ERASMUS UNIVERSITY ROTTERDAM

Erasmus School of Economics
Bachelor Econometrics and Operations Research

Bachelor Thesis in Operations Research and Quantitative Logistics

Industrial and Tramp Ship Routing with Varying Sailing Speeds

Student name	*Student ID number*
Hein Tegelberg	385981

Supervisor	*Second assessor*
Nemanja Milovanovic	Wilco van den Heuvel

July 8, 2018

Abstract

This paper looks into different approaches at solving the routing and scheduling problem present in industrial and tramp maritime shipping. Using benchmark instances, solutions are provided by a solver for a mathematical formulation of the problem and by an adaptive large neighborhood search heuristic. In addition, a layer of realism is added by introducing choices for speed tiers into the existing problem. The extended problem is also fed to a solver using the same benchmarks, and its results are analyzed in order to determine whether this is a meaningful addition to more sophisticated solving techniques in the future. These three approaches are compared with regards to performance for each of the benchmark instances. From this, conclusions are drawn on which type of approach is suitable for which type of instance. Ultimately, recommendations are made for future research into improving the approaches at solving the routing and scheduling problem that are discussed here.

Contents

1 Introduction

In the current world of growing globalism, consumers and industries expect products and services to be available as and when they desire. As a result, the demand for maritime transportation has increased worldwide. In any growing business, competition is high and therefore costs should be as low as possible. In addition, an increasing number of people voice their concerns over how large-scale transportation is affecting our planet and expect shipping companies to do something about it. This paper gives an insight in how these costs are derived in two types of maritime transportation, that are distinguished as tramp and industrial shipping.

Tramp shipping can be seen as a taxi service for cargoes, where ships pick up both mandatory cargoes that are determined in contracts and optional cargoes that are available for transport, much like a regular taxi service. In industrial shipping, the company owns the ships and the cargoes and tries to minimize the costs of performing all their required deliveries. An important part of minimizing costs and fuel consumption is found in the so-called ship routing and scheduling problem. In Hemmati et al. (2014), this optimization is executed with an approach based on solving a mathematical formulation of the problem and an approach that uses an Adaptive Large Neighborhood Search (ALNS). In addition to this, this paper introduces an approach that makes use of a realistic choice in tiers of ship sailing speed in order to further reduce fuel consumption and costs.

The problem introduced above is one that should be able to be solved relatively quickly and adequately, providing a good solution within an acceptable time frame. This paper implements and compares different approaches to the ship routing and scheduling problem, answering the question which approach is best suited to which situation. To this end, the paper includes benchmark instances that allow for proper comparison. It also does research into a meaningful and realistic implementation of varying sailing speeds and does suggestions with regards to improving this implementation.

In Section 2, previous research is reviewed and its relevance to this paper is explained. Section 3 introduces the data that will be used in this research. In this section the benchmark instances are discussed and the implementation of speed tiers is conceived. In Section 4, the methodology of this research is given, in which all three approaches to the problem are fleshed out. Section 5 presents the result of the computational experiments and comments on those. In addition, it analyzes the relevance of the choice in sailing speed. Section 6 summarizes the report and Section 7 provides recommendations for future research.

2 Literature Review

The question posed at the beginning of this research consists of a union between the field of maritime shipping, in particular tramp and industrial shipping, and the field of mathematical programming. It therefore bases itself on earlier work in these fields, which in turn is a culmination of previous research. This paper combines the findings from some of these works and extends upon them. Within maritime transportation, it is common to define three modes of operation, liner, industrial and tramp shipping (Lawrence, 1972). In this paper only industrial and tramp shipping are considered.

Because this paper aims to compare several the effectiveness of different solving methods on different problems, it is important that relevant and meaningful comparisons can be made. It has been established that in order to compare certain methods of mathematical programming based on their performance, it is important that several benchmark instances are used (Malliappi, Bennell, & Potts, 2011). These benchmarks are taken from Hemmati et al. (2014), and will be expanded upon in the Data section.

The foundations for the mathematical formulation are also found in Hemmati et al. (2014), where a formulation with constant sailing speed is given. An alternative formulation with the addition of speed alterations is provided in Norstad et al. (2011). It should be noted that the problems provided in these works are both NP-hard. In response, both papers propose heuristic methods to determine solutions that are adequate and reached within an acceptable time limit.

In Hemmati et al. (2014) an ALNS method was used as an alternative to the use of a solver for a mathematical formulation of the problem. It concluded that, although the solver was faster in solving to optimality for small instances, the ALNS method almost always also found these optimal values. In addition, the ALNS method was noticed to scale well for instances where more ships and cargoes were concerned. For some of the heuristics used in the ALNS procedure we refer to their relevant papers in Section 4.

Norstad et al. (2011) attempted to define the adaptive speed problem with an approach where sailing costs were approximated as a quadratic function of sailing speed. This led to a focus on solving optimal speeds for a ship with a given route using several additional algorithms, which proved very useful for that certain problem. This continuous function proved to be very costly to implement in our problem. How the choices in speed were eventually implemented is discussed in the following section on data.

The ALNS as described in Hemmati et al. (2014) is built up from several smaller heuristics that work together to finally provide an adequate solution. In following research, it was found that only one area of this larger framework could be improved by removing randomness (Hemmati & Hvattum, 2017). As further research has already been performed by the original author, this paper will not elaborate on improving the ALNS.

3 Data

In this section, the data that are used during this research are discussed. The benchmark instances are looked at in detail, and the derivation for the implementation of speed tiers is described.

3.1 Benchmark Instances

The importance of benchmarks was mentioned earlier, along with the promise to elaborate on these as a part of this section. For these benchmarks we look to Hemmati et al. (2014), whose authors created an instance generator and used this to determine several benchmarks. For additional information on the creation of these benchmarks, we refer the reader to the paper from which they stem, however, there are a few important aspects that will be elaborated upon here. Each instance contains the following data:

Port The port and its name, where ships can pick up and deliver cargoes.

Vessel Each vessel is specified using the following characteristics:

- **Index** The index of the vessel.
- **Home Port** The port from which the vessel starts.
- **Starting Time** The time at which the vessel becomes available.
- **Capacity** The maximal load that the vessel can carry.
- **Cargo List** A list of all cargoes that can be transported by the ship.

Cargoes Each cargo is specified using the following characteristics:

- **Index** The index of the cargo.
- **Origin Port** The port where the cargo must be picked up. This also has an earliest possible pickup time and a latest possible pickup time.
- **Destination Port** The port where the cargo must be delivered. This also has an earliest possible delivery time and a latest possible delivery time.
- **Size** The size of the cargo.
- **Charter Cost** The cost that the shipping company incurs when the cargo is not transported by their fleet.

Voyages The voyages that vessels could take between ports. Each voyage is specified using the following characteristics:

- **Vessel Index** The index of the vessel that this voyage concerns.
- **Origin Port** The port from which the vessel leaves for this voyage.
- **Destination Port** The port at which the vessel arrives from this voyage.
- **Travel Time** The time that this vessel takes to make this voyage.
- **Travel Cost** The fuel costs that this vessel incurs on this voyage.

Loading and Unloading Specifics For each vessel, for each cargo, certain variables come into play when that vessel picks up that cargo:

- **Loading Time** The time it takes to load the cargo onto the vessel at its origin port.
- **Unloading Time** The time it takes to unload the cargo from the vessel at its destination port.
- **Port Costs** Both the origin and destination port have certain costs that a vessel needs to pay when it wants to enter the port.

It should be noted that the Triangle Inequality holds. This means that for any three ports i, j, k it holds that $c_{ij} \leq c_{ik} + c_{kj}$ and $t_{ij} \leq t_{ik} + t_{kj}$, where c_{ij} and t_{ij} denote the travel costs and travel time for any voyage between ports i and j, respectively.

These instances are created with several different characteristics. The set of ports can be either in a close vicinity of one another in the so-called Short Sea set, or in a larger distance from one another - the Deep Sea set. Additionally, there is a distinction between full load cargoes and mixed load. Instances with full load cargoes have vessels with a capacity that is equal to the size of each cargo. Instances with mixed load cargoes do not have this, and might therefore have several cargoes in the hold of a vessel at any given time. The instance generator also leaves the option to choose a specific number of vessels and cargoes, allowing the user to decide on the size of the benchmark instance created. We do not create new benchmarks, but simply use the ones that are used in Hemmati et al. (2014).

3.2 Speed Tiers

It should be noted that these benchmark instances have a given travel time and travel cost on each voyage. Part of these times and costs are incurred in the ports and can be explained by harbor fees and (un)loading times. Seeing how these will not be altered due to a change in sailing speed, we only consider the influence of speed on the rest of the time and costs, that are specific to the voyage. Our research aims to implement varying speed on a voyage, which demands additional data transformations. To find what speeds could be included in this research, it is necessary to find at what speed the vessels are sailing in the benchmark instances and at what speed interval vessels are capable of sailing.

In the benchmark instance, the travel time (this is excluding loading times) from Rotterdam to Bilbao for one of the ships is given as 55 hours. Using the SEA-DISTANCES website[1], we found that this travel time corresponds to a vessel speed of 14 knots. The name given to this ship is *Handysize-BulkCarrier_1*, which is a middle-sized type of bulk carrier. Research found that the maximum service speed for the Handysize is 15 knots[2], which is the speed the ship can maintain for daily operation. This is roughly 7% higher than the current speed that is used. For our research we therefore define five different speed tiers, with steps of 7%.

The fuel consumption increase (and with it, the travel cost increase), is derived from the function given in Norstad et al. (2011) which is given as

$$c(v) = 0.0036v^2 - 0.1015v + 0.8848,$$

where $c(v)$ is the fuel consumption in tonnes per nautical mile as a function of the speed v in knots. Because this function is defined for a larger and faster vessel, with a sailing speed between 14 and 22 knots, an adaptation was made for our Handysize vessel which led to the following characteristics for our varying tiers of speed.

[1] https://sea-distances.org/

[2] https://www.quora.com/Merchant-Navy-What-is-the-average-speed-of-a-ship

Table 1: Speed tiers for Handysize vessel

Tier	Speed increase	Speed in knots	Cost increase
5	+ 7%	14.98	15.3%
4	0%	14.00	0.0%
3	-7%	13.02	-13.3%
2	-14%	12.04	-24.5%
1	-21%	11.06	-33.7%

Although not all vessels have the same speed in the instances, we assume that the percentages in each speed tier remain the same for each vessel. Using these tiers, additional data was generated for each voyage, which means there are now five possible choices for a pair of travel time and travel costs within each voyage. This means that the instance description for a voyage now reads:

Voyages The voyages that vessels could take between ports. Each voyage is specified using the following characteristics:

- **Vessel Index** The index of the vessel that this voyage concerns.
- **Origin Port** The port from which the vessel leaves for this voyage.
- **Destination Port** The port at which the vessel arrives from this voyage.
- **Speed Tier** One of five tiers with associated travel time and costs.
  - **Travel Time** The time that this vessel takes to make this voyage at this speed.
  - **Travel Cost** The fuel costs that this vessel incurs on this voyage at this speed.

Having described all the required data, it is now possible to introduce the methods in which they are used.

4 Methodology

As stated before, the aim of this paper is to compare three different approach of solving a ship routing and scheduling problem. The first approach that will be examined is through the fixed speed approach, where the problem is captured in a mathematical formulation and run through a so-called solver. This solver attempts to solve the problem to optimality, a process that takes exponentially longer as the problem becomes larger. The second approach is similar to the first, but relaxes the problem to allow for different sailing speeds. With this relaxation a new mathematical formulation is derived which is also run through a solver. The final method is an Adaptive Large Neighborhood Search (ALNS), which employs several methods that remove and add cargoes to the current solution, starting at a feasible but suboptimal initial solution.

4.1 Mathematical Formulation with Fixed Speed

We consider shipping companies with heterogeneous fleets, which means that each ship may differ with regards to which cargoes they are equipped to carry, and with regards to specifications such as hold capacity, speed and cost. In addition, each ship has a given starting point and a given starting time at which previous assignments have been fully finished. When a cargo is not carried by a ship in the fleet, so-called charter costs are incurred.

Ultimately, the decision lies in what cargoes will be assigned to which ship in the fleet, and what cargoes will not be carried by any ship in the fleet. For this we define the set of ships by V, where each ship $v \in V$ has a capacity K_v. There are n given cargoes, where cargo i originates from its loading port node i, and needs to be delivered to its unloading port node $n+i$. In addition, the volume of cargo i is given as Q_i. The sets of loading and unloading nodes are denoted by N^P and N^D, respectively. Because not every ship can visit every port, we define the set of nodes that can be visited by ship v as N_v. This set of nodes includes the initial position $o(v)$ and the (yet to be decided) final position $d(v)$. From this set N_v we derive the set of voyages that can be made by each ship v and call it A_v. We also define the set of loading nodes that can be visited by ship v as $N_v^P = N^P \cap N_v$, and the set of unloading nodes that can be visited by ship v as $N_v^D = N^D \cap N_v$.

Having defined all necessary sets, the next step is to identify what choices are available with which we can search for optimality. For each cargo i we define the binary variable y_i that is equal to 1 when this cargo is picked up by a spot charter in the industrial case or not carried in the tramp case, and 0 otherwise. For each ship v we decide on binary variables x_{ijv} that are equal to 1 when the ship sails directly from node i to node j, and 0 otherwise. The cost incurred by not having cargo i shipped by your own fleet is given as C_i^{Spot}. The cost of sailing from port i to port j with ship v is given as C_{ijv}, and the associated travel time is T_{ijv}. Each node has a time window $[\underline{T_i}, \overline{T_i}]$ in which cargoes can be picked up or delivered. This time at which service starts for ship v at node i is t_{iv} and the load onboard after completing service is l_{iv}. Given these definitions, the mathematical formulation as given in Hemmati et al. (2014) becomes as follows:

$$\min \sum_{v \in V} \sum_{(i,j) \in A_v} C_{ijv} x_{ijv} + \sum_{i \in N^P} C_i^{Spot} y_i \tag{1}$$

subject to

$$\sum_{v \in V} \sum_{j \in N_v} x_{ijv} + y_i = 1, \qquad i \in N^P, \tag{2}$$

$$\sum_{j \in N_v} x_{o(v)jv} = 1, \qquad v \in V, \tag{3}$$

$$\sum_{j \in N_v} x_{ijv} - \sum_{j \in N_v} x_{jiv} = 0, \qquad v \in V, i \in N_v \backslash \{o(v), d(v)\}, \tag{4}$$

$$\sum_{j \in N_v} x_{jd(v)v} = 1, \qquad v \in V, \tag{5}$$

$$l_{iv} + Q_j - l_{jv} \leq K_v(1 - x_{ijv}), \qquad v \in V, j \in N_v^P, (i,j) \in A_v, \tag{6}$$

$$l_{iv} - Q_j - l_{(n+j)v} \leq K_v(1 - x_{i(j+n)v}), \qquad v \in V, j \in N_v^P, (i, n+j) \in A_v, \tag{7}$$

$$0 \leq l_{iv} \leq K_v, \qquad v \in V, i \in N_v^P, \tag{8}$$

$$t_{iv} + T_{ijv} - t_{jv} \leq (\overline{T_i} + T_{ijv})(1 - x_{ijv}), \qquad v \in V, (i,j) \in A_v, \tag{9}$$

$$\sum_{j \in N_v} x_{ijv} - \sum_{j \in N_v} x_{(n+i)jv} = 0, \qquad v \in V, i \in N_v^P, \tag{10}$$

$$t_{iv} + T_{i(n+i)v} - t_{(n+i)v} \leq 0, \qquad v \in V, i \in N_v^P, \tag{11}$$

$$\underline{T_i} \leq t_{iv} \leq \overline{T_i}, \qquad v \in V, i \in N_v, \tag{12}$$

$$y_i \in \{0, 1\}, \qquad i \in N^C, \tag{13}$$

$$x_{ijv} \in \{0, 1\}, \qquad v \in V, (i,j) \in A_v. \tag{14}$$

This formulation is given to the solver, which tries to minimize the cost function (1) whilst making sure all other restrictions are upheld. Restrictions (2) makes sure that every cargo is either picked up once or handled by a charter. Restrictions (3), (4) and (5) form the flow of each ship, making sure that each ship leaves its origin once, leaves a port as often as it arrives at that port, and also reaches its final destination once. The load on board of each ship is regulated by restrictions (6) and (7) for loading and unloading ports respectively, and restrictions (8) ensure for each ship that the load on board is never more than the capacity of that ship. Restrictions (9) make sure that service times are possible with respect to travel times between ports. Restrictions (10) ensure that a cargo that is picked up by a certain ship is also delivered by that same ship. Cargoes cannot be delivered before they are picked up because of restrictions (11), and restrictions (12) give time windows for service times. Finally, restrictions (13) and (14) give binary meaning to decision variables x_{ijv} and y_i.

4.2 Mathematical Formulation with Varying Speeds

With the addition of speed tiers, the problem formulation becomes slightly more complex. Speed tiers are defined as $s \in S$, where $S = \{1, 2, 3, 4, 5\}$. A new decision variable z_{ijvs} is introduced that is equal to 1 when ship v sails directly from node i to node j at speed tier s, and zero otherwise. The cost of sailing from port i to port j with ship v at speed tier s is defined as C_{ijvs} with associated travel time T_{ijvs}. With these additions, several restrictions are also modified. First of all, objective function (1) now becomes:

$$\min \sum_{v \in V} \sum_{(i,j) \in A_v} \sum_{s \in S} C_{ijvs} z_{ijvs} + \sum_{i \in N^P} C_i^{Spot} y_i. \tag{15}$$

The relation between x_{ijv} and z_{ijvs} is given as:

$$\sum_{s \in S} z_{ijvs} = x_{ijv}, \qquad v \in V, (i,j) \in A_v, \tag{16}$$

which should be added as a new restriction. Restrictions (9) and (11) become

$$t_{iv} + T_{ijvs} - t_{jv} \leq (\overline{T_i} + T_{ijvs})(1 - z_{ijvs}), \qquad v \in V, (i,j) \in A_v, s \in S, \tag{17}$$

and

$$t_{iv} + T_{i(n+i)v5} - t_{(n+i)v} \leq 0, \qquad v \in V, i \in N_v^P, \tag{18}$$

respectively. In restrictions (17) the service times are now made feasible with regards to the new speed. Restrictions (18) make sure that cargoes are picked up before being delivered, for which they now use the shortest possible travel time (with subscript 5). Finally, the new decision variables should always be binary, which gives the last additional constraint:

$$z_{ijvs} \in \{0, 1\}, \qquad v \in V, (i,j) \in A_v, s \in S. \tag{19}$$

The mathematical formulation of the problem including variable sailing speed consists of objective function (15), subject to restrictions (2)-(8), (10), (12)-(14) and (16)-(19). The problem defined in Section 4.1 is a special case of the problem defined here, in that it sets the speed tier to be tier 4, which means that the varying speeds give a relaxation of the original problem. Because of this, the objective value when solved to optimality of the varying speeds problem will always be less than or equal to that of the original problem.

4.3 Adaptive Large Neighborhood Search

The Adaptive Large Neighborhood Search (ALNS) heuristic makes use of certain semi-random combinations of insertion and removal techniques that operate on a provided feasible solution. Each iteration the algorithm removes a certain number of cargoes from the current solution and reinserts them differently. This algorithm is described in Algorithm 4.1. The algorithm takes an initial solution S which is the equivalent of having all cargoes be transported by spot charters. Solutions are represented as a sequence of integers for each ship, including one sequence for an artificial spot charter ship. The first time an integer is encountered in a sequence means that the cargo with that index has been picked up by that ship and the second time an integer is encountered in a sequence means that the cargo with that index has been delivered by that ship.

Algorithm 4.1: ALNS(S)

Input: S, initial feasible solution

Output: S_{best}, final solution

```
S_best ← S
while Stopping criteria are not met
  { S' ← S
  | select removal and insertion heuristics based on search parameters
  | select the number of cargoes to remove and insert, q
  | remove q cargoes from S'
  | reinsert removed cargoes into S'
  | if f(S') < f(S_best)
  |   then S_best ← S'
  |
  |   else if S' gets accepted by acceptance criterion
  |   then S ← S'
  { update search parameters
return (S_best)
```

In each iteration in Algorithm 4.1, q cargoes are removed from the current solution and subsequently inserted to form a new (but not necessarily different) solution. These cargoes are removed and inserted using removal and insertion heuristics, respectively. If the new-found solution is better than the previous solution, the algorithm starts a new iteration with the new solution. When the new solution is not better, however, it might also be accepted in order to visit new neighborhoods in search of better solutions. Insertion and removal heuristics are randomly chosen each iteration using adaptive weights based on their performance. Every set number of iterations, in our case 100, these weights are redetermined. We will describe all these methods in more detail in the following paragraphs.

Removal Heuristics Each iteration, cargoes are removed using one of three methods. First of all, the number of cargoes that need to be removed needs to be determined. For this we choose randomly q in $[4, min(100, \xi n]$, where n is the total number of cargoes and ξ is a constant parameter which we set at 0.85. If this parameter ξ is too large, an entire solution might be removed. Is it too small, however, iterations would have little effect. The randomness ensures that the number of cargoes that are removed in each iteration varies, which gives us different neighborhood sizes. The q cargoes are selected by means of one of three heuristics. We employ the Shaw removal, the random removal, and the worst removal heuristics.

The Shaw removal heuristic makes use of the relatedness of cargoes. It selects one cargo at random and removes that cargo, along with its $q-1$ closest related other cargoes. This relatedness is defined based on proximity, on relatedness in delivery time limit, on size, and based on the vessels that can be used to transport the cargoes. The full formulation is as follows:

$$R(i,j) = \phi(d_{A(i),A(j)} + d_{B(i),B(j)}) + \chi\big(|T_{A(i)} - T_{A(j)}| + |T_{B(i)} - T_{B(j)}|\big)$$
$$+\psi(Q_i - Q_j) + \omega\Big(1 - \frac{|K_i \cap K_j|}{min\{|K-I|, |K_j|\}}\Big).$$

The pickup and delivery locations of a cargo i are denoted by $A(i)$ and $B(i)$ respectively. The traveling costs between two locations i and j is given as $d_{i,j}$, the last moment at which a location i can be visited is denoted by T_i and Q_i describes the size of cargo i. The set of vessels that are capable of transporting cargo i is given as K_i.

The parameters ϕ, χ, ψ and ω each give a weight to each of the measures of relatedness. For the parameters we take the following values that focus heavily on removing cargoes on the same ship and that are near one another, and less so on size and time windows: $\phi = 0.75$, $\chi = 0.1$, $\psi = 0.1$, and $\omega = 1$ (Shaw, 1997).

The random removal heuristic is very simple in that it removes q randomly chosen cargoes from the solution.

The worst removal heuristic selects cargoes that are the largest burden on costs. It calculates for each cargo what the saving would be if it was removed from the solution completely, and does so every time the most expensive cargo is removed until q cargoes are removed from the solution.

Insertion Heuristics The cargoes that were removed also need to be reinserted. For this we employ two insertion heuristics, a greedy heuristic and a regret-k heuristic.

The basic greedy heuristic tries to find the cheapest feasible way in which any of the cargoes can be inserted in the solution. This cheapest cargo is inserted in this cheapest way, and the heuristic searches for the next cheapest feasible way to insert a cargo. This is continued until all cargoes are inserted. Note that cargoes can also be picked up by a spot charter when no other feasible insertions are available.

The regret-k heuristic calculates the so-called regret value for each cargo. This regret value is the sum of inserting this cargo in its best location, in its second best location and so on until the k-th best location. This regret value indicates how hard it is to fit this cargo into the solution. Cargoes that are harder to fit have a higher regret value and the cargo with the highest regret value is inserted at its best position. These regret values are calculated for the new solution and the process is repeated until all cargoes are represented in the solution. Each time this heuristic is called, the value of parameter k is chosen randomly in $[2, 4]$ (Hemmati & Hvattum, 2017). This randomness helps ensure that multiple neighborhoods for the solution are visited.

Adaptive Weight Adjustment During the ALNS heuristic, scores are being awarded to each of the heuristics employed. Whenever a heuristic is chosen and the solution is accepted by either being better than the previous solution or because of the acceptance criterion, their score is increased. Whenever a heuristic is selected, it also gets an increase in its appearance counter. After each set of 100 iterations the new weights are calculated and the score and appearance counters are reset. The weight for the n-th set of iterations for heuristic i is calculated as follows:

$$w_{i,n} = w_{i,n-1}(1 - r) + r\frac{\pi_i}{\theta_i},$$

where π_i and θ_i denote the score and the appearance counter, respectively, for heuristic i. The parameter r influences how much of the new weight should be based on the old weight. In our case, r was set conservatively at $r = 0.8$. In the first set of iterations all weights are equal.

Heuristic Selection With the weights previously assigned, the ALNS procedure picks two heuristics independently in each iteration. These are selected using a roulette wheel based on the weights of that type of heuristic. For example, the probability of selecting the Shaw removal heuristic is

$$p_{shaw} = \frac{w_{shaw}}{w_{shaw} + w_{worst} + w_{random}}.$$

This is done similarly for all five heuristics.

Acceptance and Stopping Criteria A new solution constructed as a result of the removal and insertion procedures is not always accepted. Solutions that are better than their predecessor are always accepted, and solutions that are worse are accepted with probability $e^{-|f-f_{new}|/T}$. Here $T > 0$ is the so-called temperature, which is implemented as it is given in (Crama & Schyns, 2003). The ALNS procedure is terminated as it reaches 3000 iterations.

With all three approaches thoroughly defined, the next section performs the computational study in which the approaches are compared to one another.

5 Computational Study

The computational study consists of two parts. The first part is where the approaches are evaluated for each of the benchmark instances, and the second part is where the solutions of the extension with varying speed are further examined.

5.1 Approach Performance Comparison

This paper aims to compare the performance of the different approaches presented in Section 4 to the ship routing and scheduling problem, for which comparison by means of a computational study is of paramount importance. We solve the MIP formulations of the problem with fixed speed and of the problem with varying speeds using CPLEX 12.6.3 in Java 8. The ALNS heuristic is also executed in Java 8, and consisted of a maximum of 3000 iterations. The systems on which the approaches were tested were identical and used a 3.5 GHz GPU with 16.0 GB of RAM and a 64-bit operating system. For each of the approaches we allowed a computation time of one hour. This hour included all the time required to initialize the program, with tasks such as reading data and presolving amongst others. In the following tables the best known results for each of the methods are presented as their objective values, along with the time at which these results were found. It should be noted that these results are not always the optimal values, but are the best results that were found in the allotted time.

Not all benchmarks that were described in Section 3.1 were actually used. Instead, only the first instance of each problem size was used in order to cut down on the total computational time. This choice is justified by the fact that the computational study of earlier research showed limited difference in results between the different instances (Hemmati, Hvattum, Fagerholt, & Norstad, 2014). In addition to this, several of the larger problem instances failed to produce any results, either not finishing the presolve phase in CPLEX or not finishing the initialization of the ALNS heuristic. Because of this, all of the larger instances were not included in the tables as none of them produced any results. These larger instances are 100/30 and 130/40 for mixed load cargoes and 90/40 and 100/50 for full load cargoes, where a/b denotes an instance with a cargoes and b ships. Some of the results of the other larger problems are presented as a dash (-), indicating that no results were found in the available time. The best results for each instance are printed in bold.

		Fixed Speed		**ALNS**		**Varying Speeds**	
# of Cargoes	# of Ships	Objective Value	Running Time	Objective Value	Running Time	Objective Value	Running Time
7	3	1,476,444	0.48s	1,477,429	0.13s	**1,319,166**	0.22s
10	3	2,083,976	0.56s	2,083,976	0.34s	**1,811,614**	2.39s
15	4	1,959,153	348.79s	1,959,153	0.84s	**1,589,154**	392.25s
18	5	2,374,420	2206.18s	2,374,420	15.57s	**2,170,525**	3570.52s
22	6	4,008,619	734.59s	4,501,362	29.18s	**3,475,420**	3399.39s
23	13	2,420,298	2723.70s	**2,277,598**	1.98s	5,046,022	2523.69s
30	6	6,796,334	1945.89s	**5,261,561**	25.70s	11,152,400	3292.17s
35	7	6,614,270	3170.35s	**5,096,837**	335.33s	15,251,900	1447.18s
60	13	38,147,200	1067.45s	**8,570,796**	3549.03s	35,285,800	2658.12s
80	20	46,770,300	1046.47s	**11,642,187**	2855.82s	-	-

Table 2: Best found results for short sea shipping instances with mixed cargo sizes

Table 2 presents the best found values when solving each short sea mixed load instance using each of the three approaches. In the smaller instances, the third approach solves to optimality and is therefore better than the other two approaches. Even when it does not solve to optimality, it still outperforms the other approaches. However, as problems get larger, the ALNS often finds a better solution more quickly than the other two.
In Table 3, the best found results for each short sea full load instance for each approach are given. In all of the smaller instances, the varying speeds approach gives the best results despite not solving to optimality. As the instances get larger, the ALNS approach becomes more effective.

Tables 2 and 3 covered the short sea problem instances, which consists of

# of Cargoes	# of Ships	Fixed Speed Objective Value	Fixed Speed Running Time	ALNS Objective Value	ALNS Running Time	Varying Speeds Objective Value	Varying Speeds Running Time
8	3	1,391,997	0.34s	1,391,997	0.34s	**1,175,972**	4.54s
11	4	1,052,463	2.51s	1,052,463	0.49s	**909,439**	25.36s
13	5	2,034,184	4.38s	2,034,184	8.12s	**1,587,865**	261.66s
16	6	3,577,005	12.40s	3,612,552	3.00s	**3,039,096**	324.67s
17	13	2,265,731	735.44s	2,268,009	8.25s	**1,861,184**	3342.57s
20	6	2,991,891	1322.84s	2,973,381	20.04s	**2,503,918**	3124.11s
25	7	3,910,308	3199.72s	**3,900,951**	15.96s	4,462,483	2976.56s
35	13	3,623,460	2342.96s	**3,010,548**	257.82s	5,490,537	2335.51s
50	20	11,493,500	2327.25s	**7,370,100**	2688.74s	17,058,700	1932,18
70	30	23,431,100	1972.52s	**10,965,966**	1270.98s	-	-

Table 3: Best found results for short sea shipping instances with full load cargoes

many ports close to one another. As a result, a relatively large part of the travel times and costs is made up of harbor costs and loading times. This means that the saving made possible by allowing varying speeds is expected to be smaller in the short sea than in the deep sea instances. The results of the deep sea instances will be covered in Tables 4 and 5.

# of Cargoes	# of Ships	Fixed Speed Objective Value	Fixed Speed Running Time	ALNS Objective Value	ALNS Running Time	Varying Speeds Objective Value	Varying Speeds Running Time
7	3	5,233,464	0.14s	5,233,464	0.61s	**3,751,347**	1.25s
10	3	7,986,248	0.17s	9,983,848	0.87s	**6,107,111**	0.81s
15	4	13,467,090	10.98s	13,467,090	3.58s	**11,562,392**	2169.94s
18	5	43,054,055	493.21s	43,197,288	2.91s	**33,593,500**	2986.84s
22	6	41,176,986	192.79s	41,176,986	9.76s	**33,182,200**	3569.80s
23	13	41,002,992	3241.90s	41,599,591	29.90s	**36,352,200**	2167.95s
30	6	20,212,000	3414.41s	19,229,713	986.32s	**17,352,000**	3539.43s
35	7	106,329,000	2197.77s	**65,923,641**	1136.86s	111,797,000	1059.15s
60	13	357,600,000	1400.84s	**89,158,268**	1815.15s	331,270,000	2804.86s
80	20	340,682,000	3413.12s	**83,094,686**	1268.28s	340,682,000	3072.83s

Table 4: Best found results for deep sea shipping instances with mixed load cargoes

The best found results in Table 4 confirm what we have seen in earlier instances, which is that the approach with varying speeds is better for smaller instances but that ALNS outperforms all other approaches once the instances get large enough.

		Fixed Speed		**ALNS**		**Varying Speeds**	
# of Cargoes	# of Ships	Objective Value	Running Time	Objective Value	Running Time	Objective Value	Running Time
8	3	9,584,863	0.05s	9,816,881	0.66s	**6,279,584**	0.17s
11	4	34,854,819	0.16s	34,854,819	5.77s	**27,957,396**	5.35s
13	5	11,629,005	0.33s	12,490,851	2.00s	**9,013,226**	42.31s
16	6	51,127,590	4.57s	51,127,590	2.66s	**41,310,188**	512.17s
17	13	17,316,720	2.79s	17,316,720	22.26s	**15,147,150**	1331.58s
20	6	16,406,738	7.97s	16,406,738	90.88s	**12,437,029**	495.71s
25	7	22,773,158	32.62s	22,811,855	315.83s	**17,508,100**	2003.80s
35	13	88,853,100	2671.10s	88,620,858	2196.04s	**72,376,000**	944.125s
50	20	43,088,500	1791.27s	42,432,039	2748.42s	**37,368,700**	3281.91s
70	30	**150,829,000**	3423.88s	152,507,132	1557.88s	159,016,000	3431.61s

Table 5: Best found results for deep sea shipping instances with full load cargoes

The results in Table 5 once again that the model with varying speeds can also outperform ALNS even when the solver does not manage to reach optimality. Only for the largest instance the fixed speed approach produced the best result, albeit with slim margins.

From these tables we find that the approach with varying speeds is always better when solved to optimality. This should not come as a surprise, as it is a relaxation of the definition of the problem of the other two approaches. For instances up to 20 cargoes and sometimes even larger, the varying speed approach outperforms the other two approaches even when it does not solve to optimality. It is interesting to see that the varying speed approach seems to perform better in the deep sea instances and the full load cargoes, rather than in short sea instances and mixed load cargoes. For larger instances, the ALNS approach produces the best solutions. It should be noted that the ALNS generally finds its best solution much quicker than the model with varying speeds. However, as this computational study was set up with a computation time of an hour, the best result found is more important than the time it took to reach that solution.

5.2 Speed Tier Analysis

For the sake of possible further research, it is of interest to know how the new possibilities that come with the relaxation provided by the model with varying speeds are utilized. If, for example, only the highest speed tier is utilized, it might mean that the penalty for a higher speed is too low compared to its decreased travel time. To this end, we collected the data with regards to choice of speed for the instances that managed to solve to optimality, which are presented in Table 6.

	# of	# of	**Speed**	**Tier**			
Instance type	Cargoes	Vessels	1	2	3	4	5
Short sea, mixed cargo	7	3	11	1	0	0	1
Short sea, mixed cargo	10	3	15	0	1	1	4
Deep sea, mixed cargo	7	3	14	0	0	0	3
Deep sea, mixed cargo	10	3	18	0	0	0	3
Short sea, full cargo	8	3	10	3	0	0	4
Short sea, full cargo	11	4	17	4	3	2	0
Short sea, full cargo	13	5	20	1	2	2	6
Deep sea, full cargo	8	3	11	2	0	0	4
Deep sea, full cargo	11	4	17	0	1	0	4
Deep sea, full cargo	13	5	19	2	2	2	4
		Total	79	8	5	4	16

Table 6: Number of times each speed tier was used in optimal solutions

From this table, we learn that the lowest speed tier is used most often. The next most commonly used speed tier is the highest tier, and the three middle tiers are used approximately the same. From this we can conclude that lower speeds are very popular for reducing costs on voyages that allow it and that higher speeds are sometimes used to squeeze in a cargo that would otherwise be picked up by a spot charter. While these results seem promising for at least some further research, it should be noted that the cost structure used is by no means exact but rather an approximation based on assumptions on the current cost structure and on vessel capabilities.

6 Conclusion

This paper reacts to a growing market for global maritime transportation, in addition to a larger concern for ecological well-being in all industries. It does so by considering the cargo ship routing and scheduling problem as it occurs in industrial and tramp shipping. Three different approaches were taken in order to solve these problems, and they were compared with regards to performance in objective value and solving speed. For a fair comparison, this paper made use of benchmark instances presented in earlier research in this field.

The first approach to the routing and scheduling problem was solving a mathematical formulation with a solver. Expanding on this formulation, the second approach was presented which extended upon the original formulation by introducing different speed tiers at which voyages could be made. This second approach solves a new extended mathematical formulation using the same solver as the first approach. The third approach made use of an adaptive large neighborhood search (ALNS) heuristic.

The benchmark instances have been solved by all three approaches, and the best solutions found within a certain time limit have been recorded. In cases where the second approach reached optimality, it outperformed the other two approaches. However, this approach also produced the best results when optimality was not reached for instances up to 20 cargoes and sometimes more. The varying speed approach remained superior over the other approaches for longer as instances became larger in the case of deep sea instances and of instances with full load when compared to the cases of their counterparts. For larger instances, the ALNS outperformed or matched the other approaches consistently.

The addition of speed tiers created a more complex, but also more realistic and more rewarding problem. Optimal solutions for this problem often included very low and high speeds, which - in addition to the reduced objective values - indicates that this extension is not trivial. Solving the cargo ship routing and scheduling problem with the help of variable sailing speeds helps reduce fuel consumption without sacrificing profit, and could prove to be a lucrative consideration in the future of maritime transportation.

7 Recommendations

During this research, several points of interest arose that would warrant further research. One of the limitations of this research is that the cost structure of the voyages and their speed tiers is built on several assumptions. One of these assumptions is that the costs incurred on a voyage solely consist of fuel costs and harbor fees. In reality, other factors might be at play that could be further researched with the help of actual shipping companies. Another assumption is that the proposed tiers with their associated percentages increase in costs and speed are identical for every type of ship. Both of these assumptions are made in this research in order to investigate whether the addition of speed tiers would be interesting, and now that it has proven to be so, these assumptions could themselves be further researched.

Another promising opportunity for further research lies in implementing an ALNS for the model with varying speeds. The ALNS approach used in this paper already produces better results than the solver of the same model, which means that an ALNS heuristic for the extended model could prove to be superior to the approach taken in this research.

In summary, adding a choice of speed to the existing routing and scheduling problem brings about lower fuel consumption and higher profits for the maritime shipping industry. This effect could be enhanced by creating a more realistic structure between sailing speed and costs based on empirical data from shipping companies. In addition, clever (ALNS) heuristics could be devised for this extension and tailored to the addition of speed tiers, further unlocking the potential of this new-found flexibility.

References

Crama, Y., & Schyns, M. (2003). Simulated annealing for complex portfolio selection problems. *European Journal of operational research*, *150*(3), 546–571.

Hemmati, A., & Hvattum, L. M. (2017). Evaluating the importance of randomization in adaptive large neighborhood search. *International Transactions in Operational Research*, *24*(5), 929–942.

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.

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

Malliappi, F., Bennell, J. A., & Potts, C. N. (2011). A variable neighborhood search heuristic for tramp ship scheduling. In *International conference on computational logistics* (pp. 273–285).

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.

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*.