Tramp Ship Routing with Transshipment Possibilities: Solution Approaches and Parameter Estimation Methods

Bachelor Thesis

Double Degree Econometrics & Economics

Operations Research and Quantitative Logistics [1]

Author: **Peter de Voogd**

Student number: **409118**

Supervisor: **Nemanja Milovanović**

Co-reader: **Wilco van den Heuvel**

July 8, 2018

[1]Erasmus School of Economics,
Erasmus University Rotterdam

Abstract

This paper presents solution methods for tramp ship routing and scheduling problems with transshipment possibilities. It presents formulations which are able to find optimal solutions for relatively small instances and heuristics which can determine low-cost solutions when the problem instances become larger. Furthermore, parameter estimation methods are presented which can be used to determine efficient parameter settings for the different heuristics. Finally, nine different sets of parameters are provided which can be adopted for different types of problems, dependent on their characteristics.

For all methods the benchmark instances as provided by Hemmati et al. (2014) are used to estimate performance. Out of the 240 problem instances provided there, new optimal values were found or confirmed for 15 instances and improved solutions were found for 59 instances. Finally, by including transshipment possibilities in the model, a cost decrease was found for 144 of the problems. In some cases, this corresponded to cost savings of well over ten percent.

Contents

Chapter 1

Introduction

1.1 Background

In 2016, over ten billion tons of goods have been shipped overseas. This corresponds to about 1.4 tons of goods for every single individual. The value of these goods amounted to over seventy percent of the total global trade. Altogether, over 700 million containers have been handled on ports worldwide for the transportation of these goods (UNCTAD, 2017).

In general, there are two ways of assigning these containers to the ships which can transport them. In the first case, the ships follow given schedules. That means that for each cargo, an optimal route should be picked, given the available schedules and ship capacities. The other option is to allow more flexible ship routing, where the ships follow the available demand. In general, the first option is referred to as liner shipping, while the second is either referred to as industrial shipping or tramp shipping (Lawrence, 1972). In industrial shipping, the cargo owner also owns the ships, whereas with tramp shipping, the ship owners generally set up contracts with the cargo owners. These contracts specify which cargoes should be shipped and include some time windows indicating when the pickup and delivery of the cargoes should take place.

This paper focuses on tramp shipping, investigating means of routing the ships to pick up and deliver a given set of cargoes. The relatively small problem instances are solved to optimality using mathematical formulations. For the larger problem instances, solving to optimality takes too long, so an adaptive large neighbourhood search (ALNS) heuristic is implemented. In order to speed up this heuristic, parameter estimation methods are described.

1.2 Problem Description

In order to transport a cargo to its destination, two options are available. The cargo can either be transported by one of the available vessels or by using the available spot charters. It is assumed that the ship owner owns a heterogeneous fleet of vessels which may have different capacities, sailing speed, transportation costs, and different starting positions and starting times. Furthermore, some cargoes and vessels might not be compatible.

If a cargo is assigned to one of the available vessels, it should be picked up and delivered within certain time intervals, while not exceeding ship capacity. If none of the vessels is used to transport this cargo, it should be transported by a spot charter at a given, fixed cost.

The cargoes can be brought to their destinations by a single vessel, but in some cases it is economically more efficient to use multiple vessels in handling a single cargo load. If that happens, the containers will have to be shipped from one vessel to another at some port. This is known as transshipment. If such a transshipping operation is performed, some costs are incurred at the relevant port and some time is spent during this operation.

The problem can be summarized using the following research question:

> *How can low-cost solutions be derived for tramp ship routing and scheduling problems with transshipment possibilities?*

Regarding this research question, we assume that it will be possible to develop mathematical formulations which can be used to find optimal solutions for problems of relatively small size, but which will not be able to come up with optimal solutions, or even low-cost solutions for larger problems. For these larger problem instances, we assume that heuristic approaches will manage to find solutions which are better than those found by using the mathematical formulation.

In summary, this paper aims to find efficient formulations and heuristics for tramp ship routing with and without transshipment.

1.3 Literature review

1.3.1 Tramp Shipping Research

Inspired by solution approaches which turned out very successful in other industries, some of the first solution methods used in the tramp shipping industry have been introduced by Appelgren, solving the ship routing problem with time windows using the Dantzig-Wolfe decomposition method (Appelgren, 1969; Dantzig, 1963) and using branch-and-bound techniques (Appelgren, 1971; Land & Doig, 1960). In his first paper, he determines a solution to the problem, which might not satisfy integrality constraints and in his second paper he describes how this solution can be converted to a feasible integer solution. For extensive discussion of subsequent literature, we refer the interested reader to the review papers brought out at a ten year interval (Ronen, 1983, 1993; Christiansen et al., 2004; Pantuso et al., 2014). We just note that most of the subsequent literature investigates how to improve the current solutions with more advanced solution methods, or how the current problems can be extended to include more real-life characteristics. The same holds for this paper, attempting to improve the solution methods that are known to date and including transshipment possibilities, but using some simplifications to keep the model simple, thereby leaving room for further extensions.

1.3.2 Extensions to Suit Real Life Characteristics

As mentioned, some simplifications are used throughout this paper. First of all, vessel speed is assumed to be constant. In reality, this need not be, as vessel speed is subject to a natural trade-off between transportation time and fuel cost. This natural trade-off is the ground for a field of research regarding speed optimization (Norstad et al., 2011). Here, time windows may not be fixed, but might be subject to more flexible bounds (Yu et al., 2017). Finally, fleet size is taken constant. In reality, this is a strategical decision and only really constant in the short-run. In the long-run, decisions need to be made about whether to add additional vessels to the available fleet (Xinlian et al., 2000).

As an alternative to transshipment operations, one might like to split up the cargo into smaller sizes, and transport all these 'subcargoes' using multiple vessels (Korsvik et al., 2011).

1.3.3 Benchmark Suites

Most of the literature in operations research is focused on finding new solution methods or improving existing solution methods for the problem under consideration. Without benchmark problem instances, the academic relevance of these papers would be smaller, as someone interested in the best solution approach for his problems would have to compare all possible methods before knowing which would work well. This is not necessary, as for most operations research problems and other mixed integer programming problems benchmark problem instances are given, which can be used to compare the performance of different solution methods.

Extensive libraries with benchmark instances can be found in Koch et al. (2011) for general mixed integer problems and Beasley (1990) for operational research benchmark suites.

Although benchmark suites are widely provided for most of the programming problems in operations research, none of these included problem instances which could directly be used for investigating solution methods regarding tramp ship routing and scheduling. Therefore, Hemmati et al. (2014) provided a benchmark suite to fill this gap and to inspire future research in this area of literature. The goal is to allow for a standardized comparison of solution methods and to inspire future research in this area of literature.

1.3.4 Developing the ALNS Heuristic

Oftentimes, a single heuristic is used throughout the entire solution method, or multiple heuristics are used sequentially in a multiple-stage approach. However, Ropke & Pisinger (2006) introduced the so-called Adaptive Large Neighbourhood Search (ALNS), where multiple heuristics are used simultaneously and weighted according to their performance in previous iterations.

In every iteration of the ALNS heuristic, some cargoes are removed from the current solution, and reinserted at certain positions. This is known as a k-interchange or k-opt method (Croes, 1958; Lin, 1965) and is known to have strong performance in many fields (Lin & Kernighan, 1973), including routing problems with time windows (Potvin & Rousseau, 1995).

The most basic k-interchange methods rely on local search. Generally, this procedure takes many small steps like interchanging two cargoes or moving a cargo between two vessels (Bräysy et al., 2004) and (Brønmo et al., 2007). These methods are able to visit many possible solutions in a short amount of time, but they might get stuck in a local minimum. To overcome this, some more advanced methods can be used, such as tabu search (Garcia et al., 1994) or branch and bound (Shaw, 1997), by temporarily visiting some infeasible solutions (Cordeau et al., 2001), or by taking larger steps. The last option is known as Large Neighbourhood Search (Shaw, 1997) and is the basis of the ALNS heuristic.

1.4 Relevance

The main contribution of this paper is that it provides methodology which can be used to derive low-cost solutions for industrial and tramp ship routing problems with time windows while allowing for transshipment. Generally, methods developed for solving routing problems without transshipment fall short for solving transshipment problems, as they do not account for dependencies between vessel schedules (Drexl, 2013).

In 2016, over 25 percent of liner shipping throughput consisted of transshipment containers (UNCTAD, 2017). However, for industrial and tramp shipping, transshipment is much less prevalent. Accordingly, research on transshipment operations in the liner shipping industry has come up with efficient options for including transshipment in the models (Wang & Meng, 2012), but for industrial and tramp shipping this is not the case. This paper aims to fill this gap in research by providing a mathematical formulation which allows for transshipment of containers and extending the ALNS heuristic to allow for transshipment. The ALNS heuristic enables determining low-cost solutions for problem instances where optimal solutions cannot be found within reasonable time.

Finally, a local search heuristic is presented to estimate the necessary parameters. Knowledge of what parameter values work well in which situations will help derive low-cost solutions very quickly. Nine different sets of parameters are provided which can be used depending on the characteristics of the problem at hand.

1.5 Outline

Next, In Chapter 2 the problem instances that were used in this paper are described. After that, Chapters 3 and 4 explain the mathematical formulation and ALNS heuristic used to solve the problem without transshipment. Then, in Chapter 5 and 6 these are extended to allow for transshipment. Afterwards, in Chapter 7 parameter estimation methods are discussed. Next, in Chapter 8 the results are presented. Finally, Chapter 9 concludes this paper.

Chapter 2

Data

In their 2014 paper, Hemmati et al. present 240 problem instances by means of which the implementation and performance of solution methods can be tested in terms of objective value and running time. These instances are used to represent multiple types of problems.

First of all, a distinction is made between deep sea and short sea shipping problems. In the first case, many of the cargoes are transported over one of the big oceans, implying that average sailing times are relatively long. In short sea shipping problems, the operations are only regional and correspondingly, average sailing times are short.

Secondly, the data distinguishes between full load problems, for which only a single cargo fits in the vessels, and mixed load problems, where the ships can hold multiple (smaller) cargoes.

Together, these two distinctions imply four different types of problems. By using different numbers of cargoes and ships, sixty problem instances are provided for each of these four types. So, altogether, 240 benchmark problem instances are given[1]. These include data on the vessels, cargoes, and ports based on real life values (Fagerholt et al., 2000).

Firstly, for each vessel, its starting port and corresponding starting time are given, as well as its capacity and a list of cargoes which it can transport. Secondly, for each cargo, the origin and destination port, size, relevant spot charter cost and time windows are given. Furthermore, for each possible trip between two ports, the transportation time and cost are given for each vessel[2]. Finally, for each port, the cost of using this port and the time induced in doing so are given per vessel.

The number of cargoes which need to be transported range between 7 and 130, the number of available vessels range between 3 and 50, and the number of ports are 39 and 86 for short sea and deep sea shipping problems, respectively (although not all ports are relevant for the smaller problem instances).

[1] The problem instances can be found at `http://home.himolde.no/~hvattum/benchmarks/`.

[2] It is important to note that the transportation time and cost in the datasets do not satisfy the triangle inequality. This implies that when moving between two ports, it may be cheaper to stop at an intermediate port rather than moving straight from one port to the other. This option is only exploited by the formulation in Chapter 5 and therefore, solutions obtained there could be cheaper than what other methods might find.

Chapter 3

Tramp ship routing

This chapter presents a mathematical formulation which can be used to solve the basic problem to optimality. The formulation is taken from Hemmati et al. (2014).

3.1 Problem Description

For now, transshipment options are disregarded. The goal is to minimize the sum of transportation costs, port costs and spot charter costs, while satisfying vessel flow constraints, weight constraints, time constraints and precedence constraints.

3.1.1 Notation

Let the set of vessels be denoted as V, the set of loading and unloading nodes be denoted as N^p and the set of nodes which vessel v can visit be denoted as N_v. Let N_v^p denote the intersection of N^p and N_v and let A_v denote the set of arcs which vessel v can traverse. Let n denote the number of cargoes. Next, let K_v be the capacity of vessel v, and $o(v)$ and $d(v)$ denote the origin node and artificial destination node of vessel v. Furthermore, let the cost and travel time of sailing from node i to j with vessel v be denoted as C_{ijv} and T_{ijv}, respectively. Let the time window of node i be bound by $\underline{T}_i$ and $\bar{T}_i$ and let the size of the relevant cargo for node i be denoted by Q_i. Then, let x_{ijv} be a binary variable equal to one if vessel v sails from node i to node j. Also, let y_c be a binary variable equal to one if cargo c is transported by a spot charter and let C_i^S be the associated cost. Finally, let l_{iv} and t_{iv} be the weight on board of vessel v after node i and the time at which it arrives there, respectively.

3.1.2 Formulation

Using the notation as explained before, the problem can be formulated as follows:

$$\min \sum_{v \in V} \sum_{(i,j) \in A_v} C_{ijv} x_{ijv} + \sum_{i \in N^p} C_i^S y_i \tag{3.1}$$

subject to

$$\sum_{v \in V} \sum_{j \in N_v} x_{ijv} + y_i = 1, \qquad i \in N^P, \tag{3.2}$$

$$\sum_{j \in N_v} x_{o(v)jv} = 1, \qquad v \in V, \tag{3.3}$$

$$\sum_{j \in N_v} x_{ijv} = \sum_{j \in N_v} x_{jiv}, \qquad v \in V,\ i \in N_v \backslash \{o(v), d(v)\}, \tag{3.4}$$

$$\sum_{j \in N_v} x_{jd(v)v} = 1, \qquad v \in V, \tag{3.5}$$

$$l_{iv} + Q_j - l_{jv} \leq K_v(1 - x_{ijv}), \qquad v \in V, j \in N_v^P, (i,j) \in A_v, \tag{3.6}$$

$$l_{iv} - Q_j - l_{(n+j)v} \leq K_v(1 - x_{i(n+j)v}), \quad v \in V, j \in N_v^P, (i, n+j) \in A_v, \tag{3.7}$$

$$0 \leq l_{iv} \leq K_v, \qquad v \in V, i \in N_v^P, \tag{3.8}$$

$$t_{iv} + x_{ijv}T_{ijv} - t_{jv} \leq \bar{T}_i(1 - x_{ijv}), \qquad v \in V, (i,j) \in A_v, \tag{3.9}$$

$$\sum_{j \in N_v} x_{ijv} = \sum_{j \in N_v} x_{(n+i)jv}, \qquad v \in V,\ i \in N_v^p, \tag{3.10}$$

$$t_{iv} + T_{i(n+i)v} - t_{(n+i)v} \leq 0, \qquad v \in V,\ i \in N_v^p, \tag{3.11}$$

$$\underline{T}_i \leq t_{iv} \leq \bar{T}_i, \qquad v \in V, i \in N_v, \tag{3.12}$$

$$x_{ijv} \in \{0,1\}, \qquad v \in V, (i,j) \in A_v, \tag{3.13}$$

$$y_i \in \{0,1\}, \qquad i \in N^C, \tag{3.14}$$

Constraints (3.2) ensure that every cargo is either transported by one of the available vessels or the spot-charter. Constraints (3.3) – (3.5) describe the vessel routing. Constraints (3.6) – (3.8) ensure that cargo weights are updated correctly and do not exceed vessel capacity. Constraints (3.9), (3.11), and (3.12) update the time variables and keep them within the required bounds. Next, Constraints (3.10) look after the precedence requirements. Finally, Constraints (3.13) and (3.14) set the binary requirements.

Chapter 4

ALNS Heuristic

To find low-cost solutions for problem instances which are too large to solve to optimality within reasonable time the ALNS-heuristic as developed by Ropke & Pisinger (2006) is used.

4.1 Procedures

First, an initial solution is generated. This is done by assigning all cargoes to the available spot charters. Next, the following steps are repeated until a certain stop-criterion is met. They will be explained in more detail throughout Sections 4.2 – 4.5.

First, a removal and insertion heuristic are selected. Then, using these heuristics, a certain number of cargoes are removed and then reinserted in the current solution. After reinsertion, if the acceptance-criterion is met, the current solution is updated and if the objective value corresponding to the new solution is better than the previous best, the best solution is updated as well.

Finally, after a predefined running time, the best solution found is returned.

4.2 Removal of Cargoes

In every iteration, q cargoes are removed, where q is a random number between four and $\min(100, \xi n)$, with n being the number of cargoes and ξ a parameter between zero and one. For the removal of cargoes, the Shaw removal heuristic, random removal, and worst removal are used.

4.2.1 Shaw Removal Heuristic

The Shaw removal heuristic depends on the similarity between cargoes (Shaw, 1997). To find q cargoes for removal, a random cargo is selected, together with $q - 1$ similar cargoes. The intuition behind this is that it might be easier to interchange the positions of similar cargoes between solutions, rather than very different cargoes. In the latter case, only the original positions and expensive positions might be possible.

Similarity is determined based on several characteristics. First of all, the minimal travel cost between the origin and destination of cargoes and the difference between arrival times are considered. This is done because cargoes which are close to one another (geographically) and are being picked up and delivered at similar times might be more easily interchanged. Secondly, the differences in size and the sets of vessels which can transport the cargoes are taken into account. This is done because interchanging the cargoes should yield a feasible solution and similarities in these regards increase the probability that this is the case.

Altogether, the similarity between two cargoes is calculated with the following equation, where a lower value of $R(i,j)$ implies more similarity between cargoes i and j.

$$\begin{aligned} R(i,j) =& \varphi(C_{o(i),o(j)} + C_{d(i),d(j)}) + \chi(|T_{o(i)} - T_{o(j)}| + |T_{d(i)} - T_{d(j)}|) \\ &+ \psi|Q_i - Q_j| + \omega\left(1 - \frac{|V_i \cap |V_j|}{\min\{|V_i|, |V_j|\}}\right) \end{aligned} \tag{4.1}$$

with $C_{a,b}$ the traveling cost between a and b, $o(c)$ and $d(c)$ the origin and destination port of cargo c, T_a the time at which node a is visited, Q_c the size of cargo c and V_c the set of vessels which can transport cargo c.

In order to make the absolute value of the parameters more meaningful, all terms are normalized by scaling the relevant variables such that they take values between zero and one. So, the value of $R(i,j)$ ranges between 0 and $2\varphi + 2\chi + \psi + \omega$. Note that the values of T_a are not given for cargoes which are transported using the available spot charters, so these are set to be the average of the lower and upper bounds to give an as accurate as possible representation of what they might have been.

Chapter 7 explains how the values of the weighting parameters can be determined.

4.2.2 Random Removal

Random removal is much simpler than Shaw removal, selecting q cargoes at random. The advantage of this method is that it runs faster than the Shaw removal method and does not require any parameter estimation. However, it relies on luck to find good solutions and therefore might not be as good at converging to an optimum. For this reason, random removal is likely to perform better at early stages of the ALNS heuristic (in terms of time) but might be outperformed later on.

4.2.3 Worst Removal

The last removal heuristic, worst removal, removes q cargoes, which are set at high cost positions. To measure this, the cost of the current solution, with and without this cargo is taken and the difference is calculated. Then, the cargo for which this difference is largest, is removed, and the process is repeated until q cargoes are removed from the current solution.

Intuitively, the difference between the Shaw removal method and worst removal, is that Shaw removal focuses on cargoes which can easily be interchanged, while worst removal focuses on cargoes which might be put at the wrong position. Furthermore, because worst removal has a tendency to remove the spot chartered cargoes first, it might outperform Shaw removal (in terms of objective value) at early stages of the ALNS heuristic.

4.3 Reinsertion of cargoes

For the insertion of cargoes, a basic greedy heuristic and a regret-k heuristic are used.

4.3.1 Greedy Reinsertion Heuristic

The basic greedy insertion heuristic inserts one cargo at a time, such that the lowest cost cargo is inserted first every time. That is, for every cargo, the insertion costs at all possible positions are calculated and the minimum of these values is computed. Then, the cargo for which this minimum is lowest, is inserted first. This implies that at every step, we choose the least costly option of all possible insertions. However, the risk of this heuristic is that for the last cargoes to be inserted, only very costly options remain. This shortcoming is the basic idea used in the next heuristic.

4.3.2 Regret heuristics

Regret heuristics try to avoid being left with very costly cargoes after inserting some cheaper ones. Intuitively, we try to select the cargo which we might regret if it wouldn't be selected now. For this, the regret-k value is calculated for all cargoes which are not yet being served by one of the vessels. This regret-k value is defined as the sum of differences between the cost of inserting the cargo at its best position and inserting it at the second best, third best, ..., and k-th best position (Tillman & Cain, 1972). Then, the cargo with the highest regret value is inserted at the cheapest position. If two cargoes have the same regret value, the one with the lowest insertion cost is chosen.

If a cargo cannot be inserted at k different positions, its regret value is set at infinity. When multiple cargoes have an infinite regret value, the one with the least feasible insertion points is chosen. When there are multiple cargoes which have the same number of insertion points, the one with the lowest insertion cost is chosen.

4.4 Heuristic Selection

So far, multiple removal and reinsertion heuristics have been presented. To explore a region which is as large as possible, all possible combinations should be allowed in every iteration. In order to do so, let a weight w_i be given for every heuristic and let the probability of selecting heuristic i be equal to w_i. To be as flexible as possible, the removal and insertion heuristics are selected and weighted separately.

In order to determine the weights of the heuristics, they will get a score π_i based on their previous performance. This score will increase most when they manage to find a new best solution. Secondly, their scores will receive a medium increase when a new solution is found which improves upon the current solution, but not on the best known solution. Finally, the relevant scores will increase a small bit when a new solution is found and accepted which does not increase the current solution. This is done to reward heuristics which manage to explore new regions of the solution space. From experiments we found that scores of 10, 4 and 1, respectively, work well and therefore these scores have been used throughout this paper.

To update the heuristic weights, dependent on their scores, the entire search is divided into separate segments. Between the different segments, the weights of the different heuristics are adjusted and the scores are reset. First, the scores are normalized by dividing them by the number of times they have been used in the previous segment, θ_{is}. Then, the weights in the next segment, $w_{i,s+1}$ are determined by balancing the previous weights and the relevant scores according to a constant parameter r:

$$w_{i,s+1} = (1-r)w_{is} + r\frac{\pi_i}{\theta_{is}} \tag{4.2}$$

The parameter r determines how important the previous weights are. If r is set at zero, the initial weights are kept throughout the entire search and if r is set at one, the new weights are determined entirely through the scores in the previous segment. Finally, the weights are scaled such that they sum to one.

4.5 Acceptance Criterion

To determine whether a new found solution should be accepted or rejected, an acceptance criterion based on simulated annealing (Kirkpatrick et al., 1983) is used.

First of all, if the new solution yields a cost reduction compared to the current solution, it is always accepted.

Next, if the new solution is worse than the current solution, it might still be accepted. The probability of acceptance equals $e^{(f_{new}-f_{best})/(T_s f_{best})}$. Here, $f_{solution}$ is the objective value corresponding to the new solution or best solution found so far. Furthermore, T_s is the current cooling temperature and T is the starting temperature. In every segment s, T_s is calculated as $T_s = cT$, where c equals the total number of segments divided by the current segment number. For example, when halfway through the heuristic, T_s would equal $2T$. In our research, ten segments are used. In summary, the probability of acceptance, depends on how much more expensive the current solution is than the current best and how far we are along the entire search.

A problem with this approach is that after getting down to a reasonably good objective value, the heuristic might struggle finding improvements, and therefore start accepting worse solutions faster than finding improvements. In this case, the heuristic might keep jumping upwards, just after we arrived at a cost similar to the best one found so far and not be able to improve any further.

To circumvent this problem, one could set T at a very large value, in order to avoid jumping up too often. However, this might make the probability of acceptance so small that hardly any worse solutions will be accepted, thereby taking away the advantage of simulated annealing. Therefore, a different approach is taken here.

Whenever a new global best solution is found, the possibility of accepting a worse solution will be taken away for n iterations. By doing so, the heuristic will get to search through the neighbourhood of this solution before possibly jumping back up to a worse solution. Secondly, the number of times a worse solution can be accepted, before finding a better solution will be fixed by a given parameter d. This helps to avoid situations where many steps to worse solutions are taken in quick succession.

Chapter 5

Tramp ship routing with transshipment

5.1 Problem description

Now, the problem is extended by allowing a single cargo to be shipped by multiple vessels before reaching its destination. In order to do so, the problem instances are extended to incorporate the time it takes to transfer a certain cargo from one ship to another and the costs obtained from doing so. Intuitively, this will have different effect depending on the number of cargoes, vessels and ports, the size of cargoes and vessels and the time and cost associated with transferring cargoes. Therefore, some additional problem instances are generated which differ in these regards.

Once again, a mathematical formulation and heuristic approach are described and implemented for small and large problem instances, respectively.

5.2 Notation

In order to allow multiple vessels to be used for the transportation of a single cargo, some new variables and parameters are required, and some changes need to be made to the formulation from Chapter 3 and the heuristic from Chapter 4.

First of all, let K denote the set of cargoes and let $o(k)$ and $d(k)$ denote the origin and destination of cargo k. Let V_k denote the set of vessels which can transport cargo k and let K_v be the set of cargoes which vessel v can transport. Also, let N_{uv} denote the intersection of N_u and N_v. Then, let z_{ijkv} be equal to one if vessel v transports cargo k over arc (i, j), and zero otherwise and let w_{ikuv} be one if cargo k is switched from vessel u to vessel v at node i and zero otherwise. Next, let t^a_{iv} and t^d_{iv} be the arrival and departure time of vessel v at node i, respectively and let T^l_{ikv} and T^u_{ikv} be the loading and unloading time of cargo k by vessel v at node i, respectively. Note that T^l_{ikv} and T^u_{ikv} were included in T_{ijv} in Chapter 3, but they need to be treated separately here because not all loading and unloading happens at the origin and destination nodes anymore. Finally, let C_{ikuv} be the cost associated with transshipping cargo k at node i using vessels u and v, where u is the vessel transporting k to i and v is the vessel transporting k from i. Here, these costs are taken as the sum of the two port costs associated with the two vessels and port i. That is, the costs correspond to the costs incurred by the two separate vessels when performing an operation at port i and it is assumed that these costs cover storage costs if the pickup vessels arrives later than the time at which the delivery vessel finishes unloading. Note that the port costs were included in C_{ijv} in Chapter 3, but are actually provided separately from C_{ijv} in the datasets provided by Hemmati et al. (2014).

5.3 Formulation

Using this notation, the complete mathematical formulation becomes:

$$\min \sum_{v \in V} \sum_{(i,j) \in A_v} C_{ijv} x_{ijv} + \sum_{i \in N^p} C_i y_i + \sum_{k \in K} \sum_{u \in V_k} \sum_{v \in V_k \setminus \{u\}} \sum_{i \in N_{uv}} C_{ikuv} w_{ikuv} \tag{5.1}$$

subject to

$$\sum_{j \in N_v} x_{o(v)jv} = 1, \quad v \in V, \tag{5.2}$$

$$\sum_{j \in N_v} x_{ijv} - \sum_{j \in N_v} x_{jiv} = 0, \quad v \in V,\ i \in N_v \backslash \{o(v), d(v))\}, \tag{5.3}$$

$$\sum_{j \in N_v} x_{jd(v)v} = 1, \quad v \in V, \tag{5.4}$$

$$\sum_{v \in V_k} \sum_{j \in N_v} z_{o(k)jkv} + y_k = 1, \quad k \in K, \tag{5.5}$$

$$\sum_{v \in V_k} \sum_{j \in N_v} z_{ijkv} - \sum_{v \in V_k} \sum_{j \in N_v} z_{jikv} = 0, \quad k \in K,\ i \in N \backslash \{o(k), d(k))\}, \tag{5.6}$$

$$\sum_{v \in V_k} \sum_{j \in N_v} z_{jd(k)kv} + y_k = 1, \quad k \in K, \tag{5.7}$$

$$\sum_{k \in K_v} z_{ijkv} Q_k - x_{ijv} K_v \leq 0, \quad v \in V, (i,j) \in A_v, \tag{5.8}$$

$$\sum_{j \in N_u} z_{jiku} + \sum_{j \in N_v} z_{ijkv} - w_{ikuv} \leq 1, \quad k \in K, u \in V_k, v \in V_k \backslash \{u\}, i \in N_{uv}, \tag{5.9}$$

$$t^d_{iv} + x_{ijv} T_{ijv} - t^a_{jv} - \bar{T}_i (1 - x_{ijv}) \leq 0, \quad v \in V, (i,j) \in A_v, \tag{5.10}$$

$$t^d_{iv} + T_{i(n+i)v} - t^a_{(n+i)v} \leq 0, \quad v \in V,\ i \in N^p_v, \tag{5.11}$$

$$\underline{T}_i < t^a_{iv} \leq \bar{T}_i, \quad v \in V, i \in N_v, \tag{5.12}$$

$$t^a_{iu} + \sum_{v \in V_k \backslash \{u\}} w_{ikuv} T^u_{iku} \leq t^d_{iu}, \quad u \in V, i \in N_u, k \in K_u, \tag{5.13}$$

$$t^a_{iu} + w_{ikuv}(T^u_{iku} + T^l_{ikv}) \leq t^d_{iv}, \quad k \in K, u \in V_k, v \in V_k \backslash \{u\}, i \in N_{uv}, \tag{5.14}$$

$$t^a_{iv} + \sum_{u \in V_k \backslash \{v\}} w_{ikuv} T^l_{ikv} \leq t^d_{iv}, \quad v \in V, i \in N_v, k \in K_v, \tag{5.15}$$

$$w_{ikuv} \in \{0,1\}, \quad k \in K, u \in V_k, v \in V_k \backslash \{u\}, i \in N_{uv}, \tag{5.16}$$

$$x_{ijv} \in \{0,1\}, \quad v \in V, (i,j) \in A_v, \tag{5.17}$$

$$y_k \in \{0,1\}, \quad k \in K, \tag{5.18}$$

$$z_{ijkv} \in \{0,1\}, \quad v \in V, k \in K_v, (i,j) \in A_v, \tag{5.19}$$

Constraints (5.5) – (5.7) ensure that every cargo is brought from origin to destination, either by the available vessels or by the spot charter, while satisfying the flow conditions. Constraints (5.8) make sure that the load on board of vessel v does not exceed its capacity. Furthermore, they ensure that if a vessel does not traverse a certain arc, no cargoes can be shipped by that means. Constraints (5.9) state that a cargo can only be shifted from one vessel to another if both are present at the respective port. Constraints (5.10) update the time between departure and arrival. Constraints

(5.11) enforce the precedence requirements. Time windows are imposed through Constraints (5.12). Constraints (5.13) – (5.15) impose that arrival times and departure times account for the time spent unloading and loading the cargoes. Finally, binary requirements are set by Constraints (5.16) – (5.19).

5.3.1 Solving Problems Without Transshipment

Note that by fixing w_{ikuv} to be zero, this formulation can also be used to solve problems without transshipment possibilities. This is not very efficient in terms of running time, though. However, by including the following additional contraints, the efficiency is greatly increased.

$$x_{o(k)jv} = z_{o(k)jkv}, \quad v \in V, k \in K_v, j \in N_v \tag{5.20}$$

$$x_{id(k)v} = z_{id(k)kv}, \quad v \in V, k \in K_v, i \in N_v \tag{5.21}$$

Including these constraints will actually allow full cargo load problem instances to be solved faster than by the original problem formulation from Chapter 3. This approach requires more computer memory though, so problem instances with 100 or more cargoes can no longer improve upon the trivial solution of transporting every cargo with the spot charters without running out of memory anymore[1]. Problems of such size are better left to heuristic approaches regardless of what formulation is being used, so we do not consider this to be a very serious drawback.

Furthermore, this adapted formulation will be used to obtain starting solutions. That is, first w_{ikuv} will be fixed to zero to solve the problem without transshipment. Then, w_{ikuv} will become a binary variable again to solve the problem with transshipment and the final solution obtained from when w_{ikuv} was zero will be used as a starting solution for this problem. Using this approach will save considerable amounts of running time.

A final advantage of this adapted formulation arises in problem instances for which the triangle inequality does not hold with regards to transportation time and cost[2]. In this case, it may be advantageous to omit equations 5.20 and 5.21 to remove any restrictions on x_{ijv}. In Chapter 3, x_{ijv} was restricted by forcing a delivery or pickup when moving to a port. Similarly, in Chapter 4, the solutions do not allow for intermediate ports to be visited. This also holds for the heuristic in Chapter 6. We have not investigated how severe this effect is, but like to note that a few percent decrease in objective value is not rare. To solve these problems, we suggest that any trips for which the triangle inequality does not hold would have their transportation time and cost replaced by the relevant (cheaper and faster) alternatives. An analysis of the correct approach in situations where either the transportation time or the transportation cost (but not both) is lower when moving through an intermediate port is left to future research[3].

[1] Using a computer with 16 GB RAM.
[2] As noted before, this is the case for some of the ports which are provided by Hemmati et al.
[3] This may for example happen when a shorter route might be available, but would require a fee.

Chapter 6

ALNS heuristic with transshipping

To include transshipments during the ALNS heuristic, an insertion method will be introduced, which will insert a given cargo by using a transshipment operation at the cheapest position and will pass the cargo on to one of the other insertion methods if no feasible positions exist.

6.1 Transshipment Insertion

First, two vessels will be selected which can pick up the cargo and deliver it, such that the pickup time is before the delivery time minus the shortest route from the pickup point to the delivery point. We will refer to these as the pickup vessel and the delivery vessel. Next, a feasible position will be selected where the pickup and delivery of the cargo will take place by the relevant vessel. Then, for both vessels, a position on their current route will be selected where the transshipment should take place and could take place when taking weight limits into account.

Given these vessels and positions, the costs of all ports at which the transshipment operation could take place given time window constraints are considered, and the port with the minimum cost is chosen. Note that these costs include the cost of picking up and delivering the cargo at its origin and destination port, respectively.

This process is repeated for all vessel combinations and positions on their routes. Finally, the vessel combination with the minimum cost is chosen and the transshipment operation is inserted at the positions and port corresponding to this lowest cost option. If no feasible insertion points exist, the cargo will be inserted by one of the other insertion methods.

6.2 Further Adjustments

Note that by including transshipment operations in the route of a given vessel, also changes the feasibility check of inserting normal cargoes. Therefore, it is recommended to keep track of which vessels have a transshipment operation in their schedule and which vessels are used in this operation, as transshipping introduces an interdependency between vessels regarding feasibility.

Because the transshipment insertion can only find transshipment possibilities for a single cargo at a time, it is quite expensive in terms of running time. Furthermore, all feasibility checks that happen after a transshipment operation has been inserted will also become more cumbersome. Therefore, this insertion heuristic will be selected on a different basis than the other heuristics. In every iteration, at most a single, random cargo will be inserted using transshipment insertion and this will always be done after all removed cargoes are served again. This is done because it will avoid an early inefficient transshipment insertion which would increase the cost of the subsequent insertions.

In every iteration, the probability of attempting a transshipment operation is set at $1-c/10$ with c the total number of segments divided by the current segment number. This is done to make the insertion of early transshipment operations less likely, as they take longer to perform and increase the running time of the subsequent iterations when successful.

Chapter 7

Parameter estimation

7.1 Shaw Removal Parameters

To determine the weights that should be used for the Shaw removal heuristic, note that the absolute values of the parameters do not matter, just their size compared to each other. Therefore, the values of these parameters can be standardized to sum up to one. Furthermore, note that the weights of individual removal heuristics also sum up to one. Therefore, the initial weights to be used in the Shaw removal heuristic can be determined by splitting it up into four separate removal heuristics, one for each component weighted by one of the parameters and then running the entire heuristic using only these removal heuristics. The weights of the different components can then be used as parameter values.

However, there might still be room for improvement. Because the weight of the different components only indicates how often a certain heuristic should be used, this yields limited information on how large the relative values of these components should be within the original heuristic. For example, it might be that a certain component is very important, so it receives a high weight. However, when setting the relating parameter to this weight value its effect might still be limited because the absolute value is not large enough. Therefore, the previous procedure will be repeated, but with the complete Shaw removal heuristic with the new weights as additional removal heuristic. Now, all components will have a twofold effect on which cargoes are chosen. First of all, they will have a direct effect as a separate removal heuristic. Secondly, they will also have an effect through the full Shaw removal heuristic.

After running the complete ALNS heuristic for this second setting, the parameter values should be adjusted again. Here, we suggest to take the estimated effect of the relative component on the previous heuristic. With weights w_s and w_c of the Shaw removal heuristic and the heuristic regarding component c respectively. The new parameter value $\hat{p}_c$ of component c can be computed as $\hat{p}_c = w_s p_c + w_c$, with p_C the previous parameter value of component c. It can easily be checked that the new parameter values sum up to one, so this procedure can be repeated until a certain degree of convergence is attained. To get a more robust estimate, this procedure is repeated several times, and the average values of the parameters are taken, scaling them afterwards to sum up to one. Due to the random nature of the ALNS heuristic, there is little guarantee about the time until convergence and the resulting optimality of the parameter values. Therefore, the final parameter values can be refined with local search methods. The next section discusses an approach based on grid search to perform such a local search, which will be used to fine-tune all parameters necessary in the ALNS heuristic.

7.2 Local Search

To fine-tune the parameter values using local search, a subset S of the problem instances is selected. Nine different subsets are used, depending on the size of the problem instances and whether or not we are dealing with a short sea or deep sea shipping problem and mixed or full cargo loads. Regarding size, we differentiate between three different cases. Firstly, there are small problem instances. For these, the ALNS heuristic almost always finds the optimal solution within a few seconds as long as the parameter values are reasonably good. Therefore, considering these instances, no difference is made between the other characteristics (short sea or deep sea shipping and full or mixed cargo loads) of the problem at hand. Here, problem instances with at most twenty cargo loads are considered to be small. Furthermore, there are average and large problem instances. Average problem instances contain at most fifty cargo loads. The remaining problem instances are considered to be large.

This way, every average or large subset contains fifteen or twenty problem instances. Out of these, forty percent of the subset (six or eight problem instances) is used to estimate the best parameter settings and the remaining problem instances are used to evaluate how well the obtained parameter setting perform compared to the starting values. Per subset, the parameters are estimated separately.

For all parameters used in the ALNS heuristic, the current value is kept track of, as well as a lower and upper bound, and a step size. In order to improve the parameter values, in every step of the local search heuristic one parameter will be selected which may be changed if this yields an improvement in performance of the ALNS heuristic. First, the performance will be tested with the original value. Secondly, the same will be done with the original value decreased by the current step size, while staying above the lower bound. Thirdly, this will be repeated with the original value increased by the current step size, while staying below the upper bound.

To estimate the performance of current parameter values, the problem instances from the relevant subset are run for a certain running time, after which the best objective value is returned. It is important to note that the choice of running time is quite important. If it is too short, random effects will be more important than the small difference in parameter values. However, if the running time is too long, the objective value is expected to converge towards the optimal objective value and we might not be able to differentiate between the performance of the different settings.

Also, the total running time of one iteration might become too large. Therefore, we suggest that it is generally more useful to add more problem instances or iterations than to increase the running time. However, if time is not an issue, increasing the running time might turn out to yield a positive effect. In our estimations, the running time is set to be x seconds, where x is the number of cargo loads which should be transported.

After running the ALNS heuristic for all relevant problem instances and parameter settings, the average of the objective values is taken. This yields three mean values: one for the original parameter values, and one for the given parameter increased or decreased by the step size. The performance of a given setting is considered to outperform another setting if its average objective value is ϵ percent lower than the other. Again, the choice of ϵ is quite important. If it is too small, the local search might become more of a random walk (albeit somewhat in the right direction) as too many changes are accepted. However, if ϵ is too large, the neighbourhood might become too small to find any improvements. In our estimations ϵ is put at 0.05 percent.

Depending on the obtained mean values, several cases can be distinguished. If only one of the new settings outperforms the original setting, the relevant parameter value is changed to the new value. If both parameters outperform the original value, the one with the lowest objective value will be chosen.[1] If the original parameter value outperformed the other two, no step will be taken.

Every time a step is taken in a certain direction, another step will be taken in the same direction for as long as this keeps yielding an improvement in average objective value. To save time, only the mean value of the newest setting will be determined by running the ALNS heuristic. For the others, the mean value which was previously found, will be reused. This also avoids that many successive steps in the same direction will be taken, thereby keeping the search local and saving more time.

At this point, it is assumed that no further improvements can be made with the current step size, so it will become twice as small (rounded upwards for parameters which can only take integer values). Due to random effects, this is not necessarily true, but an analysis of the effect of this simplification is left to future research. We suffice to say, that if the starting parameter values are accurate enough, these random effects will have limited consequences.

If none of the previously described cases apply, that is, none of the settings outperforms the other two significantly, the value of the parameter and the step size will remain at their original values.

Table 7.1 presents the starting values, lower and upper bound and step sizes of all parameters relevant for the ALNS heuristic. For convenience, it also mentions the (sub)section where the relevant parameter is explained. Besides the Shaw removal weights, whose starting values are determined as explained in Section 7.1, all values presented in Table 7.1 are obtained by observation and trial-and-error approaches and therefore there should be plenty of room for improvement.

[1] This should not happen too often as long as ϵ is large enough. If it does happen often, consider increasing ϵ.

Table 7.1: Parameter values, bounds and step sizes

Parameter	ξ	φ	χ	ψ	ω	K	r	T	n	d
Starting value	0.7	0.35	0.44	0.14	0.07	6	0.2	2500	50	2
Lower bound	0	0	0	0	0	1	0	1	0	0
Upper bound	1	1	1	1	1	15	1	10000	1000	6
Step size	0.1	0.1	0.1	0.1	0.1	1	0.1	200	10	1
Section	4.2	4.2.1	4.2.1	4.2.1	4.2.1	4.3.2	4.4	4.5	4.5	4.5

All values used in the local search heuristic. Presented are for each parameter, its starting value, the lower and upper bound set for this parameter, and the step size which was used. Also mentioned is the section where the purpose of the parameter is explained.

Chapter 8

Results

8.1 Mathematical Formulation Without Transshipment

First, all 240 problem instances were solved using the formulation in Chapter 3 for mixed cargo loads and the formulation in Chapter 5 adjusted as in Section 5.3.1 for full cargo loads. The maximal running time was set at one hour. The results of this are summarized in Table 8.1. There, for the different problem instances the average gap of the five relevant problems are presented, as well as the average running time in seconds. The gap is calculated with respect to the individual best solutions as in Panels (a) of Appendix A.

Table 8.1: Average gap and total running time of the formulation without transshipment

		Shortsea full		Deepsea full				Shortsea mixed		Deepsea mixed	
C	V	Gap	Sec.	Gap	Sec.	C	V	Gap	Sec.	Gap	Sec.
8	3	0.00	0.30	0.00	1.01	7	3	0.00	0.40	0.00	0.21
11	4	0.00	5.64	0.00	0.78	10	3	0.00	1.51	0.00	0.67
13	5	0.00	11.79	0.00	0.90	15	4	0.00	107.90	0.00	158.35
16	6	0.00	60.25	0.00	2.28	18	5	0.24	1773.80	0.00	1449.37
17	13	0.00	25.85	0.00	4.23	22	6	2.14	3388.69	0.08	1886.54
20	6	0.00	7.13	0.00	4.75	23	13	5.04	3663.28	0.06	1489.58
25	7	0.00	77.60	0.00	12.76	30	6	8.62	3627.95	0.79	3627.68
35	13	0.00	166.60	0.00	751.80	35	7	54.10	3607.72	19.57	3604.39
50	20	1.53	3911.89	0.00	1766.01	60	13	211.32	3605.35	140.53	3604.43
70	30	177.54	3609.37	3.76	3611.20	80	20	251.48	3613.72	200.79	3610.63
90	40	278.41	3643.99	4.88	3623.62	100	30	347.72	3626.63	326.71	3629.05
100	50	272.54	3641.96	6.06	3638.04	130	40	345.72	3674.03	360.15	3689.98

Summary of the results obtained by running the formulations for solving tramp ship routing problems without transshipment possibilities as explained in Chapters 3 and 5. Presented are for the specified number of cargoes (C) and vessels (V) the average gap to the best solutions known in percentages for the five problems with these number of cargoes and vessels and the average running time in seconds.

From Table 8.1 it can be seen that full cargo load problems can be solved to optimality for problem instances of up to 35 cargo loads. For problem instances with 50 cargo loads, optimal solutions where found for all deep sea shipping problems. Also, solving deep sea shipping instances with up to 100 full cargo loads can be solved reasonably well. We believe this is due to the fact that the feasible region of these problems is significantly smaller than those of the others.

For mixed cargo load problems, optimality is proven for all but one instances of up to 18 cargo loads. It can be seen that full cargo load problem instances can be solved much more easily than mixed cargo load problems. This is most likely due to the difference in size of the feasible regions, as many more solutions are possible for mixed cargo load problems. Still, we believe that future research may reveal more promising methods in solving mixed problems as well.

8.2 ALNS Heuristic Without Transshipment

Next, the parameter estimation methods as explained in Chapter 7 were performed. This was done for the eight subsets mentioned earlier (not the one with the smallest cargo loads). The results of the local search are summarized in Table 8.2. It is advised to use the average value for each parameter whenever using the ALNS heuristic for problem instances which have only mediocre similarity to the eight subsets used here.

For full cargo loads the value of φ has no meaning as the cargo sizes are always the same. However, the values are not set at zero, because then the values of the other Shaw parameters would not be comparable between mixed and full cargo load problems (together, the Shaw parameters sum up to one) and consequently, their averages would not be accurate. So, they are set at the average of the four mixed cargo load instances. Then, the Shaw parameters of the full cargo load problems are scaled so that the sum is one and all parameter values are comparable.

Table 8.2: Final parameter values per subset

Subset	ξ	φ	χ	ψ	ω	K	r	T	n	d
SM22–35	0.675	0.391	0.355	0.190	0.063	6	0.150	2100	20	1
SM60–130	0.650	0.261	0.391	0.116	0.232	4	0.250	2900	1	3
SF25–50	0.650	0.388	0.183	0.186	0.244	6	0.200	3200	82	1
SF70–100	0.600	0.337	0.382	0.186	0.096	5	0.250	2500	60	2
DM22–35	0.750	0.368	0.368	0.175	0.088	6	0.175	2600	20	0
DM60–130	0.675	0.299	0.374	0.262	0.065	6	0.200	2700	30	0
DF25–50	0.625	0.366	0.380	0.186	0.069	7	0.119	2200	93	1
DF70–100	0.663	0.396	0.287	0.186	0.132	2	0.125	2300	50	1
Average	0.661	0.351	0.340	0.186	0.124	5	0.184	2563	45	1

Final values of the parameters used in the ALNS heuristic after running the local search heuristic for the different subsets. Also provided are the averages of these values. Subsets are denoted depending on whether they include shortsea (S) or deepsea (D) shipping problems, full (F) or mixed (M) cargo loads, and the number of cargoes (lowest number of cargoes in subset – highest number of cargoes in subset). The average value of φ (indicated by an asterisk) is taken over only the mixed cargo load instances.

For all subsets, the average objective value of the final parameter setting was compared to the average objective value of the starting parameter values. It was found that the average cost decrease was about 539,000 euros per problem instance. This accounted to a decrease of about 1.11 percent.

After determining the final parameter values, the problem instances which had not been used for estimating the parameters were run using the ALNS heuristic for the relevant parameter values from Table 8.2 as explained in Chapter 4. For the small problem instances (up to 18 mixed cargo loads and

20 full cargo loads) it was assumed before that any reasonable parameter settings would generally lead to the optimal value being found. Therefore, for these problems, the average parameter values as in Table 8.2 were used. In every run, the total running time was set at $5x$, where x is the number of cargoes plus the number of vessels in that problem. The results of this are summarized in Table 8.3, where the gap to the best solution is presented, and the time at which the last improvement occurred. The values are taken as an average over ten runs. For problem instances which were used for parameter estimation, the average parameter values as in Table 8.2 are taken. This is done to avoid using these problem instances to estimate the parameters and then use these parameters to run the same problems. Doing so would mean that the problem instances would need to be known before parameter estimation, whereas we assume that the parameter estimation is based on historical data[1].

Table 8.3: Average gap and total running time of the ALNS heuristic without transshipment

		Shortsea full		Deepsea full				Shortsea mixed		Deepsea mixed	
C	V	Gap	Sec.	Gap	Sec.	C	V	Gap	Sec.	Gap	Sec.
8	3	0.00	0.01	0.00	0.00	7	3	0.00	0.05	0.00	0.00
11	4	0.00	3.75	0.00	0.66	10	3	0.00	0.01	0.00	0.01
13	5	0.00	1.56	0.00	0.03	15	4	0.00	3.17	0.00	3.26
16	6	0.00	5.44	0.00	1.94	18	5	0.00	5.22	0.00	12.15
17	13	0.00	16.02	0.00	4.33	22	6	0.20	17.75	0.03	18.98
20	6	0.00	17.69	0.00	1.52	23	13	0.23	35.65	0.00	5.63
25	7	0.02	11.97	0.00	21.53	30	6	1.39	35.73	0.67	42.73
35	13	0.04	60.25	0.47	51.77	35	7	1.95	69.94	0.56	68.95
50	20	0.23	177.11	0.15	132.73	60	13	1.59	209.11	2.37	180.61
70	30	0.58	349.93	0.17	366.43	80	20	1.24	338.66	1.93	370.68
90	40	0.44	503.99	0.35	578.48	100	30	1.16	484.87	1.59	549.09
100	50	0.33	634.83	0.38	677.70	130	40	1.52	687.08	3.19	743.89

Summary of the results obtained by running the ALNS heuristic for solving tramp ship routing problems without transshipment possibilities as explained in Chapter 4 ten times. Presented are for the specified number of cargoes (C) and vessels (V) the average gap to the best solutions known in percentages and the average running time in seconds until the last improvement.

From Table 8.3 it can be seen that the heuristic manages to find reasonably strong solutions for all problem instances. For almost all problem instances with more than 50 full cargo loads, new best solutions are found in at least one of the ten iterations. For mixed problem instances, this is

[1] Although the average parameter values also depend on these problem instances, we assume that the overall effect of this on the final objective values is negligible in comparison to random effects.

the case for about half of them. Altogether, the ALNS heuristic found 59 solutions which yielded an improvement with respect to the previously best known objective values.

8.3 Mathematical Formulation With Transshipment

Next, all 240 problem instances were solved using the formulation in Chapter 5. First, the problems are solved using the adjusted approach as explained in Section 5.3.1 for full cargo loads. This way, a solution is found which does not allow for transshipment. Then, this solution is used as an initial solution for the transshipment problem. The maximal running time is set at one hour for both approaches. The final results are summarized in Table 8.1. There, for the different problem instances the average gap of the five relevant problems is presented, as well as the average running time in seconds. The gap is calculated with respect to the individual best solutions as in Panels (b) of Appendix A.

Table 8.4: Average gap and total running time of the formulation with transshipment

		Shortsea full		Deepsea full				Shortsea mixed		Deepsea mixed	
C	V	Gap	Sec.	Gap	Sec.	C	V	Gap	Sec.	Gap	Sec.
8	3	0.00	757.18	0.00	19.01	7	3	0.00	32.54	0.00	35.58
11	4	0.22	3079.56	0.14	3721.74	10	3	0.54	3285.75	0.00	64.65
13	5	6.99	3610.02	0.19	2363.25	15	4	9.45	5021.43	6.82	3244.43
16	6	5.35	3617.76	5.19	3611.85	18	5	4.74	6811.00	11.32	4721.57
17	13	2.30	3649.84	0.59	3619.01	22	6	10.80	7228.94	14.26	5806.22
20	6	1.35	3634.48	1.93	3618.58	23	13	9.92	7230.48	19.78	5108.68
25	7	3.60	3768.10	2.80	3680.12	30	6	15.06	7259.35	9.33	6634.47
35	13	0.82	4100.58	2.61	4571.13	35	7	57.30	7255.71	29.18	7210.98
50	20	1.62	7257.73	1.09	6498.55	60	13	208.45	7274.60	144.54	7284.50
70	30	147.70	7276.48	4.79	7263.23	80	20	250.76	7322.83	200.74	7317.89
90	40	284.82	7321.44	4.81	7332.86	100	30	278.73	7291.85	323.43	7314.70
100	50	280.61	7339.26	5.96	7308.60	130	40	342.66	7361.18	277.60	7332.75

Summary of the results obtained by running the formulations for solving tramp ship routing problems without transshipment possibilities as explained in Chapter 5. Presented are for the specified number of cargoes (C) and vessels (V) the average gap to the best solutions known in percentages and the average running time in seconds.

At first it may seem that the objective values in Table 8.4 are slightly worse than those in Table 8.1 as the average gap tends to be somewhat larger. However, this gap is taken with respect to objective values which are usually significantly lower than those of the original problem. In fact, because of the two stage approach taken here, the final objective value in this section are almost always lower than those which were obtained when solving the problem without transshipment.

In terms of running time, the results in Table 8.4 are clearly a lot worse than those in Table 8.1. In fact, even for most small problems, no optimal solutions are being found, so the running time is generally at least an hour. Therefore, we recommend to stick with the two stage approach adopted in this paper. First, the original problem should be run until an optimal solution is found, or the optimality gap is small enough. Then, using this solution as a starting point, the transshipment formulation should be run for the remaining time.

Note that the choice of whether or not to include transshipment operations in the model mostly depends on the available running time. If only limited time is available, it is advised to exclude transshipment possibilities and focus on fine-tuning the available solution methods for the simpler model. However, if time is not much of an issue, significant decreases in shipping costs can be found by including transshipments, so this should be the preferred option.

8.4 ALNS Heuristic With Transshipment

Finally, the ALNS heuristic was run while including transshipment operations as explained in Chapter 6. For all problem instances, the average parameter values as in Table 8.2 have been used. In every run, the total running time was set at $10x$, where x is the number of cargoes plus the number of vessels in that problem. The results of this are summarized in Table 8.5, where the gap to the best solution is presented, and the time at which the last improvement occurred. The values are taken as an average over ten runs.

Table 8.5: Average gap and total running time of the ALNS heuristic with transshipment

		Shortsea full		Deepsea full				Shortsea mixed		Deepsea mixed	
C	V	Gap	Sec.	Gap	Sec.	C	V	Gap	Sec.	Gap	Sec.
8	3	0.00	0.59	0.00	1.30	7	3	0.00	0.45	0.00	0.91
11	4	0.01	0.63	0.99	1.19	10	3	0.00	1.57	0.06	7.59
13	5	0.31	0.12	0.00	2.32	15	4	2.34	4.55	1.63	17.93
16	6	1.23	5.80	1.08	1.00	18	5	0.94	31.53	2.33	11.41
17	13	0.07	19.40	0.06	0.44	22	6	2.69	130.05	4.00	134.55
20	6	0.00	20.34	0.53	48.01	23	13	1.73	154.10	3.36	182.54
25	7	0.41	108.92	0.50	83.44	30	6	3.17	155.23	3.98	191.21
35	13	0.22	130.02	1.47	286.15	35	7	4.14	202.58	6.07	195.89
50	20	0.30	268.79	0.46	285.32	60	13	3.12	505.48	6.18	450.19
70	30	1.84	637.27	0.46	665.51	80	20	3.35	654.81	4.02	689.12
90	40	1.47	951.99	0.48	982.49	100	30	3.28	887.74	4.53	928.15
100	50	1.49	1092.60	0.57	1087.58	130	40	3.09	1215.69	6.13	1388.66

Summary of the results obtained by running the ALNS heuristic for solving tramp ship routing problems with transshipment possibilities as explained in Chapter 6 ten times. Presented are for the specified number of cargoes (C) and vessels (V) the average gap to the best solutions known in percentages and the average running time in seconds.

From Table 8.5, it can be seen that the heuristic manages to find reasonably good solutions for most problem instances. Compared to Table 8.3, there are more problem instances for which the gap is more than zero percent. However, the corresponding objective values still tend to be at least as good as those presented there. Note that the last improvement usually happens quite well in advance of the maximum allowed time, indicating that the heuristic tends to get stuck in (local) optima. To increase performance, one might like to try tweaking the parameters (especially those which are most relevant when running time increases: r, T, n, and d.) by local search methods as in Chapter 7 using a higher running time, or increasing the number of segments. Altogether, by using transshipment operations, improvements in objective value have been found for 144 out of 240 problem instances. In some cases, this led to cost reductions of well over ten percent.

Note that for some problem instances, this cost reduction might not require transshipment operations. First of all, only the transshipment formulation managed to benefit from detours for which the triangle inequality did not hold[2]. Secondly, the transshipment heuristic might be able to find solutions which the original heuristic did not find, even though these did not include transshipment. This could for example happen when the transshipment heuristic would avoid getting stuck in a local optimum by temporarily including transshipping solutions. Finally, as the heuristic uses random approaches to determine its solutions, it might be that the transshipment heuristic would benefit from having a different random number stream leading to some 'lucky finds'.

[2]Note that this can be easily circumvented by ensuring that the triangle inequality does hold for all problem instances. For example, when $C_{ijv} > Cikv + C_{kjv}$ for some i, j, k, v, we can set $C_{ijv} = C_{ikv} + C_{kjv}$ without loss of generality (one could always just sail past a certain port).

8.5 Model Recommendations

Four different models have been used in this paper. First of all, a mathematical formulation was used to solve tramp shipping problems without transshipment. Generally, this model managed to find good solutions of problems with up to 50 cargo loads. We advice that this model should be used for smaller firms which do not have access to the most advanced software or need to make their decisions on a more real-time basis. For example, when orders are not known a priori, but come in as a continuous flow, time may be a serious issue, and therefore this simple model might be the best choice.

Secondly, a heuristic approach was presented to efficiently solve larger problems without transshipment. This model managed to find good solutions for all problems at hand, regardless of their size. We advice that this model should be used for larger firms which need to make their decisions on a real-time basis, or are time-constrained for other reasons.

Thirdly, the tramp shipping and scheduling problem with transshipment was solved with another mathematical formulation. Using the model without transshipment as an initial solution, this approach found reasonably good solutions for problems with up to 50 cargo loads. However, it can not guarantee optimality of the objective values which are being found, except for the smallest of problem instances. Therefore, we advice that this model should only be used by small firms which have a lot of time to determine the best solutions.

Finally, the extended problem was solved with a heuristic approach. As this model only requires a moderate adjustment to the heuristic without transshipment, it generally manages to find solutions which are just as good as those found by the original model. We suggest that large firms which have a reasonable amount of time, should stick with this option. However, even when time is an issue, this model should be considered. For example, by decreasing the probability of a transshipment insertion, or by only allowing transshipment insertions after a given amount of running time, this model can be used to run almost as fast as the previous model. Only when running time is extremely limited, or when problem instances become very large, will this lead to significantly worse objective values.

Chapter 9

Conclusion

9.1 Summarizing Remarks

Throughout this paper, mathematical formulations and heuristical approaches were used to solve tramp ship routing and scheduling problems with and without transshipment possibilities.

Regarding the problems without transshipment possibilities, it was found that optimal solutions can be determined for problems with smaller numbers of cargo loads, but that trying to solve the problem to optimality with larger numbers of cargo loads (more than 50), tends to cause very long running times. For these cases, the presented heuristic provides much better solutions.

By including transshipment operations, this problem became even more severe, generally only finding optimal solutions for the smallest of problem instances. Once again, the adjusted heuristic manages to find reasonably good solutions even for the largest of instances.

Regarding the parameter values necessary to run these heuristics, nine different sets of parameters have been provided, which have been shown to perform well at deriving low-cost solutions. Furthermore, a simple local search heuristic has been presented, which can be used to fine-tune these parameter settings to the problems at hand.

Altogether, this paper has determined or confirmed new optimal values for 15 problem instances and found improved solutions for 59 problem instances. Also, by including transshipment possibilities in the model, an improvement in costs was found for 144 problem instances. In some cases, this corresponded to cost savings of well over ten percent. The final objective values that were determined in this paper can be used as a benchmark to estimate future solution approaches regarding tramp shipping problems with or without transshipment possibilities.

9.2 Simplifications and Future Research Possibilities

As already mentioned in Section 1.3.2, this paper has assumed that vessel speed is constant, that time windows and fleet sizes are fixed, and that cargoes can not be split over multiple vessels.

This need not be in reality, so future research should be performed to develop models suitable for including these kinds of features in the models presented in this paper.

Furthermore, this paper has not performed an exhaustive investigation of including transshipment in the ALNS heuristic. More sophisticated insertion and removal methods might be available, or two stage approaches which do not allow transshipment at the first stage might turn out to outperform the methodology presented here. The results obtained in this paper can serve as a benchmark for this sort of research.

Finally, full cargo load problems were solved with much more efficiency than mixed cargo load problems. To a certain extent, this is inherent to the structure of the problems, as full cargo loads allow for a much smaller feasible region. However, it may be possible to tweak the presented formulations such that they manage to perform well for these type of problems as well.

We hope future research will shed light on these matters.

References

Appelgren, L. H. (1969). A column generation algorithm for a ship scheduling problem. *Transportation Science*, *3*(1), 53.

Appelgren, L. H. (1971). Integer programming methods for a vessel scheduling problem. *Transportation Science*, *5*(1), 64.

Beasley, J. E. (1990). Or-library: distributing test problems by electronic mail. *Journal of the operational research society*, *41*(11), 1069–1072.

Bräysy, O., Hasle, G., & Dullaert, W. (2004). A multi-start local search algorithm for the vehicle routing problem with time windows. *European Journal of Operational Research*, *159*(3), 586–605.

Brønmo, G., Christiansen, M., Fagerholt, K., & Nygreen, B. (2007). A multi-start local search heuristic for ship scheduling—a computational study. *Computers & Operations Research*, *34*(3), 900–917.

Christiansen, M., Fagerholt, K., & Ronen, D. (2004). Ship routing and scheduling: Status and perspectives. *Transportation science*, *38*(1), 1–18.

Cordeau, J.-F., Laporte, G., & Mercier, A. (2001). A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational research society*, *52*(8), 928–936.

Croes, G. A. (1958). A method for solving traveling-salesman problems. *Operations research*, *6*(6), 791–812.

Dantzig, G. B. (1963). *Linear programming and extensions. princeton landmarks in mathematics and physics.* Princeton University Press Princeton.

Drexl, M. (2013). Applications of the vehicle routing problem with trailers and transshipments. *European Journal of Operational Research*, *227*(2), 275–283.

Fagerholt, K., Heimdal, S., & Loktu, A. (2000). Shortest path in the presence of obstacles: An application to ocean shipping. *Journal of the operational research society*, *51*(6), 683–688.

Garcia, B.-L., Potvin, J.-Y., & Rousseau, J.-M. (1994). A parallel implementation of the tabu search heuristic for vehicle routing problems with time window constraints. *Computers & Operations Research*, *21*(9), 1025–1033.

Hemmati, A., Hvattum, L. M., Fagerholt, K., & Norstad, I. (2014). Benchmark suite for industrial and tramp ship routing and scheduling problems. *INFOR: Information Systems and Operational Research*, *52*(1), 28–38.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, *220*(4598), 671–680.

Koch, T., Achterberg, T., Andersen, E., Bastert, O., Berthold, T., Bixby, R. E., ... others (2011). Miplib 2010. *Mathematical Programming Computation*, *3*(2), 103.

Korsvik, J. E., Fagerholt, K., & Laporte, G. (2011). A large neighbourhood search heuristic for ship routing and scheduling with split loads. *Computers & Operations Research*, *38*(2), 474–483.

Land, A. H., & Doig, A. G. (1960). An automatic method of solving discrete programming problems. *Econometrica: Journal of the Econometric Society*, 497–520.

Lawrence, S. A. (1972). *International sea transport: the years ahead.* Lexington Books.

Lin, S. (1965). Computer solutions of the traveling salesman problem. *The Bell system technical journal*, *44*(10), 2245–2269.

Lin, S., & Kernighan, B. W. (1973). An effective heuristic algorithm for the traveling-salesman problem. *Operations research*, *21*(2), 498–516.

Norstad, I., Fagerholt, K., & Laporte, G. (2011). Tramp ship routing and scheduling with speed optimization. *Transportation Research Part C: Emerging Technologies*, *19*(5), 853–865.

Pantuso, G., Fagerholt, K., & Hvattum, L. M. (2014). A survey on maritime fleet size and mix problems. *European Journal of Operational Research*, *235*(2), 341–349.

Potvin, J.-Y., & Rousseau, J.-M. (1995). An exchange heuristic for routeing problems with time windows. *Journal of the Operational Research Society*, *46*(12), 1433–1446.

Ronen, D. (1983). Cargo ships routing and scheduling: Survey of models and problems. *European Journal of Operational Research*, *12*(2), 119–126.

Ronen, D. (1993). Ship scheduling: The last decade. *European Journal of Operational Research*, *71*(3), 325–333.

Ropke, S., & Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation science*, *40*(4), 455–472.

Shaw, P. (1997). A new local search algorithm providing high quality solutions to vehicle routing problems. *APES Group, Dept of Computer Science, University of Strathclyde, Glasgow, Scotland, UK*.

Tillman, F. A., & Cain, T. M. (1972). An upperbound algorithm for the single and multiple terminal delivery problem. *Management Science*, *18*(11), 664–682.

UNCTAD. (2017). Review of maritime transport. *United Nations, Geneva*.

Wang, S., & Meng, Q. (2012). Liner ship fleet deployment with container transshipment operations. *Transportation Research Part E: Logistics and Transportation Review*, *48*(2), 470–484.

Xinlian, X., Tengfei, W., & Daisong, C. (2000). A dynamic model and algorithm for fleet planning. *Maritime Policy & Management*, *27*(1), 53–63.

Yu, B., Peng, Z., Tian, Z., & Yao, B. (2017). Sailing speed optimization for tramp ships with fuzzy time window. *Flexible Services and Manufacturing Journal*, 1–23.

Appendix A

Best Known Results

A.1 Short Sea Shipping With Mixed Cargo Loads

Table A.1: Individual best known results for short sea shipping problems with mixed cargo loads

C	V	Instance #1	Instance #2	Instance #3	Instance #4	Instance #5
7	3	**1,476,444**	**1,134,176**	**1,196,466**	**1,256,139**	**1,160,394**
10	3	**2,083,965**	**2,012,364**	**1,986,779**	**2,125,461**	**2,162,453**
15	4	**1,959,153**	**2,560,004**	**2,582,912**	**2,265,396**	**2,230,861**
18	5	**2,374,420**	**2,987,358**	**2,301,308**	**2,400,016**	**2,813,167**
22	6	**3,928,483**	3,683,436	3,264,770	3,228,262	3,770,560
23	13	2,276,832	2,255,469*	2,362,503	2,250,110	2,325,941
30	6	4,958,542	4,549,708*	4,106,293	4,449,449*	4,528,514
35	7	4,893,734*	4,533,265*	4,433,847	4,580,935	5,511,661
60	13	8,202,138	8,055,970	7,651,685	8,593,410	8,950,046
80	20	10,376,392	10,387,253	9,763,401	11,440,433*	10,968,104*
100	30	12,763,888*	13,057,536	12,088,444	13,772,836*	13,442,180*
130	40	16,524,192	16,713,067	15,862,154	17,305,841	18,533,293

(a) Results without transshipment

C	V	Instance #1	Instance #2	Instance #3	Instance #4	Instance #5
7	3	**1,476,444**	**1,134,176**	**1,033,363**	**1,256,139**	**1,160,394**
10	3	2,083,965	**2,012,364**	1,934,340	2,125,461	**2,162,453**
15	4	1,936,608	2,360,492	2,328,306	2039618	2,050,572
18	5	2,235,790	2,938,697	2,218,157	2,400,016	2,536,775
22	6	3,518,821	3,203,696	3,127,321	3,154,561	3,448,296
23	13	2,141,795	2,182,939	2,258,687	2,146,753	2,233,245
30	6	4,544,156	4,347,257	4,047,325	3,984,392	4,386,450
35	7	4,698,306	4,465,485	4,407,901	4,572,472	5,232,303
60	13	8,202,138	8,055,970	7,651,685	8,579,722	8,950,046
80	20	10,376,392	10,387,253	9,730,948	11,319,367	10,964,239
100	30	12,763,888	13,057,536	12,088,444	13,772,836	13,442,180
130	40	16,524,192	16,713,067	15,862,154	17,305,841	18,533,293

(b) Results with transshipment

This table presents the best known objective values for the individual problem instances regarding short sea shipping instances with mixed cargo loads. For the given number of cargoes (C) and vessels (V), Panel A.1a presents the results without transshipment possibilities and Panel A.1b presents the results with transshipment possibilities. Known optimal values are indicated with bold face font. Objective values for the model without transshipment which were not known before are indicated with an asterisk.

A.2 Short Sea Shipping With Full Cargo Loads

Table A.2: Individual best known results for short sea shipping problems with full cargo loads

C	V	Instance #1	Instance #2	Instance #3	Instance #4	Instance #5
8	3	**1,391,997**	**1,246,273**	**1,698,102**	**1,777,637**	**1,636,788**
11	4	**1,052,463**	**1,067,139**	**1,212,388**	**1,185,465**	**1,310,285**
13	5	**2,034,184**	**2,043,253**	**2,378,283**	**2,707,215**	**3,011,648**
16	6	**3,577,005**	**3,560,203**	**4,081,013**	**3,667,080**	**3,438,493**
17	13	**2,265,731**	**3,154,165**	**2,699,378**	**2,806,231**	**2,910,814**
20	6	**2,973,381**	**3,206,514**	**3,197,445**	**3,342,130**	**3,156,378**
25	7	**3,833,588**	**3,673,666**	**4,238,213**	**4,260,762**	**4,069,693**
35	13	**2,986,667**	**3,002,973***	**3,084,339**	**3,952,461**	**3,293,086**
50	20	7,258,266*	7,468,471*	6,938,306	8,933,847*	7,324,910*
70	30	10,055,496*	10,503,191	10,254,939*	10,875,943*	10,901,313*
90	40	13,435,834*	13,924,937*	12,698,188*	14,463,335*	13,650,213*
100	50	13,858,105*	14,704,972*	13,200,886*	14,909,717*	14,092,518*

(a) Results without transshipment

C	V	Instance #1	Instance #2	Instance #3	Instance #4	Instance #5
8	3	1,229,641	**1,246,273**	**1,698,102**	**1,777,637**	**1,636,788**
11	4	1,052,463	1,066,216	**1,212,388**	1,185,465	1,296,163
13	5	1,964,747	2,039,953	2,378,283	2,286,767	2,660,730
16	6	3,374,308	3,552,909	3,683,011	3,512,373	3,263,892
17	13	2,265,731	3,127,954	2,699,312	2,806,231	2,630,810
20	6	2,973,381	3,003,860	3,197,445	3,342,130	3,156,378
25	7	3,577,044	3,673,666	3,887,696	4,200,414	4,054,946
35	13	2,986,667	3,002,973	2,979,011	3,945,020	3,281,337
50	20	7,258,266	7,429,425	6,933,707	8,933,847	7,324,795
70	30	9,733,077	10,355,251	10,064,810	10,722,498	10,567,126
90	40	13,060,273	13,872,407	12,493,428	14,098,963	13,508,880
100	50	13,723,843	14,398,494	12,853,611	14,498,729	13,786,822

(b) Results with transshipment

This table presents the best known objective values for the individual problem instances regarding short sea shipping instances with full cargo loads. For the given number of cargoes (C) and vessels (V), Panel A.2a presents the results without transshipment possibilities and Panel A.2b presents the results with transshipment possibilities. Known optimal values are indicated with bold face font. Objective values for the model without transshipment which were not known before are indicated with an asterisk.

A.3 Deep Sea Shipping With Mixed Cargo Loads

Table A.3: Individual best known results for deep sea shipping problems with mixed cargo loads

C	V	Instance #1	Instance #2	Instance #3	Instance #4	Instance #5
7	3	**5,233,464**	**6,053,699**	**5,888,949**	**6,510,656**	**7,220,458**
10	3	**7,986,248**	**7,754,484**	**9,499,357**	**8,617,192**	**8,653,992**
15	4	**13,467,090**	**12,457,251**	**12,567,396**	**11,764,241**	**10,833,640**
18	5	**43,054,055**	25,068,287	**29,211,238**	**32,281,904**	**40,718,028**
22	6	**41,176,718**	**37,236,363**	**38,215,238**	34,129,809	**46,379,332**
23	13	**41,002,992**	28,014,147	**29,090,422**	**33,685,274**	**38,664,843**
30	6	19,227,093	16,802,142*	21,183,928*	21,076,728	24,490,671
35	7	65,082,675	54,836,748*	56,243,874*	61,354,812*	63,930,423*
60	13	81,709,586	75,787,340*	92,585,918	90,465,188*	88,769,627*
80	20	72,019,446	74,476,462	78,918,099	76,459,495	75,378,149
100	30	153,747,613	153,603,159*	152,338,747	156,960,390*	162,084,771
130	40	239,877,031	234,909,061	244,239,529	228,680,203	246,883,618

(a) Results without transshipment

C	V	Instance #1	Instance #2	Instance #3	Instance #4	Instance #5
7	3	**5,233,464**	**6,053,699**	**5,888,949**	**6,510,656**	**7,220,458**
10	3	**7,986,248**	**7,754,484**	**9,402,796**	**8,617,192**	**8,653,992**
15	4	13,110,773	11,215,880	**12,567,396**	11,398,082	10,833,640
18	5	39,942,617	22,218,783	27,114,309	32,281,904	40,477,390
22	6	37,753,588	33,616,022	35,943,245	31,401,361	42,026,770
23	13	36,304,032	24,796,616	27,613,634	28,693,963	34,347,973
30	6	17,483,566	15,597,111	19,154,274	18,872,426	21,435,290
35	7	58,144,707	52,343,630	51,737,427	54,580,329	55,227,235
60	13	79,833,749	75,787,340	89,020,981	87,875,772	87,737,359
80	20	71,966,996	74,476,462	76,387,945	76,459,495	75,378,149
100	30	153,747,613	153,603,159	152,338,747	156,510,180	162,084,771
130	40	239,877,031	234,909,061	244,239,529	228,680,203	246,883,618

(b) Results with transshipment

This table presents the best known objective values for the individual problem instances regarding deep sea shipping instances with mixed cargo loads. For the given number of cargoes (C) and vessels (V), Panel A.3a presents the results without transshipment possibilities and Panel A.3b presents the results with transshipment possibilities. Known optimal values are indicated with bold face font. Objective values for the model without transshipment which were not known before are indicated with an asterisk.

A.4 Deep Sea Shipping With Full Cargo Loads

Table A.4: Individual best known results for deep sea shipping problems with full cargo loads

C	V	Instance #1	Instance #2	Instance #3	Instance #4	Instance #5
8	3	**9,584,863**	**9,369,654**	**4,596,681**	**6,899,730**	**6,815,253**
11	4	**34,854,819**	**25,454,434**	**29,627,143**	**33,111,680**	**28,175,914**
13	5	**11,629,005**	**11,820,655**	**9,992,593**	**12,819,619**	**10,534,892**
16	6	**51,127,590**	**44,342,796**	**45,391,842**	**39,687,114**	**42,855,603**
17	13	**17,316,720**	**12,194,861**	**12,091,554**	**12,847,653**	**13,213,406**
20	6	**16,406,738**	**16,079,401**	**17,342,200**	**16,529,748**	**17,449,378**
25	7	**22,773,158**	**20,206,329**	**19,108,952**	**22,668,675**	**23,036,603**
35	13	**86,951,609**	**83,422,071**	**83,898,591**	**91,970,481**	**91,123,040***
50	20	**41,310,946***	**37,784,994***	**39,841,724***	**43,941,098**	**41,947,437***
70	30	142,752,237*	135,180,647*	162,812,826*	155,947,901*	157,007,780*
90	40	191,302,974*	190,305,108*	211,396,320*	211,046,180	198,557,461*
100	50	206,463,427*	208,105,662*	217,952,003*	221,093,259*	224,011,198*

(a) Results without transshipment

C	V	Instance #1	Instance #2	Instance #3	Instance #4	Instance #5
8	3	**9,584,863**	**9,351,269**	**4,596,681**	**6,899,730**	**6,815,252**
11	4	**34,603,069**	25,454,434	29,265,506	**33,111,680**	28,175,913
13	5	11,521,266	**11,802,270**	**9,992,593**	**12,819,619**	10,534,892
16	6	44,349,354	44,249,373	42,809,297	39,190,807	41,416,201
17	13	17,316,719	12,194,861	12,091,554	12,607,670	13,073,810
20	6	16,044,137	16,042,592	17,214,206	16,243,080	16,525,126
25	7	20,955,087	20,206,329	19,108,952	21,620,713	22,929,054
35	13	86,190,351	79,001,666	81,536,565	89,348,906	90,446,500
50	20	41,310,946	37,545,541	39,253,054	42,526,120	41,947,437
70	30	136,245,501	135,103,989	162,621,840	155,947,901	156,982,669
90	40	191,302,974	190,142,522	211,396,320	211,046,180	198,487,194
100	50	206,420,024	208,105,662	217,952,003	221,093,259	224,011,198

(b) Results with transshipment

This table presents the best known objective values for the individual problem instances regarding deep sea shipping instances with full cargo loads. For the given number of cargoes (C) and vessels (V), Panel A.4a presents the results without transshipment possibilities and Panel A.4b presents the results with transshipment possibilities. Known optimal values are indicated with bold face font. Objective values for the model without transshipment which were not known before are indicated with an asterisk.