SOLVING THE BARGE ROUTING AND SCHEDULING PROBLEM WITH A HYBRID METAHEURISTIC

A Master Thesis Conducted at the Cofano Software Solutions

DWARAGANATH PRABAKARAN

May 2020

Industrial Engineering & Management

Production & Logistics Management

University of Twente, Enschede, the Netherlands

University Supervisors

Dr. Eduardo Lalla-Ruiz

Dr. Martijn Mes

Company Supervisor

Ir. Leon de Vries

Preface

This thesis report is written to fulfil my graduation requirement for the master study in Industrial Engineering and Management at the University of Twente, Enschede. It documents my research work during my study, from February 2018 until May 2020. These two years have taught me so much and made me look at my future with hope and confidence. I would like to take this opportunity to thank all the people who made this possible.

First and foremost, I would like to express my deepest gratitude to my thesis supervisor, Dr. Eduardo Lalla-Ruiz, for allowing me to start the thesis under his guidance. He shaped me to be a better researcher throughout the journey of my thesis. His guidance has been an inspiration to push my limits. I would also like to thank my second supervisor, Dr. Martijn Mes, for his inputs and critical feedback on my research work.

I would like to thank Cofano software solution for providing me with the opportunity to do my master graduation assignment at their company. I would like to extend my sincere gratitude to my external supervisor from Cofano Ir. Leon de Vries, for his support throughout the research. I would also like to thank Roland Westerduin for his guidance in answering all my clarifications and helping me with the data to perform the research.

In truth, I could not have achieved this without a strong support group. I thank all my friends who stand by me throughout my happy and tough times. I appreciate your kindness and support throughout these two years, without which I would not have managed to come this far.

My final thanks are to my parents who have been my pillar of support throughout my life. Thanks for your care and effort in bringing me up to be a better individual.

Dwaraganath Prabakaran
May 2020

Management summary

Cofano software solution is considering to develop a barge routing and scheduling system that will be used for planning of containers on barges and decide the barge schedule between terminals based on the container demand. The routing and scheduling system should be effective and efficient compared to the current manual planning concerning the operation cost of barges. This research focuses on using optimizing techniques to develop an algorithm for the routing and scheduling of barges. The organization is also interested in investigating the transshipment opportunities where containers are dropped in intermediate terminals to be later transported to their destinations by different barges. The transshipment enables better utilization of the barges and better consolidations of the containers.

We start the research by investigating the existing problems in routing and scheduling of the barges and identifying the type of research problem to be solved. The systematic literature review identifies the problem to be a pick-up and delivery problem with transshipment opportunities. After analysing different solution approaches discussed in the literature, the decision was made to use a metaheuristic solution approach for designing the algorithm. The solution approach makes use of a randomized greedy search procedure for solution construction and an adaptive large neighbourhood search for solution improvement to solve the pick-up and delivery problem. The algorithm uses different destruction and repair strategies to create and analyse different neighbour solutions in order to find a better solution. As the deliverable of this research, a MIP (Mixed Integer Programming) model and an algorithm that uses hybrid metaheuristic to solve the pick-up and delivery problem is designed and implemented. The functioning of the mathematical model and the algorithm is validated using the different testing approaches. The mathematical model is used to evaluate and compare the performance of the proposed solution algorithm in terms of solution quality.

The results of the performance of the algorithm on the randomly generated test instances show that the algorithm can always achieve the solution which is equal to or better than the best integer solution from CPLEX. For the small test instance with 5 container request and without transshipment terminal, both the algorithm and MIP model were able to attain a gap of 0%. For the largest problem instances with 30 container requests and one transshipment terminal, the gap between the best integer solution from CPLEX and the best solution from algorithm reaches -83.5%. However, the gap between the lower bound (LB) and the best algorithm solution was about 76%. In comparison, the gap between LB and the CPLEX was still 95%, indicating that the algorithm was able to converge close to the optimality compared to the CPLEX solution. The LB and the CPLEX solutions were found for a maximum run time of 2 hours for large size instances with 20-30 container request with an average gap value of 90%. In comparison, the average gap between the LB and the algorithm for large size instances considered was about 72% indicating that the algorithm was able to achieve better solution compared to the CPLEX even for larger problem instances with less time.
When comparing the performance of the single start and multi start-scenarios of the algorithm, the gap between the best solution for singe start and multi-start scenarios were

found to be 1%. Whereas, the gap between the average solutions from single-start and multi-start scenarios were found to be 12%, indicating that the multi-start procedure performing consistently better than the single-start procedure.
Experimentation with real-life data instances for scenarios considering the transshipment and without transshipment options are performed using the algorithm. The test data is generated from a barge service route. The results indicate that the algorithm was able to generate solutions for real-life instances.

The following inferences are made based on the results reported in this thesis:

- The proposed metaheuristic solution approach is efficient in achieving a better quality of solution compared to the MIP model, even with lesser computation time.
- The multi-start procedure used in the algorithm ensures reliable results for the test instances compared to the single start procedure.
- Identification of transshipment and consolidation opportunities during container transportation leads to improvement in the solution cost.
- Consideration of transshipment opportunities complicates the routing and scheduling problem by increasing the solution options to be explored. The time taken to solve the problem with transshipment opportunity is enormous.
- The tradeoff for the transshipment operation is between the travel cost of barges to perform direct shipment and the additional container handling cost at the transshipment terminals to consolidate and transport the containers using transshipment strategy.

Given the results of the research, we provide the following recommendations for Cofano software solutions:

- The implementation of the solution from the algorithm needs synchronization with the solution to problems such as berth allocation problem, container stacking problem and terminal restrictions, etc. Hence, Cofano needs to appropriately address the coordination of the planning system with the integration of solutions from the above-discussed problems to provide a realistic and comprehensive solution to the barge scheduling problem.
- The restriction to consider only specific container requests that contains less number of containers for transshipment during the planning is recommended. Transshipment is advantageous when the benefit from additional handling cost at transshipment terminal is more than the benefit from direct shipment. The container requests with less number of containers are easy to be handled at transshipment terminal, and the cost to handle is also less compared to request with more number of containers. Size of the pick-up and delivery problem with transshipment is reduced when a limited number of container requests are checked for transshipment opportunities as the run time required for iterating only these specific container requests for transshipment is less.
- It is also recommended to use the solution algorithm for fitting new container requests into the existing barge schedule and create an adjusted schedule that does not deviate more from the existing barge schedule.

Future research:

- The future demand forecast should be used to investigate the potential terminals to be considered as transshipment terminals or hub terminals in the service network using the barge planning algorithm.
- Different lengths of planning horizon should be considered, and their impact on the consolidation opportunities of container request should be analysed using the planning algorithm.
- This research investigates the less explored area of transshipment using barges in inland transportation. The consideration of transshipment opportunities for intermodal transportation (using different modes such as barges, trucks and trains) needs to be investigated in future.

Table of Contents

Preface i

Management summary ii

Definitions iv

Chapter 1: Introduction 1

1.1. Company and stakeholders description 2

1.2. Problem description 3

1.2.1. Problem cluster 4

1.2.2. Core problem 5

1.3. The objective of the research 5

1.3.1. Research goal 5

1.3.2. Scope and limitations 6

1.4. Research questions 6

1.5. Research approach 8

1.5.1. Research deliverables 8

1.6. Conclusion 9

Chapter 2: Problem Context 10

2.1. Container demand in the Northwestern European region 10

2.2. Service network of barge operator 11

2.2.1. Process flow for customer request 12

2.2.2. Problems in routing and scheduling of barges 13

2.3. Research problem to be solved 15

2.4. Business and technical requirements 16

2.5. Conclusion 16

Chapter 3: Literature Review 17

3.1. Identifying the category of the research problem 17

3.1.1. Levels of planning decisions 17

3.1.2. Types of shipping problems 18

3.1.3. Different routing and scheduling problems in liner shipping 19

3.1.4. Identification of Problem category. 20

3.2. Classification of the available literature for routing and scheduling problems 20

3.2.1. Nature of demand 23

3.2.2. Mode of transportation 23

3.2.3. Fleet composition 24

3.2.4. Time dimension 24
3.2.5. Transshipment scenarios considered 25
3.2.6. Objective functions 27
3.3. Analysis of the solution methods used in literature 27
3.3.1. Exact solution approaches discussed in the literature 27
3.3.2. Heuristic solution approach discussed in the literature 28
3.4. Performance evaluation techniques used in literature 30
3.5. Selection of solution approach 30
3.6. Related works 31
3.7. Conceptual framework 32
3.8. Conclusion 33
Chapter 4: Design and Implementation of Solution Approach 34
4.1. Pick-up and delivery problem with transshipment 34
4.2. Assumptions considered for modelling 36
4.3. Graphical representation of the PDPT 36
4.4. Example for graphical representation of the PDPT 38
4.5. Mathematical model 42
4.6. Heuristic solution approach 47
4.6.1. Solution representation 49
4.6.2. Operations performed by the algorithm 50
4.6.3. Phase one heuristic 54
4.6.4. Phase two heuristic 58
4.6.5. Selecting the degree of destruction 62
4.6.6. Adaptive weight adjustment 63
4.6.7. Stopping criteria 64
4.7. Conclusion 64
Chapter 5: Validation of Solution Approach 65
5.1. Content of the research deliverables 65
5.1.1. Input data sheet template 65
5.1.2. Mathematical model 65
5.1.3. Algorithm (Two-phase heuristic solution approach) 65
5.2. Evaluation of the solution approach 66
5.2.1. Evaluating the functionality and working of the solution approach 66
5.2.2. Evaluating the performance of the solution approach 66
5.2.3. Data generation 67
5.2.4. KPI for comparing the performance 69

5.2.5. Discussion 71
5.3. Performance of heuristics solution approach on the instances with real-life demand data 71
5.3.1. Experiment design 72
5.3.2. Data generation 72
5.3.3. Experiment results 73
5.3.4. Discussion 74
5.4. Conclusion 75
Chapter 6: Analysis and Recommendations 76
6.1. Analysing the performance of the multi-start algorithm 76
6.2. Analyzing the convergence of the solution search space 78
6.3. Analyzing the benefits of transshipment opportunities 82
6.4. Recommendations from the research 87
6.5. Conclusion 88
Chapter 7: Conclusion 89
7.1. Future Scope: 90
Bibliography 91
Appendix 96
Appendix A 96
Appendix B 101
Appendix C 102
Appendix D 103
Appendix E 107
Appendix F 109

Definitions

Port	A location with a harbour or access to navigable water where ships load or unload
Port terminal	Part of port dedicated to activities such as cargo loading and unloading. There can be multiple terminals in a port that are close to each other
Inland terminal	An intermediate terminal connecting seaport and inland destinations. They act as transshipment points of sea cargo to inland destinations with connections to the road, train, and barges
Hinterland transportation	The movement of containers from a port terminal to the inland terminal (inbound) and vice-versa (outbound) through different transportation means
Inland transportation	Transportation of containers by ships (barges) via inland waterways (such as canals, rivers, and lakes) between inland terminals and seaports terminals
Transshipment	Transshipment is the process of shipment of goods or containers to one or more intermediate location followed by delivery to the final destination by different barges
Barge	A long flat-bottomed boat for carrying freight on canals and rivers, either under its power or towed by another
Container	A large metal box of a standard design and size used for the transport of goods by road, train, sea, or air
TEU - Twenty-foot Equivalent Unit	It refers to the standard unit for describing the capacity of unit cargo. One TEU is a container with a length of 20 feet
Barge operating system	The app handles data related to the barge schedule regarding the terminals to visit and containers to be loaded and unloaded at each loading and unloading location
Terminal operating system	The app handles data related to the terminal operation. The schedule for barges that are visiting the terminal, containers to be handed in the terminals and gate moves of a terminal
Planning system	Application responsible for the planning of the container on the barge schedule

Chapter 1: Introduction

The first chapter of the thesis introduces the research conducted at Cofano software solution. This chapter helps to understand the need, motivation and the objective of the research. A brief introduction about the global container supply chain and the hinterland container supply chain is provided first. The different stakeholders considered in this thesis are introduced in Section 1.1. Following this, a brief description of the problem and identification of the core problem is provided in Section 1.2. Later, the objective of the research is explained along with the scope and limitations in Section 1.3. Finally, the research questions are explained in Section 1.4, followed by the research approach in Section 1.5.

International trade has paved the way for global economic growth. Maritime transportation has been the centre of international trade. Around 80% of the volumes of goods exchanged in the world are transported via sea (UNCTAD, 2008). The term linear shipping refers to the transportation of cargo with the help of the large ships operating in scheduled routes between different ports. The liner shipping at the international level facilitates the transferring of goods at a lower cost and greater energy efficiency than any other mode of transportation.

Injection of the standard containers was made during 1955 by Malcom P. McLean, a trucking entrepreneur from North Carolina, USA. He bought a steamship company with the idea of transporting entire truck trailers along with their cargo inside. It was much simpler and quicker to have one container that could be lifted from a vehicle directly on to a ship without unloading its contents. Use of containers improved the efficiency of intermodal transportation (World shipping council, 2019). Due to the capability of transporting a large volume of containerized cargo, the liner container ship can be identified as the most efficient transportation mode for handling containerized cargo. As the world economy is growing, and due to an increase in globalization, liner shipping companies are dealing with significant growth in the volume of containers transported every year.

Containerized cargo is bought to the deep-sea terminals by the ocean carriers. The ocean carriers refer to large ships carrying containers that travel through international waters connecting different ports in different countries. The capacities of these ships have grown with years. Modern vessels can handle up to 20,000 TEU (e.g., OOCL Hong Kong - 21,413 TEUs, Madrid Maersk - 20,568 TEUs). (Network, 2019). The term foreland refers to the seaside of the port, and hinterland refers to the land side of the port. The deep seaport terminals are port terminals that handle the import and export cargo between the foreland and the hinterland. The import container refers to the containers that flow from the foreland to the hinterland. The import containers are brought to the seaport by the ocean carriers and are carried from the seaport to the inland terminals, followed by the last leg delivery to the customer location or warehouse. The container transportation from the port to the hinterland is performed by different modes of transportation like trains, trucks, and barges. The export containers refer to the containers that flow from the hinterland to the foreland. The export containers originate from the customer location or warehouse and are transported to the inland terminal through trucks. The barges or trains collect these containers and transport them to the deep-sea terminals or port terminals. The cargo is then fed to the foreland to be exported to various countries by ocean carriers. The flow of containers in a global container supply chain is represented in Figure 1. The empty container is bought to the customer location or warehouse to load the export cargo, and the empty containers are also returned to the empty container depot once the import cargo is unloaded at the customer location. The empty container flow is not included in Figure 1.

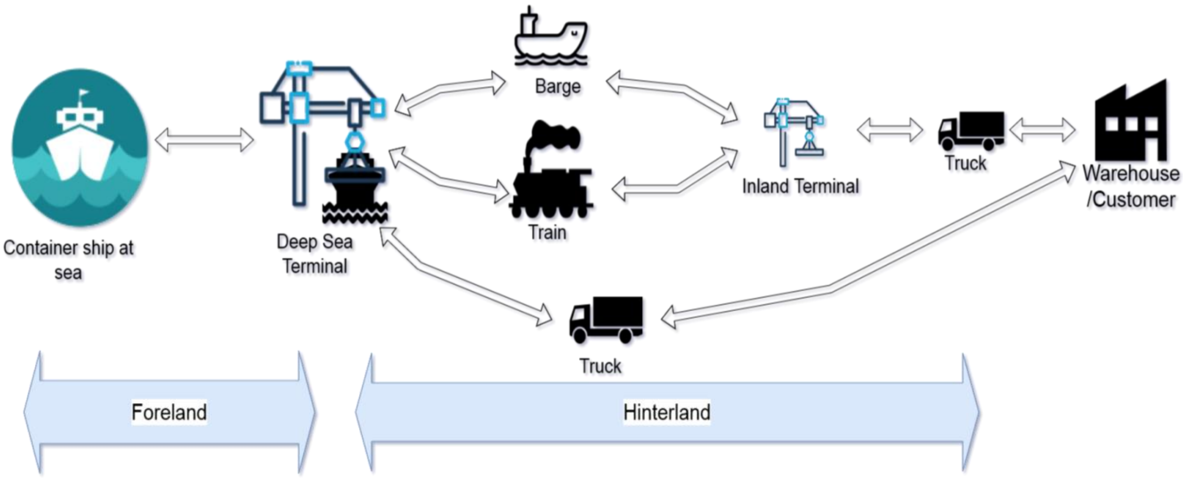

Figure 1: Container flow supply chain

1.1. Company and stakeholders description

This research focuses on the transportation of containers in the hinterland region by inland transportation (barge). The hinterland service network and container transportation of a client of Cofano Software Solutions is considered in this research. Different stakeholders who are involved in the hinterland container supply chain and the interaction between these stakeholders are introduced in this section.

Cofano Software Solutions is a software company based in Sliedrecht and Enschede. They offer web-based business software that gives the user direct insight into all relevant logistics and process information. Cofano provides several flexible standard products in the field of QHSE management and transport & logistics; perfected by years of experience in the process and maritime industry. Cofano aims to provide logistics solutions both effectively and efficiently as possible to the wishes of their partners. Cofano Software Solutions strives to provide shipping companies with schedules on routes that enable transshipment operations at terminals between the origin and destination of the containers in the hinterland region.

Barge operator is an independent logistics solutions provider who operates multimodal transports for handling cargo in the European Unionbarge operator connects seaports in the ARA (Antwerp, Rotterdam, and Amsterdam) region with the European hinterland with service routes connecting up to Basel (Switzerland). Based on the type of load, time constraints, and the destination, the barge operator creates the barge schedule to transport the containers.

Other essential stakeholders for hinterland container transportation are the terminal operators. The terminal includes both the port and inland terminals. The difference is that the inland terminals are located far away from the port area or the sea. The port terminals are located near the sea (mouth of the river), providing the seagoing vessels access to visit the port for loading and unloading the containers. The export containers loaded in an inland terminal need to be transported to multiple seaport terminals. Similarly, the set of import containers stacked in a port terminal has to be transported to one or more inland terminals by barges that operate in the hinterland region.

Last but not least, relevant stakeholders are the customers. Customers are businesses who request transportation of a container between terminals. The request can be either for export container request or an import container request or an empty container request. The empty containers are shipped from an empty container depot to an inland or port terminal. The customer requests are satisfied by the barge operator by shipping them to the destination before an agreed due date.

1.2. Problem description

Barge operater operates barges that transport containers between different locations as specified in the customer request with the help of different transportation modes available with them. Barge operator owns and operates barges that are used for container transportation in the hinterland waterways. The planning for the barges and scheduling of containers on these barges are handled in the barge planning system. The planning system receives the container request from a customer to transport a container from an origin terminal to a destination terminal. The planner operating the planning system is responsible for the creation of the barge schedule and allocating the containers to the barge schedule. The planning system is not an automated system. The planners manually plan the barge schedule based on the container request. Barge operators makes use of different logistic support software's that are provided by Cofano in order to support their logistic operation. One of the Cofano software is the Barge Operating System which enables barge operator to track and monitor their barge performance. The Cofano's terminal operating system is also installed in several terminals.

There is a two-way information transfer happening between the software of Cofano and that of barge operator. The Barge Operating System feeds the planning system with the data concerning the movement and position of barges with respect to different terminals. In return, the schedule of different terminals to be visited by the barges is provided to Barge Operating System by the planning system. Similarly, the information about the containers that should be loaded/ unloaded in each terminal is shared with the terminal operating system from the planning system. The data regarding the container availability at terminals and terminal operations such as gate moves and bookings are shared to the planning system from terminal operating system. The planners make use of the information from the Barge Operating System and terminal operating system that are shared to planning system to create the barge schedule in the planning system. This manual planning is done with the support of information from Cofano's software.

The planners have to do the scheduling process manually within the limited time available for the planning. A massive number of solution options needs to be evaluated manually before selecting a schedule because of the manual planning done for the barge operation. Due to this reason, the schedule chosen by manual planning might not be an optimal plan for the routing and scheduling problem. Moreover, there is a policy of a fixed number of weekly trips that are available between regions, and it is not efficient to operate the barges at regular intervals, even during the off-peak seasons. There is a need for improving the planning of barge schedules based on the demand of the containers rather than having a fixed schedule for every week and trying to fit the container demand to the available schedule. There is also a need for improvement in the planning process due to the high number of containers handled by barges. Another reason for the need for improvement in the planning process is due to the emerging competitors in barge transportation in Northwestern Europe. The planners need to consider the consolidation opportunity of containers and transshipment opportunities for containers that are available throughout the service network to improve the efficiency of the barge schedule. The term transshipment refers to the shipment of goods or containers to one or more intermediate destinations following its way to the final destination. The trend towards transshipment ports and the use of a hub and spoke network for inland waterways has been investigated by many new barge operators. Hence, for cost-effective, sustainable operation and to remain competitive in the transportation business, it is necessary for the barge operators to investigate the above trends as well as to improve the planning process by automating them. Cofano software solution is interested in studying the scheduling system and improving it. This improved scheduling system should incorporate the daily container demand along with analyzing the consolidation and transshipment opportunities and generate a barge schedule for a shorter planning horizon. The solution approach should focus on improving the barge capacity utilization and total cost reduction while creating an enhanced feasible solution that performs better than the current schedules concerning total cost.

1.2.1. Problem cluster

From the list of problems identified in the above analysis of the planning process, we construct the problem cluster, as shown in Figure 2. The root cause of the action problem is identified by performing a study of the action problem that was provided by the company (i.e., reducing the barge operating cost by improving the routing and scheduling). The issues identified are arranged according to the cause and effect relationship to find the core problem that needs to be focused on this thesis research. There could be high operation cost for barges when the plan is prepared to handle containers without consolidation and transshipment opportunities. There is additional complexity in the planning problem caused by the consideration of transshipment opportunities and consolidation options as they increase the number of the solution to be analyzed. The difficulty in solving the planning problem is due to the limited time available for the planners to prepare the schedule manually.

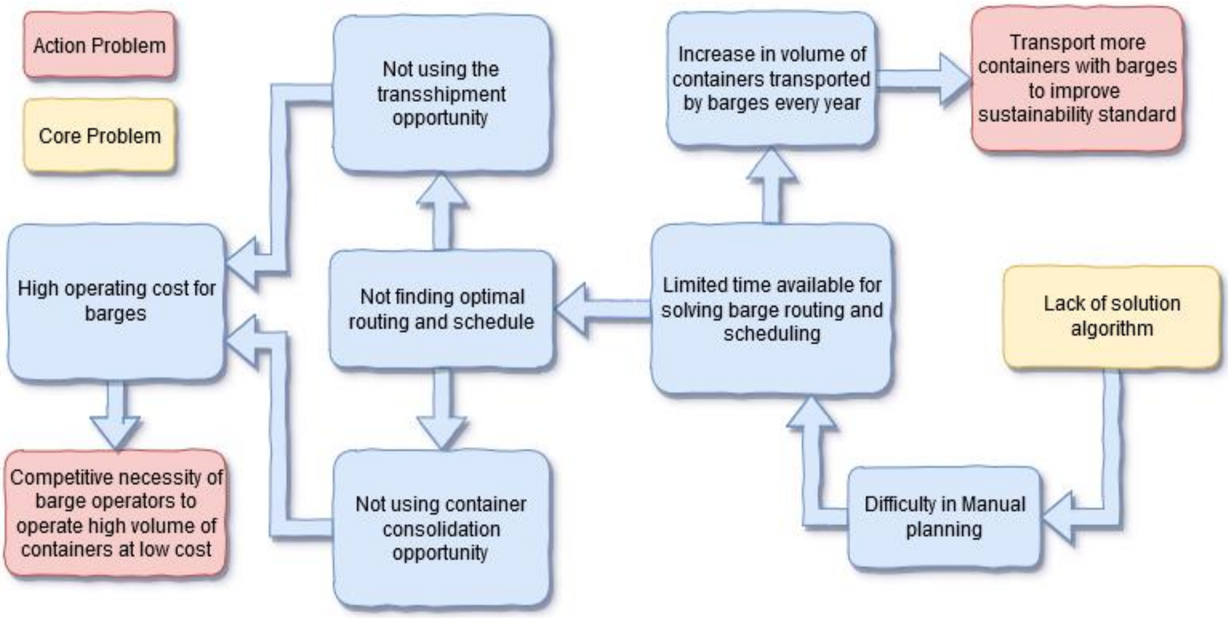

Figure 2: Problem cluster

1.2.2. Core problem

After analyzing the problem cluster, it is evident that the manual planning is stressful considering the vast data and a large number of a possible solution that might arise in the routing and scheduling problem. There is a lack of decision support systems for the planners to explore the enormous solution opportunities available in the complex service network. Hence, to make it easier for the planners to explore the vast solution options and derive an improved cost solution, we decided upon creating an algorithm for analyzing this extensive solution options to determine the best routing and scheduling solution in the given time.

The lack of scheduling algorithms for barge routing and scheduling problem is identified as the core problem from the problem cluster. This research intends to develop an algorithm to find the best feasible solution for the barge routing and scheduling problem. The algorithm will help the planners for creating an improved plan by identifying the transshipment and container consolidation opportunities within the limited time available for planning.

1.3. The objective of the research

In order to stay focused on the main objective of the research, a goal statement is identified. This section provides the research goal with the scope and limitations that are considered for the research.

1.3.1. Research goal

The goal of this research is to use optimization techniques to create a routing and scheduling algorithm for container transportation. The opportunities for transshipment and container consolidation are analyzed in the process, to benefit from economies of scale.

1.3.2. Scope and limitations

The problem of optimizing the routing and scheduling of container transportation in the entire service network is very complicated. Considering the time and the amount of work required for the level of a master thesis, it is decided to narrow the scope of the problem to limit the focus of this thesis to the following:

Scope:

- The scope is confined only to the barge operation in the hinterland area, based on the data obtained from past container transportation requests in the ARA region. Hence, the list of terminals, service networks, and container types will be used from the same data set to develop and evaluate the routing and scheduling algorithm.
- The research investigates the transshipment opportunities in the hinterland transportation service network.
- The research focuses only on the problem of cost-based optimization of transporting a container request from their origin to a destination. The performance related to automating the planning process and human operation time that is saved due to the automation is out of scope.

Limitations:

- Terminal operations that are not related to barges will not be considered in the scope of the research. (e.g., container stacking, crane and queue scheduling)
- Empty container repositioning and reuse is an emerging optimization area in barge routing and scheduling. This area will not be included in this research. Both the empty container request and the full container request will be treated as the same.
- The barge operators may operate intermodal transportation, and there can be various modes of alternatives available to transport a container from an origin to destination. The research is limited only to barge transportation mode.

1.4. Research questions

The core problem identified is solved systematically by answering the main research question defined for this research. The series of research sub-questions are defined later to answer the main research question. The main research question corresponding to the goal of the research is as follows:

How can optimization techniques be used for improving the routing and scheduling of barges in the transportation network?

The optimization techniques enable in finding the optimal routing and scheduling solution to the problem. An algorithm that identifies the best solution in the given time using the optimization techniques should be developed as a part of the solution approach. There are five research sub-questions framed to answer the main research question, as explained below.

1. Analyzing the current situation of the problem context

The first set of research question focuses on the ongoing operation of the inland transportation network. The different stakeholders who are involved throughout the inland transportation and their interaction are studied. The roles of planners and the planning process for the routing and scheduling operations are analyzed. The KPIs that are used for the performance measures along with the technical and business requirements for the solution approach is identified.

RQ 1: How is the barge routing and scheduling system working in the existing service network design?

a. *Who are the stakeholders involved in the service network?*
b. *How does the current booking and scheduling process of the inland transportation network work?*
c. *What are the complexities of the current planning process?*
d. *What are the KPIs, technical and business requirements for the solution to the barge routing and scheduling problem?*

2. Literature review and analysis

The second set of research question deals with the identification and understanding of the solution approach available in current works of literature to the barge routing and scheduling problem. Literature that gives insight into the different hinterland operation problems and creation of service network design are analyzed. The approach to a solution method and its implementation are also explained.

RQ 2: What have been proposed in the literature for solving the barge routing and scheduling problem?

a. *What are the different approaches for solving the barge routing and scheduling problem discussed in various works of literature?*
b. *What solution approach is suitable for solving barge routing and scheduling problem?*
c. *What are the advantages and disadvantages of the solution approach considered?*

3. Implementation of the solution approach

The next research question deals with the implementation of the solution approach. The algorithm for solving the routing and scheduling problem is designed as a result of answering the research question.

RQ 3: How should the solution approach be implemented for the barge routing and scheduling problem?

a. *What are the KPI measures that should be captured to measure the performance of the solution?*
b. *What routing and scheduling strategies should be considered to design the solution approach?*

4. Experimentation and evaluation

Once the Implementation phase is completed, the assessment of the solution approach and its performance is validated. This is followed by the experiment design to identify the different experimental setup and performing the experiments.

RQ4: How does the solution approach perform for different scenarios for the existing service network?

a. *How do we validate/ evaluate the performance of the solution approach?*
b. *What are the different scenarios and experimental setup that need to be considered to analyze the solution approach?*
c. *How does the solution approach perform for these experimental setup considered?*

5. Recommendations and conclusions

The last research question answers the and findings from the results of the experiments and recommendations made to Cofano based on the results.

RQ 5: What are the recommendations from the results of the experiments?

a. *What are the pros and cons inferred from the performance of the solution approach?*
b. *What are the recommendations provided to Cofano based on the results of the experiments?*

1.5. Research approach

The research is divided into different phases, each corresponding to answering different research questions. The main research question that is defined in Section 1.4 is systematically solved by using these different research phases. The first phase is the problem identification and planning phase discussed in Chapter 1. This phase is followed by the problem analysis phase, which is related to sub-questions 1 and 2 and explained in Chapter 2 and 3. The next step corresponds to the solution generation phase, which is answered by solving sub-question 3, where different alternatives to the solution approach are analyzed, and a selected solution approach is implemented. Consequently, the answer to sub-question 4 discusses the evaluation of the performance of the solution approach and the experimentation conducted with this solution approach. The explanation for the same can be found in Chapter 5. The last phase of the research is the recommendation and conclusion phase that includes the recommendation for implementing the findings from the solution approach and are answered by the final set of sub-questions. The flow diagram of the research approach explaining the different phases of the research is represented in Figure 3.

1.5.1. Research deliverables

The list of deliverables for the research includes:

- An algorithm for finding the best routing and scheduling option for the given number of container request in a planning horizon.
- Results from the numerical experiments conducted with the real-life data showing the performance of the algorithm.

1.6. Conclusion

A brief introduction to the research problem and the research goal has been provided in this chapter. The identification of different stakeholders involved in the research and their interactions were analysed. The research questions were formulated in a sequential approach to solve the main research problem. The research approach section offers a brief outline of the actions and deliverables from each phase of the study.

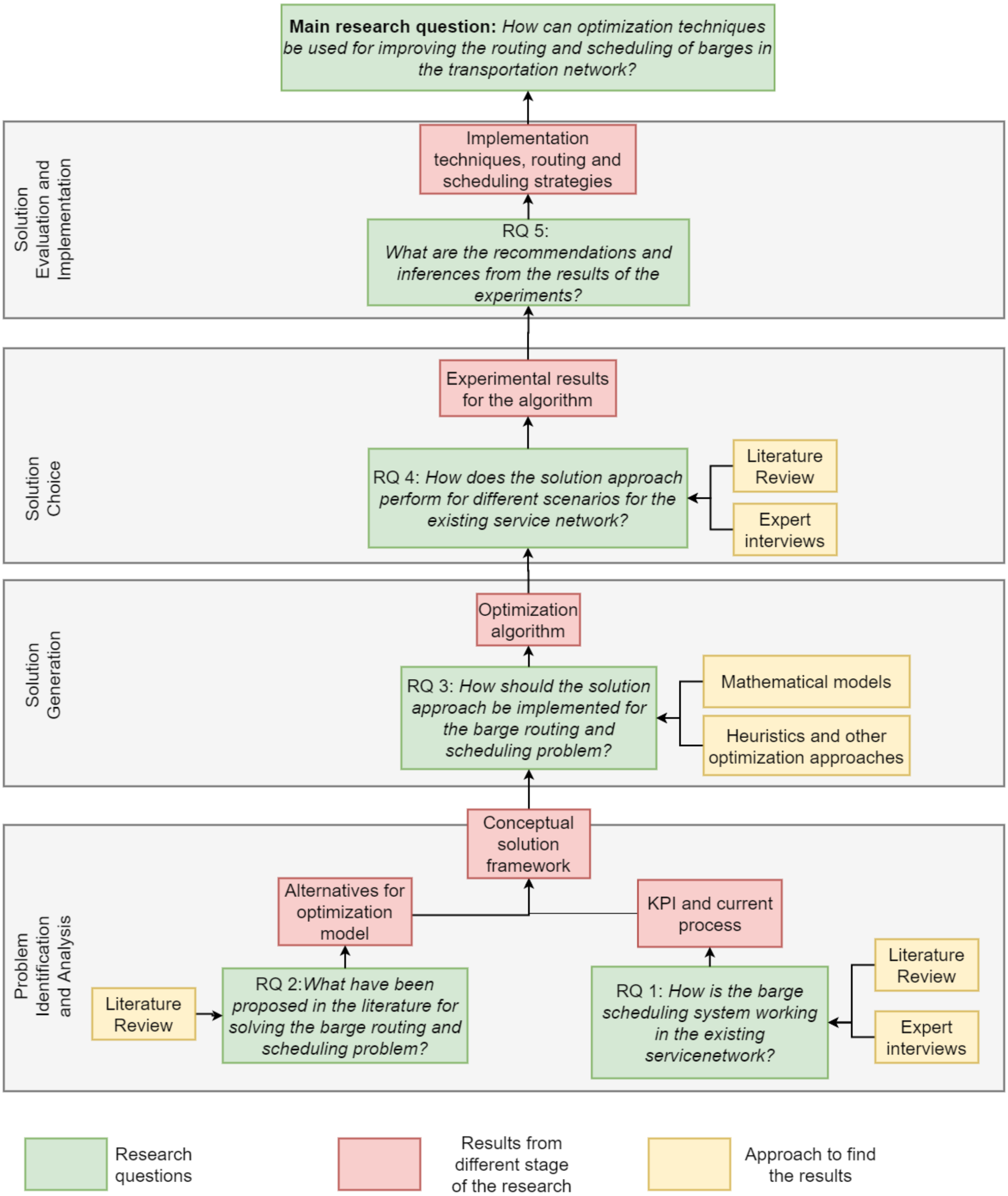

Figure 3: Research design

Chapter 2: Problem Context

The detailed description of the problem is provided in this chapter. The current transportation service network that is considered for the research is explained in Section 2.1 and Section 2.2. The working of the current booking process is explained in Section 2.3. The complexity in solving the routing and scheduling problem is given in Section 2.4. Followed this, an introduction to the nature of the problem at hand is discussed in Section 2.5. Finally, Section 2.6 describes the technical and business-specific requirements for the solution approach of the research.

2.1. Container demand in the Northwestern European region

This research is focused on the hinterland container supply chain in Northwestern Europe and more concentrated towards the inland waterway (Barges) mode of transportation. Port of Rotterdam and Antwerp are the top two ports located in Western Europe, holding more than 50% of the market share in the Hamburg-Le Havre (HLH) range. The Hamburg-Le Havre (HLH) range includes ten important ports in Northwestern Europe. (Port Authority of Rotterdam, 2018). The Port of Rotterdam handled 14.5 million TEU, and the Port of Antwerp handled 11.1 million TEU during the year 2018. The port of Rotterdam achieved a throughput of 240.7 million tons in the first six months of 2019. Container throughput, one of the strategic priorities of the Port Authority, rose by 4.8% in tons, (+6.4% in TEU) by comparison with the first six months of 2018, which is a new throughput record for the Port of Rotterdam. The share of containers amounted to 32% of total throughput in the first half of 2019. The sharp increase in container throughput over 2018 was primarily due to the rise in transshipment at ports; in other words, intercontinental cargo transported to and from European destinations via Rotterdam (Port of Rotterdam Authority, 2019). The port is ideally situated for inland shipping because of the Maas and Rhine rivers. A fundamental requirement of a successful, competitive hinterland transport system is the ability to offer services, which are cost-effective, reliable and have a short transit time (Visser, Konings, Wiegmans, & Pielage, 2009). One of the critical factors for these ports to operate such a high volume of cargo would be the high-quality national waterway network in the Netherlands and Belgium. The ports are located at the Rhine estuary which offers access up to Switzerland and to major consumer and industrial regions in Germany that generate large volumes of container export and import (Konings, Kreutzberger, & MaraŠ, 2013).

The vision statement of the Port of Rotterdam states that the modal split of 45% of the container transportation should be achieved by barge transportation by 2035. Similarly, the Port of Antwerp defined their ambition to make a container transportation modal split of 42% through barges by 2030 (Source: Port of Rotterdam, Port of Antwerp web sites). This vision statement emphasizes the focus on container transportation through barges to increase sustainability in transportation operations.

2.2. Service network of barge operator

After discussing the growth and importance of container transportation in the hinterland region, we focus on the research specific case. The barge operator who operates multimodal transportation from the ports of ARA (Antwerp, Rotterdam, and Amsterdam). The operation of the multimodal transportation network itself is a very complex area to be explored. Hence, the scope of the research is narrowed to container transportation on barges. Barge operator owns barges with different cargo-carrying capacity varying from 20 TEU to more than 250 TEU. There are various scheduled service routes for the barges operated. The barge service connects different inland terminals in the European region with the two major seaports, namely, Port of Rotterdam and Port of Antwerp. Figure 4 represents the Rhine service route, which is one of the longest service routes operated by the barge operator. The service is between the Port of Rotterdam in the Netherlands and the inland terminals at Basel in Switzerland. The service also connects the Port of Antwerp in Belgium with the inland terminals at Basel. There are weekly sailing schedules for the barges between these terminals and the list of possible intermediate terminals that might be visited by these barges during the voyage represented in Figure 4.

Censored

Figure 4: Rhine service route by barge operator

The above network is one of the service routes operated by the barge operator. Figure 5 represents the complete service network of all barge service along with intermediate terminals represented by different colour lines. From Figure 5, it can be seen that there is a significant concentration between the Port of Rotterdam and Antwerp since these are the terminals corresponding to the port hinterland. The mass of the network is not dense around the other inland terminals such as Basel.

Censored

Figure 5: Barge service network of barge operator

2.2.1. Process flow for customer request

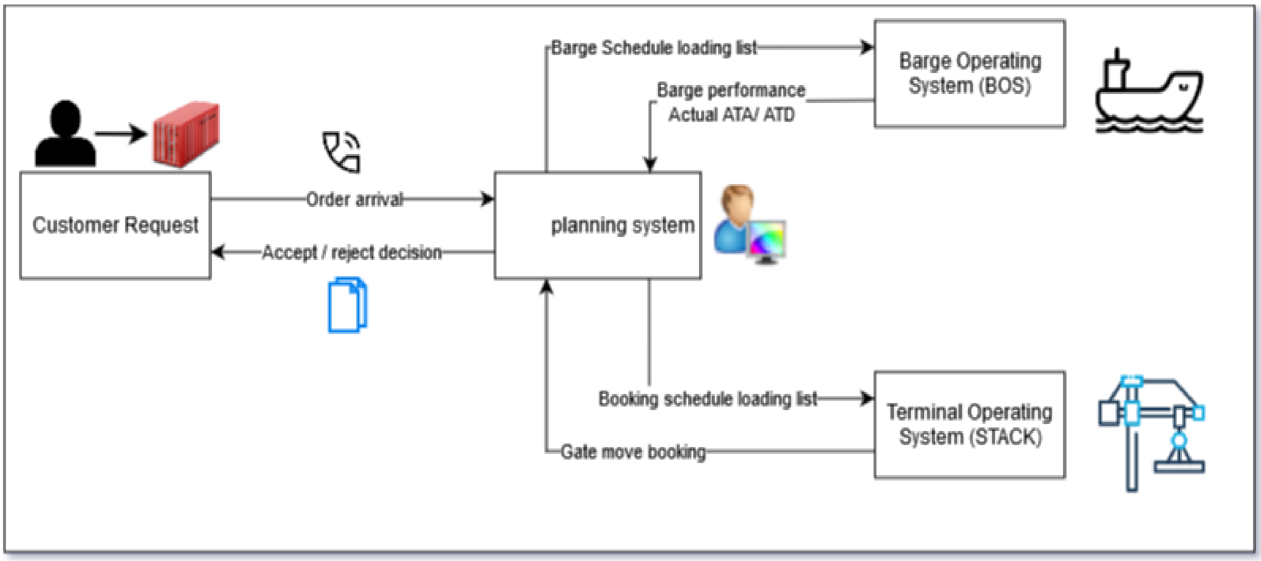

Figure 6: Process flow for customer request processing

Figure 6 visualizes the process flow of container transportation request. The customer requests to transport containers from a pick-up location to a delivery location is received by the planning system from different customers. The planner decides to transport the containers on the available barge schedule and make the assignments in planning system accordingly. The information about actual barge arrival time and the gate moves of a terminal from Barge Operating System and terminal operating system are inputs for planners to plan the container request on the available barge schedule. The planners do the planning and scheduling process through manual planning. The customer request can either be accepted by the planners if a request can be satisfied by the available resource or can be rejected by the planners if the resource to fulfil the request is not available. However, a customer request is fulfilled in most of the cases through an alternative arrangement by using a different mode of transport such as trains or trucks operated by barge operator. Delegating the transportation of a container request by finding a different transport company to handle the booking request rather than rejecting is also followed sometimes. This action enables a better customer retention rate.

2.2.2. Problems in routing and scheduling of barges

The planners do the planning for container allocation to the weekly schedule of the barges in the service routes. The planners often try to consolidate the containers that are headed to the same destination. However, the barge visiting the port should visit more than one port terminal to load and unload the containers. The schedule for the barge to visit several terminals leads to triggering a domino effect during real-life operations. The expected and actual time of the arrival and departure of barges at a terminal is not always as planned. When there is a disturbance in arrival time of a barge in one terminal, the expected time of arrival and departure for the remaining terminals are affected. This disturbance causes the change in the schedule of the following barges visiting the terminal. These changes are monitored and corrected during the real-time operation of the barges. Also, when a barge enters a terminal, there is a set of waiting times, berthing times and handing times associated with it. Change in the actual schedule makes it difficult to operate the barge as expected by the initial plan. The barge has to stay for one or more days at the port to handle the containers due to these disturbances, which in turn leads to high operating costs. Hence, there is a need to reduce the number of terminals visited by a barge during a port visit to reduce the uncertainty in the actual plan. This reduction in the number of terminal visits can be achieved only by better consolidation of the container transportation requests.

The literature (Fazi, Fransoo, & Van Woensel, 2015) analyse the ratio between the import and export containers in the Rotterdam region. It is stated that the ratio of import to export is 2:1, meaning that the cargo import is twice the amount of cargo export. This unbalanced ratio causes the need for transporting more cargo towards the inland and underutilizing the resource capacity during the voyage towards the port terminals. There are also scenarios where the empty containers are returned to the empty container depot. These activities are considered as non-value adding and should be minimized by reducing the distance of empty container transportation. Further, the demand for container transportation does not remain stable over time. It varies with the season as the demand for different products varies with season. There will be limited operations during the off-peak season when the water level of the rivers are low, and some regions of the river might not be accessible. Hence, there is a need for a dynamic barge schedule based on the container demand.

Several competitors of barge operate in the ARA region. From the findings of Section 2.1, it can be inferred that the volume of containers handled by the major ports such as Rotterdam and Antwerp are high. To gain a competitive advantage over this increasing volume of the container, the planners need to create a plan that operates the barges to serve these large number of container requests at a lower operating cost.

To overcome the problems mentioned above, the planners need to consider the opportunity of consolidation and transshipment that is available throughout the service network. There are different transshipment and consolidation strategies available which makes the routing and scheduling process more complicated. We consider the below illustrative problem as explained by (Crainic, 2003) where a container is to be transported from terminal A to terminal D. There are barge schedules available between different terminals represented as S1, S2, S3, etc.

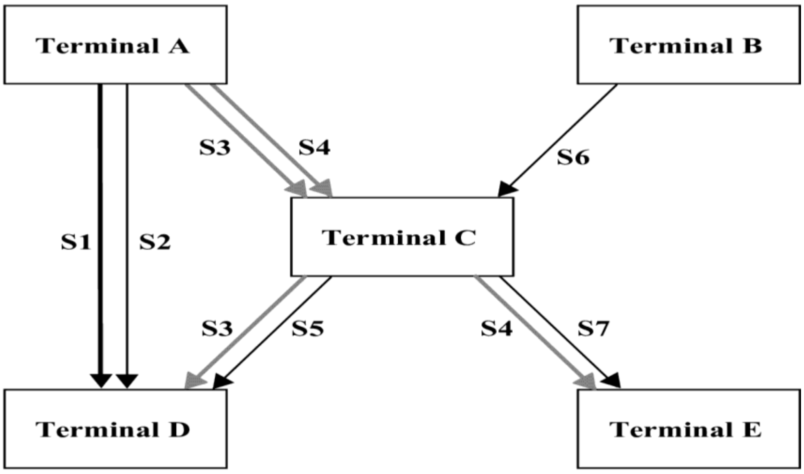

Figure 7: Illustrative problem; Scenarios for container consolidation and transportation in a service network design (Crainic, 2003)

From Figure 7, it can be inferred that the container can be routed in different ways to be transported from terminal A to terminal D.

1. The first strategy is to consolidate the container with the other containers going directly from terminal A to terminal D in barges that are operating in schedule S1 or S2. This is a direct shipment strategy.
2. The second strategy is to transport the container in the barge that is travelling in schedule S3. In this scenario, the container remains in the barge throughout the voyage and is not handled at terminal C.
3. The third strategy is to transport the container in the barge operating in schedule S4 to reach from terminal A to terminal C. The barge drops the container at terminal C. It continues its voyage to terminal E with the remaining containers. Another barge which is scheduled for S5 or S3 collects the container from the terminal C and makes its voyage to reach the destination terminal D.

4. The final strategy is similar to the third strategy; the difference is that the consolidation of the container happens both at terminal A and terminal C. Hence, the container is transported from terminal A to terminal C through S3 or S4. There is another consolidation process at terminal C. The container has to wait for the other containers arriving from terminal B to terminal C that are later transported to terminal D through schedule S3 or S5.

Transshipment operations performed at the intermediate terminals during a voyage expand the scope of shipping services and enables container consolidation at the intermediate transshipment terminals. Correspondingly, the shipping companies may benefit from economies of scale. Introducing such transshipment operations may be more beneficial in terms of costs and flexibility. It brings some challenges in routing and scheduling the barges, where containers can be put on many different (sub) routes to reach their final destination and complexity in coordination between barges to perform the transshipment. The coordination refers to the synchronization of schedules of the barges visiting the transshipment terminal. The opportunities for container consolidation and transshipment increases when the export and import scenarios are considered simultaneously.

There are also other restrictions, such as, a barge visiting a terminal need to satisfy a minimum call size of container requests that must be loaded or unloaded. Every container request has a due date before which the container should be transported to its destination location. Violating this due date agreement, the service providers should pay the penalty for the time they have delayed.

There is a set of empty container depots that are also considered as terminals. These terminals are the points from which an empty container is transported to a client location for loading the cargo into the container and further used for transportation. The empty container terminal is also a point to which a customer returns an empty container after use. It is considered not to differentiate the regular terminal from the empty container terminal in this research as the full and empty container transportation request are treated similarly.

2.3. Research problem to be solved

The goal of this research is to provide an algorithm that gets the input of the container transportation request and makes use of the resources in the service network to create a barge routing and scheduling plan for a period considered. The problem is a pick-up and delivery problem for container request with transshipment opportunities. There is also a time window that represents the arrival date and due date for the container transportation request, which should be satisfied in the plan. The routing and scheduling plan should include the sequence of terminals visited by each barge available in the service network and the containers that should be picked up and delivered during these visits. The algorithm should make use of advanced optimization techniques to derive the routing and scheduling plan.

2.4. Business and technical requirements

The scheduling algorithm created for solving the routing and scheduling problem will be used by Cofano to serve their customers who operate their barges in different service networks. There are specific technical and business requirements for the algorithm that are expected from Cofano as follows:

- The objective of the routing and scheduling plan created should be to minimize the overall operational cost for container transportation.
- There should be a minimum number of containers handled by each barge at a terminal during its visit. This minimum number of container request handled during the visit is referred to as the minimum call size.
- There should be transshipment terminals considered that can be used for the transshipment purpose where the containers can be dropped for other barges to be picked up.
- There are due dates before which each container should be transported to the destination location. Violating this due date leads to a penalty based on the time violated.

2.5. Conclusion

The chapter introduced the service network of the barge operation. The activites such as the current booking and scheduling procedure for container transportation were explained in this chapter. The analysis of different problems faced by the planners in planning the barge schedule and issues related to the complexity in routing and scheduling of barges were discussed with examples. Various business and technical specifications mentioned by the problem owners were identified. Thus, the working of the barge routing and scheduling system in the existing service network answers the first set of research questions related to the existing configuration of the problem context.

Chapter 3: Literature Review

A systematic literature review is performed in this chapter, similar to the method of Preferred Reporting Items for Systematic Review and Meta-Analysis (PRISMA) (Moher, 2009). The procedure followed during the systematic literature review and the methods used for the review process is explained in Appendix A. The literature review is performed to identify the nature of the research problem and to analyze different solution approaches available for the research problem. Section 3.1 identifies the category of the research problem by analysing various works of literature. Section 3.2 describes the different classification of literature problems based on the problem attributes followed by an analysis of solution techniques such as exact and heuristic methods to solve the problem in Section 3.4. The brief discussion on the selection of the solution method from the literature is provided in Section 3.5. The conceptual framework, which is the result of the first phase of the research, is provided in Section 3.7.

3.1. Identifying the category of the research problem

The different levels of planning decisions explained in the literature are analyzed first. Various type of shipping problems identified from different work of literature is discussed later. The result of this analysis is the category of the research problem to be focused in this research.

3.1.1. Levels of planning decisions

The long haul freight transportation refers to transporting cargo over a longer distance in the supply chain. Transportation of containers using barges in the hinterland can be considered as long haul freight transportation. The planning decisions of a shipping problem in the long haul freight transportation are categorized into three levels, namely strategic, tactical and operational (Crainic, 2002).

- The strategic decisions are medium or long term decisions and are based on the aggregated information over time. The knowledge of the future is limited during the strategic level decision. Some of the shipping problems falling under the strategic level include transportation system design, selection of service area, fleet composition and choice of port.
- The tactical decisions are medium-term decisions that are based on strategic decisions. The information at the tactical level is more reliable than during the strategic phase. The tactical level decision is more concerned about the service network design. The choices such as fleet deployment, timetable creation, container flow assignment, repositioning of the fleet for the next planning period are included in this level.
- The operational level decisions are short term decisions. The operational level decision can further be classified as offline and online operational decisions. The offline decisions are influenced by the tactical level decisions. Offline decisions in barge scheduling include the problems like the sequence of terminals visited by barges and allocation of containers to barges. The online-operational decision refers to the decisions taken in real-time operations, with the decrease in problem uncertainty. These decisions are performed by the local management, yardmasters and dispatchers.

Some of the choices include the transition from the old planned schedule to a new adjusted schedule in real-life sipping operations.

Further detailed explanation about the levels of decision phases can be found in (Crainic, 2002), (Crainic, 2000), (Kjeldsen, 2012).

We consider the research problem defined in Chapter 2. The container demand for the planning horizon is considered and an effective schedule for the barges to transport the containers during the planning horizon is to be made. The scheduling decision is influenced by the resource capacity available, and the service network considered from the tactical level decision. The real-time barge operation is monitored and adjusted based on the route and schedule generated as a result of the scheduling process. This real-time monitoring and adjustment are referred to as the online-operational level decision. After analysing the different levels of planning decisions and from the above justification, we can categorise the problem of creating the barge schedule to be an offline operational level decision problem.

3.1.2. Types of shipping problems

The different types of shipping problems available in the work of literature are analysed in this section. Authors (Christiansen, Fagerholt, Nygreen, & Ronen, 2013) classify the shipping problems based on the operations into the liner, industrial and tramp shipping.

Liner shipping

The liner shipping involves a fixed route based on a published schedule between regions to maximize the profit from the transportation of cargo. This type of shipping problem is often compared to the bus service operated between areas. In liner shipping, the ship travels from an origin terminal to a destination terminal visiting a set of intermediate terminals during the voyage. The service network can also be considered similar to a liner shipping problem.

Industrial shipping

The next type of shipping operation is Industrial shipping. The industrial operator owns and operates the ships to transport their cargo in the supply chains to reduce the operating cost.

Tramp shipping

The last category of shipping problem is the tramp shipping, where the shipment contract to ship mandatory cargo based on the Contract of Affreightment (COA) is shipped with the available fleets (Christiansen et al., 2013). Tramp shipping is compared to an operation that is similar to the taxi cab service. Opportunity to serve additional optional cargo that is generated on the spot apart from the mandatory cargo is also considered during the tramp shipping. This extra spot cargo is considered to be one of the differences between the tramp and industrial shipping (Brønmo, Christiansen, & Nygreen, 2007).

The research problem is identified to be related to the offline operational level decision problem corresponding to the liner type of shipping. Hence, the literature associated with solving the liner shipping problems for inland waterways is focused.

3.1.3. Different routing and scheduling problems in liner shipping

This section analyzes the different routing and scheduling problems in liner shipping and identifying the category of the problem corresponding to our research problem.

Service frequency planning

Determining the service frequency for a liner shipping route is one of the key problems faced by the shipping industry concerning the tactical level of the planning decisions. Authors (Riessen, Negenborn, Dekker, & Lodewijks, 2015), (Crainic, 2002), (Crainic & Kim, 2006), (Crainic, 2000), (Kjeldsen, 2012) deal with deciding the service frequency for the service network design as output or decision variable from the mathematical model for different demand patterns. Authors (Fu, Liu, & Xu, 2010) analyse the impact of shuttle frequency on the waiting time per container. Authors (Konings, 2006), (Konings et al., 2013) analysed the hub and spoke model and define a relation between travel distance and service frequency to offer an efficient service for different barge capacity. The relation between the length of the spoke connection, and barge productivity related to the service frequency is analysed.

Barge rotation planning

Authors (Notteboom & Konings, 2004) explain the nature of the existing liner operation of barges in the Northeastern European hinterland region where vessels sail between seaport of Rotterdam and Antwerp and dedicated regions in the hinterland (Lower, Middle and Upper Rhine river basin). The analysis based on a line bundling loop system where 4-6 terminal calls in the hinterland region and average of as high as up to 10 terminal calls at the port region are observed. Considering the problems faced in the port region of the liner barge network, (S. Li, Negenborn, & Lodewijks, 2017) analyze the effect that is caused by the plan generated by the individual vessel agent of a ship. Adjustment of the plan in real-life operation causes more waiting time because of the lack of corporation between different vessels that are visiting a terminal. This leads to domino effects that make the total sojourn time and total waiting time of all vessels visiting the port to increase substantially and conflict with the rotational plan. They propose a central coordination system that communicates between different vessel agents to plan the rotation plan for barges in the port region.

Pick-up and delivery problem

Authors (Christiansen et al., 2013), (Crainic & Kim, 2006), (Fazi et al., 2015), (Korsvik, Fagerholt, & Laporte, 2010), (Lin & Tsai, 2014), (Brønmo et al., 2007) and (A. Caris, Macharis, & Janssens, 2011) discuss the pick-up and delivery problem in liner shipping using ships and barges where a set of cargo is picked up from origin location and delivered to the destination location. These problems are modelled similarly to the general pick-up and delivery problems. The routes travelled by ship along with the list of cargo handled during each terminal visit is identified. Authors (Lin & Liu, 2011) and (Stålhane, Andersson, Christiansen, & Fagerholt, 2014) explain the pick-up and delivery problems in tramp shipping where combined routing and freight allocation decisions are made.

Authors (Alfandari et al., 2019), (Braekers, Caris, & Janssens, 2013) and (Maraš, Lazić, Davidović, & Mladenović, 2013) discuss the special case of pick-up and the delivery problem of liner shipping in inland transportation where round trips are made between the port terminal in the mouth of the river and the inland terminal which is considered as the last terminal in the river liner route. The sequence of intermediate terminals visited during upstream and downstream voyages during the round trip is identified as the output. The cargo that is handled at each terminal visit is identified as well.

Empty container repositioning

Authors (Choong, Cole, & Kutanoglu, 2002), (Alfandari et al., 2019), (Braekers et al., 2013), (Maraš, 2008) and (An, Hu, & Xie, 2015) made an extension to the pick-up and delivery problem with optimizing the routing and scheduling for barges in inland waterways along with considering the repositioning of empty container in the hinterland supply chain. Our research does not focus on the empty container repositioning problem. Hence, further investigation regarding the topic is avoided.

Detention and Demurrage problems

The next set of problems in liner shipping discussed in the literature is related to demurrage and detention. The demurrage period is the penalty-free period in which a container can be stored in a sea terminal. It starts after the release of a container to the sea terminal. The detention period is a penalty-free period that starts when the container leaves sea terminal and ends when a container is expected to be returned to an empty container depot. Authors (Fazi & Roodbergen, 2018) and (Fazi, 2014) analyses the impact of the routing decision based on detention and demurrage penalty on the inland transportation network by making use of different container consolidation strategy. Our research does not consider the detention and demurrage period into account. Hence further investigation is avoided.

3.1.4. Identification of Problem category.

We focus on the research problem at hand. Our research problem focuses on finding a pick-up and delivery sequence for the container request and scheduling the barges based on the actual container demand. It is evident from Chapter 2 that the research problem is a pick-up and delivery problem for container cargo with consideration of transshipment opportunities. The analysis from the literature also suggests the same. The research problem is identified as a pick-up and delivery problem of containers using barges corresponding to the offline operational level decision in liner shipping. The literature corresponding to the transshipment scenarios for the pick-up and delivery problem using barges are minimal. Hence, the decision to investigate the transshipment opportunity while solving the routing and scheduling problem is considered.

3.2. Classification of the available literature for routing and scheduling problems

This section deals with the classification of the literature identified for the routing and scheduling problems based on the problem attribute. The attributes are the problem-specific character used to classify the literature. The methodology used is similar to the classification framework used by (Bierwirth & Meisel, 2015), where different attributes have been identified in order to classify the literature found for the scheduling problems. The attributes of our research are derived based on the characters defined by (Kjeldsen, 2012) during their classification of the liner ship routing and scheduling problems. A notation with six different attributes is used for representing the problem classification. The solution methods used in each problem (both exact and heuristic methods) are also identified for each literature work during the classification.

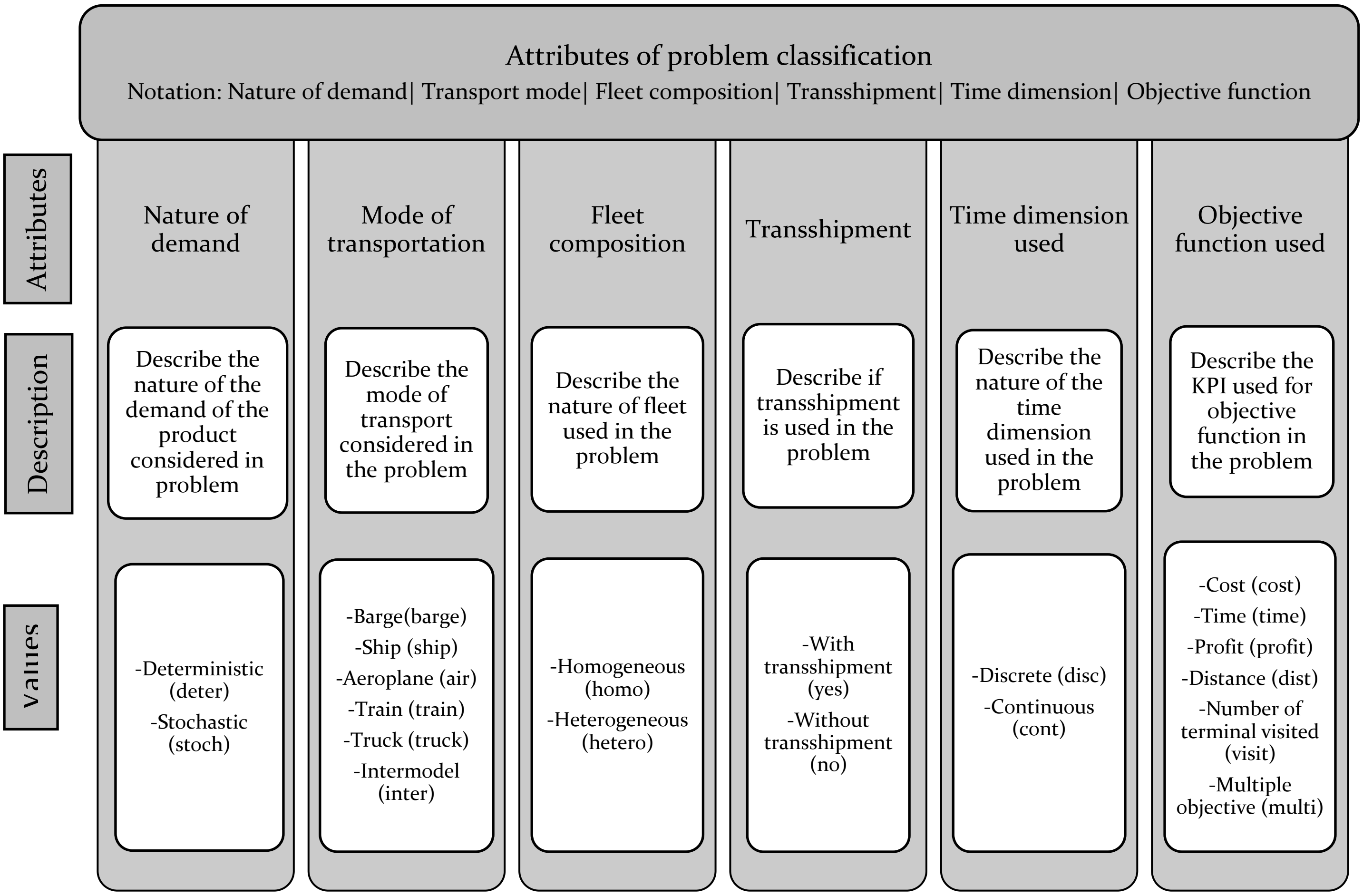

Figure 8: Classification of attributes, their description and values

Figure 9 represents the attributes, description and their values that are considered for the classification of the works of literature. Each term in the notation represents the value of an attribute. A short description of the attribute and the values it can take are summarized in Figure 9. The exact and heuristics solution methods used in each literature work are also differentiated and summarized in Table 1. The discussion on each attribute and solution methods from the work of literature is provided later.

Table 1: Problem classification for the literature

Serial.no	Reference	Problem classification	Solution method
1	(Christiansen et al., 2013)	deter | ship | hetero | cont | transfer | multi	MIP | -
2	(Riessen et al., 2015)	deter | inter | hetero | cont | transfer | cost	MIP | -
3	(Crainic, 2002)	deter | train | homo | disc | transfer | cost	MIP | -
4	(Crainic & Kim, 2006)	deter | inter | hetero | disc | transfer | cost	MIP | -
5	(Crainic, 2000)	deter | train | homo | disc | transfer | cost	MIP | -
6	(Rivera, 2018)	stoch | inter | homo | disc | transfer | cost	- | ADP
7	(Fazi, 2014)	deter | barge | hetero | cont | transfer | cost	MIP | MCMC
8	(Liu & Pang, 2014)	deter | ship | homo | cont | direct | cost	MIP | -
9	(Foss et al., 2016)	deter | ship | hetero | disc | transfer | cost	MILP | Iterative Local Search
10	(Zweers et al.,, 2019)	deter | barge | homo | cont | direct | cost	ILP | Two stage Heuristic
11	(Moccia et al., 2006)	deter | train | homo | cont | transfer | cost	MILP | Column generation
12	(An Caris, 2011)	deter | barge | hetero | disc | transfer | time	- | Simulation Model
13	(González-Ramírez et al., 2009)	deter | truck | hetero | cont | transfer | cost	- | Local Search
14	(Castillo-Villar et al., 2014)	deter | ship | hetero | disc | direct | cost	MIP | LNS
15	(L. Li et al., 2015)	deter | inter | hetero | disc | transfer | cost	RIFC | Simulation Model
16	(Zenker, Emde, & Boysen, 2016)	stoch | barge | homo | cont | direct | cost	- | DP
17	(Pruijn, 2018)	deter | barge | hetero | cont | direct | time	- | Priority based algorithm
18	(Konings et al., 2013)	deter | barge | hetero | cont | transfer | multi	- | Simulation Model
19	(De Armas et al., 2015)	deter | ship | hetero | disc | direct | cost	MIP | GRASP-VNS
20	(Sharypova, 2014)	deter | barge | hetero | cont | transfer | cost	MIP | Metaheuristics
21	(Davidovi, 2011)	deter | barge | homo | cont | direct | profit	MILP | Multi start Local search
22	(Hemmati et al., 2014)	deter | ship | hetero | cont | direct | cost	MIP | ALNS
23	(Korsvik et al., 2010)	deter | ship | hetero | cont | direct | profit	- | Tabu search
24	(Kjeldsen, 2012)	deter | ship | hetero | cont | transfer | cost	MIP | Column Generation
25	(Alfandari et al., 2019)	deter | barge | homo | cont | direct | profit	MIP | -
26	(Braekers et al., 2013)	deter | barge | homo | cont | direct | profit	MIP | -
27	(Kim et al., 2008)	deter | inter | hetero | cont | transfer | cost	MIP | -

Serial.no	Reference	Problem classification	Solution method
28	(S. Li et al., 2017)	deter | barge | hetero | cont | direct | time	MIP | LNS
29	(Lin & Tsai, 2014)	deter | ship | hetero | cont | transfer | cost	MIP | Lagrangian relaxation
30	(Maras et al., 2013)	deter | barge | homo | cont | direct | profit	MIP | VNS
31	(An et al., 2015)	deter | barge | homo | cont | direct | cost	MIP | GA
32	(Bronmo et al., 2007)	deter | ship | hetero | cont | direct | profit	MIP | Column generation
33	(Lin & Liu, 2011)	deter | ship | hetero | disc | direct | profit	MIP | GA
34	(Stalhane et al., 2014)	deter | ship | hetero | cont | direct | profit	MIP | Branch and price
35	(A. Caris et al., 2011)	stoch | barge | hetero | cont | transfer | time	- | Simulation model
36	(Qu & Bard, 2012)	deter | air | hetero | cont | transfer | cost	MIP | GRASP-ALNS
37	(Rais, Alvelos, & Carvalho, 2014)	deter | truck | hetero | cont | transfer | cost	MIP | -
38	(Takoudjou et al., 2012)	deter | truck | hetero | cont | transfer | cost	MIP | -
39	(Hagen, 2016)	deter | truck | hetero | cont | transfer | cost	MIP | ALNS
40	(Vornhusen et al., 2014)	deter | truck | hetero | cont | transfer | cost	MIP | -
41	(Masson et al., 2017)	deter | truck | hetero | cont | transfer | cost	- | ALNS

3.2.1. Nature of demand

The first attribute discussed is the nature of the demand data used in the literature. In most of the literature, the demand is considered to be deterministic. Deterministic demand refers to knowing the exact amount of demand that is going to arrive during the planning period. Stochastic demand refers to flexible demand based on a distribution where the exact amount of demand is not known accurately but can be defined by a distribution function. Only a few works in the literature are found about vehicle routing and scheduling based on stochastic demand for hinterland transportation. The stochastic cases are found mostly in simulation models developed for simulating the behaviour of inland transportation.

3.2.2. Mode of transportation

The next attribute discussed is the mode of transportation used. Most of the literature searches were confined to barge and ship mode. However, to find concepts about the different cargo consolidation approaches, other modes such as trains, aeroplanes, trucks, and intermodal transportation were also investigated. Authors (Crainic, 2002) and (Crainic, 2000) discusses the long haul transportation of freights through the train and are interested in finding the service frequency for the schedule in a service network. Authors (Creemers, Woumans, Boute, & Beliën, 2017) uses an effective algorithm for the detection of bundling, back-hauling, and round-trip opportunities, as well as “collect-and-or-drop” opportunities for container consolidation using truck mode. Authors (González-Ramírez et al., 2009) employs two types of consolidation strategies, namely consolidation options in vehicle and consolidation options at a terminal. For vehicle consolidation, a heuristic procedure for less than a truckload (LTL) based on insertion and reallocation strategy is used. For terminal consolidation, the p-median approach is used to find the effective terminal location for consolidation. Authors (Notteboom & Konings, 2004) explain the difference between the handling of containers at a terminal using train and barge transportation modes. It is discussed that the handling or consolidation of containers in train networks can be either by horizontal operations (i.e., shunting of wagons) or on vertical operations (i.e., the loading/ unloading of containers). In inland barge networks, the regrouping of containers (transfer) requires vertical container handling operations by crane at barge terminals. Horizontal transfer in barges might only occur when an operator uses push barges in view of regrouping large container batches. This condition limits the horizontal container handling in barge transportation. Most of the literature works selected are pertaining to the ship and barge mode of transportation in liner shipping. Few works of literature in tramp shipping with the consolidation and transshipment opportunities are considered as well. The transshipment and consolidation concepts are explained in detail while discussing the transshipment attribute.

3.2.3. Fleet composition

The next attribute discussed is the composition of the fleet used in the literature. Most of the literature uses a heterogeneous fleet where the fleet is not the same as one another. The difference might be due to varying capacity or travel speed, among others. The heterogeneous fleet also involves using intermodal transportation where high capacity modes such as barges and trains for the long haul and low capacity mode such as trucks that are used for a short leg. The varying capacities (heterogeneous) for individual transportation modes are considered as well (e.g., different capacities for different barges). Some literature makes use of a homogeneous fleet where they consider only one type of vehicle with the same vehicle capacity and travel speed. A homogeneous fleet reduces the complexity of the problem and enables the easier formulation of a mathematical model. The vessel capacity is restricted and not homogeneous in real-life scenarios for barge transportation due to variations in draft limitations and other physical conditions in segments of the river (Notteboom & Konings, 2004).

3.2.4. Time dimension

The next attribute to consider is the nature of the time dimension used in the literature. This character is mainly considered to analyze the representation of the container flow or vessel flow used in different works of literature. Most of the literature used a discrete-time interval. The problems are formulated based on node and arc models. Hence, discrete-time intervals are easily adapted for these models. Few works of literature make use of continuous-time intervals where the defined time of operation is added to the time parameters in the mathematical model. Almost all the literature considered to have a defined planning horizon for which the planning is done. The planning horizon is a defined time interval for which the demand for transporting products are considered. Authors (Cheung & Chen, 1998) perform experiments with rolling horizons and conclude that a longer planning horizon is not necessarily better than a shorter one. Receding horizon control (RHC) determines a control action over a prediction horizon by making a prediction and performing optimization and implement control for the current time step based on the predictions as explained in (L. Li et al., 2015).

The representation of the solution in the literature is discussed based on the time dimension considered. Authors (Riessen et al., 2015) and (Crainic, 2000) classifies the service network design problems for intermodal container network into Link-based network flow model (LBNF) and path-based network design model (PBND) for single and multiple commodities. There are three different types of representations based on the time dimension used in the problem.

Continuous-time with the terminal as nodes and vehicle route as a link:
Authors (Sharypova, 2014), (Crainic, 2002), (Crainic & Kim, 2006) and (Crainic, 2000) represents the shipment problem where the pick-up and delivery terminals are denoted as nodes. The cargo has an origin and destination corresponding to the terminal node. The travelling time between terminals and service time at a terminal is used to calculate the arrives time and departure time of the ship at a terminal node. The available time window of the terminals is represented as the time window associated with a node. An arc or link between node pairs represents the path and direction of the voyage to reach from one node to another.

Discrete-time with the terminal as nodes and vehicle route as a link:
Authors (Foss et al., 2016), (Lin & Liu, 2011) and (Kjeldsen, 2012) represent the problem concerning a time-space network. Each node represents the location of the ship at a time. In order to visualize this, a 2-dimensional graph with time on the *x*-axis and terminals on the *y*-axis is considered. Nodes represent the points on the graph denoting the terminal at a point in time. An arc connecting two nodes can be interpreted as the position or terminal where the ship is present at that point of time. There are different sets of arcs used based on the activity that is performed by the ship (travelling, loading and unloading operations or idle).

Continuous-time with cargo origin and destination as nodes and vehicle route as a link:
Authors (Christiansen et al., 2013), (Fazi et al., 2015), (Edirisinghe, Bowers, & Agarwal, 2010) and (Qu & Bard, 2012) represent cargo origin and destination as nodes and considers a node pair for every individual cargo to be transported. Hence, there are $2n$ nodes for n cargo to be transported (first n nodes representing pick-up/ origin points and other n nodes representing delivery/ destination points). Authors (Alfandari et al., 2019) uses two sets of the arc for distinguishing the full and empty containers flows. Authors (Moccia et al., 2006) extends this formulation by considering an additional set of nodes representing the possible pick-up time window and delivery time window for every cargo. For instance, each day of the week is considered as a pick-up time window node. Hence, the cargo can be picked up during any one of the days by visiting any one of the pick-up time window nodes. An arc connecting the origin node to one of the pick-up time window nodes with zero travel cost ensures that the pick-up operation during the pick-up time window. Similarly, the arc connecting the delivery node to a delivery time window node is also considered.

3.2.5. Transshipment scenarios considered

The attribute considered next is transshipment opportunities considered during transportation. Literature work either considers the direct shipment of cargo from origin to destination or includes transshipment in the cargo route. We are more interested in how different literature approaches the problem by allowing the transshipment and consolidation rather than literature without transshipment.

A collaborative hub network model is created for fast-moving goods by (Groothedde, Ruijgrok, & Tavasszy, 2005). Large barges are used for transporting the cargo by combining them. The inference suggests that consolidation in hub and spoke network allows more efficient and more frequent shipping with achieving economies of scale for fast-moving goods. Authors (Konings, 2006), (Konings & Maras, 2011) and (Sharypova, 2014) dealt with the investigation of the hub and spoke models for barge service network design and its performance compared to the other models. They also compare the performance of the transportation models based on different hub location to varying distances from the port. It is found that the higher the distance of hub location from the port, the better is the possibility of exploiting economies of scale through consolidation. Authors (Notteboom & Konings, 2004) explain the structure of the river systems, which is similar to a treelike structure with limited or no lateral connections between the different branches. Hence the possibility of having a hub and spoke system in inland waterways is complicated compared to a train system which is having many lateral connections.

Several kinds of literature explicitly investigate the potential locations for consideration of a hub terminal or transshipment terminal in the Northwestern Europe region. Authors (Notteboom & Konings, 2004) discuss Duisburg terminal, which is a clear example for growing inland hub. Containers are transported by large vessels between the port region and Duisburg, and small vessels are used to commute Middle- and Upper Rhine regions. (van der Houwen) investigate the inland terminals to determine the potential location for intermodal consolidation where the TCT Venlo, TCT Belgium, DELETE Duisburg, Albersdam and Valburg terminals are considered for investigation. The study concludes that the potential region for intermodal terminal could be around Gorinchem and Moerdijk to serve the port of Rotterdam and Antwerp and its hinterland region. Authors (Konings et al., 2013) compared the hub location at Rotterdam Eem-/Waalhaven port area, Dordrecht (70 km from Rotterdam) and Nijmegen (135 km from Rotterdam). The conclusion from the research suggests that the cost performance was better when the hub location is far from the port region (at Nijmegen) as this creates more scope for economies of scale in the hub-seaport link. The hub considered at Nijmegen could serve both the port of Antwerp and Rotterdam region. Authors (A. Caris et al., 2011) compare the consolidation hub in the port region of the Antwerp and hinterland region of Antwerp. The port terminals in the Antwerp region are clustered into two clusters and the different port hub strategies of having the hub in the left cluster; right cluster and multi-cluster (both left and right cluster) are investigated. The results conclude that having the hub in both the left and right clusters of port terminals is the most interesting strategy. The conclusion from the research suggests that having a hub at the port area and operating inter-terminal shuttle in port for container consolidation is more beneficial.

The general pick-up and delivery problem with time windows and transshipment opportunities using other modes of transportation discussed in the literature is also investigated. Authors (Takoudjou et al., 2012), (Vornhusen et al., 2014) consider pick-up and delivery problem with one transshipment per cargo. Authors (Qu & Bard, 2012) provides a solution for pick-up and delivery problem with multiple transfers.

3.2.6. Objective functions

The next attribute discussed is the objective function considered in the literature. The objective function is the parameter that the problem owner is interested in optimizing. Most of the literature uses a cost-based objective function. The use of a cost-based objective is very common in liner shipping. Whereas, tramp shipping uses profit maximization objectives in many cases. There are few works of literature about another type of objective function for ship routing and scheduling problems such as distance, travel time, number of terminals visited and environmental impact. However, different KPI are indirectly considered while optimizing the main objective by adding a penalty factor to the main objective function. This KPI includes the utilization of the resource, reduction of turnaround time, an increase of service frequency, reduction of multiple terminal visits, reduction of late delivery and reduction of total travel distance. Few works of literature use the reduction of fuel consumption as well as reduction of emission from the fuel consumed during transportation as the objective.

3.3. Analysis of the solution methods used in literature

The solution approaches that are used in different works of literature are discussed in this section. The solution approaches are distinguished into the exact solution approaches, and the heuristics solution approaches for solving different routing and scheduling problems.

3.3.1. Exact solution approaches discussed in the literature

Authors (Christiansen et al., 2013) provide a mathematical model for network design problems for both liner and tramp shipping. Authors (Edirisinghe et al., 2010) and (Brønmo et al., 2007) provides the general formulation for the pick-up and delivery problem for shipping cargo. Authors (Qu & Bard, 2012), (Rais et al., 2014) and (Hagen, 2016) consider the pick-up and delivery problem with transshipment opportunities formulated using the mathematical model. Authors (Vornhusen et al., 2014) and (Takoudjou et al., 2012) consider a different mathematical formulation for the pick-up and delivery problem with single transshipment for a cargo. Authors (Fazi et al., 2015) and (Fazi, 2014) formulate a mathematical model for the Hinterland Allocation Problem (HAP) which is an adaptation of the pick-up and delivery problem where routing and scheduling of cargo from an inland terminal to a cluster of sea terminals are made for a planning time horizon with the cost-based objective function.

Authors (Davidovi, 2011), (Maraš, 2008), (Maraš et al., 2013) and (An et al., 2015) propose a Mixed integer linear programming (MILP) for the linear shipping operating between a port terminal and its hinterland terminals. The terminals are considered to be in series. The mathematical model is solved to obtain the optimal barge routing and scheduling plan to maximize the profit. A combinatorial formulation of the same problem is proposed with determining the calling sequence during the upstream and downstream of the barge trip in the liner route. Authors (Alfandari et al., 2019) adopts a tighter Mixed Integer Programming (MIP) model for the same problem with separate arc for full and empty container flow.

Author (Sharypova, 2014) models the direct service network design problem using a node and arc-based mathematical model which takes into account the cargo release date, the vehicle restriction to travel in a service route and synchronization of vehicle moves during the container transshipment. Author (Kjeldsen, 2012) considers a similar formulation with the restriction on the number of terminal visits without considering a planning horizon.

Authors (Lin & Liu, 2011) and (Stålhane, et al., 2014) provide a MIP formulation for tramp shipping operating with COA and Vendor Managed Inventory (VMI) contracts. Authors (Castillo-Villar et al., 2014) and (De Armas et al., 2015) provide a mathematical formulation for the tramp shipping problem considering a discretized time window where the nodes corresponding to possible pick-up time for contracts are considered. The ship should visit one or more terminals and transport the cargo from the origin to the destination terminal in a sequential manner (cargo is dropped before picking up the next contract cargo) with a cost minimization objective.

3.3.2. Heuristic solution approach discussed in the literature

The mathematical models that were used by different literature discussed in the previous section for the different routing and scheduling problems are hard to be solved to optimality for larger real-life problem instances. Hence, there are different heuristic solutions approaches used for solving them to near optimality, which is discussed in this section. Different metaheuristics and matheuristics discussed in different literature as solution approaches are explained below.

Authors (Foss et al., 2016) and (Archetti & Speranza, 2014) provide a general classification for different matheuristics available in solving the routing problem along with iterative matheuristics that use two phases for constructing and improving the solution. They also describe the different approaches that are available in matheuristics for the routing problem. Authors (Zweers et al., 2019) develop a two-stage Integer Linear Programming (ILP) based heuristic where the model is relaxed and solved in the first phase followed by solving the ILP without relaxation in the second phase. The solutions are compared to the planner's algorithm by mimicking the planner's logic to schedule the barges in real-life scenarios. Authors (Creemers et al., 2017) propose a matching algorithm where the transshipment terminals are identified based on the distance parameters. The matching algorithm uses a bounding box approach to confine the terminals to be analyzed with sorting techniques to find the attractive nearest terminal for cargo consolidation.

Authors (Hemmati et al., 2015) solves the pick-up and delivery problem of maritime cargo with a time window by using an Iterative Cargo Generating and Routing heuristic (ICGR). The heuristic is an adaptation of an Adaptive Large Neighborhood Search (ALNS) heuristic performed iteratively. K-means clustering algorithm is used to partition the cargo groups into clusters of similar groups. Later, ALNS is used to solve the selected potential cluster and reducing the solution time for large problems.

Authors (Van der Hagen, 2016) and (Masson et al., 2017) discuss an ALNS heuristic for solving the Pick-up and Delivery Problem with Time Window (PDPTW) and transfer opportunities. Authors (Qu & Bard, 2012) propose a Greedy randomized adaptive search procedure combined with ALNS (GRASP-ALNS) heuristic for solving the PDPTW with a transfer, which uses the GRASP heuristic for diversification and ALNS for intensification purposes. The above heuristic use different removal strategies, namely Shaw, random and worst-case removal and insertion strategies namely greedy, regret-k, random insertion, to create a neighbour solution.

Author (Pruijn, 2018) uses a priority matrix-based solution approach for container transportation in the hinterland region using liner shipping without time window. Authors (Castillo-Villar et al., 2014) use variable neighbourhood search heuristic (VNS) to solve the ship routing and scheduling problem in tramp shipping with variable speed and discretized time window. Authors (De Armas et al., 2015) solve the same problem with the help of GRASP heuristic for the construction phase and VNS for the improvement phase. It is inferred that the GRASP-VNS approach performs better concerning the time and solution quality.

Author (Sharypova, 2014) propose metaheuristics for the Scheduled Service Network Design problem (SSND) with synchronization for the transshipment of containers. The metaheuristics first try to solve the problem with dummy vehicles using a MILP to obtain a feasible solution. This is followed by an evaluation of neighbours using 1, 2 and 3 vehicles neighbour structure followed by the intensification and diversification of the search. There is a post-optimization search phase that is performed after the intensification and diversification phase to ensure that there is no better solution once the intensification and diversification phase stops. Author (Kjeldsen, 2012) solves the ship routing and scheduling problem with a column generation based heuristic which uses a separate set of columns for transshipment options.

Authors (Homayouni & Fontes, 2018) discuss the working of different general metaheuristics that are available for optimizing the maritime operations. They also use the Ant Colony Optimization (ACO) algorithm for solving a liner ship routing problem. Authors (Fazi & Roodbergen, 2018) use a hybrid simulated annealing method with tabu lists, run with constant temperatures in a multi-start approach to solving the ship routing and scheduling problem in liner shipping along with consideration of the detention and demurrage factors. Authors (Fazi et al., 2015) and (Fazi, 2014) consider the pick-up and delivery problem as a container consolidation problem and later use it to solve the hinterland allocation problem. The variable size bin packing problem with time constraint (T-VS-BPP) is formulated for container consolidation. The initial solution is generated by the First Fit Type Algorithm (FFTA) for the bin packing problem. Later the (Markov Chain Monte Carlo) MCMC algorithm which is similar to simulated annealing but with a fixed temperature is used to solve the hinterland allocation problem which is an adaptation of the pick-up and delivery problem. A multi-start local search is employed for generation of the neighbor.

Authors (Korsvik et al., 2010) consider a tabu search based heuristic for ship routing and scheduling problems. A continuous diversification mechanism, along with the possibility to consider an infeasible solution by violating the capacity and time window for generating the neighbour solution, to explore a large solution space. The heuristic also considers intra-route optimization, where the sequence of every ship is optimized individually. The final intensification phase consists of five local search operators and the results of the heuristic yield better solution compared to other multi-start local search heuristic.

Authors (An et al., 2015) uses a genetic algorithm (GA) to solve the service network design problem for liner transportation in inland waterways along with empty container repositioning. Authors (Lin & Liu, 2011) uses the GA for solving combined ship allocation, routing and freight assignment problems in tramp shipping. Authors (Danloup, Allaoui, & Goncalves, 2018) provides a comparison between Large Neighborhood Search (LNS) and the GA for solving the PDPTW with the transfer. It can be inferred from the results that both the heuristics provide a reasonably good solution for larger problem instances.

Authors (Caris, 2011) and (Caris et al., 2011) propose a hub-based container consolidation model for the port of Antwerp. A simulation model is built using the Arena simulation software to represent the operation in the hinterland of the Antwerp region and its port.

3.4. Performance evaluation techniques used in literature

After analyzing different solution methodologies, the approach that different literature uses to evaluate the performance of their solution method is analyzed. Authors (Van der Hagen, 2016) and (Masson et al., 2017) uses the benchmark instances for the PDPTW generated by (Li and Lim, 2003) to evaluate the solution quality. Authors (Qu & Bard, 2012) generated test instances for PDPT and used to evaluate the solution approach to solve the generated instances. Author (Sharypova, 2014) generates test instances by plotting the port terminal and inland terminal in the graph and generating the cargo demand. The above literature works use the gap value between the objective function of exact and heuristic method solutions for comparing the performance of the heuristic.

Author (Pruijn, 2018) uses real-life data generated from the dataset and compare the performance between the planner's algorithm and the priority-based algorithm. Authors (Fazi et al., 2015) consider the benchmark instances of VS-BPP for testing the performance of the heuristic for the bin packing problem. The demand data for three months of real-life demand from the inland terminal located in Veghel is used along with other cost values for actual analysis of the results.

Authors (Castillo-Villar et al., 2014) and (De Armas et al., 2015) consider tramp shipping with a discretized time window. The problem is evaluated using test instances that differ with contract size, number of ships considered and number of nodes per window. The literature work uses the same test instances that are generated by an instance generator for testing the performance. The comparison of the performance is made by computing the gap between the mathematical model solution and the heuristic solutions. Authors (Hemmati et al., 2014) provides a benchmark suite for industrial and tramp ship routing and scheduling problems. The instance generator which was developed as a part of the research is used to generate both deep sea and short sea instances. The option to vary the number and size of cargo generated, its time window, option to switch between balanced and not balanced regional demand and market conditions are also included to generate real-life instances.

3.5. Selection of solution approach

The problem identified is a pick-up and delivery problem with the consideration of transshipment opportunities. First, the notation of the research problem is provided, followed by the discussion of the selected solution method. From the problem context discussed in Chapter 2, it is evident that the problem deals with handling the deterministic demand for the container transportation by the barges of a heterogeneous fleet along with the transshipment opportunities during a planning period. The main objective of the solution should be towards the minimization of the total cost for container transportation. That being said, the notation misses one attribute value, which is the time dimension used in the solution method. There is a need for synchronization of the schedule of barges visiting the transshipment terminals, and most of the literature has handled this with the help of a continuous-time dimension. Hence, the decision to use a continuous dimension is taken.

The classification of the research problem studied in this thesis is:

deter | barge | hetero | cont | transfer | cost

The pick-up and delivery problem, along with transshipment opportunities, increase the complexity of the problem where a large solution space should be analyzed. From the solution methods used in the literature which are discussed above, it is evident that metaheuristics such as simulated annealing and tabu search techniques perform well for a reasonably large search space. For a very large solution space, LNS and ALNS techniques are well suited. The PDPTW with transshipment discussed in (Qu & Bard, 2012) uses a GRASP-ALNS heuristic to handle very large solution space. Hence, our decision to use the multi-start GRASP-ALNS similar to the heuristic proposed by (Qu & Bard, 2012) with the adaptation to the other business requirements requested by Cofano is considered as the selected solution method for our problem. A large amount of literature uses the comparison of gap value between the exact and heuristic solution approach to evaluate the performance of the solution approach. Hence, the decision to use the gap value between the exact and heuristic solution approach is selected as the performance evaluation technique.

3.6. Related works

This section compares the characters of the research problem with similar works found in the literature. Two works of literature were identified to be more related to the problem and solution approach defined in this thesis research.

Table 2: Comparison of the research problem with related works

Reference	**(Sharypova, 2014)**	**(Fazi, 2014)**	**Thesis research problem**
Scheduled service	Yes	No	No
Transportation mode	Barge	Barge	Barge
Type of Problem	SSND with transshipment	Hinterland allocation problem	PDP with transshipment
Transshipment	Yes	No	Yes
Minimum call size	No	No	Yes
Objective function used	Cost Minimization	Cost Minimization	Cost Minimization
Test instances	Test instances based on real-life data	Test instances based on real-life data	Test instances based on real-life data

Author (Sharypova, 2014) considers the SSND problem with synchronization and transshipment constraints for container transportation using barges in the hinterland. The problem considers scheduled services for barges between predefined terminal pairs. The synchronization of barge arrival and departure with regards to the transshipment of cargo are discussed as well. Author (Fazi, 2014) considers the Hinterland Allocation Problem (HAP) which is a variant of the pick-up and delivery problem for barges. Author (Fazi, 2014) do not discuss any transshipment opportunities in their work. All the containers are transported through direct shipment. The research problem can be considered similar to (Fazi, 2014) with additional transshipment opportunities, and similar to (Sharypova, 2014) but without a scheduled service where a barge can travel between any terminals in the service network without restriction. There is also a call size restriction considered in the research, referring to the minimum number of containers served by the barge during a terminal visit. The difference between the related works and the research problem is summarized in Table 2.

3.7. Conceptual framework

As explained in Section 1.5, the deliverable from the first phase of the research after performing the problem analysis and literature review is the conceptual framework. The conceptual framework is similar to the existing communication framework. The conceptual framework helps us to visualize the flow of inputs, outputs and decisions that should be considered for our solution approach. The conceptual framework explained in Figure 9 contains three elements. The first set of elements represent the input. This element corresponds to the input data fed to the scheduling algorithm. The second element represents the decisions that are to be taken on the input data. The third set of elements correspond to the output generated as a result of the decisions.

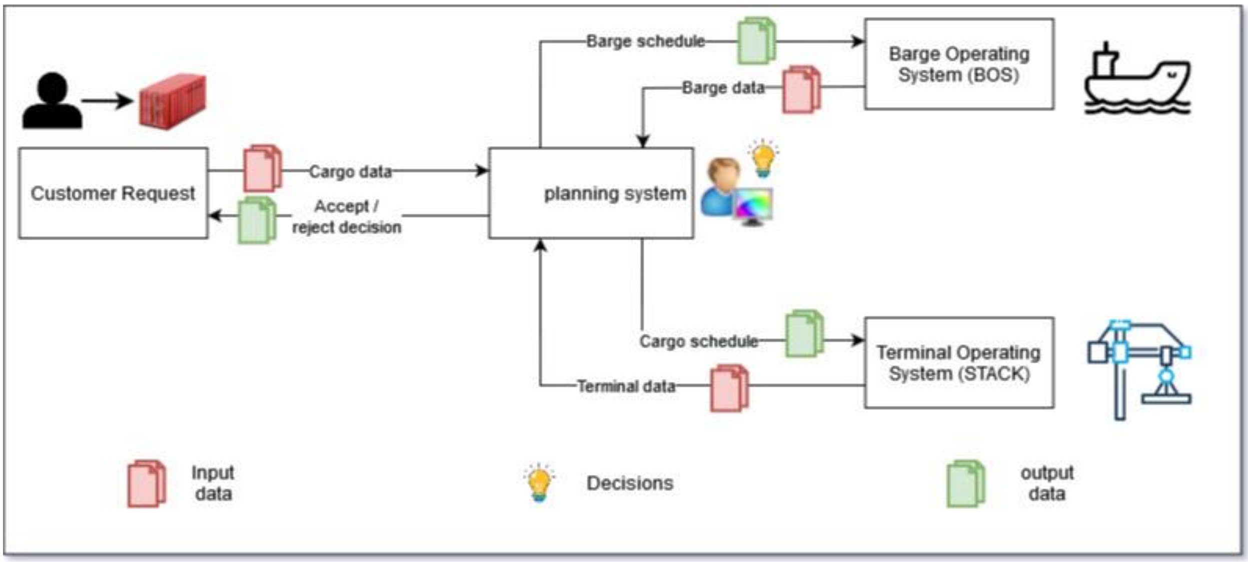

Figure 9: Conceptual solution framework

The input data provided from the terminal operating system contains details of the terminals that are considered during the planning horizon. The cargo data related to the container request are the input from the customer. The input data from the barge operating system contains the list of available barges, their capacity and other barge related data. The decision represents the solution from the algorithm that is to be designed as a part of the solution approach of the research. The algorithm uses GRASP-ALNS to decide on the selection of the best solution. The solution related to terminal visit sequence for each barge is provided as the output to the barge operating system. The decision of accepting or rejecting the container transportation request is provided to the customer as output. The solution related to the list of containers that are loaded/ unloaded from each terminal is sent as output to the terminal operating system.

3.8. Conclusion

The second set of research questions related to the solution approaches available in the current literature, and the process of selecting a solution approach for the research was answered in this chapter. The comparison of the pros and cons of different solution approaches were made. GRASP-ALNS heuristic was selected as the solution approach due to its capability for providing a high-quality solution within reasonable times. Related works that are similar to the research were identified and discussed. The conceptual framework has been developed as a result of the answers from the research question 1 and 2. The conceptual framework explaining the input, output and decision elements of the solution approach has been delivered as the result of the first phase of the research.

Chapter 4: Design and Implementation of Solution Approach

A formal introduction to the problem variant and a detailed explanation about the solution methodology is given in this chapter. Section 4.1 gives a brief introduction to the problem variant considered in the thesis. Section 4.2 includes the assumptions considered for modelling the research problem. After explaining the graphical representation of the node and arc formulation of the pick-up and delivery problem in Section 4.3, the mathematical model for the problem variant is discussed in Section 4.4. Section 4.5 explains the heuristic solution approach used for designing the algorithm. All the different components and strategies used for the algorithm design are explained in this chapter.

4.1. Pick-up and delivery problem with transshipment

The problem at hand is identified as a pick-up and delivery problem, which is an extension of the vehicle routing problem, where a set of cargo needs to be transported from origin location to destination location. There are different variants of the pick-up and delivery problem explained by (Battarra, 2014) such as:

- One-to-many-to-one (1-M-1), where cargo is transported from a depot to many customers and back from multiple customers to a depot.
- One-to-one (1-1), where the cargo is transported from a depot to a customer and from a customer back to the single depot.
- Many-to-many (M-M), where the cargo may originate from different origin points and delivered to different delivery points.

The research problem belongs to the many-to-many (M-M) category where there is no single fixed origin or destination point for representing the depot or customer location. Any terminal can act as an origin as well as a destination location.

The pick-up and delivery problem has different variant based on the ways the cargo is handled as explained by (Battarra, 2014) such as:

- Handling of a cargo transportation request by one vehicle from the point of pick-up until the point of delivery referred to as direct shipment.
- Handling of a cargo transportation request by more than one vehicle which is referred to as transshipment.
- Splitting of single shipment into different smaller shipments and transporting it in multiple vehicles referring to the split load.

The problem considered in this chapter is an M-M variant of pick-up and delivery problem with the transshipment (PDPT). The problem is defined for a service network that includes a set of loading and unloading terminals belonging to different regions. There is a set of container transportation requests that arises during a planning period. The containers should be picked up from different origin locations *O* and delivered to different destination locations *D*. The loading and unloading operations are performed at the respected terminals. There is an *O-D* pair representing the origin and destination location for each container request. There is no differentiation that a terminal is dedicated only for loading or only for unloading in the many-to-many variant. In the M-M variant of the PDP, individual cargo pick-up and destination location are unique. The hinterland case differs from the traditional M-M variant problem with respect to the terminal location. Multiple cargos can originate from the same loading location and can be destinated to a different destination. Cargos loaded from different loading location can be destinated to the same destination as well. The container request is always transported from a loading terminal to a different unloading terminal. Hence, a vehicle can visit a terminal multiple times to handle a set of container requests.

The next consideration is the barges which serve as the carriers for the containers between the terminals. There is a set of barges available during the considered planning horizon that can travel between any terminal pair. The goal of the solution approach is to find an optimal way of assigning the containers to the barges and to operate these barges between different terminals to load/pick-up the containers from their origin terminal and unload/ drop the containers at the destination. The objective of the problem is to minimize the total cost of transportation.

The mathematical model considered for this research is an adaptation of the PDPTW with transshipment proposed by (Qu & Bard, 2012). In addition to the constraints of the general pickup and delivery problem with transshipment there are additional constraints added to the PDPTW model proposed by (Qu & Bard, 2012). The distinction of the model proposed for this research compared to the model proposed in the literature are as follows.

- The variant of the PDP problem discussed in the literature is designed with a time window for pick-up and delivery points. The research problem considers a due date constraint, where, failing to deliver a container before the due date is subjected to penalty based on the lateness in delivery.
- The literature model considers the delivery of all the container request considered for the planning horizon. The option to either transport a container request or rejecting a container request is incorporated in the proposed research model.
- The M-M variant of the PDP considers unique pick-up and delivery location for handling each of the container requests. The research problem considers the terminals that can be visited multiple times for both pick-up and delivery operations. Hence, a penalty based on the number of visits is imposed in order to limit the number of terminal visits made by the barges.
- There is a restriction on the maximum number of transshipments allowed per container request in addition to the constraints used for the variant of PDPTW with transshipment.
- The heuristic solution approach of this research considers a minimum call size restriction for the number of containers handled during a terminal visit. This restriction ensures that there are always a certain number of containers handled during each terminal visit. The minimum call size restriction is not discussed in any of the literature identified during the literature study.

4.2. Assumptions considered for modelling

The problem considered in this research is a PDPT for barges. The objective is to transport the containers either by direct shipment or by transshipping through transshipment terminals using the available barges. Several assumptions are considered to simplify the modelling of the problem and structuring it, such as:

- There are *n* container transportation requests of standard size (TEU or FEU representing 20 or 40 feet equivalent unit) available to be transported during the planning horizon.
- There are *t* terminals involved in the service network without storage and berth capacity restriction.
- There are *b* barges available with different capacity, operating between *t* terminals.
- The container can be transhipped only once per transshipment terminal. There is a limit for the maximum number for transshipment that can be allowed per container request before it reaches the destination.
- There are two cost factors considered for the terminals. First is a fixed cost factor for a barge to enter a terminal. Second is a variable cost for handling containers based on the number of containers handled in each terminal during the barge visit.
- There are two cost factors considered for the barge operation. First is a fixed cost for using a barge during a planning horizon. Second is a variable cost for operating the barges, based on the distance travelled.
- Penalty cost is considered for not delivering the container as well as for late delivery of the container to the destination beyond the due date.

4.3. Graphical representation of the PDPT

The PDPT can be described in a graphical representation with a graph $G = (N, A)$ where N represents the set of all nodes considered and A represents the set of all arcs connecting each node pairs. We consider the set of all terminals in the service network as $T = \{1, 2 \ldots t\}$ and the set of barges considered during the planning horizon as B = {1, 2... b}. We define n to be the number of container transportation requests considered during the planning horizon. Hence, there is n number of O-D pairs considered in the node-set, where $O = \{1, 2 \ldots n\}$ represents the set of origin nodes of the container request and $D = \{n+1, n+2 \ldots 2n\}$ represents the destination nodes. The arc $i \rightarrow j$ connecting node i and j such that $i, j \in N$ denotes that the node j is visited after serving the node i. If the same barge visits a pick-up node i and delivery node j for a container request, the transportation is considered as direct shipment.

The next type of transportation is transshipment, where two or more barges transport the container with the help of a transshipment terminal. The first barge picks up the container from the origin terminal and drops at a transshipment terminal. A different barge handles the container from the transshipment terminal to another transshipment terminal or to its destination terminal. The transshipment drop refers to the unloading action of a container at an intermediate transshipment terminal. The nodes corresponding to the transshipment drop are defined as the transshipment drop nodes represented by set $TD = \{(2n+1), (2n+2) \ldots (2n+nt)\}$. The transshipment pick-up refers to the loading of a container in a barge at a transshipment terminal. The nodes corresponding to the transshipment pick-up are defined as transshipment pick-up nodes represented by set *TP = {(2n+nt+1), (2n+nt+2) … (2n+2nt)}*. Transshipment pick-up for a container can occur after the respected container is dropped in a transshipment drop terminal. Hence, we have a transshipment pick-up *TP* and transshipment drop *TD* node pair for each container request at each terminal. We define the set of transshipment pick-up nodes corresponding to an individual container request *i* at a terminal *a* as $TP_i = i + a * n + (t + 1) * n$ where $i \in O$ and $a \in T$. Similarly, the set of all transshipment drop nodes for a container request *i* is represented as $TD_i = i + a * n + n$ where $i \in O$ and $a \in T$. Hence, the set of all transshipment pick-up nodes are grouped as $TP = TP_1 \cup TP_2 .. \cup TP_n$. The set of all transshipment drop nodes are denoted as $TD = TD_1 \cup TD_2 .. \cup TD_n$.

We define the starting terminal for all the barges as a set *R*. The nodes corresponding to the starting position of the barges are defined as *R = {(2n+2nt+1), (2n+2nt+2) … (2n+2nt+b)}*. The end destination of the barges is an artificial terminal which is the last node that is visited by the barges during the voyage. The index of the node is represented as *E = {2n+2nt+b+1}*. The distance between any terminal in the service network and the artificial end terminal is zero. This artificial terminal with a zero distance is considered due to the assumption that the barge need not end its voyage at any specific terminal.

The set *N* consist of all the set of nodes defined above *N = O ∪ D ∪ TD ∪ TP ∪ R ∪ E*. Each node has a terminal location mapped to it as explained above. The arc between any two nodes represents the movement of the barge from the first node to the second node. For instance, the arc between a node pair *i* → *j* represents the movement of the barge from the terminal of node *i* to the terminal of node *j*. However, certain arcs do not make sense such as the arc from an end node *E* to any other node as it is not logical for a barge to move from the final terminal to any point further. Such arcs are excluded by considering the valid inequalities in the mathematical model.

The distance between two nodes is considered to be the distance between the corresponding terminals of the nodes. If two nodes in a node pair are said to be from the same terminal, then the distance between the nodes is considered as zero. Each node has a load value assigned to it. The load value is either positive or negative, depending on the node. If a node is a loading/ pick-up node, then the load value is considered as positive (i.e., all the origin and the transshipment pick-up node for a container request have a positive load value). If a node is an unloading/ drop node, then the load value is considered as a negative (i.e., all the destination and the transshipment drop node for a container request have a negative load value).

Each container request has an arrival time after which it is available at the origin terminal for transportation and due date before which the container request should reach the destination terminal. The arrival time is a hard constraint. No origin, transshipment and destination nodes for a container request can be visited before the arrival time. The due date is a soft constraint. A container can be delivered to the destination node even after the due date. However, there will be a penalty for delivering a container request after the due date, depending on the lateness.

The barges used for transportation are considered to be heterogeneous; they vary in the capacity of cargo each barge can handle and speed of travel. There is a maximum capacity limit for each barge above which the containers cannot be loaded. Hence, the capacity restriction is a hard constraint and cannot be violated.

Now, we discuss the decision variables that are used in the mathematical model. The first decision variable is a binary variable which denotes the sequence of nodes that are visited by each barge. X_{ijk} is equal to 1 if a barge k travels from node i to node j where $k \in B$ and $i, j \in N$. The next decision variable Y_{ik} is also a binary variable whose value is equal to 1 if the node i is visited by barge k where $k \in B$ and $i \in N$. The value is zero otherwise. Any node can be visited only once in the solution except the imaginary end terminal node where all the barges end its voyage.

The next decision variable H_i is a binary variable whose value is 1 if a container request i is considered to be transported in any of the barges. The value is zero if the request is not transported. When the barge service does not transport a container request, then it is assumed to be transported by a truck or any other service provider, and the penalty for doing so is imposed upon not transporting it. The decision variable L_{ik} represents the load of containers that are available in the barge k when it completes the service at node i. When a barge visits a node, it means that the operation corresponding to the node is being performed (i.e., loading/ unloading of the cargo at origin/ destination or transshipment terminals). Hence, the total load in a barge should not exceed the capacity of the barge. The last variable corresponds to the service start time T_i at node i. It is assumed that the starting terminal for the barge is the corresponding node in set R with the service start time equal to zero for these nodes.

4.4. Example for graphical representation of the PDPT

We consider an example problem where there are three barges B = *{1, 2, 3}*, four terminals represented by the set *T = {1=A, 2=B, 3=C, 4=D}*, and 9 container request to be transported during a planning horizon. Each container request has an origin and destination node. The set of origin nodes are *O* = {1, 2, 3, 4, 5, 6, 7, 8, 9} and the destination nodes are *D* = {10, 11, 12, 13, 14, 15, 16, 17, 18}. There are three starting nodes for barges namely *R = {r1 = 91, r2 = 92, r3 = 93}* and a destination node for barge is E = {94}.

For each container request and each terminal, there is a node pair representing the transshipment pick-up and transshipment drop operations. The transshipment drop nodes for container request 1 at terminal A is $TD_{1A} = 19$ and transshipment pick-up node for container request 1 at terminal A is $TP_{1A} = 55$. Hence, the set of transshipment drop nodes for container request 1 corresponding to terminal *{A, B, C, D}* is represented as $TD_1 = \{19, 28, 37, 46\}$. Similarly, the transshipment pick-up nodes for container request 1 corresponding to terminal *{A, B, C, D}* are represented as $TP_1 = \{55, 64, 73, 82\}$. The node-set *TP = {55, 56 …90}* contains all the transshipment pick-up nodes for all the container requests at all terminals and *TD = {19, 20 …54}* contains all the transshipment drop nodes for all the container request at all the terminals. The visualization of the nodes is explained in Figure 10.

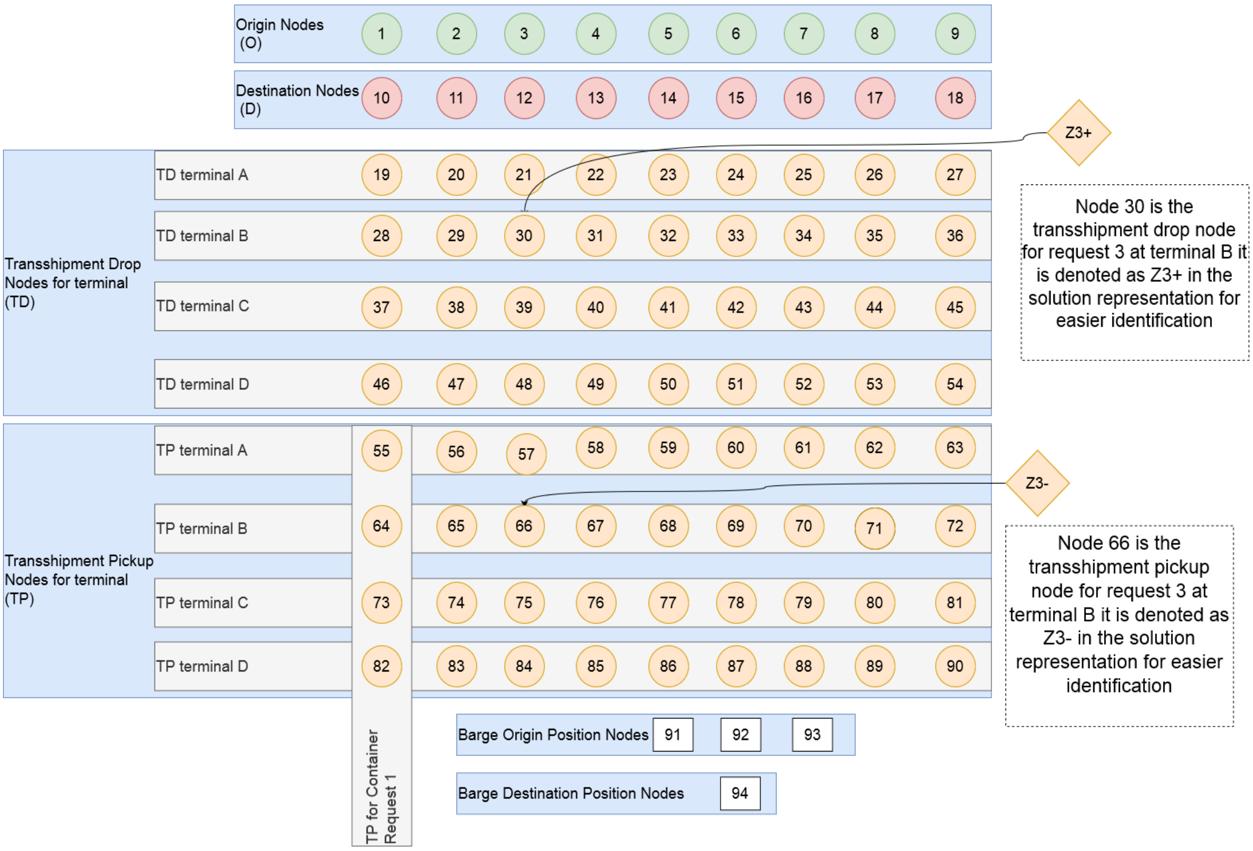

Figure 10: list of nodes consider in graph G

Now, we consider a solution sequence and represent it with the node and arc formulation. For easy visualization of the transshipment scenario, request 3 is only considered to be transshipped in the solution.

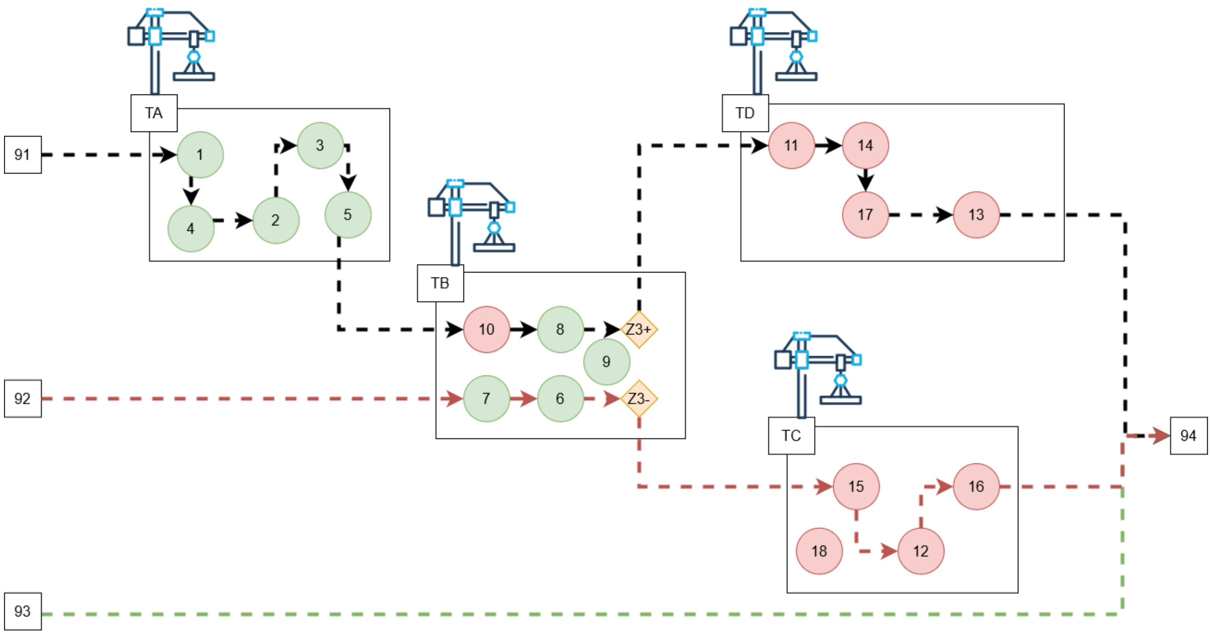

Figure 11: Visualization of the solution route for barges through terminals

The route taken by different barges is denoted in different colours. Each barge starts its voyage at its start terminal nodes {91, 92 and 93} and ends the voyage at the common final destination node (94). A barge picks up and drops container requests along its route. The pick-up nodes are denoted as green, and the drop nodes are denoted as red inside the box that represents the respected terminal to which they belong. It can be inferred that barge three starts its voyage at node 93 and ends at node 94, without transporting any containers during the journey. The transshipment for container request 3 occurs at terminal B. Hence, the transshipment pick-up node for container request 3 at terminal B (node 66 represented as Z3-) and the transshipment drop node for the container request 3 at terminal B (node 30 represented as Z3+) are only included (orange diamond nodes) in Figure 11. Barge one picks up the container request three from its origin terminal TA (node 3) and drops in transshipment terminal *TB* (node *Z3+)*. Barge two picks up the container request three from the transshipment terminal *TB* (node *Z3-)* and drops at the destination terminal TC (node 12).

Now, we visualize the sequence of the nodes visited by each barge separately in Figure 12. The decision variable denoting the course of the visit of barges (e.g., $X_{ijk} = 1$ where i denotes the first node in the node pair, j denotes the second node in the node pair, and k denotes the barge number) is indicated as connection arcs between the node pairs. It can be noted that the service start time of the node *Z3-* is after the service completion of the *Z3+* indicating that the transshipment pick-up operation for request three at terminal *B* happens after the container request three has been dropped in transshipment terminal by the first barge. It can be inferred that any destination node can only be visited after the service at origin and all the transshipment nodes are completed. For instance, node 10 (destination node for container request 1) is visited after the service completion of node 1.

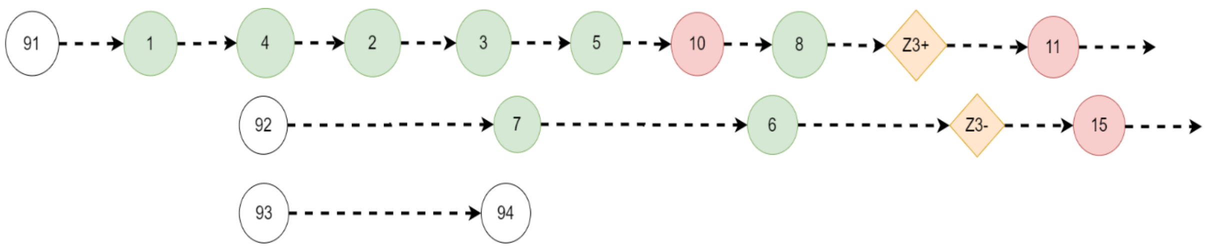

Figure 12: Visualization of individual solution route sequence and nodes visited during the route

The course of the terminals visited by each barge based on the inference from the solution considered previously is visualized in Figure 14. The route of each barge is differentiated with a different colour in Figure 14.

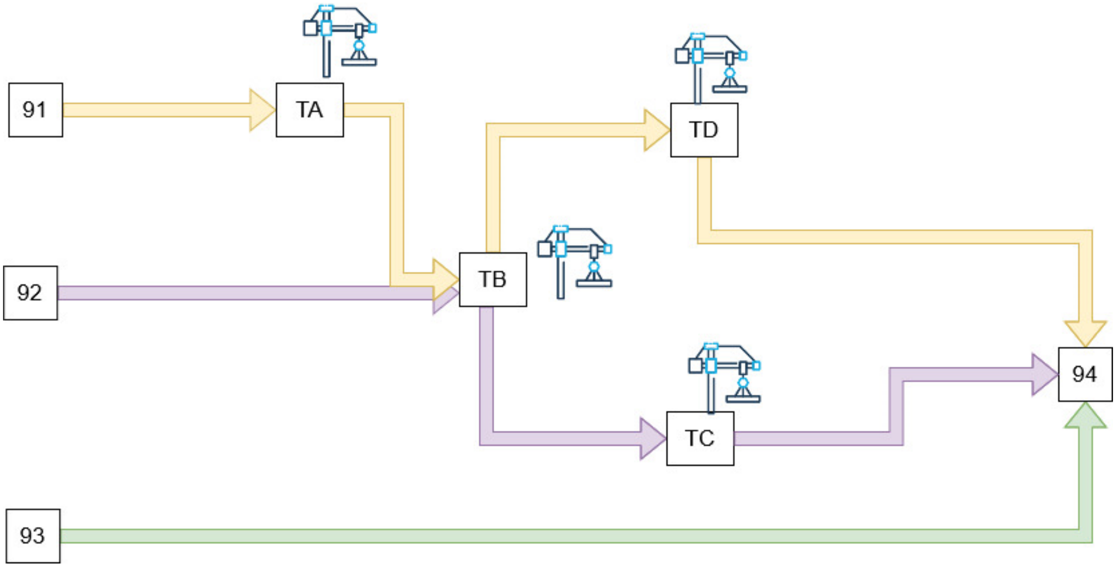

Figure 13: Visualization of sequence of terminals visited by each barge

The last visualization represents the various operations that are happening in the terminal B. The nodes and transshipment operations are explained in detail in Figure 14. The route of barge one is represented with orange arrows, and the black line represents the sequence of nodes visited by barge one. The route of barge two is represented by a purple arrow, and the red line represents the sequence of nodes visited by barge two. The pick-up node for container request nine is not visited by any barge indicating that the request is not transported in any of the barge schedules.

The cost for the solution represented in the example is computed based on different parameters such as the number of barges used, route the barges travel, the terminals visited by each barge, the containers handled by the barges at various terminals and the penalty for late delivery of container requests. The breakdown of the total cost function is explained with the objective function in the next section.

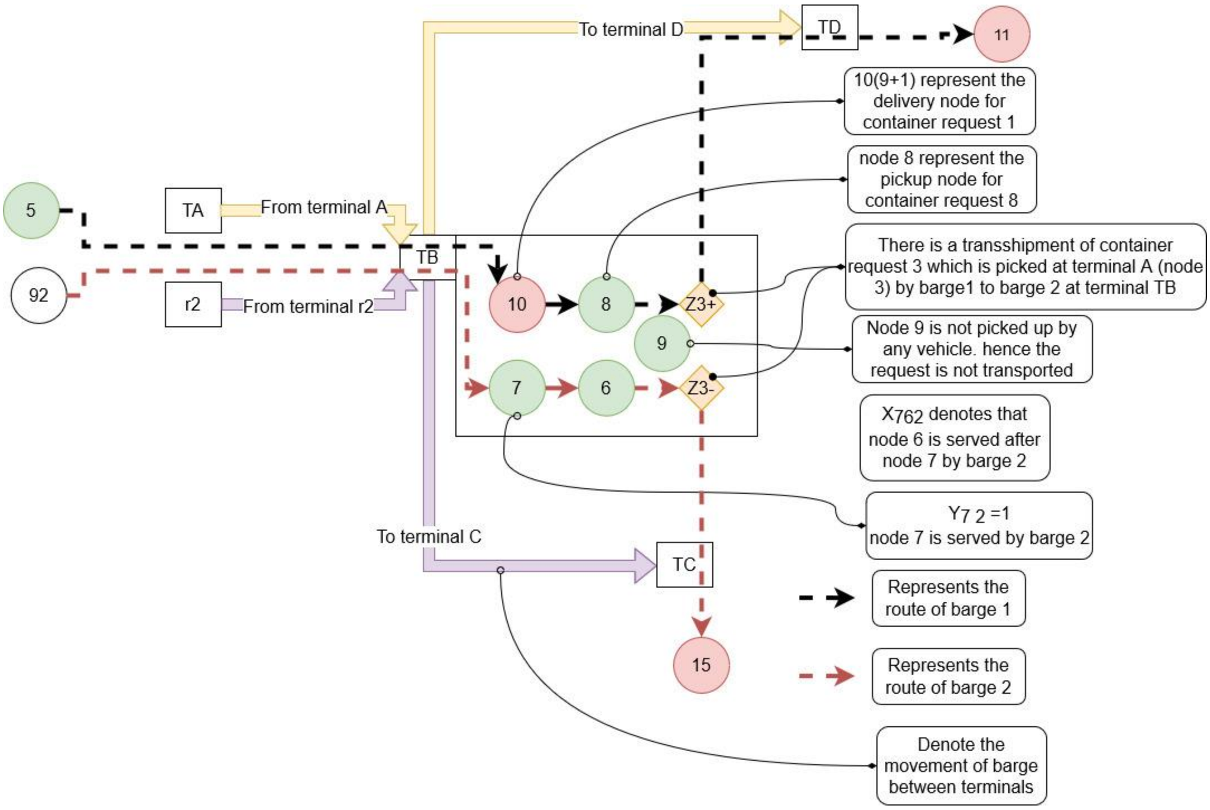

Figure 14: Visualization of operations performed at terminal B by barge 1 and barge 2

4.5. Mathematical model

This section discusses the node and arc-based formulation of a mathematical model for the PDPT concerning graph *G = (N, A)* explained earlier. First, the indices and the sets used in the mathematical model are defined. Later, the parameters considered in the mathematical formulation are emphasised. Finally, the decision variables used in the model are explained. The mathematical model is provided by defining the objective function, followed by the definition of the constraints used. The inequalities are described towards the end, along with the limits for different variables used. The explanation about the objective function and the constraints are provided after the mathematical formulation.

Indices and set:

n Index for the number of container transportation request during the planning horizon
b Index for the number of barges available during the planning horizon
t Index for the number of terminals considered in the service network

O *{1, 2 … n}* Set of all nodes denoting the origin of the container transportation request
D *{n+1, n+2 … 2n}* Set of all nodes denoting the destination of the container transportation request
TD_{ia} $\{(i + a * n + n)\}$ Transshipment drop node for container request *i* at the terminal *a*
TP_{ia} $\{(i + a * n + (t + 1) * n)\}$ Transshipment pick-up node for container request *i* at the terminal *a*
TD_i $\{TD_{i1}, TD_{i2} \dots TD_{it}\}$ Set of transshipment drop nodes for container request *i* at all the terminals

TP_i $\{TP_{i1}, TP_{i2} \dots TP_{it}\}$ Set of transshipment pick-up nodes for container request i at all the terminals
TD *{2n+1, 2n+2,....(2n+nt)}* Set of transshipment drop off node for all terminals and container requests $= \{TD_1, TD_2 \dots TD_n\}$
TP *{(2n+nt)+1, (2n+nt)+2....(2n+2nt)}* Set of transshipment pick-up node for all terminals and container requests $=\{TP_1, TP_2 \dots TP_n\}$
C *{1, 2 … (2n+2nt)}* Set of all container nodes corresponding to origin, destination and transshipment nodes = *(O U D U TD U TP)*
R *{2n+2nt+1, 2n+2nt+2,.. 2n+2nt+b}* Set of starting location for barges = *{r1, r2 … rb}*
E *{2n+2nt+b+1}* Destination location for barges
N *{O U D U TD U TP U R U E}* Set of all nodes = *{C U R U E}*
T *{1, 2 … t, t+1}* Set of all terminals (t+1 denotes the artificial end terminal in the service network)
B *{1, 2 … b}* Set of all barges available

Parameters:

Parameter	*Description*	*Data Type*
$node\ terminal_i$	Terminal number corresponding to the node $i \in N$	Integer
$dist\ \ terminal_{gh}$	Distance between the terminal $g \in T$ and terminal $h \in T$	Integer
hub_t	1 if terminal $t \in T$ is a transshipment terminal 0 otherwise	Binary
cap_k	Capacity of barge $k \in B$	Integer
$fixed\ cost\ barge_k$	Fixed cost of operating barge $k \in B$	Integer
$var\ cost\ barge_k$	Variable cost for barge k $\in B$	Integer
$cost\ _{terminal\ entry}$	Fixed cost for entering a terminal	Integer
$cost_{terminal\ handling}$	Cost for handling unit container at a terminal	Integer
$penalty\ _{late}$	The variable cost for a unit time late delivery of container	Integer
$penalty_{no\ delivery\ i}$	Fixed cost for not transporting container request $i \in O$	Integer
$unit\ travel\ time_k$	Time taken to travel unit distance by barge $k \in B$	Integer
$is\ different\ terminal_{ij}$	1 if node $i \in N$ and $j \in N$ belong to a different terminal 0 otherwise	Binary
$dist_{ij}$	Distance between terminal of node $i \in N$ and terminal of node $j \in N$	Integer
β	Maximum number of transshipment allowed per request	Integer
s	Service time to handle a container at a terminal	Integer
l_i	Load of the container handled at node $i \in N$	Integer
$arrival_i$	Arrival time of container transfer request $i \in O$	Integer
dd_i	The due date for the container transfer request $i \in O$	Integer
M	Big integer value	Integer

The parameter $node\ terminal_i$ and $dist\ \ terminal_{gh}$ are used to compute the parameter $dist_{ij}$. Hence, these two parameters are not used in the mathematical model directly.

Variables:

Variables	*Description*	*Data Type*
H_i	1 if container request originating at node $i \in O$ is not transported 0 otherwise	Binary
L_{ik}	Loads present in barge $k \in B$ after completing the service at $i \in N$	Integer

X_{ijk}	1 if barge $k \in B$ travels from node $i \in N$ to node $j \in N$ 0 otherwise	Binary
Y_{ik}	1 if barge $k \in B$ visit node $i \in C$ 0 otherwise	Binary
T_i	Start time of service for a node $i \in N$	Integer
$Nr\ Visit_k$	Number of terminal visits made by barge $k \in B$ during its voyage	Integer
B_i	1 if time difference between actual delivery and due date is positive $i \in O$ 0 otherwise for	Binary
$Late_i$	Delay in delivery time for request $i \in O$	Integer

The variable $Late_i$ is zero if the delivery is before the due date. $Late_i$ is a positive value which is equal to the difference between actual delivery time of a container and due date if the container request is delivered after the due date.

Objective Function

$$Minimize\ \sum_{k\in B}\sum_{i\in R}\sum_{j\in C} X_{ijk} * fixed\ cost\ barge_k + \sum_{k\in B}\sum_{i\in N}\sum_{j\in N} X_{ijk} * var\ cost\ barge_k * dist_{ij} + \sum_{k\in B}\sum_{i\in C} Y_{ik} * cost_{\text{terminal handling}} + \sum_{k\in B} Nr\ Visit_k * cost_{\text{terminal entry}} + \sum_{i\in O} H_i * penalty_{No\ delivery\ of\ i} + \sum_{i\in O} Late_i * penalty_{late} \quad (1)$$

Container handling constraint

$$\sum_{k\in B} Y_{ik} + H_i = 1 \qquad \forall i \in O \qquad (2)$$

Barge flow constraint

$$\sum_{j\in N} X_{r_k jk} = 1 \qquad \forall k \in B \qquad (3)$$

$$\sum_{i\in N} X_{iek} = 1 \qquad \forall k \in B \qquad (4)$$

$$\sum_{i\in N} X_{irk} + \sum_{j\in N} X_{ejk} = 0 \qquad \forall k \in B, \forall r \in R \qquad (5)$$

$$\sum_{i \in N} X_{ijk} - \sum_{i \in N} X_{jik} = 0 \qquad \forall k \in B, \forall j \in C \qquad (6)$$

$$X_{iik} = 0 \qquad \forall i \in N, \forall k \in B \qquad (7)$$

$$\sum_{i \in N} X_{ijk} = Y_{jk} \qquad \forall k \in B, \forall j \in C \qquad (8)$$

$$\sum_{k\in B} Y_{ik} \le 1 \qquad \forall i \in C \qquad (9)$$

$$\sum_{k\in B} Y_{ik} = \sum_{k\in B} Y_{i+nk} \qquad \forall i \in O \qquad (10)$$

Barge capacity restriction constraints

$$L_{ik} = 0 \qquad \forall k \in B, \forall i \in R\ U\ E \qquad (11)$$

$$L_{jk} \ge (L_{ik} + l_j) - M(1 - X_{ijk}) \qquad \forall k \in B, \forall i \in N, \forall j \in N,\ i \ne j \qquad (12)$$

$$L_{jk} \le M * \sum_{i\in N, i \ne j} X_{ijk} \qquad \forall k \in B, \forall j \in N \qquad (13)$$

$$L_{jk} \le (L_{ik} + l_j) + M(1 - X_{ijk}) \qquad \forall k \in B, \forall i \in N, \forall j \in N, i \ne j \qquad (14)$$

$$0 \le L_{ik} \le cap_k \qquad \forall k \in B, \forall i \in N \qquad (15)$$

Container arrival time restriction and lateness in delivery constraints

$arrival_{\ i} * Y_{ik} \leq T_i$ $\forall i \in O,\ \forall k \in B,$ (16)

$Late_i \geq (T_{i+n} + s - dd_i) - M * (1 - B_i)$ $\forall i \in O$ (17)

$Late_i \leq M * B_i$ $\forall i \in O$ (18)

$Late_i \leq (T_{i+n} + s - dd_i) + M * (1 - B_i)$ $\forall i \in O$ (19)

$(T_{i+n} + s - dd_i) \geq 0 - M * (1 - B_i)$ $\forall i \in O$ (20)

$(T_{i+n} + s - dd_i) < M * B_i$ $\forall i \in O$ (21)

Container flow continuity and transshipment continuity constraints

$Y_{ik} \leq \sum_{j \in TD_i} Y_{jk} + Y_{i+n\,k}$ $\forall i \in O, \forall k \in B$ (22)

$Y_{i+nk} \leq \sum_{j \in TP_i} Y_{jk} + Y_{ik}$ $\forall i \in O, \forall k \in B$ (23)

$\sum_{j \in TD_i} Y_{jk} \leq Y_{ik} + \sum_{j \in TP_i} Y_{jk}$ $\forall i \in O, \forall k \in B$ (24)

$\sum_{j \in TP_i} Y_{jk} \leq Y_{i+nk} + \sum_{j \in TD_i} Y_{jk}$ $\forall i \in O, \forall k \in B$ (25)

$\sum_{k \in B} Y_{TD_{it}\,k} = \sum_{k \in B} Y_{TP_{it}\,k}$ $\forall i \in O, \forall t \in T$ (26)

$\sum_{k \in B} Y_{TD_{it}\,k} \leq 1$ $\forall i \in O, \forall t \in T$ (27)

$\sum_{j \in TD_i} \sum_{k \in B} Y_{jk} \leq_{\beta}$ $\forall i \in O$ (28)

$\sum_{i \in O} \sum_{k \in B} Y_{TD_{it}\,k} \leq M * hub_{\ t}$ $\forall t \in T$ (29)

Time continuity constraints

$T_i = 0$ $\forall i \in R$ (30)

$(T_i + s + unit\ travel\ time_k * Dist_{ij}) - T_j \leq M * (1 - X_{ijk})$ $\forall j \in N, \forall i \in N/R$ $i \neq j,$ $\forall k \in B$ (31)

$(T_i + unit\ travel\ time_k * Dist_{ij}) - T_j \leq M * (1 - X_{ijk})$ $\forall j \in N, \forall i \in R$ $i \neq j,$ $\forall k \in B$ (32)

$T_j \leq M * \sum_{\forall k \in B} \sum_{i \in N} X_{ijk}$ $\forall j \in N$ (33)

$\left(T_{TD_{it}} + s\right) - T_{TP_{it}} \leq M * \left(1 - Y_{TD_{it}k}\right) + M * (1 - Y_{TP_{it}\,s})$ $\forall t \in T, \forall k, s \in B, \forall i \in O$ (34)

$(T_i + s) - T_{TD_{it}} \leq M * \left(1 - Y_{TD_{it}s}\right) + M * (1 - Y_{ik})$ $\forall t \in T, \forall k, s \in B, \forall i \in O$ (35)

$T_{TD_{it}} \leq M * \sum_{k \in B} Y_{TD_{it}\,k}$ $\forall t \in T,\ \forall i \in O$ (36)

$\left(T_{TP_{it}} + s\right) - T_{i+n} \leq M * \left(1 - Y_{TP_{it}s}\right) + M * (1 - Y_{i+nk})$ $\forall t \in T, \forall k, s \in B, \forall i \in O$ (37)

$T_{TP_{it}} \leq M * \sum_{\forall k \in B} Y_{TP_{it}\,k}$ $\forall t \in T,\ \forall i \in O$ (38)

$\left(T_{TP_{ih}} + s\right) - T_{TD_{it}} \leq M * \left(1 - Y_{TD_{it}k}\right) + M * \left(1 - Y_{TP_{ih}k}\right)$ $\forall t, h \in T, \forall k \in B, \forall i \in O$ (39)

$Nr\ Visit_k = \sum_{i \in N} \sum_{j \in N} is\ different\ terminal_{ij} * X_{i\,j\,k}$ $\forall k \in B$ (40)

Invalid arcs constraints

$X_{i+n\,i\,k} = 0$ $\forall i \in O, \forall k \in B$ (41)

$\sum_{j \in TP_i} X_{j\,i\,k} = 0$ $\quad \forall\, i \in O, \forall\, k \in B$ (42)

$\sum_{j \in TD_i} X_{j\,i\,k} = 0$ $\quad \forall\, i \in O, \forall\, k \in B$ (43)

$\sum_{j \in TP_{i-n}} X_{i\,j\,k} = 0$ $\quad \forall\, i \in D, \forall\, k \in B$ (44)

$\sum_{j \in TD_{i-n}} X_{i\,j\,k} = 0$ $\quad \forall\, i \in D, \forall\, k \in B$ (45)

$\sum_{j \in TP_i} X_{i\,j\,k} = 0$ $\quad \forall\, i \in O, \forall\, k \in B$ (46)

$\sum_{j \in TD_{i-n}} X_{\,j\,i\,k} = 0$ $\quad \forall\, i \in D, \forall\, k \in B$ (47)

$X_{\,TP_{it}\,TD_{it}\,k} = 0$ $\quad \forall\, i \in O, \forall\, t \in T\,, \forall\, k \in B$ (48)

$\sum_{h \in T} X_{\,TD_{it}TP_{ih}\,k} = 0$ $\quad \forall\, i \in O, \forall\, t \in T\,, \forall\, k \in B$ (49)

Variables Limit

$X_{ijk},\ Y_{ik}\,,\ H_i,\ B_i\ \in\ \{0,1\}$ *Binary variables* (50)

$T_i,\ L_{ik},\ Late_i,\ Nr\ Visit_k \quad \geq 0$ *Positive integer variables* (51)

The objective function (1) minimizes the total cost of transportation of all the containers during the planning horizon. The total cost is composed of three cost factors namely cost related to barge operation (fixed barge transportation cost + variable barge transportation cost), cost related to terminal operation (container handling cost at a terminal +terminal visiting cost) and cost related to penalty values (the penalty for not delivering the container + penalty for late delivery).

Constraint (2) ensures that every container request is handled. The container requests are either accepted to be transported by any barge or rejected. Constraints (3) and (4) ensure that each barge starts from the starting terminal node and ends at the common ending terminal node. Constraints (5), and (7) restrict the invalid arcs that connect the start terminal and end terminal nodes with other nodes that are logically not possible (i.e., connecting end terminal node to other nodes, connecting any other node to a start terminal node, etc.). Constraint (6) ensures the flow continuity between the nodes for individual barges. Constraints (8) and (9) ensure all nodes should be visited only once. Constraint (10) ensures that, if a container request is picked up, then it should be delivered, but not necessarily by the same barge. Constraints (11) – (15) compute the load on a barge after completing the service at a node and ensures the load on any barge is not greater than the capacity limit of the barge. Constraint (16) ensures that all containers should be served only after the arrival time. Constraints (17) - (21) compute the lateness value for the delay in delivering a container to its destination beyond the due date. Constraints (22) and (23) ensure that if a container request is picked up from the origin terminal, then it should be dropped at the destination by direct shipment or dropped at a transshipment terminal to be transshipped later. Constraints (24) and (25) ensure continuity between nodes during transshipment. Constraint (26) ensures continuity of a container request in a transshipment terminal. Constraints (27) and (28) restrict the number of transshipments per terminal and the number of transshipments per container request. Constraint (29) ensures that the transshipment operation can take place only at the transshipment terminal. Constraints (30) – (33) ensure the time continuity between two nodes that are connected by an arc. Constraints (34) – (39) ensure the precedence relation concerning the service start time for the transshipment operations at transshipment terminals. The precedence relations considered are as follows:

- All the transshipment operations should happen between the pick-up operation at the origin terminal and delivery operation at the destination terminal.
- Transshipment drop for a container request should happen only after the completion of the previous pick-up operation either at an origin terminal or at any transshipment pick-up terminal for that container request.
- Transshipment pick-up for a container request at a terminal should happen only after the transshipment drop for that container request at that terminal.

Constraint (40) computes the number of visits made by each barge from the solution sequence. Constraints (41) – (49) restrict the invalid arcs that are possible between any two nodes (i.e., the container destination node cannot be visited before any of the pick-up or drop nodes for that container request, container origin node cannot be visited after any of the pick-up or drop nodes for that container request, transshipment drop node for a container request at a terminal cannot be visited after the transshipment pick-up node for that container request at that terminal). Constraints (50) and (51) define the limit of the variables that are used in the mathematical model.

4.6. Heuristic solution approach

The mathematical model described in the previous section might not be able to solve large problem instances to optimality within a limited time available. The computation time for the mathematical model will be huge with MIP solvers. We propose a multi-start GRASP-ALNS heuristics-based algorithm that is used to solve the PDPT. The heuristic is an adaptation of the GRASP-ALNS proposed by (Qu & Bard, 2012). The algorithm is a hybrid metaheuristic that uses a two-phase heuristics approach with a multi-start local search procedure. The multi-start local search procedure helps us to explore a large amount of the search space in a progressive way to search for the optimal solutions in a given amount of time.

The first phase is considered as the construction phase of the solution. The initial solution, which represents the sequence of operation for each barge, is constructed with different construction heuristics as a result of the first phase. A greedy approach is followed in constructing the solution with randomness in selecting the next cargo to be added to the solution. The greedy procedure is explained in the design phase of the algorithm. The randomness enables diversification of the search space with a different initial solution created each time the first phase of the algorithms is run. The first phase of the algorithms is to run multiple times to construct multiple initial solutions as a process of the multi-start local search.

Phase two is an intensification phase where the local search is performed on the initial solution constructed. The local neighbourhood of the initial solution is searched to find a better solution than the initial solution. The neighbours are generated based on the destruction and repair strategies performed sequentially. The algorithm propagates through the search space, searching in the direction to find a better solution than the initial solution considered for the iteration. The second phase of the algorithm is run until the defined stop condition after which the first phase of heuristic generates the next initial start point of multi-start local search. This process of diversification and intensification is repeated until the GRASP-ALNS algorithm reaches its overall stop criteria.

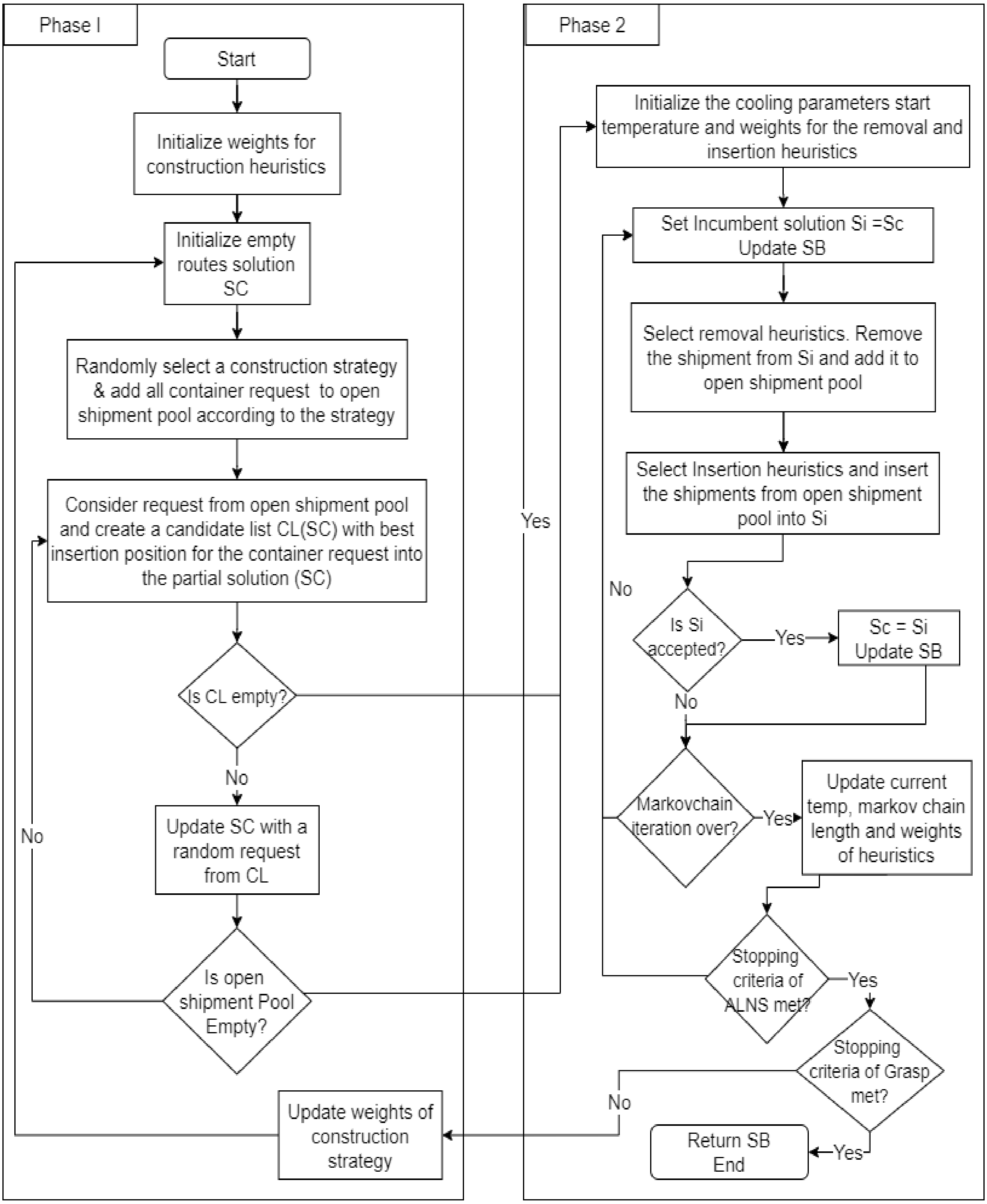

Figure 15: GRASP-ALNS flow diagram

The heuristics used for construction, destruction, and repair of the solution are selected randomly based on the performance of these heuristics over time. The performances of each heuristics are tracked with the help of weight parameters. The weight parameters are updated after a defined number of iterations based on the performance of the heuristics. The detailed explanation about the heuristics and the weight parameters are provided in the following sections. The overview of the working of the GRASP-ALNS algorithm is provided in Figure 16.

The best solution value is updated every time the algorithm finds an improved solution that is better than the previous best solution. When the algorithm terminates, the final best solution is returned as the output of the algorithm.

4.6.1. Solution representation

The solution to the PDPT consists of three parts:

1. Set of routes R, where $r \in R$ is the sequence of the route travelled by a barge from the start terminal node to the end terminal node, visiting different container nodes in between. The nodes visited in between represent the pick-up and drop operation of containers at different terminals. If no node is visited in between, then the barge is not used in the solution.
2. There are service start time and the capacity of load on the barge after completion of service for every node visited in the solution sequence.
3. The last part of the solution corresponds to the open shipment pool, which consists of the list of container requests that are not included in the solution route. These are the list of containers that are not transported by the barges.

The objective function considered is the total cost of transportation which includes three types of costs parameters. The total cost parameter is similar to the objective function defined in the mathematical model, namely the transportation cost associated with transporting the container request by barges, the operation cost associated with the terminal operation and the penalty cost for not transporting a container as well as late delivery of the container request. The total cost of transportation is computed based on the above three parts of the solution.

The first solution parameter, which is the solution route sequence refers to the sequence of nodes visited by each barge which corresponds to the sequence in which different containers are picked up and dropped at a different terminal. The solution is considered to be a sequence of array of node indices that are visited by the barge. There is always a drop node j corresponding to a pick-up node i for any container request $i \rightarrow j$ in the solution sequence (i.e., a container is always dropped at a terminal if it is picked up).

Figure 16 is an example representation of the solution sequence for three barges considered. Each barge starts from their respected start terminal and visits unique nodes and ends in the common end terminal. No node is visited more than once except for the end terminal node. The barge three is not utilized, hence the sequence is from the start terminal node to the end terminal node directly without visiting any other node.

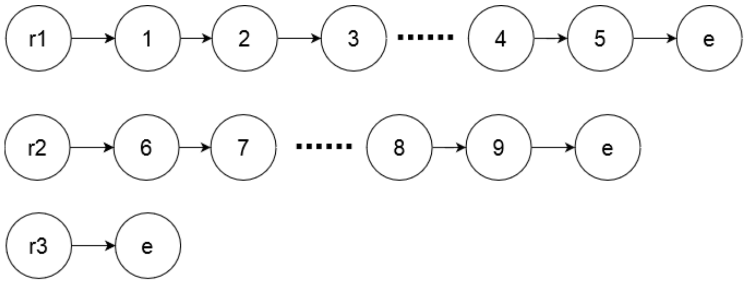

Figure 16: Solution sequence representation - Example

The service start time of a node in the solution corresponds to the time at which the pick-up or drop operation of the corresponding node starts. The continuity in time for various activities is maintained with respect to the sequence of nodes visited.

- The time continuity for solution sequence refers to the time continuity between any two consecutive nodes visited in any route. For instance, when we consider any two successive nodes in the solution sequence, the service start time for the second node is always greater than or equal to the service completion time of the first node.
- The time continuity for a container transportation route refers to the time continuity between the pickup and drop operations for the respected container. For instance, the pick-up operation of the container is performed before the drop operation. The transshipment operations are performed between the initial pick-up and final delivery. The transshipment pick-up operation at a transshipment terminal is always performed after the transshipment drop operation at that terminal.

The time continuity concerning the precedence relation for the solution sequence and the container transportation route is always maintained. The last part of the solution refers to the open shipment pool that consists of the container requests which are not assigned to any barge in the solution sequence.

From the three solution parameters discussed above the inference about the container route (route taken by each container) and the barge sequence (the course of each barge) can be deduced. The container request route contains the time at which the container is handled at different terminals and the respected barge that is transporting the request. If transshipment is performed for a container request, the route contains more than one leg of transportation. The details about each leg of transport are considered in the container route. The barge sequence includes the sequence of terminals visited by the barge during the planning horizon. The arrival time and the number of containers handled at each terminal by the barges are included in the barge sequence.

4.6.2. Operations performed by the algorithm

There are three primary operations performed by the algorithm during the development of the solution. These operations are performed by different heuristics to construct a new solution or identify a neighbour solution. The operations are insertion of a container request into the solution sequence, removal of the container request from the solution sequence and updating of service start time and load values for the nodes after performing insertion or removal. This section contains a detailed explanation of how each action is performed.

Insertion of a request into solution

Requests are always represented with the help of a node pair. The first node in the node pair represents the pick-up of the request and the second node represent the drop action. If we use transshipment in between the pick-up from the origin terminal and drop at the destination, then we consider an extra pair of nodes for each transshipment operation representing the transshipment pick-up and transshipment drop nodes. To assign a container request to a barge, we insert the pick-up and drop node of the container request into the route sequence for the barge. Multiple methods can do the insertion.

Single insertion

The single insertion refers to the insertion of the origin and destination nodes for a container request into a barge route. The barge is responsible for picking up the container at the origin terminal and delivering it to the destination terminal. The delivery should always happen after the pick-up operation. Hence, when we insert *i*→*j* into a route *r*, the node *j* can be inserted at any position after the insertion position of node *i*. Figure 17 represents the route of the barge before and after the insertion of the container request *i*→*j*. The pick-up node *i* is represented in green and drop node *j* is represented in red colour.

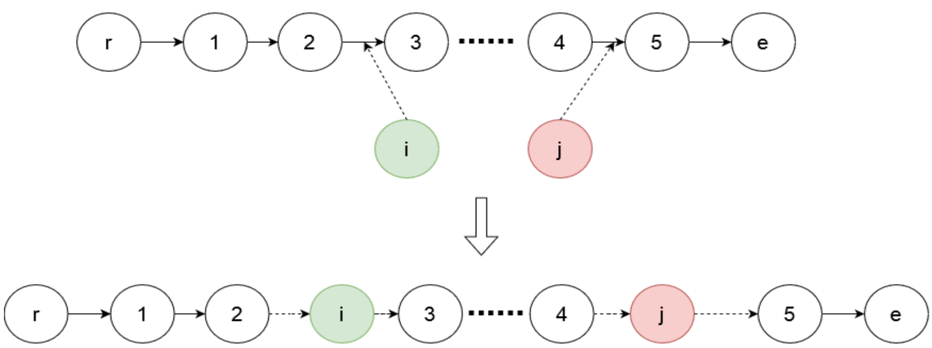

Figure 17: Single insertion – Example

Double insertion

The double insertion is the basic version for the container transportation with single transshipment. The container request *i* →*j* is split and inserted into two different routes of the solution sequences. The container request *i*→*j* is picked from the origin node *i* and dropped at the transshipment drop node *t* by the first barge. The barge which performs the second leg of the transportation picks up the container at the transshipment pick-up node *t'*. The barge transports the container from transshipment pick-up node *t'* to the destination node *j*. Thus, the container route is represented as *i*→ *t*→ *t'*→ *j* where t and t' are the transshipment drop and transshipment pick-up nodes for the container request *i*→*j*.

Hence, the insertion is made in two steps. The first pick-up and drop node pair *i*→ *t* is inserted into the first route. The second pick-up and drop node pair (*t'*→ *j*) is inserted into the second route. Figure 18 represents the route *r1* and *r2* for two different barges before and after the insertion of the container request. It can be noted that the t_{iA}, which is the transshipment drop node for container request *i* at terminal *A* can only be inserted after the position where the origin node i is inserted. Also, the transshipment pick-up node t'_{iA} is inserted before the position of insertion of the destination node j. The service start time of the transshipment pick-up node t'_{iA} is always greater than or equal to the service completion time of the previous node served (node 9 in Figure 18) as well as the transshipment drop node t_{iA}.

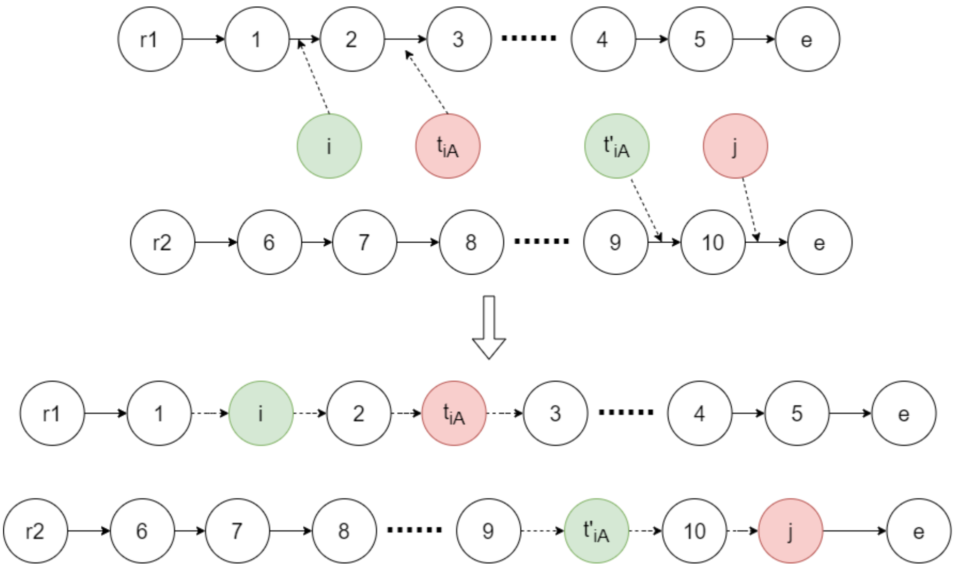

Figure 18: Double insertion – Example

Removal of a request from solution

When a container request $i \rightarrow j$ is removed from the solution, all the nodes that correspond to the pick-up and drop of the respected container request are removed. If container request $i \rightarrow j$ is selected to be removed from the solution, then, there could be two possible cases for removal.

Direct route container request removal

When a single barge directly ships the container request without transshipment, the origin node i and the destination node j is removed from the route and the container request $i \rightarrow j$ is added to the open shipment pool. Figure 19 represents the route sequence before and after the removal of the container request $i \rightarrow j$ from the solution.

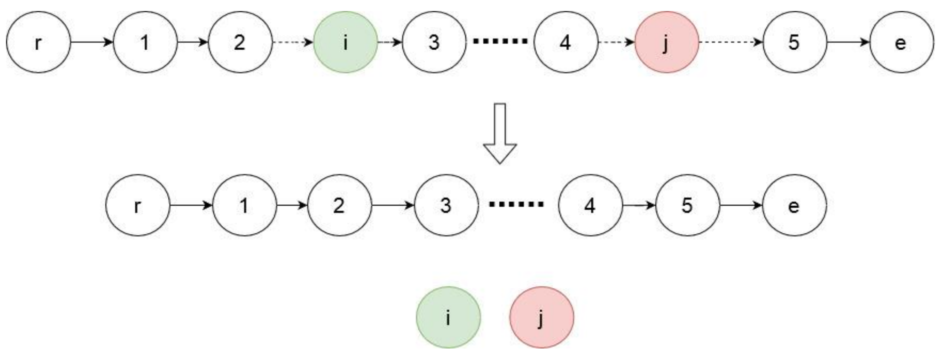

Figure 19: Direct route container request removal – Example

Transshipment route container request removal

When the containers request $i \rightarrow j$ is transported through transshipment, more than two nodes correspond to the pick-up and drop operation of the container request. In the case of transshipment, all the nodes corresponding to the container request $i \rightarrow j$ are removed from the solution sequences. Figure 20 explains the route *r1* and *r2* for two barges before and after the removal of the container request $i \rightarrow j$. It can be observed that $i \rightarrow t_{iA}$ and $t'_{iA} \rightarrow j$ is removed from the solution sequence and the container request $i \rightarrow j$ is added to the open shipment pool. The container request $i \rightarrow j$ that is added to the open shipment pool might later be inserted by single insertion or double insertion, based on the best improvement to the total cost.

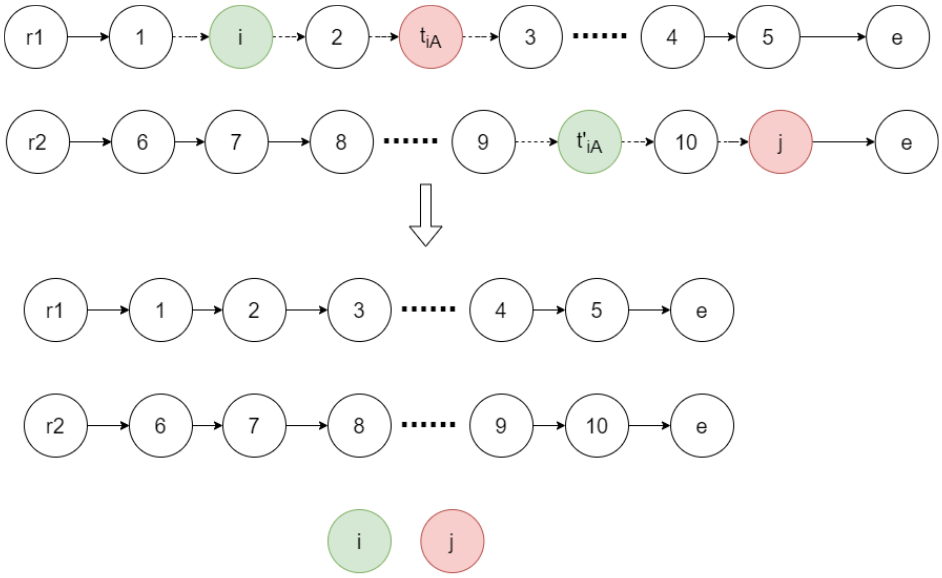

Figure 20: Transshipment route container request removal – Example

Updating the service start time and load values for the nodes

The service start time, and the load after completion of service for the nodes are updated after performing the insertion or removal operations. The 'Update service time and load' procedure gets the solution sequence and the list of affected nodes (NL) as input and updates the service start time and load value for the nodes. The affected nodes list (NL) contains the index of the node inserted into the solution in case of insertion. The NL contains the indices of the successor node to the nodes removed in case of removal. The procedure propagates by iterating through all the node indices in the affected node list. The service start time and load after service for the node are updated based on the previous node values. If there is a violation in the capacity loaded, then the solution is considered as infeasible. In case the affected node is a transshipment pick-up node, then the service start time for the corresponding transshipment drop node is also checked to ensure the time continuity in the container route (i.e., the service start time for transshipment pick-up node should be after the service completion time of the corresponding transshipment drop node).

When an affected node is updated, the service start time and the load value for the successive node will also change. Hence the successive node is now added to the list of the affected node to be updated next. This process is repeated until the last node visited by the barge is iterated. It should be noted that, when we insert a cargo with a single transshipment, there are two pick-up nodes and two drop nodes being introduced into the solution sequence. Hence, two routes in the solutions sequence are affected. Pick-up nodes in both these routes should be considered in the affected nodes list.

There is also a case where we update the service start time of a transshipment drop node for a container request during the propagation. The container request considered will have a transshipment pick-up node in another route which might be affected due to the change in the service start time of this transshipment drop node. Hence, the transshipment pick-up node corresponding to this transshipment drop node of the container request should also be added to the affected node list and its service time and load parameter should be corrected later.

4.6.3. Phase one heuristic

The phase one heuristic represented on the left side of Figure 16 is referred to as the construction phase of the algorithm. Input data corresponding to the barges, container request and terminals are read from the excel file. The heuristic starts with initializing empty routes for barges and adding all the container requests to the open shipment pool. The basic insertion procedure similar to the insertion procedure explained by (Qu & Bard, 2012) is used to check for direct and transshipment opportunities for each container request. The container request that needs to be inserted and the partial solution sequence are provided as the input for the basic insertion procedure. The procedure tries to add the container request into all feasible positions in the partial solution sequence and return a defined number of best possible insertion options. All the infeasible insertions are ignored. The procedure is repeated for every container request in the open shipment pool in order to create the initial solution. The pseudocode for the basic insertion and the first phase is provided in the following sections.

Procedure: Basic insertion

Input: shipment request pair $i \rightarrow j$, current partial solution *SC,* max number of solution to return n_{max}

Output: the set of solution *SL* created with inserting the selected $i \rightarrow j$ to *SC*

Step 1: Check single insertion for $i \rightarrow j$ in solution *SC*

For all route $r \in R$

For all insertion position of *i, j* in route *r*

Set *So* as the solution after inserting $i \rightarrow j$ to *SC*

Put *NL* as empty, add node *i* and *j* to *NL.*
If (*Update service time and load* (*So, NL*)), then
Remove $i \rightarrow j$ from open shipment pool of *So;*
Put $SL \leftarrow SL \cup So$
If $(|SL| < n_{max})$, then
then sort *SL* in ascending order of total cost
Else,
Sort *SL* in ascending order of total cost.
Remove the worst solution from *SL* such that $(|SL| < n_{max})$

Step 2: Check double routes for $i \rightarrow j$ in solution *SC* $(i \rightarrow t \rightarrow t' \rightarrow j)$

For all hub terminal

For all route $r_1 \in R$

For all insertion position of *i, t* in route r_1

For all route $r_2 \in R$

For all insertion position of *t', j* in route r_2

Set *So* as solution after inserting $i \rightarrow j$ to *SC*

Put *NL* as empty, add node *i, t, t'* and *j* to *NL*.
If (*Update service time and load* (*So, NL*)), then
Remove $i \rightarrow j$ from open shipment pool of *So*.
Put $SL \leftarrow SL \cup So$
If $(|SL| < n_{max})$, then
Sort *SL* in ascending order of total cost
Else,
Sort *SL* in ascending order of total cost
Remove the worst solution from *SL* such that $(|SL| < n_{max})$,

Step 3: Return *SL*;
Stop;

Procedure: Phase one heuristic (construction phase)

Input: All customer request (origin, destination, arrival, due date, load) and barges b. max length of the candidate list of solution *(l)* from which a random solution is selected

Output: Solution *SC* (the route for each barge), set of requests not delivered in open shipment pool (*U)*

Step 0: Initialize *SC* = empty route from start terminal node for the barge to end terminal node
Step 1: Select a construction strategy randomly based on the weights of the strategies
Add the request to the open shipment pool in the order based on the construction strategy selected. Open shipment Pool (*U*) contains all unassigned container requests
Step 2: Build candidate list *CL* with potential container request and best insertion position
Put *CL* as empty
Put iterated request (IR) for *CL* as empty
For ($i \rightarrow j \in U$)
Select the next request ($i \rightarrow j$) from *U*
Call Basic Insertion procedure ($i \rightarrow j$, *SC*) and set solution list *SL* = *SC*;
If (*SL* = not empty), then
Add container request $i \rightarrow j$ to IR
//Add solutions in *SL to CL*
Put $CL \leftarrow CL \cup SL$
If ($|CL| < l$), then
Sort *CL* in ascending order of total cost
Else,
Sort CL in ascending order of total cost
Remove the worst solution from *CL* such that ($|CL| < l$),

If (IR = full), then
Go to step 3
Else, continue until all request in *U* are iterated

Step 3: If (*CL* = empty), then
Return current solution *SC;*
Stop;
Step 4: Select a new solution S randomly from *CL* with probability of $1/|CL|$;
Remove request $i \rightarrow j$ associated with *S* from *U;*
Set *SC* = *S;*
If (*U* = empty), then
Return current solution *SC;*
Stop;
Else, Go to Step 2.

The first phase of the heuristics starts with a solution that contains an empty route. The barges start from the starting node and end at the end terminal node without visiting any other node in the empty route. A set of container requests from the open shipment pool are analyzed, and one container request is selected and inserted into the solution sequence for every iteration. This selection of container requests and insertion into the solution sequence is made sequentially to construct the initial solution sequence. The next request to be inserted is selected from a list of a potential request identified called the candidate list. The candidate list is constructed by considering the request one by one from the open shipment pool and adding the request and its best feasible insertion position details. The requests are iterated in the same order as they are present in the open shipment pool. The iteration is stopped when the desired number of potential requests are identified for the next insertion. If there is no feasible insertion possible for a request, the request is skipped, and the next request in the open shipment pool is checked.

The basis insertion procedure explained before is used to identify the best potential insertion positions for each container request. The basic insertion procedure returns more than one potential insertion position for each request. The number of insertion positions returned depends on the parameter settings of the heuristics. The candidate list stores the values of the container request indices, total cost value after insertion of a request, and the position of the insertion. The candidate list is always sorted in the ascending order of the total cost value. The sorting ensures that the solution corresponding to the best possible insertion is always listed on the top. Phase one heuristics is a random greedy procedure, where a random request from the top candidate list is selected to be inserted next. The selected request is inserted into the partial solution sequence, and the candidate list is cleared. This process is repeated until the stop condition for the first phase is reached.

Construction heuristics

The construction heuristics look for the best possible insertion positions for each container request from the open shipment pool into the constructed solution. The number of container requests considered for building the candidate list affects the computation time as well as the initial solution constructed. When all the container requests in the open shipment pool are iterated for generating the candidate list, it might increase the computation time as there might be a huge number of container requests in a real-life case. Hence, there is a restriction on the count of container request considered for the generation of the candidate list. The number of container request considered to generate the candidate list is controlled by the parameter settings of the heuristics. Thus, the order of the container request in the open shipment pool decides the request that is iterated next. There can be various strategies to define the order of the container request in an open shipment pool, as explained by different construction strategies below.

Earliest Due data - We sort the container request based on the due date of each request. The container request with the earliest due date is added to the first of the open shipment pool and the container request with a later due date is added towards the end. This method enables the construction heuristics to consider the insertion of request that is due earlier first.

Earliest arrival time - We sort the container request based on the arrival time of the request to its origin terminal and add to the open shipment pool. This strategy enables us to assign all the containers that are immediately available for transportation into the solution rather than selecting the container which might arrive at a later point of time.

Cargo Id – We sort the container request based on the cargo id of the request. Cargo id is the id provided to a container transportation request when a customer makes a booking. Thus the strategy is similar to the first come first serve strategy, where the container request that arrives first to the planners is given more priority than the request that has arrived later.

Random order - We sort the container request in random order and store it in the open shipment pool and start assigning the request that is available first in the stored order sequence. This strategy ensures a high amount of randomness in the initial solution constructed, enabling diversification of the initial solution.

Container request from terminal pair with a maximum number of containers transported between them assigned first - A matrix with the number of containers transported between terminal pairs is created. The maximum value of the matrix element is identified. This value corresponds to the terminals between which the maximum number of containers are transported. The container requests originating from the terminal corresponding to the row and destined to the terminal corresponding to the column of the matrix are identified and added to the open shipment pool. This operation is repeated for the next maximum value in the matrix until all the matrix elements are iterated. This method ensures that the barge makes use of transshipment opportunities for minimum size request.

4.6.4. Phase two heuristic

The second phase of the algorithm described in the right side of Figure 16 is an adaptive large neighbourhood search procedure which represents the improvement phase of the algorithm. The solution constructed from the construction phase is fed as the initial solution for the second phase. The heuristic makes use of different destruction and repair procedures to identify the next potential neighbour. The destruction and repair procedure refers to destroying the solution and reconstructing it again to complete the solution. The identified neighbour is then compared with the initial solution from which the neighbour solution was created. The neighbour solution is accepted or rejected based on cost improvement. Phase two heuristic enables the intensification of the search process by searching all the local neighbourhood of a solution considered. The phase two heuristic makes use of the simulated annealing technique for accepting the bad solution with a certain acceptance probability and intensifying the search in order to avoid getting stuck in local optima.

There are various strategies used for destruction and repairing of the solution. The strategies are selected randomly based on the selection probability. The probability values for the strategies are derived from the weights for the respected strategies. These weight values are updated based on the performance of the strategies in finding a better new solution during the run time. This update of weight is done with the adaptive weights adjustment techniques which help in improving the performance of the heuristics. The best solution found during the search process is updated for every iteration. The output of phase two heuristic is the final best solution and the list of container request in the open shipment pool referring to the container request that is not transported in any barge schedule. This section deals with explaining the different destruction and repair strategies used in the algorithm and the techniques used to update the weights of the strategies. The pseudocode for the second phase of the algorithm is provided next.

Procedure: Phase 2 heuristic (improvement phase - ALNS procedure)

Input: Initial solution from construction phase *SC*, stop condition for heuristic

Output: Solution *SB* (the route for each vehicle), set of requests not delivered in *U*

Step 0: Initialize cooling parameter, initialize *Si* = *SC*, Initialize the weights for the destruction and repair heuristics and degree of destruction.

If (*Cost (SC)* < *Cost (SB)),* then set *SB=SC*

Step 1: initialize iteration number =1,

Step 2: Set *S* = *Si;* select the degree of destruction μ

Step 3: Solution destruction

Randomly select a destruction heuristic based on selection probability

Remove μ requests from routes of *S;* add them to the open shipment pool *U;*

Step 4: Solution repair

Randomly select a repair heuristic based on selection probability

Insert open requests in *U* to routes of *S*;

Step 5: Check and accept the neighbour solution

```
If (Cost (S) < Cost (Si)), then
        Set Si = S;       // accept neighbor solution if it is better
        If (Cost (S) < Cost (Sb)), then set SB = S;
        Increment weight for destruction and repair heuristics
Else If (Cost (S) > Cost (Si)), then
        Find the acceptance probability value
        If (random probability <= acceptance probability), then
                set Si = S;       // accept worse neighbor solution with probability
                Increment weight for destruction and repair heuristics
        Else, we keep the solution Si without changing it

Increment iteration count

If (iteration < Markov chain length), then
         Go to Step 2.
Else,
        Decrease current temp, update heuristics selection probability
        If (current temp < end temp) then,
                Stop phase two heuristics and return SB
        Else, Go to Step 1.
```

Destruction heuristics

The destruction heuristics refers to the procedure with which a certain degree of the solution is destroyed. The destruction in the PDPT problem is performed by removing a certain number of container requests that are transported from the solution route and adding them to the list of containers that are not transported. There is a degree of destruction associated with any destruction performed. The degree of destruction refers to the extent to which the solution is destroyed. The more the degree, the more the solution is destroyed. The degree of destruction plays an important role in deciding the neighbour solution that is considered for comparison. A low degree of destruction creates a neighbour that is a closer neighbour to the initial solution considered before the destruction. A larger degree of destruction provides a neighbour that is far from the initial solution. Hence, the level of degree of destruction is selected based on the neighbour we want to generate. The degree of destruction in our algorithm refers to the number of container request removed from the solution sequence. Now, we see about the different destruction heuristics that are considered for the algorithm.

Random removal - the first removal/ destruction heuristic is based on randomly removing the container request from the solution sequence. All the container requests that are transported in the solution sequence are considered with equal probability of selection and removed one by one until the required degree of destruction count is achieved. The logic behind selecting this heuristic is to introduce randomness in the neighbour solution generated.

Minimum call size request removal – this heuristic considers the barge visit to a terminal where a minimum number of containers are handled (barge visit with minimum call size) and remove the container requests that are served during the visit. This process is repeated until the count for the degree of destruction is achieved. The logic behind selecting this heuristic is to find a better consolidation opportunity for the container request that are served with lesser call size.

Hub with minimum container transshipment – This heuristic considers the transshipment terminal in which a minimum number of container requests are handled through transshipment. The container requests that are handled in the transshipment terminal are removed from the solution to check for an opportunity to handle them in a different transshipment terminal. There could be a cost improvement option when handling the containers at different transshipment terminals. If there are not enough container requests identified through this heuristic, the remaining count for reaching the degree of destruction is achieved through random removal heuristic.

Worst removal heuristic – This heuristic considers the container request that causes the worst change in objective when removed from the solution sequence. We remove the worst container request and add it to the open shipment pool. The logic behind doing this is to destruct the solution without affecting the route that is performing better in the solution sequence. The total cost, excluding the penalty factor, is considered for this heuristic to identify the worst containers. This enables the identification of the container request that is causing the largest detour in the route and identifying a better alternative route for the removed container request.

Shaw's removal heuristic – This heuristic tries to remove the requests that are similar to each other. By considering the requests that are similar to each other and removing them, there is a high chance that they can be reinserted into a different route which could lead to better performance in terms of the total cost of transportation. The heuristic consists of the following steps:

1. Randomly select a seed which corresponds to a container request that is available in the solution and place them in a list of request to be removed.
2. Find the relatedness measure between the individual requests in the solution and the list of requests to be removed.
3. Select the request with the least value for relatedness function (more related request) and add to the list of requests to be removed. This method enables to select and group the similar request from the solution sequence.
4. Stop, if the degree of destruction is achieved. If not, then repeat step 2 and 3 until the degree of destruction is achieved.

The relatedness measure is the measure of similarity between the container request. If the list of request to be removed is defined as a set LRR then relatedness measure for a container request $i \rightarrow j$ is defined as:

$$R_{i\rightarrow j} = \sum_{k\rightarrow l \in LRR} \left(\beta_1 * \left(D_{ik} + D_{jl}\right) + \beta_2 * \left((arrival_i - arrival_k) + (dueDate_j - dueDate_l)\right)\right)$$

Where the D_{ik} refers to the distance between the terminals corresponding to the node *i* and *k*. We also consider the difference in arrival time and the due date for the container request. β_1 and β_2 refer to the weights for the distance and time factors considered for the relatedness measure function. The relatedness function tries to group the request that is arriving together and having a similar due date, or the request with origin and destination terminals that are close to each other. Hence, the lower the value of the relatedness measure, the more similar the container requests are. We select the container request $i \rightarrow j$ with the minimum value of the relatedness measure function to be removed ensuring the request $i \rightarrow j$ is more similar to the request available in the set LRR that are to be removed.

Repair heuristics

The repair heuristics refer to the procedure with which the destroyed solution is reconstructed or repaired to complete the new neighbour solution. The construction of the new neighbour solution is done with a single objective in mind, which is to improve the total transportation cost of the container request. The heuristics are performed by adding the container request into the solution one by one, until no other feasible insertion of a request from open shipment pool is available, or until all the requests from open shipment pool is inserted into the solution. The different repair/ insertion heuristics considered in the algorithm are as follows.

Random insertion heuristic – This heuristic is similar to the random construction heuristic. A random container request is selected from the open shipment pool and is inserted into the best possible position in the solution sequence. The insertion can be direct or with transshipment depending on which strategy provides more benefit.

Greedy best insertion heuristic – This heuristic iterate through all the container requests in the open shipment pool checking for all the possible insertion position possible. The insertion that causes the best improvement to the total cost is selected and inserted into the solution. Thus we select the container request in a greedy way to have the most improvement while adding the next request into the solution.

Regret-K insertion heuristic – We consider each request in the open shipment pool and return the *k* best possible insertion position for each request. The regret value for each container request $i \rightarrow j$ is computed by finding the difference between the total cost of inserting the request in the best possible insertion position and the worst possible insertion position among the k insertion positions returned. The container request having the maximum regret value is selected to be inserted into the solution next. If a container request has only one possible feasible insertion position, then the regret value is considered to be the total cost after the insertion. This ensures that the container request with only one possible insertion position is given a higher priority.

Most constrained insertion – The most constrained insertion heuristics consider container requests from the open shipment pool with a weight function computed with the sum of the distance between the origin and destination terminals and the difference between the due date and arrival date for the request. This weight function is used to identify the request that will be difficult to accommodate at the later point of time in repairing the solution and insert them first. The below equation gives the weight function. The container request with the maximum weight function is selected first to be inserted at the best possible insertion position into the solution.

$$MC_{i \rightarrow j} = (\beta_1 * D_{ik}) + (\beta_2 / (dueDate_j - \ arrival_i))$$

The insertion heuristic is used only to select the next request that should be inserted into the solution. The insertion is made using the basic insertion procedure to identify the best possible position to insert the request and update it accordingly.

Acceptance strategy for the neighbour solution

During the second phase of the algorithm, a neighbour solution is always accepted if it is better than the initial solution. There is a possibility of accepting a neighbour solution even if the solution is not better than the initial solution from which the neighbour is generated. Such a worse solution is accepted with an acceptance probability that keeps decreasing over iterations ensuring the intensification of the search procedure. Simulated annealing is used as a method to accept the worst solution with an acceptance probability. Simulated annealing procedure starts with initializing the start temperature and gradually decreasing the temperature until it reaches an end temperature below which the heuristic is stopped. During the decrease of the temperature, the acceptance probability of accepting a worse solution also decreases. Thus, ensuring more number of worse solutions are accepted during the initial iteration. Only a few worst solutions are accepted during the final phase of the search when the temperature nears the end temperature. There is Markov chain length referring to the number of iterations that are performed with a specific temperature value before decreasing the temperature. The below equation calculates the acceptance probability:

$$Acceptance\ probability = e^{\frac{current\ solution - neighbor\ solution}{current\ temperature}}$$

A random probability value is generated when a worse solution is identified. If the random probability is less than the acceptance probability, the worse solution is accepted. Else, the worse solution is ignored, and the heuristics continue with the generation of the next neighbour.

4.6.5. Selecting the degree of destruction

The degree of destruction is the degree to which the solution is destroyed. When the degree is large, it leads to a neighbour solution which is far from the initial solution. When the degree is small, it might lead to a neighbour solution which is close to the initial solution considered. Hence, it is recommended to start with a low degree of destruction to evaluate the close neighbour without skipping them. When there is no improvement found from the near neighbour, then, we try to increase the degree of destruction to expand the search space. The basic idea is to search the local search space and find the optimal local solution first using a lower degree of destruction. Once the algorithm is stuck in a local minimum, we increase the degree of destruction to come out of the local minimum trap and search for other solution space. Once an improved solution is found after the increment of the degree of destruction, the degree of destruction value is reset to the lowest value to start searching the nearest neighbours. There are minimum and maximum values for the degree of destruction and an increment factor defined as the input setting for the algorithm from the user. For instance, if the minimum value is 5, and the maximum value is 15 with the increment factor of 5, the values for the degree of destruction considered are {5, 10 and 15}. The initial degree of destruction is set to 5. When no improvement in the new solution is observed for a predefined number of iteration, the degree of destruction value is incremented to 10 and further to 15. When an improvement in solution is found during the iteration, the degree of destruction is again reset to 5.

4.6.6. Adaptive weight adjustment

Adaptive weight adjustment for improvement heuristics

Each improvement heuristics in phase two of the algorithm have a selection probability computed based on the weight of the heuristics, and the heuristics are selected randomly during the run time based on these probability value. The algorithm makes use of an adaptive adjustment of the weight for the destruction and repair heuristics during the run. The probability p_i of selecting a heuristics for a given weight w_i is defined by the following function:

$$p_i = \frac{w_i}{\sum_{i \in I} w_i}$$

Set *DH* is the set of all destruction heuristics when computing the probability for destruction heuristics *i*. Set *RH* is the set of all repair heuristics when calculating the probability for repair heuristics *i*. The initial weights of the improvement heuristics are initialized during the start of the second phase of the algorithm.

For Markov chain number of iterations in the second phase, we keep track of the number of times each destruction and repair heuristics are performed. When a neighbour solution is better than the initial solution we increment the weight for the respected destruction and repair heuristics by a factor of σ_1 that leads to an improved solution. This weight improvement is also performed when a worse solution is accepted, but with a different increment factor of σ_2 (such that $\sigma_1 > \sigma_2$). When we decrease the temperature after Markov chain number of iterations, we update the probability of selecting the destruction and repair heuristics based on the increased weight derived from their performance.

$$w_i = w_i * (1 - \rho_i) + \frac{\rho_i * \pi_i}{n_i}$$

Where, w_i is the adaptive weight for calculating the selection probability of the heuristic, π_i is the increased weight for the heuristic during the Markov chain number of iterations and n_i is the number of times the heuristic is used. ρ_i corresponds to the roulette wheel parameter indicating the percentage of weight changed due to the performance of a heuristics. If ρ_i is zero, the weight is updated to the initial value, and the effect of performance of the heuristics is not considered. If $\rho_i = 1$, then, all the weights are updated completely based on the performance during the previous Markov chain number of iterations. The value of ρ_i is selected between 0 and 1.

Adaptive weight adjustment for construction heuristics

Similar to the improvement heuristics, the construction heuristics also have weights that are initialized during the start of the algorithm and respected probability values calculated with the same formula as explained for improvement heuristics. The weights of the construction heuristics are updated for every iteration when the algorithm generates a solution that is better than the best solution of the previous iteration. If there is no improvement in the best solution for an iteration, then the heuristics weights are unaffected.

4.6.7. Stopping criteria

The stopping criteria refer to the conditions that instruct the algorithm to stop and move to the next phase of the algorithm or end the algorithm. Three stop conditions are defined for different phases of the algorithm as follows:

1. The Grasp-ALNS algorithm performs the two phases of the algorithm sequentially and stops after completing a specific number of iteration or if the maximum run time limit is reached. The algorithm stops if any one of the stop conditions occurs, which every stop condition is happening first.
2. The first phase of the algorithm is performed as many times as the Grasp-ALNS algorithm to construct the initial solution. This phase of the algorithm stops when all the requests from the open shipment pool are inserted into the solution or if there is no feasible request in the open shipment pool to be inserted into the solution.
3. The second phase of the algorithm makes use of a simulated annealing procedure. The procedure is performed until the temperature reaches the end temperature or if there is no improvement found in solution after a specified number of iterations.

4.7. Conclusion

The variant of the PDPT problem identified has been modelled as a MIP and a two-phase algorithm. The two-phase algorithm performs an adaptive large neighbourhood search procedure to improve the solution value. Various business constraints and restrictions were considered during the design of the solution approach. The research questions regarding the identification of the KPI for capturing the performance were answered. A cost-based objective function was defined to compute the total cost of transportation. The research question concerning the development of a solution approach for routing and scheduling of barges has been answered with the help of the designed algorithm.

Chapter 5: Validation of Solution Approach

This chapter presents the evaluation of the performance of the algorithm and the mathematical model developed for the barge routing and scheduling problem. The content of the research deliverables is given in Section 5.1, followed by the evaluation of the solution approach in Section 5.2 based on virtually generated test instances. The experimentation on the real-life data with the designed algorithm is performed in Section 5.3. and the results are discussed.

5.1. Content of the research deliverables

Apart from the routing and scheduling algorithm which uses metaheuristic approach to solve the barge routing and scheduling problem, there is a template designed to feed the data to the algorithm and a MIP model designed with AIMMS[1] to compute the exact solution to the PDPT problem for performance comparison. The content of these deliverables is explained in detail below.

5.1.1. Input data sheet template

The input for both, the mathematical model and the algorithm, is provided with the help of the excel file. Hence, to design a standard approach for the input data and easy readability of the input data, an excel template is created. There are four sets of datasheets provided as input to the mathematical model and the algorithm, namely 1. General Data, 2. Cargo Data, 3. Barge Data, and 4. Terminal Data. The excel template contains a default sheet named 'General Data' which contains buttons for generating the other datasheet of the template. The excel template uses macros and generates three different input data template sheets where users can enter the cargo, barge, and terminal related input data, respectively. The working of the template sheet and the content of the individual data sheet is explained in detail in Appendix B.

5.1.2. Mathematical model

The mathematical model is a MIP model and is designed using the AIMMS software which supports a wide range of mathematical optimization problem solvers. The mathematical model explained in Section 4.4 is implemented in AIMMS, and the input for the AIMMS model is created with the help of the excel template. The AIMMS model makes use of CPLEX[2] solver and is run in a windows machine with 8 GB of memory and an Intel i7 CPU with 1.80 GHz core.

5.1.3. Algorithm (Two-phase heuristic solution approach)

The algorithm that uses a multi-start GRASP-ALNS explained in Section 4.5 is implemented using Java and compiled with Eclipse mars and run in a windows machine with 8 GB of memory and an Intel i7 CPU with 1.80 GHz core. The java package, along with the hierarchy

[1] Advanced Interactive Multidimensional Modeling System (AIMMS) software which is an optimization-based application development software

[2] IBM ILOG CPLEX Optimization Studio is an optimization software package

of the classes and the methods implemented in each class are explained in Appendix C. The parameter settings applicable for the algorithm are also explained in Appendix C.

5.2. Evaluation of the solution approach

The evaluation of the solution approach is performed in two steps. The first step is the evaluation of the correctness of the solution approach. The correctness of the solution approach is evaluated by testing the functionality and working of the mathematical model and heuristics. The second step is the evaluation of the effectiveness of the solution approach. The effectiveness of the solution approach with respect to the performance is measured with the help of virtually generated test instances.

5.2.1. Evaluating the functionality and working of the solution approach

To evaluate the functionality of the constraints and the methods used in designing the solution approach, we generate small scenario-based test instances and subject the mathematical model and the heuristics to test. There are two types of test performed to evaluate the different functionality and working of the mathematical model and the heuristics.

- Black-box testing: This type of testing refers to the testing of the specific functionality of the model without peering into the internal structure or working. The test scenarios with the scenario-specific test data as input are created to check the compliance with each functionality. Functionalities such as barge capacity restriction, time continuity for loading and unloading of containers, continuity of transshipment of containers, arrival time and due date constraints, etc. are tested under black-box testing. Both the mathematical model and the heuristics are tested as a part of black-box testing.
- White-box testing: White-box testing refers to the testing of internal structure or working of an application. This type of testing is done only for the heuristics. The working of each construction heuristics, destruction heuristics, repair heuristics, cost calculation, node service time and load update, insertion of container nodes into the existing solution, initialization of heuristics weight and update of heuristics weights are tested as a part of white-box testing.

The black-box testing is performed to validate if the business requirements and constraints are properly handled by the mathematical model and the heuristics, whereas white-box testing is used to verify the working of the heuristics. The test scenarios used for the black-box and white-box testing are explained in detail in Appendix D.

5.2.2. Evaluating the performance of the solution approach

In order to evaluate the performance of the heuristics and compare the performance with the results of the mathematical model, small instances of test data are generated for the pick-up and delivery problem with transshipment. There are various sizes of test instances generated which are differentiated by the four sets of parameters. The parameters *cn, bn* and *tn* denote the number of container request considered, the number of barges considered and the number of terminals considered for the test instances. The parameter *hn* denotes the number of hub terminals considered among the terminals in the service network. The size of the test instances and the run time used in the problems from similar literature are analysed before designing the test instances.

Table 3: Test instances size considered in related works

Referred Literature	Transshipment	Problem size			Run time	
		cn	*bn*	*tn*	CPLEX	Metaheuristics
(Fazi, 2014)	No	28 - 109	2 - 7	4	4 hr	20,000,000 iteration
(Sharypova, 2014)	Yes	5 - 50	5 - 10	5-7	10 hr	10 hr

Author (Fazi, 2014) did not consider the transshipment opportunities in the HAP model. The problem instances were based on real-life demand data. It can be inferred that solving a PDPT for small size instances can be very time consuming using a MIP model. Authors (Sharypova, 2014) could not find a feasible solution for their problem instances using a CPLEX even with 10 hours of run time for large size instances. Authors (Qu & Bard, 2012) consider the PDPT problem with different terminals for each pick-up and drop operation. Findings from (Qu & Bard, 2012) suggest that it can take up to 2 hours for solving small instances of the problem with eight customer request, two carriers and one transshipment terminal using CPLEX.

The heuristic model proposed by (Sharypova, 2014), which considers synchronization of transshipment is run 10 hours for instances with 30-50 cargo size. Author (Fazi, 2014) model is also run for a large number of iterations. Hence, the consideration of the container request between 5 – 30 container request with 2 - 5 barges and 4 - 6 terminals and one transshipment terminal is considered for the research in order to achieve a good quality of results for MIP model and compare the perfromance of the heuristic. The mathematical model proposed in the research are run for a maximum of 3600 seconds for small instances and 7200 seconds for large instances the heuristic proposed in the research are run until the stop condition set in the parameter settings.

5.2.3. Data generation

The test instances for evaluating the performance are generated based on the virtual service network with a random container demand. There are three types of terminals that are considered in the virtual service network. The first type of terminals is the seaport terminal. The seaport terminals are usually clustered and are concentrated to a port region. For example, the port of Rotterdam contains more than 30 port terminals where barges can load the cargo. The second type of terminals is inland terminal which are considered to be the terminals from which the export cargo are loaded, and import cargo are delivered. These terminals are scattered throughout the hinterland, and the number of terminals concentrated in a region is very less. The third type of terminals is the intermediate terminal or transshipment hub terminal. The transshipment terminal can be either a seaport terminal or an inland terminal. The decision to decide a terminal to be treated as a transshipment terminal is based on how effective it can facilitate transshipment opportunities. From the literature research conducted, it can be inferred that the transshipment terminals are considered to be close to the seaport and in between the seaport and the inland terminals. There is no use in having a transshipment terminal towards the origination of a river(opposite end to the mouth of the river). Hence, it is decided to consider the intermediate terminal or the seaport terminal as transshipment terminals. The size of the problem instances are dependent on the instance parameter such as number of container request considered (*cn*), number of barges considered *(bn)*, number of terminals in the service network *(tn)* and number of hub/ transshipment terminals among the terminals *(hn)*. There are 10 test instances that have been generated. The size of the problem is increased gradually by increasing these parameter values.

The instances are generated with a service network similar to the example network designed in Figure 21, and the transshipment terminal is considered in the intermedial location. The distance between terminals is considered proportionate to the distance shown in the figure.

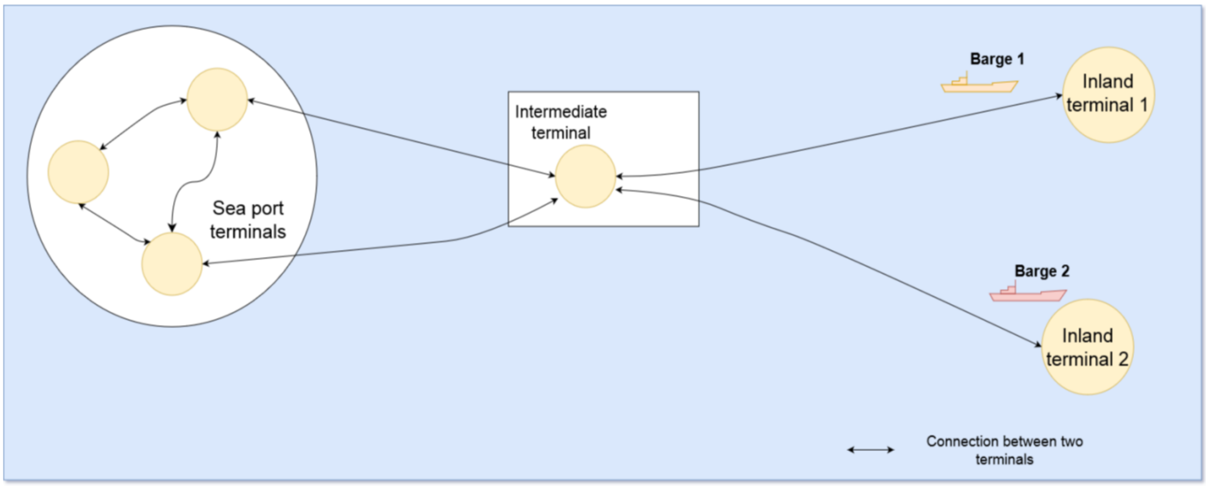

Figure 21: Example for service network with port and inland terminals used for creating the test instances

The origin and destination for the container request are selected based on random probability. There are a number of import container request generated from a random seaport terminal to random inland terminals. There is also export container request generated from a random inland terminal to a random seaport terminal. The ratio of the import to the export container request is maintained equal, i.e., the equal probability is considered for import and export container demand. There are also scenarios with a container request picked up at a seaport terminal and delivered to another seaport terminal or picked up at an inland terminal and delivered to another inland terminal. These scenarios are considered with a minimum probability of occurrence. The container request route is the route from the origin terminal to the destination terminal. The transshipment opportunity is created in the service network design when two container request route overlap. The decision to consider the import and export container request in the service network was made to inject the opportunity of transshipment by considering the overlapping of container routes. The due date for each container request is greater than the time taken to travel the distance between the origin and destination terminals. The barges are assumed to be present in any one of the random terminals during the start of the planning horizon.

The parameter settings for the heuristics are set based on a trial and error method. The algorithm is run for one iteration with different parameter settings by varying the cooling parameters, and the time taken for the solution space to converge is analysed. The parameter setting which achieves maximum convergence with a minimum run time is selected for each test instance. The algorithm is then run with multiple start points for the selected settings. The parameter settings used for the different test instances are explained in Appendix E.

5.2.4. KPI for comparing the performance

The MIP model constructed with AIMMS using a CPLEX solver and the heuristics model compiled in eclipse is used to solve the test instances generated. The decision is made to compare three different solution approaches for test instances such as:

1. MIP model constructed with AIMMS and run with CPLEX solver.
2. Grasp algorithm (Construction phase algorithm) compiled in Eclipse.
3. GRASP-ALNS algorithm (Construction phase + Improvement phase) compiled in Eclipse.

The gap value, which represents the percentage of gap between the two solutions, are used to compare the performance of the different solution methods. The gap value is computed with the below formula:

$$Gap = \frac{solution\ 1 - solution\ 2}{solution\ 2} * 100$$

Solution 1 and solution 2 in the above formula are the solutions to be compared. A positive value in the gap represents that solution 2 is better than solution 1 by the percentage of gap computed. Similarly, negative gap value indicates the solution 1 is better than solution 2. There are four different gaps calculated for the comparison, as explained below:

- Gap 1 - The gap between best lower bound for the MIP model vs best objective value found with the MIP model for the run time.
- Gap 2 – The gap between the best objective value found with the MIP model for the run time vs best objective from GRASP heuristic.
- Gap 3 – The gap between the best objective value found with the MIP model for the run time vs best objective from GRASP -ALNS heuristic.
- Gap 4 – The gap between the best objective value found with GRASP heuristic vs best objective from GRASP -ALNS heuristic.

The lower bound for a minimization problem refers to the minimum value, which is less than any feasible objective value for the problem. It is observed that the lower bound of the MIP model converges very slowly for most of the instances. During the execution of the MIP, the best CPLEX solution remained unchanged and the lower bound kept on increasing over time for most of the small and medium-sized instances, meaning, the best solution found by CPLEX was more close to the optimal solution than the predicted gap value. The convergence of the LB and the UB was very slow because the size of the PDPT is vast due to the consideration of transshipment. Hence, the comparison of the heuristic solution and the best objective value found using the MIP model is made for the computation of the gap values. The comparison of the best objective of MIP model with the heuristic enables us to compare the two solution approaches with respect to computation time and solution quality. The test instances are run once using AIMMS to solve the mathematical model. The test instances are run ten times with GRASP and five times using the multi-start GRASP-ALNS heuristic. The best objective value for test instances, gap values, the average objective cost for multiple runs, and the time taken to compute them are provided in Table 4.

Table 4: Results of test instances for performance comparison

ID	*Instance*				*MIP*		*GRASP*				*Multi-start GRASP-ALNS*				
	cn	*bn*	*tn*	*hn*	*Best solution*	*Gap 1 (%)*	*Best solution*	*Gap 2 (%)*	*Average obj*	*Average computation time (s)*	*Best solution*	*Gap 3 (%)*	*Gap 4 (%)*	*Average obj*	*Average computation time (s)*
I1	5	3	4	0	642.5	0.00%	642.5	0.00%	720.8	2	642.5	0.00%	0.00%	642.5	3
I2	5	3	4	1	642.5	31.24%	642.5	0.00%	711.4	3	642.5	0.00%	0.00%	642.5	17
I3	10	3	5	0	1069.5	62.15%	840.0	-21.46%	919.8	4	835.0	-21.93%	-0.60%	835.0	9
I4	10	3	5	2	835.0	56.78%	840.0	0.60%	905.3	52	835.0	0.00%	-0.60%	860.5	1000
I5	15	3	5	0	1270.0	82.28%	1027.5	-19.09%	1054.3	5	1027.5	-19.09%	0.00%	1027.5	19
I6	15	3	5	1	1160.0	87.07%	1030.0	-11.21%	1089.3	163	1027.5	-11.42%	-0.24%	1068.4	1163
I7	20	5	6	0	2113.0	84.81%	1027.0	-51.40%	1075.3	16	1027.0	-51.40%	0.00%	1046.1	47
I8	20	5	6	1	2738.0	88.28%	1063.0	-61.18%	1272.0	747	1041.0	-61.98%	-2.07%	1073.5	1508
I9	30	4	6	0	7474.8	84.81%	1234.0	-83.49%	1385.4	58	1227.0	-72.32%	-0.57%	1245.8	205
I10	30	4	6	1	7647.9	95.79%	1328.0	-82.64%	1956.2	1000	1262.0	-83.50%	-4.97%	1593.0	3600
	Avg				2559.3		967.5		1109.0		956.7			1003.5	

5.2.5. Discussion

There are different size of test instances considered for evaluation of the performance of the solution approach. The columns corresponding to the gap value for each solution approach is used to compare the performance of the respected solution approach. Analysing the gap value of the MIP model for different problem instances indicates that the problem size increases with the increase in the number of containers, the number of barges and number of terminals. Gap 1, which is the gap between the lower bound and the best solution found using a CPLEX solver for the instance I1, is zero. The zero-gap indicated that the optimal solution to the problem instance is identified. It takes 11 seconds to reach the optimal solution for the CPLEX solver to solve I1. Gap 2 representing the gap between the best solution of the MIP and the solutions of construction phases of the algorithm. The results of the GRASP are close to the MIP solution for smaller instances. The results of the GRASP are better than the best solution of the MIP for larger instances. It can be noted that the GRASP-ALNS give equal or better results compared to both MIP and GRASP solutions in almost all the instances considered. The gap between the best solutions from GRASP and the GRASP-ALNS is considerably small for instances without transshipment terminal compared to scenarios with transshipment terminal. The improvement phase of the GRASP-ALNS algorithm always tries to find a better solution than the solution found by the construction phase.

It should be noted that the gap value is computed with the best solution among the multiple runs. However, the difference between the best objective value and the average objective value is considerably high for GRASP, indicating that the GRASP generates a random greedy solution to initialize the search. Hence, the solution is not consistent when GRASP is used for generating the best solution. The comparison between the best objective and the average objective value of the GRASP-ALNS suggests that the solution of the two-phase heuristics is consistent for most of the problem instances. The gap value between the MIP solution and the GRASP-ALNS solution is less than -50 %, indicating that the GRASP-ALNS is more effective for larger problem instances compared to the solution from MIP even for less run time. The inference from the comparison of the computation time for the instances with and without transshipment suggest that the problem with transshipment are very time-consuming. The high run time is because of checking all the possible insertion position of the container request in different barge routes through the transshipment terminal. The algorithm also should update different barge routes due to the synchronization of transshipment operations.

5.3. Performance of heuristics solution approach on the instances with real-life demand data

The heuristics solution approach discussed in Section 4.6 is considered, and the real-life data are used to check the feasibility of the GRASP-ALNS algorithm to generate solutions for real-life demand instances. This section explains the different experimental setup considered for the analysis and discuss the results from the experiments.

There is a business requirement related to the container call size handled by the barge during each terminal visit. The call size refers to the number of containers handled by a barge during its visit to a terminal. A barge can visit a terminal one or more time in the solution sequence generated by the MIP model. Hence, it is difficult to distinguish the containers handled during these multiple visits of a barge to a terminal. Due to this reason, the computation of the number of containers handled by a barge during each terminal visit becomes difficult. This constraint is ignored in the mathematical model discussed in Section 4.4. The constraints are ignored in the heuristic solution approach explained in Section 4.6 in order to compare the results of heuristics and the MIP model. We extend the heuristic solution approach to incorporate the call size requirement by adding a penalty cost to the total cost objective in case of violating the minimum call size requirement.

5.3.1. Experiment design

As discussed in Chapter 1, the objective of the research is to develop an algorithm for solving the barge routing and scheduling problem. The main goal has been satisfied with the help of the multi-start GRASP-ALNS algorithm designed for solving the PDPT problem.

The additional scope defined in the research proposal was to explore the advantages of considering the transshipment opportunities during the routing and scheduling of barges. Hence, the experimentation using the real-life demand data is performed with and without the transshipment terminals included in the actual transportation service network of the barge operator. The extended version of the heuristics is used to run the real-life data instances, and the performance of different test instances are observed. There are two sets of experimental setups considered for the experimentation phase, as follows:

- Transportation of containers in the service network without any transshipment terminals.
- Transportation of containers in the service network with transshipment terminals.

5.3.2. Data generation

In order to achieve a comparison between the two experimental setups, test instances based on the real-life demand are generated. The results of the heuristics solution for the test instances are compared for the two setups defined. From the findings of Section 5.2., it can be inferred that the model with 30 container request can take hours to get an optimal solution with transshipment opportunities. There can be more than hundreds of containers handled on a weekly basis during the real-life operations. To bridge this gap, we perform data analysis on the existing container demand that is handled in real-life and generate test instances that could possibly represent the real-life container demand and solved in a reasonable amount of time.

Cofano provides the source data from the Barge Operating System.The data consist of the barge schedule of the barge service for different service network. The data contains contains the container type, its loading location, destination location, the respected barge that is transporting the container, expected time of pick-up and expected time of delivery for the containers. The source data is cleaned and visualized in order to get a final data set corresponding to one of the service networks of the barge operator. The service network selection is made by identifying the terminals that are handling the maximum volume of containers during a stable demand period based on data visualization. From this data set, there are several test instances generated for the experiment purpose. The container request from a client arrives in a huge volume, i.e., 50-100 containers belonging to the same customer arrives at a terminal at the same time and needs to be transported to the destination before the same due date. These container loads are not split and are usually carried together by the same barge.

The decision to consider these container request belonging to the same client and arriving at the same time as a single container request is taken in order to reduce the size of the problem without affecting the real-life demand. The load of the aggregated container request is equal to the sum of loads of all the individual containers considered in the request. The handling time for this container request is considered to be the time taken to handle the aggregated set of containers at a terminal.

Each instance consists of four sets of data. The first set of data is the general data which includes the general cost-related data and other general information such as the number of container request, number of barges and terminals considered for the problem. The second set of data includes the terminal data, such as distance matrix and the transshipment terminals considered in the network. The third set of data is related to the barge, its speed, capacity, and cost of transportation. The fourth set of data represent the list of container request, its origin and destination location along with the load, arrival and due date values. The data related to individual cost components are assumed similar to the values used in (Sharypova, 2014) and (Fazi, 2014). The consideration of other datas and the procedure for generating the test instances from the real-life source data set and selection of the transshipment terminal is explained in detail in Appendix E.

5.3.3. Experiment results

With the extended version of the GRASP-ALNS algorithm which includes penalty cost for violating the minimum call size, and considers the group of containers to be transported together as a single container request, we experiment on the test instances created. The two experimental setups, with and without consideration of transshipment terminals, are run using the GRASP-ALNS algorithm for four different test instances generated. It is to be noted that the instance one (S1) and instance two (S2) are small instances. The third (S3) and fourth (S4) instance are huge instances. The instances are generated based on the actual proportion of demand from different terminals considered in the service network, i.e., proportionate to the real-life demand.

The test instances are run five times to check the reliability of the results. The parameters referring to the size of the test instances and the total unit load of containers considered in the test instances are summarized in Table 5. The decision to consider only one transshipment terminal for the service network is made after analysing the considerable run time of the algorithm with multiple transshipment terminals. The average objective cost value and the average computation time for the test instances are also summarized in the table. The different parameter settings used for the problem instances are explained in Appendix E. The parameter turning is performed similarly to the method explained in Section 5.2.3. The multi-start algorithm is run for one iteration to check the convergence of the solution space for a different set of parameters. The set of the parameter that achieves maximum convergence in minimum time is selected as the parameter settings.

Table 5: Results of test instances for experiments

Instance	*Instance*			*Without transshipment terminal*			*With transshipment terminal*		
	cn	*bn*	*tn*	*Best obj*	*Average obj*	*Average computation time (s)*	*Best obj*	*Average obj*	*Average computation time (s)*
S1	[redacted] Censored								
S2									
S3									
S4									

5.3.4. Discussion

As mentioned earlier, the test instances S1 and S2 are small-sized instances designed to test the performance of the algorithm with respect to solution quality. It is easier to solve smaller instances and find the best solution within a reasonable amount of time due to the small search space. The algorithm is run for one hour run time or completion of five iterations of the multi-start GRASP-ALNS whichever is earliest. Results from test instances S1 and S2 show that both the best objective and the average objective values are better for scenarios with transshipment terminals considered. This is the result of using transshipment opportunities in the container route. However, when comparing the time taken to solve the models with transshipment and without transshipment, there is a huge difference between the average time spent to solve the instances for the two setups. The time difference is observed even for small instance with six container request. The difference increases in a larger scale for instances with more number of container requests. The instance S4 with transshipment performs only one or two iterations of the multi-start GRASP-ALNS algorithm for the stop criteria mentioned. The difference between the average objective value and the best objective value for the first three instances without transshipment are small, indicating that the solution is reliable for multiple runs of the test instances. The test instance S4 consider more number of containers that are transported in a service network with more number of terminals. The instance S4 is having a larger difference between the average and the best solution for the scenario with transshipment. This is because the search space for the problem with transshipment increases with an increase in the number of container request and the number of barges. In order to achieve a consistent solution, the search parameters should be increased to run the algorithm for more number of iterations. This increase in the number of iteration leads to finding a better optimal solution compared to the solutions found. But, the run time for a large number of iteration will be very high. The parameter settings of the algorithm for the instance S4 are decided in such a way that the algorithm performs at least one complete run of the GRASP-ALNS within the maximum time limit. Even within these less number of iterations, the algorithm is able to achieve the solution for the scenario with transshipment which is close to the values of the solution without transshipment. Hence, the inference can be made that the consideration of the transshipment opportunities may lead to cost improvement if transshipment opportunities are available in the considered planning horizon and service network. But, the identification of this improvement comes with the expense of high solution computation time.

5.4. Conclusion

To summarize the chapter, the contents of different deliverables of the research has been explained along with the experimental design and the results for the test instances. The discussion about the test instances generation procedure and results of the test instances has been defined in this chapter. The evaluation of performance has been made by comparing the results of the exact and heuristic solution approach. The answer to research sub-questions related to the experimental setups considered, and the performance of the solution approach performs for different experimental setups has been answered in this chapter.

Chapter 6: Analysis and Recommendations

The recommendation for Cofano based on the analysis of the results from the experiments is provided in this chapter. Section 6.1 describes the analysis performed by comparing the performance of the single-start and multi-start approach. The performance of the algorithm to converge through the solution search space is provided in Section 6.2. Section 6.3 gives the advantages and disadvantages of considering transshipment opportunities along with the cost tradeoff for selecting the transshipment options. Section 6.4 explains the recommendation to Cofano for implementing the solution approach along with other potential benefits that could be harvested using the solution approach.

6.1. Analysing the performance of the multi-start algorithm

The GRASP-ALNS algorithm considered is a multi-start algorithm. The two-phase algorithm is a restart after the completion of an iteration and runs until a stop condition is met. The metaheuristics search through the available search space and propagate towards the direction of the best solution. The multi-start condition is implemented in the algorithm in order to initiate the search at different starting points and search for the local optimal solutions near the start point. An analysis to compare the solutions obtained from a single start point and multiple start point is conducted. The results from GRASP-ALNS algorithm for a single start point and ten start point on the artificially generated test instances is provided in Table 6.

Table 6: Single-start vs multi-start performance

Set	Single start			Multi-start		
	Best Obj	Average Obj	Average time (Sec)	Best Obj	Average Obj	Average time (Sec)
I1	642.5	695.5	2	642.5	642.5	5
I2	642.5	764.5	3	642.5	642.5	17
I3	835	898	3	835	835	9
I4	835	908.5	74	835	860.5	1121
I5	1027.5	1127	6	1027.5	1027.5	19
I6	1100	1177	229	1027.5	1068.4	1442
I7	1041	1084.4	14	1027	1046.1	47
I8	1041	1206.9	145	1041	1073.5	1636
I9	1227	1623.7	41	1227	1245.8	205
Average	**932.4**	**1053.9**	**57**	**922.7**	**937.9**	**500**

The results of the multi-start scenarios are from Table 4. The instance I10 is not considered in this analysis as the time for completing multiple iterations is very high. The single start scenarios for the test instances are run with a slow cooling parameter such that the second phase of the algorithm performs an extensive search on the local neighbourhood of the single initial solution generated. The test instances are run for five times. The best objective value, average objective value and average computation time are summarized in Table 6. The best objective value among the five iterations is compared with the average objective values in Figure 22 and Figure 23.

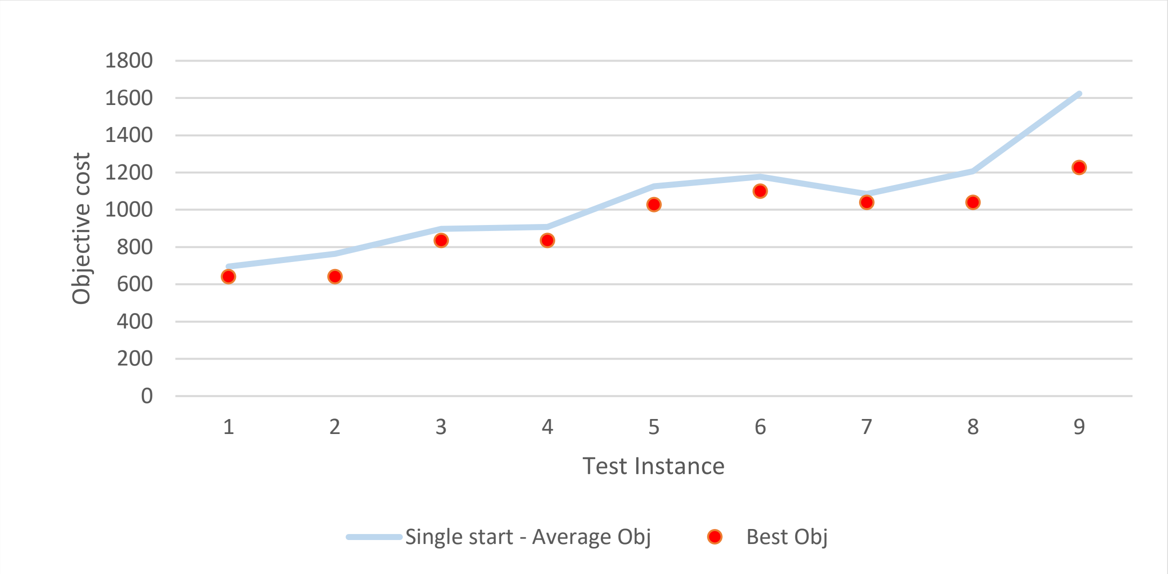

Figure 22:Performance- single-start heuristics

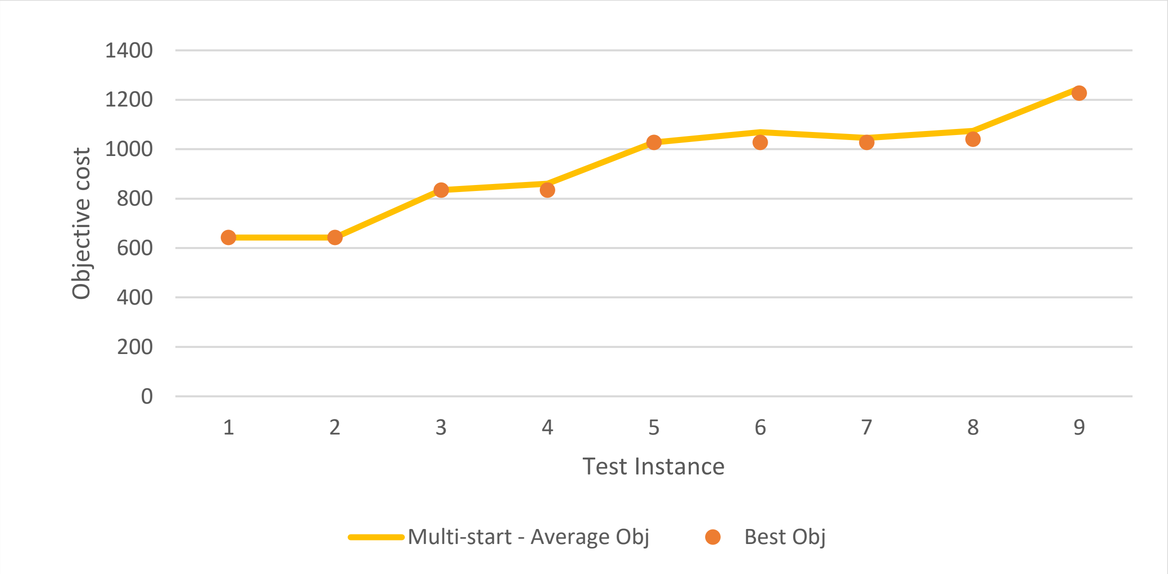

Figure 23: Performance - multi-start heuristics

From Figure 22 and Figure 23, it can be inferred that the gap between the best objective and the average objective values for the test instances are small for multi-start scenarios compared to the single start scenario. The multi-start scenario leads to more reliable results compared to the single start scenario. The comparison between the average objective of the two scenarios in Figure 24 also indicates that the average performance of the multi-start is better compared to the single start scenario. The solution of multi-start is always associated with the high computation time for searching the search space. The multi-start algorithm performs a fast convergence of the solution space with fast cooling parameters, whereas the single start, the heuristics should explore through the search space in order to move to the next local neighbour space. Hence, it is recommended to prefer fast convergence with multi-start rather than the slow convergence with a single start.

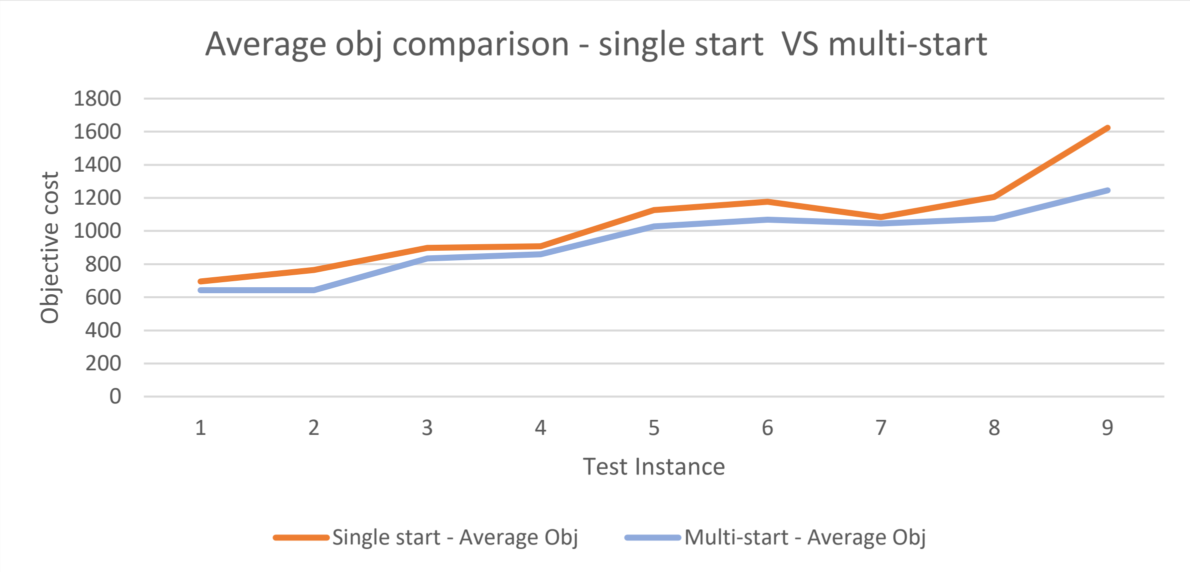

Figure 24: Average objective value comparison

6.2. Analyzing the convergence of the solution search space

The variation of cooling parameters and its effect on the solution quality and computation time is explained in this section. The second phase of the algorithm representing the improvement phase is responsible for the convergence of the solution search space. The speed of the convergence can be controlled by the cooling temperature parameters used in the algorithm. When the algorithm is run for more number of iterations, the solution search space explored is more, leading to a better chance for identifying the best solution. There is an option for the user to tune the run time of the algorithm based on the quality of the solution required. This is achieved by varying the cooling parameter setting based on the time available to generate the solution. i.e., if the time available to create the solution is less, the settings can be manipulated to stop the algorithm within a few iterations. But the solution obtained might not be better than the solution with more run time. Making the algorithm run for a huge number of iterations for a problem with a small search space leads to visiting the same solution again and again. Hence, it is important to decide on the parameter settings based on the solution quality and time available to generate the solution.

Now, we discuss the performance of the heuristics with respect to the intensification and the diversification of the search through the solution search space. This analysis enables us to understand the convergence of the solution space by the algorithm and decide on better parameter settings. The multi-start GRASP-ALNS algorithm developed as a solution approach to the pickup and delivery problem is a sophisticated search algorithm which constructs the initial solution with the help of different construction heuristics. This process is adapted in order to diversify the starting points of the solution search space. The ALNS improvement heuristics perform the intensification phase. The improvement heuristics make use of the simulated annealing procedure, to diversify and intensify within the local search space. This technique enables the algorithm to converge through the solution space in a faster manner.

Some drawback of using metaheuristics is that the solution iterated can be revisited multiple times. One cannot predict if the best solution found by the metaheuristics is the overall best solution until the exact solution from the MIP model is known. However, the metaheuristics have the advantage in terms of solution computation time compared to the MIP models.

In order to visualize the convergence of the solution search space, the objective cost computed from the solutions of each iteration for the instances S2 and S3 without transshipment is plotted in a graph. Both the instances are run with seven multi-start points for two different cooling parameters. The first run is conducted with a slow convergence parameter performing more number of iterations and the second run is conducted with a fast convergence parameter performing less number of iterations. Table 7 represents the cooling parameter used for 2 different runs of instance S2. Figure 25 and Figure 26 denote the objective cost for different iterations of instance S2 with a different set of cooling parameters.

Table 7: Cooling parameters - Instance S2

Instance – S2 (Without transshipment)	Run 1	Run 2
Start temp	100	50
Markov chain length	100	30
Number of no improvement to stop	50	10
Decrement factor	0.99	0.55
End temp	0.1	1
Time (s)	6	4

[redacted]

Figure 25: Performance of Instance S2 - Run 1

Censored

Figure 26: Performance of Instance S2 - Run 2

It can be observed that GRASP-ALNS algorithm starts the initial solution with a high objective cost value (represented as high spikes in the graph) that is found as a result of the construction heuristics. The start solution is improved by the improvement phase until the stop condition. The parameter settings of the algorithm control the termination of the algorithm. It can be inferred from Figure 25 that the solution converges within 20 iterations. The rest of the 80 iterations performed to conduct the local search does not lead to an improvement in the solution. The algorithm visits the same solution again and again during this phase, as the number of local neighbour for small size instances is less. The inference from Figure 26 is that the second phase of the algorithm is stopped after a few iterations of the saturation point where no improvement in solution is found. Hence, for small instances like S2, the need for running the algorithm for more number of iterations is not the right option due to a small search space. The best objective obtained from the two runs for the instance S2 is the same.

Table 8 denotes the cooling parameter used for the instance S3, and Figure 27 and Figure 28 represent the objective cost for different iterations.

Table 8: Cooling parameters- Instance S3

Instance S3 (without transshipment)	Run 1	Run 2
Start temp	100	50
Markov chain length	100	20
Number of no improvement to stop	10	10
Decrement factor	0.95	0.55
End temp	0.1	1
Time (s)	390	101

[redacted]

Censored

Figure 27: Performance of Instance S3 -Run 1

[redacted]

Censored

Figure 28: Performance of Instance S3 - Run 2

Figure 27 represents the slow convergence of solution space where the number of the local neighbourhood is more. There could be a possibility where the local neighbourhood is visited again and again for the first run. The inference from Figure 28 explains that the convergence is within 30 iterations. The second phase of the algorithm is stopped after 30 iterations for the second run. Hence, the number of solutions visited repeatedly is reduced. The comparison of the best objective indicates that the objective value could be computed even with a fast convergence parameter. The multi-start procedure ensures that the solution computed is reliable even with the fast convergence.

The algorithm facilitates the option to control the search process for finding an optimal solution by varying the parameter settings. Altering these settings enables the user to get the best possible solution to the pickup and delivery problem within the time available to generate the schedule. However, the consideration of transshipment increases the time required for computing the optimal solution with transshipment. Hence, the decision to investigate the transshipment opportunities should be made based on time available for planning. It is recommended to use the algorithm without considering the transshipment terminals for large instances and investigate transshipment opportunities only when an adequate amount of time is available to generate the solution. The inclusion and exclusion of the transshipment terminals can be done easily by changing one binary variable corresponding to whether the terminal considered is a hub terminal or not.

6.3. Analyzing the benefits of transshipment opportunities

The transshipment scenarios are generated when cargo routes overlap. The overlap in the container route leads to the consolidation of the containers at transshipment terminals by the barges transporting them. The solution to the problem instances is analysed in order to identify the tradeoff between the direct shipment and transshipment scenarios. The transshipment option is selected if the objective cost is better with transshipment compared to direct shipping. The comparison between the individual cost parameters of the instances S1 and S2 are considered. Test instances S3 and S4 are not considered for the analysis as the solution space is huge and the solutions for scenarios with transshipment needs more computation time for finding the best solution whereas the algorithm performs consistently for smaller instances. Moreover, all the container request are transported without any penalty cost in the instances S1 and S2. The exclusion of the penalty cost for the instances S1 and S2 are observed because of using more number of barges for transporting few container requests between less number of terminals in the service network compared to the instances S3 and S4. Hence, S1 and S2 are the potential instances for analysing the main cost components for the transshipment tradeoff. The different cost components of the objective function are plotted in a graph for the test instances S1 and S2. The penalty cost components are ignored as the penalty cost incurred for all the runs are zero. Table 9 represents the average cost components for the objective function for the test instance S1. The comparison between the scenario with transshipment terminal and without transshipment terminal is analysed.

Table 9: Cost components - Instance S1

[redacted]

Censored

Figure 29 represents the comparison between the average values of the total cost component with and without transshipment.

[redacted]

Figure 29: Objective cost comparison -Instance S1

Figure 30 represents the share of individual cost components towards the total objective cost function for the test instance S1. The penalty cost components such as late delivery penalty, not delivery penalty and violation of call size penalty are not included in the graph as the values are zero. The instance S1 use the same number of barges and perform the same number of terminal visits. Hence, the trade-off between other cost components contributing to the advantage in transshipment is analysed. Figure 31 provides a comparison between the container handling cost and barge travel cost components for scenarios with and without transshipment. The handling cost refers to the cost of lifting and handling the containers during the loading and unloading at a terminal. The travel cost refers to the cost incurred by the barge to travel from a terminal to another terminal.

[redacted]

Figure 30: Cost components - Instance S1

[redacted]

Censored

Figure 31: Travel cost VS Handling cost - Instance S1

It can be inferred from Figure 31 that the cost of travel is more for the scenario without transshipment, and cost of handling containers is more for the scenario with transshipment. When analysing the solution route of the barge, it is inferred that the cost-benefit from additional handling of the containers at the transshipment terminal is more than the cost-benefit incurred from the direct shipment. We now analyse the test instance S2 with similar cost breakdown used above. Table 10 provides the cost breakdown of the transportation scenarios with and without transshipment for test instance S2. The penalty cost components are zero, and the number of barges used is the same for the scenarios with and without transshipment. However, the difference between the cost of travel for barges is very less when analysing the scenarios with and without transshipment. Hence, the comparison is made between the cost components of average container handling cost and the average cost incurred for the terminal visit.

Table 10: Cost components - Instance S2

[redacted]

Censored

Figure 32 represents the comparison between the average total cost for scenarios with and without transshipment for test instance S2. The cost difference is not as huge compared to the test instance S1. But the scenario with transshipment performs better than the scenario without transshipment. Figure 33 represents the individual cost components of the objective function for the instance S2, excluding the penalty components. The results from the cost components indicate that the cost of using the barge are the same indicating that the number of barges used for the solution with and without transshipment is same. The difference in the cost is observed for the components such as the number of terminals visited, the total number of container handling operation performed for the scenarios with and without transshipment. There is a deviation in the average travelling cost of the barges, but this deviation is very small.

Censored

Figure 32: Objective cost comparison -Instance S2

Censored

Figure 33: Cost components - Instance - S2

Figure 34 compares the average cost component of container handling and terminal visit cost from the results of the test instance S2. The terminal visit cost is an indirect penalty cost implemented in the solution approach in order to benefit from reducing the number of terminals visited by the barge. It can be inferred that the container handling cost for the scenario without transshipment is less than the scenario with transshipment. This is also due to the handling of additional containers in transshipment terminals similar to the case explained, for instance, S1. The terminal visit cost is less for the scenario with transshipment compared to the scenario without transshipment. This effect is due to the consolidation of containers at the transshipment terminals where the containers are dropped by a barge at an intermediate terminal in order for another barge to pick-up. When the second barge picking the transshipped container is already destined to the delivery terminal of the container, the additional visit to the delivery terminal by the first barge is reduced. Hence, there is a decrease in the total number of terminals visited due to the consolidation activities happening at the transshipment terminal.

[redacted]

Figure 34: Handling cost VS terminal visit cost - Instance S2

From the above analysis, it is evident that the tradeoff is usually made between the cost saved in travelling additional routes by different barges with direct shipping versus the cost of handling the containers at transshipment terminal and shipping with transshipment scenarios. The increase in the number of container handling operation due to transshipment leads to other problems such as stacking of containers at transshipment terminals and increasing the workload at transshipment terminals. The main findings from the results of the experiments are listed in Figure 35.

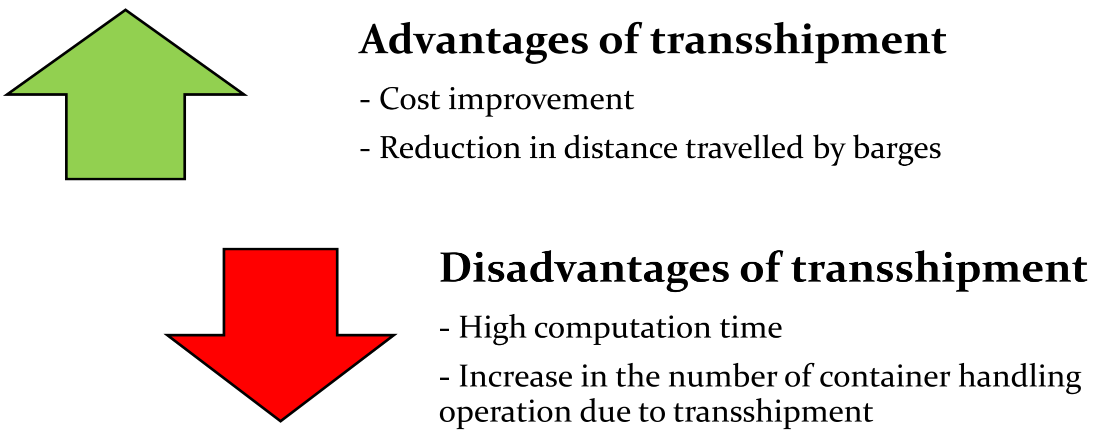

Figure 35: Pros and cons of transshipment

The complexity of the pickup and delivery problem increases with the increase in the number of container request and the number of barges considered. The complexity further increases with the increase in the number of transshipment terminals considered in the service network. When there are more transshipment terminals in the problem, the options for transporting each container through each transshipment terminal is analysed. This leads to a very high run time when multiple transshipment terminals are considered. For instance, to complete two iterations of the two-phase heuristic, it takes more than 1 hour for the instance with 30 container requests, 4 barges, 6 terminals and one transshipment terminal. The same behaviour is observed for the real-life instance with 14 container requests, 5 barges and 12 terminals with one transshipment terminal. Similar inferences are made by (Qu & Bard, 2012) for the problem instances used by them.

6.4. Recommendations from the research

The primary goal of this research, to develop a routing and scheduling algorithm for the barges operating in the service networks using optimization techniques is satisfied with the GRASP-ALNS algorithm developed as a part of the solution approach. The output parameters discussed in the conceptual framework in Section 3.7 such as barge schedule for the barge operating system, container schedule for the terminal operating system and the decision to accept or reject a container request is obtained as the output of the algorithm. The objective function computed based on these output parameters is used to evaluate the different alternative solutions. The best alternative with the lowest cost objective is selected as the proposed solution option for the pickup and delivery problem.

Recommendations for implementing the solution to the existing system

The existing barges in the service network are operating between different regions of NorthWest Europe on a fixed weekly schedule. There are several restrictions for barges to not visit a terminal due to various reasons such as geographical restriction and personal reasons for the barge operators. There are other limitations such as berth capacity restrictions for barges to load and unload containers in the terminal, storage capacity for storing containers in the terminals, assignment of resource at the container terminal to handle the containers etc. The solution to the barge routing problem from the algorithm does not take these limitations into account while computing the route. The penalty cost and the operational cost at the terminal were considered based on the assumptions from related works and experts openion and these cost may vary between different operators in real-life when compared to the parameters used for the study. The solution to the barge routing problem should be adapted to these restrictions and limitations in order to make use of the schedule generated by the algorithm effectively. To implement the algorithm to Cofano's software, Cofano needs to appropriately address the coordination of involved operations in order to fill the gap between the solution of the algorithm and the other limitations discussed above. Hence, the solution to pick-up and delivery problems can be incorporated into functional integration approaches similar to those discussed in (Lalla-Ruiz et al., 2015) and (Exposito-Izquierdo et at., 2019), in order to provide a comprehensive and realistic solution.

Recommendations for adding a new request to the existing barge schedule

The algorithm uses two different phases, one for construction and the second for the improvement of the constructed schedule; the improvement phase acts as an alternative for Cofano to investigate in improving the existing schedule of barges. The unassigned container request is stored in an open shipment pool and is inserted into the partial solution. This improvement phase can be used to fit the new container request into the created barge schedule and adjust the current barge schedule.

Recommendation for improving the computational time of the algorithm

The computation time for the algorithm to iterate all the possible insertion position for a container request through each transshipment terminal is very high. (Qu & Bard, 2012) make use of cache memory in their heuristics approach for the PDPT problem in order to prevent repeated checking of the same container request insertion into the solution sequence during different iteration. This is achieved by storing the best insertion position for each container request and the solution sequence. The insertion position is retrieved from the stored solution during the later stage rather than performing the insertion and removal again. The technique helps them to improve computation time even with considering the transshipment opportunities.

Since the tradeoff for transshipment is between the cost of travelling extra distance with direct shipment and cost of handling container request at a transshipment terminal, the transshipment opportunities are advantageous only for the group of container request with less number of containers as the cost to handling these containers at transshipment terminal is minimal. Hence, only the container request with less number of containers can be considered for transshipment rather than the entire set of container requests. Considering the cache memory for insertion strategy along with iterating the transshipment option only for selected request with less number of containers can lead to a huge improvement in the reduction of computation time to PDPT problem.

Recommendations for investigating the potential transshipment hub locations in the service network design

The other recommendation related to the transshipment is the investigation of the potential hub terminals in the service network. The potential hub or transshipment terminals can be identified using the algorithm based on the futuristic demand data. Simulation models based on the schedules from the algorithm can be used to check if the hub terminals considered can be a potential option for the transshipment of cargo.

6.5. Conclusion

The last set of research questions related to the inference from the results of the experiments based on the trade-off between individual cost components, the performance of the algorithm and the drawbacks of the heuristics were answered in this chapter. Insights for implementing the solution algorithm to develop a realistic barges route and schedule were explained. The recommendation to further improve the performance of the algorithm by considering the selected container request for transshipment opportunities and improve the solution generation time has been suggested.

Chapter 7: Conclusion

This chapter concludes the thesis research by summarising the findings of the research. Suggestions for future work are discussed later. The research is focused on developing a scheduling algorithm that generates a solution to the barge routing and scheduling problem. Different sub-research questions answered in different chapters of this thesis helped us to answer the main research question. The main research question and the sub-questions are as follows:

Main research question: *How can optimization techniques be used for improving the routing and scheduling of barges in the transportation network?*

1. *How is the barge routing and scheduling system working in the existing service network design?*
2. *What has been proposed in the literature for solving the barge routing and scheduling problem?*
3. *How should the solution approach be implemented for the barge routing and scheduling problem?*
4. *How does the solution approach perform for different scenarios for the existing service network design?*
5. *What are the recommendations from the results of the experiments?*

In order to answer the main research question, the basic subquestions pertaining to the working of existing operations in the service network were answered. The nature of the problem was identified as a pickup and delivery problem with transshipment. The first research sub-questions were answered in Chapter 2 by detailed analysis of the problem context and the existing operations.

The next step was the identification of the problem category by analysing the different works of literature. The category of the problem was identified to be an offline operational level decision problem related to linear shipping using barges. Various solution approaches used in different works of literature were analysed. The GRASP-ALNS heuristic approach was selected as the solution method due to its proofed capability for providing a high-quality solution within reasonable times for similar problems.

The pickup and delivery problem with transshipment opportunity tackled by Cofano was modelled as a MIP model. The GRASP-ALNS metaheuristic was used to design a two-phase heuristic solution approach. The first phase of the GRASP-ALNS algorithm makes use of different construction heuristics to construct the initial solution. The second phase of the algorithm performs the improvement of the constructed solution by destroying the solution and reconstructing it sequentially. Various destruction and repair heuristics have been implemented in order to generate a neighbour solution. The designed solution approach make use of simulated annealing as a way to intensify the search and accept the worsening solutions to move out of the local optima and continue the search to improve the solution. The third set of research questions were answered as a result of the designed algorithm.

The constructed solution approaches were evaluated with different testing techniques. The performance of the MIP model and algorithm were compared with the help of the test instances generated. Different experimental setups representing the inclusion of the transshipment terminals and exclusion of the transshipment terminals in the existing service network were considered to evaluate the performance of the GRASP-ALNS algorithm on the real-life instances.

Final discussion based on the inference from the results and analysis of the experiments and the algorithms' performance was explained in Chapter 6. The drawbacks of the proposed algorithm, along with the advantages, were discussed in Chapter 6. The recommendation for Cofano software solution to implement the solution from the algorithm was provided along with the discussions of other potential benefits from the solution approach answering the last set of sub-research questions.

The main research question was answered by answering the five sub-research questions, sequentially. Thus, optimization techniques have been used to solve the barge routing and scheduling problem by developing a hybrid metaheuristic and solving the container pickup and delivery problem with transshipment.

7.1. Future Scope:

Several recommendations for Cofano software solutions have been suggested based on the developed barge scheduling algorithm. Few recommendations can be suggested as the future scope of the research, such as:

- Considering the pickup and delivery problem and analysing the effect of having a centralised or decentralised transshipment terminal in the service network can be a challenging area to be investigated. Analysis of different hub configuration can also be performed for the hinterland service network considered.
- A real-life solution to the barge scheduling problem involves synchronization of solution from different problems such as berth allocation problem, container stacking problem, incorporation of terminal entry restrictions and terminal operating hours into the solution approach. Hence, the scope of the problem and solution approach can be incorporated into functional integration approaches similar to those as in (Lalla-Ruiz et al., 2015) and (Exposito-Izquierdo et at., 2019), so that holistic solutions can be provided.
- The proposed solution approach considers heterogeneous fleet of barges that vary with speed, capacity and operational cost. Hence, the model can be easily adapted for multiple modes of transportation. The intermodal transportation problem can be considered and solved by adapting the proposed heuristic approach.
- The length of the planning horizon affects consolidation opportunities. There is more number of consolidation opportunities for a larger planning horizon compared to a shorter one. But, the uncertainty in the demand of container restricts the opportunity for consideration of the larger planning horizon. Investigating the length of the planning horizon and its effect on consolidation is a challenging area to be explored.

Bibliography

Alfandari, L., Davidović, T., Furini, F., Ljubić, I., Maraš, V., & Martin, S. (2019). Tighter MIP models for Barge Container Ship Routing. *Omega (United Kingdom)*, *82*, 38–54. https://doi.org/10.1016/j.omega.2017.12.002

An, F., Hu, H., & Xie, C. (2015). Service network design in inland waterway liner transportation with empty container repositioning. *European Transport Research Review*, *7*(2). https://doi.org/10.1007/s12544-015-0157-5

Archetti, C., & Speranza, M. G. (2014). A survey on matheuristics for routing problems. *EURO Journal on Computational Optimization*, *2*(4), 223–246. https://doi.org/10.1007/s13675-014-0030-7

Bierwirth, C., & Meisel, F. (2015). A follow-up survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, *244*(3), 675–689. doi.org/10.1016/j.ejor.2014.12.030

Braekers, K., Caris, A., & Janssens, G. K. (2013). Optimal shipping routes and vessel size for intermodal barge transport with empty container repositioning. *Computers in Industry*, *64*(2), 155–164. doi.org/10.1016/j.compind.2012.06.003

Battarra, M., Cordeau, J.F., Iori, M., (2014). Pick-up-and-delivery problems for goods transportation. Vehicle routing: problems, methods, and applications. *MOS/SIAMseries on optimization, 2nd ed. 161–192.*

Brønmo, G., Christiansen, M., & Nygreen, B. (2007). Ship routing and scheduling with flexible cargo sizes. *Journal of the Operational Research Society*, *58*(9), 1167–1177. https://doi.org/10.1057/palgrave.jors.2602263

Caris, A., Macharis, C., & Janssens, G. K. (2011). Network analysis of container barge transport in the port of Antwerp by means of simulation. *Journal of Transport Geography*, *19*(1), 125–133. https://doi.org/10.1016/j.jtrangeo.2009.12.002

Caris, A. (2011). Simulation and optimisation of intermodal barge transport networks. *4or*, *9*(2), 211–214. https://doi.org/10.1007/s10288-010-0142-8

Castillo-Villar, K. K., González-Ramírez, R. G., Miranda González, P., & Smith, N. R. (2014). A heuristic procedure for a ship routing and scheduling problem with variable speed and discretized time windows. *Mathematical Problems in Engineering*, *2014*. https://doi.org/10.1155/2014/750232

Cheung, R. K., & Chen, C.-Y. (1998). A Two-Stage Stochastic Network Model and Solution Methods for the Dynamic Empty Container Allocation Problem. *Transportation Science.*

Choong, S. T., Cole, M. H., & Kutanoglu, E. (2002). Empty container management for intermodal transportation networks. *Transportation Research Part E: Logistics and Transportation Review*, *38*(6), 423–438. https://doi.org/10.1016/S1366-5545(02)00018-2

Christiansen, M., Fagerholt, K., Nygreen, B., & Ronen, D. (2013). Ship routing and scheduling in the new millennium. *European Journal of Operational Research*, *228*(3), 467–483. https://doi.org/10.1016/j.ejor.2012.12.002

Crainic, T. G. (2000). Service network design in freight transportation. *European Journal of Operational Research*, *122*(2), 272–288. https://doi.org/10.1016/S0377-2217(99)00233-7

Crainic, T. G. (2002). *Long-Haul Freight Transportation*. https://doi.org/10.1007/0-306-48058-1

Crainic, T. G., & Kim, K. H. (2006). Intermodal Transportation. *Transportation Research Record*, (November).

Creemers, S., Woumans, G., Boute, R., & Beliën, J. (2017). Tri-vizor uses an efficient algorithm to identify collaborative shipping opportunities. *Interfaces*, *47*(3), 244–259. https://doi.org/10.1287/inte.2016.0878

Danloup, N., Allaoui, H., & Goncalves, G. (2018). A comparison of two meta-heuristics for the pick-up and delivery problem with transshipment. *Computers and Operations Research*, *100*, 155–171. https://doi.org/10.1016/j.cor.2018.07.013

Davidovi, T. (2011). Combinatorial Formulation Guided Local Search for Inland Waterway Routing and Scheduling. *13th IASTED International Conference on Control and Applications*. https://doi.org/10.2316/P.2011.729-091.

De Armas, J., Lalla-Ruiz, E., Expósito-Izquierdo, C., Landa-Silva, D., & Melián-Batista, B. (2015). A hybrid GRASP-VNS for ship routing and scheduling problem with discretized time windows. *Engineering Applications of Artificial Intelligence*, *45*, 350–360. https://doi.org/10.1016/j.engappai.2015.07.013

Edirisinghe, N. C. P., Bowers, M. R., & Agarwal, A. (2010). Generalised pick-up and delivery problem with dynamic time windows. *International Journal of Operational Research*, *8*(1), 85–109. https://doi.org/10.1504/IJOR.2010.033105

Exposito-Izquierdo, C., de Armas, J., Lalla-Ruiz, E., Melian-Batista, B., & Marcos Moreno-Vega, J. (2019). Multi-stage approach for the transshipment of import containers at maritime container terminals. *IET Intelligent Transport Systems*, *13*(4), 714-728. https://doi.org/10.1049/iet-its.2018.5147

Fazi, S. (2014). *Mode selection, routing and scheduling for inland container transport*. https://doi.org/10.6100/IR777919

Fazi, S., Fransoo, J. C., & Van Woensel, T. (2015). A decision support system tool for the transportation by barge of import containers: A case study. *Decision Support Systems*, *79*, 33–45. https://doi.org/10.1016/j.dss.2015.08.001

Fazi, S., & Roodbergen, K. J. (2018). Effects of demurrage and detention regimes on dry-port-based inland container transport. *Transportation Research Part C: Emerging Technologies*, *89*(January), 1–18. https://doi.org/10.1016/j.trc.2018.01.012

Foss, E., Myklebust, T. N., & Andersson, H. (2016). *A New Optimization-based Algorithm for a Maritime Inventory Routing Problem* (Norwegian University of Science and Technology). Retrieved from https://brage.bibsys.no/xmlui/bitstream/handle/11250/2433815/15396_FULLTEXT.pdf?sequence=1&isAllowed=y

Fu, Q., Liu, L., & Xu, Z. (2010). Port resources rationalization for better container barge services in Hong Kong. *Maritime Policy and Management*, *37*(6), 543–561. https://doi.org/10.1080/03088839.2010.514955

González-Ramírez, R. G., Askin, R. G., R. Smith, N., & Villalobos, R. (2009). Shipment consolidation by terminals and vehicles. *Revista de Matemática: Teoría y Aplicaciones*, *16*(1), 178–187. https://doi.org/10.15517/rmta.v16i1.1426

Groothedde, B., Ruijgrok, C., & Tavasszy, L. (2005). Towards collaborative, intermodal hub networks. A case study in the fast moving consumer goods market. *Transportation Research Part E: Logistics and Transportation Review*, *41*(6 SPEC. ISS.), 567–583. https://doi.org/10.1016/j.tre.2005.06.005

Hagen, L. van der. (2016). An adaptive large neighborhood search heuristic for the Pick-up and Delivery Problem with Time Windows and Scheduled Lines. *Computers and Operations Research*, *72*, 12–30. https://doi.org/10.1016/j.cor.2016.01.018

Hemmati, A., Hvattum, L. M., Fagerholt, K., & Norstad, I. (2014). Benchmark suite for industrial and tramp ship routing and scheduling problems. *Infor*, *52*(1), 28–38. https://doi.org/10.3138/infor.52.1.28

Hemmati, A., Stålhane, M., Hvattum, L. M., & Andersson, H. (2015). An effective heuristic for solving a combined cargo and inventory routing problem in tramp shipping. *Computers and Operations Research*, *64*, 274–282. https://doi.org/10.1016/j.cor.2015.06.011

Homayouni, S. M., & Fontes, D. B. M. M. (2018). *Metaheuristics for Maritime Operations*. ISTE Ltd, John Wiley & Sons, Inc.

Houwen, A. van der. (n.d.). *The port of Rotterdam and the use of intermodal transportation*.

Kim, H. J., Chang, Y. T., Lee, P. T. W., Shin, S. H., & Kim, M. J. (2008). Optimizing the transportation of international container cargoes in Korea. *Maritime Policy and Management*, *35*(1), 103–122. https://doi.org/10.1080/03088830701849084

Kjeldsen, K. H. (2012). *Routing and Scheduling in Liner Shipping* (AARHUS UNIVERSITY). https://doi.org/10.1287/trsc.l030.0036

Konings, R. (2006). Hub-and-Spoke Networks in Container-on-Barge Transport. *Transportation Research Record*, 23–32.

Konings, R., Kreutzberger, E., & MaraŠ, V. (2013). Major considerations in developing a hub-and-spoke network to improve the cost performance of container barge transport in the hinterland: The case of the port of rotterdam. *Journal of Transport Geography*, *29*(2013), 63–73. https://doi.org/10.1016/j.jtrangeo.2012.12.015

Konings, R., & Maras, V. (2011). Hub-and-spoke networks to increase the scope of container-on-barge transport. *Transportation Research Record*, (2222), 90–97. https://doi.org/10.3141/2222-12

Korsvik, J. E., Fagerholt, K., & Laporte, G. (2010). A tabu search heuristic for ship routing and scheduling. *Journal of the Operational Research Society*, *61*(4), 594–603. https://doi.org/10.1057/jors.2008.192

Lalla-Ruiz, E., de Armas, J., Expósito-Izquierdo, C., Melián-Batista, B., & Moreno-Vega, J. M. (2015). A Multi-stage Approach Aimed at Optimizing the Transshipment of Containers in a Maritime Container Terminal. In R. Moreno-Díaz, F. Pichler, & A. Quesada-Arencibia (Eds.), *Computer Aided Systems Theory – EUROCAST 2015 : 15th International Conference, Las Palmas de Gran Canaria, Spain,* (pp. 255-262).

Li, L., Negenborn, R. R., & De Schutter, B. (2015). Intermodal freight transport planning - A receding horizon control approach. *Transportation Research Part C: Emerging Technologies*, *60*, 77–95. https://doi.org/10.1016/j.trc.2015.08.002

Li, S., Negenborn, R. R., & Lodewijks, G. (2017). Planning inland vessel operations in large seaports using a two-phase approach. *Computers and Industrial Engineering*, *106*, 41–57. https://doi.org/10.1016/j.cie.2017.01.027

Lin, D. Y., & Liu, H. Y. (2011). Combined ship allocation, routing and freight assignment in tramp shipping. *Transportation Research Part E: Logistics and Transportation Review*, *47*(4), 414–431. https://doi.org/10.1016/j.tre.2010.12.003

Lin, D. Y., & Tsai, Y. Y. (2014). The ship routing and freight assignment problem for daily frequency operation of maritime liner shipping. *Transportation Research Part E: Logistics and Transportation Review*, *67*, 52–70. https://doi.org/10.1016/j.tre.2014.04.003

Liu, J., & Pang, K. W. (2014). An integrated model for ship routing with transshipment and berth allocation. *IIE Transactions (Institute of Industrial Engineers)*, *46*(12), 1357–1370. https://doi.org/10.1080/0740817X.2014.889334

Maraš, V. (2008). Determining optimal transport routes of inland waterway container ships. *Transportation Research Record*, (2062), 50–58. https://doi.org/10.3141/2062-07

Maraš, V., Lazić, J., Davidović, T., & Mladenović, N. (2013). Routing of barge container ships by mixed-integer programming heuristics. *Applied Soft Computing Journal*, *13*(8), 3515–3528. https://doi.org/10.1016/j.asoc.2013.03.003

Masson, R., Lehuédé, F., & Péton, O. (2017). An Adaptive Large Neighborhood Search for the Pick-up and Delivery Problem with Transfers. *Computers and Operations Research*, *88*(2010), 1–14. https://doi.org/10.1016/j.cor.2017.06.012

Moccia, L., Cordeau, J.-F., Laporte, G., Ropke, S., & Valentini, M. P. (2006). Modeling and Solving a Multimodal Transportation Problem with Flexible-time and Scheduled Services. *Networks*, *47*(1), 26–36. https://doi.org/10.1002/net

Moher, D. L. (2009). *Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement. PLoS medicine, pp. 6(7), e1000097.*

Network, M. N. (2019, October). *10 World's Biggest Container Ships in 2017. Opgehaald van https://www.marineinsight.com/know-more/10-worlds-biggest-container-ships-2017/*

Notteboom, T., & Konings, R. (2004). Network dynamics in container transport by barge. *Belgeo*, (4), 461–477. https://doi.org/10.4000/belgeo.13402

Port Authority of Rotterdam. (2018). *Facts and Figures Port of Rotterdam.*

Port of Rotterdam. (2014*). Progress report 2014 - Port Vision 2030.*

Port of Rotterdam. (2019). *Port of Rotterdam web page. Opgehaald van https://www.portofrotterdam.com/en*

Port of Rotterdam Authority. (2019). *Half yearly report 2019.*

Pruijn, S. E. (2018). *An algorithm for scheduling containers on barges*. Retrieved from http://essay.utwente.nl/74684/

Qu, Y., & Bard, J. F. (2012). A GRASP with adaptive large neighborhood search for pick-up and delivery problems with transshipment. *Computers and Operations Research*, *39*(10), 2439–2456. https://doi.org/10.1016/j.cor.2011.11.016

Rais, A., Alvelos, F., & Carvalho, M. S. (2014). New mixed integer-programming model for the pick-up-and-delivery problem with transshipment. *European Journal of Operational Research*, *235*(3), 530–539. https://doi.org/10.1016/j.ejor.2013.10.038

Riessen, B. Van, Negenborn, R. R., Dekker, R., & Lodewijks, G. (2015). Service network design for an intermodal container network with flexible transit times and the possibility of using subcontracted transport. *International Journal of Shipping and Transport Logistics*, *7*(4), 457–478. https://doi.org/10.1504/IJSTL.2015.069683

Rivera, A. E. P. (2018). Anticipatory Freight Scheduling in Synchromodal Transport. https://doi.org/10.3990/1.9789036545716

Saeedi, H., Wiegmans, B., Behdani, B., & Zuidwijk, R. (2017). European intermodal freight transport network: Market structure analysis. *Journal of Transport Geography*, *60*, 141–154. https://doi.org/10.1016/j.jtrangeo.2017.03.002

Sharypova, K. (2014). Optimization of Hinterland Intermodal Container Transportation. https://doi.org/10.6100/IR766077

Stålhane, M., Andersson, H., Christiansen, M., & Fagerholt, K. (2014). Vendor managed inventory in tramp shipping. *Omega (United Kingdom)*, *47*, 60–72. https://doi.org/10.1016/j.omega.2014.03.004

Takoudjou, R. T., Deschamps, J. C., & Dupas, R. (2012). A MIP formulation for the pick-up and delivery problem with time window and transshipment. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, *14*(PART 1), 333–338. https://doi.org/10.3182/20120523-3-RO-2023.00109

UNCTAD. (2008). The modal split of international goods transport. Transport Newsletter. *UNCTAD/SDTE/TLB/MISC/2008/1.*

Visser, J., Konings, R., Wiegmans, B., & Pielage, B.-J. (2009). A new hinterland transport concept for the port Rotterdam: Organisational or technological challenges? *CD Delft. 3450. 31-15.*

Vornhusen, B., Wang, X., & Kopfer, H. (2014). Vehicle routing under consideration of transhipment in horizontal coalitions of freight carriers. *Procedia CIRP*, *19*(C), 117–122. https://doi.org/10.1016/j.procir.2014.05.008

World shipping council. (2019). Opgehaald van History of Containerization: *http://www.worldshipping.org/about-the-industry/history-of-containerization*

Zenker, M., Emde, S., & Boysen, N. (2016). Cyclic inventory routing in a line-shaped network. *European Journal of Operational Research*, *250*(1), 164–178. https://doi.org/10.1016/j.ejor.2015.10.067

Zweers, B. G., Bhulai, S., & van der Mei, R. (2019). Optimizing barge utilization in hinterland container transportation. *Naval Research Logistics*, *66*(3), 253–271. https://doi.org/10.1002/nav.21837

Appendix

Appendix A

<u>The systematic literature review process</u>

According to the method of systematic literature review explained by Preferred Reporting Items for Systematic Review and Meta-Analysis (PRISMA), (Moher, 2009), there are four phases involved in the systematic literature review. The first phase is the identification phase, where the research papers are identified from multiple databases. Scopus and Web of Science databases were primarily used for the literature search. The search was conducted based on the keywords with various criteria such as citation count, dates of publication of the article and language of the article (restricted to English) set as filters. The second phase is related to the screening process, where duplicate articles are removed. EndNote, which is an industry-standard software tool for publishing and managing bibliographies, citations and references, was used for this screening phase. The final list after removal of duplicates literature is generated. The third phase is a screening phase where each research paper is accessed for the relevance to the scope of our research. Relevant articles were filtered based on the significance of the abstract to our research. Google Scholar, another related article from magazines, newsletters, and annual reports and other websites were also used for additional information

This section describes the procedure used to identify the search phrase and the elimination strategies to exclude unwanted literature. The literature review starts with the identification of the keywords to search the relevant literature from the selected databases. The goal of the research is to create a routing and scheduling algorithm for the planning of barges. Hence, the first search phrase is related to the same. We use the search phrase such as ship scheduling and routing, hinterland container supply chain, as well as barge scheduling and routing. There is the identification of container consolidation and transshipment problem followed by the barge scheduling problem to schedule the containers on the barges. The next set of literature analysis is made for container consolidation strategies and scheduling strategies. This is followed by investigating the transshipment opportunities in barges and inland transportation. In addition to the above search phrase, the 'hub and spoke' network for barge operation is used to identify literature related to the advantages of the transshipment and use of transshipment hub concerning the inland transportation. The phrase pick-up and delivery problem and transshipment are used to get an insight into the mathematical modelling of the pick-up and delivery problem with transshipment. The final sets of search phrases are related to the optimization of routing and scheduling problems. The phrase 'optimization' is also used with the phrase 'container'; to identify the works of literature that are related to container bundling and scheduling optimization problems. The words such as "barge", "container" and "inland" are used with the "AND" operator to search for literature that is more specific to the research problem. The lists of search phrases used in the different databases are displayed in Table 11 and Table 12.

Other literature works, such as thesis projects and web articles related to the research, were also identified. The lists of additional literature identified were added to the final list. The next level of the filter was done by eliminating the articles that are having less connection to the research by reading the abstract. For removing the article which is not related to the scope of the research, it is decided to restrict the routing and scheduling problem to the barge transportation only. However, train, truck and cargo ship modes of transportation are considered in a few cases to investigate the consolidation and transshipment opportunities that have been used in different literature. Literature works related to the metaheuristics and matheuristics that have been developed for general routing problems were also included in the final list.

As a part of the fourth and final phase, a literature review with the key findings from research papers that were identified through the systematic literature review process is consolidated and explained in Chapter 3. This systematic procedure is done for easy consolidation and easy handling of the knowledge gained through literature review. The results of each phase are explained with the help of the below tables.

Search Protocol

Table 11: Search protocol for Scopus database

Search string	Scope	Date of search	Data range	Include	Nr. of entries
Search protocol for Scopus					226
"Barge" AND "scheduling"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	33
"Barge" AND "routing"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	19
"Ship routing" AND "Ship scheduling"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	7
"Hinterland" AND "container" AND "chain" AND "barge"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	7
"container on barge transport"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	4
"container consolidation"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	10
"Container" AND "allocation" AND "Barge"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	8
"Container" AND "Routing" AND "Barge"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	4
"Container" AND "bundling" AND "Barge"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	4
"Container" AND "Scheduling" AND "Barge"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	8

"Hub and Spoke" AND "Barge"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	8
"tramp shipping"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	35
"Collaborative shipping"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	2
"Transshipment" AND "Barge" AND "Container"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	3
"Transshipment" AND "container" AND "Inland"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	18
"crow shipping" OR "crow shipping"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	0
"Inland" AND "shipping" AND "container" AND "optimization"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	26
"optimization" AND "routing" AND "Container" AND "scheduling" AND "inland"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	1
"Optimization" AND "barge" AND "routing" AND "scheduling"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	2
"heuristics" AND "inland" AND "container"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	16
"heuristics" AND "barge" AND "routing"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	3
"Pick-up and delivery problem" and "transshipment"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	8

Table 12: Search protocol for Web of Science Database

Search string	Scope	Date of search	Data range	Include	Nr. of entries
Search protocol for Web of science					196
"Barge" AND "scheduling"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	13
"Barge" AND "routing"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	10
"Ship routing" AND "Ship scheduling"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	8
"Hinterland container chain" AND "barge"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	6
"container on barge transport"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	2

"Container consolidation"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	4
"container" AND "consolidation" AND "Inland"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	4
"Container" AND "allocation" AND "Barge"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	12
"Container" AND "Routing" AND "Barge"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	11
"Container" AND "bundling" AND "Barge"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	4
"Container" AND "Scheduling" AND "Barge"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	12
"Hub and Spoke" AND "Barge"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	6
"tramp shipping"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	31
"Collaborative shipping"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	2
"Transshipment" AND "Barge"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	2
"Transshipment" AND "container" AND "Inland"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	9
"Inland" AND "shipping" AND "container" AND "optimization"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	17
"crow shipping" OR "crowshipping"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	0
"optimization" AND "Container" AND "transportation" AND "inland"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	26
"Optimization" AND "barge" AND "routing" AND "scheduling"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	1
"heuristics" AND "inland" AND "container" AND "transportation"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	4
"heuristics" AND "barge" AND "routing"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	2
"Pick-up and delivery problem" and "transshipment"	Title, keyword and abstract	10/30/2019	1990-2019	Cited at least once	10

Review process

Table 13: Review process for literature

Description	Variable	Count
Total in endnote from Scopus	X	226
Total in endnote from Web of Science	Y	196
Total in endnote	A = X + Y	422
Duplicates articles	B	-177
Total after duplicate removal	C = A + B	245
Article with less connection to the scope of research	D	-197
Removed after reading the abstract, introduction, and findings	E	-13
Additional literature added	F	28
Total selected for review	G = C +D + E + F	63

Table 14: Final literature count

Description	Count
Literature with specific routing problems and solution techniques	41
Literature with general concepts	22
Total	63

Appendix B

The excel template data file is used to generate input datasheet. The excel file opens with "General Data" as the first sheet of the excel file. The help text on how to use the excel file and its buttons are explained in the note section. The content of the individual sheets is described below.

First Sheet: General data

The general data is the default sheet available in the excel template. The general data about the problem is provided in the general datasheet. The general datasheet contains the data such as number of container request, number of barges and number of terminals considered during the planning horizon along with other general data such as terminal handling time, cost of using a barge during planning horizon and the penalty value for unit late delivery of the container to the destination.

Second Sheet: Cargo data

The cargo data contains an individual record for each container request that is to be transported. The count of records depends on the number of containers considered in the general datasheet. The cargo datasheet contains the container id, its origin terminal and destination terminal, arrival time and the due date for the container request, the load of the container and the penalty value for not delivering during the planning horizon.

Third Sheet: Barge data

The barge data contains the individual records for each barge that are available during the planning horizon. The number of records for the barge data sheet depends on the number of barges parameter considered in the general datasheet. The barge datasheet contains barge related data such as starting terminal of the barge, speed of the barge, capacity limit of the barge and the cost of travel for the barge per unit distance.

Fourth Sheet: Terminal data

The terminal datasheet contains the number of terminals that are considered in the service network during the planning horizon. The count of the records in the terminal data sheet depends on the number of terminals parameter in the general datasheet. The terminal datasheet contains the distance matrix which contains the distance between any two-terminal pair considered and a binary variable denoting whether the terminal can be used as a hub terminal or not.

The user can use the available buttons to generate the template for a specific datasheet or all the datasheets at once. It should be noted that the data that are present in the datasheet will be cleared and an empty template will be created when the user creates any specific datasheet. The empty template will be based on the input parameter values provided in the General Datasheet.

Appendix C

This appendix has been removed from the public version.

Appendix D

This section contains the test scenarios for the black box and white box testing conducted on the solution approach.

Table 15: Results for Blackbox testing test cases

Scenario ID	Scenario Description	Result
1	Cargo optimal flow: cargo should be transported in the optimal route considering the barge starting location	Pass
2	Cargo optimal flow: cargo should be transported in the optimal route considering the barge operating cost	Pass
3	Cargo optimal flow: cargo should be transported in the optimal route considering the barge operating speed	Pass
4	Cargo Sequence flow: Cargo should be picked up if it is dropped	Pass
5	Cargo Sequence flow: Cargo should be Dropped if it is picked up	Pass
6	Cargo Sequence flow: Cargo should be transshipped only if it is picked up from origin terminal	Pass
7	Cargo Sequence flow: Cargo should be picked up from a transshipment terminal if it is dropped to a transshipment terminal	Pass
8	Cargo Sequence flow: Cargo should be picked up before it is being dropped	Pass
9	Cargo Sequence flow: Cargo should be dropped at a transshipment terminal before it is picked up from a transshipment terminal	Pass
10	Cargo Sequence flow: All the pick-up and drop operations at origin and transshipment terminals should be completed before the final delivery	Pass
11	Cargo Sequence flow: All the pick-up operation should happen after the arrival time of the respected cargo	Pass
12	Transshipment limit constraint: Transshipment should happen only in the hub terminal	Pass
13	Transshipment limit constraint: Transshipment should not happen in other terminals which are not hub terminal	Pass
14	Transshipment limit constraint: Max number of transshipment should not exceed the limit of max allowed transshipment for a cargo	Pass
15	Transshipment limit constraint: cargo should be transshipped only once per transshipment terminal	Pass
16	Transshipment limit constraint: the tradeoff between handling cost at a transshipment terminal and direct shipment cost should be considered while transshipping	Pass
17	No delivery penalty: Cargo should be shipped if it is optimal to be shipped	Pass
18	No delivery penalty: Cargo should not be shipped if it is optimal to be not shipped	Pass
19	No delivery penalty: penalty should be included for not shipped cargo if cargo is not shipped and the penalty for other cargo should not be added for which the cargo is shipped	Pass
20	Late delivery penalty: late delivery penalty should be considered if it is optimal	Pass
21	Late delivery penalty: late delivery penalty should not be considered if late delivery is not optimal solution rather the cargo should not be shipped and should pay no delivery penalty	Pass
22	Load of ship: if the cargo cannot be loaded, then the cargo should not be loaded even if it is the optimal route	Pass

23	Load of ship: if cargo is from the same loading terminal, then, the ship should load only up to the max load limit per voyage	Pass
24	Number of visits: Ship should not consolidate the cargo originating from the same terminal when the cargo having different arrival time and serve them in a different visit to have a tradeoff between the number of visits and penalty for delivering cargo late if the penalty is too high	Pass
25	Number of visits: Ship should consolidate the cargo and serve in the same visit for the cargo having different arrival time to have a tradeoff between the number of visits and penalty for delivering cargo late if the penalty is not high	Pass
26	Cargo delivery: Cargo should be delivered as early as possible depending on the optimal cost	Pass
27	one cargo, 2 terminal and one vessel scenario	Pass
28	Multiple cargo, 2 terminal and one vessel scenario	Pass
29	Multiple cargoes, multiple terminals, and one vessel scenario	Pass
30	Multiple cargoes, multiple terminals and multiple vessels with direct transport opportunity scenario	Pass
31	Multiple cargoes, multiple terminals and multiple vessels with Transshipment opportunity scenario	Pass
32	Transshipment opportunity: Pick-up or drop location in one route same as pick-up or delivery location of different customer in another route	Pass
33	Transshipment opportunity: two cargo route using the same hub transshipment location	Pass
34	Transshipment opportunity: two cargo route overlap but not with common pick-up and delivery	Pass
35	Cargo optimal flow: cargo should be transported in the optimal route with storing the cargo in a barge from the pick-up until it reaches the destination with serving in between routes	Pass

Table 16: Results for Whitebox testing scenarios

Serial No	Test scenario	Description	Status
1	Construction Heuristics 1: Cargo ID construction heuristics initialization	The cargo id should be added to the open shipment pool in the cargo id order	Pass
2	Construction Heuristics 2: EDD construction heuristics initialization	The cargo should be added to the open shipment pool in their earliest due date order	Pass
3	Construction Heuristics 3: EAT construction heuristics initialization	The cargo should be added to the open shipment pool according to their earliest time order	Pass
4	Construction Heuristics 4: Random Cargo order construction heuristics initialization	The cargo should be added to the open shipment pool in random order from the available pool	Pass
5	Construction Heuristics 5: Max Cargo between terminal pairs One	Consider the terminal from which maximum cargo originates and consider terminals to which maximum cargo are destined and create a matrix with a count of cargo between these two terminals considered. Add the cargo to the open shipment pool from the highest order to the lowest order	Pass
6	Construction Heuristics 6: Max Cargo between terminal pairs Two	Create a matrix with a count of cargo between any pair of origin and destination terminals considered. Add the cargo to the open shipment pool in descending order of the count from this matrix	Pass
7	Destruction heuristics 1: Random removal	Remove the cargo randomly from the solution	Pass
8	Destruction heuristics 2: minimum call	Remove cargo that is served in the visit with a minimum call	Pass

	size removal	size by the barge	
9	Destruction heuristics 3: minimum cargo handled in transshipment terminal removal	Remove the cargo that is transshipped in a transshipment terminal. The transshipment terminal with the lease number of transshipments performed is selected first	Pass
10	Destruction heuristics 4: worst cargo removal	Remove cargo which when removed leads to worst improvement in the total cost	Pass
11	Destruction heuristics 5: Shaw's cargo removal	Remove cargo that is similar to each other from the solution	Pass
12	Repair heuristics 1: Random insertion	Select random cargo from open shipment pool and add to the best possible insertion position	Pass
13	Repair heuristics 2: Greedy insertion	Select the cargo from open shipment pool that leads to maximum improvement when added to the solution and add to the solution	Pass
14	Repair heuristics 3: Regret K insertion	Consider K best insertion position for each cargo from open shipment pool and add the cargo with most regret value if not inserted	Pass
15	Repair heuristics 4: Most constrained the first insertion	Select the cargo that is having the highest constraint function (cargo that is difficult to be inserted at later point of time) and insert it first	Pass
16	Weight updating for destruction heuristics and repair heuristics	1. The weight should be updated with sigma one increment if there is a better solution found than the solution considered for the iteration 2. The weight should be updated with sigma two increments if there is not a better solution found than the solution considered for the iteration but the bad solution found is accepted 3. The weight should not be updated if there is not a better solution found than the solution considered for the iteration and the bad solution found is not accepted The number of times the selected heuristics used should be incremented by a count of one	Pass
17	Weight updating for construction heuristics	When the construction heuristics selected leads to an improvement in the best solution considered so far, then the weight of the construction heuristics should be updated	Pass
18	Probability updating for destruction and repair heuristics	The probability of all the destruction and repair heuristics should be updated based on the weights. The update should happen after performing a predefined number of iterations (Markov chain length	Pass
19	Probability updating for construction heuristics	The probability of selection of the construction heuristics should be updated based on the weight of the construction heuristics during the start of every iteration	Pass
20	Cooling parameter updating: current temperature updating	The current temperature should be decremented after performing a Markov chain number of iteration until it reaches end temperature	Pass
21	Accepting solution	1. A solution which is better than the iteration solution should always be accepted 2. A solution which is worse than the iteration solution should be accepted if the random probability is less than the acceptance probability 3. A solution which is worse than the iteration solution should	Pass

		be rejected if the random probability is more than the acceptance probability	
22	Single insertion strategy	The singe insertion strategy should check all the possible insertion position of the origin and destination nodes for cargo in all the routes of the solution sequence that are feasible	Pass
23	Double insertion strategy	The double insertion strategy should check all the possible insertion position of the origin node and the transshipment drop node in the route one and all the possible insertion position of transshipment pick-up node and a destination node in route two for all the routes and all the transshipment terminals in the solution sequence There can be a loop scenario happening that affects the future nodes visited in the route. This infeasible insertion should be avoided	Pass
24	Construction Heuristics: working	The cargo in the open shipment pool should be considered and inserted into the best possible position in the greedy approach from the candidate list	Pass
25	Removal heuristics: working	The count of cargo based on the selected degree of freedom from the solution sequence should be removed from the solution and the service start time and load values for the new solutions should be updated accordingly	Pass
26	Insertion heuristics: working	The selected cargo should be inserted according to the insertion strategy (direct or transshipment). The solution sequence and the service start time and load values should be updated accordingly	Pass
27	Stop condition: Phase One Heuristics	The cargo should be added from the open shipment pool into the solution until the open shipment pool becomes empty or until there is no feasible way to insert any of the cargo from the open shipment pool into the solution route.	Pass
28	Stop condition: Phase Two heuristics	Phase two heuristics starts with a starting temperature 1. The algorithm should end if the current temperature is less than the end temperature set in the settings 2. The algorithm should end if there are no improvements in the solution for a predefined number of iteration. this should happen irrespective of the end temperature	Pass
29	Stop Condition: GRASP-ALNS algorithm	1. The algorithm should perform phase one and phase two heuristics for a predefined number of the iteration count. 2. The algorithm should stop after performing the phase two heuristics if the run time is above the predefined time value set in the settings. This should happen irrespective of the number of iterations performed	Pass

Appendix E

For generating the test instances to evaluate the performance of the heuristics, a small transportation service network is similar to Figure 21 is considered. There are three types of terminals that are considered in the service network. The first type of terminals is the seaport terminal. The seaport terminals are usually clustered and are concentrated to a port region. For example, the port of Rotterdam contains more than 30 port terminals where barges can load the cargo. The second type of terminals is inland terminal, which is considered to be the terminals from which the export cargo are loaded, and import cargo are delivered. These terminals are scattered throughout the hinterland, and the number of terminals concentrated in a region is very less. The third type of terminals is the intermediate terminal or transshipment hub terminal. The transshipment terminal can be either a seaport terminal or an inland terminal. The decision to fix a terminal as a transshipment terminal is based on how effective a terminal can facilitate transshipment opportunities. From the literature research conducted, it can be inferred that the transshipment terminals are considered to be close to the seaport and in between the seaport and the inland terminals. There is no use in having a transshipment terminal towards the origination of a river(opposite end to the mouth of the river). Hence, it is decided to consider the intermediate terminal or the seaport terminal as transshipment terminals.

The size of the problem instances are dependent on the instance parameter such as number of container request considered (*cn*), number of barges considered *(bn)*, number of terminals in the service network *(tn)* and number of hub/ transshipment terminals among the terminals *(hn)*. There are 10 test instances that have been generated. The parameters are gradually increased throughout the test instances. The instances are considered with and without the transshipment hub terminals considered. The instances are generated with a service network similar to the example network designed in Figure 21, and the transshipment terminal is considered in the intermedial location and the port location if there are more than one transshipment terminals. The distance between terminals is considered proportionate to the distance shown in the figure. Size of test instances generated is summarized in Table 17. The performance settings used for different test instances for the multi-start and single start are tabulated in Table 18 and Table 19.

Table 17: Parameters for test instance used to compare the performance of the solution approach

Set	*Instance parameter*			
	cn	*bn*	*tn*	*Hn*
I1	5	3	4	0
I2	5	3	4	1
I3	10	3	5	0
I4	10	3	5	2
I5	15	3	5	0
I6	15	3	5	1
I7	20	5	6	0
I8	20	5	6	1
I9	30	4	6	0
I10	30	4	6	1

Table 18: Parameter settings used for different instance – multi-start

Parameter	Instance									
	1	2	3	4	5	6	7	8	9	10
maxAlgorithmIterationCount	5	5	10	10	10	10	10	10	10	10
maxAlgorithmRunTimeSec	1000	1000	1000	1000	1000	1200	1000	1600	1000	3600
NrOfRequestConsideredForCandidateList	2	2	4	4	4	4	5	4	7	5
MaxBestSolutionForConstruction	5	5	5	5	5	5	5	5	15	5
MaxCandidateListForConstruction	5	5	5	5	5	5	5	5	15	5
minDegreeOfDestruction	1	1	1	1	1	1	1	1	1	1
maxDegreeOfDestruction	2	2	4	4	4	4	4	5	6	4
degreeOfDestructionIncrement	1	1	1	1	1	1	1	1	2	1
incrementDegreeOfDestructionForNoImprovement	5	5	5	5	5	5	8	5	8	3
sigmaOne	3	3	3	3	3	3	3	3	3	3
sigmaTwo	1	1	1	1	1	1	1	1	1	1
rouletteWheelParameter	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
betaOne	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
betaTwo	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
StartTemp	50	50	50	100	100	100	100	50	50	50
markovChainLength	15	10	30	20	30	20	50	20	30	15
numberOfNoImprovementForStopSA	10	15	20	10	20	10	30	10	20	10
decrementFactor	0.5	0.5	0.75	0.7	0.75	0.75	0.75	0.55	0.75	0.55
endTemp	1	5	1	5	1	5	1	10	1	10

Table 19: Parameter setting used for different instance - Single start

Parameter	Instance								
	1	2	3	4	5	6	7	8	9
maxAlgorithmIterationCount	1	1	1	1	1	1	1	1	1
maxAlgorithmRunTimeSec	1000	1000	1000	1000	1000	1000	1000	1000	1000
NrOfRequestConsideredForCandidateList	2	2	4	4	5	5	7	7	10
MaxBestSolutionForConstruction	5	5	5	5	7	7	10	10	10
MaxCandidateListForConstruction	5	5	5	5	7	7	10	10	10
minDegreeOfDestruction	1	1	1	1	1	1	1	1	1
maxDegreeOfDestruction	2	2	4	4	5	5	5	5	7
degreeOfDestructionIncrement	1	1	1	1	1	1	1	1	1
incrementDegreeOfDestructionForNoImprovement	5	5	5	5	7	7	7	7	7
sigmaOne	3	3	3	3	3	3	3	3	3
sigmaTwo	1	1	1	1	1	1	1	1	1
rouletteWheelParameter	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
betaOne	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
betaTwo	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
StartTemp	50	50	50	50	100	50	100	100	100
markovChainLength	30	30	50	50	70	50	50	50	100
numberOfNoImprovementForStopSA	20	20	25	25	30	30	30	30	50
decrementFactor	0.75	0.75	0.75	0.75	0.9	0.8	0.95	0.8	0.95
endTemp	0.1	0.1	0.1	1	0.1	1	0.1	1	0.1

Appendix F

This appendix has been removed from the public version.