$ and $o \in \{\text{F, P, W, N}\}$. We then consider all maximal loadings for each connector based on the compatibility of the resources and the size of the connector. The loadings can be found in Wagenvoort (2023).

2.7.2 Results

In this section, the computational results are presented. The algorithms are implemented in Java using CPLEX 12.10. The experiments are executed on the Dutch national SurfSARA Lisa cluster consisting mostly of nodes with 16 core Intel Xeon 6130 processors and 96GB RAM. We choose a cut-off point of one hour for the branch-and-price algorithm. Whenever we refer to the gap, we mean the optimality gap between the lower bound and the best integer solution.

For the labelling algorithm, parameters must be defined. Namely, we have to determine the maximum number of labels to store at each node, the maximum number of labels to store at the sink, and the threshold for switching from the heuristic labelling algorithm to the exact MIP. Based on early experiments, we choose the following parameters. We allow, in each pricing problem, at most ten labels at the sink and at most five at all other nodes, *i.e.* at most ten columns per connector are added in each iteration. When no route with negative reduced costs smaller than

Table 2.5: Overview of connector types used for the artificial instances. For each connector type we indicate the (un)loading times in minutes, the capacity for resources, the fuel capacity, the fuel consumption rate (per minute), the refuel rate (per minute), the speed while being (un)loaded in nautical miles per hour, and the type of (un)loading locations it can access.

Type	Loading Time (min.)	Unloading Time (min.)	Capacity	Fuel Capacity	Fuel Consumption (per min.)	Refuel Rate (per min.)	Speed Loaded (knts)	Speed Empty (knts)	Location Compatibility
Large	15	15	150	10	0.005	0.15	10	12	dock, beach
Medium	10	10	75	10	0.005	0.15	20	25	dock, beach
Small	5	5	25	10	0.01	0.15	30	30	dock, davit, beach
Helicopter	5	5	8	10	0.01	0.15	125	125	landing platform (SB), landing zone (LA)

Table 2.6: Overview of resource types used for the artificial instances, their size, and what connector can carry them.

		Connector Compatibility			
Name	Size	Large	Small	Medium	Helicopter
Pax	1	X	X	X	X
VehA	60	X			
VehB	40	X			
VehC	35	X			
VehD	30	X	X		
VehE	25	X	X		
VehF	20	X	X		

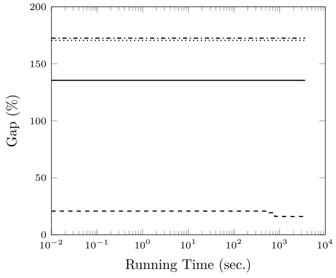
-0.05 is found by the labelling algorithm, we switch to the MIP and use the MIP until the node is solved to optimality.

The remainder of the section is structured as follows. First, the effect of using the bounds and the labelling algorithm is analysed. Second, the performance of the branch-and-price algorithm and greedy heuristic are discussed. Finally, the performance of the artificial instances are analysed to compare with the findings of the instances from the Royal Netherlands Navy.

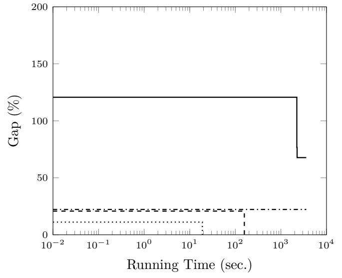
The Effect of Bounds and Labelling

Intuitively, using the bounds to strengthen the formulation as explained in Section 2.5.4 and using heuristic pricing in combination with the exact MIP as explained in Section 2.5.2, speeds-up the branch-and-price algorithm. To validate our intuition, some instances with different characteristics and demand are run for different configurations. We run each instance for both the most and least fractional branching rule with a cut-off point of one hour and use the result of the branching strategy with the lowest running time or lowest gap after one hour. The change in the optimality gap over time for this branching strategy is presented in Figure 2.4.

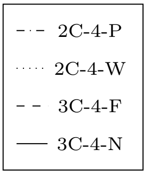
Figure 2.4a shows that none of the instances terminate within an hour. In fact, only for one instance, instance 3C-4-F, the gap decreases within an hour. For three of the four instances, the lower bound obtained by solving the machine scheduling problem as explained in Section 2.5.4, is better than the initial root node without the bound. Therefore, in Figure 2.4c, we see a decrease in the gap for these instances. When both bounds are added, see Figure 2.4b, we see that instances 3C-4-F and 2C-4-W now terminate within an hour. Note that the gap instantly drops to zero from the initial gap. Since we are using a depth first approach, it can occur that the lower bound within the branching tree could not have been updated before termination of the algorithm. This results in an instant drop in the gap. We also see that the gap

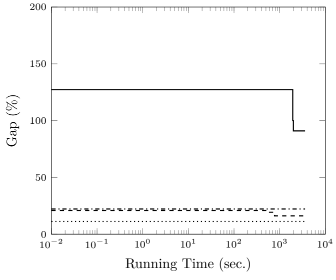


(a) Without bounds and without labelling.

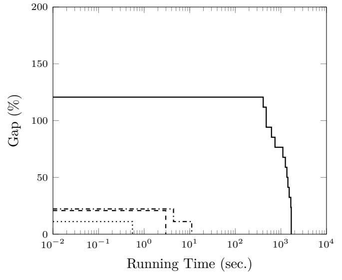


(b) With both bounds and without labelling.





(c) With lower bound and without labelling.



(d) With both bounds and with labelling.

Figure 2.4: The change in the optimality gap, the gap between the lower bound and best integer solution, over time in the branch-and-price algorithm with a cut-off point of one hour for different configurations. For the configurations we consider with/without actively using the lower and upper bound as described in Section 2.5.4, and with/without heuristic labelling as described in Section 2.5.2.

obtained for instance 3C-4-N when both bounds are used is lower compared to the gap when only the lower bound is used. Figure 2.4d shows the change in the gap over time when the heuristic labelling algorithm is used in combination with the exact MIP. We see that now also instance 3C-4-N terminates and that all other instances terminate faster.

Results of the Branch-and-Price Algorithm and Greedy Heuristic

In this section, the results of the branch-and-price algorithm and the greedy heuristic are presented. Here both bounds and the labelling heuristic are used in the branch-and-price algorithm. We analyse the performance of the branch-and-price algorithm and the greedy heuristic based on the instance characteristics, namely, the instance type (A, B, or C), and which operational constraints are present in the instance. Summarising results for the different instance types are given in Table 2.7, full results

Table 2.7: Performance measures of the branch-and-price algorithm with a cut-off point of one hour. This includes the percentage of instances that terminate within an hour, the average number of columns generated, the average number of times the labelling heuristic is used per node, the average number of times the exact MIP is used per node, and percentiles of the total running times. Here o denotes the set of operational constraints regarding the order of the resources that are present, namely either the full set (F), only the priority constraints (P), only the resource set constraints (W), or neither of these constraints (N).

							Running Time (sec.)				
o	Type	Num. Instances	Term. <1 h (%)	Avg. # Columns	Avg. # Heur	Avg. # MIP	min	25^{th}	50^{th}	75^{th}	max
F	A	3	100	140.00	5.00	1.00	0.14	0.19	0.24	0.29	0.34
	B	7	100	6422.29	5.50	1.00	0.36	0.43	0.65	2.21	2.84
	C	8	87.50	2045.88	7.26	1.12	0.65	0.99	1.23	2.28	3600
	Total	18	94.44	3623.71	6.27	1.06	0.14	0.49	1.10	2.45	3600
P	A	3	100	27058.50	1.42	1.00	14.81	588.84	1162.87	1736.90	2310.93
	B	12	58.33	15306.13	3.06	1.00	0.42	136.76	3600	3600	3600
	C	9	22.22	22156.00	1.31	1.02	10.64	3600	3600	3600	3600
	Total	24	50.00	19656.33	2.10	1.01	0.42	714.30	3600	3600	3600
W	A	9	100	165.60	5.20	1.80	0.13	0.13	0.18	0.70	1.02
	B	7	100	448.33	5.44	1.00	0.42	0.46	0.63	0.74	1.06
	C	8	100	960.63	5.92	1.00	0.27	0.55	1.04	2.91	8.17
	Total	24	100	589.63	5.58	1.21	0.13	0.42	0.70	1.10	8.17
N	A	9	100	7556.00	2.80	1.00	0.05	0.10	4.31	21.08	58.81
	B	12	100	3060.00	1.78	1.01	0.27	1.38	6.80	93.55	182.08
	C	9	77.78	182202.30	3.87	1.00	0.31	1.31	88.25	2643.81	3600
	Total	30	93.33	66659.00	2.72	1.01	0.05	0.57	7.65	120.33	3600
Total		96	84.38	23781.08	4.12	1.07	0.05	0.55	1.64	182.08	3600

can be found in the supplementary materials.

In Table 2.7 we see that all small instances (type A) terminate within an hour and that, with the exception of the instances with only priority constraints (P), all terminate within a minute. All type B instances terminate within approximately three minutes, with the exception of the instances with only priority constraints (P). For type C instances, all the instances with only the resource set constraints (W) terminate within an hour. We thus see, both from the percentage of instances that terminate within an hour and the running times, that the instances with only the priority constraints are hardest to solve. On average, 84% of all instances could be solved within an hour of which 75% are solved in approximately three minutes.

Regarding the usage of the MIP after heuristic labelling, we observe that the average number of times the MIP is used is only slightly above one, indicating that the heuristic labelling algorithm is able to find good negative reduced costs columns. We see that the number of generated columns varies a lot. If the number of columns is high, this can be caused by two things, the number of iterations at a node is high resulting in many routes, and/or the number of nodes in the branching tree is high resulting in many nodes that are solved. For the 'P' and 'N' instances, we see that,

Table 2.8: The effect of priority level constraints and resource set constraints on the performance of the greedy heuristic with a cut-off point of one hour for the branch-and-price algorithm on the instances from the Royal Netherlands Navy. Here *o* denotes the set of operational constraints regarding the order of the resources that are considered, namely either the full set (F), only the priority constraints (P), only the resource set constraints (W), or neither of these constraints (N). For each set of operational constraints, we denote for each instance type separately as well as for all instances types combined, the number of such instances, the % of these instances for which we know the greedy heuristic found the optimal solution, and the % of instances for which the BP found improvement. For the instances for which improvement was found, the average gap with the solution from the greedy heuristic is given as well as the average time it took to find the best integer solution.

					Instances with Improvement	
o	Instance Type	Num. Instances	Greedy Optimal (%)	BP finds Improvement (%)	Avg. Gap Greedy (%)	Avg. T. till Best Sol. (sec.)
F	A	3	100	0	-	-
	B	7	85.71	14.29	14.29	2.69
	C	8	87.50	12.50	10.00	5.85
	Total	18	88.89	11.11	12.14	4.27
P	A	3	66.67	33.33	127.27	14.81
	B	12	50.00	16.67	38.96	91.64
	C	9	11.11	55.56	24.89	31.93
	Total	24	37.50	33.33	41.21	44.72
W	A	9	100	0	-	-
	B	7	100	0	-	-
	C	8	100	0	-	-
	Total	24	100	0	-	-
N	A	9	88.89	11.11	63.64	8.51
	B	12	33.33	66.67	44.35	50.05
	C	9	33.33	55.56	24.18	484.24
	Total	30	50.00	46.67	38.52	202.15
Total		96	66.67	25.00	37.22	133.19

although the labelling heuristic is only used between two and three times on average, the number of columns is significantly larger compared to the 'F' and 'W' instances. These results thus show that, as we would expect, these less constrained instances have more feasible routes making the problem harder to solve.

Comparing the results of the branch-and-price algorithm with the solutions of the greedy heuristic, we find that for two thirds of the instances the greedy solution is proven to be optimal. Table 2.8 shows the percentage of instances for which the greedy solution is proven to be optimal, the percentage of instances for which the branch-and-price algorithm finds improvement, and the average gap with the greedy solution and average time to obtain these improvements for these instances. We see from these results that the greedy heuristic performs best when there are resource sets (F and W). In the cases with resource set constraints and no priority levels (W), all greedy solutions are optimal, and proven to be optimal within an hour. When priority levels do exist (F), the greedy heuristic finds the best and optimal solution in almost 90% of the cases. When an improvement is found, the solution of the greedy heuristic has, on average, a makespan that is 12% higher compared to the optimum. These improvements are found in a few seconds. Although instances with both constraints (F) are more constrained compared to instances with only resource set constraints (W), we observe that improvements are found in F instances while for all W instances the solution of the greedy heuristic is optimal. As the greedy heuristic is an iterative method, it does not anticipate on future trips. Hence, it might select connectors at the end of a priority level that are required at the start of the next priority level. The branch-and-price algorithm is then able to find a solution where these essential connectors are not used at the end of a priority level resulting in a better solution.

For instances with no resource set constraints (P and N), we see that the greedy heuristic performs worse. Furthermore, not all large instances (type B and C instances) without resource set constraints terminate within an hour. This is the case for all instances with resource set constraints. Hence, we see that while there are instances where no improvement compared to the solution of the greedy heuristic is found, the solution of the greedy heuristic is also not proven to be optimal. In 37.50% and 50.00% of the instances with (P) and without (N) priority constraints, respectively, the greedy solution is known to be optimal. In 33.33% and 46.67% of the instances, respectively, we find an improvement. On average, the gaps between the solution of the greedy heuristic and the solution of the branch-and-price algorithm are 40%. If we compare the gaps for the different instances types (types A, B and C), we see that the smaller the instances, the larger the gaps. This can be explained by the smaller makespan, *i.e.* a difference in the makespan of one time period gives a larger gap for smaller instances where the makespans are smaller. When improvements can be found, these are found within five minutes in most of the cases. Only for the largest instances (type C) when there are no operational constraints regarding the order of delivery (N), there are instances where it takes more than five minutes.

Overall, we thus see that the branch-and-price algorithm is able to solve the majority of the instances in limited time. We see that instances with only priority constraints are hardest to solve, while instances with both constraints or only resource set constraints are easiest to solve. Although priority constraints restrict the solution space compared to having neither of these constraints, they do not restrict it as much as resource set constraints while still having to consider some coordination between the schedules of the different connectors. Furthermore we find that when resource set constraints exist, the greedy solution is often optimal. When these constraints

Table 2.9: The effect of priority level constraints and resource set constraints on the performance of the greedy heuristic with a cut-off point of one hour for the branch-and-price algorithm on the artificial instances. Here o denotes the set of operational constraints regarding the order of the resources that are considered, namely either the full set (F), only the priority constraints (P), only the resource set constraints (W), or neither of these constraints (N). For each set of operational constraints, we denote the % of these instances for which we know the greedy heuristic found the optimal solution, and the % of instances for which the BP found improvement. For the instances for which improvement was found, the average gap with the solution from the greedy heuristic is given as well as the average time it took to find the best integer solution.

o	Greedy Optimal (%)	BP finds Improvement (%)	Instances with Improvement: Avg. Gap Greedy (%)	Instances with Improvement: Avg. T. till Best Sol. (sec.)
F	20	40	25.48	1032.01
P	40	20	47.83	104.43
W	20	80	42.05	32.09
N	20	80	30.10	15.00
Average	25	55	35.22	214.25

are not present, improvements were found compared to the solution of the greedy heuristic in about 40% of the instances. These improvements are on average found within a few minutes and have an average gap of approximately 40% with the solution of the greedy heuristic.

Artificial Instances

The results in Section 2.7.2 show that the greedy heuristic performs well when instances are constrained, especially when resource set constraints exist. In this section, we compare the performance of the greedy heuristic and branch-and-price algorithm for the artificial instances and compare this finding with the results from the instances from the Royal Netherlands Navy.

The 20 artificial instances are run under the same configurations as the instances from the Royal Netherlands Navy. The full results of the artificial instances can be found in the supplementary materials. Summarising results comparing the performance of the greedy heuristic and branch-and-price algorithm can be found in Table 2.9. The results in this table show that for some instances the greedy heuristic finds the optimal solution, but that quite often improvement is found. When improvement is found, the average gap between the greedy heuristic and the best solution found by the branch-and-price algorithm is quite large, on average 35%. Furthermore, generally this solution is found relatively soon, on average within a couple of minutes.

When we compare the results of the artificial instances in Table 2.9 with the results of the instances from the Royal Netherlands Navy in Table 2.8, we notice the difference in the results for the different types of instances. Namely, while the greedy heuristic is optimal for all instances with resource set constraints from the Royal Netherlands Navy, this is not the case for the artificial instances. A potential reason is that instances from the Royal Netherlands Navy are constructed by experts that have knowledge of the capacities and the potential loadings on the connectors that can be encoded in defining the resource sets.

2.8 Conclusion

In this chapter, we provide a formulation for the Ship-to-Shore Problem that allows for coordination between the connectors. We prove that the Ship-to-Shore Problem is NP-hard, even in restricted special cases. We develop (i) a branch-and-price algorithm, and (ii) a tailored greedy heuristic. Our branch-and-price algorithm makes use of an upper and lower bound and we incorporate a pricing heuristic that improves the running time of the algorithm. We investigate, using data from the Royal Netherlands Navy, under which circumstances, which method is preferred. We find that the branch-and-price algorithm is able to solve the majority of the instances within an hour and that it performs best in very constrained cases in terms of the coordination of delivering sets of resources. We also find that the greedy heuristic is often able to find the optimal solution in such restricted cases. However, in less restricted cases the branch-and-price algorithm finds an improvement compared to the greedy algorithm in approximately 40% of the instances. For those instances, the average gap with the greedy heuristic is around 40% and those improvements are found within a few minutes.

We then use artificial instances to compare with the instances from the Royal Netherlands Navy. We find no difference in the performance of the greedy heuristic under different circumstances for these instances. This shows that potentially the greedy heuristic performs well when coordination between the resources is required due to a bias in the instances constructed by experts in the field. The artificial instances do confirm the ability of the branch-and-price algorithm to find improvements fast. Namely, for the instances where improvement compared to the solution from the greedy heuristic is found, the average gap with the solutions of the branch-and-price algorithm is 35% and found within a few minutes.

Therefore, in practice when these coordinating constraints exist, current practices mimicked by the greedy heuristic, might perform well. However, the branch-and-price algorithm is able to find large improvements to the solutions of the greedy heuristic fast, hence it is worthwhile to try and find a better schedule as large gains can be made.

There are a number of options for further research related to our model. As distances and therefore travel times are large compared to the (un)loading times in these instances, we chose to set the time period length such that (un)loading for all connectors can take place within one time period. Depending on the application, the loss in exactness can be more significant and a shorter time period length is preferred. In that case, this would require small changes to the model, namely, arcs should be added between a node for an (un)loading location in time period t and the node corresponding to the same location in time period $t + 1$. Furthermore, when (un)loading at a location takes $n > 1$ time periods, constraints should be added that require a connector to remain at this location for n time periods if it is visited.

The results show that the greedy heuristic performs well in case the problem is very constrained, but there may be opportunity for better heuristics in different situations. For less constrained problems, one can apply additional local search steps to the solution of the greedy heuristic, or running the greedy heuristic multiple times with a randomisation parameter, can result in better outcomes. To test whether a tailored heuristic works well for other similar applications, the branch-and-price algorithm can be used as a benchmark.

By using discrete time periods, some slack occurs in the schedule. A potential benefit of this slack is that it can serve as a buffer in case a delay occurs. However, the moments at which this slack occurs are not chosen and therefore accounting for delays while constructing the schedule can result in a lower expected makespan. Hence, for future research, it would be interesting to incorporate the uncertainty about the travel and (un)loading times to construct schedules with a lower expected makespan.

Appendix

2.A Notation Problem Definition

In this section, an overview of the notation used in the problem description (Section 2.2) and the description of the time-space network (Section 2.2.1) is given.

Table 2.10: List of sets and parameters in the problem description.

Set	Explanation
$\mathcal{C}$	Set of connectors
$\mathcal{D}_i$	Set of unloading locations at landing area $i \in \Lambda$
$\mathcal{D}$	Set of unloading locations such that $\mathcal{D} = \cup_{i\in\Lambda} D_i$
$\mathcal{P}_i$	Set of loading locations at sea base $i \in \Sigma$
$\mathcal{P}$	Set of loading locations such that $\mathcal{P} = \cup_{i\in\Sigma} P_i$
$\mathcal{L}^c$	Set of feasible loadings for connector $c \in \mathcal{C}$
$\mathcal{M}$	Set of resource types
$\mathcal{S}$	Set of resource sets
Σ	Set of sea bases
Λ	Set of landing areas
Parameters	**Explanation**
g_c	Refuelling rate for connector $c \in \mathcal{C}$
h_c	Fuel consumption rate for connector $c \in \mathcal{C}$
n_m	Demand for resource type $m \in \mathcal{M}$
Q_c	Fuel capacity for connector $c \in \mathcal{C}$
t_{ijc}	Travel time between locations $i, j \in \mathcal{P} \cup \mathcal{D}$
t'_{ic}	(Un)loading time at location $i \in \mathcal{P} \cup \mathcal{D}$ for connector $c \in \mathcal{C}$
Π	The number of priority levels

Table 2.12: List of sets and parameters in the time-space network.

Set	Explanation
$\mathcal{A}$	Set of arcs
$\mathcal{D}'$	Set of unloading nodes, $\mathcal{D}' = \{(i,t) \mid i \in \mathcal{D}, t = 1,\ldots,T\}$
$\mathcal{N}$	Set of nodes, $\mathcal{N} = \mathcal{P}' \cup \mathcal{D}' \cup \mathcal{W}' \cup \tau \cup \tau'$
$\mathcal{P}'$	Set of loading nodes, $\mathcal{P}' = \{(i,t) \mid i \in \mathcal{P}, t = 1,\ldots,T\}$
$\mathcal{W}$	Set of waiting locations, $\mathcal{W} = \{(i,c) \mid i \in \Sigma \cup \Lambda, c \in \mathcal{C}\}$
$\mathcal{W}'$	Set of waiting nodes, $\mathcal{W}' = \{(j,t) \mid j \in \mathcal{W}, t = 1,\ldots,T\}$
τ	Set of starting nodes, $\tau = \{(\tau_c, 1) \mid c \in \mathcal{C}\}$
τ'	Set of ending nodes, $\tau' = \{(\tau_c, T) \mid c \in \mathcal{C}\}$
Parameters	**Explanation**
T	Number of time periods in the time-space network
τ_c	Start location of connector $c \in \mathcal{C}$
τ'_c	End location of connector $c \in \mathcal{C}$

2.B Mathematical Formulation of the Pricing Problem

In the Pricing Problem (PP), we aim to find a route such that the reduced costs (2.22) are minimised. Additionally to the notation introduced in Table 2.1, the notation in Table 2.14 is used in the PP.

Table 2.14: List of sets, variables and parameters for the Pricing Problem.

Set	Explanation
$\mathcal{A}$	Set of arcs
$\mathcal{D}'$	Set of landing area nodes
$\mathcal{L}$	Set of loadings
$\mathcal{P}'$	Set of sea base nodes
$\mathcal{T}$	Set of time periods
$\mathcal{W}'$	Set of waiting nodes
Variables	**Explanation**
a_{ij}	A binary variable equal to 1 if arc $(i, j) \in \mathcal{A}$ is used, 0 otherwise
b_m	The number of resources of type $m \in \mathcal{M}$ transported
d	The last delivery period
f_i	A binary variable equal to 1 if the connector refuels at node $i \in \mathcal{P}'$, 0 otherwise
u_i	The fuel level upon reaching node $i \in \mathcal{N}$
z_{li}	A binary variable equal to 1 if loading $l \in \mathcal{L}$ is delivered to $i \in \mathcal{D}'$, 0 otherwise
Parameters	**Explanation**
g_c	The refuelling rate per time period for connector $c \in \mathcal{C}$
h_c	The fuel consumption rate per time period for connector $c \in \mathcal{C}$
k_{li}	A binary parameter equal to 1 if loading $l \in \mathcal{L}$ is available at/heading to location $i \in \mathcal{N}$, 0 otherwise
n_{ml}	The number of resources from type $m \in \mathcal{M}$ in loading $l \in \mathcal{L}$
$p_{l\pi}$	A binary parameter equal to 1 if loading $l \in \mathcal{L}$ has priority level $\pi = \{1, \ldots, \Pi\}$, 0 otherwise
Q_c	The fuel capacity of connector $c \in \mathcal{C}$
τ_c	The starting node of connector $c \in \mathcal{C}$
τ'_c	The ending node of connector $c \in \mathcal{C}$

We can then define the PP for a specific connector $c \in \mathcal{C}$ as follows:

$$
\begin{aligned}
\min\ & - \sum_{m \in \mathcal{M}} \lambda_m^{(2.2)} b_m + \lambda^{(2.3)} \\
& + \sum_{i \in \mathcal{N}(c)} \sum_{j \in \mathcal{N}(c):(i,j) \in \mathcal{A}(c)} \lambda_i^{(2.4)} a_{ij} \\
& + \sum_{t=1}^{T} \sum_{\pi=1}^{\Pi} \sum_{l \in \mathcal{L}(c,\pi)} \sum_{i \in \mathcal{D}'(c,t)} \lambda_{ct\pi}^{(2.5)} z_{li}
\end{aligned}
$$

$$
\begin{aligned}
& + \sum_{s\in\mathcal{S}} \sum_{l\in\mathcal{L}(c,s)} \sum_{t=1}^{T} \sum_{i\in\mathcal{D}'(c,t)} \left(\lambda_{st}^{(2.10)} + \lambda_{st}^{(2.11)}\right) z_{li} \\
& + \lambda_c^{(2.15)} d && (2.27)\\
\text{s.t.} \quad & \sum_{j\in\mathcal{N}(c):(\tau,j)\in\mathcal{A}(c)} a_{\tau j} = 1 && (2.28)\\
& \sum_{i\in\mathcal{N}(c):(i,\tau')\in\mathcal{A}(c)} a_{i\tau'} = 1 && (2.29)\\
& \sum_{i\in\mathcal{N}(c):(i,j)\in\mathcal{A}(c)} a_{ij} - \sum_{i\in\mathcal{N}(c):(j,i)\in\mathcal{A}(c)} a_{ji} = 0 & j\in\mathcal{N}(c)\setminus\{\tau,\tau'\} \quad & (2.30)\\
& f_i \le \sum_{j\in\mathcal{N}(c):(i,j)\in\mathcal{A}(c)} a_{ij} & i\in\mathcal{P}'(c) \quad & (2.31)\\
& u_\tau = Q_c && (2.32)\\
& u_j \le Q_c - h_c(t_j - t_i)a_{ij} & (i,j)\in\mathcal{A}(c) \quad & (2.33)\\
& u_j \le u_i + g_c f_i - h_c(t_j - t_i)a_{ij} + Q_c(1-a_{ij}) & (i,j)\in\mathcal{A}(c) \quad & (2.34)\\
& \sum_{l\in\mathcal{L}(c)} z_{lj} = \sum_{i\in\mathcal{P}'(c)\cup\mathcal{W}'(c):(i,j)\in\mathcal{A}(c)} a_{ij} & j\in\mathcal{D}'(c) \quad & (2.35)\\
& 2z_{lj} \le k_{lj} + \sum_{i\in\mathcal{P}'(c)\cup\mathcal{W}'(c):(i,j)\in\mathcal{A}(c)} k_{li}a_{ij} & l\in\mathcal{L}(c), j\in\mathcal{D}'(c) \quad & (2.36)\\
& b_m \le n_m & m\in\mathcal{M} \quad & (2.37)\\
& b_m \le \sum_{l\in\mathcal{L}(c,m)} \sum_{i\in\mathcal{D}'(c)} n_{ml} z_{li} & m\in\mathcal{M} \quad & (2.38)\\
& \sum_{l\in\mathcal{L}(c,\pi)} \sum_{j\in\mathcal{D}'(c,t)} z_{lj} \le y_{\pi t} & t\in\mathcal{T}, \pi\in\{1,\dots,\Pi\} \quad & (2.39)\\
& \sum_{\pi=1}^{\Pi} y_{\pi t} \le 1 & t\in\mathcal{T} \quad & (2.40)\\
& T_\pi^{start} \le t y_{\pi t} + T(1-y_{\pi t}) & t\in\mathcal{T}, \pi\in\{1,\dots,\Pi\} \quad & (2.41)\\
& T_\pi^{end} \ge t y_{\pi t} & t\in\mathcal{T}, \pi\in\{1,\dots,\Pi\} \quad & (2.42)\\
& T_{\pi+1}^{start} \ge T_\pi^{end} & \pi\in\{1,\dots,\Pi\} \quad & (2.43)\\
& \sum_{i\in\mathcal{D}'(c)} \sum_{l\in\mathcal{L}(c,s)} z_{li} \le 1 & s\in\mathcal{S} \quad & (2.44)\\
& \sum_{i\in\mathcal{D}'(c,t)} \sum_{l\in\mathcal{L}(c)} t z_{li} \le d & t\in\mathcal{T} \quad & (2.45)\\
& a_{ij} \in \mathbb{B} & (i,j)\in\mathcal{A}(c) \quad & (2.46)\\
& b_m \in \mathbb{N}^+ & m\in\mathcal{M} \quad & (2.47)
\end{aligned}
$$

$$d \in \mathbb{N}^+ \tag{2.48}$$

$$T_\pi^{start}, T_\pi^{end} \in \mathbb{N}^+ \qquad \pi \in \{1, \ldots, \Pi\} \tag{2.49}$$

$$f_i \in \mathbb{B} \qquad i \in \mathcal{P}'(c) \tag{2.50}$$

$$u_i \in \mathbb{R}^+ \qquad i \in \mathcal{N}(c) \tag{2.51}$$

$$w_{st}^{start}, w_{st}^{end} \in \mathbb{B} \qquad s \in \mathcal{S}, t \in \mathcal{T} \tag{2.52}$$

$$y_{\pi t} \in \mathbb{B} \qquad t \in \mathcal{T}, \pi \in \{1, \ldots, \Pi\} \tag{2.53}$$

$$z_{li} \in \mathbb{B}. \qquad l \in \mathcal{L}(c), i \in \mathcal{D}'(c) \tag{2.54}$$

The aim is to minimise the reduced costs (2.22) as represented by the objective (2.27).

Constraints (2.28), (2.29) and (2.30) ensure that a connector departs from its starting location, terminates at its ending location and ensure flow conservation throughout the network. The connector can only be refuelled at a pick-up location if this location is visited (constraints (2.31)). It is assumed that all connectors start with a full tank (constraint (2.32)). Constraints (2.33) and (2.34) update the fuel level throughout the network. They ensure that the fuel level is set to $\min\{u_i + g_c, Q_c\}$ after refuelling at $i \in \mathcal{P}'(c)$ and that no constraint is imposed on the fuel level if the arc is not used, but decreased if an arc is used.

Constraints (2.35) and (2.36) ensure that the connector can only be assigned one loading for each trip from a SB to a LA and that a loading can only be assigned when the loading is available at the SB and the connector is heading to the LA corresponding to the destination of the loading. Constraints (2.38) set b_m to the number of resources of type $m \in \mathcal{M}$ that are transported in the selected loading. This variable is upper bounded by the number of resources of type m that have to be transported in constraints (2.37) to avoid deducting the reduced costs of the resource in the objective too much.

To satisfy the priority order, constraints (2.39)-(2.43) are imposed and to ensure that there is at most one delivery for each resource set, constraints (2.44) are imposed. If within a route, multiple deliveries with resource types from the same resource set occur, this route cannot be used as this will violate the resource set constraints. Imposing this constraint will avoid generating routes that cannot be used in a feasible RMP solution. The last delivery period of the route is determined in constraints (2.45).

2.C Artificial Instances

This section contains the contents of the artificial instances. Tables 2.16 - 2.18 are the contents of the supply sets. Table 2.19 are the contents of the demand sets.

Table 2.16: Available connectors in the supply sets for the artificial instances.

	Number of Connectors			
Supply Set	Large	Medium	Small	Helicopter
S1	2	2	0	4
S2	2	0	0	0
S3	2	2	0	4
S4	4	4	4	4
S5	2	0	1	2

Table 2.17: Sea base locations in the supply sets for the artificial instances.

Supply Set	Number	Docks per SB	Davits per SB	Landing Platforms per SB
S1	2	1	0	1
S2	2	2	0	0
S3	2	2	0	2
S4	2	2	4	2
S5	1	1	0	1

Table 2.18: Landing area locations in the supply sets for the artificial instances.

Supply Set	Number	Beaches per LA	Number	Landing Zones per LA
S1	1	2	1	2
S2	1	2	0	0
S3	2	2	1	2
S4	1	2	1	2
S5	1	2	1	2

2.D Results

This section contains the full results. For each instance and branching rule, we report the objective value of the solution of the greedy heuristic, the initial lower bound, the number of loadings, the number of nodes and arcs in the time-space network, and output from the branch-and-price algorithm. The output from the branch-and-price algorithm contain the objective value, the lower bound of the algorithm, the running time of the algorithm, the time after which the best integer solution was found, if applicable, and the gap, if applicable.

Table 2.19: Demand sets for the artificial instances. When the resource set equals ∅, this implies that these resources are not part of any resource set.

Demand Set	Name	Origin	Destination	Quantity	Priority Level	Resource Set
D1	VehE	LPD1	LA1	11	1	∅
	Pax	LPD2	LA1	100	1	∅
	Pax	LPD1	LS1	106	1	∅
	VehE	LPD1	LA1	1	1	A
	Pax	LPD1	LA1	4	1	A
	VehA	LPD1	LA1	4	2	B
	VehF	LPD1	LA1	1	2	B
	VehC	LPD2	LA1	1	2	C
	VehF	LPD2	LA1	1	2	C
	VehE	LPD2	LA1	2	2	∅
	VehE	LPD1	LA1	1	2	∅
D2	VehF	LPD1	LA1	4	1	∅
	VehF	LPD2	LA1	1	1	∅
	VehF	LPD1	LA1	2	1	A
	VehC	LPD1	LA1	1	1	A
	VehE	LPD1	LA1	1	1	A
	VehE	LPD2	LA1	1	1	B
	VehF	LPD2	LA1	1	1	B
	VehC	LPD2	LA1	1	1	B
	VehB	LPD1	LA1	1	2	C
	VehC	LPD1	LA1	1	2	C
	VehF	LPD1	LA1	2	2	C
	VehE	LPD1	LA1	1	2	C
	VehC	LPD2	LA1	2	2	D
	VehE	LPD2	LA1	2	2	D
	VehB	LPD2	LA1	1	2	D
	VehF	LPD2	LA1	2	2	∅
	VehF	LPD1	LA1	2	2	∅
D3	VehF	LPD1	LA1	5	1	∅
	Pax	LPD2	LA1	1	1	∅
	Pax	LPD1	LS1	2	1	∅
	VehF	LPD1	LA2	1	1	∅
	Pax	LPD1	LA1	1	1	A
	VehF	LPD1	LA1	1	1	A
	VehE	LPD2	LA2	1	1	∅
	VehD	LPD1	LA2	2	2	B
	VehF	LPD1	LA2	2	2	B
	VehE	LPD1	LA2	2	2	∅
	VehF	LPD2	LA1	2	2	∅
	VehF	LPD1	LA2	4	2	∅
	VehF	LPD1	LA1	4	2	∅
	VehE	LPD2	LA1	2	2	∅
	VehE	LPD1	LA1	2	2	∅
D4	VehF	LPD1	LA1	10	1	∅
	Pax	LPD2	LA1	80	1	∅
	Pax	LPD1	LS1	100	1	∅
	VehF	LPD1	LA1	1	1	A
	Pax	LPD1	LA1	4	1	A
	VehC	LPD2	LA1	4	2	B
	VehF	LPD2	LA1	1	2	B
	VehF	LPD2	LA1	1	2	∅
D5	VehE	LPD1	LA1	4	1	A
	VehF	LPD1	LA1	1	1	A
	Pax	LPD1	LA1	4	1	A
	VehA	LPD1	LA1	1	1	∅
	VehF	LPD1	LA1	2	1	∅
	Pax	LPD1	LS1	80	1	∅
	VehB	LPD1	LA1	1	2	∅
	VehE	LPD1	LA1	2	2	B
	VehF	LPD1	LA1	2	2	B
	VehE	LPD1	LA1	3	2	∅
	Pax	LPD1	LA1	4	2	∅
	VehC	LPD1	LA1	2	2	C
	Pax	LPD1	LA1	1	2	C

Table 2.20: Performance of the greedy heuristic and branch-and-price algorithm for artificial instances with a cut-off point of one hour.

						Branch-and-Price Algorithm					
Instance	Greedy Solution	Initial LB	#Loadings	#Nodes	#Arcs	Branching Rule[1]	Makespan	LB	T (sec.)	T till Int. Sol. (sec.)	Gap (%)
A1-F	34	33	22	268	8976		34	33	3600		3.03
A1-P	34	33	14	268	8976		34	33	3600		3.03
A1-W	33	23	48	260	8448	DM	23	23	25.94	25.94	
A1-N	33	23	32	260	8448	DM	23	23	15.32	15.32	
A2-F	34	20	43	188	7816	DL	23	20	3600	2042.23	15.00
A2-P	34	20	23	188	7816	DL	23	20	3600	104.43	15.00
A2-W	34	20	331	188	7816	DL	23	20	3600	78.47	15.00
A2-N	33	20	43	182	7324	DM	21	20	3600	8.93	5.00
A3-F	33	23	52	434	22465	DL	32	23	3600	21.8	39.13
A3-P	33	23	30	434	22465		33	23	3600		43.48
A3-W	23	20	61	294	10205	DM	20	20	5.18	5.18	
A3-N	23	20	27	294	10205	DM	20	20	34.76	34.76	
A4-F[2]	15	15	15	266	5388						
A4-P[2]	15	15	10	266	5388						
A4-W[2]	15	15	22	266	5388						
A4-N[2]	15	15	18	266	5388						
A5-F	34	20	61	205	4825		34	20	3600		70.00
A5-P	21	20	23	127	1874	DM	21	21	0.45		
A5-W	34	20	1895	205	4825	DM	21	21	33.30	18.78	
A5-N	22	20	43	133	2053	DM	21	21	4.32	0.94	

[1]: D = depth first, L = least fractional, M = most fractional
[2]: UB = LB, hence the branch-and-price algorithm is not used

Table 2.21: Performance of the greedy heuristic and branch-and-price algorithm for instances with $o = \mathrm{F}$ with a cut-off point of one hour.

						Branch-and-Price Algorithm					
Instance	Greedy Solution	Initial LB	#Loadings	#Nodes	#Arcs	Branching Rule[1]	Makespan	LB	T (sec.)	T till Int. Sol. (sec.)	Gap (%)
1B-1-F	42	41	38	308	12134	DL	42	41	0.74		
1B-1-F	42	41	38	308	12134	DM	42	41	0.36		
1B-2-F	27	25	38	192	4816	DL	27	25	3600		8.00
1B-2-F	27	25	38	192	4816	DM	27	25	2.84		
1C-3-F	41	10	10	461	24224	DL	41	10	1.42		
1C-3-F	41	10	10	462	24224	DM	41	10	1.10		
1C-4-F	14	9	19	264	5608	DL	14	9	0.82		
1C-4-F	14	9	19	264	5608	DM	14	9	0.65		
1C-5-F	41	19	21	484	26204	DL	41	19	1.54		
1C-5-F	41	19	21	484	26204	DM	41	19	1.32		
2B-2-F	41	25	16	366	16080	DL	41	25	1.09		
2B-2-F	41	25	16	366	16080	DM	41	25	0.49		
2B-4-F	8	6	62	164	2811	DL	8	6	3600		33.33
2B-4-F	8	6	62	164	2811	DM	7	7	2.70	2.69	
2B-5-F	26	25	53	428	22127	DL	26	25	2.13		
2B-5-F	26	25	53	428	22127	DM	26	25	1.73		
2C-3-F	36	25	26	448	23933	DL	36	25	1.37		
2C-3-F	36	25	26	448	23933	DM	36	25	1.15		
2C-4-F	11	9	60	182	3649	DL	10	9	3600	370.36	11.11
2C-4-F	11	9	60	182	3649	DM	10	9	3600	5.85	11.11
2C-5-F	26	19	51	344	14243	DL	26	19	1.07		
2C-5-F	26	19	51	344	14243	DM	26	19	0.65		
3A-1-F[2]	41	41	18	132	4452						
3A-2-F	27	25	18	82	1743	DL	27	25	0.19		
3A-2-F	27	25	18	82	1743	DM	27	25	0.14		
3A-5-F	26	11	33	146	4752	DL	26	11	0.59		
3A-5-F	26	11	33	146	4752	DM	26	11	0.34		
3B-2-F	59	41	61	210	11151	DL	59	41	1.26		
3B-2-F	59	41	61	210	11151	DM	59	41	0.65		
3B-5-F	26	11	33	146	4752	DL	26	11	0.61		
3B-5-F	26	11	33	146	4752	DM	26	11	0.36		
3C-3-F	54	26	154	288	18654	DL	54	26	2.09		
3C-3-F	54	26	154	288	18654	DM	54	26	1.54		
3C-4-F	27	11	232	346	14876	DL	27	11	4.96		
3C-4-F	27	11	232	346	14876	DM	27	11	4.50		

[1]: D = depth first, L = least fractional, M = most fractional
[2]: UB = LB, hence the branch-and-price algorithm is not used

Table 2.22: Performance of the greedy heuristic and branch-and-price algorithm for instances with $o = \text{P}$ with a cut-off point of one hour.

						Branch-and-Price Algorithm					
Instance	Greedy Solution	Initial LB	#Loadings	#Nodes	#Arcs	Branching Rule[1]	Makespan	LB	T (sec.)	T till Int. Sol. (sec.)	Gap (%)
1B-1-P[2]	41	41	12	292	10934		41				
1B-2-P	26	25	12	184	4436	DL	26	25	1.10		
1B-2-P	26	25	12	184	4436	DM	26	25	0.77		
1B-4-P	17	9	41	338	9280	DL	17	9	3600		88.89
1B-4-P	17	9	41	338	9280	DM	17	9	3600		88.89
1B-5-P	33	25	77	388	16860	DL	33	25	3600		32.00
1B-5-P	33	25	77	388	16860	DM	33	25	3600		32.00
1C-3-P	39	25	9	438	21808	DL	28	25	3600	53.22	12.00
1C-3-P	39	25	9	438	21808	DM	29	25	3600	16.51	16.00
1C-4-P	14	9	12	264	5608	DL	11	9	3600	14.54	22.22
1C-4-P	14	9	12	264	5608	DM	11	9	3600	9.65	22.22
1C-5-P	39	25	16	460	23676	DL	35	25	3600	14.54	40.00
1C-5-P	39	25	16	460	23676	DM	39	25	3600		56.00
2B-1-P[2]	42	42	11	368	16186						
2B-2-P	26	25	11	216	5670	DL	26	25	0.86		
2B-2-P	26	25	11	216	5670	DM	26	25	0.42		
2B-4-P	8	6	37	164	2811	DL	8	6	3600		33.33
2B-4-P	8	6	37	164	2811	DM	7	6	3600	1.19	16.67
2B-5-P	26	25	31	344	14243	DL	26	25	3600		4.00
2B-5-P	26	25	31	344	14243	DM	26	25	3600		4.00
2C-3-P[2]	25	25	14	280	9304						
2C-4-P	11	9	35	182	3649	DL	9	9	13.59	13.59	
2C-4-P	11	9	35	182	3649	DM	9	9	10.64	10.64	
2C-5-P	25	19	29	330	13097	DL	25	19	3600		31.58
2C-5-P	25	19	29	330	13097	DM	25	19	3600		31.58
3A-1-P[2]	26	11	33	146	4752						
3A-2-P	26	25	13	78	1581	DL	26	25	3600		4.00
3A-2-P	26	11	13	78	1581	DM	26	25	2310.93		
3A-5-P	25	11	22	140	4368	DL	21	11	3600	16.13	90.91
3A-5-P	25	11	22	140	4368	DM	11	11	14.81	14.81	
3B-1-P[2]	74	74	116	264	17586						
3B-2-P[2]	57	57	58	196	9732						
3B-4-P	19	11	96	248	7652	DL	19	11	3600		72.73
3B-4-P	19	11	96	248	7652	DM	19	11	3600		72.73
3B-5-P	25	11	22	140	4368	DL	21	11	3600	12.63	90.91
3B-5-P	25	11	22	140	4368	DM	11	11	10.76	10.76	
3C-3-P	41	26	146	210	9970	DL	41	26	3600		57.69
3C-3-P	41	26	146	210	9970	DM	33	26	3600	71.62	26.92
3C-4-P	26	12	212	346	14876	DL	26	12	3600		116.67
3C-4-P	26	12	212	346	14876	DM	26	12	3600		116.67
3C-5-P	41	39	205	236	12432	DL	41	39	3600		5.13
3C-5-P	41	39	205	236	12432	DM	41	39	3600		5.13

[1]: D = depth first, L = least fractional, M = most fractional

[2]: UB = LB, hence the branch-and-price algorithm is not used

Table 2.23: Performance of the greedy heuristic and branch-and-price algorithm for instances with o = W with a cut-off point of one hour.

						Branch-and-Price Algorithm					
Instance	Greedy Solution	Initial LB	#Loadings	#Nodes	#Arcs	Branching Rule[1]	Makespan	LB	T (sec.)	T till Int. Sol. (sec.)	Gap (%)
1A-1-W[2]	25	25	5	164	3574						
1A-2-W	11	10	5	64	632	DL	11	10	0.21		
1A-2-W	11	10	5	64	632	DM	11	10	0.13		
1A-5-W[2]	9	9	9	84	832						
1B-1-W[2]	41	41	136	292	10934		41				
1B-2-W	26	25	136	184	4436	DL	26	25	0.92		
1B-2-W	26	25	136	184	4436	DM	26	25	0.55		
1C-3-W	11	9	28	270	8480	DL	11	9	0.46		
1C-3-W	11	9	28	270	8480	DM	11	9	0.27		
1C-4-W	25	9	30	108	1124	DL	25	9	3.66		
1C-4-W	25	9	30	108	1124	DM	25	9	2.86		
1C-5-W	25	18	46	292	9564	DL	25	18	1.04		
1C-5-W	25	18	46	292	9564	DM	25	18	0.94		
2A-1-W[2]	25	25	5	198	4762						
2A-2-W	25	9	5	206	5168	DL	25	9	0.25		
2A-2-W	25	9	5	206	5168	DM	25	9	0.18		
2A-5-W	10	9	9	120	1667	DL	10	9	0.13		
2A-5-W	10	9	9	120	1667	DM	10	9	0.15		
2B-2-W	40	25	29	356	15218	DL	40	25	0.78		
2B-2-W	40	25	29	356	15218	DM	40	25	0.42		
2B-4-W	7	6	68	142	2073	DL	7	6	3600		16.67
2B-4-W	7	6	68	142	2073	DM	7	6	0.42		
2B-5-W	31	18	62	414	20693	DL	31	18	1.05		
2B-5-W	31	18	62	414	20693	DM	31	18	0.75		
2C-3-W	25	9	32	98	1187	DL	25	9	0.92		
2C-3-W	25	9	32	98	1187	DM	25	9	0.53		
2C-4-W	10	9	66	164	2915	DL	10	9	3600		11.11
2C-4-W	10	9	66	164	2915	DM	10	9	0.55		
2C-5-W	25	18	60	330	13097	DL	25	18	1.31		
2C-5-W	25	18	60	330	13097	DM	25	18	1.15		
3A-1-W[2]	41	41	18	132	4452						
3A-2-W	27	25	219	82	1743	DL	27	25	1.37		
3A-2-W	27	25	219	82	1743	DM	27	25	1.02		
3A-5-W	25	11	274	140	4368	DL	25	11	1.08		
3A-5-W	25	11	274	140	4368	DM	25	11	0.70		
3B-2-W	43	41	266	146	5423	DL	43	41	1.83		
3B-2-W	43	41	266	146	5423	DM	43	41	1.06		
3B-5-W	25	11	274	140	4368	DL	25	11	1.08		
3B-5-W	25	11	274	140	4368	DM	25	11	0.70		
3C-3-W	53	25	1021	282	17890	DL	53	25	9.54		
3C-3-W	53	25	1021	282	17890	DM	53	25	8.17		
3C-4-W	26	11	1204	346	14876	DL	26	11	4.19		
3C-4-W	26	11	1204	346	14876	DM	26	11	3.08		

[1]: D = depth first, L = least fractional, M = most fractional
[2]: UB = LB, hence the branch-and-price algorithm is not used

Table 2.24: Performance of the greedy heuristic and branch-and-price algorithm for instances with o = N with a cut-off point of one hour.

						Branch-and-Price Algorithm					
Instance	Greedy Solution	Initial LB	#Loadings	#Nodes	#Arcs	Branching Rule[1]	Makespan	LB	T (sec.)	T till Int. Sol. (sec.)	Gap (%)
1A-1-N[2]	25	25	3	164	3574						
1A-2-N	11	10	3	64	632	DL	11	10	0.14		
1A-2-N	11	10	3	64	632	DM	11	10	0.12		
1A-5-N[2]	9	9	5	84	832						
1B-1-N[2]	41	41	13	292	10934						
1B-2-N	26	25	13	184	4436	DL	26	26	843.98		
1B-2-N	26	25	13	184	4436	DM	26	26	2.57		
1B-4-N	15	9	75	246	5688	DL	11	9	3600	49.66	22.22
1B-4-N	15	9	75	246	5688	DM	9	9	120.33	120.33	
1B-5-N	33	25	77	388	16860	DL	32	25	3600	23.93	28.00
1B-5-N	33	25	77	388	16860	DM	26	25	93.55	68.09	
1C-3-N	11	9	24	102	1376	DL	11	9	0.55		
1C-3-N	11	9	24	102	1376	DM	11	9	0.31		
1C-4-N	10	9	28	156	2192	DL	9	9	1.75	1.75	
1C-4-N	10	9	28	156	2192	DM	9	9	0.45	0.44	
1C-5-N[2]	18	18	38	192	4144						
2A-1-N[2]	25	25	5	198	4762						
2A-2-N	10	9	4	56	462	DL	10	9	0.09		
2A-2-N	10	9	4	56	462	DM	10	9	0.05		
2A-5-N[2]	9	9	7	96	1124						
2B-1-N	41	25	12	364	15881	DL	26	26	8.42	7.61	
2B-1-N	41	25	12	364	15881	DM	26	26	0.61	0.36	
2B-2-N	26	25	12	216	5670	DL	25	25	0.67	0.67	
2B-2-N	26	25	12	216	5670	DM	25	25	0.27	0.27	
2B-4-N	8	6	31	164	2811	DL	6	6	1.66	1.66	
2B-4-N	8	6	31	164	2811	DM	6	6	1.37	1.37	
2B-5-N	25	17	28	330	13097	DL	18	18	22.60	21.13	
2B-5-N	25	17	28	330	13097	DM	25	17	3600		47.06
2C-3-N	11	9	13	98	1187	DL	11	9	3600		22.22
2C-3-N	11	9	13	98	1187	DM	11	9	3600		22.22
2C-4-N	11	9	29	182	3649	DL	9	9	2.16	2.16	
2C-4-N	11	9	29	182	3649	DM	9	9	4.73	4.72	
2C-5-N[2]	18	18	26	222	5912						
3A-1-N[2]	41	41	75	132	4452						
3A-2-N	26	25	75	78	1581	DL	26	25	3600		4.00
3A-2-N	26	25	75	78	1581	DM	26	25	58.81		
3A-5-N	18	11	102	98	2128	DL	11	11	8.82	8.82	
3A-5-N	18	11	102	98	2128	DM	11	11	8.51	8.51	
3B-1-N[2]	57	57	109	196	9732						
3B-2-N[2]	41	41	109	132	4452						
3B-4-N	18	11	153	234	6812	DL	11	11	182.08	182.08	
3B-4-N	18	11	153	234	6812	DM	11	11	411.52	411.52	
3B-5-N	18	11	102	98	2128	DL	11	11	6.86	6.86	
3B-5-N	18	11	102	98	2128	DM	11	11	6.80	6.80	
3C-3-N	27	25	437	126	3642	DL	26	26	88.25	69.06	
3C-3-N	27	25	437	126	3642	DM	26	26	102.31	59.68	
3C-4-N	25	11	1030	332	13700	DL	19	11.30	3600	2365.74	59.28
3C-4-N	25	11	1030	332	13700	DM	14	14	1687.62	1649.01	
3C-5-N	41	32	525	236	12432	DL	39	32	3600	700.53	39.00
3C-5-N	41	32	525	236	12432	DM	40	32	3600	117.62	25.00

[1]: D = depth first, L = least fractional, M = most fractional
[2]: UB = LB, hence the branch-and-price algorithm is not used

Chapter 3

Evaluating Ship-to-Shore Schedules using Simulation

This chapter is based on Wagenvoort et al. (2025b).

3.1 Introduction

During military operations in coastal regions, resources are brought from large amphibious ships to the shore using smaller ships and helicopters, called connectors. It is essential to carry out the transportation of the resources efficiently to facilitate the earliest possible start of the tasks on land. Hence, the aim is to schedule the connector trips to the shore such that the makespan, the duration of the operation, is minimised. Therefore, the need for a fast algorithm to construct a schedule for this transportation problem arises. This problem is called the Ship-to-Shore Problem and is described in Chapter 2. Various versions of this problem have been studied, *e.g.*, considering single or multiple resources that should be transported (Christafore Jr., 2017; Danielson, 2018; Strickland, 2018; Villena, 2019; Wagenvoort et al., 2025a).

In the Ship-to-Shore Problem, the aim is to find, for each connector, a route that should be executed such that the makespan is minimised. A route defines a set of round-trips between a sea base (SB) and a landing area (LA) location that should be executed as well as the resources that should be transported in each trip. Based on interviews with experts at the Defence, Safety and Security unit of the Netherlands Organisation for Applied Scientific Research (TNO), we have identified various constraints that have to be adhered to in a schedule. These can be split into constraints regarding the connectors, the delivery of the resources to the shore, and the (un)loading of the resources.

Connectors have a space and weight capacity that determines what set of resources can be simultaneously transported. There can be specific limitations in the way connectors can be loaded, *e.g.*, the load of the connector should be balanced and resources might have to be secured which can only be done at limited spots, restricting the number of ways connectors can be loaded. Furthermore, the connectors have a fuel capacity and might therefore have to be refuelled at a sea base. The speed of a connector can be dependent on the weight of the load on the connector.

Resources have different priority levels that determine a partial ordering of the delivery of resources to the shore. Here a strict ordering exists where all resources with a higher priority should be delivered before resources with a lower priority are delivered. Additionally, certain resources can belong to the same resource set and have to be delivered at the same time or closely after each other, called a *resource set constraint*. An example of a resource set is a unit that trained together plus the vehicles containing their personal supplies. To avoid the situation where either the vehicle or the personnel has to wait on the shore, we impose that these are delivered together or closely after each other (Pagonis and Cruikshank, 1992). The time interval in which resources from a resource set are delivered is called a *delivery wave*. Furthermore, at a sea base and landing area, a limited number of (un)loading spots is available. Hence, there is a limit on the number of connectors that can be (un)loaded at the same time.

In Chapter 2, we consider the ship-to-shore problem as described above, and design an exact branch-and-price algorithm, as well as a greedy heuristic. Christafore Jr. (2017), Danielson (2018), Strickland (2018), and Villena (2019) also study the ship-to-shore problem. Compared to our problem, they do not consider all constraints regarding the coordination of the delivery of resources. Therefore, we use schedules constructed using the method of Chapter 2.

Given a schedule for the Ship-to-Shore Problem, preparations are made accordingly. This implies that switching the order of resources in which they are loaded is not always possible (Vhilen, N, 2022). Once the resources are placed in a certain order on the large amphibious warfare ships, resources scheduled to be transported earlier may block other resources that are scheduled to be transported later. Furthermore, the schedule is communicated to the staff on the ship and they make preparations accordingly. This prevents them from departing to the shore significantly before their planned departure time as they are not ready for departure and might have other conflicting tasks. Therefore, when executing a schedule for the transport operation, the schedule is followed as closely as possible. However, travel times and (un)loading times are stochastic and weather conditions can be different than predicted, also affecting the travel times of the connectors. This means that delays can occur, which can propagate through the schedule as the order in which the resources are loaded onto a connector is fixed when a schedule is executed and connectors are not allowed to depart ahead of time.

Research on the Ship-to-Shore Problem has assumed deterministic parameters regarding the speed and the (un)loading time of the connectors. The resulting schedules might therefore not be robust for delays and this can greatly affect the duration of the transportation of the resources. In general, adding slack to a schedule can help capture these delays and can therefore be beneficial for the realised makespan. One way to add slack is to schedule with more conservative parameters. However, adding too much slack, by being too conservative in the parameters, or by adding slack at the wrong moments, can negatively affect the realised makespan as the order of the connectors is fixed and connectors cannot depart before their scheduled time.

In Chapter 2, we make use of a time-space network to model the Ship-to-Shore Problem. The time-space network consists of nodes that correspond to a location at a certain discrete time period. Arcs connecting nodes correspond to transitions through both time and space. The inputs are deterministic and the length of the discrete time period is set to the maximum (un)loading time of all connectors such that all connectors can be (un)loaded within one time period. Discrete time periods create buffer time in the schedule when a connector can (un)load faster or when the travel time between two locations is not equal to an integer multiple of the time period length. This buffer time could help capture delays. However, the buffer time occurs due to the design choice for the time-space network and is not incorporated as slack by the model to avoid delays. Therefore, it might not be sufficient to handle the delays and it is of interest how well such a schedule performs.

We are therefore interested in the following questions:

Q1: How well does a schedule generated using discrete time periods perform when parameters are stochastic?

Q2: What is the trade-off between using a schedule constructed using more conservative parameters and the realised makespan?

Q3: What is the effect of being less rigid in the execution of a schedule?

Simulation models can be used to model and evaluate the behaviour of a system over time (Scheidegger et al., 2018). Therefore, they can be used to assess the robustness of a planning as well as the effectiveness of new policies (Corpuz et al., 2017; Jnitova et al., 2017; Siswanto et al., 2023). Simulation models can be particularly useful when limited data is available about the performance of a system, as is the case for the Ship-to-Shore Problem which is usually only executed once (Gu et al., 2021).

Horne and Irony (1994) use a simulation to analyse the trade-off between the number of connectors, (un)loading positions, and travel time between the SB and LA. They do however not consider different types of connectors that can use different (un)loading spots, priorities, and resource sets. We therefore develop a simulation model in which a schedule is given as input and its execution is evaluated given uncertainty in the speed and (un)loading time as well as changes in the predicted weather conditions that affect the travel time. The simulation model follows the schedule, by adhering to the order in which connectors are loaded at an SB, and by only allowing connectors to start a limited amount of time ahead of schedule. Furthermore, the simulation model considers the constraints that arise in the Ship-to-Shore Problem.

We use the simulation model to analyse the performance of schedules constructed using the approach from Chapter 2. We find that uncertainty in the parameters has a significant effect on the performance of the schedule. Hence, the buffer time in the schedule arising from the discrete time periods does not suffice to capture these delays. In fact, on average, connectors arrive late in over 30% of their trips. Using a conservative schedule can improve the performance when a more conservative (un)loading time is used. However, using more conservative (un)loading times can produce worse results, especially when you are not interested in the average performance, but a worst-case performance. Being less rigid in the execution by allowing connectors to depart a limited time ahead of schedule generally has a positive effect on the performance of the schedule. However, this comes at the cost of violating the resource set constraints.

This chapter is organised as follows. We describe how we simulate the execution of a schedule and the types of uncertainties we consider in Section 3.2. In Section 3.3, we describe the set-up of the experiments. We report and discuss the results of the experiments in Section 3.4 and give a conclusion in Section 3.5.

3.2 Discrete-Event Simulation

We use a discrete-event simulation in which we take a schedule as input and model the movement of the connectors over time while incorporating uncertainty in deviations from the wind and current used in generating the schedules, the speed of the connectors, and the (un)loading time. The schedule specifies for each connector what trips it should make and thus when it should start (un)loading, when it should depart, and what it should carry in each trip from an SB to an LA. We model the movement of the connectors using the following events: Arrival at an SB, departure from an SB, arrival at an LA, departure from an LA. During the simulation, certain policies have to be adhered to while executing the schedule. In this section, we describe the types of uncertainty we consider in the simulation, the policies we have to adhere to, and the output of the simulation model. An overview of the procedure for each event is given in Appendix 3.A.

3.2.1 Uncertainty in the Ship-to-Shore Problem

During the planning of the Ship-to-Shore Problem, deterministic parameters are used, while in reality these are stochastic. Furthermore, the weather conditions can differ from the expected weather conditions, affecting the speed of the connectors.

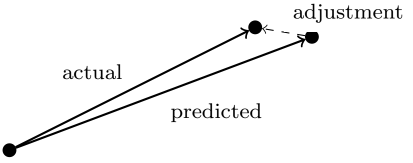


Figure 3.1: Vector representation of the effect of a change in the current/wind in direction compared to the predicted current/wind.

In this section, we explain the types of uncertainty that arise in the execution of the Ship-to-Shore Problem in more detail.

The speed of a connector is used to determine the required travel time between the locations. The speed can be constant or dependent on the weight of the load it carries. However, a connector does not always travel at maximum speed due to, for example, small navigation errors or detours. This will result in a net speed slightly below the maximum speed. We therefore determine the net speed of a connector for each trip made by the connector according to a distribution that is input to the simulation.

Furthermore, the travel time is dependent on the weather conditions. Namely, the current and/or wind affect the speed at which a connector travels. In the planning phase, the predicted water current and/or wind can be taken into account. However, the actual current or wind can be different. We consider a change in the current or wind as denoted by Figure 3.1. Namely, if a planning was made with the net speed and direction according to the predicted current/wind, and the actual current/wind deviates, the speed and direction will have to be adjusted according to the vector labelled as 'adjustment' in Figure 3.1. This affects the travel time as the connector has to adjust its direction for the change in the current/wind (see Example 3.1).

Example 3.1. *Two examples are given in Figure 3.2. Here the angle of the change in the current and/or wind is denoted by α and the strength is r nautical miles. If the direction and strength of the current and/or wind is not taken into account, the connectors will not end up at the landing area (LA), but to the left of it. Therefore, the connectors will have to steer to the right in order to arrive at the LA. The connector will thus travel at a speed v in the denoted direction and have a realised speed of v' towards the LA.*

This means that in case A the realised speed v' is higher than the speed of the connector, and in case B the realised speed v' is lower than the speed of the connector. When returning to the SB from the LA, the reverse will hold.

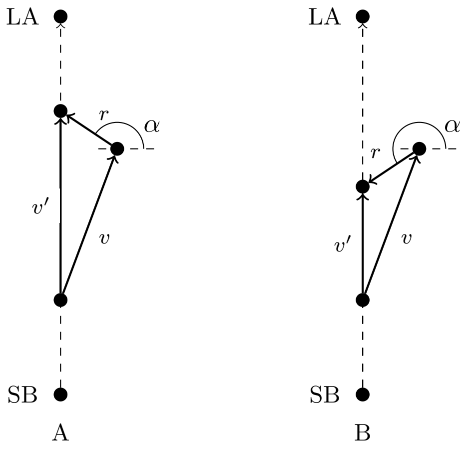


Figure 3.2: Vector representation of the effect of a current or wind in direction α and with strength r on the speed v.

Besides changes in the speed of the connectors, changes in the (un)loading time of a connector at a sea base/landing area can occur. Delays in (un)loading are caused by small delays in placing or removing the resources on/from the connectors. Furthermore, small repairs and refuelling can be done at the sea base which could cause delays. At the landing area, connectors can get stuck in sand and therefore take longer to depart the landing area. Therefore, we determine the (un)loading time of a connector for each time it (un)loads according to a distribution that is input to the simulation.

3.2.2 Simulation Policies

We consider the following simulation policies.

Firstly, we adhere to the (un)loading ordering provided by the schedule. Due to the uncertainty in the simulation parameters, it is possible that a connector that is scheduled to (un)load at a later time than another connector, arrives first. Preparations are made according to the schedule, *e.g.*, at the sea base the resources are gathered that should be loaded onto the connector. In this case, the resources for the next loading should be gathered instead including personnel, which can result in additional delays. Furthermore, this could lead to violations of the priority or resource set constraints. Hence, a connector cannot be (un)loaded before the connectors that are scheduled to (un)load before have finished or started (un)loading.

Secondly, we limit the time the connectors can be ahead of schedule. For the same reasons as mentioned above, being far ahead of schedule can cause difficulties as preparations have not been completed yet. Therefore, we set a limit to how far ahead of schedule a connector can be (un)loaded.

Finally, we adhere to all constraints of the Ship-to-Shore Problem, if possible. This implies that we adhere to the capacity constraints at the SBs and LAs. Namely, there is a maximum number of connectors that can be (un)loaded at the same time at a SB or LA. If a connector arrives and should be (un)loaded next, but all (un)loading spaces are occupied, the connector has to wait. Additionally, some connectors are only compatible with certain (un)loading locations and they are therefore not able to (un)load at a free spot that is not compatible. Furthermore, we adhere to the constraints regarding the ordering of resources. Namely, a strict priority ordering of the resources exists which implies that a connector carrying priority $\pi + 1$ resources will wait until all priority π resources are delivered. Finally, we adhere to the resource set constraints, if possible. These constraints ensure that resources that belong to the same resource set are delivered at the same time or closely after each other. Hence, if it is known in advance that a connector taking part in a delivery wave is delayed, the start of the wave will be postponed. However, it is possible that a connector taking part in a delivery wave experiences a delay when another connector is already delivering resources from that wave. In that case, the wave might be broken up and we consider this a violation of the resource set constraint.

3.2.3 Output of the Simulation Model

Our main interest is the realised makespan when executing a schedule with stochastic parameters. Besides the makespan, we also determine, from the simulation, the percentage of time a connector is late for loading at an SB and the percentage of time a connector is late for delivery at the LA. Additionally, we determine the average delay at an SB or at an LA. Finally, since resource set constraints might be violated, we determine the percentage of delivery waves that are violated. We consider a delivery wave violated when the time between the completion of a delivery till the time of the start of the next delivery of that resource set differs by more than the length of one discrete time period. An overview of the performance measures we consider is given in Table 3.1.

Table 3.1: An overview of the performance measures determined in the simulation model.

Performance Measure	Description
Standardised Makespan	The standardised time until all resources are brought to land, *i.e.*, the realised makespan divided by the planned makespan
Late_{SB}	The percentage of times a connector is late at an SB to be loaded
Late_{LA}	The percentage of times a connector is late at an LA to be unloaded
Delay_{SB}	The average time in minutes a connector arrives after the scheduled time at an SB
$\text{Delay}_{SB,l}$	The average time in minutes a connector arrives after the scheduled time at an SB, given that it was late
Delay_{LA}	The average time in minutes a connector arrives after the scheduled time at an LA
$\text{Delay}_{LA,l}$	The average time in minutes a connector arrives after the scheduled time at an LA, given that it was late
Wave	The percentage of waves that were violated

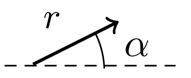


Figure 3.3: Change in the current/wind.

3.3 Experimental Design

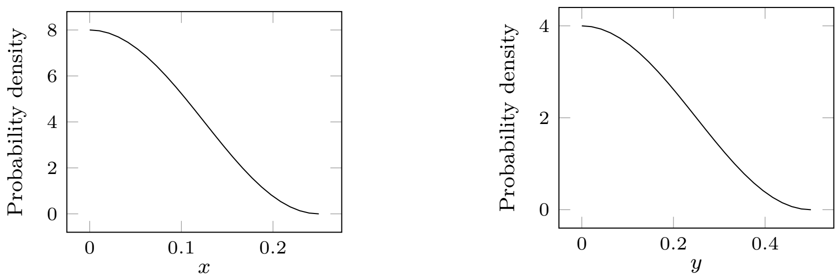
In this section, we describe the design of our experiments. We first describe the distributions we use for the different parameters. Then, we describe which experiments and tests we perform to answer our research questions.

3.3.1 Simulation Parameters

Input for the simulation are the distributions for the stochastic parameters defined in Section 3.2. Here we determine the change in the current/wind at the start of each replication, *i.e.*, this effect is fixed within a replication. The travel time and (un)loading time uncertainty is considered for each individual trip and (un)loading activity. The distribution of the change in current/wind and the speed and (un)loading time is an input to the simulation and can thus be varied. We consider the following distributions in our experiments.

For the change in the current/wind, we consider a direction as an angle, α, and speed in nautical miles, r, as given in Figure 3.3. Here we take $\alpha \sim U(0, 360)$ degrees and $r \sim U(0, 1)$ nautical miles. This implies that each direction is equally likely to occur as well as each speed up to one nautical miles.

For the speed of the connectors, we consider a deviation of the maximum speed.



(a) Distribution for the speed uncertainty.

(b) Distribution for the loading uncertainty.

Figure 3.4: Simulation parameter data.

In other words, the speed the connector has during a trip is equal to $s(1-x)$ where s is the maximum speed of the connector and x a realisation of $X \sim f_X(x) = 4\cos(4\pi x) + 4$ for $0 \leq x \leq 0.25$. This implies that a connector will have a speed of at least 75% of its maximum speed. Furthermore, small reductions in the speed are likely, while large reductions in the speed are unlikely. The distribution of the reduction in the speed is given in Figure 3.4a.

For the (un)loading time of the connectors, we consider an addition to the minimum required (un)loading time. In other words, the time it takes for the connector to be (un)loaded is equal to $t(1+y)$ where t is the minimum required time and y a realisation of $Y \sim f_Y(y) = 2\cos(2\pi y) + 2$ for $0 \leq y \leq 0.5$. This implies that a connector will take at most 1.5 times its minimum required time to be (un)loaded. Furthermore, small increases in the time are likely, while large increases in the time are unlikely. The distribution of the additional time is given in Figure 3.4b.

To make a fair comparison between the simulation output of different schedules, we use different streams of random numbers. Namely, we use one stream of random numbers to determine the deviation of the wind and current, such that the n^{th} replication of each simulation has the same weather conditions. Furthermore, we specify for each connector two separate streams of random numbers, one for the travel times and one for the (un)loading times.

3.3.2 Experiments

In the simulation model, we take a schedule as input. We then simulate the execution of the schedule according to the policies defined in Section 3.2.2 and the distribution for the parameters in Section 3.3.1 for 100,000 replications.

To answer our first research question, we analyse the performance of a schedule constructed using discrete time periods. We therefore compare the simulation output with the planned makespan using a t-test. We analyse the performance with respect to the makespan, the percentage of times a connector is late to (un)load, the average delay when (un)loading, and the percentage of times a wave is violated.

To test the performance of the schedule with respect to the makespan, we test whether the average realised makespan deviates significantly from the planned makespan. Hence, we test the null hypothesis $H_0 : \bar{X} = X_0$ against the alternative hypothesis $H_1 : \bar{X} > X_0$ using a one-sided t-test. Here X_0 is the planned makespan and $\bar{X}$ is the average makespan obtained from the simulation model. To test the performance of the schedule with respect to delays and wave violations, we test whether the average of the metric deviates significantly from 0 and thus use the above described one-sided t-test with $X_0 = 0$ and $\bar{X}$ the average value of the corresponding metric.

Our second research question relates to the trade-off between using more conservative parameters and using the most optimistic parameters in the construction of a schedule. We therefore generate schedules using more conservative parameters by running the branch-and-price algorithm from Chapter 2 using these more conservative parameters for the speed of the connectors or for the (un)loading times. We then simulate the execution of these schedules and compare them to the simulated performance of the base schedule, that is, the schedule created using the maximum speeds and (un)loading times equal to the maximum (un)loading time of all connectors. We test whether these differ significantly from each other using a paired t-test. This test is possible as the samples are dependent due to the usage of different streams of random numbers for different purposes. For example, each i^{th} replication has the same weather conditions because these are generated using a separate stream of random numbers.

Let X_1 and X_2 denote the samples of the execution of the two schedules. Let $x_i = x_{1i} - x_{2i}$ be the difference of pair i between sample 1 and 2 and $\bar{X}_D$ the average of the difference between the pairs. As we are interested in whether the samples are significantly different, we define the null hypothesis $H_0 : \bar{X}_D = 0$ and the alternative hypothesis $H_1 : \bar{X}_D \neq 0$ and apply the paired t-test.

However, in the paired t-test, we compare the averages with each other, while the schedule is only executed once and we thus do not observe the average realised makespan when executing a schedule in practice. We therefore additionally consider a quantile comparison to see whether the schedules perform significantly different at the 95th percentile. We follow the approach of Wilcox et al. (2014)

to determine whether the 95th percentiles of two samples are significantly different (see Algorithm 3.1). In this test, random samples are generated from each original sample and the difference between the 95th percentile in each of the paired samples is determined. Then, the one-sided p-value is determined by counting the number of times the difference is negative or equal. We do this based on $B = 2000$ draws and samples of size $m = 1000000$. The generalised p-value, p^*, can then be determined as $p^* = 2 \min \{p, 1-p\}$ and can be used for a two-sided test.

Algorithm 3.1 Quantile Comparison

Input: x_i, y_i for $i = 1, \ldots, n$ the samples that we wish to compare, B and m are parameters denoting the number of samples and the number of draws per sample, respectively

Output: p, the p-value

for $j = 1 : B$ **do**
  for $k = 1 : m$ **do**
    Let $U_1 \sim U(1, m)$ and $U_2 \sim U(1, m)$.
    Set $x'_k = x_{U_1}$ and $y'_k = y_{U_2}$.
  end for
  Let θ^*_x and θ^*_y be the 95th quantiles of x' and y'.
  Let $d_j = \theta^*_y - \theta^*_x$.
end for
Let $A = \sum_{j=1}^{B} I_{(-\infty,0)}(d_j)$ and $C = \sum_{j=1}^{B} I_{[0]}(d_j)$.
Let $p = \frac{A+0.5C}{B}$.
Let $p^* = 2\min\{p, 1-p\}$.

The final research question relates to the effect of being less rigid in the execution of a schedule. Therefore, we give an additional input to the simulation model defining how many minutes a connector is allowed to be ahead of schedule. We simulate the execution of each schedule while being allowed to be $x \in \{10, 30, 60\}$ minutes ahead of schedule. We then compare the performance of these schedules with the execution of the schedule with $x = 0$. We perform both the paired t-test and the quantile comparison on the samples similarly as for the second research question.

3.4 Results

In this section, the results of our simulation model are presented, where a 5% significance level is used for all tests. The experiments are conducted using 15 instances of which the data is provided by the Royal Netherlands Navy. The instances all have multiple SBs and LAs that are 15 nautical miles apart. There are 4 to 6 connectors available in each instance. The number of resources that are transported during the operation ranges from 20 to 244 items and vary from people to large vehicles. These resources are transported in up to 15 trips taking between 6.5 and 20 hours.

The schedules are generated using deterministic values. However, the conditions during the execution of the schedule can vary as described in Section 3.2.1. Therefore, we perform two experiments. In both experiments, schedules are constructed using deterministic parameters. In the first experiment, we use a schedule constructed using the most optimistic parameters, namely the maximum speed and minimum (un)loading time, which we call the base schedule. We then analyse the effect of stochastic parameters when executing the base schedule using the simulation. In the second experiment, we use more conservative parameters for the speed or (un)loading time to construct a schedule. We then compare the simulation output of these schedules with the simulation output of the base schedule to analyse the effect of using more conservative parameters in the schedule construction. Finally, we analyse the effect of being less rigid in executing a schedule by allowing connectors to (un)load and depart ahead of schedule.

3.4.1 The Performance under Stochastic Parameters

To analyse the effect of stochastic parameters on the execution of a schedule, we simulate the execution of a schedule 100000 times for each instance. The schedule taken as input is generated using the most optimistic parameters for the speed and (un)loading time of the connectors. A summary of the simulation output can be found in Table 3.2. Here, for each of the performance measures, the average mean, the between sample standard error and the worst-case within-sample standard error is given as well as the percentage of instances for which the null hypothesis of the t-test defined in Section 3.3.2 is rejected. We report the worst-case within-sample standard error to give an indication of how much the replications for a given instance could deviate. We see that the within-sample standard errors are low. Therefore, in the remainder of this chapter, we only report the between sample standard errors.

We see that on average, the operation takes 1.8% longer than planned and that for, on average, 47% of the trips, the connector arrives late at the SB. The average delay of these late arrivals is a bit over 5 minutes. The number of trips for which the connector arrives late at the LA is lower, however the delay at the LA is approximately one minute larger than at the SB. This can be caused by the policy that connectors cannot be loaded before the scheduled time at an SB, while it is possible to start unloading at an LA before the scheduled time. Therefore, connectors that arrive

Table 3.2: Descriptive statistics of the simulation output of the base schedules when you are not allowed to be ahead of schedule. For each performance measure we give the average mean, the between sample standard error (SE), the worst-case within-sample standard error (SE_w), and the percentage of instances for which the null hypothesis is rejected.

Performance Measure	Mean	SE	SE_w	% Reject H_0
Standardised Makespan	1.018	0.003	9.6×10^{-5}	100
$Late_{SB}$ (%)	47.04	2.944	0.064	100
$Late_{LA}$ (%)	31.63	3.767	0.089	100
$Delay_{SB}$ (min)	2.68	0.514	0.019	100
$Delay_{SB,l}$ (min)	5.38	0.889	0.032	100
$Delay_{LA}$ (min)	3.10	1.100	0.030	100
$Delay_{LA,l}$ (min)	6.31	1.133	0.034	100
Wave (%)	0.63	0.271	0.040	0

earlier at an LA can already unload and depart to the SB and therefore potentially arrive before the scheduled time, which is less likely to occur at LAs. Waves are rarely violated, which is also caused by the inability of connectors to depart before schedule. Therefore, having a significant time difference between two consecutive deliveries is not likely to occur.

In the t-tests, we test whether the realised performance measures differ significantly from the predicted performance measures. In other words, whether the realised makespan is significantly larger than the planned makespan, the percentage of times a connector is late is significant, the delays are significant, and the percentage of waves that are violated is significant. We find that for all instances, the makespan is significantly larger than planned and that connectors incur significant delays. However, the percentage of waves that are violated is not significant. The results show that the buffer time arising in the schedule from the discrete time periods is not sufficient to capture delays when the most optimistic parameters are used to construct the schedule.

We thus find that executing the schedule takes significantly longer than the planned duration. Namely, on average, the realised makespan is 1.8% higher than the planned makespan. However, if connectors would always travel at average speed and have an average (un)loading time, the realised makespan would be on average 2.8% higher. Hence, although the realised makespan is significantly higher, the variability in the speed and (un)loading time results in a better solution compared to when the average speed and (un)loading time is observed.

3.4.2 The Performance of more Conservative Schedules

To analyse the effect of using more conservative parameters, we determine a schedule using more conservative parameters for the speed of connectors or the (un)loading time using the branch-and-price algorithm of Chapter 2. Using a paired t-test we analyse whether the more conservative parameters have a positive, negative, or insignificant effect on the realised makespan.

Table 3.3 shows the average simulation output for a given reduction in the speed and increase in the (un)loading time. The percentage of the maximum speed or minimum (un)loading time is given by p_s and p_t, respectively. The corresponding statistical value is given by v_s and v_t, respectively, where 'pctl' is used as short for percentile. The average realised makespan is scaled to the planned makespan according to the base schedule constructed using the most optimistic parameters. We see that the average makespan over the instances is increasing in the conservativeness for the (un)loading time. We see that, generally, the number and value of delays is decreasing as well as the probability that waves are violated. This is not surprising, since connectors are not allowed to be ahead of schedule and more time is scheduled for (un)loading. This allows for capturing delays better.

Planning more conservatively in terms of the speed, results in a realised makespan that is only slightly higher than the average realised makespan for the base schedule given in Table 3.2. This is likely caused by the fact that there is already some buffer time to capture delays in the basic schedule due to the discretisation of the time periods. Using a slightly slower speed can therefore result in the same optimal schedule. For example, if first 2.5 time periods are required and therefore 3 are scheduled, a slight reduction in the speed used for planning might result in requiring 2.7 time periods for which still 3 time periods are scheduled. If this is the case for all connectors, the optimal solution is the same resulting in an insignificant output of the paired t-test. If the speed reduction does result in requiring an additional time period in order to transport the resources, this has a relatively large effect as a full extra time period is scheduled for each trip between a sea base and landing area, and/or reverse. For example, if first 2.9 time periods are required and therefore 3 are scheduled, a slight reduction in the speed can result in requiring 3.1 time periods and therefore 4 need to be scheduled when this connector is used resulting in a different optimal solution.

Since there is no effect of different parameters for the speed in our results, we jointly report them in the remainder of this chapter.

Table 3.4 gives the results of the paired t-test. Here, for a given percentage of the maximum speed and minimum (un)loading time, Table 3.4 indicates the percentage of instances a positive, negative, or insignificant change in the realised makespan is observed. We see that being a bit more conservative in the (un)loading time,

Table 3.3: Descriptive statistics of the simulation output of the schedules using more conservative parameters for the speed (s) or (un)loading time (t). The percentage of the maximum speed and minimum (un)loading time used is given by p_s and p_t, respectively. The corresponding statistical value is given by v_s and v_t, respectively. For each performance measure, the average mean is given with the corresponding standard error in brackets. The makespan is scaled with the planned makespan for that schedule.

v_s	p_s(%)	v_t	p_t(%)	Standardised Makespan	Late$_{SB}$	Late$_{LA}$	Delay$_{SB}$	Delay$_{SB,l}$	Delay$_{LA}$	Delay$_{LA,l}$	Wave
-	100	5 pctl.	101.3	0.994	52.809	60.855	4.828	8.656	10.218	15.739	2.095
				(0.006)	(2.407)	(4.834)	(0.814)	(1.106)	(1.324)	(0.898)	(0.895)
-	100	10 pctl.	102.5	0.998	49.260	57.342	3.819	7.233	8.502	13.735	1.960
				(0.005)	(2.479)	(4.784)	(0.609)	(0.861)	(1.058)	(0.673)	(0.831)
-	100	15 pctl.	103.8	1.002	45.906	53.510	3.075	6.203	7.098	12.074	1.781
				(0.005)	(2.531)	(4.629)	(0.421)	(0.639)	(0.831)	(0.467)	(0.747)
-	100	25 pctl.	106.3	1.014	39.339	44.588	1.881	4.388	4.736	9.252	1.355
				(0.005)	(2.55)	(3.993)	(0.168)	(0.341)	(0.493)	(0.201)	(0.555)
-	100	mean	113.2	1.023	42.158	29.900	2.882	6.368	3.015	7.632	1.008
				(0.005)	(2.085)	(2.969)	(0.376)	(0.508)	(0.399)	(0.315)	(0.444)
5 pctl.	99.4	-	100	1.019	46.095	28.403	2.217	4.620	2.045	5.230	0.632
				(0.005)	(2.943)	(1.492)	(0.152)	(0.308)	(0.204)	(0.226)	(0.271)
10 pctl.	98.7	-	100	1.019	46.095	28.403	2.217	4.620	2.045	5.230	0.632
				(0.005)	(2.943)	(1.492)	(0.152)	(0.308)	(0.204)	(0.226)	(0.271)
15 pctl.	98.1	-	100	1.019	46.095	28.403	2.217	4.620	2.045	5.230	0.632
				(0.005)	(2.943)	(1.492)	(0.152)	(0.308)	(0.204)	(0.226)	(0.271)
25 pctl.	96.8	-	100	1.019	46.095	28.403	2.217	4.620	2.045	5.230	0.632
				(0.005)	(2.943)	(1.492)	(0.152)	(0.308)	(0.204)	(0.226)	(0.271)
mean	93.4	-	100	1.019	46.095	28.403	2.217	4.620	2.045	5.230	0.632
				(0.005)	(2.943)	(1.492)	(0.152)	(0.308)	(0.204)	(0.226)	(0.271)

Table 3.4: Results of the paired T-test and quantile comparison of the standardised makespan comparing the simulation output of the schedule with more conservative parameters for the speed (s) or (un)loading time (t) with the output of the base schedule. The percentage of the maximum speed and minimum (un)loading time used is given by p_s and p_t, respectively. The corresponding statistical value is given by v_s and v_t, respectively. We denote the percentage of instance for which there is a positive, negative, or insignificant effect. A positive effect implies that the schedule outperforms the base schedule. Here $*$ corresponds to the case $v_s \in$ {5 pctl,10 pctl,15 pctl,25 pctl,mean} and $p_s \in$ {99.4,98.7,98.1,96.8,93.4}.

				Paired T-test			Quantile comparison		
v_s	p_s(%)	v_t	p_t(%)	pos.	neg.	insign.	pos.	neg.	insign.
-	100	5 pctl	101.3	93.33	6.67	0	86.67	6.67	6.67
-	100	10 pctl	102.5	93.33	6.67	0	80.00	13.33	6.67
-	100	15 pctl	103.8	93.33	6.67	0	46.67	20.00	33.33
-	100	25 pctl	106.3	93.33	6.67	0	6.67	93.33	0
-	100	mean	113.2	6.67	93.33	0	6.67	93.33	0
$*$	$*$	-	100	0	6.67	93.33	0	6.67	93.33

can have a positive effect on the realised makespan. When the speed is decreased, the realised makespan on all instances is higher or insignificantly different from the realised makespan of the basic schedule. The results thus suggest using a slightly more conservative (un)loading time when constructing a schedule, and therefore a slightly longer time period length, is a more effective way to construct a schedule.

The negative and insignificant results of the schedules constructed using more conservative speeds is likely caused by the two situations that can occur. Namely, a slight reduction can either result in the same optimal schedule as the number of required time periods to travel between the SBs and LAs remains the same. This gives an insignificant difference in the realised makespan as the schedules, and hence the simulation outputs, are identical. Alternatively, a slight reduction can result in scheduling a full additional time period when scheduling a trip from a sea base to a landing are, and/or reverse. This can result in a negative effect on the realised makespan as connectors are not allowed to be ahead of schedule and planning a full additional time period can result in a schedule that is too conservative.

Increasing the (un)loading time of the connector affects the schedule in two ways. Firstly, the time scheduled for (un)loading increases, adding more buffer time in the schedule for delays. Secondly, the time required for a connector to travel between an SB and an LA can both increase or decrease. For example, if initially 15-minute time periods are used and it takes 40 minutes to travel from an SB to an LA, 45 minutes (3 time periods) are scheduled. If the time period length is increased to 20 minutes, 40 minutes are scheduled (2 time periods), reducing the scheduled time. If however the time period length is increased to 17.5 minutes, 52.5 minutes are scheduled (3 time periods), increasing the scheduled time. Therefore, changing the time period

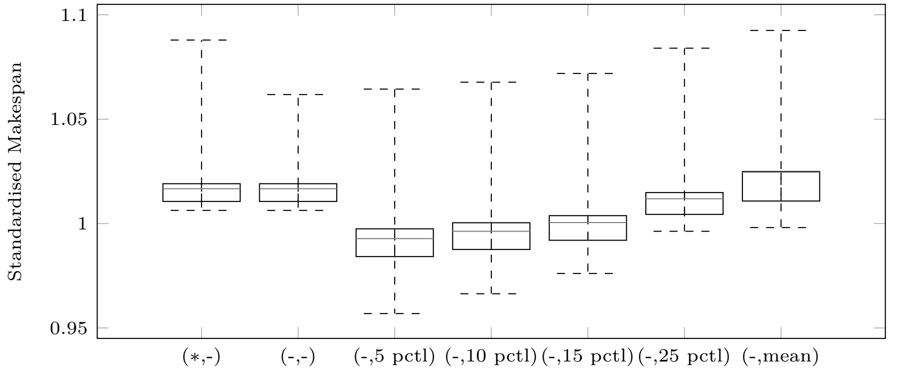


Figure 3.5: Simulation output for the average realised makespan scaled to the planned makespan of the base schedule. Here the horizontal axis denotes the types of schedule used of the form (v_s,v_t) and $*$ corresponds to the case $v_s \in \{5 \text{ pctl},10 \text{ pctl},15 \text{ pctl},25 \text{ pctl},\text{mean}\}$ and $p_s \in \{99.4,98.7,98.1,96.8,93.4\}$.

length will always increase the time scheduled for (un)loading, but can both increase or decrease the time scheduled for travelling between the locations.

These effects can be seen in Figure 3.5. Here we present the distribution of the average realised makespan of the instances for the types of schedules. These show that schedules constructed using a more conservative speed can result in a schedule with a higher makespan and therefore higher realised makespan, but are also likely to result in the same schedule and therefore same realised makespan. When the time period length is adjusted, resulting schedules can both be faster and slower.

In practice, a schedule is only executed once and the mean is thus not observed. Therefore, we are also interested in a worst-case performance and we analyse the performance at the 95th percentile. When looking at the quantile comparison, we see that only for the small increases in the time period length, the conservative schedule performs better and that more often the results are insignificant or negative. Hence, the best performance is observed when being only slightly conservative in the (un)loading time for connectors.

3.4.3 Price of Being Rigid

The previous results are generated given the policy that a connector is not allowed to be ahead of schedule, which is imposed as preparations are made according to the schedule. This can however have a large effect on the realised makespan as can be seen in Figure 3.6 which provides an example of an instance that shows the distribution

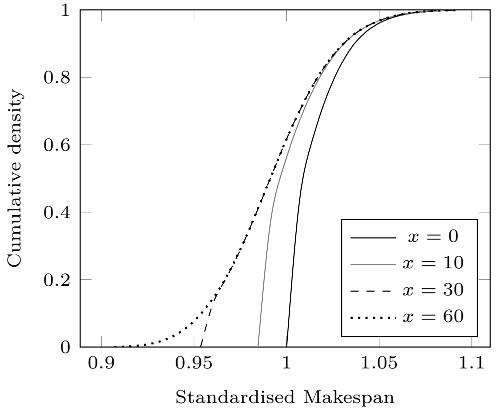
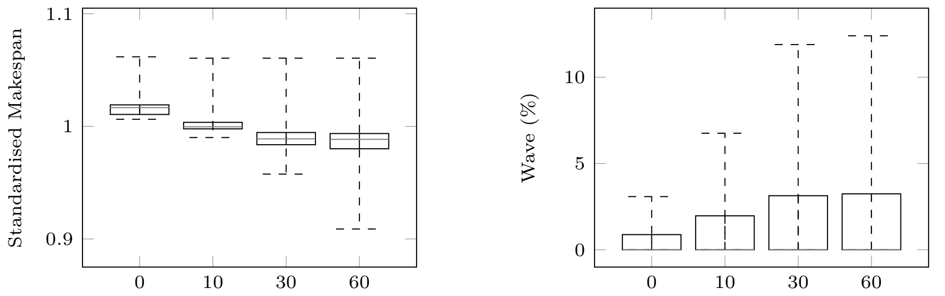


Figure 3.6: The effect on the distribution of the makespan when you are allowed to be x minutes ahead of time.

of the realised makespan as a fraction of the planned makespan. The figure shows that being allowed to be ahead of schedule has a positive effect on the expected makespan, but that the effect stagnates the more you are allowed to be ahead of schedule. Namely, the density curve is steep when you are allowed to be no or a limited time ahead of schedule, as there is a high chance you are close to the lowest possible makespan you can attain. However, when you are allowed to be significantly ahead of schedule the effect stagnates as you are not likely to be significantly ahead of schedule, as can be seen by the scenario $x = 60$. When comparing at the 95th percentile, the performance seems to be similar. Therefore, we analyse the effect of being allowed to be x minutes ahead of time, for $x \in \{10, 30, 60\}$, using a paired t-test and quantile comparison with respect to the scenario in which you are not allowed to be ahead of schedule.

The simulation output when you are allowed to be 10, 30 or 60 minutes ahead of schedule, can be found in Appendix 3.B. We highlight some results in Figure 3.7 regarding the change in the realised makespan and percentage of wave violations for the base schedule. Figure 3.7a shows a decreasing trend in the realised makespan when the minutes you are allowed to be ahead of schedule increases. We see that for some instances this flexibility does not provide much reduction in the realised makespan, but for others the makespan can decrease significantly.

Figure 3.7b shows the percentage of wave violations. We see an increasing trend in the wave violations, hence, the more you are allowed to be ahead of schedule, the higher the probability of violating the resource set constraints. This increase comes



(a) Simulation output for the base schedules for the average realised makespan when you are allowed to be $x \in \{0, 10, 30, 60\}$ minutes ahead of schedule.

(b) Simulation output for the base schedules for the percentage of wave violations when you are allowed to be $x \in \{0, 10, 30, 60\}$ minutes ahead of schedule.

Figure 3.7: Simulation output for the base schedules when you are allowed to be $x \in \{0, 10, 30, 60\}$ minutes ahead of schedule.

from the fact that connectors only wait if they know the other connectors joining in the wave will be delayed. If they are still on track, a connector can depart earlier, which can result in a wave violation if the later connectors experience a delay while loading or travelling. When you are allowed to be ahead of schedule, you are thus at a higher risk of violating a resource set constraint. This can result in undesirable and dangerous situations where units that trained together arrive separately from each other and/or the vehicles and other resources they require for their further operation on land. This can result in delays in the operations on land and blockages at the landing area.

In the paired t-test, we compare the realised performance measure when you are not allowed to be ahead of schedule with the corresponding realised performance measure when you are allowed to be $x \in \{10, 30, 60\}$ minutes ahead of schedule. We find that for all instances, and all schedules, the realised makespan is significantly lower when you are allowed to be ahead of schedule. However, the results from the quantile comparison show that at the 95th percentile, the realised makespans do not differ significantly. Hence, being less rigid will on average ensure the duration of the operation is significantly lower, but in the worst-case the performance is the same. For the basic schedule, the average realised makespan is 1.8% longer than scheduled (Table 3.2), which reduces to 0.4%, -0.8%, or -1.3% when you are allowed to be 10, 30, or 60 minutes ahead of schedule, respectively.

We also find that the percentage of times a connector is late to (un)load reduced significantly when you are allowed to be ahead of schedule. Furthermore, the average

Table 3.5: Results of the paired T-test comparing the simulation output for the conditional delay at an SB ($\text{Delay}_{SB,l}$) of the schedule where you are allowed to be x minutes ahead of schedule with the output of the schedule with $x = 0$. We denote the percentage of instances for which there is a positive, negative, or insignificant effect. A positive effect implies that being allowed to be x minutes ahead of schedule outperforms $x = 0$. The percentage of the maximum speed and minimum (un)loading time used is given by p_s and p_t, respectively. The corresponding statistical value is given by v_s and v_t, respectively.Here $*$ corresponds to the case $v_s \in \{5 \text{ pctl}, 10 \text{ pctl}, 15 \text{ pctl}, 25 \text{ pctl}, \text{mean}\}$ and $p_s \in \{99.4, 98.7, 98.1, 96.8, 93.4\}$.

				$x = 10$			$x = 30$			$x = 60$		
v_s	p_s(%)	v_t	p_t(%)	pos.	neg.	insign.	pos.	neg.	insign.	pos.	neg.	insign.
-	100	-	100	80.00	20.00	0	80.00	6.67	13.33	80.00	6.67	13.33
-	100	5 pctl	101.3	46.67	53.33	0	46.67	53.33	0	46.67	53.33	0
-	100	10 pctl	102.5	46.67	53.33	0	46.67	53.33	0	46.67	53.33	0
-	100	15 pctl	103.8	46.67	53.33	0	46.67	53.33	0	46.67	53.33	0
-	100	25 pctl	106.3	46.67	53.33	0	46.67	53.33	0	46.67	53.33	0
-	100	mean	113.2	73.33	20.00	6.67	86.67	0	13.33	86.67	0	13.33
*	*	-	100	80.00	20.00	0	80.00	6.67	13.33	80.00	6.67	13.33

delay of connectors is significantly lower as well. However, the average delay given that you are late, is not always significantly lower. The conditional delays are given in Tables 3.5 and 3.6 which show that for some instances the effect on the conditional delay is negative or insignificant.

We can identify two types of delays. First, incidental delays, which are delays occurring at one event. For example, a delay during the last trip of a connector, or a small delay upon arrival at a location, due to a longer (un)loading time or travel time, that can be compensated before it arrives at its next location. Second, propagating delays, which are delays that cause delays in the next arrivals too. For example, a connector can have a delay during one of its first trips, which causes subsequent delays as it requires (almost) the full time period to (un)load and/or (almost) the full time periods to travel between the sea base and landing area.

When a connector is allowed to depart before schedule, the first type of delay is less likely to occur. Namely, a connector can be ahead of time when starting a trip, and therefore still finish on time even though it takes longer to (un)load or travel. However, the second type can still occur, namely, delays at the start can still propagate through the schedule as the connector did not benefit (much) from being allowed to depart before the scheduled time. This can negatively affect the conditional delay.

Table 3.7 shows the results of the paired t-test on the wave violations. It shows that the results are either insignificant or negative. In the cases where the wave violations do not differ significantly, no wave violations were present. After inspection of these solutions, it appeared that the resource sets were small and either all resources

Table 3.6: Results of the paired T-test comparing the simulation output for the conditional delay at an LA ($\text{Delay}_{LA,l}$) of the schedule where you are allowed to be x minutes ahead of schedule with the output of the schedule with $x = 0$. We denote the percentage of instances for which there is a positive, negative, or insignificant effect. A positive effect implies that being allowed to be x minutes ahead of schedule outperforms $x = 0$. The percentage of the maximum speed and minimum (un)loading time used is given by p_s and p_t, respectively. The corresponding statistical value is given by v_s and v_t, respectively.Here $*$ corresponds to the case $v_s \in \{5 \text{ pctl}, 10 \text{ pctl}, 15 \text{ pctl}, 25 \text{ pctl}, \text{mean}\}$ and $p_s \in \{99.4, 98.7, 98.1, 96.8, 93.4\}$.

				$x = 10$			$x = 30$			$x = 60$		
vs	$p_s(\%)$	t	$p_t(\%)$	pos.	neg.	insign.	pos.	neg.	insign.	pos.	neg.	insign.
-	100	-	100	100	0	0	100	0	0	100	0	0
-	100	5 pctl	101.3	46.67	53.33	0	33.33	66.67	0	33.33	66.67	0
-	100	10 pctl	102.5	66.67	33.33	0	53.33	46.67	0	53.33	46.67	0
-	100	15 pctl	103.8	80.00	20.00	0	80.00	20.00	0	80.00	20.00	0
-	100	25 pctl	106.3	100	0	0	100	0	0	100	0	0
-	100	mean	113.2	100	0	0	100	0	0	100	0	0
$*$	$*$	-	100	100	0	0	100	0	0	100	0	0

Table 3.7: Results of the paired T-test comparing the simulation output for the wave violations of the schedule where you are allowed to be x minutes ahead of schedule with the output of the schedule with $x = 0$. We denote the percentage of instances for which there is a positive, negative, or insignificant effect. A positive effect implies that being allowed to be x minutes ahead of schedule outperforms $x = 0$. The percentage of the maximum speed and minimum (un)loading time used is given by p_s and p_t, respectively. The corresponding statistical value is given by v_s and v_t, respectively.Here $*$ corresponds to the case $v_s \in \{5 \text{ pctl}, 10 \text{ pctl}, 15 \text{ pctl}, 25 \text{ pctl}, \text{mean}\}$ and $p_s \in \{99.4, 98.7, 98.1, 96.8, 93.4\}$.

				$x = 10$			$x = 30$			$x = 60$		
v_s	$p_s(\%)$	v_t	$p_t(\%)$	pos.	neg.	insign.	pos.	neg.	insign.	pos.	neg.	insign.
-	100	-	100	0	33.33	66.67	0	33.33	66.67	0	33.33	66.67
-	100	5 pctl	101.3	0	33.33	66.67	0	33.33	66.67	0	33.33	66.67
-	100	10 pctl	102.5	0	33.33	66.67	0	33.33	66.67	0	33.33	66.67
-	100	15 pctl	103.8	0	33.33	66.67	0	33.33	66.67	0	33.33	66.67
-	100	25 pctl	106.3	0	33.33	66.67	0	33.33	66.67	0	33.33	66.67
-	100	mean	113.2	0	33.33	66.67	0	33.33	66.67	0	33.33	66.67
$*$	$*$	-	100	0	33.33	66.67	0	33.33	66.67	0	33.33	66.67

were transported on one large connector, or the wave was small and had high priority such that they were transported immediately at the start of the operation and hence they could not experience large enough delays for there to be a violation. In the other cases, we observe a negative effect on the wave violations caused by the higher risk of violating a resource set constraint when you are allowed to be ahead of schedule.

3.5 Conclusion

In this chapter, we developed a discrete event simulation model that can be used to analyse the performance of a deterministically constructed schedule when these parameters are actually stochastic. This simulation model follows certain policies specifying the order of events and the amount of time connectors are allowed to be ahead of schedule and can be used to evaluate the realised makespan, the percentage of times a connector is late, the average observed delays of connectors, and the percentage of times a resource set constraint is violated. Using such a simulation model can give interesting insights into the effect of using more conservative parameters in the scheduling phase as well as the effect of using different policies in the execution of a schedule. Especially in situations where a schedule is usually only executed once, as in the Ship-to-Shore Problem, a simulation model can be used to gather insights as new policies cannot be evaluated in experiments.

We use the simulation model to evaluate the performance of a schedule under uncertainty regarding the speed and (un)loading times of connectors as well as the weather conditions. We use a schedule constructed using discrete time periods to analyse whether the room in the schedule is able to capture these delays. Furthermore, we analyse the effect of using more conservative parameters in the construction of a ship-to-shore schedule and the effect of being less rigid in the execution of a schedule, *i.e.*, by allowing connectors to depart earlier than the predicted scheduled time.

Using schedules constructed using the branch-and-price algorithm with discretised time periods from Chapter 2, we find that the realised makespan is significantly higher (1.8%) when parameters are stochastic, hence, the room in the schedule arising from the usage of discretised time periods does not suffice to compensate the delay incurred. In more than 30% of the trips, a connector arrives significantly late at the sea base or landing area. However, the resource set constraints are not significantly often violated.

For the same instances, we generated schedules constructed using the branch-and-price algorithm using more conservative parameters. Here we either used a more conservative speed or a more conservative (un)loading time. We find that using a more conservative speed does not result in a significantly better performance. In fact, this resulted in either an insignificant or a negative effect on the realised makespan. Using a more conservative speed can result in the exact same schedule due to the discretisation of the time periods. When a more conservative (un)loading time is used, we found for most instances a positive significant effect. However, being too conservative can have a counterproductive effect. Since a schedule is not likely to be executed often, the mean performance is never observed. Therefore, we also look at a worst-case performance by evaluating the performance at the 95th percentile. We find that this negative effect being too conservative can have, is observed earlier when the worst-case is considered.

One of the simulation policies is that a connector cannot be ahead of schedule. This can have a large effect on the realised makespan as only negative effects of the stochastic parameters are observed. We therefore consider what happens when the execution of a schedule is less rigid, *i.e.*, a connector is allowed to be a limited time ahead of schedule. Here we considered 10, 30, and 60 minutes. This has a positive significant effect on the realised makespan and the percentage of time a connector is late. However, this comes at a risk of violating the resource set constraints.

In practice, if a slight increase in the probability of violating resource set constraints is acceptable, we recommend adopting a less rigid approach in the execution of a schedule. Our findings indicate that allowing departures to be up to ten minutes ahead of schedule can lead to significant reductions in the realised makespan. This strategy offers more consistent reduction compared to constructing the schedule using more conservative parameters. However, if minimising wave violations is a priority, we recommend using more conservative parameters for the (un)loading time in constructing the schedule. Although these showed both positive and negative effects on the realised makespan, we observe only negative and insignificant results when adjusting the speed of the connectors. To limit negative effects, we advise analysing the buffer time in connector trips before applying more conservative parameters. Furthermore, we advise using only slightly longer (un)loading times to maximise the chance of a smaller expected and worst-case realised makespan.

Appendix

3.A Events in Discrete-Event Simulation

For each of the events in the discrete-event simulation, we present a flow chart (Figure 3.8 - 3.11) describing the procedure that is followed. Here, t is the time at which the event takes place, t_{plan} is the time at which the next (un)loading event is scheduled to start, and δ is the amount of time a connector is allowed to be ahead of schedule. The simulation is initialised by scheduling the first arrival event for each connector.

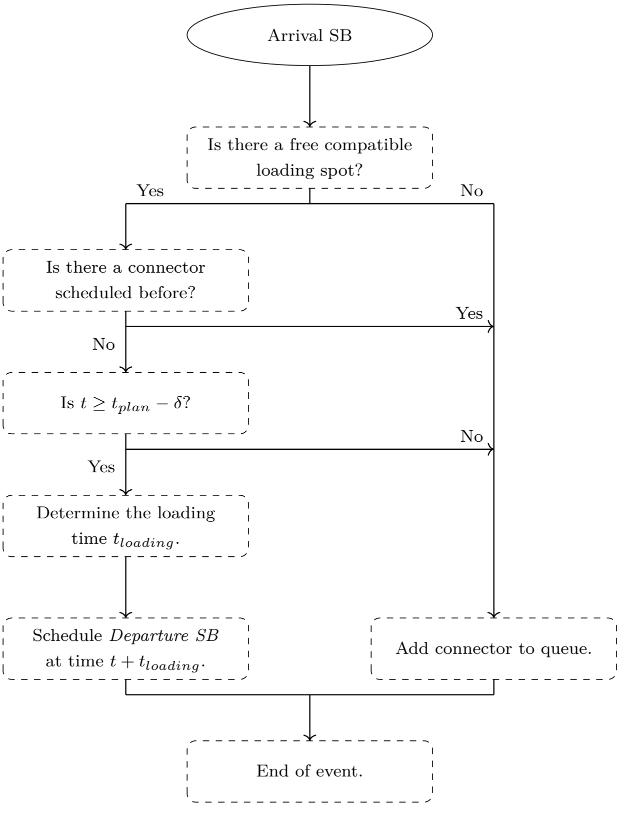


Figure 3.8: Flow chart for the event *Arrival SB*. Here the current time is t, t_{plan} is the time at which the loading is scheduled to start, and δ is the amount of time a connector is allowed to be ahead of schedule.

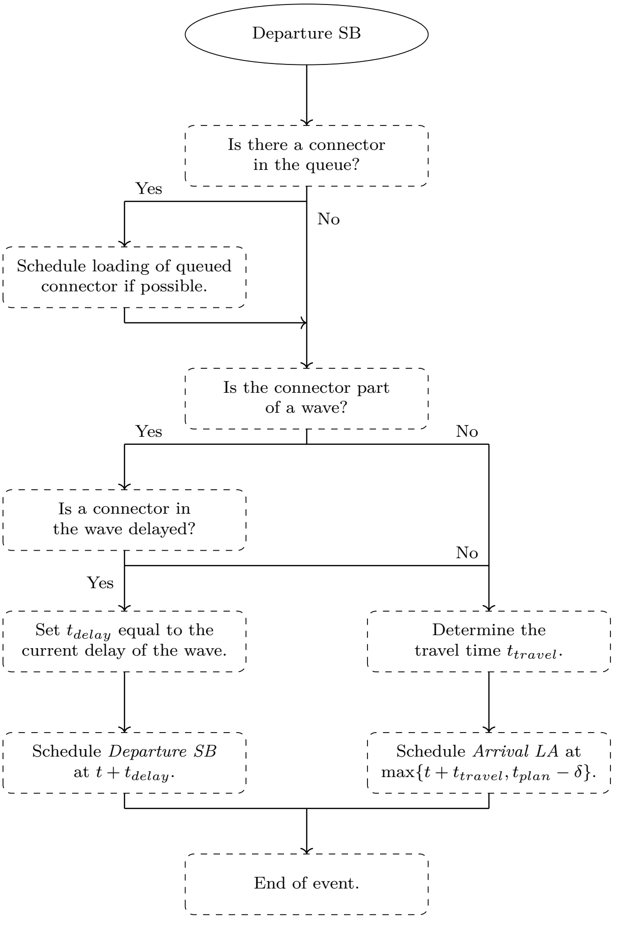


Figure 3.9: Flow chart for the event *Departure SB*. Here the current time is t, t_{plan} is the time at which the next loading is scheduled to start, and δ is the amount of time a connector is allowed to be ahead of schedule.

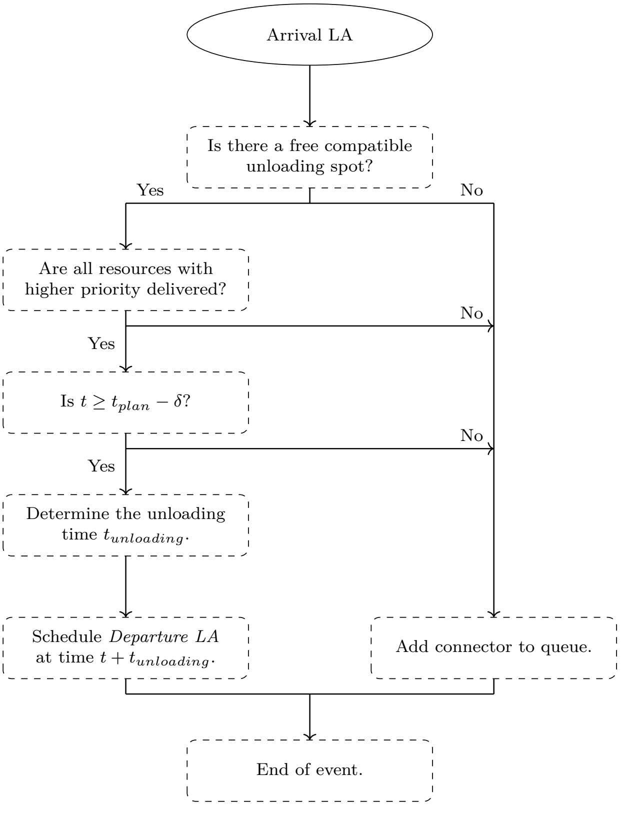


Figure 3.10: Flow chart for the event *Arrival LA*. Here the current time is t, t_{plan} is the time at which the unloading is scheduled to start, and δ is the amount of time a connector is allowed to be ahead of schedule.

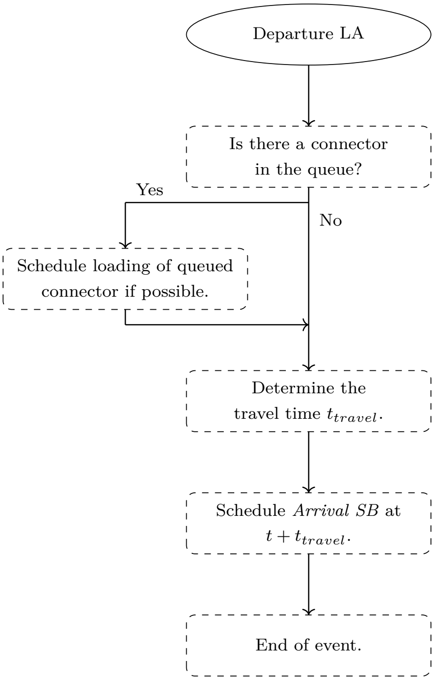


Figure 3.11: Flow chart for the event *Departure LA*. Here the current time is t, t_{plan} is the time at which the next unloading is scheduled to start, and δ is the amount of time a connector is allowed to be ahead of schedule.

3.B Simulation Output

Table 3.8: Descriptive statistics of the simulation output when you are allowed to be $x = 10$ minutes ahead of schedule. The makespan is scaled with the planned makespan for that schedule. The percentage of the maximum speed and minimum (un)loading time used is given by p_s and p_t, respectively. The corresponding statistical value is given by v_s and v_t, respectively. Here $*$ corresponds to the case $v_s \in \{5 \text{ pctl}, 10 \text{ pctl}, 15 \text{ pctl}, 25 \text{ pctl}, \text{mean}\}$ and $p_s \in \{99.4, 98.7, 98.1, 96.8, 93.4\}$.

v_s	p_s(%)	v_t	p_t(%)	Makespan	Late$_{SB}$	Late$_{LA}$	Delay$_{SB}$	Delay$_{SB,l}$	Delay$_{LA}$	Delay$_{LA,l}$	Wave
-	100	-	100	1.004	25.142	22.093	1.720	4.642	2.536	5.279	1.245
				(0.004)	(2.549)	(4.176)	(0.554)	(0.887)	(1.108)	(1.17)	(0.526)
-	100	5 pctl	101.3	0.991	39.634	56.285	4.393	9.653	9.765	15.863	2.304
				(0.006)	(2.766)	(4.806)	(0.797)	(0.967)	(1.255)	(0.807)	(0.992)
-	100	10 pctl	102.5	0.993	35.556	51.796	3.361	7.914	7.906	13.56	2.259
				(0.006)	(2.331)	(4.443)	(0.575)	(0.72)	(0.962)	(0.582)	(0.971)
-	100	15 pctl	103.8	0.996	31.076	47.308	2.582	6.529	6.427	11.619	2.180
				(0.006)	(2.055)	(4.101)	(0.38)	(0.478)	(0.717)	(0.371)	(0.934)
-	100	25 pctl	106.3	1.004	22.982	35.632	1.360	4.141	3.840	8.278	1.897
				(0.005)	(1.663)	(3.011)	(0.132)	(0.168)	(0.36)	(0.147)	(0.805)
-	100	mean	113.2	1.012	25.827	23.331	1.887	5.421	2.379	6.149	1.696
				(0.005)	(1.961)	(2.224)	(0.251)	(0.319)	(0.291)	(0.217)	(0.723)
$*$	$*$	-	100	1.005	22.987	18.277	1.182	3.733	1.446	4.086	1.245
				(0.005)	(1.796)	(1.167)	(0.087)	(0.113)	(0.134)	(0.117)	(0.526)

Table 3.9: Descriptive statistics of the simulation output when you are allowed to be $x = 30$ minutes ahead of schedule. The makespan is scaled with the planned makespan for that schedule. The percentage of the maximum speed and minimum (un)loading time used is given by p_s and p_t, respectively. The corresponding statistical value is given by v_s and v_t, respectively. Here $*$ corresponds to the case $v_s \in \{5 \text{ pctl}, 10 \text{ pctl}, 15 \text{ pctl}, 25 \text{ pctl}, \text{mean}\}$ and $p_s \in \{99.4, 98.7, 98.1, 96.8, 93.4\}$.

v_s	p_s(%)	v_t	p_t(%)	Makespan	Late_{SB}	Late_{LA}	Delay_{SB}	$\text{Delay}_{SB,l}$	Delay_{LA}	$\text{Delay}_{LA,l}$	Wave
-	100	-	100	0.992	24.239	21.358	1.674	4.516	2.494	5.150	2.140
				(0.005)	(2.65)	(4.202)	(0.556)	(0.895)	(1.109)	(1.177)	(0.92)
-	100	5 pctl	101.3	0.989	39.579	55.354	4.388	9.641	9.693	15.943	2.346
				(0.007)	(2.742)	(4.673)	(0.793)	(0.958)	(1.249)	(0.784)	(1.012)
-	100	10 pctl	102.5	0.990	35.467	50.73	3.353	7.894	7.828	13.612	2.352
				(0.007)	(2.295)	(4.269)	(0.568)	(0.706)	(0.951)	(0.558)	(1.015)
-	100	15 pctl	103.8	0.992	30.948	46.247	2.571	6.499	6.352	11.633	2.353
				(0.007)	(2.014)	(3.925)	(0.372)	(0.459)	(0.702)	(0.344)	(1.015)
-	100	25 pctl	106.3	0.997	22.739	34.036	1.343	4.076	3.734	8.205	2.326
				(0.006)	(1.679)	(2.784)	(0.126)	(0.152)	(0.342)	(0.143)	(1.002)
-	100	mean	113.2	1.004	24.736	22.567	1.816	5.245	2.328	5.998	2.355
				(0.006)	(1.838)	(2.085)	(0.228)	(0.27)	(0.275)	(0.179)	(1.014)
$*$	$*$	-	100	0.992	21.976	17.475	1.130	3.588	1.401	3.945	2.140
				(0.005)	(1.883)	(1.089)	(0.078)	(0.107)	(0.125)	(0.11)	(0.92)

Table 3.10: Descriptive statistics of the simulation output when you are allowed to be $x = 60$ minutes ahead of schedule. The makespan is scaled with the planned makespan for that schedule. The percentage of the maximum speed and minimum (un)loading time used is given by p_s and p_t, respectively. The corresponding statistical value is given by v_s and v_t, respectively. Here $*$ corresponds to the case $v_s \in \{5 \text{ pctl}, 10 \text{ pctl}, 15 \text{ pctl}, 25 \text{ pctl}, \text{mean}\}$ and $p_s \in \{99.4, 98.7, 98.1, 96.8, 93.4\}$.

v_s	p_s(%)	v_t	p_t(%)	Makespan	Late_{SB}	Late_{LA}	Delay_{SB}	$\text{Delay}_{SB,l}$	Delay_{LA}	$\text{Delay}_{LA,l}$	Wave
-	100	-	100	0.987	24.238	21.357	1.674	4.516	2.494	5.150	2.334
				(0.007)	(2.65)	(4.202)	(0.556)	(0.895)	(1.109)	(1.177)	(1.007)
-	100	5 pctl	101.3	0.985	39.579	55.321	4.388	9.641	9.692	15.951	2.346
				(0.01)	(2.742)	(4.671)	(0.793)	(0.958)	(1.249)	(0.782)	(1.012)
-	100	10 pctl	102.5	0.987	35.467	50.719	3.353	7.894	7.827	13.615	2.354
				(0.01)	(2.295)	(4.267)	(0.568)	(0.706)	(0.951)	(0.557)	(1.016)
-	100	15 pctl	103.8	0.989	30.948	46.229	2.571	6.499	6.351	11.636	2.36
				(0.009)	(2.014)	(3.924)	(0.372)	(0.459)	(0.702)	(0.344)	(1.018)
-	100	25 pctl	106.3	0.993	22.739	34.027	1.343	4.076	3.733	8.206	2.376
				(0.008)	(1.679)	(2.782)	(0.126)	(0.152)	(0.342)	(0.143)	(1.026)
-	100	mean	113.2	1.000	24.735	22.566	1.816	5.245	2.327	5.998	2.427
				(0.009)	(1.838)	(2.085)	(0.228)	(0.27)	(0.275)	(0.179)	(1.047)
$*$	$*$	-	100	0.987	21.975	17.474	1.130	3.588	1.401	3.945	2.334
				(0.007)	(1.883)	(1.089)	(0.078)	(0.107)	(0.125)	(0.11)	(1.007)

Chapter 4

Policies for the Generalised Resupply Problem

An earlier version of this chapter appeared as Wagenvoort et al. (2023).

4.1 Introduction

Timely replenishment of essential commodities, such as food, fuel and medical supplies, is a critical component of military operations. It is vital that the stock of the commodities never drops below a critical level for a sustained period of time. Traditionally, large units stay together, requiring the delivery of commodities to a few locations to sustain these units above the critical level. These resupply operations are typically performed using motorised (off-terrain) vehicles. Recent advances in drone technology make fast resupply at remote locations increasingly possible, and are particularly useful to deliver medical supplies and small spare parts.

Developments in military strategy favour operating in smaller, dispersed units (Dictorate of Land Concepts and Design, 2007). In these operations, numerous small units at different locations must be sustained. This shift in the operating style poses significant challenges on the replenishment of the units. An interesting tactical question is how many vehicles are needed to sustain all stocks of the commodities above the critical level. Although motivated by a military context, this is also an interesting question in other applications in which locations require a periodic resupply.

We consider a periodic resupply problem for a single commodity. In this problem, which we call the Generalised Capacitated Resupply Problem (GCRP), we have a set of locations with associated capacities, deterministic demand rates, and the required number of time periods to perform a resupply operation at this location. The goal is to determine whether the stock levels can be indefinitely sustained above a critical threshold by a set of vehicles that have an associated payload. The locations can either be homogeneous or heterogeneous in their parameters and each vehicle may or may not have a sufficient payload to fully restock locations to maximum capacity. Hence, upon a visit of a vehicle, a location may be only partially restocked. We consider, in addition to the GCRP, three variants where the locations are homogeneous in terms of their capacity and demand rate and/or their resupply time.

The GCRP is a generalisation of the Windows Scheduling Problem, which involves assigning a set of jobs, each with a specific processing time and a window, to a minimum number of machines (Bar-Noy et al., 2012). The window of a job represents the maximum number of time periods between two executions of a job, *i.e.*, each job must be scheduled at least once in any set consecutive time periods with a length equal to the window of that job. So, independent of the time period in which we execute a job, the jobs has to be scheduled again within its window. This corresponds to the case of GCRP in which the payload of a vehicle is larger or equal to the capacity of a location, *i.e.*, after a visit, the location is always fully restocked.

The Windows Scheduling Problem is studied with either homogeneous or heterogeneous job lengths. For instances with homogeneous job lengths, a 2-approximation exists (Bar-Noy and Ladner, 2003). For instances with heterogeneous job lengths, but with homogeneous windows, the problem can be reduced to a Bin Packing Problem. This results in a solution with at most $(1 + \epsilon)OPT + 1$ machines, using a polynomial-time approximation scheme with input parameter ϵ and runtime $O(n \log(1/\epsilon)) + O_\epsilon(1)$, where $O_\epsilon(1)$ is a function that only depends on $1/\epsilon$ (Fernandez de La Vega and Lueker, 1981). If job lengths are heterogeneous, the problem is sometimes referred to as the Generalised Windows Scheduling Problem. This case can be reduced to the case with homogeneous windows. This results in a schedule requiring at most $2(1 + \epsilon)OPT + \log w_{max}$ machines (Bar-Noy et al., 2012), where w_{max} is the maximum window. Bar-Noy et al. also present an 8-approximation for the problem with heterogeneous inputs.

Other related problems are the Pinwheel Scheduling Problem (Holte et al., 1989) and the Periodic Latency Problem (Coene et al., 2011). In the Pinwheel Scheduling Problem, a set of jobs with an associated period have to be scheduled. In each time period, we can schedule one job. The question is whether there exists a schedule such that each job is executed at least once in any set of consecutive time periods with a length equal to the period of that job. Note that this problem is concerned about feasibility only.

The continuous version of the Windows Scheduling Problem, in which time continuously passes while travelling, is known as the Periodic Latency Problem. In this problem, a set of clients with associated windows and travel times are given. Coene et al. consider two variants. In the first, the goal is to find one perpetual route that maximises the total collected profit. In the second, the goal is to minimise the number of vehicles needed to serve all clients. The decision version of these problems are known to be PSPACE-complete (Ho and Ouaknine, 2015), whereas the complexity of the Windows Scheduling Problem and the Pinwheel Scheduling Problem is open.

From our problem's perspective, the above mentioned problems all implicitly assume that each time a visit to a location is scheduled, its inventory is restocked to its full capacity. In our problem, it is possible that a vehicle's payload is not sufficient to restock a location to its full capacity, introducing interplay between payload and capacity. Also, the UAV Resupply Scheduling Problem considered in Arribas et al. (2023) has some similarities with our model, but again locations are restocked (recharged) to their full capacity at each visit.

In this chapter, we formally introduce the GCRP and variants of the problem based on the homogeneity/heterogeneity of the locations in terms of the capacity and demand and/or the resupply time. We provide complexity results for the GCRP and two of its variants. We then provide policies with an approximation guarantee for the different variants of the problem and describe the complexity of finding the required number of vehicles.

The problem description and computational complexity analysis can be found in Section 4.2. The policies for the different variants can be found in Section 4.3 - 4.6. We end with a conclusion in Section 4.7.

4.2 Problem Description and Complexity

In this section, we formally introduce the Generalised Capacitated Resupply Problem (GCRP) and the variants of this problem considered in this paper in Section 4.2.1. Thereafter, we discuss the complexity of these problems in Section 4.2.2.

4.2.1 Problem Description

We consider a single commodity problem as there is often a commodity that is most important in terms of size and weight. We consider discrete time periods, and at the start of the first time period, each location starts with an initial stock at maximum capacity. Vehicles with a given payload resupply locations by making round trips from the depot to the locations. If a location is resupplied, the payload from the vehicle is added to the stock at the location up to its capacity. At the start of a time period, before a potential resupply takes place, consumption decreases the stock level of locations by the demand rate. If the stock drops below zero after consumption, we call this a *stock-out*, a situation we want to avoid. We formally define the decision variant of our problem as follows:

Generalised Capacitated Resupply Problem (GCRP)
Instance: A set of $n \in \mathbb{N}$ locations $N = \{1, \ldots, n\}$ to be supplied, $m \in \mathbb{N}$ vehicles available for resupply, vectors $c \in \mathbb{N}^n$, $r \in \mathbb{N}^n$, and $t \in \mathbb{N}^n$ with the maximum capacity, the deterministic demand rates, and the number of required time periods to resupply per location, respectively, and $p \in \mathbb{N}$ the maximum payload of the vehicles.
Question: Is it possible to perform resupply of the locations with the given number of vehicles, such that there is never a time period during which a stock-out occurs and where locations can only be resupplied up to their capacities?

We are interested in policies and their approximation guarantee for the optimisation variant of this problem, *i.e.*, we seek the minimum number of vehicles m such that no stock-out occurs. We denote the minimum number of required vehicles by m^* and the number of vehicles required for Policy A by $m(A)$. We assume without loss of generality that $r_i < p$, *i.e.*, the vehicle payload is greater than the demand at a location. If the demand does exceed the payload, we can assign dedicated vehicles to these locations and derive an instance with $m' = m - \sum_{i \in N'} t_i \left\lfloor \frac{r_i}{p} \right\rfloor$, $c'_i = c_i - p \left\lfloor \frac{r_i}{p} \right\rfloor$, and $r'_i = r_i - p \left\lfloor \frac{r_i}{p} \right\rfloor$ for all $i \in N'$, where $N' = \{i \mid r_i \geq p\}$. This instance has $r'_i < p$ for all $i \in N'$.

In the long-run, it is not necessary to make an assumption about the proportion of the resupply time required for the onward and return journey, respectively. However, this is important for the start-up period. We assume that the resupply operation is executed at the end of the resupply time. Therefore, for feasibility, it is necessary to assume that $t_i \leq \frac{c_i}{r_i}$ for all $i \in N$. Furthermore, we assume without loss of generality that $t_i \leq \frac{p}{r_i}$ for all $i \in N$. When $t_i > \frac{p}{r_i}$, we know that in any t_i consecutive time periods location i has a demand of more than p. Hence, to be feasible in the long-run, we need to visit location i at least once in any consecutive t_i time periods. We can thus assign dedicated vehicles, where each vehicle causes a decrease in the demand rate of $\frac{p}{t_i}$.

In this problem, locations are defined by their capacity, demand rate, and the number of time periods it requires to be resupplied, which potentially consists of travel time and (un)loading time. We consider three variants of this problem where the locations are homogeneous in terms of the capacity and demand rate and/or the number of required time periods to perform the resupply operation. If input parameters are homogeneous, we drop their corresponding indices. We define the Capacitated Resupply Problem (CRP) as the version of the GCRP where the resupply times are homogeneous, *i.e.*, $t_i = t$ for all $i \in N$. The Generalised Homogeneous Capacitated Resupply Problem (GHCRP) assumes homogeneous capacities and demand rates compared to the GCRP, *i.e.*, $c_i = c$ and $r_i = r$ for all $i \in N$. Finally, the Homogeneous Capacitated Resupply Problem (HCRP) assumes that all input parameters are homogeneous. An overview of these variants and their corresponding sections is given in Figure 4.1.

Finally, whenever capacities and demand rates are homogeneous (GHCRP and HCRP), we assume that $p \leq c$, since vehicles can only restock locations up to their capacity. Whenever the resupply times are homogeneous (CRP and HCRP), we assume these are unit length. If an instance has $t > 1$ for all $i \in N$, we can derive

	$t_i = t$	Arbitrary t_i
Arbitrary c_i and r_i	CRP Section 4.5	GCRP Section 4.6
$c_i = c$ $r_i = r$	HCRP Section 4.3	GHCRP Section 4.4

Figure 4.1: Overview of the variants of the Generalised Capacitated Resupply Problem and their corresponding sections.

an instance with $t' = 1$ and $r'_i = r_i t$ for all $i \in N$. Since all parameters are integer, r'_i is integer too and the instance is valid.

4.2.2 Complexity

In this section, we present complexity results for (variants of) the GCRP and give inapproximability results, *i.e.*, we show that there is likely no polynomial-time approximation algorithm with an approximation ratio below a certain threshold. We first show that the GHCRP is NP-hard by a reduction from the Bin Packing Problem. Then, we show that the CRP is intractable by a reduction from the Pinwheel Scheduling Problem. Finally, we show that the GCRP is strongly NP-complete by a reduction from 3-Partition.

In the reductions, we apply the sufficient condition outlined in Lemma 4.1, which relies on the fact that the timing of the visits matters if the capacity is restrictive, *i.e.*, $c_i < r_i p$ for $i \in N$.

Lemma 4.1. *To avoid stock-outs, it is necessary to visit location $i \in N$ at least once every $\frac{c_i}{r_i}$ time periods and sufficient to visit each location at least once every $\frac{p}{r_i}$ time periods. Hence, if $c_i = p$, it is both necessary and sufficient to visit location i once every $\frac{c_i}{r_i} = \frac{p}{r_i}$ time periods.*

Proof. After a visit to location i, its stock level is at least equal to p and at most equal to c_i. This stock lasts at least $\frac{p}{r_i}$ and at most $\frac{c_i}{r_i}$ time periods. Therefore, a visit once every $\frac{c_i}{r_i}$ time periods is necessary to avoid a stock-out. If $p \leq c_i$, a visit at least once every $\frac{p}{r_i}$ ensures that no stock-out occurs. Hence, if $c_i = p$, it is both necessary and sufficient to have a visit once every $\frac{c_i}{r_i} = \frac{p}{r_i}$ time periods. ■

To show that the GHCRP is NP-hard, we use a reduction from the Bin Packing Problem, which is strongly NP-complete (Garey and Johnson, 1979). In this problem we have n items with sizes $s_1, \dots, s_n$ and bins of size B. The question is whether we can fit these items in m bins.

Theorem 4.1. *The GHCRP is strongly NP-hard.*

Proof. Given an instance of the Bin Packing Problem, *i.e.*, n items with sizes $s_1, \dots, s_n$ and bins of size B, create an instance of the GHCRP as follows. Create a location for each item, where location i has $t_i = s_i$, $r = 1$ and $c = B$. Let the payload of the vehicles be equal to $p = B$. Now, we can verify there exists a feasible resupply schedule for this instance of the GHCRP on m vehicles if and only if there is a feasible solution with m bins for the original instance of the Bin Packing Problem.

If there is a feasible solution for the original instance of the Bin Packing Problem, then all items fit into m bins of size B. The locations corresponding to the items in a bin can be executed by a vehicle with a recurring schedule of B time periods. Since $\frac{p}{r} = B$, no stock-outs occur by Lemma 4.1. If there is a feasible schedule for GHCRP, then each location is visited at least once during the first B periods as $\frac{c}{r} = B$. Therefore, for each vehicle a bin can be created with the items corresponding to the locations visited within the first B periods of the schedule of the vehicle, removing any duplicates arbitrarily. Hence, there exists a feasible schedule on m vehicles for the GHCRP if and only if there exists a feasible schedule with m bins for the Bin Packing Problem. ■

For the Bin Packing Problem, it is NP-hard to determine whether it is possible to use two bins as opposed to three. This implies that for any $\alpha < \frac{3}{2}$, there is no polynomial-time α-approximation for the GHCRP, unless P=NP (Vazirani, 2001).

Now, we show that the CRP is intractable by a reduction from the Pinwheel Scheduling Problem. Recall that in this problem, we have n jobs with periods $w_1, \dots, w_n$. In each time period, we can schedule one job. The question is whether we can construct a perpetual schedule such that job i is scheduled in any period of w_i consecutive time periods. It is not known whether the Pinwheel Scheduling Problem is NP-complete, or even contained in NP. It was shown by Jacobs and Longo, 2014 that there cannot be a polynomial time exact algorithm for the Pinwheel Scheduling Problem, unless Satisfiability can be solved in expected time $O(n^{\log n \log \log n})$, which is deemed unlikely.

Theorem 4.2. *There is no polynomial-time exact algorithm for the CRP, unless the Satisfiability Problem can be solved in expected time* $O(n^{\log n \log\log n})$.

Proof. Given an instance of the Pinwheel Scheduling Problem, *i.e.*, $w_1, \dots, w_n$, create an instance of the CRP as follows. Create a location for each job, where location i has $r_i = (1/w_i) \prod_{j=1}^{n} w_j$, and $c_i = c = \prod_{j=1}^{n} w_j$. Furthermore, we set $p = \prod_{j=1}^{n} w_j$, and $m = 1$. Now, we can verify there exists a feasible resupply schedule for this instance of the CRP if and only if there is a feasible schedule for the original instance of the Pinwheel Scheduling Problem.

If there is a feasible solution to the original instance of the Pinwheel Scheduling Problem, then all jobs fit onto one machine and therefore job i is executed at least once per w_i periods, for all $i = 1, \dots, n$. For a solution to the CRP, it is both necessary and sufficient to schedule a visit to location i once every $\frac{p}{r_i}$ periods, for all $i = 1, \dots, n$, as $c_i = p$ (Lemma 4.1). Since $\frac{p}{r_i} = w_i$, it thus also holds that a solution is feasible if each location is visited at least once every w_i periods. Therefore, there exists a feasible schedule for the created instance of the CRP if and only if there exists a feasible schedule for the original instance of the Pinwheel Scheduling Problem. ■

The proof above shows it is hard to distinguish instances of the CRP where one vehicle is sufficient, and instances for which at least two vehicles are necessary. This implies that for any $\alpha < 2$, there is no polynomial-time α-approximation for minimising the number of vehicles in the CRP, unless we can solve Satisfiability in expected time $O(n^{\log n \log\log n})$. Using a similar reduction, one can show that the same holds for the Windows Scheduling Problem with unit job lengths. For the Windows Scheduling Problem with unit job lengths, a 2-approximation is known (Bar-Noy and Ladner, 2003).

Next, we show that the GCRP is strongly NP-hard by a reduction from 3-Partition, which is known to be strongly NP-complete (Garey and Johnson, 1979). In this problem, we have $3n$ numbers, $a_1, \dots, a_{3n}$, for which $B/4 < a_i < B/2$ with $B = \frac{1}{n} \sum_{i=1}^{3n} a_i$. The question is whether we can partition these numbers in n disjoint subsets such that each subset contains exactly 3 numbers and the sum of the numbers in each subset is equal to B, *i.e.*, whether there exist $S_1, \dots, S_n \subset \{1, \dots, 3n\}$ such that $S_j \cap S_{j'} = \emptyset$, $|S_j| = 3$ and $\sum_{i \in S_j} a_i = B$ for each j, j'. We show that GCRP is strongly NP-hard, even for the case with only one vehicle and homogeneous capacities. The same holds for homogeneous demand rates, as these can be scaled in the reduction. A similar reduction to Windows Scheduling on one machine was used by Bar-Noy and Ladner (2003).

Theorem 4.3. *The GCRP is strongly NP-hard, even on one vehicle and with homogeneous capacities.*

Proof. Given an instance of 3-Partition, *i.e.*, $3n$ numbers, $a_1, \ldots, a_{3n}$, for which $B/4 < a_i < B/2$ with $B = \frac{1}{n}\sum_{i=1}^{3n} a_i$ for all $i \in \{1, \ldots, 3n\}$, create an instance of GCRP as follows. Create a location for each number, where location i has $t_i = a_i$, $r_i = 1$, and $c_i = n(B+1)$. Create another location with $t_{3n+1} = 1, r_{3n+1} = n$, and $c_{3n+1} = n(B+1)$. Let the payload of the vehicles be equal to $p = n(B+1)$. Now, we can verify there exists a feasible resupply schedule for the instance of GCRP on one vehicle if and only if we can partition the numbers of the original instance of 3-Partition in subsets of size 3, *i.e.*, if there exist subsets $S_1, \ldots, S_n \subseteq \{1, \ldots, 3n\}$ with $|S_j| = 3$ for all j, $S_j \cap S_{j'} = \emptyset$ for all j, j', and $\sum_{i \in S_j} a_i = B$ for all j.

If there is a 3-Partition of the original instance, then we can construct a resupply schedule on one vehicle following a periodic schedule of length $n(B+1)$ as follows. Schedule a visit to location $3n+1$ during time periods $B+1, 2(B+1), \ldots, n(B+1)$. We can schedule visits to the locations corresponding to the numbers in subset S_j between the $(j-1)$th and jth visit to location $3n+1$, since these take exactly B time periods to visit. Since location $3n+1$ is visited every $\frac{c_{3n+1}}{r_{3n+1}} = \frac{p_{3n+1}}{r_{3n+1}} = B+1$ time periods, and each other location is visited every $\frac{c_i}{r_i} = \frac{p_i}{r_i} = n(B+1)$ time periods, this schedule is feasible by Lemma 4.1.

If there is a feasible resupply schedule on one vehicle, then locations $1, \ldots, 3n$ must be visited at least once during the first $n(B+1)$ time periods as $\frac{c_i}{r_i} = n(B+1)$ for all $i \in \{1, \ldots, 3n\}$. Location $3n+1$ must be visited at least once every $B+1$ time periods since $\frac{c_{3n+1}}{r_{3n+1}} = B+1$. During the first $n(B+1)$ periods, the minimum required number of time periods to avoid a stock-out is thus equal to $\sum_{i=1}^{3n} t_i + n \cdot t_{3n+1}$. Since $\sum_{i=1}^{3n} t_i + n \cdot t_{3n+1} = \sum_{i=1}^{3n} a_i + n = n(B+1)$, these lower bounds on the number of visits during the first $n(B+1)$ time periods are tight. Location $3n+1$ must thus be visited exactly n times during the first $n(B+1)$ time periods. Since $\frac{c_i}{r_i} = \frac{p_i}{r_i} = B+1$, these visits should be evenly spread such that location $3n+1$ is visited exactly once every $B+1$ time periods. Let S_j be the set of locations visited between the $(j-1)$th and jth visit to location $3n+1$ for $j \in \{1, \ldots, n\}$. Since each location is visited exactly once during the first $n(B+1)$ time periods and all B time periods should be used, $S_j \cap S_{j'} = \emptyset$ for all j, j' and $\sum_{j \in S_j} t_i = \sum_{j \in S_j} a_i = B$. Furthermore, $|S_j| = 3$ must hold since $B/4 < t_i < B/2$. $S_1, \ldots, S_n$ thus form a 3-Partition of the original instance. ■

The proof above shows it is hard to distinguish instances of GCRP where one vehicle is sufficient, and instances for which at least two vehicles are necessary. This implies that for any $\alpha < 2$, there is no polynomial-time α-approximation algorithm for minimising the number of vehicles in the GCRP, unless P = NP.

4.3 Homogeneous Capacitated Resupply Problem

In this section, we consider the HCRP, the variant of the GCRP in which all input parameters are homogeneous. Specifically, we have n locations each with a capacity of c and a demand rate of r that can be resupplied within one time period by vehicles with a payload of p. We describe some simple analytical policies from which the required number of vehicles immediately follows in Section 4.3.1. In Section 4.3.2, we present an optimal policy.

We observe that for the HCRP, the total supply per time period (at most mp) should exceed the total demand per time period (nr), otherwise the stocks will keep decreasing in the long-run. It follows that $m^* \geq \left\lceil \frac{nr}{p} \right\rceil$.

4.3.1 Simple Analytical Policies

The first policy is based on the fact that when the capacity is large enough, the precise timing of the visits does not matter, as long as each region is visited on average once every $\frac{p}{r}$ periods.

Policy 4.1 (Wrap Around (WA) Policy)**.** *Define the schedule length as p. Initially, let the first vehicle be the active vehicle. In each step, schedule the next location for the next r time periods on the active vehicle. If the schedule of the active vehicle is full, i.e., all p time periods are allocated, continue on the next vehicle, which becomes the active vehicle.*

Mathematically, during time period τ, vehicle j will visit location

$$1 + \left\lfloor \frac{(\tau - 1 \bmod p) + (j-1)p}{r} \right\rfloor .$$

For example, if $r = 3$ and $p = 5$, vehicle 1 visits location 1 in the first three time periods and location 2 in time periods 4 and 5, whereas vehicle 2 visits location 2 in the first time period, location 3 in time periods 2, 3, and 4, and location 4 in time period 5, and so on.

Theorem 4.4. *Policy 4.1 is optimal with $m(WA) = \left\lceil \frac{nr}{p} \right\rceil$ vehicles if $\frac{c}{r} \geq p$.*

Proof. Consider the first time period of a location where the stock starts decreasing below the maximum stock level c. Since it takes at least $\frac{c}{r} \geq p$ time periods for a stock-out to occur, and there are fewer than p time periods until the next visit since the schedule length is p, no stock-out can occur. The total decrease in stock level is at most $(p-r+1)r$. Then, the location is visited in the next r time periods where the stock level can increase by $rp-(r-1)r$. After these visits the location is at full capacity again. We establish that no location can run out of stock. ■

If the capacity is not large enough and the timing of the visits matters ($c < rp$), then Policy 4.1 can result in an infeasible schedule. In that case, we can use the sufficient condition in Lemma 4.1 to define two different policies. The first policy is the No Migration Policy in which we assign locations to a vehicle such that each vehicle executes a recurring schedule visiting a subset of the locations. The second policy is the Shift Policy, in which vehicles execute a schedule visiting all locations, but the schedules of the vehicles are shifted compared to each other.

Policy 4.2 (No Migration (NM) Policy)**.** *Divide the locations in the minimum number of groups of size at most $\left\lfloor \frac{p}{r} \right\rfloor$. Then, each vehicle visits the locations in one group in a recurring schedule.*

Mathematically, during time period τ, vehicle j will visit location

$$1 + (j-1)\left\lfloor \frac{p}{r} \right\rfloor + \left((\tau - 1) \bmod \left\lfloor \frac{p}{r} \right\rfloor\right).$$

For example, the first vehicle visits locations $1, \ldots, \left\lfloor \frac{p}{r} \right\rfloor$, vehicle two visits locations $\left\lfloor \frac{p}{r} \right\rfloor + 1, \ldots, 2\left\lfloor \frac{p}{r} \right\rfloor$, and so on.

Policy 4.3 (Shift (SH) Policy)**.** *Let each vehicle visit the same sequence of locations $1, \ldots, n$, but vary the starting point of each vehicle such that location 1 is visited $\left\lfloor \frac{p}{r} \right\rfloor$ time periods later each time.*

Mathematically, during time period τ, vehicle j will visit location

$$1 + \left(\tau - 1 + (j-1)\left\lfloor \frac{p}{r} \right\rfloor\right) \bmod n.$$

For example, if $n = 10$, $p = 9$ and $r = 3$, vehicle 1 starts at location 1, vehicle 2 starts at location 8, vehicle 3 starts at location 5, and vehicle 4 starts at location 2. Then, location 1 is visited in time period 1, 4, 7, and 10.

Theorem 4.5. *Policies 4.2 and 4.3 are 2-approximations with $m(NM) = m(SH) = \left\lceil \frac{n}{\left\lfloor \frac{p}{r} \right\rfloor} \right\rceil$ vehicles, and optimal if $\frac{p}{r} \in \mathbb{N}$.*

Proof. From Lemma 4.1, it follows that no stock-out occurs with Policies 4.2 and 4.3. In Policy 4.2 each vehicle has a periodic schedule of length $\left\lfloor \frac{p}{r} \right\rfloor$ and in Policy 4.3 each vehicle operates with a shift of length $\lfloor \frac{p}{r} \rfloor$ compared to each other. Therefore, these policies result in a solution with $\left\lceil \frac{n}{\lfloor \frac{p}{r} \rfloor} \right\rceil$ vehicles. Using that $\lfloor x \rfloor \geq x/2$ for $x \geq 1$, and $\lceil nx \rceil \leq n \lceil x \rceil$ for $n \in \mathbb{N}$, we get an approximation guarantee of

$$m(\mathrm{NM}) = m(\mathrm{SH}) = \left\lceil \frac{n}{\lfloor \frac{p}{r} \rfloor} \right\rceil \leq \left\lceil \frac{n}{\frac{1}{2}\frac{p}{r}} \right\rceil \leq 2 \left\lceil \frac{nr}{p} \right\rceil \leq 2m^*$$

If $\frac{p}{r} \in \mathbb{N}$, $\lfloor \frac{p}{r} \rfloor = \frac{p}{r}$ and therefore $m(\mathrm{NM}) = m(\mathrm{SH}) = \left\lceil \frac{n}{\lfloor \frac{p}{r} \rfloor} \right\rceil = \left\lceil \frac{n}{\frac{p}{r}} \right\rceil = m^*$. ■

The next example shows that the analysis for both Policy 4.2 and Policy 4.3 is tight.

Example 4.1. *Consider an instance with $n = 10$, $r = 10$, $p = 19$, and $c = 30$. Then, Policy 4.2 and 4.3 both result in a schedule with $m = \left\lceil \frac{n}{\lfloor p/r \rfloor} \right\rceil = 10$ vehicles. However, the optimal number of vehicles is equal to 6 with the following schedule. Apply the idea of Policy 4.3 to 5 vehicles with a shift of $\lceil p/r \rceil = 2$. Then, add a sixth vehicle with a schedule that is out of sync with the other schedules. This implies that both policies are a factor $\frac{5}{3}$ off. In general, we can set $r = n$, $p = 2n - 1$, and $c = 3n$, with n even. Then, Policy 4.2 and 4.3 use n vehicles, whereas there is an optimal schedule that uses $n/2 + 1$ vehicles.*

For all policies presented in this section, the number of required vehicles can be found by using elementary operations.

4.3.2 A Greedy Policy

Up until now we considered policies for which the required number of vehicles can be easily derived. Alternatively, we can consider a policy that assigns vehicles based on the current stock level. We consider a greedy policy that aims to maximally postpone stock-outs.

Policy 4.4 (Greedy (G) Policy). *Assign the m locations with the lowest current stock to each of the m vehicles.*

Mathematically, during time period τ, vehicle j will visit location

$$(j - 1 + (\tau - 1)m) \mod n + 1.$$

Theorem 4.6. *Policy 4.4 is optimal, i.e., $m(G) = m^*$.*

Proof. Given the current stock level x_i^τ of a location i at time period τ, and assuming no future restocks occur, the time until a stock-out occurs can be expressed as $\frac{x_i^\tau}{r}$. In order to detect a stock-out, we only need to be concerned with the lowest stock level $x_{\min}^\tau = \min_{i=1\ldots n} x_i^\tau$.

First, we argue that the total stock level over all locations increases by the maximum amount possible with the greedy policy. Consider another policy that prefers restocking a location i which has a higher stock level than a location j that is not restocked by that policy. We thus have $x_i^\tau > x_j^\tau$. It is clear to see that the maximum amount that can be restocked at location i is at most the amount that can be restocked at location j, as $c - x_i^\tau < c - x_j^\tau$.

Second, we argue that the greedy policy *maximally postpones* stock-outs. Consider another policy that at time period τ prefers restocking a location i which has a higher stock level than a location j that is not restocked by that policy. In case $x_j^{\tau+1} > x_{\min}^{\tau+1}$, location j is not critical and $x_{\min}^{\tau+1}$ will be equal for both policies. In case $x_j^{\tau+1} = x_{\min}^{\tau+1}$, the $x_{\min}^{\tau+1}$ of the greedy policy will be greater than or equal to that value of the other policy.

We conclude that among all policies, the greedy policy maximally postpones stock-outs as it maximally increases the total stock at the most critical locations. As locations are otherwise identical, the greedy policy can avoid stock-outs with m^* vehicles. ■

If n/m is integer, the greedy schedule visits each location exactly once in each period of n/m consecutive time periods. It is easy to check if no stock-out can occur. If n/m is not integer, the time between two visits to a location in the greedy schedule is either $\lfloor n/m \rfloor$ or $\lceil n/m \rceil$. Observe that for the HCPR, the greedy policy coincides with a round-robin policy, where a cyclical pattern follows. For each location this results in a schedule with n_s periods of $\lfloor n/m \rfloor$ time periods and n_l periods of $\lfloor n/m \rfloor +$ 1 time periods. To check whether this schedule is feasible, Theorem 4.7 can be used.

Theorem 4.7. *If $n/m \notin \mathbb{N}$, the following statements hold.*

- *If $\lceil n/m \rceil \leq \lfloor p/r \rfloor$, then the greedy schedule is feasible.*
- *If $\lceil n/m \rceil \geq \lfloor p/r \rfloor + 2$, then the greedy schedule is infeasible.*
- *Else, the greedy schedule is feasible if and only if*

$$\min\{c,\ (n_s+1)p - n_s\lfloor p/r \rfloor r\} \geq n_l\left(\lfloor p/r \rfloor + 1\right)r - (n_l - 1)p. \tag{4.1}$$

Proof. If $\lceil n/m \rceil \leq \lfloor p/r \rfloor$, then every location is visited at least once every $\lfloor p/r \rfloor$ time periods. Hence, by Lemma 4.1, the schedule is feasible.

If $\lceil n/m \rceil \geq \lfloor p/r \rfloor + 2$, then the number of time periods between any two visits to a location is at least $\lfloor p/r \rfloor + 1$ which therefore consumes more than p between any two consecutive visits. Thus, the stock will eventually drop below zero.

Else, the refill after n_s periods of length $\lfloor p/r \rfloor$ time periods should be at least the decrease in n_l periods of length $\lfloor p/r \rfloor + 1$ time periods. If this inequality holds, the greedy schedule is feasible. It is also necessary, since a violation will lead to a stock-out. ■

By using a binary search, we can solve the optimisation variant by solving the decision variant $O(\log n)$ times. For this, Theorem 4.7 can be used which takes at most $O(n)$ time as the schedule produced by the Greedy Policy has a period of at most n time periods. More precisely, when $\lceil n/m \rceil = \lfloor p/r \rfloor + 1$, we can find n_s and n_l from the periodic schedule with length at most $\mathrm{lcm}(n, m)/m$. Hence, we can find m^* for HCRP in $O(n \log n)$ time.

4.4 Generalised Homogeneous Capacitated Resupply Problem

In this section, we consider the GHCRP, the variant of the GCRP where the demand rate and capacity are homogeneous. Namely, we have n locations each with a capacity of c and a demand rate of r. It takes t_i time periods to resupply location $i \in N$ by vehicles with a payload of p. First, in Section 4.4.1, we describe the effect of heterogeneous resupply times on the simple policies for the HCRP defined in Section 4.3. Second, in Section 4.4.2, we describe another policy that can be applied to the GHCRP by reducing the problem to a Bin Packing Problem.

4.4.1 HCRP Policies Applied to the GHCRP

In Section 4.3.1, we defined simple analytical policies for the HCRP. In the GHCRP, compared to the HCRP, we have heterogeneous resupply times. Therefore, the HCRP policies cannot be (directly) applied. In this section, we discuss the effect of heterogeneous resupply times for each of these policies.

First, we consider the Wrap Around Policy (Policy 4.1). This policy is based on the fact that, if the capacity is large enough, the exact timing of the visits does not matter. It is then sufficient to schedule at least r visits every p periods for each location. This policy is optimal for the HCRP if $c \geq rp$. When resupply times are heterogeneous, this is not the case, as the order of the visits matters due to the varying lengths of the visits we need to schedule, as illustrated in Example 4.2.

Example 4.2. *Consider an instance with* $n = 5$, $r = 2$, $p = 7$, $c = 20$, *and resupply times of 1, 1, 2, 3, and 3, respectively. From the Wrap Around Policy defined in Policy 4.1, we need to allocate two visits for each location on a periodic schedule of length* 7. *We start by scheduling visits to location* 1 *during time periods* 1 *and* 2 *on vehicle* 1. *Then, we schedule visits to location* 2 *during time periods* 3 *and* 4. *For location* 3, *for which it takes two time periods to resupply, we can schedule the first visit during time periods* 5 *and* 6. *However, we need to schedule the second visit on vehicle* 2, *leaving an empty time period on vehicle* 1. *Continuing like this, we find the schedule with* 4 *vehicles depicted in Figure 4.2.*

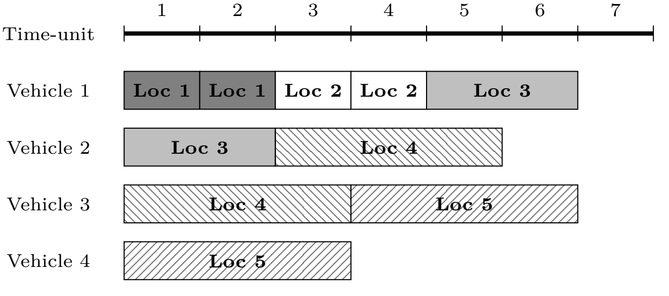


Figure 4.2: Illustration of the Wrap Around Policy for Example 4.2.

On the other hand, an optimal solution only requires 3 *vehicles, as shown by Figure 4.3.*

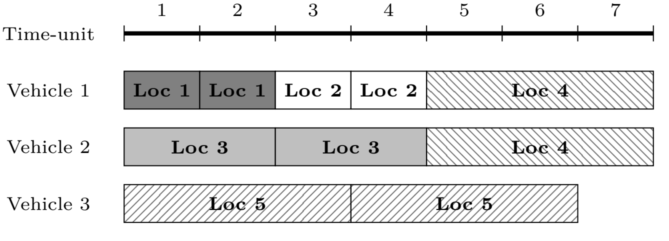


Figure 4.3: Optimal solution for Example 4.2.

Second, we consider the No Migration Policy (Policy 4.4). This policy is based on Lemma 4.1, which states that a schedule is feasible if a location is visited once every $\left\lfloor \frac{p}{r} \right\rfloor$ time periods. This policy results in a schedule that requires at most twice as many vehicles compared to the optimal solution, and is optimal if $\frac{p}{r} \in \mathbb{N}$. Similar to the Wrap Around Policy, the order in which the visits are scheduled matters when the resupply times are heterogeneous, as illustrated in Example 4.3.

Example 4.3. *Consider an instance with* $n = 5$*,* $r = 2$*,* $p = 7$*,* $c = 10$*, and resupply times of 1, 1, 2, 2, and 3, respectively. For the No Migration Policy defined in Policy 4.2, we need to allocate one visit for each location on a periodic schedule of length* $\lfloor \frac{p}{r} \rfloor$*. Here,* $\lfloor \frac{p}{r} \rfloor = 3$*. We start by scheduling a visit to location* 1 *on vehicle* 1*, which takes place in time period* 1*. Then, we can schedule a visit to location* 2 *during time period* 2*, but we are not able to schedule a visit to location* 3 *on vehicle* 1 *anymore, as this requires* 2 *time periods. This leaves an empty time period on vehicle* 1*. Continuing like this, we find the schedule given in Figure 4.4 on* 4 *vehicles.*

However, an optimal solution only requires 3 *vehicles, as shown in Figure 4.5.*

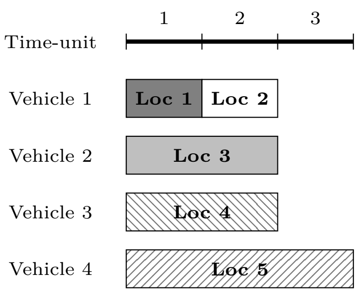


Figure 4.4: Illustration of the Wrap Around Policy for Example 4.3.

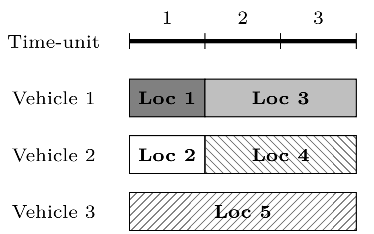


Figure 4.5: Optimal solution for Example 4.3.

For both the Wrap Around Policy and the No Migration Policy, the order in which the visits are scheduled thus matters when resupply times are heterogeneous. Using the Wrap Around Policy or the No Migration Policy can be seen as solving a Bin Packing Problem. For the No Migration Policy, this comes at the price of imposing a strict deadline of scheduling a visit at least once every $\lfloor \frac{p}{r} \rfloor$ consecutive time periods. The price of imposing a strict deadline of $\lfloor \frac{p}{r} \rfloor$ is discussed in Section 4.4.2.

Finally, we consider the Shift Policy (Policy 4.3). In this policy, which is also based on Lemma 4.1, vehicles execute the same schedule, but shifted relatively compared to each other. Due to the heterogeneous resupply times, this can result in issues in the start-up period as the times of the locations that are shifted might not add-up to the length of the shift, which equals $\lfloor \frac{p}{r} \rfloor$. This is illustrated in Example 4.4.

Example 4.4. *Consider an instance with $n = 5$, $r = 2$, $p = 7$, $c = 7$, and resupply times of 1, 1, 2, 2, and 3, respectively. From the Shift Policy defined in Policy 4.3, each vehicle visits the same sequence of locations, but with a shift of $\lfloor \frac{p}{r} \rfloor$ compared to each other. Here, $\lfloor \frac{p}{r} \rfloor = 3$. Vehicle 1 visits locations 1 up to 5 in order. Then, the schedule of vehicle 2 is shifted forward, i.e., it starts with location 5 such that location 1 is visited exactly 3 time periods later. However, when we shift the schedule for vehicle 3, vehicle 3 can only start in the second time period at the start of the operation, as can be seen in Figure 4.6. While in the long-run the schedule is feasible, a stock-out occurs during the start-up period as the capacity is depleted before the first visit to location 3 is completed.*

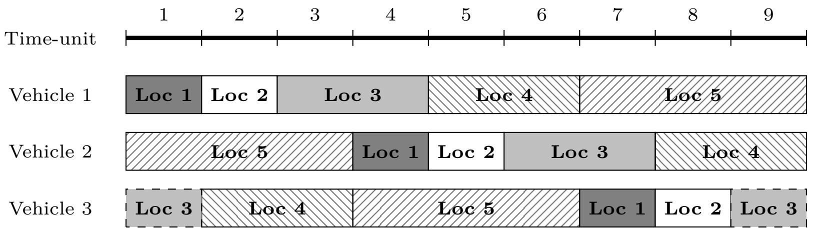


Figure 4.6: Illustration of the Shift Policy for Example 4.4.

For some instances, this issue could be resolved by altering the sequence of the locations. However, this is not always possible. In that case, an adjusted version of the policy can be used in which additional vehicles are added such that also in the start-up period locations are visited frequently enough.

Policy 4.5 (Adjusted Shift (ASH) Policy)**.** *Let each vehicle visit the same sequence of locations $1, \ldots, n$, but vary the starting point of each vehicle such that location 1 is visited $\lfloor \frac{p}{r} \rfloor$ time-units later each time. If $c < 2p$ and a shift does not align with the start of a new resupply operation, a new vehicle is added that executes the sequence starting with the misaligned job.*

Theorem 4.8. *Policy 4.5 is a 4-approximation with at most $m(ASH) \leq 2 \left\lceil \frac{\sum_{i \in N} t_i}{\lfloor \frac{p}{r} \rfloor} \right\rceil$ vehicles, a 2-approximation if $\frac{p}{r} \in \mathbb{N}$ or $c \geq 2p$, and optimal if both conditions hold.*

Proof. We first prove that, in general, the policy is a 4-approximation. Then, we discuss the conditions under which the approximation ratio is reduced by a factor of 2.

From Lemma 4.1, it follows that no stock-out occurs if each location is visited at least once per $\lfloor \frac{p}{r} \rfloor$ time periods. Whenever a shift of $\lfloor \frac{p}{r} \rfloor$ would violate this condition and the capacity is not sufficient to capture the start-up period, an additional vehicle is scheduled. Therefore, the Adjusted Shift Policy results in a feasible solution.

For the GHCRP, a lower bound on the number of required vehicles is equal to $\left\lceil \frac{\sum_{i \in N} rt_i}{p} \right\rceil$. In Policy 4.5, vehicles operate with a shift of length $\lfloor \frac{p}{r} \rfloor$ compared to each other and an additional vehicle is added whenever there is a misalignment in the shift. Therefore, the policy results in at most $2\left\lceil \frac{\sum_{i \in N} t_i}{\lfloor \frac{p}{r} \rfloor} \right\rceil$ vehicles. We thus get an approximation guarantee of

$$m(\text{ASH}) = 2\left\lceil \frac{\sum_{i \in N} t_i}{\lfloor \frac{p}{r} \rfloor} \right\rceil \leq 2\left\lceil \frac{\sum_{i \in N} t_i}{\frac{1}{2}\frac{p}{r}} \right\rceil \leq 4\left\lceil \frac{\sum_{i \in N} rt_i}{p} \right\rceil \leq 4m^*.$$

If $\frac{p}{r} \in \mathbb{N}$, $\lfloor \frac{p}{r} \rfloor = \frac{p}{r}$ and therefore $\left\lceil \frac{\sum_{i \in N} t_i}{\lfloor \frac{p}{r} \rfloor} \right\rceil = \left\lceil \frac{\sum_{i \in N} rt_i}{p} \right\rceil$, resulting in a 2-approximation. If $c \geq 2p$, no additional vehicles are required and hence $m(\text{ASH}) = m(\text{SH})$, resulting in a 2-approximation. When both conditions hold, the policy is thus optimal. ■

To determine the number of required vehicles, we have to construct the schedule consisting of n visits on at most n vehicles. Therefore, we can find $m(AHS)$ for GHCRP in $O(n^2)$ time, which is polynomial in the input size.

No additional vehicles are required if all resupply times are a power of 2. Here, we can construct a feasible schedule by arranging locations in ascending order of resupply times and using a shift of $\frac{p}{r}$ rounded down to the nearest power of 2. This leads to a feasible schedule, since the shift length is larger than or equal to all resupply times and, due to the order of the visits, no visit will be disrupted by applying this shift. Hence, the policy is then a 2-approximation and optimal if $\frac{p}{r}$ is also a power of 2. This is illustrated by Example 4.5.

Example 4.5. *Consider an instance with $n = 5$, $r = 2$, $p = 8$, $c = 8$, and resupply times of 1, 1, 1, 2, and 4, respectively. Note that all resupply times are a power of 2 and that $\frac{p}{r} = \frac{8}{2} = 4$ is also a power of 2.*

Then, we can apply the shift policy with a shift of $\frac{p}{r} = 4$ where the locations are visited in ascending order of resupply times. This gives the schedule given in Figure 4.7, which is optimal.

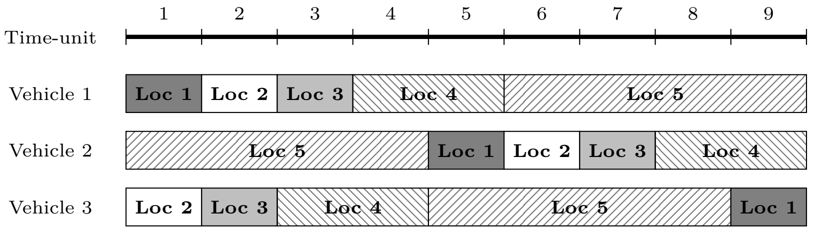


Figure 4.7: Illustration of the Shift Policy for Example 4.5.

4.4.2 Bin Packing Algorithm

An instance of the GHCRP can be solved by converting it to a Bin Packing Problem (Policy 4.6).

Policy 4.6 (Bin Packing (BP) Policy)**.** *Construct from an instance of GHCRP, $I = \{p, c, r, t_1, \ldots, t_n\}$, an instance of Bin Packing, $I^B = \left\{ \frac{t_1}{\lfloor \frac{p}{r} \rfloor}, \ldots, \frac{t_n}{\lfloor \frac{p}{r} \rfloor} \right\}$ with unit capacity bins. Let $S_1, \ldots, S_m$ be a solution to I^B, where S_i is the subset of items that are packed in bin i. Then, let vehicle $j \in \{1, \ldots, m\}$ execute a recurring scheme visiting the locations corresponding to the items in S_j in order.*

Policy 4.6 imposes the restriction that each location is visited exactly once per $\lfloor \frac{p}{r} \rfloor$ time periods, and therefore results in a feasible schedule by Lemma 4.1. By Theorem 4.9, when $c \leq 2p$, the price of imposing this strict deadline for the GHCRP is a factor 3.

Theorem 4.9. *When $c \leq 2p$, there exists a schedule using $3m^*$ vehicles for GHCRP such that each location is visited exactly once during any $\lfloor \frac{p}{r} \rfloor$ consecutive time periods.*

Proof. In an optimal solution, each location is visited at least once in the first $\lfloor \frac{c}{r} \rfloor \leq 2\lfloor \frac{p}{r} \rfloor$ time periods. Now, consider an arbitrary vehicle, and create three vehicles in the new schedule. The first vehicle repeatedly visits the locations that are (completely) visited by time period $\lfloor \frac{p}{r} \rfloor$ by the original vehicle. The second vehicle repeatedly visits the locations that are (completely) visited after time period $\lfloor \frac{p}{r} \rfloor$ until time period $2\lfloor \frac{p}{r} \rfloor$ by the original vehicle. The third vehicle repeatedly visits the location that is not scheduled yet, *i.e.*, the location whose visit partially takes place during the first $\lfloor \frac{p}{r} \rfloor$ time periods and partially during the second $\lfloor \frac{p}{r} \rfloor$ time periods in the schedule of the original vehicle, if it exists. Otherwise, it remains empty. Since we assume that $t_i \leq \frac{p}{r_i}$ for all $i \in N$, this vehicle also repeats its schedule after at most $\lfloor \frac{p}{r} \rfloor$ time periods.

If we perform this procedure for each vehicle in the optimal schedule, we obtain the desired structure using $3m^*$ vehicles. ■

Corollary 4.1. *For any $\epsilon > 0$, Policy 4.6 results in $m(BP) \leq 3(1+\epsilon)m^* + 1$ vehicles when $c \leq 2p$.*

Proof. By reducing the problem to a Bin Packing instance, a strict deadline of $\lfloor \frac{p}{r} \rfloor$ is imposed. This comes at the cost of a factor 3 by Theorem 4.9, *i.e.*, $OPT(I^B) \leq 3m^*$, where $OPT(I^B)$ denotes the optimal number of bins for instance I^B. From an instance of Bin Packing, a solution with $(1+\epsilon)OPT(I^B) + 1$ vehicles can be found in polynomial-time (Fernandez de La Vega and Lueker, 1981). Therefore, we get at most $(1+\epsilon)OPT(I^B) + 1 \leq 3(1+\epsilon)m^* + 1$ vehicles. ■

The Bin Packing algorithm of Fernandez de La Vega and Lueker (1981) is a polynomial-time approximation scheme with a runtime of $O(n \log 1/\epsilon) + O_\epsilon(1)$, where $O_\epsilon(1)$ is only dependent on $1/\epsilon$. Therefore, for a given $\epsilon > 0$, Policy 4.6 has a computational complexity of $O(n \log 1/\epsilon) + O_\epsilon(1)$.

If we combine Theorem 4.8 and Corollary 4.1, we observe that we have a 2-approximation for the GHCRP when $c \geq 2p$, and otherwise we can obtain a schedule using a number of vehicles of at most

$$\min\{4m^*, 3(1+\epsilon)m^* + 1\}.$$

4.5 Capacitated Resupply Problem

In this section, we consider the CRP, the variant of the GCRP where the resupply times are homogeneous. Recall that if resupply times are homogeneous, we can assume these are unit length as we can transform an instance with $t > 1$ to an instance with $t' = 1$. We thus consider an instance with n locations that take one time period to resupply by vehicles with a payload p. Location i has a capacity of c_i and a demand rate of r_i where $r_i < p$ as we can transform an instance with $r_i \geq p$ to an instance with $r'_i < p$. The input size of the CRP is dependent on n. Therefore, a policy is polynomial if the runtime is polynomial in n.

Each location should on average be visited at least once every $\frac{p}{r_i}$ periods to ensure the total supply is at least equal to the total demand. When the capacity is restrictive, $c_i \leq p$, we can tighten this bound to $\frac{c_i}{r_i}$. Observe that at least $\frac{r_i}{p}$ time should be spent on locations where the capacity is not restrictive, *i.e.*, on locations in $\mathcal{A} = \{i \in N \mid p < c_i\}$, and at least $\frac{r_i}{c_i}$ time should be spent on locations where the capacity is restrictive, *i.e.*, on locations in $\mathcal{B} = \{i \in N \mid p \geq c_i\}$. This results in a lower bound on the number of vehicles of $\left\lceil \sum_{i\in\mathcal{A}} \frac{r_i}{p} + \sum_{i\in\mathcal{B}} \frac{r_i}{c_i} \right\rceil$. By dividing the locations in these two groups, one group where the capacity is not restrictive and one group where it is, we can define a 2-approximation policy. This policy generalises the 2-approximation by Bar-Noy and Ladner. An example of this policy is given in Example 4.6.

Policy 4.7 (Power of 2 (Po2) Policy)**.** *Let $\mathcal{A} = \{i \in N \mid p < c_i\}$ and $\mathcal{B} = \{i \in N \mid p \geq c_i\}$. Let $d_i = \frac{p}{r_i}$ for $i \in \mathcal{A}$ and $d_i = \frac{c_i}{r_i}$ for $i \in \mathcal{B}$. Round each d_i down to the nearest power of 2 and denote this by d'_i. Use a periodic schedule of length $\max_{i=1,\dots,n} d'_i$ periods. Assign the demand locations to vehicles in non-decreasing order of d'_i, adding a new vehicle whenever all time slots of the previous vehicle are full. This results in a schedule with $m = \left\lceil \sum_{i\in N} \frac{1}{d'_i} \right\rceil$ vehicles.*

Theorem 4.10. *Policy 4.7 is a 2-approximation with $m(Po2) \leq 2\left\lceil \sum_{i\in\mathcal{A}} \frac{r_i}{p} + \sum_{i\in\mathcal{B}} \frac{r_i}{c_i} \right\rceil$ vehicles for $\mathcal{A} = \{i \in N \mid p < c_i\}$ and $\mathcal{B} = \{i \in N \mid p \geq c_i\}$.*

Proof. In Policy 4.7, the period of each location, d_i, is rounded down to a factor of 2 denoted by d'_i. Scheduling a visit every d'_i periods results in a feasible schedule by Lemma 4.1, as $d'_i \leq d_i = \min\left\{\frac{p}{r_i}, \frac{c_i}{r_i}\right\}$ periods. Locations are added to vehicles in non-decreasing order and all periods d'_i are factors of 2. Hence, following the above described algorithm gives a schedule with $m = \left\lceil \sum_{i\in N} \frac{1}{d'_i} \right\rceil$ vehicles. This implies that we have at least $m-1$ full vehicles and we cannot construct a schedule for the instance where locations $i \in N$ have periods d'_i using fewer vehicles.

We then find the following:

$$m(\text{Po2}) = \left\lceil \sum_{i\in N} \frac{1}{d'_i} \right\rceil \leq \left\lceil 2\sum_{i\in N} \frac{1}{d_i} \right\rceil \leq 2\left\lceil \sum_{i\in N} \frac{1}{d_i} \right\rceil = 2\left\lceil \sum_{i\in\mathcal{A}} \frac{r_i}{p} + \sum_{i\in\mathcal{B}} \frac{r_i}{c_i} \right\rceil \leq 2m^*.$$

Here we use that $\frac{1}{d'_i} \leq \frac{2}{d_i}$ as we round d_i down to the nearest factor of 2. Hence, the above described algorithm is a 2-approximation. ∎

Determining the number of required vehicles requires calculating d'_i for each location, which can be done in $O(n)$ time. This bound is tight by Example 4.6.

Example 4.6. *Consider an instance with $n = 5$ locations. The demand rates and capacities of the locations are given in Table 4.1. The vehicles have a payload of $p = 10$.*

It holds that $\mathcal{A} = \{i \in N | p < c_i\} = \{1, 3\}$ *and* $\mathcal{B} = \{i \in N | p \geq c_i\} = \{2, 4, 5\}$*. We can then determine* d_i *and* d'_i *by setting* $d_i = \min\{\frac{p}{r_i}, \frac{c_i}{r_i}\}$ *and rounding it down to the nearest power of 2 to define* d'_i*. These parameters are also given in Table 4.1.*

Figure 4.8 shows the schedule following from Policy 4.7. This schedule uses two vehicles, while a feasible schedule using one vehicle exists, such as the schedule in Figure 4.9. Therefore, the approximation bound is tight.

Table 4.1: The demand rate, capacity, and algorithm parameters.

Location	r_i	c_i	d_i	d'_i
1	4	20	$\min\{\frac{10}{4}, \frac{20}{4}\} = \frac{5}{2}$	2
2	1	5	$\min\{\frac{10}{1}, \frac{5}{1}\} = 5$	4
3	1	20	$\min\{\frac{10}{1}, \frac{20}{1}\} = 10$	8
4	1	10	$\min\{\frac{10}{1}, \frac{10}{1}\} = 10$	8
5	1	9	$\min\{\frac{10}{1}, \frac{9}{1}\} = 9$	8

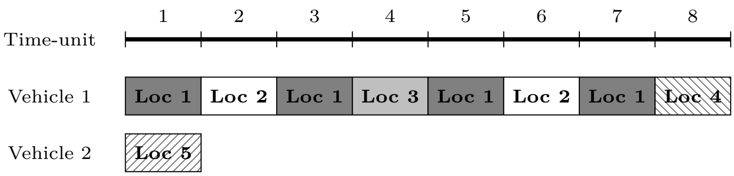


Figure 4.8: Illustration of the Po2 Policy for Example 4.6.

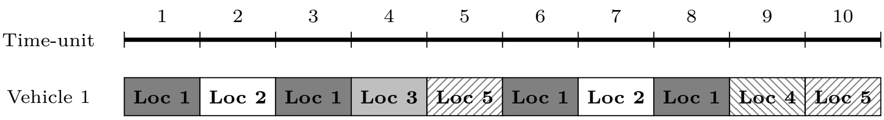


Figure 4.9: Optimal solution for Example 4.6.

4.6 Generalised Capacitated Resupply Problem

In this section, we consider the GCRP, *i.e.*, the most general problem in which capacities, demand rates, and resupply times are heterogeneous. Namely, we have n locations that are resupplied by vehicles with payload p, where location $i \in N$ has capacity c_i, demand r_i and resupply time t_i. In Section 4.6.1, we discuss using a reduction to the CRP or the GHCRP to construct policies for the GCRP. Then, we discuss the relation with the Windows Scheduling Problem in Section 4.6.2.

4.6.1 Reduction to Simplified Variant

In this section, we discuss the reduction of the GCRP to one of its simplified variants, the CRP. The CRP is a variant of the GCRP where resupply times are homogeneous and can be solved by a 2-approximation policy (Policy 4.7). To solve the GCRP, we can therefore reduce the instance to CRP-instances based on the resupply times (Policy 4.8).

Policy 4.8 (Reduce (R) Policy)**.** *Let T be the set of unique resupply times. Create instances $I_\tau = \{i \in N \mid t_i = \tau\}$ for $\tau \in T$ and transform each instance to an equivalent instance with unit-length resupply times. Each instance I_τ is now a CRP-instance and can be solved using Policy 4.7. Finally, merge all schedules using the original resupply times.*

For an instance I of the GCRP, we can define a lower bound similar to the lower bound defined for the CRP in Section 4.5. Namely, if we let $\mathcal{A} = \{i \in N \mid p < c_i\}$ and $\mathcal{B} = \{i \in N \mid p \geq c_i\}$, then all locations $i \in \mathcal{A}$ should be visited on average once every $\frac{p}{r_i}$ time periods and all locations $i \in \mathcal{B}$ should be visited once every $\frac{c_i}{r_i}$. Therefore, the vehicles need to spend at least $\frac{t_i r_i}{p}$ time on locations $i \in \mathcal{A}$ and at least $\frac{t_i r_i}{c_i}$ time on locations $i \in \mathcal{B}$. This implies that a lower bound on the required number of vehicles is equal to $\left\lceil \sum_{i \in \mathcal{A}} \frac{t_i r_i}{p} + \sum_{i \in \mathcal{B}} \frac{t_i r_i}{c_i} \right\rceil$.

Theorem 4.11. *Policy 4.8 uses at most $2m^* + |T|$ vehicles, where T is the set of unique resupply times.*

Proof. In Policy 4.8, the instance is split in instances I_τ for $\tau \in T$ such that all locations in instance I_τ have a resupply time equal to τ. Remember that we can transform any instance with homogeneous resupply times to an instance with unit-length resupply times. Specifically, for each instance I_τ, we can set $t = 1$ and $r'_i = \tau r_i$ for each location $i \in I_\tau$. Each instance I_τ is then a CRP-instance that can be solved using Policy 4.7, which results in a feasible solution by Theorem 4.10. For location $i \in I_\tau$, it then follows that $d_i = \min\left\{\frac{p}{r'_i}, \frac{c_i}{r'_i}\right\} = \min\left\{\frac{p}{\tau r_i}, \frac{c_i}{\tau r_i}\right\}$. Let $\mathcal{A}_\tau$ and $\mathcal{B}_\tau$ be the set of locations $i \in I_\tau$ for which $p < c_i$ and $p \geq c_i$, respectively. Then,

$$m(\mathrm{R}) = \sum_{\tau \in T} \left\lceil \sum_{i \in I_\tau} \frac{1}{d'_i} \right\rceil \leq \sum_{\tau \in T} \left(\sum_{i \in I_\tau} \frac{1}{d'_i} + 1 \right) = \sum_{\tau \in T} \sum_{i \in I_\tau} \frac{1}{d'_i} + |T|$$
$$\leq 2 \sum_{\tau \in T} \sum_{i \in I_\tau} \frac{1}{d_i} + |T| = 2 \sum_{\tau \in T} \left(\sum_{i \in \mathcal{A}_\tau} \frac{\tau r_i}{p} + \sum_{i \in \mathcal{B}_\tau} \frac{\tau r_i}{c_i} \right) + |T|$$

$$= 2\left(\sum_{i\in\mathcal{A}} \frac{t_i r_i}{p} + \sum_{i\in\mathcal{B}} \frac{t_i r_i}{c_i}\right) + |T| \leq 2m^* + |T|.$$

■

We can thus solve the GCRP by reducing it to multiple CRP-instances. Finding the number of required vehicles can therefore be done in $O(n^2)$ time, since at most n instances of CRP are constructed and the number of vehicles required for a CRP instance can be found in $O(n)$ time. However, when the number of unique resupply times is large, the policy can perform poorly due to the large number of instances created. Therefore, we can alternatively reduce the number of unique resupply times at the cost of scheduling longer resupply times (Policy 4.9).

Policy 4.9 (Round & Reduce (RR) Policy)**.** *For each location $i \in N$, round t_i up to the nearest power of 2 and denote this by t_i'. Then, let T' be the set of rounded resupply times and create instances I_τ for each $\tau \in T'$, where each instance consists of the locations with the same resupply time. After scaling the resupply times, each instance I_τ is now a CRP-instance and can be solved using Policy 4.7.*

Corollary 4.2. *Policy 4.9 uses at most $4m^* + \lceil \log t_{max} \rceil$ vehicles, where $t_{max} = \max_{i\in N} t_i$.*

Proof. In Policy 4.9, the instance is split in instances I_τ for $\tau \in T'$, where T' is the set of rounded resupply times. Since for each I_τ we can set $t = 1$ and $r_i' = \tau r_i$ for all $i \in I_\tau$, each instance I_τ is then a CRP-instance that can be solved using Policy 4.7. This results in a feasible solution by Theorem 4.10. For location $i \in I_\tau$, it then follows that $d_i = \min\left\{\frac{p}{r_i'}, \frac{c_i}{r_i'}\right\} = \min\left\{\frac{p}{\tau r_i}, \frac{c_i}{\tau r_i}\right\}$. Then,

$$\begin{aligned} m(\text{RR}) &= \sum_{\tau\in T'} \left\lceil \sum_{i\in I_\tau} \frac{1}{d_i'} \right\rceil \leq \sum_{\tau\in T'} \left(\sum_{i\in I_\tau} \frac{1}{d_i'} + 1\right) = \sum_{\tau\in T'} \sum_{i\in I_\tau} \frac{1}{d_i'} + |T'| \\ &\leq 2 \sum_{\tau\in T'} \sum_{i\in I_\tau} \frac{1}{d_i} + |T'| = 2\sum_{\tau\in T'} \left(\sum_{i\in\mathcal{A}_\tau} \frac{\tau r_i}{p} + \sum_{i\in\mathcal{B}_\tau} \frac{\tau r_i}{c_i}\right) + |T'| \\ &= 2\left(\sum_{i\in\mathcal{A}} \frac{t_i' r_i}{p} + \sum_{i\in\mathcal{B}} \frac{t_i' r_i}{c_i}\right) + |T'| \leq 4\left(\sum_{i\in\mathcal{A}} \frac{t_i r_i}{p} + \sum_{i\in\mathcal{B}} \frac{t_i r_i}{c_i}\right) + |T'| \\ &\leq 4m^* + \lceil \log t_{max} \rceil. \end{aligned}$$

■

Since it holds that $|T'| \leq |T|$, we can decrease the additive term at the cost of rounding the resupply times. This policy also requires $O(n^2)$ time to find the number of required vehicles.

The problem can also be reduced to the GHCRP, which is the variant of the GCRP where locations are homogeneous in terms of their capacity and demand rate. Using a similar approach to Policy 4.8 requires creating instances for each unique combination of capacity and demand rate, resulting in many instances. To reduce the number of instances, both the capacity and demand rate would have to be adjusted, which comes at a higher cost than solely adjusting the resupply time. Since the approximation guarantee for the CRP is better than for the GHCRP, this approach is unlikely to perform better than a reduction to the CRP.

Alternatively, a reduction to the GHCRP can be made by imposing a strict deadline on the visits to each region. This creates an instance of the Windows Scheduling Problem, which we discuss in Section 4.6.2.

4.6.2 The Relation to the Windows Scheduling Problem

An instance of the Windows Scheduling Problem consists of n jobs with windows $w_1, \ldots, w_n$ and lengths $\ell_1, \ldots, \ell_n$. The aim is to find the minimum number of machines such that each job i, which takes ℓ_i time periods to execute, is scheduled at least once in any w_i consecutive time periods. This problem relates to the GCRP in which visits of length t_i have to be scheduled to locations to avoid stock-outs. The difference between the Windows Scheduling Problem and the GCRP is that there is more flexibility in the timing of visits in the GCRP. Namely, while on average a visit should take place at least once every $\frac{p}{r_i}$ time periods, locations can have capacity to keep stock above the level of the vehicle payload. Therefore, the time between visits can sometimes be longer if compensated by a shorter time between visits at other times.

Since the Windows Scheduling Problem and the GCRP are related, a natural thing would be to consider a reduction to this problem. Namely, by reducing an instance of the GCRP, $I = \{p, (c_1, r_1, t_1), \ldots, (c_n, r_n, t_n)\}$, to an instance of the Windows Scheduling Problem, $I^W = \{(t_1, w_1), \ldots, (t_n, w_n)\}$, with $w_i = \min\left\{\left\lfloor \frac{c_i}{r_i} \right\rfloor, \left\lfloor \frac{p}{r_i} \right\rfloor\right\}$. For all locations where $w_i < \left\lfloor \frac{c_i}{r_i} \right\rfloor$, this comes at the price of imposing a strict deadline of scheduling a visit once every $\left\lfloor \frac{p}{r_i} \right\rfloor$ time periods.

It is unknown what the price of imposing a strict deadline of $\left\lfloor \frac{p}{r_i} \right\rfloor$ for all $i \in N$ is. However, we expect this price to be low for locations $i \in N$ for which c_i is relatively low compared to p, as the flexibility in scheduling the visits to these locations is limited. This is also the case for the GHCRP, where imposing a strict deadline of $\left\lfloor \frac{p}{r} \right\rfloor$ costs a factor 3 if the capacity is low ($c \leq 2p$). If the capacity is large, there is more flexibility in scheduling the visits and we expect the price of imposing a strict deadline to be higher. Since on average a visit should be scheduled once every $\frac{p}{r_i}$ time periods, we expect the price to be bounded. Furthermore, since the precise timing of the visits is not important when c_i is large enough, a version of the Wrap Around Policy (Policy 4.1) can be used in that case.

4.7 Conclusion

In this chapter, we introduced the Generalised Capacitated Resupply Problem (GCRP), which we proved to be strongly NP-hard through a reduction from 3-Partition. In the GCRP, vehicles with a given payload have to resupply a set of locations with a given capacity, demand rate, and resupply time. We aim to find the minimum number of vehicles such that no stock-out occurs at any of the locations.

In addition to the GCRP, we considered three variants of this problem in which the locations are homogeneous in terms of the capacity and demand rate (GHCRP), the resupply times (CRP), or both (HCRP). For each of the variants, we describe policies that can be used to find a feasible schedule, and provide the corresponding approximation guarantees. An overview of these results is given in Figure 4.10.

For the GCRP policies, we reduce the instance to an instance of CRP, where resupply times are homogeneous, and use the corresponding 2-approximation policy. Alternatively, a reduction to the related Windows Scheduling Problem can be used. Here, a strict deadline on when the next visit to a location should have been made is imposed. The price of imposing this strict deadline is an interesting open question for future research.

	$t_i = t$	Arbitrary t_i
Arbitrary c_i and r_i	$2m^*$ CRP	$2m^* + \lvert T \rvert$ $4m^* + \lceil \log t_{max} \rceil$ GCRP
$c_i = c$ $r_i = r$	m^* HCRP	$c < 2p$: $4m^*$ $3(1+\epsilon)m^* + 1$ $c \geq 2p$: $2m^*$ GHCRP

Figure 4.10: Upper bounds on the number of required vehicles for the policies for the GCRP, GHCRP, CRP and HCRP.

Chapter 5

Outbreak prevention in low- and middle-income countries: Investing in local health facilities or in mobile laboratories?

5.1 Introduction

Public health experts have increasingly recognised the vital role of *surveillance*, *i.e.*, the collection and testing of samples to identify and monitor the spread of pathogens (Gensheimer et al., 1999; Kelly-Cirino et al., 2019). Emergencies such as the Ebola outbreaks in 2014 and the COVID-19 pandemic have highlighted persistent gaps in surveillance infrastructure, particularly in low- and middle-income countries (LMICs) (Aborode et al., 2021; Worsley-Tonks et al., 2022). Mobile laboratories (labs) have emerged as an innovative solution to this challenge (Praesens Care, 2024; Racine and Kobinger, 2019). Mobile labs are vans equipped with advanced diagnostic technologies that enable rapid on-site testing. They can travel across a country, thereby enhancing surveillance in multiple regions.

However, the implementation of mobile labs faces several challenges. The first challenge is the high cost of these labs, coupled with chronic underfunding of surveillance systems (Micah et al., 2023). Second, the value of mobile labs compared to investments in local health facilities remains unclear. Although research has explored the use of mobile health units (Alban et al., 2022; Breugem et al., 2023; De Vries et al., 2021), it treats them as stand-alone solutions. Understanding when mobile labs can add value requires evaluating them within the existing health system and considering the alternative of improving local health facilities. This comparison is essential to communicate their value to stakeholders. Finally, equity is an important concern in public health resource allocation (Cooper et al., 2018; Smith, 2015). Resource allocations that fail to account for equity concerns between multiple regions may lack the necessary support from regional stakeholders, even if they offer nationwide benefits.

This chapter aims to assess the value of mobile labs while considering limited budgets, existing health systems, and equity concerns. Specifically, we consider the perspective of a health ministry tasked with distributing a fixed budget across multiple regions, either to improve local health facilities or to deploy mobile labs. We evaluate the impact of these solutions on the speed of outbreak detection, which is crucial given the exponential spread of diseases at the onset of an outbreak (Kermack and McKendrick, 1927). We first consider identical regions, after which we consider regions that differ in the quality of their local health facilities.

We find that mobile labs can offer significant value when budget is tight. From our analytical analysis we find that the deployment of mobile labs can then result in at least a 10% additional reduction in outbreak detection time. Our numerical analysis, based on real-world data, shows that the time reduction can be significantly higher in practice. However, even when the regions are identical, the allocated budgets to the regions may differ. For an equitable solution, mobile labs are preferred when budget is tight, while investments in local health facilities are superior when there is a large budget. When regions differ in the current quality of their health facilities, mobile labs add relatively more value in regions where health facilities are of lower quality. This can lead to inverse inequality, where regions with initially higher-quality health facilities face longer times until outbreak detection.

The main contributions of this chapter are twofold. First, we assess the value of mobile labs while considering investing in the existing health system. Second, we introduce an equity perspective, addressing a critical issue that has been largely overlooked in previous research. By considering these important challenges, our approach offers a more realistic view of when mobile labs can improve the performance of existing health systems.

The remainder of this chapter is structured as follows. In Section 5.2, we discuss related literature. We formulate the problem and the underlying assumptions in Section 5.3. We discuss the value of mobile labs and potential equity concerns in Section 5.4 and conclude with Section 5.5.

5.2 Literature Review

This research corresponds to the literature on disease surveillance and modelling, and on resource/budget allocation problems.

5.2.1 Disease Surveillance and Modelling

While health experts have stressed the importance of disease surveillance for outbreak prevention, recent outbreaks have highlighted gaps in the surveillance infrastructure (Aborode et al., 2021; Worsley-Tonks et al., 2022). Different solutions have been raised to improve surveillance in LMIC, which include improved data management and data access (Fleming et al., 2021), training programs (André et al., 2017), and increasing diagnostic capacity (Worsley-Tonks et al., 2022). Mobile labs have been identified as an alternative way of improving surveillance by offering high-quality diagnostic capacity (Racine and Kobinger, 2019). They have proven to be successful in outbreak response (Fall et al., 2020; Grolla et al., 2011), but their value in disease surveillance compared to investments in local health facilities remains unclear. We consider a budget allocation problem in which we consider both investments in local health facilities and in mobile labs.

To assess the spread of an infectious disease, compartment models are often used (Kermack and McKendrick, 1927). The non-linearity of compartment models makes analytical analysis challenging or even infeasible. As a result, approximations or simulations are frequently used (Harweg et al., 2022; Rao and Brandeau, 2021). In our study, we focus on allocating a limited budget to enhance surveillance of disease spread in low- and middle-income countries. Since diseases spread exponentially at the onset of an outbreak (Kermack and McKendrick, 1927), it is critical to implement preventive measures to reduce the spread of the infectious disease as soon as possible. Therefore, instead of approximating the number of infectious, we evaluate how a given budget allocation affects the detection time.

In order to evaluate the value of mobile labs compared to investment in local health facilities, we thus require the effect of a given mobile lab visit frequency or investment in local health facilities on the detection time. The effect of a given healthcare investment on health outcomes is often unclear (Nixon and Ulmann, 2006). (Hensher et al., 2024) have shown that diminishing marginal returns on investments in health-care. Therefore, we assume diminishing returns of investments in surveillance on the detection time.

5.2.2 Resource and Budget Allocation Problems

In resource and budget allocation problems, scarce resources have to be distributed among competing purposes to optimise a cost or benefit function (Patriksson, 2008). This research relates most to resource allocation problems in epidemic control and visit allocation problems.

Resource allocation problems in epidemic control focus on distributing scarce resources to control epidemics through prevention or treatment programs (Brandeau, 2004). The aim is to minimise the impact of an epidemic by determining how and to whom these resources should be allocated. For instance, decisions may involve how to distribute vaccines among different populations (Duijzer et al., 2018a; Enayati and Özaltın, 2020), or selecting which type of vaccine should be used (Duijzer et al., 2018b). A general model for allocating scarce resources across populations to mitigate the effects of an epidemic is presented by Brandeau et al. (2003), who analyse various cost functions and derive optimality conditions.

Visit allocation problems consider optimal assignment of limited visiting capacity of mobile units across different regions. Objectives in such problems often relate to the return time of the mobile unit. For example, De Vries et al. (2021) investigate the relationship between the mean return time of mobile units and the demand for family planning services, aiming to determine the optimal visit frequency for various regions. Dhanjal-Adams et al. (2016) focus on minimising disturbances to birds through patrol scheduling, examining both linear and logarithmic relationships between site disturbance and the number of visits. Breugem et al. (2023) analyse the effects of providing multiple services with a single mobile unit compared to using separate units for each service on the overall demand.

Compared to visit allocation problems, we also consider the alternative of investing in local health facilities. This requires not only determining how to allocate the limited budget among regions, but also deciding the type of investment that should be used.

5.3 Problem Formulation

We consider a country's health ministry that needs to allocate a budget B across different regions to improve disease surveillance. For each region, the budget can be spent on either improving local health facilities or on scheduling mobile lab visits. Investing in local health facilities results in a permanent improvement in surveillance, while mobile labs provide a periodic improvement for each region since they travel between multiple regions. In this section, we define the benefits of these two types of investments in improving surveillance and introduce the corresponding mathematical model.

Health ministries typically define a threshold for the number of infected individuals required to implement preventive measures, such as social distancing for airborne diseases and mosquito nets for malaria (World Health Organisation, 2022a). We define *outbreak detection* as the time when this threshold is observed. Since diseases spread exponentially at the onset of an outbreak (Kermack and McKendrick, 1927), the goal is to detect outbreaks as quickly as possible. Therefore, we define the benefit of an investment in surveillance as the expected reduction in outbreak detection time.

We consider a set of m regions. Let p_i be the probability of an outbreak occurring in region i. The benefit of allocating a budget x_i to region i, assuming an outbreak occurs, is denoted by $b_i(x_i)$. We then define the budget allocation problem as follows, in which the aim is to maximise the expected benefit of the budget allocation:

$$\begin{aligned} \max \ & \sum_{i=1}^{m} p_i b_i(x_i) \\ \text{s.t.} \ & \sum_{i=1}^{m} x_i \leq B \\ & 0 \leq x_i \leq C_i \qquad \forall i = 1, \ldots, m, \end{aligned} \tag{5.1}$$

where C_i represents an upper bound on the budget that can be allocated to region i.

Since the budget allocated to a region can be spent either on improving local health facilities or on scheduling mobile lab visits, the total benefit is given by:

$$b_i(x_i) = \max_{A \in \{M,F\}} \left\{ b_i^A(x_i) \right\}, \tag{5.2}$$

where $b_i^A(x_i)$ denotes the benefit of spending x_i on $A \in \{M, F\}$, with M corresponding to mobile labs and F to local health facilities. We define these benefit functions in Sections 5.3.1 and 5.3.2.

5.3.1 The Benefit of Mobile Lab Visits

We calculate the benefit of mobile lab visits as the reduction in outbreak detection time achieved when x_i budget is spent on mobile lab visits to region i. We define the cost of acquiring a mobile lab as C^M. Following the notation of Breugem et al. (2023), we express $\frac{x_i}{C^M}$ as the proportion of the lab's visit capacity allocated to region i over a standardised period. Over a planning horizon of T time periods, this is equivalent to scheduling $\frac{x_i}{C^M}T$ visits to the region.

We make the following assumptions to derive the benefit function. First, the planning horizon is the time frame during which a disease is prevalent, *i.e.*, an outbreak is equally likely in any given time period during the planning horizon. For instance, the planning horizon might be the summer for malaria and the winter for COVID. Therefore, the benefit of the planned mobile lab visits is independent of the timing of the first visit.

Second, we assume that in at least one region mobile labs are more efficient in detecting outbreaks compared to the current local health facilities. This assumption is necessary for mobile labs to have potential value. Otherwise, the entire budget should always be allocated to investments in local health facilities. This assumption holds for LMICs, where local health facilities often have limited surveillance capab-

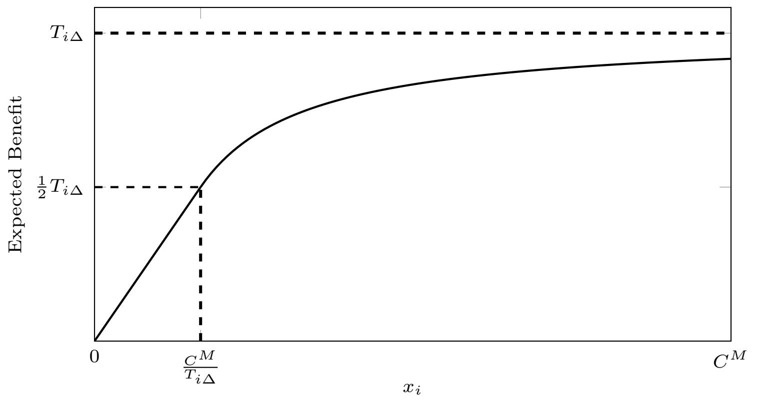


Figure 5.1: The benefit function for allocating a budget x_i to mobile lab visits in region i.

ilities. The potential expected reduction in the detection time from using mobile labs in region i is denoted as $T_{i\Delta} = T_i^F - T_i^M$, where T_i^F and T_i^M represent the expected outbreak detection time in region i using local health facilities and mobile labs, respectively. We assume that $T_{i\Delta} > 0$ for all $i \in \{1, \ldots, m\}$.

Finally, we assume that the parameters T_i^F and T_i^M are known. In practice, these could be estimated by, for example, using an SIR model or one of its variants (Kermack and McKendrick, 1927) to determine the time it takes to reach the threshold for the number of infected people beyond which the government will impose preventive measures.

Using these assumptions, we define the benefit function for mobile labs as:

$$b_i^M(x_i) = \begin{cases} \frac{1}{2}T_{i\Delta}^2 \frac{x_i}{C^M} & 0 \leq x_i \leq \frac{C^M}{T_{i\Delta}} \\ T_{i\Delta} - \frac{C^M}{2x_i} & \frac{C^M}{T_{i\Delta}} \leq x_i \leq C^M, \end{cases} \tag{5.3}$$

with visits to the region spread equally over the planning period (see Appendix 5.A for the derivation).

We observe that the benefit of mobile labs is concave in the amount of budget invested. Namely, it first increases linearly and then increases at a diminishing rate as more budget is allocated. This concavity is illustrated in Figure 5.1.

5.3.2 The Benefit of Investment in Local Health Facilities

LMICs often face two main challenges related to the surveillance capacity of local health facilities. First, many people do not get tested due to limited access, especially in poor, rural areas (World Health Organisation, 2022b). Second, samples often need to be sent elsewhere for testing, as local health facilities lack on-site laboratories. This leads to delays in obtaining test results (Yadav et al., 2021). Investing in local health facilities can enhance surveillance by improving access to testing and reducing delays. For example, access can be improved by using motorbikes to collect samples from remote areas. Additionally, acquiring diagnostic equipment or using motorbikes to transport samples faster and more frequently to laboratories can reduce the delay in obtaining test results. $b_i^F(x_i)$ denotes the benefit of investing x_i in these measures.

We note that no benefit is obtained when the budget is zero, *i.e.*, $b_i^F(0) = 0$. We make the following assumptions about the benefit function for local health facilities. First, investing in local health facilities should provide at least the same benefit as permanently allocating a mobile lab. Mathematically, this can be expressed as $b_i^F(C^M) \geq b_i^M(C^M) = T_{i\Delta} - \frac{1}{2}$.

Second, we assume diminishing marginal returns. Research on the effect of health expenditure on health outcomes has shown diminishing marginal returns (Hensher et al., 2024; Nixon and Ulmann, 2006). Furthermore, research on the effect of improving accessibility and capacity has also shown diminishing returns (Levinson and Wu, 2020; Palvannan and Teow, 2012). Mathematically, we assume that $\frac{d}{dx_i} b_i^F(x_i) > 0$ and $\frac{d^2}{dx_i^2} b_i^F(x_i) < 0$.

5.4 Analysis and Results

We are interested in (i) the value of mobile labs, *i.e.*, the additional reduction in detection time they enable, and (ii) when inequity can arise and the price of fairness, *i.e.*, the percentage difference between the optimal and equitable solutions. We analyse the budget allocation problem both analytically and numerically.

For the numerical analysis, we use data on the spread of COVID-19 and the deployment of mobile labs in Mombasa County, Kenya, by Praesens Care, a company that designs and deploys mobile labs (Praesens Care, n.d.). Mombasa County consists of six sub-counties, of which two sub-counties are urban regions (Mvita and Nyali). Praesens Care deployed mobile labs in Mombasa County to $m = 13$ health facilities spread over the six sub-counties (see Table 5.1). On average, these health facilities monitor approximately 9,700 households, equivalent to almost 38,000 people using an average household size of 3.9 (Statista, 2020). Samples taken at these health facilities are tested at a public hospital, which is, on average, located 4.7 km from the health facilities. During the 16-week pilot of Praesens Care, 3,000 patients were targeted

Table 5.1: The number of monitored households and distance to the nearest level-4 health facility for the 13 health facilities in Mombasa, Kenya.

Sub-County	Health Facility	Monitored Households[1]	Distance Nearest Public Hospital (km)[2]
Likoni	NYS Health Center	–[3]	5.9
	Mbuta Dispensary	2200	5.7
Jomvu	Jomvu Model	25000	8.1
	Miritini CDF Health Center	35110	8.5
Kisauni	Junda Dispensary	15000	4.1
Mvita	Mvita Health Center	2000	1.9
	Majengo Dispendary	1997	1.5
	Ganjoni CDC Health Center	2000	3.9
Changamwe	Magongo Health Center	12500	4.5
	Bokole CDF Health Center	6110	5.2
Nyali	Kongowea Dispensary	5210	2.6
	Maweni Health Center	5555	4.4
	Ziwa La Ngombe Health Center	3999	4.9
Average		9723.42	4.71

[1]: obtained from Ministry of Health, Kenya (n.d.)
[2]: obtained from Praesens Care (n.d.)
[3]: not available

for various tests, which is equivalent to almost 40 tests per day. The turnaround time of the tests offered ranged from 10 minutes to 3 hours. The same tests had a turnaround time of 30 minutes to 24 hours at the public hospital.

To numerically analyse the value of mobile labs, we need to calculate the expected detection time for local health facilities and permanently allocated mobile labs. We model the spread of COVID-19 in Kenya using an SEIR model (Cooke, 1967) and determine the number of positive tests per day based on the testing capacity to find the expected detection time for local health facilities and mobile labs. The SEIR model and its parameters are given in Appendix 5.C from which we derive a potential reduction in the detection time for mobile labs of $T_{\Delta} = 4.42$ weeks. We normalise the value of the budget such that one unit of budget corresponds to the price of one mobile lab.[1] We refer to this case using the parameters in Appendix 5.C as the base case. The benefit function for mobile labs is represented by the black line in Figure 5.2.

We use a logarithmic benefit function for investment in local health facilities defined as $b^F(x) = d \ln(x+1)$. The grey area in Figure 5.2 shows the benefit function of investment in local health facilities for $d \in [10, 20]$, where the expected benefit is increasing in d. This range ensures that the assumptions stated in Section 5.3 hold. We use $d = 15$ (dashed line) for the numerical analysis in this section and consider budget levels $B \in \{0, 0.05, 0.1, \ldots, m \times C^M\}$. Appendix 5.D, presents a sensitivity analysis on the parameter d and the SEIR model parameters, which shows similar trends to those presented in this section.

[1]The price of a mobile lab (C^M) is approximately 0.5 million Euros.

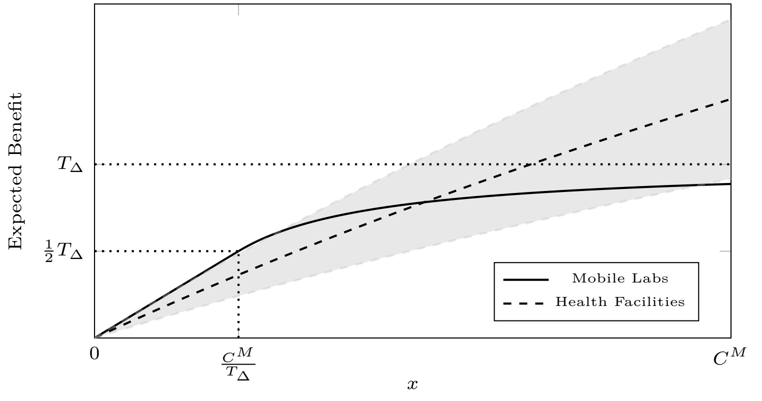


Figure 5.2: Benefit functions for mobile labs and investment for the base case. The grey region shows the range in which the benefit function for health facilities lies for $d \in [10, 20]$, while the black dashed line shows the benefit function for $d = 15$.

We first consider the case where all regions are identical in Section 5.4.1. This simplified version allows us to analytically analyse the benefit that can be obtained using mobile labs. Then, we consider the case where the regions are non-identical in Section 5.4.2.

5.4.1 Identical Regions

In this section, we assume all regions are identical, *i.e.*, $T_{i\Delta} = T_\Delta$, $C_i = C$, $p_i = p$, and $b_i(x_i) = b(x_i)$ for all $i \in \{1, \ldots, m\}$. We first describe the conditions an optimal solution adheres to. Then, we discuss the value of mobile labs, potential equity concerns, and the price of fairness.

Given that all regions are identical, all regions that receive mobile lab visits are visited with equal frequency and all regions that receive investment in local health facilities receive the same budget (see Theorem B1 in Appendix 5.B). This allows us to simplify the model. Let $\mathcal{S}^M = \left\{x \in [0, C] \mid b^M(x) > b^F(x)\right\}$ be the set of budget levels for which the benefit is larger for mobile lab visits, $\mathcal{S}^F = \left\{x \in [0, C] \mid b^F(x) > b^M(x)\right\}$ be the set of budget levels for which the benefit is larger for investments in local health facilities, and $\mathcal{S}^I = \left\{x \in [0, C] \mid b^M(x) = b^F(x)\right\}$ for which the benefit is equal for both types of investment. If two regions are allocated the same investment type, they are assigned the same budget. That is, if regions i and j receive the same investment type, then $x_i = x_j$. Based on this, we can formulate the problem as follows:

$$\begin{aligned} \max\ & nb^M(x^M) + (m-n)b^F(x^F) \\ \text{s.t. } & nx^M + (m-n)x^F \leq B \\ & x^M \in \mathcal{S}^M \cup \mathcal{S}^I \\ & x^F \in \mathcal{S}^F \cup \mathcal{S}^I \\ & 0 \leq n \leq m \\ & n \in \mathbb{N}, \end{aligned}$$

This problem has a non-linear objective function and an integer decision variable, which makes it difficult to solve. However, the number of regions that receive mobile lab visits, n, is bounded by m. We can thus consider the problem for each possible value of n and solve the corresponding problem. If $n = 0$, all regions are assigned investment in local health facilities, each receiving a budget of $\frac{B}{m}$. If $n = m$, all regions receive a budget of $\frac{B}{m}$ each for mobile lab visits. For intermediate values of n, we have:

Theorem 5.1. *Given the number of regions that receive mobile lab visits, n, the budget is allocated such that either the marginal benefit of additional investment is equal for both investment types ($b^{M'}(x^M) = b^{F'}(x^F)$), or such that the budget for at least one investment type is allocated to a level where the benefit functions intersect ($x^M \in \mathcal{S}^I \vee x^F \in \mathcal{S}^I$).*

We compare the reduction in outbreak detection time when mobile labs are included or excluded from the solution space to analyse the value of mobile labs. Corollary 5.1 shows that mobile labs can detect outbreaks significantly faster (proof in Appendix 5.B).

Corollary 5.1. *If the benefit function for investment in local health facilities is logarithmic, i.e., $b^F(x) = d\ln(x+1)$, mobile labs can reduce the outbreak detection time by at least 10% compared to investments in local health facilities alone.*

Mobile labs thus add value when they are assigned to at least one region in the optimal solution, *i.e.*, when $n > 0$. The question is whether this added value is significant. Our numerical analysis reveals that the value of mobile labs can be significantly higher than the 10% lower bound stated in Corollary 5.1. Figure 5.3 shows the optimal n, the number of regions receiving mobile lab visits (dotted line, right y-axis), and the expected benefits in the optimal solution (solid line, left y-axis). It also displays the expected benefits of the counterfactual scenario in which

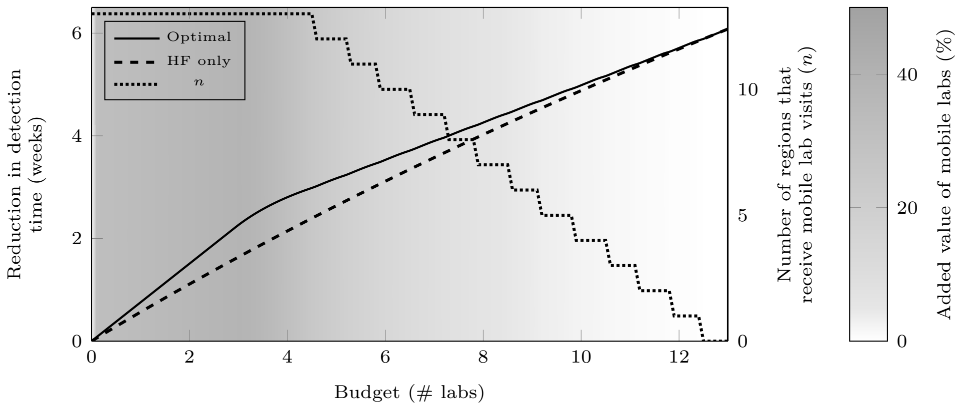


Figure 5.3: The expected reduction in the detection time in the optimal solution and the solution when all budget is invested in local health facilities (HF) for the base case (right y-axis). The dotted line (left y-axis) shows the number of regions that receive mobile lab visits. The shaded area shows the value of mobile labs, *i.e.*, the additional benefit that can be obtained in the optimal solution compared to only considering investment in local health facilities.

the entire budget is allocated to investment in local health facilities (dashed line, left y-axis). The shaded area shows the value of mobile labs, *i.e.*, the additional benefit in the optimal solution compared to the counterfactual scenario. A darker colour indicates a greater value of mobile labs.

The optimal solution allocates at least one mobile lab, unless the budget is very high (equivalent to dedicating almost one mobile lab per region). Mobile labs provide the most value at lower budget levels, where all regions receive mobile lab visits. As the budget increases, the value rises up to 40%, then decreases as higher budgets enable significant investments in some local health facilities.

In Appendix 5.D, we present a sensitivity analysis to the SEIR parameters that affect the potential time reduction of mobile labs, T_Δ, for the base case with $d = 15$. This shows similar trends, *i.e.*, the value of mobile labs is largest for small budget levels and (almost) negligible for very large budget levels where investment in local health facilities is prioritised.

In an equitable solution, identical regions should receive the same budget. However, Theorem B1 implies that identical regions may receive different budgets in an optimal solution. Namely, if $n \in [1, m-1]$, the regions are split into two groups. One group receives mobile lab visits and the other group receives investment in local health facilities. The two groups are allocated different amounts of budget, which creates inequality between the regions. As indicated by the dotted line in Figure 5.3

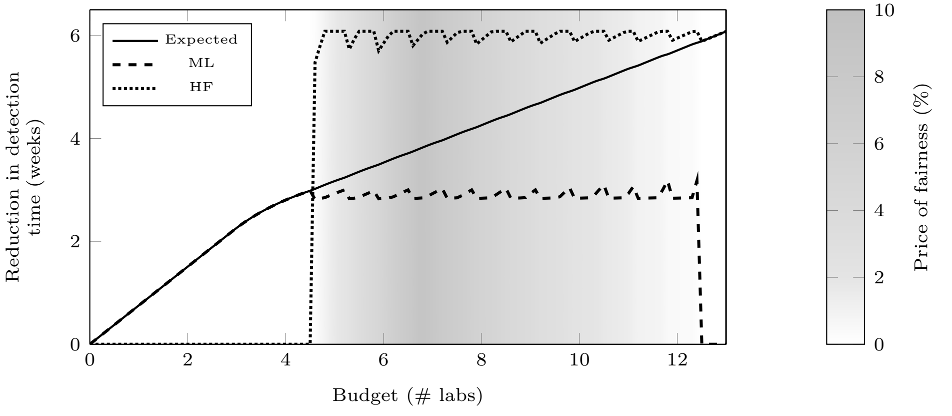


Figure 5.4: The expected reduction in the detection time in the optimal solution and the expected reduction in the detection time for regions that receive mobile lab visits (ML) and investment in local health facilities (HF) for the base case. The shaded area shows the price of fairness, *i.e.*, the percentage difference between the optimal and equitable budget allocation in which all regions receive the same budget.

all regions receive the same investment type only when the budget is tight or very large.

Figure 5.4 shows the expected reduction in the detection time for all regions, the regions that receive mobile lab visits, and the regions that receive investment in local health facilities. The shaded area shows the price of fairness, *i.e.*, percentage difference between the optimal and equitable solution in which all regions receive the same budget. A darker colour corresponds to a higher price of fairness.

For small and very large budget levels, the price of fairness is zero as all regions receive the same investment type (mobile labs for small budget levels and investment in local health facilities for very large budget levels). For medium budget levels, the regions are treated differently as they receive different types of investment. The regions that receive investment in local health facilities receive a budget that is between 2.5 and 3 times as high, which corresponds to a time reduction that is almost twice as high. The price of fairness is at most 8.9%.

We thus see that when mobile labs add most value (small budget levels), the optimal solution is equitable since all regions receive mobile lab visits. The optimal solution is also equitable at the other extreme, where there is enough budget for significant investments in all regions. Equity becomes a concern at medium to high budget levels, where large investments in some regions lead to significant improvements but limit resources for others.

5.4.2 Non-identical Regions

The simplified version in Section 5.4.1 in which all regions are identical, allowed us to give an analytical bound on the added value of mobile labs. In many contexts, particularly in low- and middle-income countries, the accessibility of health facilities and the availability of equipment can vary across regions (Lewis et al., 2019; Peters et al., 2008). As a result, the baseline detection times differ between regions. In this section, we incorporate this differentiation into the model by considering two types of regions, regions with low-quality local health facilities and regions with high-quality local health facilities, which we denote by low-resource and high-resource regions, respectively. We first discuss the value of mobile labs and then potential equity concerns. For the numerical analysis, we assume that the health facilities in the urban sub-counties (Mvita and Nyali) are of higher quality than the health facilities in the other sub-counties. We then have $m_H = 6$ high-resource regions and $m_L = 7$ low-resource regions. We assume that testing is faster in high-resource regions, *e.g.*, due to better access to public hospitals. We assume a testing duration of $d_L^F = 10$ days for the low-resource regions and of $d_H^F = 7$ days for the high-resource regions. In Appendix 5.E, we present the results when the regions differ in terms of the testing capacity.

The benefit of mobile labs is dependent on the current quality of the local health facilities as low-resource regions have a higher detection time T_i^F. This results in a larger potential gain from mobile labs, $T_{i\Delta}$, and therefore a larger benefit for the same visit frequency. The benefit function for investment in local health facilities is not directly dependent on the current detection time. We assume that regions with lower quality local health facilities can be compensated through investments, *i.e.*, we treat higher quality local health facilities as if some investments in the region have already been made.

Similar to the case where regions are identical, we can find the following condition for an optimal solution, where $\mathcal{S}_i^I = \left\{x \in \left[0, C^M\right] \mid b_i^M(x) = b_i^F(x)\right\}$ is the set of budget levels for which the benefit is the same for mobile labs and investment in local health facilities for region i:

Corollary 5.2. *In an optimal solution, regions get assigned a budget such that either the marginal benefit of additional investment is equal to the marginal benefit of other regions, or at a budget level for which the benefit functions intersect, i.e., $x_i \in S_i^I$ for region i.*

Figure 5.5 indicates the added value of mobile labs at different budget levels. We

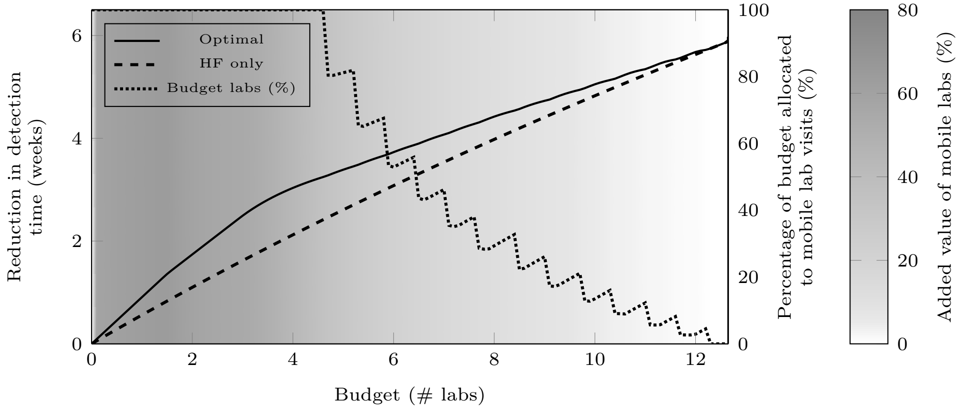


Figure 5.5: The expected reduction in the detection time in the optimal solution and the solution when all budget is invested in local health facilities (HF) (right y-axis) for the base case with testing durations $d_L^F = 10$ days and $d_H^F = 7$ days. The dotted line (left y-axis) shows the percentage of the available budget that is allocated to mobile lab visits. The shaded area shows the value of mobile labs, *i.e.*, the additional benefit that can be obtained in the optimal solution compared to only considering investment in local health facilities.

find similar results as for the case where regions are identical. Mobile labs are used unless the budget is very high and their added value is largest for small budget levels (over 60% at its maximum).

To analyse potential equity concerns in the optimal solution for the two types of regions, we consider two regions – one low-resource region and one high-resource region – and examine the implications for the division of the budget and the expected detection time for both types of investment.

Figure 5.6 shows the optimal budget allocation for mobile labs visits. The grey area (left y-axis) shows the percentage of the budget that is allocated to the low-resource region. The lines (right y-axis) show the detection time for the two regions. At low budget levels, the entire budget is allocated to the low-resource region, reducing the detection time gap between the regions. However, the gap widens in the opposite direction as the budget increases further. At medium and high budget levels, the frequency of mobile lab visits is sufficient to make them the primary factor in detecting outbreaks in both regions, making detection time independent of the quality of local health facilities. As a result, both regions receive the same budget and have the same detection time.

Figure 5.7 shows the optimal budget allocation for investment in local health facilities. For small budget levels, the budget is allocated to the low-resource region

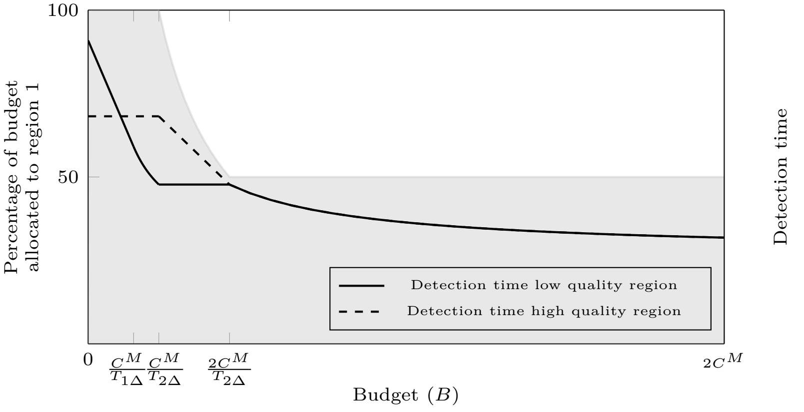


Figure 5.6: Graphical illustration of the budget allocation for mobile labs to two regions with different quality local health facilities. The grey area shows the percentage of the total budget allocated to the low-resource region (left y-axis). The lines show the detection time for the two regions (right y-axis).

which causes its detection time to converge with that of the high-resource region. For medium and large budgets, the percentage of the budget allocated to the low-resource region decreases. Once this region is sufficiently compensated for its initial disadvantage, the remaining budget is distributed between both regions, resulting in equal detection times.

We thus see that non-identical regions can receive different budget levels, even if the investment type is the same. While there is no universally accepted definition of equity (de Oliveira et al., 2024), the expected detection time for each region serves as a reasonable measure in our context. Specifically, we consider a solution to be equitable when identical regions receive the same budget, and the detection times of the regions do not differ more than the current difference in the detection times. Using this definition, we observe that when all regions receive mobile lab visits, inverse inequality can arise, where regions with higher current detection times, have a significantly lower detection time in the optimal solution.

Corollary 5.3. *When regions differ in the current quality of local health facilities, inverse inequality can arise, where regions with currently a higher detection time have a significantly lower detection time in the optimal solution.*

Figure 5.8 shows the expected reduction in the detection time for different regions. The shaded area shows the price of fairness, *i.e.*, percentage difference between the optimal and equitable solutions as defined above. A darker colour corresponds to a

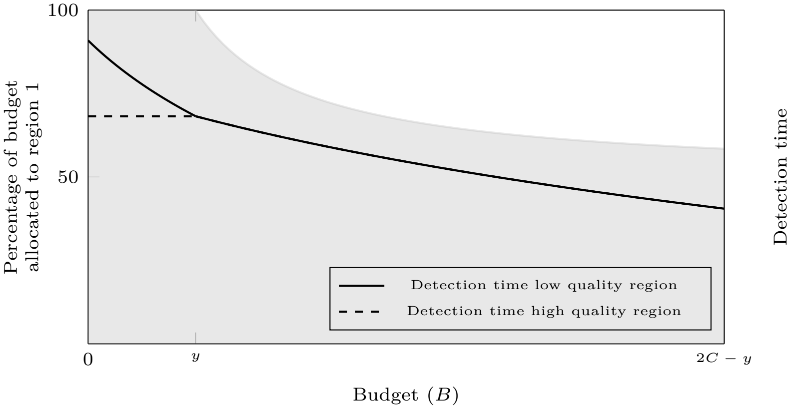


Figure 5.7: Graphical illustration of the budget allocation for investment in local health facilities to two regions with different quality local health facilities. The grey area shows the percentage of the total budget allocated to the low-resource region (left y-axis). The lines show the detection time for the two regions (right y-axis).

higher price of fairness.

For small budget levels, all budget is allocated to the low-resource regions. This leads to inverse inequality, as the reduction in the detection time exceeds the current difference in the detection time of almost half a week. This puts the high-resource regions at a disadvantage. For medium budget levels, investments are made in the local health facilities of some high-resource regions. The expected reduction in the detection time is then larger for the high-resource regions, which results in more inequality compared to the current situation. The price of fairness is at most 7.9%, *i.e.*, the expected reduction in the detection time is at most 7.9% lower for an equitable solution compared to the optimal solution.

Mobile labs thus add most value when the budget is small. The budget allocation can be inequitable since inverse inequality can arise. In an equitable solution, mobile labs can however still add value. For small budget levels, the value of mobile labs in the equitable solution is more than 30%.

In Appendix 5.E, we present the results when the regions differ in the testing capacity ($c_L^F = 5$ versus $c_H^F = 10$). We find, similar to the results in this section, that mobile labs add most value when the budget is small. While investments in high-resource local health facilities are also prioritised for medium budget levels, this does not necessarily result in an inequitable budget allocation. Inequality thus only occurs for small budget levels.

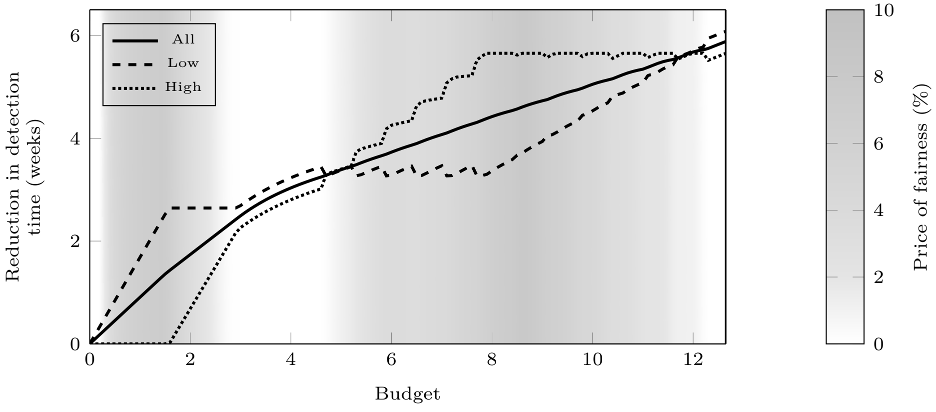


Figure 5.8: The expected reduction in the detection time for all regions, the low-resource regions, and the high-resource regions for the base case with testing durations $d_L^F = 10$ days and $d_H^F = 7$ days. The shaded area shows the price of fairness, *i.e.*, the percentage difference between the optimal and equitable expected benefit.

5.5 Conclusion

Recent global health crises, such as the Ebola outbreaks and the COVID-19 pandemic, revealed gaps in the surveillance, *i.e*, the collection and testing of samples to identify and monitor the spread of pathogens, in low- and middle-income countries. Mobile laboratories (labs), which provide on-site diagnostic testing capabilities, have emerged as a potential solution to enhance surveillance across multiple regions. However, their high costs and uncertain comparative value to investments in local health facilities complicate decision-making for health ministries. Moreover, equity concerns regarding resource allocation between different regions are important, as inequitable solutions may fail to gain the necessary support. Addressing these issues is crucial for improving outbreak detection capabilities and ensuring that resources are allocated effectively and fairly within constrained budgets.

We consider a budget allocation problem in which a limited budget must be allocated across different regions, either to improve the local health facilities or to schedule mobile lab visits. We aim to maximise the reduction in the detection time such that preventive measures can be imposed quickly. We analyse the value of mobile labs, *i.e.*, the additional reduction in detection time they enable, and the price of fairness, *i.e.*, the percentage difference between the optimal and equitable solutions. We analysed the problem both analytically and using data from the deployment of mobile labs by Praesens Care in combination with data on the spread of COVID-19 in Mombasa County, Kenya.

We find that large benefits can be obtained by including mobile labs in the surveillance system, especially at small budget levels. We identify a few potential equity concerns in the optimal budget allocation but find that even in equitable budget allocations, mobile labs can add significant value when the budget is tight. Here we use an outcome-based measure for equity where we consider a solution equitable based on the post-intervention detection time. We discuss advantages and disadvantages of this method and discuss the effect of using an allocation-based measure for equity where equity is defined based on the allocated budget.

We highlight three potential areas for future research. First, we have used some assumptions regarding the benefit function for investments in local health facilities. While we have some support for choosing a concave benefit function, it would be interesting to (empirically) analyse the effect of investments on the detection time of local health facilities. Second, regions may differ in other ways that affect the detection time. We have considered differences in the testing duration, *i.e.*, the time it takes to obtain the test result once a sample is taken, and the testing capacity. While these showed similar trends, it would be interesting to analyse which regional characteristics affect the detection time of an infectious disease and whether these follow similar trends. Finally, we considered a one-time investment as a first step in analysing the value of mobile labs compared to investment in local health facilities. In reality, the budget may become available over multiple time periods. In this case, mobile labs can be used as an intermediate solution until enough budget is available to make significant improvements in local health facilities, which can reduce (temporary) inequity. It would be interesting to analyse the value of mobile labs and the effect of using mobile labs on the equity of the budget allocation in a multi-period setting.

Appendix

5.A Derivation of the Benefit Function of Mobile Labs

We derive the benefit function (5.3) for a single region and therefore drop the subscript i denoting the region. We are interested in the expected benefit, measured as the expected time reduction in outbreak detection, when x budget is spent on mobile lab visits. Here we use the notation of Breugem et al. (2023). We let $\frac{x}{C^M}$ be the fraction of a mobile lab that is allocated to the region, where C^M is the cost of acquiring a mobile lab. The mean return time of the lab to the region is then equal to $\frac{C^M}{x}$ periods, *i.e.*, a visit is scheduled on average once every $\frac{C^M}{x}$ periods.

By our assumptions, it takes T^F time periods to detect an outbreak by local health facilities and T^M time periods to detect an outbreak by a mobile lab that is permanently allocated to a region. The difference in outbreak detection time is defined by $T_\Delta = T^F - T^M > 0$. The benefit of scheduling mobile lab visits can be defined as $\max\{T_\Delta - J, 0\}$, where J is equal to the time until the first mobile lab visit to the region after the disease has been active for T^M periods.

Let t_i be the time between visit $i-1$ and i and assume the disease has not been identified during the $(i-1)^{\text{th}}$ visit and is identified during the i^{th} visit at the latest. Since the disease is equally likely to emerge in each time period, J is a random variable with probability density function $f_J(j) = \frac{1}{t_i}$ for $j \in [0, t_i)$. Let $E[b](t_i)$ denote the expected benefit of scheduling the next visit after t_i time periods. We consider two cases. First, we consider the case where $t_i \leq T_\Delta$. Second, we consider the case where $t_i > T_\Delta$.

If $t_i \leq T_\Delta$, the outbreak will be detected by the next mobile lab visit. Namely, given that the outbreak was not detected during the previous mobile lab visit, less than T^M time periods had passed since the first infection. Therefore, if $t_i \leq T_\Delta$, local health facilities will not detect the outbreak before the next mobile lab visit. The benefit is then equal to $T_\Delta - J$ such that

$$E[b \mid t_i \leq T_\Delta](t_i) = E[T_\Delta - J] = T_\Delta - E[J] = T_\Delta - \frac{t_i}{2}.$$

If $t_i > T_\Delta$, the outbreak can either be detected by the mobile lab during the next visit, or by the local health facilities. It then holds that

$$\begin{aligned}
E[b \mid t_i > T_\Delta](t_i) &= E[\max\{T_\Delta - J, 0\} \mid t_i > T_\Delta](t_i) \\
&= \int_0^{t_i} f_J(j) \max\{T_\Delta - j, 0\} dj \\
&= \int_0^{T_\Delta} \frac{1}{t_i}(T_\Delta - j) dj + \int_{T_\Delta}^{t_i} 0 \times \frac{1}{t_i} dj \\
&= \left[\frac{1}{t_i} T_\Delta j - \frac{1}{t_i} \times \frac{1}{2} j^2\right]_0^{T_\Delta} = \frac{1}{t_i} T_\Delta^2 - \frac{1}{2}\frac{1}{t_i} T_\Delta^2 = \frac{1}{2} T_\Delta^2 \frac{1}{t_i}.
\end{aligned}$$

We thus find the following expected benefit given that we will detect the outbreak at the latest during the next visit scheduled in t_i time periods:

$$E\left[b\right](t_i) = \begin{cases} \frac{1}{2} T_\Delta^2 \frac{1}{t_i} & t_i > T_\Delta \\ T_\Delta - \frac{1}{2} t_i & t_i \leq T_\Delta. \end{cases}$$

For a given budget, x, $n = \frac{C^M}{x} T$ visits are scheduled during a planning horizon T. Therefore, it should hold that the average return time is equal to $\frac{C^M}{x}$ and $\frac{1}{n} \sum_{i=1}^{n} t_i = \frac{C^M}{x}$. We now show that in an optimal solution, visits are equally spread, *i.e.*, $t_i = \frac{C^M}{x}$ for all $i = 1, \ldots, n$.

Consider scheduling two visits in t_1 and t_2 time periods, for $t_1 < t_2$. It should hold that $\frac{t_1 + t_2}{2} = \frac{C^M}{x}$. Since we assume the disease is equally likely to emerge in each period, the expected benefit is defined as

$$\frac{t_1}{t_1 + t_2} E[b](t_1) + \frac{t_2}{t_1 + t_2} E[b](t_2).$$

We show that the expected benefit is lower than or equal to scheduling two visits in $\frac{C^M}{x}$ time periods, $E[b]\left(\frac{C^M}{x}\right)$, based on the value of T_Δ relative to t_1 and t_2.

1. If $T_\Delta < t_1 < t_2$, we find

$$\begin{aligned}
\frac{t_1}{t_1 + t_2} E[b](t_1) + \frac{t_2}{t_1 + t_2} E[b](t_2) &= \frac{t_1}{t_1 + t_2} \frac{1}{2} T_\Delta^2 \frac{1}{t_1} + \frac{t_2}{t_1 + t_2} \frac{1}{2} T_\Delta^2 \frac{1}{t_2} \\
&= \frac{1}{t_1 + t_2} T_\Delta^2 = \frac{1}{2} T_\Delta^2 \frac{x}{C^M} = E[b]\left(\frac{C^M}{x}\right)
\end{aligned}$$

2. If $t_1 < t_2 \leq T_\Delta$, we find

$$\frac{t_1}{t_1 + t_2} E[b](t_1) + \frac{t_2}{t_1 + t_2} E[b](t_2)$$

$$
\begin{aligned}
&= \frac{t_1}{t_1+t_2}\left(T_\Delta - \frac{1}{2}t_1\right) + \frac{t_2}{t_1+t_2}\left(T_\Delta - \frac{1}{2}t_2\right) \\
&= \frac{T_\Delta(t_1+t_2)}{t_1+t_2} - \frac{1}{2}\frac{t_1^2+t_2^2}{t_1+t_2} = T_\Delta - \frac{1}{2}\frac{t_1^2+t_2^2}{t_1+t_2} \\
&< T_\Delta - \frac{1}{4}(t_1+t_2) = T_\Delta - \frac{C^M}{2x} = E[b]\left(\frac{C^M}{x}\right),
\end{aligned}
$$

where $\frac{t_1^2+t_2^2}{t_1+t_2} > \frac{1}{2}(t_1+t_2)$ holds since $(t_1-t_2)^2 = t_1^2+t_2^2-2t_1t_2 > 0$.

3. If $t_1 \le T_\Delta < t_2$, we find

$$
\begin{aligned}
\frac{t_1}{t_1+t_2}E[b](t_1) + \frac{t_2}{t_1+t_2}E[b](t_2) &= \frac{t_1}{t_1+t_2}\left(T_\Delta - \frac{1}{2}t_1\right) + \frac{t_2}{t_1+t_2}\frac{1}{2}T_\Delta^2\frac{1}{t_2} \\
&= \frac{t_1T_\Delta - \frac{1}{2}t_1^2 + \frac{1}{2}T_\Delta^2}{t_1+t_2}.
\end{aligned}
$$

We now consider two cases:

(a) If $T_\Delta \le \frac{C^M}{x}$, we find

$$
\begin{aligned}
\frac{t_1T_\Delta - \frac{1}{2}t_1^2 + \frac{1}{2}T_\Delta^2}{t_1+t_2} &= \left(t_1T_\Delta - \frac{1}{2}t_1^2 + \frac{1}{2}T_\Delta^2\right)\frac{x}{2C^M} \\
&\le \frac{1}{2}T_\Delta^2\frac{x}{C^M} = E[b]\left(\frac{C^M}{x}\right),
\end{aligned}
$$

where $t_1T_\Delta - \frac{1}{2}t_1^2 + \frac{1}{2}T_\Delta^2 \le T_\Delta^2$ holds since

$$
\begin{aligned}
t_1T_\Delta - \frac{1}{2}t_1^2 + \frac{1}{2}T_\Delta^2 \le T_\Delta^2 &\iff t_1^2 - 2t_1T_\Delta + T_\Delta^2 \ge 0 \\
&\iff (t_1 - T_\Delta)^2 \ge 0.
\end{aligned}
$$

(b) If $T_\Delta > \frac{C^M}{x}$, we find

$$
\begin{aligned}
\frac{t_1T_\Delta - \frac{1}{2}t_1^2 + \frac{1}{2}T_\Delta^2}{t_1+t_2} &= \frac{t_1T_\Delta + t_2T_\Delta - t_2T_\Delta - \frac{1}{2}t_1^2 + \frac{1}{2}T_\Delta^2}{t_1+t_2} \\
&= \frac{T_\Delta(t_1+t_2)}{t_1+t_2} - \frac{t_2T_\Delta + \frac{1}{2}t_1^2 - \frac{1}{2}T_\Delta^2}{t_1+t_2} \\
&= T_\Delta - \frac{t_2T_\Delta + \frac{1}{2}t_1^2 - \frac{1}{2}T_\Delta^2}{t_1+t_2}.
\end{aligned}
$$

Then,

$$
\begin{aligned}
& T_\Delta - \frac{t_2 T_\Delta + \frac{1}{2}t_1^2 - \frac{1}{2}T_\Delta^2}{t_1 + t_2} \leq E[b]\left(\frac{C^M}{x}\right) \\
\iff & T_\Delta - \frac{t_2 T_\Delta + \frac{1}{2}t_1^2 - \frac{1}{2}T_\Delta^2}{t_1 + t_2} \leq T_\Delta - \frac{C^M}{2x} = T_\Delta - \frac{1}{4}(t_1 + t_2) \\
\iff & t_2 T_\Delta + \frac{1}{2}t_1^2 - \frac{1}{2}T_\Delta^2 \geq \frac{1}{4}(t_1 + t_2)^2.
\end{aligned}
$$

For a given t_1 and t_2, $\frac{1}{4}(t_1+t_2)^2$ is a constant and $t_2 T_\Delta + \frac{1}{2}t_1^2 - \frac{1}{2}T_\Delta^2$ is a parabola in T_Δ. Therefore, it suffices to show that this condition holds for both $T_\Delta = t_2$ and $T_\Delta = \frac{C^M}{x} = \frac{t_1+t_2}{2}$.
If $T_\Delta = t_2$, we get

$$
t_2^2 + \frac{1}{2}t_1^2 - \frac{1}{2}t_2^2 = \frac{1}{2}t_1^2 + \frac{1}{2}t_2^2 \geq \frac{1}{4}(t_1 + t_2)^2,
$$

where we use that $\frac{1}{2}t_1^2 + \frac{1}{2}t_2^2 > \frac{1}{4}(t_1 + t_2)$ since $(t_1 - t_2)^2 = t_1^2 + t_2^2 - 2t_1t_2 > 0$.
If $T_\Delta = \frac{t_1+t_2}{2}$, we get

$$
\begin{aligned}
& t_2\left(\frac{t_1+t_2}{2}\right) + \frac{1}{2}t_1^2 - \frac{1}{2}\left(\frac{t_1+t_2}{2}\right)^2 \geq \frac{1}{4}(t_1+t_2)^2 \\
\iff & t_2\left(\frac{t_1+t_2}{2}\right) + \frac{1}{2}t_1^2 - \frac{1}{8}(t_1+t_2)^2 \geq \frac{1}{4}(t_1+t_2)^2 \\
\iff & t_2^2 + t_1^2 + t_1t_2 - \frac{1}{4}(t_1+t_2)^2 \geq \frac{1}{2}(t_1+t_2)^2 \\
\iff & \frac{1}{4}t_1^2 + \frac{1}{4}t_2^2 + t_1t_2 - \frac{1}{2}t_1t_2 \geq \frac{1}{2}t_1t_2 \iff \frac{1}{4}t_1^2 + \frac{1}{4}t_2^2 \geq 0.
\end{aligned}
$$

In an optimal solution visits are thus equally spread. We can write the benefit function in terms of the budget spend, x, as follows:

$$
b^M(x) = E\left[b\right]\left(\frac{C^M}{x}\right) = \begin{cases} \frac{1}{2}T_\Delta^2 \frac{x}{C^M} & 0 \leq x \leq \frac{C^M}{T_\Delta} \\ T_\Delta - \frac{C^M}{2x} & \frac{C^M}{T_\Delta} \leq x \leq C^M. \end{cases}
$$

5.B Mathematical Proofs

Theorem B1. *Regions that get the same type of investment (mobile lab visits or investment in local health facilities), get the same amount of budget.*

Proof. Consider an optimal solution $x_1^*, \ldots, x_m^*$, where there exist regions i and j

such that $x_i^* < x_j^*$, and both regions i and j receive the same type of investment. Since both benefit functions are concave, we have:

$$2b^A(\bar{x}) > b^A(x_1^*) + b^A(x_2^*) \qquad \text{for} \qquad \bar{x} = \frac{x_1^* + x_2^*}{2}, \quad A \in \{M, F\}.$$

This creates a contradiction, which implies that the optimal solution cannot have two regions with unequal budget that receive the same type of investment. Hence, the budget must be the same for regions receiving the same type of investment. ∎

Theorem B2. *Let $h_1 : \mathbb{R} \to \mathbb{R}$ and $h_2 : \mathbb{R} \to \mathbb{R}$ be convex, decreasing, and differentiable functions. Let $n, m \in \mathbb{N}$ such that $n < m$. Furthermore, let $b > 0$ be such that $h_1(b) = h_2(b)$ and let $0 < a < b$ be such that $h_1'(a) = h_2'(b)$. Consider the following budget allocation problem:*

$$\begin{aligned} \min\ & nh_1(x_1) + (m-n)h_2(x_2) \\ s.t.\ & nx_1 + (m-n)x_2 \leq B \\ & a \leq x_1 \leq b \\ & b \leq x_2 \leq c. \end{aligned}$$

Then,

(a) if $B < na + (m-n)b$, there exists no feasible budget allocation.

(b) if $B = na + (m-n)b$, the only feasible, and hence optimal, solution is $x_1 = a$ and $x_2 = b$.

(c) if $B > na + (m-n)b$, then either of the following cases occur:

(c.1) $x_1 \in (a, b)$ and $x_2 \in (b, c)$ such that $B = nx_1 + (m-n)x_2$ and $h_1'(x_1) = h_2'(x_2)$.

(c.2) $x_1 = b$ and $x_2 \in (b, c)$ such that $B = nx_1 + (m-n)x_2$ and $h_1'(b) < h_2'(x_2)$.

(c.3) $x_1 \in (a, b)$ and $x_2 = c$ such that $B = nx_1 + (m-n)x_2$ and $h_1'(x_1) > h_2'(c)$.

(c.4) $x_1 = b$ and $x_2 = c$.

Proof. To prove Theorem B2, we prove each of the cases, displayed in Figure 5.9, separately:

(a) As $x_1 \geq a$ and $x_2 \geq b$, the minimum required budget is equal to $na + (m-n)b$. Hence, if $B < na + (m-n)b$, there exists no feasible solution.

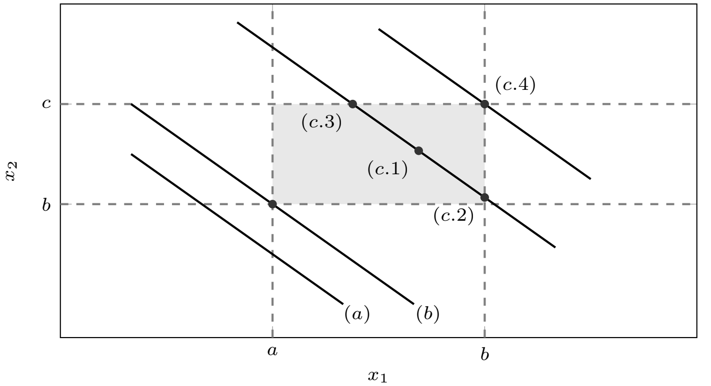


Figure 5.9: Overview of the cases of Theorem B2. The grey area shows the feasible range for x_1 and x_2. The black lines correspond to different budget constraints. A feasible budget allocation lies within the grey area, on or below the black line.

(b) As $x_1 \geq a$ and $x_2 \geq b$, if $B = na + (m-n)b$, the only feasible, and hence optimal, solution is $x_1 = a$ and $x_2 = b$.

(c) Both $h_1(x)$ and $h_2(x)$ are convex functions, hence the objective is convex. Furthermore, the constraints in the problem are affine. Therefore, the problem is convex. Furthermore, as $B > na + (m-n)b$, there exists an $\epsilon > 0$ such that $x_1 = a + \epsilon$ and $x_2 = b + \epsilon$ is feasible with $nx_1 + (m-n)x_2 < B$, $a < x_1 < b$, and $b < x_2 < c$. Hence, Slater's conditions hold. Therefore, any point that satisfies the Karush-Kuhn-Tucker (KKT) conditions is an optimal solution. We therefore first define the KKT conditions, which are then used to prove optimality for each of the cases and argue why another solution is not possible.

The Lagrangian function for this problem is

$$\begin{aligned}\mathcal{L}(x_1, x_2, \lambda_1, \ldots, \lambda_5) = {} & nh_1(x_1) + (m-n)h_2(x_2) \\ & + \lambda_1(nx_1 + (m-n)x_2 - B) + \lambda_2(a - x_1) + \lambda_3(x_1 - b) \\ & + \lambda_4(b - x_2) + \lambda_5(x_2 - c).\end{aligned}$$

Thus, for any optimal solution, there exists $\lambda_1, \ldots, \lambda_5 \in \mathbb{R}$ such that

$$\begin{aligned}& nh_1'(x_1) + \lambda_1 n - \lambda_2 + \lambda_3 = 0 \\ & (m-n)h_2'(x_2) + (m-n)\lambda_1 - \lambda_4 + \lambda_5 = 0 \\ & nx_1 + (m-n)x_2 \leq B\end{aligned}$$

$$a \leq x_1 \leq b$$
$$b \leq x_2 \leq c$$
$$\lambda_1, \ldots, \lambda_5 \geq 0$$
$$\lambda_1(nx_1 + (m-n)x_2 - B) = 0$$
$$\lambda_2(a - x_1) = 0$$
$$\lambda_3(x_1 - b) = 0$$
$$\lambda_4(b - x_2) = 0$$
$$\lambda_5(x_2 - c) = 0.$$

We now consider the different cases:

(c.1) If $x_1 \in (a,b)$ and $x_2 \in (b,c)$, then the KKT conditions hold for $\lambda_2 = \cdots = \lambda_5 = 0$, and $\lambda_1 = -h_1'(x_1) = h_2'(x_2)$ such that $nx_1 + (m-n)x_2 = B$ and $h_1'(x_1) = h_2'(x_2)$.

(c.2) If $x_1 = b$ and $x_2 \in (b,c)$, then the KKT conditions hold for $\lambda_2 = \lambda_4 = \lambda_5 = 0$, $\lambda_1 = -h_2'(x_2)$ and $\lambda_3 = nh_2'(x_2) - nh_1'(x_1)$. In other words, as $\lambda_1 > 0$, $n_1x_1 + n_2x_2 = B$, and as $\lambda_3 \geq 0$, $h_1'(x_1) < h_2'(x_2)$.

(c.3) If $x_1 \in (a,b)$ and $x_2 = c$, then the KKT conditions hold for $\lambda_2 = \lambda_3 = \lambda_4 = 0$, $\lambda_1 = -h_1'(x_1)$ and $\lambda_3 = (m-n)h_1'(x_1) - (m-n)h_2'(x_2)$. In other words, as $\lambda_1 > 0$, $n_1x_1 + n_2x_2 = B$, and as $\lambda_3 \geq 0$, $h_1'(x_1) > h_2'(x_2)$.

(c.4) If $x_1 = b$ and $x_2 = c$, then the KKT conditions hold for $\lambda_1 = \lambda_2 = \lambda_4 = 0$, $\lambda_3 = -nh_1'(x_1)$, and $\lambda_5 = -(m-n)h_2'(x_2)$.

The remaining cases are $x_1 = a$ and/or $x_2 = b$.

- If $x_1 = a$ and $x_2 = b$, then the KKT conditions hold for $\lambda_3 = \lambda_5 = 0$, $\lambda_2 = nh_1'(x_1) + \lambda_1 n$ and $\lambda_4 = nh_2'(x_2)$. As it should hold that $\lambda_2 \geq 0$ and $\lambda_4 \geq 0$, it should hold that $\lambda_1 > 0$ because $h_1'(x) < 0$ and $h_2'(x) < 0$. However, as $B > na + (m-n)b$, this provides a contradiction as we require $\lambda_1 = 0$.
- If $x_1 = a$ and $x_2 \in (b,c)$, then the KKT conditions hold for $\lambda_3 = \lambda_4 = \lambda_5 = 0$, $\lambda_1 = -h_2'(x_2)$, and $\lambda_2 = nh_1'(x_1) - nh_2'(x_2)$. As $\lambda_2 > 0$, this implies that $h_1'(x_1) > h_2'(x_2)$. However, as $h_1'(a) = h_2'(b)$ and $h_2(x)$ has diminishing marginal returns by assumption, $h_1'(a) < h_2'(x_2)$ for $x_2 > b$ and we have a contradiction.

- If $x_1 \in (a, b)$ and $x_2 = b$, then the KKT conditions hold for $\lambda_2 = \lambda_3 = \lambda_5 = 0$, $\lambda_1 = -h_1'(x_1)$, and $\lambda_4 = (m-n)h_2'(x_2) - (m-n)h_1'(x_1)$. As $\lambda_4 > 0$, this implies that $h_1'(x_1) < h_2'(x_2)$. However, as $h_1'(a) = h_2'(b)$ and $h_1(x)$ has diminishing marginal returns by assumption, $h_1'(x_1) \leq h_2'(b)$ for $x_1 > a$ and we have a contradiction.

$\blacksquare$

Theorem 5.1. *Given the number of regions that receive mobile lab visits, n, the budget is allocated such that either the marginal benefit of additional investment is equal for both investment types ($b^{M'}(x^M) = b^{F'}(x^F)$), or such that the budget for at least one investment type is allocated to a level where the benefit functions intersect ($x^M \in \mathcal{S}^I \vee x^F \in \mathcal{S}^I$).*

Proof. Consider the minimisation version of the problem, *i.e.*, minimise $-nb^M(x^M) - (m-n)b^F(x^F)$. Since $b^M(x)$ and $b^F(x)$ are concave, increasing, and differentiable, the functions $-nb^M(x)$ and $-(m-n)b^F(x)$ are convex, decreasing, and differentiable. By applying Theorem B2 from Appendix 5.B, it follows that the budget allocation will be such that either the marginal benefit of spending an additional euro is the same across all investment types, or at least one of the budget values will be set at an intersection point between the two benefit functions. $\blacksquare$

Theorem B3. *If the benefit of investment in local health facilities is defined by a logarithmic function, i.e., $b^F(x) = d\ln(x+1)$ for some $d > 0$, then the benefit functions of investing in local health facilities and mobile labs intersect at most twice.*

Proof. It holds that both benefit functions are concave and that $b^M(0) = b^F(0) = 0$. Furthermore, by our assumptions, it must hold that $b^M(C^M) < b^F(C^M)$.

For the benefit of mobile lab visits the marginal benefit is defined as

$$\frac{d}{dx}b^M(x) = \begin{cases} \frac{1}{2}T_\Delta \frac{1}{C^M} & 0 \leq x \leq \frac{C^M}{T_\Delta} \\ \frac{C^M}{2x^2} & \frac{C^M}{T_\Delta} \leq x \leq C^M, \end{cases}$$

where it holds that $\frac{1}{2}T_\Delta \frac{1}{C^M} \geq \frac{C^M}{2x^2}$ for any $x \geq \frac{C^M}{T_\Delta}$.

For the benefit of investment in local health facilities the marginal benefit is defined as

$$\frac{d}{dx}b^F(x) = \frac{d}{x+1}.$$

First, we determine how many times the functions can intersect. For the range $x \in \left[0, \frac{C^M}{T_\Delta}\right]$, $b^M(x)$ is a straight line. Therefore, the benefit functions intersect at most twice. Since $b^M(0) = b^F(0)$, there is at most one intersection point on $x \in \left(0, \frac{C^M}{T_\Delta}\right]$.

For the range $x \in \left[\frac{C^M}{T_\Delta}, C^M\right]$, it holds that

$$\begin{aligned}\frac{d}{dx}b^M(x) = \frac{d}{dx}b^F(x) &\iff \frac{C^M}{2x^2} = \frac{d}{x+1}\\ &\iff 2dx^2 = C^M(x+1)\\ &\iff x = \frac{C^M \pm \sqrt{C^{M^2} + 8dC^M}}{4d} = \frac{C^M\left(1 \pm \sqrt{1 + \frac{8d}{C^M}}\right)}{4d}.\end{aligned}$$

Since $d > 0$ and $C^M > 0$, it holds that $\sqrt{1 + \frac{8d}{C^M}} > 1$. This implies that $x = \frac{C^M\left(1 - \sqrt{1 + \frac{8d}{C^M}}\right)}{4d} < 0$ and that this point is outside the range $x \in \left[\frac{C^M}{T_\Delta}, C^M\right]$. Therefore, the slopes of the benefit functions are equal on at most one point in the range $x \in \left[\frac{C^M}{T_\Delta}, C^M\right]$. Therefore, there are at most two intersection points on the range $x \in \left[\frac{C^M}{T_\Delta}, C^M\right]$.

We can identify the following three cases:

- $b^M(x) \geq b^F(x)$ for all $x \in \left[0, \frac{C^M}{T_\Delta}\right]$.

  Since $b^M(C^M) < b^F(C^M)$, the benefit functions must intersect exactly once on the range $x \in \left[\frac{C^M}{T_\Delta}, C^M\right]$.

- $b^M(x) \leq b^F(x)$ for all $x \in \left[0, \frac{C^M}{T_\Delta}\right]$.

  Since it must hold that $b^M(C^M) < b^F(C^M)$, the function must either never intersect or intersect exactly twice on the range $x \in \left[\frac{C^M}{T_\Delta}, C^M\right]$.

- $b^M(x) \leq b^F(x)$ for all $x \in [0, \bar{x}]$ and $b^M(x) \geq b^F(x)$ for all $x \in \left[\bar{x}, \frac{C^M}{T_\Delta}\right]$.

  Since it must hold that $b^M(C^M) < b^F(C^M)$, the function must intersect one more time on the range $x \in \left[\frac{C^M}{T_\Delta}, C^M\right]$.

Hence, given a logarithmic benefit function for investment in local health facilities, the benefit functions intersect at most twice. ■

Corollary 5.1. *If the benefit function for investment in local health facilities is logarithmic, i.e.,* $b^F(x) = d\ln(x+1)$*, mobile labs can reduce the outbreak detection time by at least 10% compared to investments in local health facilities alone.*

Proof. By Theorem B3, it follows that a logarithmic benefit function intersects at most twice. Consider the case where they intersect once, *i.e.*, $b^F(x) = d\ln(x+1)$ with $d \leq \frac{1}{2}T_\Delta^2 \frac{1}{C^M}$. Then, by Theorem B3, it holds that

$$b(x) = \max\left\{b^M(x), b^F(x)\right\} = \begin{cases} b^M(x) & x \leq \bar{x} \\ b^F(x) & x \geq \bar{x} \end{cases}$$

for $\bar{x} \in \left(\frac{C^M}{T_\Delta}, C\right)$.

The optimality gap, *i.e.*, the additional reduction that can be obtained by deploying mobile labs instead of investing in local health facilities, is equal to

$$\frac{b^M(x) - b^F(x)}{b^F(x)}.$$

Since $b^M(x) \geq b^F(x)$ and $\frac{d}{dx}b^M(x) = \frac{1}{2}T_\Delta^2\frac{1}{C^M} \geq \frac{d}{dx}b^F(x)$ for all $x \in \left[0, \frac{C^M}{T_\Delta}\right]$, it follows that $\max_{x\in[0,C]}\left\{b^M(x) - b^F(x)\right\} \geq b^M\left(\frac{C^M}{T_\Delta}\right) - b^F\left(\frac{C^M}{T_\Delta}\right)$. Therefore,

$$\max_{x\in[0,C]}\frac{b^M(x) - b^F(x)}{b^F(x)} \geq \frac{b^M\left(\frac{C^M}{T_\Delta}\right) - b^F\left(\frac{C^M}{T_\Delta}\right)}{b^F\left(\frac{C^M}{T_\Delta}\right)} \geq \frac{b^M\left(\frac{C^M}{T_\Delta}\right) - b^F\left(\frac{C^M}{T_\Delta}\right)}{b^M\left(\frac{C^M}{T_\Delta}\right)}$$

By using a Taylor expansion, $\exists t \in (0,1)$ such that $b^F(x) = b^F(0) + x\frac{d}{dx}b^F(0) + \frac{1}{2}x^2\frac{d^2}{dx^2}b^F(tx)$. Therefore,

$$b^F(x) = dx - \frac{1}{2}x^2\frac{d}{(tx+1)^2} < d\left(x - \frac{1}{2}x^2\frac{1}{(x+1)^2}\right) < \frac{1}{2}T_\Delta^2\frac{1}{C^M}\left(x - \frac{1}{2}\frac{x^2}{(x+1)^2}\right).$$

At $x = \frac{C^M}{T_\Delta}$, we thus have $b^F\left(\frac{C^M}{T_\Delta}\right) < \frac{1}{2}T_\Delta - \frac{1}{4}C^M\frac{1}{\left(\frac{C^M}{T_\Delta}+1\right)^2}$ and $b^M\left(\frac{C^M}{T_\Delta}\right) = \frac{1}{2}T_\Delta$, which gives

$$\begin{aligned}\max_{x\in[0,C]}\frac{b^M(x) - b^F(x)}{b^F(x)} &\geq \frac{b^M\left(\frac{C^M}{T_\Delta}\right) - b^F\left(\frac{C^M}{T_\Delta}\right)}{b^M\left(\frac{C^M}{T_\Delta}\right)} > \frac{\frac{1}{2}T_\Delta - \frac{1}{2}T_\Delta + \frac{1}{4}C^M\frac{1}{\left(\frac{C^M}{T_\Delta}+1\right)^2}}{\frac{1}{2}T_\Delta} \\ &= \frac{\frac{1}{4}C^M\frac{1}{\left(\frac{C^M}{T_\Delta}+1\right)^2}}{\frac{1}{2}T_\Delta} = \frac{1}{2}\frac{C^M}{T_\Delta}\frac{1}{\left(\frac{C^M}{T_\Delta}+1\right)^2}.\end{aligned}$$

This expression is positive on $[0, C]$ with a maximum at $\frac{C^M}{T_\Delta} = 1$ where the additional time reduction is at least 12.5%. We find that for $\frac{C^M}{T_\Delta} \in (0.056, 17.94)$, it holds that $\frac{1}{2}\frac{C^M}{T_\Delta}\frac{1}{\left(\frac{C^M}{T_\Delta}+1\right)^2} \geq 10\%$. Hence, for a wide range of values of C^M and T_Δ, the additional time reduction is at least 10%. ■

Corollary 5.3. *When regions differ in the current quality of local health facilities, inverse inequality can arise, where regions with currently a higher detection time have a significantly lower detection time in the optimal solution.*

Proof. Consider assigning mobile lab visits to region with different quality local health facilities, and let the regions be sorted such that $T_{1\Delta} \geq T_{2\Delta} \geq \cdots \geq T_{m\Delta}$. Let i and j be such that $T_{i\Delta} > T_{j\Delta}$, *i.e.*, region i has lower quality local health facilities and therefore a larger detection time. Region j will not be assigned budget until $\frac{d}{dx}b_i(x_i) = \frac{d}{dx}b_j(0) = \frac{1}{2}T_{j\Delta}^2\frac{1}{C^M}$. This holds for $x_i = \frac{C^M}{T_{j\Delta}}$ for which $b_i\left(\frac{C^M}{T_{j\Delta}}\right) = T_{i\Delta} - \frac{1}{2}T_{j\Delta}$. The detection time for region i is then equal to

$$T_i^F - T_{i\Delta} + \frac{1}{2}T_{j\Delta} = T_i^F - \left(T_i^F - T^M\right) + \frac{1}{2}\left(T_f^F - T^M\right) = \frac{1}{2}\left(T_j^F + T^M\right).$$

Region j has a detection time of T_j^F, and since $T^M < T_j^F$ by definition, the detection time of region i is smaller than the detection time of region j.

We consider a solution equitable when the difference in the detection time is at most equal to the current difference in the detection time. Since the current difference is equal to $T_i^F - T_j^F$, inverse inequality arises when $T_j^F - \frac{1}{2}\left(T_j^F + T^M\right) > T_i^F - T_j^F$, *i.e.*, when $T^M < 3T_j^F - 2T_i^F$. ■

5.C SEIR Model

The SEIR model is an extension of the SIR model used to model exposure to a disease before someone is infectious. Let S be the fraction of the population that is susceptible, E the fraction of the population that is exposed to the disease, but not yet infectious, I the fraction of the population that is infectious, and R the fraction of the population that is recovered. Let β_0 be the transmission rate of the disease, α the rate of transition from being exposed to being infectious ($1/\alpha$ is the incubation time), and μ the rate of recovery ($1/\mu$ is the infectious period). Then, we can define the SEIR model as follows:

$$\frac{dS}{dt} = -\beta_0 SI$$
$$\frac{dE}{dt} = \beta_0 SI - \alpha E$$

$$\frac{dI}{dt} = \alpha E - \mu I$$
$$\frac{dR}{dt} = \mu I.$$

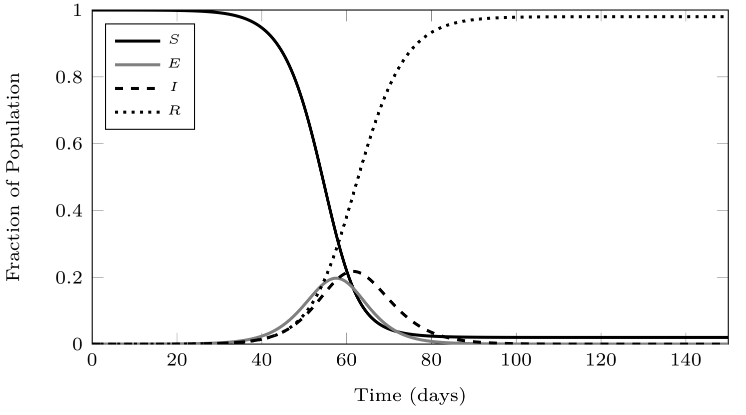
When a pre-defined threshold for the number of positive tests is reached, (non-pharmaceutical) preventive measures are imposed. These reduce the transmission rate β_0 to β_1 (with a delay).

To find the detection time, we determine the number of positive tests each day using a binomial distribution. In other words, we randomly determine the number of successes given a number of trials (tests) and a probability p (the probability that someone is positive). We assume that people with symptoms, the infectious population I, are more likely to get tested. Furthermore, we assume that, since testing capacity is very limited and many people will not know they have been infected, the recovered population might also be tested. We define the probability that someone who is tested is positive as

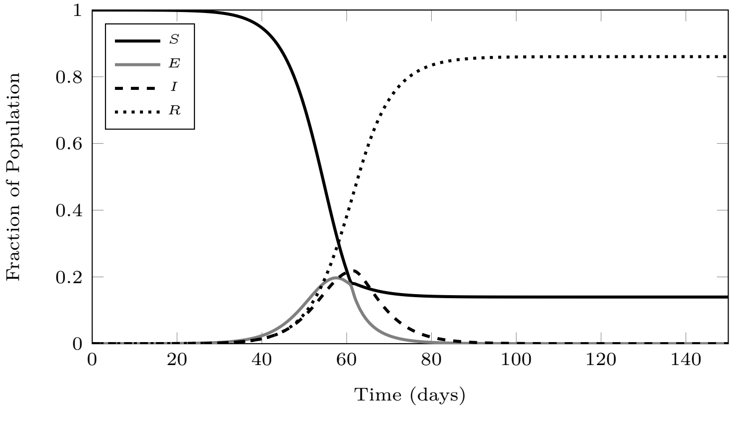
$$p = P(positive \mid tested) = \frac{E + mI}{S + E + mI + R},$$

where m is a multiplier denoting how much more likely someone with symptoms is to get tested.

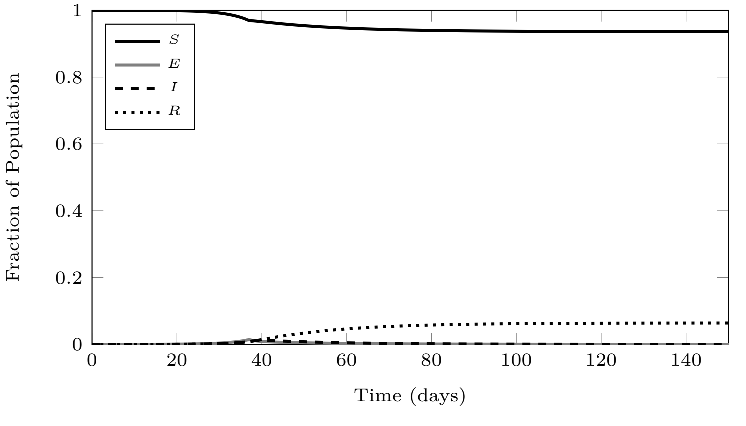
Table 5.2 shows the data used in the base case, which we use in our main results. We vary the parameters to analyse the effect of our parameters on our results. Using the parameters of Table 5.2, we model the spread of the disease using the SEIR model. Without testing, we see that almost everyone gets infected (Figure 5.10a). When testing is done by the local health facility, the disease is identified after approximately 54 days. This reduces the transmission rate after 61 days, which is shown in the S and E curves. This reduces the fraction of the population that gets infected from 98% to 86%, a reduction of more than 4500 people. When testing is done by a mobile lab, the disease is identified after approximately 23 days. Since measures are imposed after 30 days when the exposed and infectious population are still quite small, the disease is prevented from spreading too much. The total fraction of the population that gets infected is 6.4% in this case, a reduction of almost 35000 people.



(a) Without testing



(b) With testing by the local health facility



(c) With testing by the mobile lab

Figure 5.10: Evolution of the spread of the disease over time with and without testing.

Table 5.2: Parameters used to model the spread of COVID and determining the number of positive tests per day by health facilities (HF) or mobile labs (ML) for the base case.

Parameter	Value	Description	Note
α	0.25	Rate of transitioning from E to I	Incubation time is $1/\alpha = 4$ days[1]
μ	0.2	Recovery rate	Infectious period is $1/\mu = 5$ days[1]
R_0	3.77	Basic reproduction rate (β/μ)	Estimated with a 90% confidence interval of $[3.49, 4.02]$[1]
β_0	0.754	Rate of transmission	Calculated from estimated R_0[2]
β_1	0.126	Rate of transmission after measures	Based on a 82% reduction in the reproduction rate[3]
N	38000	Population size	Avg. number of monitored households × avg. household size[4,5]
e_0	1	Initial number of exposed people	
τ	100	Threshold for the number of infected people	
δ	7	Days until preventive measures are effective	
c^M	40	Testing capacity per day of the ML	Avg. number of patients reached per day during the pilot[6]
d^M	0	Days to get test results of a ML	Equal to the turnaround time[6]
m^M	1000	Multiplier for determining p for ML	Due to the mobility, more targeted tests can be done
c^F	10	Testing capacity per day of a HF	Reflects lower access and lower quality tests compared to ML
d^F	7	Days to get test results for a HF	Based on the turnaround time[6] and delay in delivering samples
m^F	100	Multiplier for determining p for HF	Due to limited access, samples are less likely from I

[1]: (Iyaniwura et al., 2022)
[2]: follows from $R_0 = \frac{\beta}{\mu}$ (Van den Driessche, 2017)
[3]: (Flaxman et al., 2020)
[4]: (Ministry of Health, Kenya, n.d.)
[5]: (Statista, 2020)
[6]: (Praesens Care, n.d.)

5.D Sensitivity Analysis Numerical Results - Identical Regions

The results obtained in Section 5.4 are for the base case presented in Appendix 5.C and for a benefit function for investment of local health facilities with $d = 15$, *i.e.*, $b^F(x) = 15\ln(x+1)$. We now present a sensitivity analysis on the parameter d used in the benefit function for investment in local health facilities and the parameters from the SEIR model that affect the potential time reduction of mobile labs, T_Δ.

First, we consider the effect of the parameter d. Figure 5.11 shows the value of mobile labs for different values of d. We see that, regardless of the value of d, mobile labs add most value when the budget is tight. When d is small, *i.e.*, when a lot of budget is required to reduce the detection time, we can attain time reductions that are approximately twice as high compared to only investing in local health facilities. When d is large, *i.e.*, small investments in local health facilities result in significant reductions in the detection time, the value of mobile labs is smaller.

Second, we consider the effect of the transmission rate. In the base case, we used a transmission rate of $\beta_0 = 0.754$, which was determined from an estimated

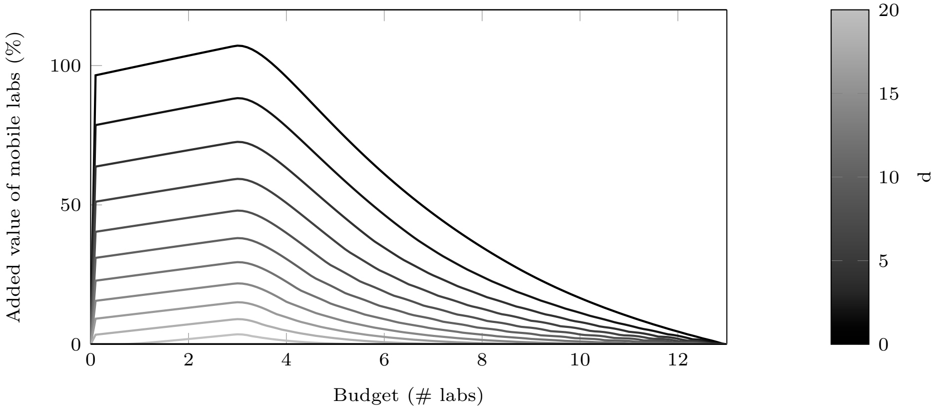


Figure 5.11: The value of mobile labs, the additional reduction in the detection time we can achieve using mobile labs, for $d \in [10, 20]$.

basic reproduction rate $R_0 = 3.77$ and a recovery rate $\mu = 0.2$ (Iyaniwura et al., 2022), using that $R_0 = \frac{\beta}{\mu}$ (Van den Driessche, 2017). Iyaniwura et al. (2022) find a 90% confidence interval for the basic reproduction rate in Kenya of $[3.49, 4.02]$. We therefore consider the case with $\beta_{0,min} = 0.698$ and $\beta_{0,max} = 0.804$, which provides a 90% confidence interval for the transmission rate.

Figure 5.12 shows the number of regions that receive mobile lab visits (black, left y-axis) and the value of mobile labs (grey, right y-axis) for the three different transmission rates. We see that mobile labs are used unless the budget is very high. The threshold after which all budget is allocated to investments is lower for higher transmission rates. For all transmission rates, mobile labs add most value when the budget is small and all regions receive mobile labs. When the transmission rate increases, the value of mobile labs decreases. This is caused by a reduction in the potential reduction in the detection time from using mobile labs. Namely, when the transmission rate increases, the disease spreads faster, reducing the detection time. Since diseases spread exponentially at the onset of an outbreak (Kermack and McKendrick, 1927), this reduces the detection time for health facilities more than for mobile lab visits.

Third, we consider the effect of the multiplier for determining the probability that a test is positive for the local health facilities. We assume less targeted tests are performed when the access to local health facilities is lower. In the base case, we used $m^F = 100$ and $m^M = 1000$.

Figure 5.13 presents the number of regions that receive mobile lab visits and the

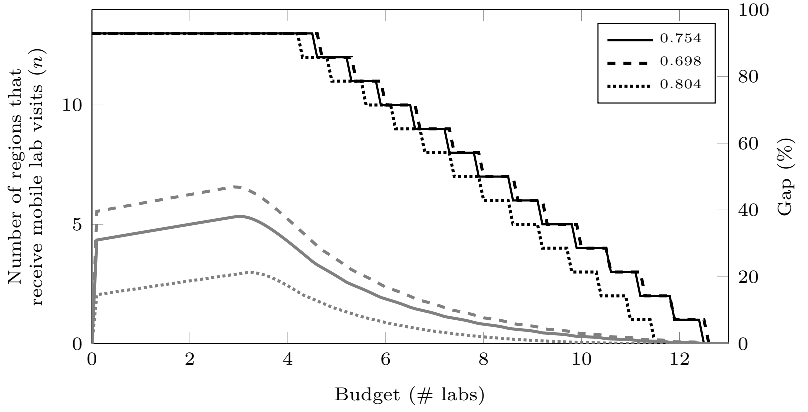


Figure 5.12: The number of regions that receive mobile lab visits (black, left y-axis) and the value of mobile labs (grey, left y-axis), *i.e.*, the additional benefit that can be obtained in the optimal solution compared to only considering investment in local health facilities for various transmission rates, β_0.

value of mobile labs for $m^F \in \{10, 50, 100, 250\}$. Similar to earlier findings, mobile labs add most value when the budget is small and they are used unless the budget is very high. We see that mobile labs add more value when the multiplier is smaller, as expected. This is caused by a decrease in the potential time reduction of mobile labs, T_Δ, since health facilities have a lower detection time. For large multipliers (small potential time reductions) the gap can become zero as is the case for $m^F = 250$. In this case investment in local health facilities will dominate While there is still a potential time reduction ($T_\Delta = 3.71$), investment in local health facilities dominates scheduling mobile lab visits in this case, *i.e.* $b^M(x) \leq b^F(x) = 15\ln(x+1)$ for all $x \in [0, C^M]$.

Next, we consider the effect of the quality of the local health facility. Figure 5.14 shows the effect of the duration to obtain the test results, d^F. In the base case, we took $d^F = 7$ days and now present the results for $d^F \in \{2, 5, 7, 10, 14\}$. A longer duration to obtain test results increases the detection time for health facilities and therefore increases the potential time reduction for mobile labs. Larger durations thus result in a higher value of mobile labs. Figure 5.15 shows the effect of the testing capacity at local health facilities, c^F. We present the results for $c^F \in \{5, 10, 15, 20\}$, where $c^F = 10$ corresponds to the base case. When local health facilities are able to execute more tests per day, they will detect the disease earlier. We see thus that the value of mobile labs increases when the number of tests decreases. For $c^F = 15$, we see that the number of regions that receive mobile lab visits first increases, and

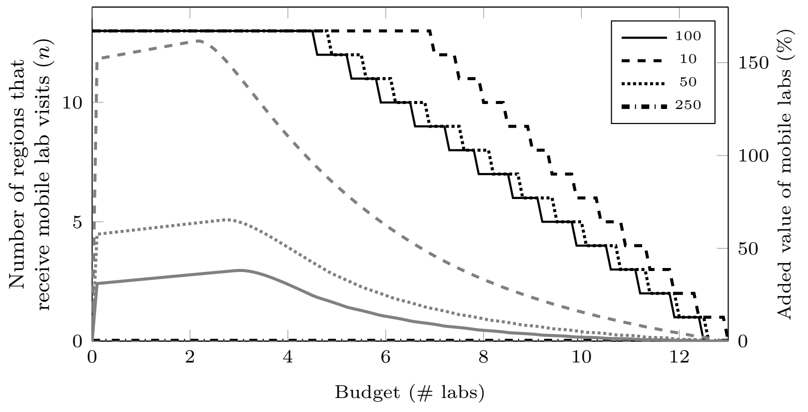


Figure 5.13: The number of regions that receive mobile lab visits (black, right y-axis) and the value of mobile labs (grey, left y-axis), *i.e.*, the additional benefit that can be obtained in the optimal solution compared to only considering investment in local health facilities. for various multipliers for local health facilities, m^F. Note that for $m^F = 250$, $n = 0$ for all budget levels and therefore the value of mobile labs is zero at all budget levels.

then decreases. For the corresponding benefit function for mobile lab visits, the benefit function for investing in local health facilities is larger for both small and large budgets and smaller for medium-sized budgets. Therefore, when the budget is small or large, (almost) all budget is assigned to investing in local health facilities. For a medium budget, (almost) all budget is assigned to mobile labs, resulting in an additional reduction in the detection time of almost 6%.

Finally, we analyse the effect of τ, the detection threshold. In the base case we assumed a threshold of $\tau = 100$ positive tests. Figure 5.16 presents the results for $\tau \in \{50, 100, 150, 200, 380\}$, where $\tau = 380$ corresponds to a positive test for 1% of the population. When the detection threshold increases, it takes longer to detect the disease. We observe that this increase in detection time is larger for local health facilities. Therefore, for increasing threshold τ, the potential time reduction for mobile labs increases. This results in a larger value of mobile labs for larger thresholds. For $\tau = 50$ positive tests, we find a value of 21%, which increases to almost 240% for $\tau = 380$.

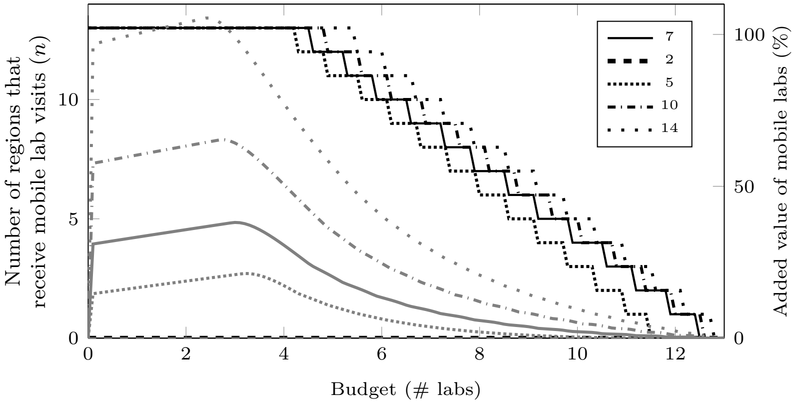


Figure 5.14: Number of regions that receive mobile lab visits (black, left y-axis) and the value of mobile labs (grey, right y-axis), *i.e.*, the additional benefit that can be obtained in the optimal solution compared to only considering investment in local health facilities, for various durations to obtain the test results for local health facilities, d^F. Note that for $d^F = 2$, $n = 0$ for all budget levels and therefore the gap is zero for all budget levels.

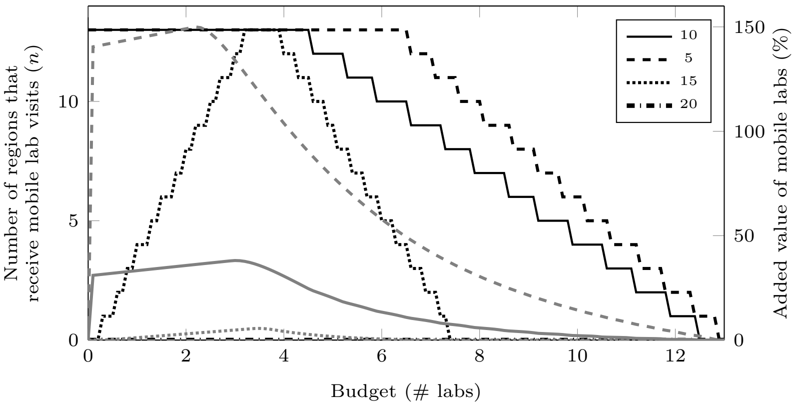


Figure 5.15: The number of regions that receive mobile lab visits (black, left y-axis) and the value of mobile labs (grey, right y-axis), *i.e.*, the additional benefit that can be obtained in the optimal solution compared to only considering investment in local health facilities, for various testing capacities at local health facilities, c^F. Note that for $c^F = 20$, $n = 0$ for all budget levels and therefore the gap is zero for all budget levels.

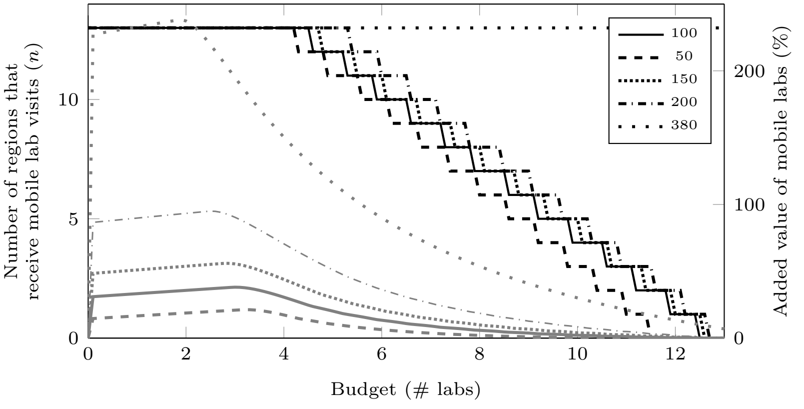


Figure 5.16: The number of regions that receive mobile lab visits (black, left y-axis) and the value of mobile labs (grey, right y-axis), *i.e.*, the additional benefit that can be obtained in the optimal solution compared to only considering investment in local health facilities, for various detection thresholds, τ.

5.E Additional Numerical Results - Non-Identical Regions

The results in Section 5.4.2 are for the base case with $d = 15$ where lower quality regions have a testing duration of $d_L^F = 10$ days and higher quality regions have a testing duration of $d_H^F = 7$ days. We now present the results for differences in the testing capacity ($c_L^F = 5$ tests versus $c_H^F = 10$ tests).

Figure 5.17 and 5.18 show the results for differences in the testing capacity. Figure 5.8 shows the percentage of the budget allocated to mobile lab visits (dotted line, right y-axis). The solid and dashed line (left y-axis), show the expected benefit in the optimal solution and the solution in which only investment in local health facilities is considered. The shaded area shows the value of mobile labs, *i.e.*, the additional benefit in the optimal solution compared to the solution in which all budget is allocated to investment in local health facilities. A darker colour corresponds to a larger value of mobile labs. Figure 5.18 shows the expected reduction for all regions, the low-resource regions, and the high-resource regions. The shaded area shows the price of fairness, *i.e.*, the percentual difference between the optimal and equitable solution in which identical regions receive the same budget and the difference in the detection time between the two types off regions is at most equal to the current difference. A darker colour corresponds to a higher price of fairness.

Similar to our findings in Section 5.4.2, we find that lower quality regions are

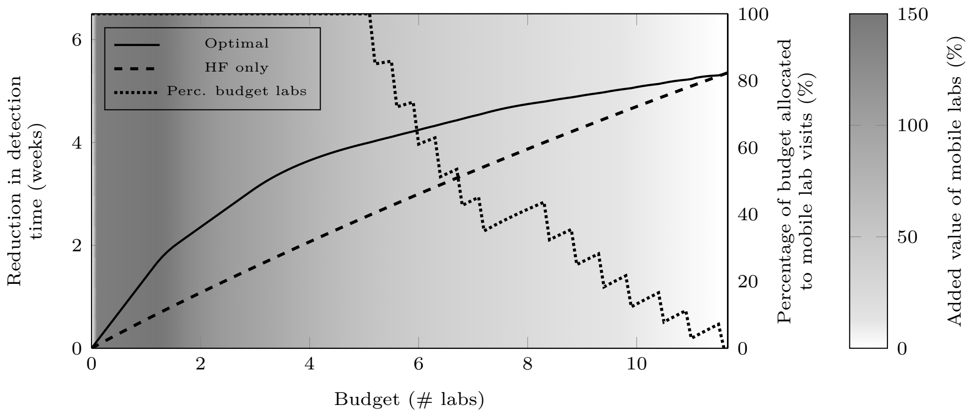


Figure 5.17: The expected reduction in the detection time in the optimal solution and the solution when all budget is invested in local health facilities (HF) (right y-axis) for $c_L^F = 5$ and $c_L^F = 10$. The dotted line (left y-axis) shows the percentage of the available budget that is allocated to mobile lab visits. The shaded area shows the value of mobile labs, *i.e.*, the additional benefit that can be obtained in the optimal solution compared to only considering investment in local health facilities.

prioritised for small budget levels. Mobile labs add most value when the budget is tight and are used unless the budget is very large. Currently, the difference in the detection time is around 1.5 weeks. We thus observe inverse inequality for low budget levels since the reduction in the detection time for low-resource regions surpasses 1.5 weeks before budget is allocated to high-resource regions. The price of fairness is less than 14% for the small budget levels.

When the budget is sufficiently high to invest in the local health facilities in some regions (for medium budget levels), investment in the local health facilities of high-resource regions is prioritised over those in low-resource regions. While in Section 5.4.2 we observed higher reductions in the detection time for high-resource regions, and thus more inequity, this is not the case for this scenario. Therefore, for medium and large budget levels, the only inequity that arises is through treating identical regions differently. The price of fairness is thus low for medium and large budget levels (at most 3.5%).

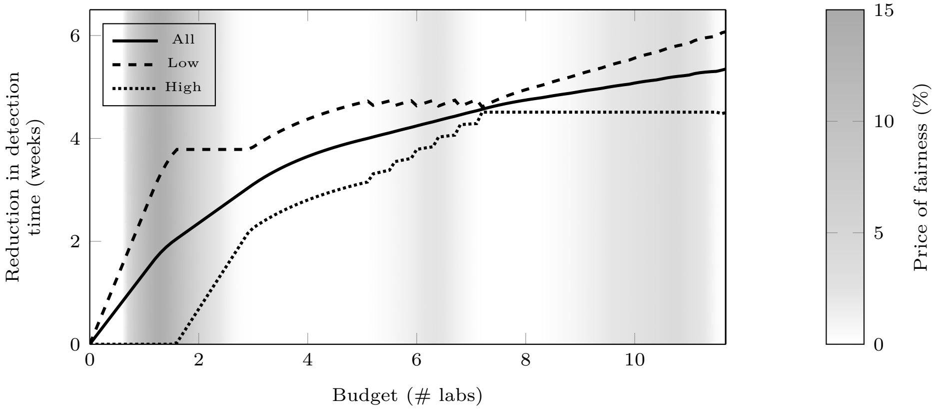


Figure 5.18: The expected reduction in the detection time for all regions, the low-resource regions, and the high-resource regions for the base case with $c_L^F = 5$ and $5_H^F = 10$. The shaded area shows the price of fairness, *i.e.*, the percentual difference between the optimal and equitable expected benefit.

Chapter 6

Conclusion and future research

In this thesis, we considered optimisation problems with applications in military and humanitarian logistics. These problems often differ in the objective and/or requirements compared to related problems in the private sector. Therefore, problems with applications in military and humanitarian logistics require tailored approaches as solution approaches in the private sector can usually not be (directly) applied.

In this chapter, we present the main findings of this thesis. Then, we discuss the practical implications and ideas for future research.

6.1 Main Findings

In Chapter 2, we considered the construction of schedules for the *Ship-to-Shore Problem*. In this problem, resources have to be transported from large amphibious warfare ships to the shore, using smaller ships and helicopters, called *connectors*. The aim is to minimise the time required to transport all resources, while adhering to various requirements. We proved that this problem is NP-hard and we formulated the problem such that coordination between the delivery of complementary resources can be imposed, *e.g.*, to ensure that personnel and their vehicles are delivered closely after each other. We presented two solution approaches, an exact and a heuristic method that mimics current practice. To evaluate their performance, we conducted experiments on both real-world and artificial instances, considering different practical constraints. Our results demonstrated that the heuristic method performed particularly well on instances with coordination requirements, which are common in practice due to the need for synchronisation in the delivery of resources. For most real-world instances, the heuristic produced solutions that were optimal, as confirmed by the exact method. In cases where significant improvements were possible, the exact method provided these within a reasonable time.

In Chapter 3, we considered the evaluation of schedules constructed for the Ship-to-Shore Problem such as those of Chapter 2. These schedules are constructed using deterministic parameters and might therefore not be robust to delays. On the other hand, the schedules constructed in Chapter 2 are made using discrete time periods which can result in room in the schedule to capture delays. We developed a simulation model that considers uncertainty in the speed of connectors, the (un)loading times of connectors, and changes in the weather conditions compared to the predictions. In this simulation model, the schedule is followed as closely as possible, as is done in practice. We analysed the effect of uncertainty on the performance of a schedule and found that significant delays can be incurred. We constructed schedules using more conservative parameters for the speed and (un)loading times of the connectors. We found that using a more conservative speed had an insignificant or even negative effect on the performance of the schedule. Namely, small reductions in the speed can result in the same schedule due to the discretisation resulting in an insignificant effect. Large reductions in the speed can result in a full additional time period that is scheduled, resulting in a negative effect as the additional room in the schedule is too large. Using more conservative (un)loading times, which were linked with the time period length, could however improve the realised makespan. Finally, we analysed the effect of rigidity in the execution of the schedule and found that being a bit less rigid in allowing connectors to depart ahead of time, can significantly reduce the realised makespan. This can however come at the risk of violating requirements regarding the coordinated delivery of resources.

In Chapter 4, we formally defined the Generalised Capacitated Resupply Problem in which we aim to find the minimum number of vehicles required to resupply locations with a given capacity, periodic demand, and resupply time. We introduced different variants of the problem based on the homogeneity or heterogeneity of the input parameters, and proved the problem to be NP-hard. For three of the variants, we provide an inapproximability result, *i.e.*, we provide a conditional lower bound on the approximation guarantee. We presented different policies that can be used to find a feasible schedule and gave corresponding approximation guarantees on the number of vehicles used in these schedules. When the locations are homogeneous in all parameters – capacity, demand rate, and resupply time – we present a policy that can solve the problem to optimality, *i.e.*, that results in a schedule using the optimal number of vehicles. When only the resupply times are homogeneous, our policy attains the lower bound. For instances with homogeneous capacities and demand rates, we present a policy with a constant-factor approximation, *i.e.*, the resulting number of vehicles at most a multiple of the lower bound. For the general problem in which all parameters are heterogeneous, we present an additive approximation policy. The problem is a generalisation of the Windows Scheduling Problem. Therefore, a natural thing would be to reduce an instance to an instance of the Windows Scheduling Problem and use an approximation algorithm for this problem. However, this reduces the feasible region as a more strict deadline would be imposed on the period during which a location must be visited. The price of imposing this stricter deadline is an open question.

In Chapter 5, we analysed the value of mobile labs in improving surveillance of infectious diseases in low- and middle-income countries. We formulated a budget allocation problem, where the budget is allocated to different regions that can either use it to invest in their local health facilities or in deploying mobile labs. The aim is to maximise the expected reduction in the detection time, which allows for imposing preventive measures earlier, reducing the impact of an outbreak. We also discussed potential equity concerns, which are important in allocating resources in public health. We analysed the problem both analytically and numerically for identical regions and for regions that differ in the current quality of their local health facilities. We found that for a tight budget, mobile labs can add significant value. Inequity can arise in different forms. Even when regions are identical, they can be treated differently as some receive budget to invest in their local health facilities and others to deploy mobile labs, resulting in different detection times. When regions differ in the current quality of their local health facilities, we found that inverse inequality can arise. Namely, regions that currently have lower quality local health facilities and thus higher detection times can be overcompensated for this lower quality, resulting in lower detection times. When we considered equitable budget allocations, we found that mobile labs can still add significant value for tight budgets.

6.2 Practical Implications

Chapter 2 shows that for practical instances of the Ship-to-Shore Problem, where complementary resources require coordinated delivery, current practices captured by our heuristic may often perform well. Our analysis shows that using the branch-and-price algorithm can provide significant improvements, especially for scenarios with minimal coordination requirements. We suggest to use the heuristic method when time is limited, as this often finds good solutions for practical instances. Since the branch-and-price algorithm is often able to find improvement relatively fast, we recommend using it to help identify quick improvements to the schedule, as long as time allows.

During the execution of a ship-to-shore schedule, delays can propagate through the schedule. If coordination between the deliveries is not critical, we recommend being less rigid in the execution of the schedule by allowing connectors to depart up to ten minutes ahead of schedule. This small adjustment has been shown to consistently reduce the realised makespan, providing more consistent reductions compared to using more conservative parameters. When the coordination between deliveries is critical and violations should be avoided as much as possible, we recommend using slightly longer time periods.

A concern for using dispersed operations is whether there is sufficient logistical capacity to sustain these dispersed units. Our policies for the Generalised Capacitated Resupply Problem in Chapter 4, although it is a simplified version of the actual resupply operation, provide a first step in assessing the operational feasibility of dispersed operations. These policies can be used to determine a feasible number of vehicles required to sustain a given operation. Furthermore, additional insights can be gained such as the impact of varying the number of dispersed locations or vehicle payload capacities.

In Chapter 5, we analysed the value of mobile labs and discussed potential equity concerns. Our findings indicate that mobile labs provide significant values, particularly when the budget is limited and the local health facilities are of low quality. However, when the budget is large, it is better to invest in the local health facilities. We therefore recommend considering mobile labs when the available budget is tight and the current quality of the local health facilities is low. We found that equity concerns can exist in the optimal solution. Namely, even when regions are identical, they may receive different budgets in the optimal budget allocation, creating differences in their detection times. When regions are different, either the low or high resource regions are prioritised, depending on the available budget. When the low resource regions are prioritised, inverse inequality can arise, where the low resource regions have lower detection times than the high resource regions. When the high resource regions are prioritised, the difference between the regions becomes larger. Inequitable budget allocations may lack the necessary support from local stakeholders that prevent them from being implemented. We found that the price of fairness, *i.e.*, the percentage difference between the optimal and equitable solution, is relatively low. Therefore, we recommend imposing equity constraints in the problem to ensure a fair allocation.

6.3 Future Research

The problems presented in this thesis offer several directions for future research.

Our simple greedy heuristic of Chapter 2 performs well in constrained instances for the Ship-to-Shore Problem. However, the exact method is able to find significant improvements for some instances. For such instances where the heuristic does not perform well, it would be interesting to develop an improved heuristic, such as a local search on the solution of the heuristic to find improvements faster compared to our exact approach. In the current approach, we set the time period length as the maximum of all (un)loading times, such that each connector can be (un)loaded in one time period. In scenarios where the travel times are relatively short, it could be useful to consider shorter time period lengths. This requires adjustments to the model to ensure connectors remain long enough at the short or sea base to (un)load.

In Chapter 3, we analysed the effect of uncertainty in the weather conditions, speeds, and (un)loading times and found that these can significantly effect the realised makespan. Instead of using less optimistic parameters, as evaluated in Chapter 3, a more robust schedule could be made by considering the uncertainty in the parameters in the construction of the schedule.

In Chapter 4, we studied the Generalised Capacitated Resupply Problem (GCRP) in which locations with a given capacity, demand rate, and resupply time have to be sustained using vehicles with a given payload. We present an additive approximation policy for instances in which all parameters are heterogeneous, however, we aim to find a constant-factor approximation. A related problem for which a constant-factor approximation algorithm exist, is the Windows Scheduling Problem. In this problem, the implicit assumption is made that the capacity is smaller than the payload of the vehicles, such that each time a location is visited, its inventory is restocked to its full capacity. We can therefore reduce an instance of GCRP to an instance of Windows Scheduling and use a constant-factor approximation for this problem. However, this requires imposing more strict deadlines on the visit frequency of the locations and it is unknown what the price of this is.

In Chapter 5, we used some assumptions regarding the functional form of the benefit function for investments in local health facilities. While we have some justification for choosing a concave benefit function, it would be valuable to (empirically) investigate the relationship between investments and the detection time of local health facilities. Additionally, it would be valuable to identify which regional characteristics affect the detection time. We considered regions that differ in the current quality of their health facilities based on the testing capacity or the testing duration, but other differences can have an influence as well. Finally, we consider a one-time investment to improve the surveillance as a first step in analysing the value of mobile labs while considering the alternative of investing in local health facilities. In practice, budget may become available over multiple time periods, which may affect the decision on how to allocate the budget.

References

Aborode, A.T., Hasan, M.M., Jain, S., Okereke, M., Adedeji, O.J., Karra-Aly, A. and Fasawe, A.S. (2021). 'Impact of poor disease surveillance system on COVID-19 response in africa: time to rethink and rebuilt'. *Clinical epidemiology and global health* 12, p. 100841.

Alban, A., Blaettchen, P., de Vries, H. and Van Wassenhove, L.N. (2022). 'Resource allocation with sigmoidal demands: Mobile healthcare units and service adoption'. *Manufacturing & Service Operations Management* 24.6, pp. 2944–2961.

Amrouss, A., El Hachemi, N., Gendreau, M. and Gendron, B. (2017). 'Real-time management of transportation disruptions in forestry'. *Computers & Operations Research* 83, pp. 95–105.

André, A.M., Lopez, A., Perkins, S., Lambert, S., Chace, L., Noudeke, N., Fall, A. and Pedalino, B. (2017). 'Frontline field epidemiology training programs as a strategy to improve disease surveillance and response'. *Emerging infectious diseases* 23.Suppl 1, S166.

Arribas, E., Cholvi, V. and Mancuso, V. (2023). 'Optimizing UAV Resupply Scheduling for Heterogeneous and Persistent Aerial Service'. *IEEE Transactions on Robotics.*

Assad, A.A. and Gass, S.I. (2011). *Profiles in operations research: pioneers and innovators.* Springer.

Bar-Noy, A. and Ladner, R.E. (2003). 'Windows scheduling problems for broadcast systems'. *SIAM Journal on Computing* 32.4, pp. 1091–1113.

Bar-Noy, A., Ladner, R.E., Tamir, T. and VanDeGrift, T. (2012). 'Windows scheduling of arbitrary-length jobs on multiple machines'. *Journal of Scheduling* 15, pp. 141–155.

Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W. and Vance, P.H. (1998). 'Branch-and-price: Column generation for solving huge integer programs'. *Operations Research* 46.3, pp. 316–329.

Behnke, M., Kirschstein, T. and Bierwirth, C. (2021). 'A column generation approach for an emission-oriented vehicle routing problem on a multigraph'. *European Journal of Operational Research* 288.3, pp. 794–809.

Boland, N., Hewitt, M., Marshall, L. and Savelsbergh, M. (2019). 'The price of discretizing time: a study in service network design'. *EURO Journal on Transportation and Logistics* 8.2, pp. 195–216.

Brandeau, M.L. (2004). 'Allocating resources to control infectious diseases'. *Operations research and health care: a handbook of methods and applications*, pp. 443–464.

Brandeau, M.L., Zaric, G.S. and Richter, A. (2003). 'Resource allocation for control of infectious diseases in multiple independent populations: beyond cost-effectiveness analysis'. *Journal of health economics* 22.4, pp. 575–598.

Bredstrom, D. and Rönnqvist, M. (2007). 'A branch and price algorithm for the combined vehicle routing and scheduling problem with synchronization constraints'. *NHH Dept. of Finance & Management Science Discussion Paper* 2007/7.

Breugem, T. (2020). *Crew Planning at Netherlands Railways: Improving Fairness, Attractiveness, and Efficiency.* PhD Thesis. ERIM PhD Series in Research in Management.

Breugem, T., Fan, Y., Gernert, A.K. and Van Wassenhove, L.N. (2024). 'Equity in Health and Humanitarian Logistics: A People-Centered Perspective'. *Production and Operations Management*, p. 10591478241248751.

Breugem, T., Wolters, T.S. and Van Wassenhove, L.N. (2023). *Visit Allocation Problems in Multi-Service Settings: Policies and Worst-Case Bounds.* Tech. rep.

Chardaire, P., McKeown, G.P., Verity-Harrison, S. and Richardson, S. (2005). 'Solving a Time-Space Network Formulation for the Convoy Movement Problem'. *Operations Research* 53.2, pp. 219–230.

Christafore Jr., R. (2017). *Generating ship-to-shore bulk fuel delivery schedules for the Marine Expeditionary Unit.* Master thesis.

Coene, S., Spieksma, F.C.R. and Woeginger, G.J. (2011). 'Charlemagne's Challenge: The Periodic Latency Problem'. *Operations Research* 59.3, pp. 674–683.

Cooke, K.L. (1967). 'Functional-differential equations: some models and perturbation problems'. *Differential equations and dynamical systems*, pp. 167–183.

Cooper, L.A., Purnell, T.S., Showell, N.N., Ibe, C.A., Crews, D.C., Gaskin, D.J., Foti, K. and Thornton, R.L. (2018). 'Progress on major public health challenges: the importance of equity'. *Public health reports* 133.1_suppl, 15S–19S.

Corpuz, M.Q., Rusnock, C.F., Valencia, V.V. and Oyama, K. (2017). ‘Medical readiness: evaluating the robustness of medical clinic staffing solutions’. *The Journal of Defense Modeling and Simulation* 14.4, pp. 407–419.

Cuesta, E.F., Andersson, H., Fagerholt, K. and Laporte, G. (2017). ‘Vessel routing with pickups and deliveries: an application to the supply of offshore oil platforms’. *Computers & Operations Research* 79, pp. 140–147.

Danielson, M.E. (2018). *Scheduling amphibious connectors to deliver multiple commodities*. Master thesis.

Debusmann Jr., B. (13th Mar. 2024). ‘How the US military plans to construct a pier and get food into Gaza’. *British Broadcasting Corporation (BBC)*. URL: `https://www.bbc.com/news/world-us-canada-68534370` (visited on 18/03/2024).

Desaulniers, G., Desrosiers, J. and Solomon, M.M. (2002). ‘Accelerating strategies in column generation methods for vehicle routing and crew scheduling problems’. *Essays and surveys in metaheuristics*. Springer, pp. 309–324.

Dhanjal-Adams, K.L., Mustin, K., Possingham, H.P. and Fuller, R.A. (2016). ‘Optimizing disturbance management for wildlife protection: the enforcement allocation problem’. *Journal of Applied Ecology* 53.4, pp. 1215–1224.

Dictorate of Land Concepts and Design (2007). ‘Land Operations 2021. Adaptive Dispersed Operations. The Force Employment Concept for Canada’s Army of Tomorrow’. *Government of Canada*.

Drexl, M. (2012). ‘Synchronization in vehicle routing—a survey of VRPs with multiple synchronization constraints’. *Transportation Science* 46.3, pp. 297–316.

Van den Driessche, P. (2017). ‘Reproduction numbers of infectious disease models’. *Infectious disease modelling* 2.3, pp. 288–303.

Duijzer, E., van Jaarsveld, W., Wallinga, J. and Dekker, R. (2018a). ‘Dose-optimal vaccine allocation over multiple populations’. *Production and Operations Management* 27.1, pp. 143–159.

Duijzer, E., van Jaarsveld, W. and Dekker, R. (2018b). ‘The benefits of combining early aspecific vaccination with later specific vaccination’. *European Journal of Operational Research* 271.2, pp. 606–619.

Enayati, S. and Özaltın, O.Y. (2020). ‘Optimal influenza vaccine distribution with equity’. *European Journal of Operational Research* 283.2, pp. 714–725.

Falconer, N. (1976). ‘On the size of convoys: an example of the methodology of leading wartime OR scientists’. *Journal of the Operational Research Society* 27.2, pp. 315–327.

Fall, C., Cappuyns, A., Faye, O., Pauwels, S., Fall, G., Dia, N., Diagne, M.M., Diagne, C.T., Niang, M., Mbengue, A. et al. (2020). 'Field evaluation of a mobile biosafety laboratory in Senegal to strengthen rapid disease outbreak response and monitoring'. *African Journal of Laboratory Medicine* 9.2, pp. 1–9.

Fernandez de La Vega, W. and Lueker, G.S. (1981). 'Bin packing can be solved within 1+ ε in linear time'. *Combinatorica* 1.4, pp. 349–355.

Flaxman, S., Mishra, S., Gandy, A., Unwin, H.J.T., Mellan, T.A., Coupland, H., Whittaker, C., Zhu, H., Berah, T., Eaton, J.W. et al. (2020). 'Estimating the effects of non-pharmaceutical interventions on COVID-19 in Europe'. *Nature* 584.7820, pp. 257–261.

Fleming, K.A., Horton, S., Wilson, M.L., Atun, R., DeStigter, K., Flanigan, J., Sayed, S., Adam, P., Aguilar, B., Andronikou, S. et al. (2021). 'The Lancet Commission on diagnostics: transforming access to diagnostics'. *The Lancet* 398.10315, pp. 1997–2050.

Ford, L.R. and Fulkerson, D.R. (1956). 'Maximal flow through a network'. *Canadian journal of Mathematics* 8, pp. 399–404.

Garey, M.R. and Johnson, D.S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman, San Fransisco.

Gensheimer, K.F., Fukuda, K., Brammer, L., Cox, N., Patriarca, P.A. and Strikas, R.A. (1999). 'Preparing for pandemic influenza: the need for enhanced surveillance.' *Emerging infectious diseases* 5.2, p. 297.

Gribkovskaia, I., Laporte, G. and Shlopak, A. (2008). 'A tabu search heuristic for a routing problem arising in servicing of offshore oil and gas platforms'. *Journal of the Operational Research Society* 59.11, pp. 1449–1459.

Grolla, A., Jones, S.M., Fernando, L., Strong, J.E., Ströher, U., Möller, P., Paweska, J.T., Burt, F., Pablo Palma, P., Sprecher, A. et al. (2011). 'The use of a mobile laboratory unit in support of patient management and epidemiological surveillance during the 2005 Marburg outbreak in Angola'. *PLoS neglected tropical diseases* 5.5, e1183.

Gu, L., Ryzhov, I.O. and Eftekhar, M. (2021). 'The facts on the ground: Evaluating humanitarian fleet management policies using simulation'. *European Journal of Operational Research* 293.2, pp. 681–702.

Harris, T. and Ross, F. (1955). *Fundamentals of a method for evaluating rail net capacities.* Tech. rep. The RAND Corporation, Santa Monica, California.

Harweg, T., Wagner, M. and Weichert, F. (2022). 'Agent-based simulation for infectious disease modelling over a period of multiple days, with application to an airport scenario'. *International Journal of Environmental Research and Public Health* 20.1, p. 545.

Hensher, M., Blizzard, L., Campbell, J., Canny, B., Zimitat, C. and Palmer, A. (2024). 'Diminishing marginal returns and sufficiency in health-care resource use: an exploratory analysis of outcomes, expenditure, and emissions'. *The Lancet Planetary Health* 8.10, e744–e753.

Hiermann, G., Puchinger, J., Ropke, S. and Hartl, R.F. (2016). 'The electric fleet size and mix vehicle routing problem with time windows and recharging stations'. *European Journal of Operational Research* 252.3, pp. 995–1018.

Ho, H.-M. and Ouaknine, J. (2015). 'The cyclic-routing UAV problem is PSPACE-complete'. *Proceeding of the 18th Internation Conference on Foundations of Software Science and Computation Structures*. Springer, pp. 328–342.

Holte, R., Mok, A., Rosier, L., Tulchinsky, I. and Varvel, D. (1989). 'The pinwheel: A real-time scheduling problem'. *Proceedings of the 22th Annual Hawaii International Conference on System Sciences*. Vol. 2, pp. 693–702.

Horne, G.E. and Irony, T.Z. (1994). 'Queueing processes and trade-offs during ship-to-shore transfer of cargo'. *Naval Research Logistics (NRL)* 41.2, pp. 137–151.

Irnich, S. and Desaulniers, G. (2005). 'Shortest path problems with resource constraints'. *Column generation*. Springer, pp. 33–65.

Iyaniwura, S.A., Rabiu, M., David, J.F. and Kong, J.D. (2022). 'The basic reproduction number of COVID-19 across Africa'. *Plos one* 17.2, e0264455.

Jacobs, T. and Longo, S. (2014). 'A new perspective on the windows scheduling problem'. *arXiv preprint arXiv:1410.7237*.

Jnitova, V., Elsawah, S. and Ryan, M. (2017). 'Review of simulation models in military workforce planning and management context'. *The Journal of Defense Modeling and Simulation* 14.4, pp. 447–463.

Karp, R.M. (1972). 'Reducibility among combinatorial problems'. *Complexity of computer computations*. Springer, pp. 85–103.

Kelly-Cirino, C.D., Nkengasong, J., Kettler, H., Tongio, I., Gay-Andrieu, F., Escadafal, C., Piot, P., Peeling, R.W., Gadde, R. and Boehme, C. (2019). 'Importance of diagnostics in epidemic and pandemic preparedness'. *BMJ global health* 4.Suppl 2, e001179.

Kermack, W.O. and McKendrick, A.G. (1927). 'A contribution to the mathematical theory of epidemics'. *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character* 115.772, pp. 700–721.

Kulshrestha, A., Lou, Y. and Yin, Y. (2014). 'Pick-up locations and bus allocation for transit-based evacuation planning with demand uncertainty'. *Journal of Advanced Transportation* 48.7, pp. 721–733.

Levinson, D. and Wu, H. (2020). 'Towards a general theory of access'. *Journal of Transport and Land Use* 13.1, pp. 129–158.

Lewis, T.P., Roder-DeWan, S., Malata, A., Ndiaye, Y. and Kruk, M.E. (2019). 'Clinical performance among recent graduates in nine low-and middle-income countries'. *Tropical Medicine & International Health* 24.5, pp. 620–635.

Liu, R., Tao, Y. and Xie, X. (2019). 'An adaptive large neighborhood search heuristic for the vehicle routing problem with time windows and synchronized visits'. *Computers & Operations Research* 101, pp. 250–262.

Maritime Warfare Centre (2019). *Handboek Surface Assault.*

Micah, A.E., Bhangdia, K., Cogswell, I.E., Lasher, D., Lidral-Porter, B., Maddison, E.R., Nguyen, T.N.N., Patel, N., Pedroza, P., Solorio, J. et al. (2023). 'Global investments in pandemic preparedness and COVID-19: development assistance and domestic spending on health between 1990 and 2026'. *The Lancet Global Health* 11.3, e385–e413.

Ministry of Health, Kenya (n.d.). *Kenya Master Health Facility Registry.* Last accessed November 4th 2024. URL: `https://kmhfr.health.go.ke/public/facilities`.

National Postal Museum (n.d.). *Postal Reforms that Transformed the Mail.* Last accessed November 27th 2024. URL: `https://postalmuseum.si.edu/exhibition/the-queen%E2%80%99s-own/postal-reforms-that-transformed-the-mail`.

Nixon, J. and Ulmann, P. (2006). 'The relationship between health care expenditure and health outcomes: evidence and caveats for a causal link'. *The European journal of health economics* 7, pp. 7–18.

Nowak, M.A. (2005). *The pickup and delivery problem with split loads.* Georgia Institute of Technology.

De Oliveira, F.N., Alem, D., Oliveira, F. and Leiras, A. (2024). 'Long-Lasting Insecticidal Net Campaigns for Malaria Control Considering Prioritization and Equity'. *Production and Operations Management*, p. 10591478241279247.

Pagonis, W.G. and Cruikshank, J.L. (1992). *Moving mountains: Lessons in leadership and logistics from the Gulf War.* Harvard Business School Press.

Palvannan, R.K. and Teow, K.L. (2012). 'Queueing for healthcare'. *Journal of medical systems* 36, pp. 541–547.

Patriksson, M. (2008). 'A survey on the continuous nonlinear resource allocation problem'. *European Journal of Operational Research* 185.1, pp. 1–46.

Peters, D.H., Garg, A., Bloom, G., Walker, D.G., Brieger, W.R. and Hafizur Rahman, M. (2008). 'Poverty and access to health care in developing countries'. *Annals of the new York Academy of Sciences* 1136.1, pp. 161–171.

Praesens Care (2024). *Praesens Care*. Last accessed June 28th 2024. URL: `https://praesens.care/`.

Praesens Care (n.d.). *Intervention study in Kenya*. Last accessed November 25th 2024. URL: `https://praesens.care/our-projects/intervention-study-kenya/`.

Racine, T. and Kobinger, G.P. (2019). 'Challenges and perspectives on the use of mobile laboratories during outbreaks and their use for vaccine evaluation'. *Human vaccines & immunotherapeutics*.

Rao, I.J. and Brandeau, M.L. (2021). 'Optimal allocation of limited vaccine to control an infectious disease: Simple analytical conditions'. *Mathematical biosciences* 337, p. 108621.

Rieck, J. and Zimmermann, J. (2010). 'A new mixed integer linear model for a rich vehicle routing problem with docking constraints'. *Annals of Operations Research* 181.1, pp. 337–358.

Scheidegger, A.P.G., Pereira, T.F., de Oliveira, M.L.M., Banerjee, A. and Montevechi, J.A.B. (2018). 'An introductory guide for hybrid simulation modelers on the primary simulation methods in industrial engineering identified through a systematic review of the literature'. *Computers & Industrial Engineering* 124, pp. 474–492.

Schneider, M., Stenger, A. and Goeke, D. (2014). 'The electric vehicle-routing problem with time windows and recharging stations'. *Transportation Science* 48.4, pp. 500–520.

Schrijver, A. (2002). 'On the history of the transportation and maximum flow problems'. *Mathematical programming* 91.3, pp. 437–445.

Simpson, G.G. (1947). 'Holarctic mammalian faunas and continental relationships during the Cenozoic'. *Geological Society of America Bulletin* 58.7, pp. 613–688.

Siswanto, N., Kusumawati, M., Baihaqy, A., Wiratno, S. and Sarker, R. (2023). 'New Approach for Evaluating Berth Allocation Procedures Using Discrete Event Simulation to Reduce Total Port Handling Costs'. *Operations and Supply Chain Management: An International Journal* 16.4, pp. 535–553.

Smith, M.J. (2015). ‘Health equity in public health: clarifying our commitment’. *Public Health Ethics* 8.2, pp. 173–184.

Statista (2020). *Average household size in Kenya as of 2019, by county.* Last accessed November 4th 2024. URL: https://www.statista.com/statistics/1225097/household-size-in-kenya-by-county/.

Strickland, C.W. (2018). *Generating efficient and robust schedules to deliver bulk fuel via amphibious connectors*. Master thesis.

Sutherland, R. (2021). *Video calls are the new penny post.* Last accessed November 25th 2024. URL: https://www.spectator.co.uk/article/video-calls-are-the-new-penny-post/.

Szymkiewicz, D. (1934). ‘Une contribution statistique à la géographie floristique’. *Acta Societatis Botanicorum Poloniae* 11.3, pp. 249–265.

Ticha, H.B., Absi, N., Feillet, D. and Quilliot, A. (2017). ‘Empirical analysis for the VRPTW with a multigraph representation for the road network’. *Computers & Operations Research* 88, pp. 103–116.

Tolstoı, A. (1930). ‘Metody nakhozhdeniya naimen’shego summovogo kilometrazha pri planirovanii perevozok v prostranstve [Russian; Methods of finding the minimal total kilometrage in cargotransportation planning in space]’. *Planirovanie Perevozok, Sbornik pervyı [Russian; Transportation Planning, Volume I], Transpechat’NKPS [TransPress of the National Commissariat of Transportation], Moscow*, pp. 23–55.

Ursavas, E. (2017). ‘A Benders decomposition approach for solving the offshore wind farm installation planning at the North Sea’. *European Journal of Operational Research* 258.2, pp. 703–714.

Van Wassenhove, L.N. (2006). ‘Humanitarian aid logistics: supply chain management in high gear’. *Journal of the Operational Research Society* 57.5, pp. 475–489.

Vazirani, V.V. (2001). *Approximation Algorithms*. Springer.

Vhilen, N (2022). *How can the marines learn from the Falklands War? War on the rocks.* Last accessed February 17th 2024. URL: https://warontherocks.com/2022/11/how-can-marines-learn-from-the-falklands-war/.

Villena, K. (2019). *Ship to shore transportprobleem tijdens amfibische operaties.* Bachelor thesis.

De Vries, H., Swinkels, L.E. and Van Wassenhove, L.N. (2021). ‘Site visit frequency policies for mobile family planning services’. *Production and Operations Management* 30.12, pp. 4522–4540.

Wagenvoort, M. (2023). *Ship-to-Shore Artificial Instances*. Zenodo. URL: https://doi.org/10.5281/zenodo.10216198.

Wagenvoort, M., Bouman, P., van Ee, M., Lamballais Tessensohn, T. and Postek, K. (2025a). 'Exact and heuristic approaches for the ship-to-shore problem'. *European Journal of Operational Research* 320.1, pp. 115–131.

Wagenvoort, M., Bouman, P., van Ee, M. and Malone, K.M. (2025b). 'Evaluating Ship-to-Shore Schedules using Simulation'. *Journal of Defense, Modeling and Simulation (accepted for publication)*.

Wagenvoort, M., van Ee, M., Bouman, P. and Malone, K.M. (2023). 'Simple Policies for Capacitated Resupply Problems (Short Paper)'. *23rd Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2023)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik, pp. 18:1–18:6.

Wilcox, R.R., Erceg-Hurn, D.M., Clark, F. and Carlson, M. (2014). 'Comparing two independent groups via the lower and upper quantiles'. *Journal of Statistical Computation and Simulation* 84.7, pp. 1543–1551.

World Health Organisation (2022a). *Early warning alert and response in emergencies: an operational guide*. World Health Organisation.

World Health Organisation (2022b). *Universal health coverage (UHC)*. Last accessed May 12th 2023. URL: https://www.who.int/news-room/fact-sheets/detail/universal-health-coverage-(uhc).

Worsley-Tonks, K.E., Bender, J.B., Deem, S.L., Ferguson, A.W., Fèvre, E.M., Martins, D.J., Muloi, D.M., Murray, S., Mutinda, M., Ogada, D. et al. (2022). 'Strengthening global health security by improving disease surveillance in remote rural areas of low-income and middle-income countries'. *The Lancet Global Health* 10.4, e579–e584.

Xiao, Y., Zhao, Q., Kaku, I. and Xu, Y. (2012). 'Development of a fuel consumption optimization model for the capacitated vehicle routing problem'. *Computers & Operations Research* 39.7, pp. 1419–1431.

Yadav, H., Shah, D., Sayed, S., Horton, S. and Schroeder, L.F. (2021). 'Availability of essential diagnostics in ten low-income and middle-income countries: results from national health facility surveys'. *The Lancet Global Health* 9.11, e1553–e1560.

Zachariadis, E.E., Tarantilis, C.D. and Kiranoudis, C.T. (2016). 'The vehicle routing problem with simultaneous pick-ups and deliveries and two-dimensional loading constraints'. *European Journal of Operational Research* 251.2, pp. 369–386.

Zhao, X., Ji, K., Xu, P., Qian, W.-w., Ren, G. and Shan, X.-n. (2020). 'A round-trip bus evacuation model with scheduling and routing planning'. *Transportation Research Part A: Policy and Practice* 137, pp. 285–300.

Summary

Military and humanitarian logistics present unique challenges that distinguish them from logistical problems in the private sector. While logistical challenges in the private sector often focus on cost-minimisation or profit-maximisation, military and humanitarian logistics focus on the effective and efficient use of scarce resources, often in unpredictable and challenging environments. The problems often involve unique constraints that are not present in other domains, such as the need for coordination and equity. Consequently, military and humanitarian logistics problems require tailored solution approaches. In this thesis, we study various military and humanitarian optimisation problems.

In Chapter 2 and 3, we consider the *Ship-to-Shore Problem*. In this problem, resources must be transported from large amphibious warfare ships to the shore using smaller ships and helicopters, called *connectors*. The aim is to transport the resources as soon as possible, such that they can be used on land. A transportation schedule must comply with various constraints, such as coordination of complementary resources.

In Chapter 2, we present two solution approaches to construct a schedule for this problem, an exact approach and a simple heuristic that mimics current practice. We analyse the performance of these models on both real-world and artificial instances. Our findings show that the simple greedy heuristic performs well on the practical instances involving significant coordination requirements. For less constrained instances, the exact algorithm is able to find significant improvements in reasonable time. This suggests that, in practice, the greedy heuristic can be used and can be supported by the exact branch-and-price algorithm when time allows.

In Chapter 3, we simulate the execution of deterministically constructed schedules for the Ship-to-Shore Problem. We consider uncertainty in the (un)loading times and travel times of the connectors and follow the schedule as closely as possible to replicate current practice. We analyse the effect of using more conservative parameters for the

construction of a schedule as well as the effect of being less rigid in the execution of a schedule. We find that being less rigid in the execution significantly improves the realised makespan. Even when connectors are allowed to depart ten minutes ahead of time, significantly better results can be achieved. However, this improvement comes at the cost of potentially violating coordination requirements.

In Chapter 4, we consider the *Generalised Capacitated Resupply Problem*. In this problem, a set of dispersedly located units with a given periodic demand rate, capacity, and resupply time must be sustained, *i.e.*, they must be resupplied on time such that they do not run out of stock. The aim is to find the minimum number of vehicles with a given payload required to sustain all locations. We present policies with corresponding approximation ratios, *i.e.*, guarantees on the quality of the solution, that can be used to find a feasible resupply schedule for different versions of the problem.

In Chapter 5, we analyse the value of mobile laboratories (labs) for the surveillance of infectious diseases in low- and middle-income countries compared to investments in local health facilities. Mobile labs are vans equipped with high-quality diagnostic equipment designed to perform on the spot testing of infectious diseases. Due to their mobility, they can be deployed at different locations, temporarily improving disease surveillance. We consider a budget allocation problem where a limited budget must be allocated to different regions, either for investments in local health facilities or for deploying mobile labs. We find that mobile labs can add significant value, especially when the budget is limited and the regions have low-quality local health facilities. We evaluate equity concerns and find that the price of fairness is relatively low.

Nederlandse Samenvatting (Summary in Dutch)

Militaire en humanitaire logistiek brengen unieke uitdagingen met zich mee die zich onderscheiden van logistieke problemen in de private sector. Waar logistieke vraagstukken in de private sector vaak gericht zijn op kostenminimalisatie of winstmaximalisatie, draait het in militaire en humanitaire logistiek om het effectieve en efficiënte gebruik van schaarse middelen, vaak in onvoorspelbare en veeleisende omgevingen. Deze problemen gaan gepaard met specifieke randvoorwaarden die in andere domeinen afwezig zijn, zoals de noodzaak tot coördinatie en een eerlijk verdeling van middelen. Hierdoor vereisen militaire en humanitaire logistieke vraagstukken aangepaste oplossingsmethoden. In dit proefschrift bestuderen we verschillende optimalisatieproblemen binnen militaire en humanitaire logistiek.

In hoofdstuk 2 en 3 behandelen we het *Schip-tot-Strand Probleem (Ship-to-Shore Problem)*. Dit probleem betreft het transport van middelen vanaf grote amfibische transportschepen naar de kust met behulp van kleinere schepen en helikopters, ook wel *connectoren* genoemd. Het doel is om deze middelen zo snel mogelijk aan land te brengen, zodat ze daar kunnen worden ingezet. De planning moet voldoen aan verschillende voorwaarden, waaronder de coördinatie van complementaire middelen.

In hoofdstuk 2 presenteren we twee oplossingsmethoden voor dit planningsprobleem, een exacte methode en een eenvoudige heuristiek die de huidige praktijk nabootst. We analyseren de prestaties van deze methoden op zowel instanties uit de praktijk als op artificiële instanties. Onze bevindingen laten zien dat de eenvoudige heuristiek goed presteert op instanties uit de prakrijk waarin veel coördinatie vereist is. Voor minder beperkende instanties kan het exacte algoritme binnen redelijke tijd significante verbeteringen opleveren. Dit suggereert dat in de praktijk de heuristiek effectief ingezet kan worden en kan worden ondersteund door de exacte methode

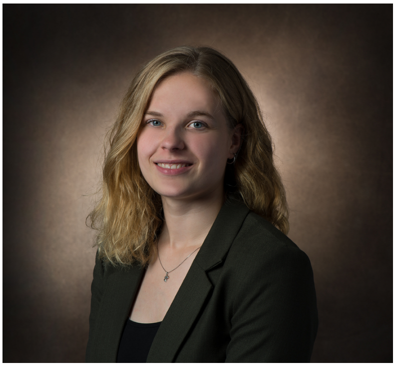
wanneer de beschikbare tijd dit toelaat.

In hoofdstuk 3 simuleren we de uitvoering van deterministisch geconstrueerde planningen voor het Schip-tot-Strand Probleem. Hierbij houden we rekening met onzekerheid in de laad- en lostijden en de reistijden van de connectoren, terwijl we het oorspronkelijke plan zo nauwkeurig mogelijk volgen om de praktijk realistisch te modelleren. We analyseren zowel het effect van de parameters bij het opstellen van een planning als de impact van een flexibelere uitvoering ervan. De resultaten laten zien dat een minder rigide uitvoering van een planning de duur van de transportoperatie aanzienlijk kan verkorten. Zelfs het toestaan van een vertrek van connectoren tot tien minuten eerder kan al tot substantiële verbeteringen leiden. Deze efficiëntiewinst gaat echter gepaard met het risico dat bepaalde randvoorwaarden met betrekking tot de coördinatie van middelen worden geschonden.

In hoofdstuk 4 behandelen we het *Gegeneraliseerde Capaciteitsbegrensde Bevoorradingsprobleem (Generalised Capacitated Resupply Problem).* Dit probleem richt zich op het bevoorraden van verspreid geplaatste eenheden met een bepaalde periodieke vraag, capaciteit en bevoorradingstijd. Dit betekent dat elke locatie tijdig bevoorraad moet worden om te voorkomen dat voorraden opraken. Het doel is om het minimale aantal voertuigen met een gegeven laadvermogen te bepalen waarmee de locaties kunnen worden onderhouden. We presenteren *regimes* met bijbehorende benaderingsratio's, d.w.z., garanties op de kwaliteit van de oplossing, die kunnen worden gebruikt om een haalbare bevoorradingsplanning op te stellen voor verschillende varianten van het probleem.

In Hoofdstuk 5 analyseren we de waarde van mobiele laboratoria voor het monitoren van infectieziekten in lage- and middeninkomenslanden, in vergelijking met investeringen in lokale gezondheidsfaciliteiten. Mobiele laboratoria zijn voertuigen uitgerust met geavanceerde diagnostische apparatuur waarmee ter plaatse testen op infectieziekten kunnen worden uitgevoerd. Dankzij hun mobiliteit kunnen ze op verschillende locaties worden ingezet, wat een tijdelijke verbetering van het toezicht op infectieziekten mogelijk maakt. We bestuderen een budgettoewijzingsprobleem waarin een beperkt budget moet worden verdeeld over verschillende regio's, hetzij voor investeringen in lokale gezondheidszorg, hetzij voor de inzet van mobiele laboratoria. Onze bevindingen tonen aan dat mobiele laboratoria van grote waarde kunnen zijn, met name wanneer het budget zeer beperkt is en de bestaande gezondheidszorginfrastructuur van lage kwaliteit is. Daarnaast evalueren we mogelijke zorgen over de eerlijkheid van een budgettoewijzing en concluderen we dat de kosten van het vereisen van een eerlijkere verdeling van het budget relatief laag zijn.

About the author



Mette Wagenvoort studied Econometrics and Operations Research with a specialisation in Operations Research and Quantitative Logistics (cum laude), and Economics and Business Economics with a specialisation in Urban, Port and Transport Economics from which she graduated in 2019. In 2020 she graduated from the master Econometrics and Management Science with a specialisation in Operations Research and Quantitative Logistics (cum laude). In 2020 she started working as a PhD candidate at the Erasmus University Rotterdam on a joint project with TNO and the Netherlands Defence Academy.

During her PhD, Mette worked on problems in military logistics using different optimisation techniques, such as branch-and-price and approximation algorithms. In the spring of 2023, she went on a research visit to INSEAD, Fontainebleau. Here she collaborated with dr. Iman Parsa and Prof.dr. Luk Van Wassenhove, where she analysed the value of mobile laboratories to improve surveillance of infectious diseases.

Portfolio

Peer-reviewed publications

Azadeh, S.S., van der Zee, J. and Wagenvoort, M. (2022). 'Choice-driven service network design for an integrated fixed line and demand responsive mobility system'. *Transportation Research Part A: Policy and Practice* 166, pp. 557–574.

Wagenvoort, M., Bouman, P., van Ee, M., Lamballais Tessensohn, T. and Postek, K. (2025a). 'Exact and heuristic approaches for the ship-to-shore problem'. *European Journal of Operational Research* 320.1, pp. 115–131.

Wagenvoort, M., Bouman, P., van Ee, M. and Malone, K.M. (2025b). 'Evaluating Ship-to-Shore Schedules using Simulation'. *Journal of Defense, Modeling and Simulation (accepted for publication).*

Wagenvoort, M., van Ee, M., Bouman, P. and Malone, K.M. (2023). 'Simple Policies for Capacitated Resupply Problems (Short Paper)'. *23rd Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2023).* Schloss-Dagstuhl-Leibniz Zentrum für Informatik, pp. 18:1–18:6.

Research visits

Research visit to INSEAD, Humanitarian Research Group, to collaborate with dr. Iman Parsa and prof.dr. Luk Van Wassenhove (March-June 2023).

PhD courses
Algorithms and Complexity at LNMB
Academic Writing at ERIM
Basic Didactics and Groupdynamics for PhDs at Risbo
Convex Analysis for Optimisation at LNMB
Hacking the Code of Publishing at ERIM
Integer Programming Methods at LNMB
Networks and Polyhedra at LNMB
Networks and Semidefinite Programming at LNMB
Open Science and Scientific Integrity at ERIM
Presentation Skills at ERIM
Randomised Algorithms at LNMB
Robust Optimisation at LNMB
Thesis Supervision at Risbo
Topics in the Philosophy of Science at ERIM

Teaching activities
Lecturer for the course Quantitative Methods for Logistics for the bachelor *Econometrics and Operations Research*, 2025
Teaching assistant for the course Simulation for the bachelor *Econometrics and Operations Research*, 2020-2025
Teaching assistant for the course Academic Skills for the bachelor *Econometrics and Operations Research*, 2021-2022
Teaching assistant for the minor Port Management & Maritime Logistics at the *Erasmus University Rotterdam*, 2022
Supervisor of 4 bachelor theses for the bachelor *Econometrics and Operations Research*, 2024-2025
Supervisor of 2 master theses for the master *Econometrics and Management Science*, 2024-2025
Co-reader of 7 master theses for the master *Econometrics and Management Science*, 2022-2025

Conference presentations

EURO 2021, Athens, Greece (online)
NATO OR&A 2021, Madrid, Spain (online)
INFORMS 2021, Anaheim, California (online)
EURO 2022, Helsinki, Finland
NATO OR&A 2022, Copenhagen, Denmark
ATMOS 2023 Amsterdam, The Netherlands
POMS 2024 Minneapolis, Minnesota, US
EURO 2024 Copenhagen, Denmark
INFORMS 2024 Seattle, Washington, US
LNMB 2025 Soesterberg, The Netherlands

Other presentations

Econometric Institute OR Seminar, July 2021, Rotterdam, The Netherlands (online)
TNO Lunch colloquium Planning & Scheduling, July 2021, The Hague, The Netherlands (online)
Econometric Institute PhD Conference, January 2024, Rotterdam, The Netherlands
Econometric Institute OR Seminar, July 2024, Rotterdam, The Netherlands

Referee activities for international journals

European Journal of Operations Research
Omega
Transportation Planning and Technology

The ERIM PhD Series

The ERIM PhD Series contains PhD dissertations in the field of Research in Management defended at Erasmus University Rotterdam and supervised by senior researchers affiliated to the Erasmus Research Institute of Management (ERIM). Dissertations in the ERIM PhD Series are available in full text through: https://pure.eur.nl. ERIM is the joint research institute of the Rotterdam School of Management (RSM) and the Erasmus School of Economics (ESE) at the Erasmus University Rotterdam (EUR).

Dissertations in the last four years

Abdelwahed, A., *Optimizing Sustainable Transit Bus Networks in Smart Cities*, Supervisors: Prof. W. Ketter, Dr P. van den Berg & Dr T. Brandt, EPS-2022-549-LIS

Alkema, J., *READY, SET, GO(AL)! New Directions in Goal-Setting Research*, Supervisors: Prof. H.G.H. van Dierendonck & Prof. S.R. Giessner, ESP-2022-555-ORG

Alves, R.A.T.R.M, *Information Transmission in Finance: Essays on Commodity Markets, Sustainable Investing, and Social Networks*, Supervisors: Prof. M.A. van Dijk & Dr M. Szymanowska, EPS-2021-532-LIS

Anantavrasilp, S., *Essays on Ownership Structures, Corporate Finance Policies and Financial Reporting Decisions*, Supervisors: Prof. A. de Jong & Prof. P.G.J. Roosenboom, EPS-2021-516-F&E

Ansarin, M., *The Economic Consequences of Electricity Pricing in the Renewable Energy Era*, Supervisors: Prof. W. Ketter & Dr Y. Ghiassi-Farrokhfal, EPS-2021-528-LIS

Aydin Gökgöz, Z. *Mobile Consumers and Applications: Essays on Mobile Marketing*, Supervisors: Prof. G.H. van Bruggen & Dr B. Ataman, EPS-2021-519-MKT

Azadeh, K., *Robotized Warehouses: Design and Performance Analysis*, Supervisors: Prof. M.B.M. de Koster & Prof. D. Roy, EPS-2021-515-LIS

Badenhausen, K., *IoT – Inducing Organizational Transformation?* Supervisors: Prof. R.A. Zuidwijk & Dr M. Stevens, ESP-2022-559-LIS

Balocco, F.A.M., *Asymmetric information in programmatic advertising: Three studies on adverse selection, mechanism choices, and fee structures*, Supervisors: Prof. T. LI & Prof. E. van heck, EPS-2023-565-LIS

Bilgin, B., *Visionary Leadership and the Pursuit of Organizational Visions*, Supervisors: Prof. D.L. van Knippenberg & Dr I.J. Hoever, EPS-2024-583-ORG

Breet, S., *A Network Perspective on Corporate Entrepreneurship: How workplace relationships influence entrepreneurial behavior*, Supervisors: Prof. J.J.P. Jansen, Prof. J. Dul & Dr L. Glaser, EPS-2022-545-S&E

Bunders, D., *Gigs of their Own: can platform cooperatives become resilient*? Supervisors: Prof. T. de Moor, Prof. A. Akkerman & Prof. P. Dykstra, EPS-2024-580-ORG

Chung, Y.S., *Valorizing Innovation through Imaginativeness in Business Venturing*, Supervisors: Prof. PH.B.F. Franses & Prof. H.P.G. Pennings, EPS-2022-561-MKT

Dijkstra, N., *Statistical Modeling Approaches for Behavioral and Medical Sciences*, Supervisors: Prof. P.J.F Groenen, Prof. H. Tiemeier & Prof. A.R Thurik, EPS-2022-539-S&E

Duijsters, J., *Change in Inter-Organizational Relationship Portfolios and Social Networks in the Context of Corporate Venturing*, Prof. V.J.A. van de Vrande, Prof. J.J.P. Jansen & Prof. P.P.M.A.R. Heugens, EPS-2024-504-S&E

Fu, G., *Agency Problems in the Mutual Fund Industry*, Supervisors: Prof. M.J.C.M. Verbeek & Dr E. Genc, EPS-2022-526-F&A

Genc, A., *The Triad of Innovation: Exploring management innovation, business model innovation, and ownership in unlocking firms' innovative potential*, Supervisors: Prof. H. W. Volberda & Prof. J. S. Sidhu, EPS-2024-579-S&E

Geradts, T. *Moving Beyond the Why and What Question: How Corporations Achieve Sustainable Development*, Supervisors: Prof. J.J.P. Jansen, Prof. J.P. Cornelissen & Prof. A. Nijhof, EPS-2021-521-S&E

Giessen, M. van der, *Co-creating Safety and Security: Essays on bridging disparate needs and requirements to foster safety and security*, Supervisors: Prof. G. Jacobs, Prof. J.P. Cornelissen & Prof. P.S. Bayerl, EPS-2022-542-ORG

Giudici, A., *Cooperation, Reliability, and Matching in Inland Freight Transport*, Supervisors: Prof. R.A. Zuidwijk, Prof. C. Thielen & Dr T. Lu, EPS-2022-533-LIS

Gunadi, M.P., *Essays on Consumers and Numbers*, Supervisors: Prof. S. Puntoni & Dr B. Van Den Bergh, EPS-2022-558-MKT

Harter, C., *Vulnerability through vertical collaboration in transportation: A complex networks approach*, Supervisors: Prof. R.A. Zuidwijk & Dr O.R. Koppius, EPS-2023-560-LIS

Hartleb, J., *Public Transport and Passengers: Optimization Models that Consider Travel Demand,* Supervisors: Prof. D. Huisman, Prof. M. Friedrich & Dr M.E. Schmidt, EPS-2021-535-LIS

Hoogendoorn, Y.N., *Vehicle Routing with Varying Levels of Demand Information*, Supervisors: Prof. A.P.M. Wagelmans, Prof. R. Dekker & Dr R. Spliet, EPS-2024-578-LIS

Hoogervorst, R., *Improving the Scheduling and Rescheduling of Rolling Stock: Solution Methods and Extensions*, Supervisors: Prof. D. Huisman & Dr T.A.B. Dollevoet, EPS-2021-534-LIS

Hulsen, M., *Wait for others? Social and intertemporal preferences in allocation of healthcare resources*, Supervisors: Prof. K.I.M. Rohde & Prof. N.J.A. van Exel, EPS-2023-556-MKT

Jellema, S.F., *Brace for Impact: Good intentions, unintended consequences, and the role of performative micro-processes in organized efforts for societal change*, Supervisors: Prof. J.P. Cornelissen & Prof. T.H. Reus, EPS-2024-587-ORG

Kalvapelle, S.G., *Breaking the Conduit: A Relational Approach to Communication in Management and Entrepreneurship*, Supervisors: Prof. J.P. Cornelissen & Prof. P.P.M.A.R. Heugens, EPS-2023-575-ORG

Kanellopoulos, I., *Navigating Digital Platform Challenges: Three Studies on Blockchain-based Platform Competition, Platform Subsidies for Non-Fungible Token Creation, and Ghosting in the Dating Market*, Supervisors: Prof. T. Li & Prof. H.W.G.M. van Heck, EPS-2024-585-LIS

Karaca, U., *Privacy in Resource Allocation Problems*, Supervisors: Prof. Ş.İ. Birbil & Prof. A.P.M. Wagelmans, EPS-2023-567-LIS

Koritarov, V.D., *The Integration of Crisis Communication and Regulatory Focus: Deconstructing and Optimizing the Corporate Message*, Promotors: Prof. C.B.M. van Riel, Dr G.A.J.M. Berens & Prof. P. Desmet, EPS-2021-522-ORG

Korman, B., *Leader-Subordinate Relations: The Good, the Bad and the Paradoxical*, Promotors: S.R. Giessner & Prof. C. Tröster, EPS-2021-511-ORG

Lam, H.H.W., *Lonely-ship:The emergence and experience of leader loneliness*, Supervisors: Prof. S.R. Giessner, Prof. M. Shemla & Dr M.D. Werner, EPS-2023-525-ORG

Legierse, W., *The Timing and Pricing of Initial Public Offerings: Evidence from the Low Countries*, Supervisors: Prof. A. de Jong & Prof. P.G.J. Roosenboom, EPS-2022-543-F&A

Leung, Y.K., *The Mind with a Touch of Madness? Mental health and well-being of entrepreneurs*, Supervisors: Prof. A.R. Thurik & Prof. I.H.A. Franken, EPS-2022-546-S&E

Li, Wei., *Competition in the Retail Market of Consumer Packaged Goods*, Supervisors: Prof. D. Fok & Prof. Ph.H.B.F. Franses, EPS-2021-503-MKT

Liu, M., Essays on Financial Disclosure and Innovation, Supervisors: Prof. J.P.M. Suijs & Dr P.Y.E. Leung, EPS-2023-582-F&A

Lieshout, R. van, *Integration, Decentralization and Self-Organization Towards Better Public Transport*, Supervisors: Prof. D. Huisman, Dr P.C. Bouman & Dr.ir J.M. van den Akker, EPS-2022-547-LIS

Liu, W., *An indigenous perspective on institutions for sustainable business in China*, Supervisors: Prof. P.P.M.A.R. Heugens & Dr. F.H. Wijen, EPS-2023-568-S&E

Mazzola, F., *Externalities in Economics and Finance: Essays on spillover effects and economic decisions*, Supervisors: Prof. W.B. Wagner & Prof. D.G.J. Bongaerts, EPS-2023-573-F&A

Mobini Dehkordi, Z., *Supply Chain Coordination with Separating and Pooling Contracts*, Supervisors: Prof. A.P.M Wagelmans, Dr W. van den Heuvel, EPS-2022-554-LIS

Musa, SM., *Making a Life on the Margins: An ethnographic account from Kutupalong*, Supervisors: Prof. P.P.M.A.R. Heugens & Prof. L. Berchicci, EPS-2024-586-S&E

Nikulina, A., *Interorganizational Governance in Projects: Contracts and collaboration as alignment mechanisms*, Supervisors: Prof. J.Y.F. Wynstra & Prof. L. Volker, EPS-2021-523- LIS

Novales Uriarte, A., *Thriving with Digitized Products: How Firms Leverage their Generative Capacity via Experimentation, Learning, and Collaboration*, Supervisors: Prof. H.W.G.M. van Heck & Prof. M.Mocker, EPS-2022-544-LIS

Orlandi, I., *Unringing the stigma bell: Investigating informational and social mechanisms behind boards of directors' appointments*, Supervisors: Prof. P.M.A.R. Heugens & Prof. V. F Misangyi, EPS-2023-564-S&E

Osrouti, F. El, *Norms versus Organizational Practices: Essays in the context of innovation, standardization, and religion*, Supervisors: Prof. H.J. de Vries & Prof H.W. Volberda, EPS-2024570-LIS

Pasparakis, A.M., *Leader-Follower Relationships in Technologically Advanced Operations*, Supervisors: Prof. M.B.M. de Koster & Dr J. de Vries, EPS-2023-571-LIS

Paul, J., *Online Grocery Operations in Omni-channel Retailing: Opportunities and Challenges*, Supervisors: Prof. M.B.M de Koster & Dr N.A.H. Agatz, EPS-2022-541-LIS

Paundra, J., *The Search for Alternatives to Private Vehicles The multi-perspective investigation of transportation mode choice preference based on policy, ICT development, and psychology*, Supervisors: Prof. W. Ketter, Dr J. van Dalen & Dr L. Rook, EPS-2023-496-LIS

Pocchiari, M., *Managing Successful and Resilient Shared-Interest Communities: The Role of Digitization Technologies and Disruptive Events*, Supervisors: Prof. G.H. van Bruggen & Dr J.M.T. Roos, EPS-2022-552-MKT

Ralcheva, A., *Crowdfunding: A Disruptive Innovation in Entrepreneurial Finance?* Supervisors: Prof. P.G.J. Roosenboom & Dr T. Lambert, EPS-2023-581-F&A

Ramezan Zadeh, M.T., *How Firms Cope with Digital Revolution: Essays on Managerial and Organizational Cognition*, Supervisors: Prof. H.W. Volberda & Prof. J.P. Cornelissen, EPS-2021-508-S&E

Ratara, C., *Behavioural and Neural Evidence for Processes Underlying Biases in Decision-Making*, Supervisors: Prof. A. Smidts & Dr M.A.S. Boksem, EPS-2022-548-MKT

Sluga, A., *Hour of Judgment: On judgment, decision making, and problem solving under accountability*, Supervisors: Prof. F.G.H. Hartmann & Dr M.A.S. Boksem, EPS-2021-520-F&A

Slob, E., *Integrating Genetics into Economics*, Supervisors: Prof. A.R. Thurik, Prof. P.J.F. Groenen & Dr C.A. Rietveld, EPS-2021-517-S&E

Speer, S.P.H., *The (Dis)Honest and (Un)Fair Brain: Investigating the Neural Underpinnings of Moral Decisions*, Supervisors: Prof. A. Smidts & Dr M.A.S. Boksem, EPS-2021-537-MKT

Stet, V.C., *(In)flexibility in Power Markets with Supply from Variable Renewable Sources*, Supervisors: Prof. J.T.J. Smit & Dr R. Huisman, EPS-2021-527-F&A

Stirnkorb, S., *Changes in the Information Landscape and Capital Market Communication*, Supervisors: Prof. E. Peek & Prof. M. van Rinsum, EPS-2021-536-F&A

Tierean, S.H., *Mind the Gap: The role of psychic distance and supplier's reputation in international buyer-supplier relationships*, Supervisors: Prof. C.B.M. van Riel, Prof. G. van Bruggen & Dr G.A.J.M. Berens, EPS-2022-551-ORG

Truong, H.M., *The Effect of Posted Prices on Sequential Auctions in B2B Multi-channel Markets*, Supervisors: Prof. H.W.G.M. van Heck, Prof. W. Ketter & Prof. A. Gupta, EPS-2021-539-LIS

Turturea, R., *Overcoming Resource Constraints: The Role of Creative Resourcing and Equity Crowdfunding in Financing Entrepreneurial Ventures,* Supervisors: Prof. P.P.M.A.R Heugens, Prof. J.J.P. Jansen & Dr I. Verheuil, EPS-2019-472-S&E

Vafa Arani, H., *Creating Shared Value: An operations and supply chain management perspective*, Supervisors: Prof. M.B.M. de Koster & Dr H. de Vries, EPS-2023-576-LIS

Waltré, E., *Leading for Performance in Adversity: Managing Failure, Negative Emotions, and Self-Threats*, Supervisors: Prof. D.L. van Knippenberg & Dr H.M.S. Dietz, EPS-2022-513-ORG

Wismans, A., *Behavioural Insights from the COVID-19 Pandemic Studies on Compliance, Vaccination, and Entrepreneurship*, Supervisors: Prof. A.R. Thurik & Prof. I.H.A. Franken, EPS-2023-562-S&E

Xiao, J., *Coordination & Control in Contemporary Organizations*, Supervisors: Prof. J. Cornelissen & Prof. D.A. Stam, EPS-2021-531-LIS

Yalcin, G., *Consumers in the Age of AI: Understanding Reactions Towards Algorithms and Humans in Marketing Research*, Supervisors: Prof. S. Puntoni & Dr A. Klesse, EPS-2022-550-MKT

Yang, A., *Corporate Bond Markets: Investor Preferences and Intermediary Frictions*, Supervisors: Prof. P. Verwijmeren & Prof. S. van Bekkum, EPS-2022-553-F&A

Zhang, Q., *Making sense of sensitivity in the workplace: Coping with contextual information in innovation and social networks*, Supervisors: Prof. D. Stam & Dr S. Tasselli, EPS-2023-574-LIS

Zhu, S., *Spare Parts Demand Forecasting and Inventory Management: Contributions to Intermittent Demand Forecasting, Installed Base Information and Shutdown Maintenance*, Supervisors: Prof. R. Dekker & Dr W.L. van Jaarsveld, EPS-2021-538-LIS

Zhou C., *Exploring The Role of Context and Interpretative Dynamics in Large-Scale Cross-Cultural Collaborations*, Supervisors: Prof. T. Simons & Dr M.D. Werner, EPS-2023-572-ORG

Zon, M. van, *Cost Allocation in Collaborative Transportation*, Supervisors: Prof. A.P.M. Wagelmans, Dr R. Spliet & Dr W. van den Heuvel, EPS-2021-530-LIS

Military and humanitarian logistics present unique challenges that distinguish them from logistical problems in the private sector. While logistical challenges in the private sector often focus on cost-minimisation or profit-maximisation, military and humanitarian logistics focus on the efficient and effective use of scarce resources, often in unpredictable and challenging environments. The problems often involve unique constraints that are not present in other domains, such as the need for coordination and equity. Consequently, military and humanitarian logistics problems differ significantly from those in the private sector, requiring tailored solution approaches.

In this thesis, we consider various military and humanitarian optimisation problems. In Chapter 2 and 3, we consider the Ship-to-Shore Problem, in which resources have to be transported from the ship to the shore using smaller ships and helicopters, called connectors, such that the duration of the transportation of these resources is minimised. A feasible schedule for the Ship-to-Shore Problem has to adhere to various constraints that require coordination between the pick-up and delivery of the resources by the connectors. In Chapter 4, we consider the Generalised Capacitated Resupply Problem. In this problem, locations with a given periodic demand, capacity, and resupply time have to be resupplied by vehicles from a central depot. The aim is to determine the minimum number of required vehicles to ensure none of the locations runs out of stock. We present simple policies with a corresponding approximation guarantee. In Chapter 5, we analyse the value of mobile laboratories for the surveillance of infectious diseases in low- and middle-income countries compared to investments in local health facilities. We consider a budget allocation problem where a limited budget must be allocated to different regions, either for investments in local health facilities or for deploying mobile laboratories. We also discuss potential equity concerns.

Research in Business and Management
ERIM Full-Time PhD
Erasmus School of Economics

Erasmus Research Institute of Management
Erasmus University Rotterdam
Burgemeester Oudlaan 50
Mandeville (T) Building
3062 PA Rotterdam, The Netherlands

P.O. Box 1738
3000 DR Rotterdam, The Netherlands
T: +31 10 408 1182
E: info@erim.eur.nl
W: www.erim.eur.nl
www.eur.nl/ese