A Ship Routing Problem with Hold and Draft Constraints

Charity Jenkins

5th June 2009

Honours Thesis

Supervisors: Natashia Boland
Heng-Soon Gan
Abstract

Maritime transportation is an important industry, with seaborne trade estimated to be up to 85% in terms of weight of total international trade in 2007.

Capital investment for ships is substantial, and the operating costs of a ship may range from thousands to tens of thousands of US dollars daily, thus the potential benefits of optimized approaches to fleet planning are significant.

Despite this, there has been relatively sparse attention paid in the literature to maritime transportation in comparison to other forms of transportation such aircraft and road.

We consider a single time-period ship routing problem with the following properties: a heterogeneous fleet, draft constraints, loading and discharging ports with pre-specified amounts of supply and demand, multiple material types, and capacity constrained holds in which different material types cannot be mixed. Our objective is to minimize the shipping costs while meeting the specified demands.

We review a mixed integer programming model for the problem, and discuss some improvement techniques that have been implemented. We also investigate an attempt at reformulating the model, to address some inherent symmetry, using decomposition techniques: a combination of Benders Decomposition and Dantzig-Wolfe Decomposition with column generation.

We find the best results for improving our original formulation are produced by a combination of methods including symmetry breaking and various types of bounding. This combination of methods performed faster than the original formulation in all attempted problem sets where it solved to completion.

The performance results for the investigated decomposition reformulation are at this stage inconclusive, and some ideas for further research are suggested.
Acknowledgements

I have many people to thank for helping me along the way through my work on this project.

Firstly, I would like to thank Heng-Soon Gan for being so fantastically supportive, for always making time to meet with me, and for giving me much advice and instruction on many aspects of the project. I will remember well our discussions; that he listened thoughtfully to all of my ideas, and never once led me to feel they were silly, improved my confidence in my work immeasurably.

To Natashia Boland, for taking the time out of her hectic schedule to devise so interesting a project for me, a student who stubbornly resisted working in any research area other than hers, I am very grateful. Her initial encouragement that I attend our research group seminars was also most valuable; not only in expanding my knowledge of Integer Programming and related subjects, but, perhaps most importantly, in making me feel a part of the group.

I would also like to thank Olivia Smith and Sophie Dickson for giving such very helpful advice and suggestions for writing this thesis. And Kerem Akartunali, for teaching me so many methods for improving MIP problems; I only wish I’d found the time to implement them all.

More generally, I would like to thank Melbourne Operations Research (MORe) for allowing me to share such an fascinating problem with them; in particular, Natashia Boland, Heng-Soon Gan and Olivia Smith for helping me to understand the problem and explaining their initial MIP framework.

Finally, I wish to thank Stuart, for introducing me to Operations Research in the first place, and for suffering me to once again disrupt our lives by returning to study.
Contents

1 Introduction ................................................. 5
2 Background on solution methodology .......................... 6
   2.1 Mixed Integer Programming (MIP) ...................... 7
   2.2 Branch-and-Bound (B&B) .............................. 8
   2.3 Column generation (CG) ................................. 10
   2.4 Benders Decomposition ................................. 13
3 Literature review ........................................... 14
4 Problem formulation .......................................... 17
   4.1 Description of problem .................................... 17
   4.2 Notation .................................................. 19
   4.3 MIP formulation .......................................... 20
5 Issues, improvements and results ............................. 24
   5.1 Symmetry breaking:⟨sb⟩,⟨sbh⟩ ............................. 24
   5.2 Limiting the creation of trip variables:⟨tsd⟩,⟨thw⟩ ............... 26
   5.3 Lower bound on number of trips:⟨lbmh⟩,⟨lbnw⟩ ...................... 30
   5.4 Variable fixing:⟨fix⟩ ....................................... 32
   5.5 Discussion of results ....................................... 33
6 Decomposing the problem .................................... 34
   6.1 Motivation for approach ................................... 35
   6.2 Notation .................................................. 35
   6.3 Master problem formulation ............................... 36
   6.4 Column generation formulation .......................... 39
   6.5 Benders Decomposition formulation ..................... 40
7 Reformulation: issues, improvements and results ............ 42
   7.1 Ordering of master and subproblems ..................... 42
   7.2 Feasible route generation .................................. 42
   7.3 Cycle elimination in CG subproblem ..................... 44
   7.4 Bounding η in the master MIP:⟨ηbound⟩ .................... 45
   7.5 Fixing in the master MIP:⟨rfix⟩ ......................... 46
   7.6 Discussion of results ....................................... 46
8 Conclusion ................................................... 48
Bibliography .................................................... 49
A List of Figures ............................................... 54
B List of Tables .................................................. 54
C Computational results ....................................... 54
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>C.1 Results of MIP improvement methods</td>
<td>55</td>
</tr>
<tr>
<td>C.2 Results of decomposition approach</td>
<td>64</td>
</tr>
<tr>
<td><strong>D Revised reformulation for $\eta$ bound</strong></td>
<td>67</td>
</tr>
<tr>
<td>D.1 Notation</td>
<td>67</td>
</tr>
<tr>
<td>D.2 Revised MIP reformulation</td>
<td>67</td>
</tr>
</tbody>
</table>
1 Introduction

Maritime transportation today is an important industry, with seaborne trade estimated to be from 65% to 85% in terms of weight of total international trade in 2007 [15]. Between 1980 and 2007, such trade has increased in terms of weight by 67% [15].

Since shipping is, by a considerable degree, the most efficient way of transporting goods in terms of energy consumption per unit transported, the current global emphasis on carbon emission reduction highlights further the importance of the maritime transportation industry [14].

Capital investment and operating costs for ships are substantial; the operating costs of a ship may range from thousands to tens of thousands of US dollars daily [15].

It is clear, then, that the transportation industry and the population in general benefit significantly from optimized approaches to maritime fleet planning: minimizing the costs (including overall fuel consumption, and therefore also emission of pollutants) of maritime transportation.

Although transportation planning has been discussed extensively in the literature, there has been relatively sparse attention paid to maritime transportation in comparison to aircraft and road transportation [15]. Given the relative importance of maritime fleet planning, further research into optimization in this field is certainly beneficial.

The problem considered in this thesis was motivated by the requirements of a manufacturing company for the bulk import of raw materials to Australia from overseas. This company sought the assistance of the team of researchers at Melbourne Operations Research (MORe), a consulting practice based at the University of Melbourne.

The manufacturing company desired an optimal schedule of routes for their ships to take, minimizing the total shipment cost while satisfying constraints they had specified. One example of their constraints is that of the draft limit, in which a ship may not dock at a port if its weight exceeds the limit set by that port. The minimization of costs instead of the maximization of profit is typical for a shipping subsidiary, where most revenue is generated through other parts of the company [47].

A schedule solution should include which ships to use, how much of each of different types of material to buy and then load from ports overseas, how much then to discharge at the ports in Australia, which ports each ship will visit and in what order (i.e. what routes they will travel), and when these transactions should be made.

An additional complication to this problem is brought about by the introduction of hold or compartment constraints for each ship. The company’s fleet comprises ships of two distinct types, differing by their size and hold configuration. Each type of ship has multiple holds, inside which different types of raw material cannot be mixed; that is, there must be no more than one type of raw material stored in each hold. These hold constraints significantly increase the problem complexity, as any ship may carry multiple types of raw materials in neighbouring holds on the same trip.

Due to this complexity in solving the problem, the researchers at MORe simplified it by dividing it into three stages. The first stage manages supply and demand, allocating for each one of a
set of two-week time-periods amounts of raw material to be loaded from ports overseas, and the corresponding amounts to be discharged at the ports in Australia. In the second stage, with the amounts to load and discharge for each port now known exactly, feasible and promising routes are determined by solving a single time-period problem over different combinations of aggregated time-periods. The routes generated in the second stage are then used to solve the entire scheduling problem in the third stage.

This thesis considers only the second stage of the divided planning problem, and thus from this point forward all mention of “the problem” will refer only to this second stage: a single time-period ship-routing problem with hold and draft constraints, having a heterogeneous fleet and predetermined amounts of one or more different raw materials to be loaded and discharged at specific loading and discharging ports.

The problem may be considered as an extension to the Vehicle Routing Problem (VRP), in which one finds the optimal set of routes for a homogeneous fleet of vehicles to deliver goods to customers, with each vehicle starting and ending at its own depot. The VRP is itself an extension to the NP-complete Traveling Salesman Problem (TSP), and so the problem at hand, by extending the VRP, is also NP-complete [39]; this means there is not known to be an algorithm that can solve it in polynomial time [18]. Since 1959, when Dantzig and Ramser introduced the VRP with their truck dispatching problem [21], many variants of the VRP, for example the Split-Delivery VRP (SDVRP) and the VRP with Time Windows (VRPTW), have been studied and advanced extensively with increasingly faster algorithms and better models [35]. However, the problem at hand complicates the VRP significantly; indeed, before even taking the hold and draft constraints into account, it could be defined in such lengthy terms as a Multi-Commodity Multi-Vehicle Split-Delivery Heterogeneous Fleet VRP.

The problem was originally formulated by the research team at MORe as a mixed integer programming problem (MIP) using the arc-based formulation outlined in Section 4. The work by Boland, Gan and Smith on this second stage problem was presented at the ASOR conference in Melbourne in December 2007 [8], and their work on the full three-stage problem was presented at the INFORMS meeting in Washington in October 2008 [32].

The outline of this thesis is as follows: In Section 2, we provide a background of solution methods related to this work. Relevant literature is then reviewed in Section 3. In Section 4, the original MIP problem formulation is defined. We investigate in Section 5 a variety of MIP reformulations and heuristic improvement methods that aim to improve the speed with which good quality solutions can be found. The performances of these methods are compared by testing on data provided by the manufacturing company. In Sections 6 and 7, we investigate some initial work on a path-based reformulation of this problem using decomposition techniques. We conclude in Section 8 with a short summary of results and suggestions for further work.

2 Background on solution methodology

A range of techniques are available for searching the feasible solution space of combinatorial problems for optimal solutions, and these can be categorized as either exact methods or heuristic
methods. An exact method is one that is guaranteed to find an optimal solution, provided feasible solutions exist, and will prove the optimality of the solution found [49]. A disadvantage with these methods, however, and hence a reason that heuristic algorithms are considered for combinatorial optimization problems, despite having no guarantee of optimality, is that their solution times can increase dramatically in proportion to the problem size [49].

In this thesis we focus on the use of mixed integer programming (MIP), an exact algorithm, and related techniques including heuristic improvement methods to solve the problem at hand.

### 2.1 Mixed Integer Programming (MIP)

MIP is based on Linear Programming (LP), a method for solving optimization problems whose feasible solution space can be defined by linear equations as a convex polytope. An LP is modelled by a linear objective function we want to minimize (e.g. representing costs) or to maximize, along with a set of linear equations and linear inequalities collectively known as constraints that define the solution space polytope.

The standard form for an LP in which we seek to minimize the objective function (known as a minimization problem) is as follows, having the constraints defined as greater-than-or-equal-to type inequalities:

$$
\min_{\mathbf{x}} \quad \mathbf{c}^T \mathbf{x} \\
\text{s.t.} \quad \mathbf{A} \mathbf{x} \geq \mathbf{b}
$$

where \( \mathbf{x} \) is a vector of decision variables (those values we may modify to improve the solution), \( \mathbf{c} \) is a cost coefficient vector, \( \mathbf{A} \) is a matrix of coefficients in the constraint set, and \( \mathbf{b} \) is a vector of bounding values.

The term s.t. stands for “subject to”, as the objective function \( \mathbf{c}^T \mathbf{x} \) is minimized subject to the set of constraints \( \mathbf{A} \mathbf{x} \geq \mathbf{b} \).

It is assumed here and throughout the remainder of this thesis that all decision variables are required to be non-negative, that is, the constraint \( \mathbf{x} \geq \mathbf{0} \) is implied for the decision vector \( \mathbf{x} \) for all MIP and LP models.

An LP can be solved using the Simplex algorithm, first proposed by George Dantzig in 1947 in [19] following his work with the military during World War II [20]. The simplex algorithm can require an exponential number of steps to solve an LP in its worst-case [42], however, in practice it rarely reaches this number and is generally very efficient.

The Simplex algorithm solves an LP by searching along the edges of the convex polytope defining the space of feasible solutions. It is sufficient to search only the boundary for a solution, because it is known that the optimal solution will correspond to either a unique extreme point or an extreme ray (entire edge) of the LP’s feasible region [55].

Another efficient algorithm, known as Karmarkar’s algorithm, was developed in 1984 and proven to solve LP problems in polynomial-time using an interior-point method [41]. An interior-point method is one that searches from the interior of the feasible solution space rather than along
the boundary. The methods discussed in this thesis, however, such as column generation and the Benders Decomposition approach, are based on the Simplex algorithm.

MIP is the extension of LP to integer-restricted variables; it began in 1958 with the work of R. E. Gomory in [36]. Gomory developed a system for generating “cutting planes”, constraints to prohibit fractional solutions to the integer-restricted variables. The convex polytope defined by cutting off these fractional solutions is known as the convex hull of the integer-feasible region.

A MIP does not immediately find its solution by solving a corresponding LP with integrality requirements removed (such an LP is known as the LP-relaxation for a MIP). It is often not suitable to simply round the LP-relaxation solution values to their nearest integer values; to imagine rounding these values in a scenario for which many integer-restricted variables are also restricted to being binary (i.e. $0 \leq x \leq 1$) may help to make this clear. Indeed, in general, the greater the absolute value the decision variables are likely to take, and the fewer of these variables there are, the more similar the LP-relaxation solution becomes to the MIP solution. Refer to [55], pp. 466–467, for good examples in which rounding the variable values from an LP-relaxation does not produce an optimal solution for the corresponding MIP.

It is required, then, to use a method such as cutting planes to restrict the solution to integer values, but this is not necessarily simple. In fact, the problem of working from the LP-relaxation solution toward the MIP solution, or simply the problem of solving a MIP, using any method is NP-hard [54].

2.2 Branch-and-Bound (B&B)

B&B was developed by Land and Doig in 1960 [43]. The technique can be described as a tree of scenarios, called nodes, where one node branches into two child nodes by considering the two sides of a “what-if” question, namely, what if for a decision variable $x$, having a fractional
LP-relaxation solution $a$, the optimal MIP solution has $x \leq \lfloor a \rfloor$, or, what if $x \geq \lceil a \rceil$. We solve an LP at each node, and continue branching on the two alternative options for fractional-valued decision variables until none remain. Certain of these nodes are pruned during the process by either the infeasibility of the LP under the new branching conditions or by using known bounds on the solution value. Finally, we will have just one non-pruned external node satisfying the integer requirements—that corresponding to the optimal MIP solution.

The bound on the solution value used to prune a node is given by the other nodal solutions; if the solution $z$ at a node $n$ is worse than the solution $y$ at another node $m$, where $y$ is an integer feasible solution, then no child node of $n$ can produce a solution better than $y$ (as each node is a relaxed version of its child nodes), and thus there is no use in continuing to branch on $n$, and it is pruned.

How to progress through the tree, that is, which of the nodes or branches to explore first in the tree, is not obvious. Of the main possibilities for node selection, depth-first, breadth-first and best-first, a depth-first strategy is traditionally chosen for solving MIP because it requires less node storage in memory, with the maximum number nodes to be stored at any time being the number of levels in the tree [25].

Options for variable selection at a node include choosing those variables to branch from with the most fractional or the least fractional values, or simply selecting variables based on their lexicographic ordering. Once a node and a variable have been selected, we then decide in which direction to branch, either toward the floor or the ceiling of the fractional value.

---

1. Note that $x$, as an integer, cannot be between $\lfloor a \rfloor$ and $\lceil a \rceil$, and so a fractional solution with $x = a$ is infeasible for the MIP; the optimal solution must have that either $x \leq \lfloor a \rfloor$ or $x \geq \lceil a \rceil$.

2. Nodes having integer solutions are also branched from no further, however, this is not pruning but fathoming. We fathom a node when its solution is optimal for its subtree. An integer solution may be pruned if its solution is worse than the current best bound.
The following example demonstrates the B&B tree for a small MIP:

\[
\begin{align*}
\min_x & \quad z = 3x_1 + 2x_2 - 4x_3 \\
\text{s.t.} & \quad 2x_1 + 3x_2 + 4x_3 \geq 15 \\
& \quad 2x_2 - 3x_1 - 2x_3 \geq 0 \\
& \quad -x_1 - x_2 - x_3 \geq -11 \\
& \quad x_1 - 2x_3 \geq 0 \\
& \quad x_1 \geq 1 \\
& \quad x_1, x_2, x_3 \in \mathbb{Z}^+ 
\end{align*}
\] (2.2.1)

The LP-relaxation solution \( \text{MIP}^{\text{LP}} \) for (2.2.1) is \( z = 7.5 \), \( x_1 = 1.5 \), \( x_2 = 3 \) and \( x_3 = 0.75 \). We call the node for this solution the root node in the B&B tree.

Figure 2.3 gives the B&B tree for (2.2.1), where we apply a depth first search with lexicographical variable selection, branching up toward ceiling values. The nodes are numbered by the order they are visited in the search.

We can see in Figure 2.3 that the first child node of the root node, node 2, where we branched with \( x_1 \geq 2 \), is the optimal solution to the MIP. But we do not realize this solution is optimal until after further branching from the other child node, node 3, where we branched with \( x_1 \leq 1 \). The branching from node 3 ultimately leads either to infeasible solutions (nodes 4 and 6) or to a solution that is pruned for being worse than the integer feasible solution we had already found (node 7).

### 2.3 Column generation (CG)

Column generation is a method used to solve LPs with a large number of variables, where only a relatively small subset of the decision variables will be basic, or nonzero, in the optimal solution. CG was first suggested by Ford and Fulkerson in 1958 for solving multi-commodity network flow problems [29], and was generalized and applied to LP in 1960 by Dantzig and Wolfe for use with their decomposition approach for large LPs [23].

The idea behind CG is that if only a subset of decision variables will be basic in the optimal solution, there is no need to include so many nonbasic variables in the problem at all. Starting with a reduced problem having only a few or perhaps none of the original decision variables, new variables can be generated at each iteration of the algorithm by solving a subproblem that determines which available variable is likely to improve the objective function the most, that is, the variable with the most negative reduced cost.

The reduced cost of a decision variable \( x \) is computed by multiplying the dual value\(^3\) of each constraint containing \( x \) by the coefficient of \( x \) in that constraint. The sum of these values represents the potential value to the problem of adding one unit for \( x \) to the basis; this sum is then subtracted from the cost coefficient of \( x \) in the objective function. So, for the problem in

\[^3\text{Refer to chapter 6 of [55] for an overview of duality theory.}\]
Figure 2.3: Example B&B tree for (2.2.1)
standard form (2.1.1), we have the reduced cost of \( x \):

\[
\mathbf{c} - \pi \mathbf{A}
\]  

(2.3.1)

where \( \pi \) is the vector of the dual values for each constraint. If the improvement added to a problem by the addition of one unit of a decision variable \( x \) is greater than the cost coefficient for \( x \), then \( x \) will have a negative reduced cost.

The dual value of a constraint is the value of its corresponding variable in the dual problem\(^4\), in which the LP is reformulated so that constraints correspond to variables and variables to constraints; it can be understood to indicate the amount a relaxation of that constraint by one unit will improve the objective value. The term \( \pi \mathbf{A} \) used to calculate the reduced cost (2.3.1) also can be seen as a formulation of the constraints corresponding to \( x \) in this dual problem. Every LP has a dual problem, which for a minimization problem is a maximization problem and vice-versa; these two problems share the same optimal objective value, and thus can be used to bound one another. At each iteration of CG we must solve the dual LP to obtain these values for the reduce cost calculation.

The term column generation may become clearer when thinking of the problem structure in its matrix form, in which each row represents a constraint and each column represents a decision variable. So, by generating columns, we are finding new decision variables to add to the problem. With this in mind, one could equally well describe the addition of new constraints or cutting planes to a problem as row generation.

See [24] for more information on CG, and in particular [5] for a description of CG applied to its most standard example problem, the Cutting Stock Problem.

A problem without a large number of variables may not seem immediately to lend itself to a CG approach, however, if it has a number of constraints \( Y \) that include only some subset of the variables, say \( X' \subseteq X \) where \( X \) is the set of all variables, it can be reformulated into a master problem and subproblem using Dantzig-Wolfe decomposition. The variables in \( X' \), along with the constraints \( Y \) containing only those variables, are moved to a subproblem, and the remaining variables and constraints form a master problem. Columns can be generated from the subproblem using CG, with each column representing a different combination of values for the variables in \( X' \), corresponding to an extreme point of the constraint set \( Y \) [31]. Using the cost coefficients for \( X' \) and the constraints \( Y \), it can be determined for each new variable generated its cost coefficient for the master objective function and its coefficients for each master constraint.

Unfortunately, CG cannot be used with MIP quite so easily as described here for LP, at least not if the decomposition leads to us having integer-restricted variables in our master problem, because the dual values used to calculate the reduced costs are found by solving an LP-master and not a MIP. To generate columns for a MIP, we must do so within the B&B tree in a process known as branch-and-price [4]. In branch-and-price, branching occurs when the CG subproblem

--

\(^4\)A brief description of duality theory is provided here. For a more detailed overview, the reader is referred to chapter 6 of [55].
finds no column with negative reduced cost to enter the basis of the master problem, and the master LP solution still does not satisfy the integrality conditions. This method has some inherent difficulties, as the constraints generated in the master problem during B&B must be taken into account by each of the subproblems [3].

2.4 Benders Decomposition

The Benders Decomposition, developed by J. F. Benders in 1962 [6], has much in common with the Dantzig-Wolfe Decomposition, in that a selection of variables are moved into a subproblem and the remaining variables kept in a master problem. However, in this case it is not variables we generate to add to the master problem but constraints; in essence, this is a row generation approach.

The idea behind Benders Decomposition is that if certain “complicating variables” can be temporarily fixed, when the remaining variables are not integer-restricted, the MIP problem reduces to an LP [33].

In the Benders Decomposition process, the master problem MIP is solved for its subset of the variables and constraints, and then the solutions to its complicating integer variables are treated as fixed in the Benders subproblem. If, with these values fixed, the subproblem LP is feasible, then a constraint is generated for the master problem to make it cost its solution appropriately for the Benders subproblem variables.

The master problem cannot apply a cost for the values of the Benders subproblem variables, as it does not itself know them. The constraints added iteratively by Benders Decomposition set a lower bound for a variable, $\eta$, in the master objective function, to account for the cost of the Benders subproblem variables under the condition in which the master MIP variables had been fixed. Essentially, these constraints force the solution of the master problem to be within an extreme point or an extreme ray of the solution space for the Benders subproblem variables as well as for its own variables.

After a constraint has been added by the Benders subproblem, the master problem is re-solved, as the change made by the new constraint to the objective function may mean that a better solution now exists. The process is repeated until the MIP finds no better solution even with the adjustments made by the latest Benders constraint to the objective.

At some iterations of the process, it may happen that the Benders subproblem is infeasible due to the fixed MIP variable values; in this case a constraint is generated for the master problem, not to adjust the cost, but to cut off this infeasible solution. This constraint is found by minimizing the amount, called the slack, by which each of the subproblem constraints are violated.

The cuts generated by the Benders subproblem are defined by the dual values of its constraints, so unlike with CG there is no issue in having a MIP master problem, and instead we usually require the Benders subproblem to be a pure LP.

See [22] for more information about Benders Decomposition, and more generally for information about the decomposition of large scale LP problems.
3 Literature review

As mentioned previously, research into maritime transportation planning has been but a small proportion of all transportation planning research in the literature. Christiansen et al. [15] have suggested some possible reasons for this: the low visibility of ships to the general population, including a majority of organizations moving cargo mainly by truck or rail, the lack of consistent structure in maritime planning problems, greater uncertainty in maritime operations (e.g. caused by weather conditions), and the long tradition and fragmentation of the ocean shipping industry.

They also observe, however, that research in ship fleet planning optimization has recently undergone significant growth, as evidenced by an increase in the number of references in review papers; a review of the last decade by Christiansen et al. [16] contained twice as many references as the review by Ronen [52] covering the previous decade. The reader may wish to refer to these review papers for a broader summary of research over the last two decades.

Research into problems similar to the one at hand seems to be only relatively recent, compared even with other maritime transportation research, with a number of relevant studies having been presented at conferences but related published work not yet to be found.

Maritime fleet optimization necessitates dedicated research because of how it differs from other transportation problems, such as aircraft, truck and train scheduling. Some major differences are in unit size, voyage length, port fees, and times of operation; a detailed comparison of these operational characteristics is made by Christiansen et al. [15].

That said, there are studies with similarities to our problem that are not maritime transportation problems, such as the real life VRP presented in conference by Repoussis et al. in 2006 [50], having a heterogeneous fleet with hold constraints for non-mixable products. There are, naturally, no draft constraints, port fees, nor comparable vehicle sizes and distances, but they would face similar issues with their material-to-hold assignment as in our problem. They propose a solution via a hybrid meta-heuristic that employs Greedy Randomized Adaptive Search Procedures (GRASP) [28] for diversification and Variable Neighborhood Search [37] for intensification in local search. A similar study was presented in a different conference by Muyldermans and Pang in 2007 [46] that also used local search heuristics.

Many problems in the literature, such as the tramp shipping study by Jetlund and Karimi in 2004 [39], the studies by Christiansen in 1999 [13], and by Fagerholt and Christiansen in 2000 [26], have the property that a large proportion of ship time is spent in ports—about 40% for the tramp carriers in short-sea operation [39]—instead of being spent travelling between the ports, and thus time-window constraints at ports are a major consideration. With our problem having a greater distance between ports, particularly between the discharging ports in Australia and the loading ports overseas, time-windows are not an issue to the same extent, and they have not been accounted for here.

There are three categories of operational modes for shipping: industrial, tramp and liner [44]. Liner shipping operates similarly to a bus line, having published timetables; in tramp shipping, carriers place their ships where they expect there to be available cargo, and these can be hired much the same as a taxi; with industrial shipping, the shipping company controls its own fleet
and meets its objectives at minimum cost. The mode of operation for the problem at hand is industrial shipping.

Maritime routing problems for bulk products can also be divided into cargo routing and inventory routing problems [2]. Cargo routing problems are mainly constrained by amounts of cargo specified by loading and discharging ports, as well as by time-windows for loading and discharging, whilst inventory routing problems are constrained by requirements that levels of products at ports be maintained. With the supply and demands having been set in a previous stage problem, the problem at hand is a cargo routing problem without time-window constraints. Most attempts at solving cargo routing problems apply set-partitioning to achieve a path-based approach, generating a set of candidate schedules and choosing among them [2]. We will investigate this approach for our problem in Sections 6 and 7.

In 2005, Bredström et al. [9] extended on earlier work [10, 11] to consider a problem that, like our problem, is a second stage focused on distribution to customers, in which a route is defined as a sequence of loading ports followed by a sequence of discharging ports. In contrast to our problem, they do not have non-mixable products to be loaded into holds. They use a genetic algorithm to find candidate schedules, where a schedule (chromosome) is a sequence of routes (genes) for all ships in a fixed fleet. As some quantities can also be transported by train or truck, and the schedule does not include this, an LP is used to measure a schedule’s approximate overall fitness value. They reported computational results for a 40-day planning horizon, noting that they generally find better and faster solutions using the genetic algorithm than using a two-hour B&B search.

Hennig and Nygreen, at a conference in 2008 [38], presented a split pickup-and-delivery problem for multiple commodities using a heterogeneous fleet. As with the Bredström et al. study [9], they do not have hold constraints. Their problem also differs from the problem at hand by having route definitions made more complex through not being divided into a distinct sequence of loading ports and discharging ports. As with this work, Hennig and Nygreen investigate both arc-based and path-based formulations for their problem, with the routes for their path-based formulation generated by an initial heuristic.

These studies by Bredström et al. [9] and Hennig and Nygreen [38] are similar to our problem in many ways, except they are without hold constraints; there are several other similar ship fleet planning optimization studies that do include hold constraints: Fox and Herden in 1999 [30], Al-Khayyal and Hwang in 2007 [2], Giesen et al. in 2007 [34], Jetlund and Karimi in 2004 [39], Fagerholt and Christiansen in 2000 [26], and Ronen in 2002 [53], however, these are largely inventory routing problems instead of cargo routing problems, and also generally include the consideration of time-windows.

The inventory routing problem considered by Fox and Herden [30] is another Australian-based study, with a heterogeneous fleet shipping non-mixable materials from a single loading port. Their problem covers multiple time periods, and assumes for simplification that there will be only one ship used per period. Fox and Herden apply a MIP solution approach, for which extra integer variables are introduced to create strong cuts, and integer and binary variables are initially relaxed for later periods. They report having produced good schedules within 2
hours computing time, and a significant improvement on the previously used manual solution technique.

Al-Khayyal and Hwang [2] extend an inventory routing model considered by Christiansen and Nygreen [17], adopting the network flow formulation of Christiansen [13]. Their problem is for a heterogeneous fleet of ships engaged in pickup-and-delivery for several liquid bulk products during inventory dependent time-windows. In their problem, they have hold constraints of a sort, but the holds are pre-dedicated to specific material types, which simplifies the allocation of materials. Al-Khayyal and Hwang formulate their problem as a MIP, running numerous randomly generated “test problems” to determine where the structure of the model can be exploited in order to solve larger instances. They note exponential growth in their solution times as the number of port visits increases, and find that a solution scheme starting with a small number of possible visits, then selectively increasing them, may lead to a more robust procedure.

Giesen et al., at a conference in 2007 [34], presented a combined multi-item inventory management and ship routing problem for bulk oil products, with hold and draft constraints for a heterogeneous fleet. Their problem includes time-windows, and determines in advance the number of visits and associated time-windows for each port. To determine the routes, schedules and hold allocations of ships, they use a MIP decomposed into three stages: a greedy or GRASP heuristic to identify an initial feasible solution, followed by a sequential optimization procedure to improve a set of ship schedules without considering hold constraints, and finally product allocations are optimized considering all material-to-hold allocation constraints. They found their approach to solve problems with up to six ships and twelve ports for a 15-day period in less than an hour of computation time.

The hold constraints for the tramp shipping problem considered by Jetlund and Karimi [39] have the additional complication that each cargo must be compatible with the coating of its hold or with the previously stored material, and that cargoes in neighbouring compartments must be nonreactive. In contrast to our problem, Jetlund and Karimi study a tramp operation with multiple time periods and time-windows; to solve their problem, they applied a heuristic method of first solving for only one ship using MIP, and then repeatedly using these solutions to obtain solutions for multiple ships.

The problem presented by Fagerholt and Christiansen [26] is a cargo routing problem with time-window constraints. They also allow for the servicing of some cargoes by spot carrier, which is similar in some ways to the use of trucks in our problem. Each of the ships in their problem has flexible cargo holds that can be partitioned into smaller holds in a variety of ways. They apply a two-phase set-partitioning approach, generating candidate schedules for each ship that include the allocation of material to holds. Candidate schedules are added to the problem if they rate well on measures of capacity utilization, using an algorithm described in another paper [27]. Fagerholt and Christiansen achieve optimal solutions with this method for several cases of a real ship planning problem.

Ronen [53] presents an inventory routing problem with hold constraints and time period considerations. As with our problem, Ronen separates the inventory planning and cargo routing into
separate stages, reducing the second stage to a cargo routing problem for multiple time periods. The possible routes in this study, however, are simple in comparison to our problem, being only a single leg from from a loading port to a discharging port. The problem is solved using a MIP for small problems, and for larger problems a heuristic that produces fast but sub-optimal solutions.

While no exact match to the problem at hand could be found in the literature, a summary of the studies considered most similar in regard to important characteristics such as hold constraints and routing type has been given here. For a more comprehensive review of all literature in maritime transportation optimization, the reader is again referred to reviews by Christiansen et al. [16] and Ronen [51, 52].

4 Problem formulation

In this section we define the problem in detail. Notation for sets, parameters and decision variables used in this problem is given in Section 4.2, and the MIP formulation is described in Section 4.3.

4.1 Description of problem

The problem at hand is a single time-period ship-routing problem with hold and draft constraints, a heterogeneous fleet, and predetermined exact amounts of different raw materials to be loaded and discharged at specific loading and discharging ports. We seek an optimal schedule of routes that minimizes the total shipment cost while satisfying the constraints.

To begin with, the problem was modelled as a MIP, with each ship journey that may or may not be used in the solution represented by a binary variable, \( \gamma_i \). The number of trips that will optimally be made is not known in advance; as there is no limit set on this number, the set \( I \) of trips included in the model will be significantly larger than the number that will ultimately be required.

A ship used for a trip may be either of two different types, and so a number of potential trips are created for each type of ship, as it is not known in advance which ship types the optimal solution will use. The type of the ship, then, is known as a function of the trip index: \( v(i) \in V \).

A ship’s type is distinguished by the following characteristics: its total capacity, \([TSC]_v\), its number of holds/compartments, \([SHN]_{vh}\), the capacity of each of its holds, \([SHC]_{vh}\), its costs for loading or discharging at specific ports, \([RMC]_{vp}\), its costs for starting a route at a specific loading port, \([PSC]_{vl}\), its cost for travelling between two particular ports, \([SEC]_{vpq}\), and its draft limit for each port, \([DL]_{vp}\).

The draft limit imposes a maximum total weight, set by each port for each type of ship, for that ship to be allowed to dock at that port. A maximum cargo weight capacity, \([LC]_{vpq}\), for any leg from a port \( p \in P \) to a port \( q \in P \) can be defined for each ship type \( v \in V \) by using the minimum of the draft limit at each of the ports and the capacity of that ship type. This consolidates what could otherwise be several capacity constraints in the model.
The types of ship available are Handysize, which is able to carry up to 28,600 tonnes of cargo, and the larger Handymax, which can carry up to 44,000 tonnes of cargo. Each ship has five holds of differing sizes.

We may recall the description of the hold constraints, that for each ship type there must be no more than one type of raw material stored in any individual hold, although neighbouring holds may contain different raw material types on the same trip.

The geographical locations of the ports, with each loading port located overseas at a considerable distance from the discharging ports in Australia, allows for an important simplification to the route structures: only routes having the property that all loading ports in the route are visited before any discharging ports are visited need to be considered. A route therefore consists of a sequence of loading ports followed by a sequence of discharging ports. No port acts as both a loading port and a discharging port.

The loading ports are also divided into regions, with each loading port belonging to exactly one region, and travel only being possible between two loading ports if they belong to the same region. A number of trip variables are created for each region as well as for each ship type, and thus the region that a ship may travel from is also known as a function of the trip index: $r(i) \in R$.

There is a maximum set for the numbers of both loading and discharging ports that can be visited in any single trip.

![Figure 4.1: Example of available legs with loading port regions](image)

Upon reaching the first discharging port in Australia, the use of trucks is available for transporting cargo between the remaining discharging ports. This can be beneficial if there is only a small amount of material to be taken to a particular port, making it inefficient to pay the travel and discharging costs for a ship.

The costs associated with sending trucks between discharging ports are calculated proportionally.
to the weight of material carried and the distance travelled. There is a soft limit to the number of trucks that may be used, with a penalty applied to the use of any additional trucks.

![Diagram](image)

Figure 4.2: Example trip involving two loading ports and three discharging ports, showing material-to-hold assignments, with the load from D2 to D3 carried by trucks

Our solution must determine the number of ships we need to use, and the routes those ships will take, including how much of each type of material is loaded or discharged at each port on the route. The solution must also find the number of trucks to use, if any, between which discharging ports they will travel and how much of each type of material they will carry.

### 4.2 Notation

The notation for this problem is as follows:

**Sets**

- \( I := \) set of trips travelled by ships;
- \( P := \) set of ports;
- \( L := \) subset of ports that are loading ports \((L \subseteq P)\);
- \( D := \) subset of ports that are discharging ports \((D = P - L)\);
- \( M := \) set of raw material types;
- \( H := \) set of hold size types;
- \( V := \) set of ship types, with \( v(i) \) the ship type used on trip \( i \in I \);
- \( R := \) set of regions for loading ports, with \( r(i) \) the loading port region for trip \( i \in I \);
- \( L_r := \) subset of loading ports that are in region \( r \ (\sum_{r \in R} L_r = L) \);
- \( I_r := \) subset of trips that are in region \( r \ (\sum_{r \in R} I_r = I) \).
Constants and Parameters

\[ \text{Supply: amount of material } m \in M \text{ to supply to discharging port } d \in D; \]
\[ \text{Buy: amount of material } m \in M \text{ to buy from loading port } l \in L; \]
\[ \text{Total Ship Capacity: total capacity of ship type } v \in V; \]
\[ \text{Ship Hold Number: number of holds of type } h \in H \text{ on ship type } v \in V; \]
\[ \text{Ship Hold Capacity: capacity of each hold of type } h \in H \text{ on ship type } v \in V; \]
\[ \text{Raw Material Cost: unit cost to load/discharge material } m \in M \text{ at port } p \in P \text{ for ship type } v \in V; \]
\[ \text{Port Start Cost: cost of beginning a route at loading port } l \in L \text{ for ship type } v \in V; \]
\[ \text{Sail-and-Entry Cost: fixed cost for sailing a leg from port } p \in P \text{ to port } q \in P \text{ for ship type } v \in V; \]
\[ \text{Draft Limit: limit for material discharged at port } p \in P \text{ by ship type } v \in V; \]
\[ \text{Leg Capacity: maximum shipload for travel between ports } p \in P \text{ and } q \in P \text{ for ship type } v \in V, \text{ defined as: } [LC]_{vpq} := \min\{[TSC]_v, [DL]_{vp}, [DL]_{vq}\}; \]
\[ \text{Maximum Discharging Ports: maximum number of discharging ports } d \in D \text{ allowed in any route } i \in I; \]
\[ \text{Maximum Loading Ports: maximum number of loading ports } l \in L \text{ allowed in any route } i \in I; \]
\[ \text{Truck Capacity: capacity of each truck; } \]
\[ \text{Truck Limit: soft limit on the total number of trucks that may be used; } \]
\[ \text{Extra Truck Factor: penalty factor for exceeding the soft truck limit; } \]
\[ \text{Material Ground Cost: cost of sending material by truck from port } p \in D \text{ to port } q \in D. \]

Decision Variables

\[ \gamma_i := \text{binary: 1 if trip } i \in I \text{ is used, 0 otherwise; } \]
\[ f_{li} := \text{binary: 1 if loading port } l \in L \text{ is the first port visited in trip } i \in I, \text{ 0 otherwise; } \]
\[ x_{pqi} := \text{binary: 1 if trip } i \in I \text{ includes the leg from port } p \in P \text{ to port } q \in P, \text{ 0 otherwise; } \]
\[ y_{mpi} := \text{real: amount of material } m \in M \text{ on the leg from port } p \in P \text{ to } q \in P \text{ in trip } i \in I; \]
\[ z_{nhi} := \text{real: amount of material } m \in M \text{ to load/discharge at port } p \in P \text{ on trip } i \in I; \]
\[ o_{mpq} := \text{integer: number of holds of type } h \in H \text{ for material } m \in M \text{ on trip } i \in I; \]
\[ o^+ := \text{real: number of trucks shipping material } m \in M \text{ from port } p \in D \text{ to port } q \in D; \]
\[ o^+ := \text{real: number of trucks used over the soft truck limit. } \]

4.3 MIP formulation

The total cost (4.3.1) is given by the cost of starting at the first port in each trip, \( \sum_{i \in I} \sum_{l \in L} [PSC]_{vi} f_{li} \), plus the approximated fixed cost of each leg in each trip, \( \sum_{i \in I} \sum_{p \in P} \sum_{q \in P} [SEC]_{vi} x_{pqi} \), plus the loading and discharging costs, \( \sum_{m \in M} \sum_{p \in P} [RMC]_{vp} z_{mpi} \), plus the cost of shipping by truck, \( \sum_{p \in D} \sum_{q \in D} \sum_{m \in M} [MGC]_{pq} [TCap] o_{mpq} \), plus the penalty for going over the soft truck
limit, \( o^+ [ETF] \max_{p \in D, q \in D} [MGC]_{qp} [TCap] \). The truck limit penalty is calculated by multiplying the penalty factor \([ETF]\) by the cost of routing each truck used above the limit over the most expensive leg.

\[
\min_{\gamma, f, x, y, z, n, o, o^+} \sum_{i \in I} \sum_{l \in L} [PSC]_{v(i)} x_{li} + \sum_{i \in I} \sum_{p \in P} \sum_{q \in D} [SEC]_{v(i)} y_{pqi} + \sum_{m \in M} \sum_{p \in D} \sum_{q \in D} [RMC]_{v(i)} z_{mpq} + o^+ [ETF] \max_{p \in D, q \in D} [MGC]_{qp} [TCap] \tag{4.3.1}
\]

The constraints are defined as follows, each followed by a short explanation:

\[
[L_C]_{v(i)} x_{pqi} - \sum_{m \in M} y_{mpqi} \geq 0 \quad \forall p, q \in P, \quad \forall i \in I \tag{4.3.2}
\]

(4.3.2) The total amount of material taken on a leg from port \( p \in P \) to \( q \in P \) on a ship of type \( v \in V \) in a trip \( i \in I \) must be less than or equal to the maximum weight capacity for that leg for that ship type.

\[\sum_{i \in I} x_{mli} = [B]_{ml} \quad \forall m \in M, \quad \forall l \in L \tag{4.3.3}\]

(4.3.3) The total amount of material \( m \in M \) to buy from loading port \( l \in L \) must be equal to the amount of that material loaded from that port.

\[\sum_{i \in I} o_{mpq} - \sum_{q \in D} o_{mpq} = [S]_{mp} \quad \forall m \in M, \quad \forall p \in D \tag{4.3.4}\]

(4.3.4) The total amount of material \( m \in M \) to supply to discharging port \( d \in D \) must be equal to the amount of that material discharged at or shipped by truck into that port, less the amount being shipped by truck away from that port.

\[
[TSC]_{v(i)} \sum_{l \in L} \sum_{d \in D} x_{ldi} - \sum_{l \in L} \sum_{d \in D} \sum_{m \in M} y_{mdl} \geq 0 \quad \forall i \in I \tag{4.3.5}
\]

(4.3.5) The total amount of materials taken over the leg joining the set of loading ports to the set of discharging ports (call this leg “the bridge”) on a ship of type \( v \in V \) in a trip \( i \in I \) must be less than or equal to the total capacity of that ship type.

\[\sum_{h \in H} [SHC]_{v(i)} m_{nhi} - \sum_{l \in L} \sum_{d \in D} y_{mldi} \geq 0 \quad \forall m \in M, \quad \forall i \in I \tag{4.3.6}\]

(4.3.6) The total amount of material \( m \in M \) taken over the bridge on a ship of type \( v \in V \) in a
trip $i \in I$ must be less than or equal to the combined capacity of all holds that are assigned to that material $m \in M$ in the same trip.

\[- \sum_{m \in M} n_{mhi} \geq -[\text{SHN}]_{v(i)h} \quad \forall h \in H, \quad \forall i \in I \quad (4.3.7)\]

(4.3.7) The number of holds of type $h \in H$ on a ship of type $v \in V$ in a trip $i \in I$ that are assigned to any materials must be less than or equal to the number of holds of that type available on that ship type.

\[o^+ - \sum_{m \in M} \sum_{p \in D} \sum_{q \in D} o_{mpq} \geq -[\text{TLim}] \quad (4.3.8)\]

(4.3.8) The number of trucks used less the number of trucks we allowed to be over the truck limit must be less than or equal to the truck limit\(^5\).

\[\sum_{l \in L} z_{mli} - \sum_{d \in D} y_{mldi} = 0 \quad \forall m \in M, \quad \forall i \in I \quad (4.3.9)\]

(4.3.9) The total amount of material $m \in M$ loaded from all ports $l \in L$ in a trip $i \in I$ must be equal to the total amount of that material discharged at all discharging ports $d \in D$ in the same trip.

\[\sum_{d \in D} z_{mdi} - \sum_{l \in L} \sum_{d \in D} y_{mldi} = 0 \quad \forall m \in M, \quad \forall i \in I \quad (4.3.10)\]

(4.3.10) The total amount of material $m \in M$ discharged at all discharging ports $d \in D$ in a trip $i \in I$ must be equal to the total amount of that material shipped over the bridge in the same trip.

\[\sum_{l \in L} f_{li} - \gamma_i = 0 \quad \forall i \in I \quad (4.3.11)\]

(4.3.11) Exactly one loading port $l \in L$ must be the first port on a trip $i \in I$ if that trip is used.

\[\sum_{l \in L_{r(i)}} f_{li} = 0 \quad \forall i \in I \quad (4.3.12)\]

(4.3.12) The first port for a trip $i \in I$ must not be from any region but that defined by the trip.

\[\sum_{l \in L} f_{li} - \sum_{l \in L} \sum_{d \in D} x_{ldi} \geq 0 \quad \forall i \in I \quad (4.3.13)\]

\(^5\)This constraint will hold as an equality as we attempt to minimize $o_{mpq}$ and $o^+$ in the objective function.
(4.3.13) In any trip \( i \in I \), only one leg travels over the bridge, due to the distance between the loading and discharging regions.

\[
\sum_{p \in P} x_{pdi} - \sum_{p \in D} x_{dpi} \geq 0 \quad \forall d \in D, \quad \forall i \in I
\] (4.3.14)

(4.3.14) A discharging port \( d \in D \) can only be departed from during a trip \( i \in I \) if it has also previously been entered from another port in the same trip.

\[
f_{li} + \sum_{p \in L} x_{pli} - \sum_{p \in P} x_{lpi} = 0 \quad \forall l \in L, \quad \forall i \in I
\] (4.3.15)

(4.3.15) A loading port \( l \in L \) that is departed from during a trip \( i \in I \) must also be entered from another loading port, unless it is the first port visited in the trip.

\[
\gamma_i - f_{li} - \sum_{p \in L} x_{pli} \geq 0 \quad \forall l \in L, \quad \forall i \in I
\] (4.3.16)

(4.3.16) A loading port \( l \in L \) that is the first port in a trip \( i \in I \) cannot also be entered from another port in the same trip, and any other loading port may be entered from another port at most once.

\[
\gamma_i - \sum_{p \in P} x_{pdi} \geq 0 \quad \forall d \in D, \quad \forall i \in I
\] (4.3.17)

(4.3.17) A discharging port \( d \in D \) may be entered from another port at most once on a trip \( i \in I \).

\[-\sum_{l \in L} \sum_{p \in P} x_{lpi} \geq -[\text{MaxLP}] \quad \forall i \in I \] (4.3.18)

\[-\sum_{p \in P} \sum_{d \in D} x_{pdi} \geq -[\text{MaxDP}] \quad \forall i \in I \] (4.3.19)

(4.3.18) - (4.3.19) The number of loading ports \( l \in L \) departed or discharging ports \( d \in D \) entered in a trip \( i \in I \) must be less than or equal to the maximum number of loading/discharging ports that are allowed to be visited in any trip.

\[
z_{mli} - \sum_{p \in P} y_{mlpi} + \sum_{p \in P} y_{mpli} = 0 \quad \forall l \in L, \quad \forall m \in M, \quad \forall i \in I
\] (4.3.20)

(4.3.20) The amount of material \( m \in M \) loaded from loading port \( l \in L \) in trip \( i \in I \) must be equal to the amount of that material shipped away from that port less the amount of that material shipped into that port in the same trip.
\[ z_{m_{d_{i}}} - \sum_{p \in P} y_{m_{p_{d_{i}}}} + \sum_{p \in P} y_{m_{d_{i}p}} = 0 \quad \forall d \in D, \quad \forall m \in M, \quad \forall i \in I \quad (4.3.21) \]

(4.3.21) The amount of material \( m \in M \) discharged at discharging port \( d \in D \) in trip \( i \in I \) must be equal to the amount of that material shipped into that port less the amount of that material shipped away from that port in the same trip.

5 Issues, improvements and results

In looking for ways to improve the speed of finding good quality solutions, several potential ways of improving the original MIP model were investigated, and these are detailed in this section.

5.1 Symmetry breaking: (sb), (sbh)

When some or all of the variables in a MIP model may be permuted without changing the problem structure, that model can be said to be symmetric with respect to those variables. In this section we investigate two attempts at breaking symmetry in the model.

5.1.1 Symmetry breaking among trips: (sb)

The use of a variable \( \gamma_{i} \) to represent the use or non-use of a trip, without that variable initially containing any information about the route structure, creates a symmetry issue in the model. If the use of a certain route should be optimal, it does not matter which trip variable \( \gamma_{i} \) that route is attached to.

Consider, for example, a solution with three trips:

<table>
<thead>
<tr>
<th>Trip Variable</th>
<th>( \gamma_{1} )</th>
<th>( \gamma_{2} )</th>
<th>( \gamma_{3} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ship Type</td>
<td>A</td>
<td>B</td>
<td>A</td>
</tr>
<tr>
<td>Route</td>
<td>( L_{1} \rightarrow L_{2} \rightarrow D_{1} )</td>
<td>( L_{1} \rightarrow D_{3} \rightarrow D_{4} )</td>
<td>( L_{3} \rightarrow D_{2} )</td>
</tr>
</tbody>
</table>

cmpared to a solution with exactly the same three trips associated with different trip variables:

<table>
<thead>
<tr>
<th>Trip Variable</th>
<th>( \gamma_{1} )</th>
<th>( \gamma_{2} )</th>
<th>( \gamma_{3} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ship Type</td>
<td>A</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>Route</td>
<td>( L_{3} \rightarrow D_{2} )</td>
<td>( L_{1} \rightarrow L_{2} \rightarrow D_{1} )</td>
<td>( L_{1} \rightarrow D_{3} \rightarrow D_{4} )</td>
</tr>
</tbody>
</table>

Each of the possible trip variable permutations is considered to be distinct by the MIP solver, when in fact all are equivalent. Searching through the branches of the branch-and-bound tree for each equivalent solution can slow down the branch-and-bound process considerably; models having large symmetry groups are difficult to solve using traditional branch-and-bound or branch-and-cut algorithms for even modestly sized problems [45]. Another problem with sym-
metry in the model is that the bounds obtained from the LP-relaxations are rather inadequate for symmetric variables [7].

There are multiple methods for breaking symmetries, such as reformulation using Dantzig-Wolfe decomposition, perturbation of the cost function, and the inclusion of symmetry-breaking inequalities in the set of constraints. Refer to [7] and [45] for more detail on symmetry in MIP. The use of Dantzig-Wolfe decomposition to deal with symmetry for this problem will be covered in Section 6.

Perturbing the cost function by assigning random coefficients to \( \gamma_i \) for each \( i \in I \), which would generally be a simple way to implement symmetry breaking, is not possible for this problem due to the division of trips into ship types; it would lead to the use of a trip of one ship type \( A \) being preferred to the use of a trip of another ship type \( B \) having a higher coefficient, even if no ships of type \( A \) would be used in the optimal solution. Furthermore, as François Margot notes in [45], cost perturbation to deal with symmetry is not very effective, especially in the case of the MIP being infeasible.

Symmetry can be dealt with using symmetry-breaking inequalities by restricting the feasible region to only one of the equivalent feasible alternatives. One relatively simple way of doing this is to enforce the use of a lexicographically maximal ordering of the variables; the polytope defined by the convex hull of these lexicographically maximal solutions is called an orbitope [40].

The following constraint can be applied for each trip \( i \in I \) to enforce that within each ship type and region, the trips with the lowest indices carry the most weight in material, i.e. the trip represented by \( \gamma_1 \) carries more than the trip represented by \( \gamma_2 \), and so on.

\[
\sum_{p \in L_{r(i)}} \sum_{q \in D} \sum_{m \in M} y_{mpq[i]} \geq \sum_{p \in L_{r(i)}} \sum_{q \in D} \sum_{m \in M} y_{mpq[i+1]} \quad \forall i \in I : v(i) = v(i+1), r(i) = r(i+1) \tag{5.1.1}
\]

This constraint ensures that out of all equivalent solutions, only the assignment of trip variables giving the ordering that is lexicographically maximal with respect to material carried will be feasible. Symmetry will still exist between ships that carry exactly the same amount of material, but this should be relatively rare, and at least a significant part of the symmetry is still taken care of by this constraint.

5.1.2 Symmetry breaking among holds: \( (sbh) \)

Another place where symmetry exists within the problem is in the assignment of materials to holds, as there are potentially a number of ways to feasibly distribute an amount of material among the holds of a ship. This is somewhat more complicated to deal with than the trip variable symmetry, as no combination of holds is definitely preferable to another in all cases.

We attempt to break this symmetry to some extent by preferring the use of larger holds to smaller holds in any material-to-hold assignment. To do this, we write a constraint in two forms for each pair of hold types: one form for the case where there are fewer or the same number of the larger hold types than the smaller (5.1.3), and one for the case where there are more of the
larger hold types (5.1.2).

\[ \sum_{m \in M} n_{imh} \geq \sum_{m \in M} n_{imj} \quad \forall i \in I, \forall h, j \in H : [\text{SHC}]_{v(i)h} > [\text{SHC}]_{v(i)j} \text{ and } [\text{SHN}]_{v(i)h} > [\text{SHN}]_{v(i)j} \]  

(5.1.2)

\[ [\text{SHN}]_{v(i)j} - \sum_{m \in M} n_{imj} \geq [\text{SHN}]_{v(i)h} - \sum_{m \in M} n_{imh} \quad \forall i \in I, \forall h, j \in H : [\text{SHC}]_{v(i)h} > [\text{SHC}]_{v(i)j} \text{ and } [\text{SHN}]_{v(i)h} \leq [\text{SHN}]_{v(i)j} \]  

(5.1.3)

In (5.1.2), where there are more of the larger hold types available, we state that more of the larger holds must be used than the smaller holds. In (5.1.3), where there are fewer or the same number of the larger hold types available than the smaller, we state that more of the smaller holds must be left unused than the larger holds.

These constraints are valid because any for solution using more smaller holds than larger holds, where there are unused larger holds available, we could easily swap the use of the smaller holds for the larger holds. The reverse, however, is not true, as the smaller holds may not be able to contain the material held in the larger holds.

It should be noted that these constraints do not entirely deal with the symmetry issue for the material-to-hold assignment in the same way that (5.1.1) did for the trip symmetry; it still happens that an amount of material can equally well be placed under multiple different configurations. A method to ensure that any combination of amounts of material can only possibly be placed into holds in one way has not been devised here, and is left for future work.

5.2 Limiting the creation of trip variables: \langle tlsd\rangle, \langle tlnw\rangle

Another issue in the model is that in general more potential trips are created than will ultimately be needed in any final solution. This makes the problem size larger than necessary, and hence harder to solve, as the additional trips extend the space of feasible solutions to be searched. Providing more trip options will also exacerbate any existing symmetries on those variables.

The number of potential trips created for each ship type \( v \in V \) and region \( r \in R \) is calculated by:

\[ \left\lceil \frac{\sum_{m \in M} \sum_{p \in L_r} [B]_{mp}}{[\text{MSL}]_v} \right\rceil \]  

(5.2.1)

where \([\text{MSL}]_v\) is the minimum load a ship of type \( v \) is expected to carry.

Assuming that we require the minimum, \([\text{MSL}]_v\), to be enforced for each trip, then (5.2.1) obviously bounds the number of trips. Even if, however, the load minimum is not to be enforced, as indeed is the case in the MIP model, \([\text{MSL}]_v\) is small enough that in practice we should generate from this method more trips than required, particularly since the ceiling and not the floor of the fraction is taken in the calculation, and we create these potential trips for each of the two ship types.

Two further methods of improving the performance of the model by reducing the number of trip variables created are investigated. Neither of the trip variable creation limits found in this
section has been proven to be a valid upper bound, however, there have not yet been any cases in practice where limiting the number of trips by these methods has made the true optimal solution infeasible.

These limits are based on the idea that, when considering the use of only one type of ship, the optimal solution will in many cases use the minimal number of trips; it follows that any feasible solution found for one type of ship is likely to use either more or the same number of trips than optimally required. To consider why this may be true, we must recall the condition that all trips must cross a bridge leg to Australia, and that the size of this leg is in general much greater and thus more expensive than those legs within a loading region or legs between discharging ports in Australia. It is beneficial in most cases, then, to use a minimal number of trips so that we cross as few bridge legs as possible.

Each of the algorithms following in this section solve a version of the main problem having a significant tightening of some constraints, e.g. allowing no mixing of materials even within a ship, which perhaps surprisingly make the problem simpler to solve; this constraint tightening will tend to increase rather than decrease the number of trips to be used, and thus further improve the probability that these limits are greater than the optimal number of trips.

5.2.1 Trip limiting by direct supply-demand matching: \(\langle \text{tlsd} \rangle\)

An LP-based algorithm is used to find \([LPUB]_v\), which represents the number of ships of type \(v \in V\) that would be required if legs existed only directly between loading ports and discharging ports, and if only one type of material were carried per ship.

\[
[LPUB]_v := \sum_{m \in M} \sum_{l \in L} \sum_{d \in D} \left\lceil \frac{x_{mld}}{\text{LC}} \right\rceil_{vld} \tag{5.2.2}
\]

where \(x_{mld}\) is found by solving the following LP:

\[
\max \sum_{m \in M} \sum_{l \in L} \sum_{d \in D} x_{mld}
\]

s.t.

\[
[B]_{ml} = \sum_{d \in D} x_{mld} \quad \forall m \in M, \forall l \in L \tag{5.2.3}
\]

\[
[S]_{md} = \sum_{i \in L} x_{mld} \quad \forall m \in M, \forall d \in D
\]

The LP (5.2.3) matches materials to loading port and discharging port pairs, creating direct trade matches between them. In (5.2.2), to calculate the number of ships required, the amount of material traded between each pair of ports is divided by the maximum load allowed to be carried along the leg between them.

Note that the objective function for (5.2.3) is a constant value, being the total overall buy and supply amount. We are in fact “satisficing”\(^6\) the problem rather than maximizing an objective function.

value.

The number of trip variables created in \( \text{tlsd} \) for each region \( r \in R \) and for each ship type \( v \in V \) is:

\[
\min \left\{ \left[ \frac{\sum_{m \in M} \sum_{p \in L} [B]_{mp}}{[MSL]_v} \right], \left( \sum_{m \in M} \sum_{p \in L} [B]_{mp} \right) \right\}.
\] (5.2.4)

Because the trips are generated for each region when they apply only to ship type, the following constraint is also added:

\[
\sum_{i \in I} \gamma_i \leq \max_{v \in V} \left( \left[ \frac{\sum_{m \in M} \sum_{p \in L} [B]_{mp}}{[MSL]_v} \right] \right).
\] (5.2.5)

### 5.2.2 Trip limiting by iterative network flow based algorithm: \( \text{tlnw} \)

A greedy iterative network flow based algorithm is used to find \( [NWUB]_{vr} \), which represents the number of ships that would be required if only one type of material were carried per ship. This is similar to \( [LPUB]_v \), but with multi-leg routes allowed instead of only a direct matching between loading ports and discharging ports.

Network flow algorithms can be solved very quickly due to having the property that despite any integer requirements for the flow, they can be solved to optimality as an LP. In this approach, however, we have added integer variables in addition to the usual network flow problem to force the flow to follow a simple path.

Let \( [NWLC]_{pq} \) be the capacity defined by the algorithm for the leg from \( p \in P \) to \( q \in P \). The algorithm is as follows, for each region \( r \in R \) and ship type \( v \in V \):

1. Simplify the network structure by adding a “dummy” bridge-node between loading and discharging ports, such that all loading port to discharging port connections are via this node.
2. Initialize \( [NWUB]_{vr} = 0 \).
3. Do for each material type \( m \):
   a. Set initial network flow based problem with initial source values \( [NWB]_l := [B]_{ml} \), initial sink values \( [NWS]_d := [S]_{md} \), and arc capacities:
      - \( [NWLC]_{pq} := [LC]_{vpq} \) \( \forall p, q \in L_r \)
      - \( [NWLC]_{p(\text{BRIDGE})} := \min\{[TSC]_v, [DL]_{vp}\} \) \( \forall p \in L_r \)
      - \( [NWLC]_{q(\text{BRIDGE})} := \min\{[TSC]_v, \min_{p \in D}\{[DL]_{vp}\}\} \) \( \forall q \in D \)
      - \( [NWLC]_{pq} := \min\{[TSC]_v, \min_{p \in D}\{[DL]_{vp}\}\} \) \( \forall p, q \in D \)
   b. Repeat until source values are zero:
      i. Flow the maximum amount of material through the network in a simple directed path.
      ii. Reset the network with updated source and sink values, with \( x_p \) the amount of material being loaded (source) or discharged (sink) at port \( p \): \( [NWB]_l := [NWB]_l - x_l \) and \( [NWS]_d := [NWS]_d - x_d \).
iii. Increment the number of trips: $[NWUB]_{vr} := [NWUB]_{vr} + 1$.

It should be noted that the capacities $[NWLC]_{pq}$ have been defined in a special way for discharging ports—they are restricted by the minimum draft limit among all discharging ports. If this were not the case, and the algorithm were to define the arc capacities for discharging ports in the same way as for loading ports (i.e. based on the drafts only of the ports being considered), it would allow each region separately to supply the same material to the same discharging ports, presumably those with the most generous draft limit, which may not be a feasible solution to the problem. Averting this issue by treating all discharging ports as being subject to the minimum draft limit among them does not necessarily produce a feasible solution when solving for each region separately, but it produces a solution which can easily be swapped for a feasible solution using the same number of trips by redirecting the routes. As the trip count is all we retain from the algorithm, this is sufficient.

![Diagram of network with ports and bridge node](image)

**Figure 5.1:** Example of network (tlhw) with ports (squares) and bridge node (circle)

The network flow based problem is solved by the following MIP:

Let $x_p$ be the amount of material loaded or discharged at port $p \in P$, let binary variable $y_{pq} = 1$ if we use leg $pq$, let $z_{pq}$ be the amount of material to carry on leg $pq$, and let the set $Q$ include
all ports plus the bridge node.

\[
\begin{align*}
\text{max} & \quad \sum_{p \in P} x_p \\
\text{s.t.} & \quad z_{pq} \leq y_{pq}[NWLC]_{pq} \quad \forall p, q \in Q \\
& \quad \sum_{q \in Q} y_{pq} \leq 1 \quad \forall p \in Q \\
& \quad \sum_{q \in Q} y_{qp} \leq 1 \quad \forall p \in Q \\
& \quad x_p = \sum_{q \in Q} z_{qp} - \sum_{q \in Q} z_{pq} \quad \forall p \in D \\
& \quad x_p = \sum_{q \in Q} z_{qp} - \sum_{q \in Q} z_{pq} \quad \forall p \in L \\
& \quad \sum_{q \in Q} \sum_{q \in Q} y_{pq} \leq \text{MaxLP} \\
& \quad \sum_{p \in Q} \sum_{q \in Q} y_{pq} \leq \text{MaxDP} \\
\end{align*}
\]  

(5.2.6)

The number of trip variables created in \(\langle tlnw \rangle\) for each region \(r \in R\) and ship type \(v \in V\) is:

\[
\min \left\{ [NWUB]_{vr}, \left[ \frac{\sum_{m \in M} \sum_{p \in L_r} [B]_{mp}}{[MSL]_v} \right] \right\}.
\]  

(5.2.7)

5.3 Lower bound on number of trips: \(\langle lbmh \rangle, \langle lbnw \rangle\)

As well as limiting the number of trips generated, it would also be valuable to bound from below the number of trips to be used so that no time is wasted considering solutions with fewer trips than necessary.

Two straightforward examples of lower bounds on the total number of trips for this problem are applied for each region \(r \in R\):

\[
\frac{\text{Total material weight}}{\text{Capacity of largest ship}} := \left[ \frac{\sum_{m \in M} \sum_{p \in L_r} [B]_{mp}}{\max_{v \in V} [TSC]_v} \right]
\]

and

\[
\frac{\text{Number of different material types}}{\text{Maximum number of holds on any ship type}} := \left[ \frac{\text{count}(m : \sum_{p \in L_r} [B]_{mp} > 0)}{\max_{v \in V} (\sum_{h \in H} [SHN]_{vh})} \right].
\]

5.3.1 Lower Bound by Material-to-Hold Assignment: \(\langle lbmh \rangle\)

A better lower bound on the number of trips can be found by considering the minimum number of trips that would be required if we were to assign the total weight of materials to holds without mixing different material types within the holds.

This bound, \([LBH]_r\), is found with the following MIP for each region \(r \in R\):

Let \(\eta_{vmh}\) be the total number of holds \(h\) over all ships of type \(v\) assigned to material \(m\), let \(\mu_{vmh}\) be the amount of material \(m\) stored in holds \(h\) on ships of type \(v\), and let \(\lambda_v\) be the number of
ships of type \( v \) to be used.

\[
\begin{align*}
\text{min} & \quad \sum_{v \in V} \lambda_v \\
\text{s.t.} & \quad \sum_{m \in M} \eta_{vmh} \leq \lambda_v [\text{SHN}]_{vh} \quad \forall v \in V, \forall h \in H \\
& \quad \mu_{vmh} \leq \eta_{vmh} [\text{SHC}]_{vh} \quad \forall v \in V, \forall h \in H, \forall m \in M \\
& \quad \sum_{v \in V} \sum_{h \in H} \mu_{vmh} = \sum_{p \in L_r} [B]_{mp} \quad \forall m \in M
\end{align*}
\]

(5.3.1)

Then the lower bound is:

\[
[LBH]_r := \sum_{v \in V} \lambda_v \quad \forall r \in R
\]

(5.3.2)

and the following constraint can be added to the main problem MIP for each region \( r \in R \):

\[
\sum_{i \in I_r} \gamma_i \geq [LBH]_r \quad \forall r \in R.
\]

(5.3.3)

Another slightly more complicated version of (5.3.1) was also developed to calculate the bound that should apply in the special case where only one type of ship is used over the entire solution, which also involves adding additional binary variables to the MIP to keep track of whether or not only one ship type is being used.

### 5.3.2 Lower Bound by Iterative Network Flow Based Algorithm: \( \langle \text{lbnw} \rangle \)

The algorithm \( \langle \text{tlnw} \rangle \) in Section 5.2.2 can be modified to create a lower bound, \( [\text{NWLB}]_r \), for the number of trips. Firstly, instead of calculating a bound for each ship type, consider only the ship type with the greatest capacity, \( \text{argmax}_{v \in V} [\text{TSC}]_v \). This will, when minimizing the number of trips used, allow us to use the least number of trips. Secondly, allow all materials to be mixed within the holds; in essence, treat all different material types as one. This material assignment relaxation also allows, out of all possible material assignment configurations, for the least number of trips to be used when minimizing the number of trips. Finally, modify the arc capacities in the network between each two discharging ports \( p \) and \( q \) to be equal to the total capacity of the ship \( [\text{TSC}]_v \). This modification allows for the possibility of using fewer trips by letting material between the discharging ports be transported by trucks instead of by ships. Given this reasoning, the arc capacity between discharging ports could theoretically be considered infinite (until we bound the use of trucks), however, it is sufficient to limit the capacity to \( [\text{TSC}]_v \), as this is the most that could possibly be taken over the bridge on any trip.

The algorithm is as follows for each region \( r \in R \):

1. Simplify the network structure by adding a “dummy” bridge-node as with (tlnw).
2. Initialize \( [\text{NWLB}]_r = 0 \). Set ship type \( v = \text{argmax}_{v \in V} [\text{TSC}]_v \).
(a) Set initial network flow based problem with initial source values $[NW B]_l := \sum_{m \in M} [B]_{ml}$, initial sink values $[NW S]_d := \sum_{m \in M} [S]_{md}$, and arc capacities:

- $[NW LC]_{pq} := [LC]_{pq} \quad \forall p, q \in L$
- $[NW LC]_{p(BRIDGE)} := \min\{[TSC]_v, [DL]_{vp}\} \quad \forall p \in L$
- $[NW LC]_{q(BRIDGE)} := \min\{[TSC]_v, [DL]_{vq}\} \quad \forall q \in D$
- $[NW LC]_{pq} := [TSC]_v \quad \forall p, q \in D$

(b) Repeat until source values are zero:

i. Flow the maximum amount of material through the network in a simple directed path.

ii. Reset the network with updated source and sink values, with $x_p$ the amount of material being loaded (source) or discharged (sink) at port $p$: $[NW B]_l := [NW B]_l - x_l$ and $[NW S]_d := [NW S]_d - x_d$.

iii. Increment the number of trips: $[NW LB]_r := [NW LB]_r + 1$.

Unlike \langle llnw \rangle, which does not produce a true upper bound on the number of trips, the bound on trips generated by \langle lnnw \rangle is truly a lower bound; this is because the algorithm \langle lnnw \rangle minimizes the total number of ships used in a relaxed problem, whereas \langle llnw \rangle does not seek to maximize.

The following constraint can be added to the main problem MIP for each region $r \in R$:

$$\sum_{i \in I_r} \gamma_i \geq [NW LB]_r \quad \forall r \in R.$$

(5.3.4)

5.4 Variable fixing: \langle fix \rangle

The fixing heuristic described in this section is based on a simple technique called LP-and-fix. In LP-and-fix, the LP-relaxation is solved, and some or all variables found to integer in the LP are fixed in a resulting MIP LP-FIX [48]. MIP LP-FIX will either be infeasible based on the fixing, or will be able to provide an LP-and-fix heuristic solution whose objective function can be used as an upper bound for MIP.

An issue with LP-and-fix is determining which of the integer variables should actually be fixed [1]. If too many variables are fixed there is a greater risk that MIP LP-FIX will be infeasible, but if too few variables are fixed then MIP LP-FIX may be so similar to MIP that it is no faster to solve and therefore provides no improvement\footnote{In fact, when being used to bound MIP, solving too similar a MIP LP-FIX and then solving MIP afterward would effectively double the solution time, if not for the time saved in MIP from the bound provided by MIP LP-FIX}.

In this variation of LP-and-fix, we focus on fixing the binary variables governing the use of legs in trips, $x_{pqi}$. On inspection of some results produced by the LP, it was found that the LP solution was fairly weak and only few of these variables took the value 1, although many were zero. In order to get a tighter formulation, it was decided to fix in the following way, acknowledging the risk that this method may cause infeasibility in MIP LP-FIX in some cases:

$$x_{pqi}^{LP} > 0.5 \quad \rightarrow \quad x_{pqi}^{LP-FIX} = 1$$
If MIP\textsuperscript{LP-FIX} is feasible, its objective function is used to bound MIP. If it is not feasible, then it is ignored and MIP is run as when no fixing algorithm is in place.

5.5 Discussion of results

In Table C.1 of the Appendix, we show the results of each solution method for several problem instances.

The problem difficulty appears to be proportional to the problem size, that is, to the numbers of rows and columns in each problem set. A problem becomes hard when it reaches the size of around 1,500 rows and 2,000 columns, and problems with over 2,000 rows and over 3,000 columns are considered very hard.

It is difficult to provide the percentages by which any method improves the solution time in the best case, due to the imposition of a time limit on how long the computation may run. However, through inspection of the results we can determine which methods appear to be consistently performing well for the largest possible range of problem sets.

The greatest improvements made by any single method are by symmetry breaking among trips \textlangle sb\textrangle, and the fixing heuristic \textlangle fix\textrangle. As we would expect, the solution time is most improved by combinations of methods that include both of \textlangle sb\textrangle and \textlangle fix\textrangle.

A symmetry breaking approach would be expected to produce a tree with fewer nodes than a method without symmetry breaking, as symmetrically equivalent branches do not need to be searched. This generally seems to be the case, and where it does not occur, it could be explained by some “luck” in searching the tree for the methods without symmetry breaking.

The good performance of the fixing heuristic \textlangle fix\textrangle is surprising, as it generally produces a very weak bound for the MIP objective function, sometimes over three times the optimal solution value.

We would expect to see the fixing heuristic producing a better bound when combined with symmetry breaking, due to a tighter formulation for the LP-relaxation, however, the results suggest that the opposite of this is occurring.

The fixing heuristic in general is very quick to run, so that the overall solution time can be improved by fixing inclusive of running the heuristic. However, in a couple of cases the heuristic did not perform so quickly, running all the way up to the 300 second time limit, as evidenced in the results by total solution times of around twice this limit. As this method is used for an upper bound on the objective function value, one recommendation would be to limit the time spent on the fixing heuristic to only a small fraction of the total time spent, and then to take the best solution produced at time of finish as a bound.

If the problem is being made significantly simpler to solve by only a weak bound from \textlangle fix\textrangle, then this would suggest that there is much to be gained by finding even stronger bounds for the MIP objective function.
Another method of bounding that was considered but not attempted for this problem is to run a version of the problem made much simpler by either tightening constraints (for an upper bound) or relaxing constraints (for a lower bound). This is a similar idea to that behind the networks $\langle \text{tlnw} \rangle$ and $\langle \text{lbwn} \rangle$, but applied to the objective function instead of to the number of trips.

The upper limits and lower bounds on the number of trips do not appear to be making any consistent improvement to the solution time. These methods bring an improvement for some combinations of methods, but only modestly.

Of the two lower bounds applied to trip variables, $\langle \text{lbwn} \rangle$ shows the best performance.

Neither of the trip limiting approaches produce a stronger limit for any problem set than the original trip limit, (5.2.1); we can tell this from the results as the number of columns are not reduced from the Base method in any implementation of $\langle \text{tlsd} \rangle$ or $\langle \text{tlnw} \rangle$, which they would be if fewer trip variables had been created. Any improvement to the solution time brought about by trip limiting must be entirely from the constraint (5.2.5) added by $\langle \text{tlsd} \rangle$.

As the solution time is so improved by additional bounding and constraining, it is worth investigating other methods of applying constraints to this problem, such as the addition of valid inequalities or cutting planes. These could act to strengthen the LP-relaxation toward its optimal integer values, in a process known as cut-and-branch. See [12] for a overview of valid inequalities for MIP.

The symmetry breaking method applied for the material-to-hold assignment does not perform reliably well, but shows an improvement in some cases. It is not surprising that this method should not show a dramatic improvement, however, as it does not completely remove the symmetry inherent in the hold composition.

Overall, the combination of all methods, $\langle \text{sb} \rangle + \langle \text{sbh} \rangle + \langle \text{tl} \rangle + \langle \text{lb} \rangle + \langle \text{fix} \rangle$, gives the best improvement. This approach is not the fastest to run for all problem sets, but it consistently appears in the top few of the best approaches. When $\langle \text{sb} \rangle + \langle \text{sbh} \rangle + \langle \text{tl} \rangle + \langle \text{lb} \rangle + \langle \text{fix} \rangle$ is outperformed by another method for a problem set, it is not clear what properties of that problem set caused this to be so. If such problem properties could be ascertained, then one could apply a solution technique dependent on these.

Finally, it is worth noting that all unfinished problems, including very hard problems, have final solutions within 3% of the best known solution, even though the reported gap from the lower bound may be much higher. This shows that the solution time is spent mostly on finding a lower bound to prove that the solution is optimal, rather than in actually finding a good solution. This strengthens the recommendation to introduce better bounds for the objective function. In addition, once these improved bounds have been applied, we may wish to limit the solution time even further, as we will be likely to have a good solution within a short time.

6 Decomposing the problem

In this section we describe our initial attempt at a decomposition approach. Any new notation required for this approach is defined in Section 6.2.
Recall from Section 2.3 that column generation (CG) is a method of adding variables to a master problem based on which of them will most likely improve the objective function value, and from Section 2.4 that Benders Decomposition is a method of adding constraints to a master problem based on which costs are not being accounted for or which constraints from the subproblem are being violated.

6.1 Motivation for approach

As mentioned in Section 5.1.1, another way to deal with symmetry among MIP variables is to reformulate the problem using a Dantzig-Wolfe Decomposition. Using this approach for the problem at hand, we can move away from our symmetrical arc-based formulation into a many-variable path-based formulation that will require CG. If, instead of using our trip variable $\gamma_i$, we can define a variable that represents how many times to use a particular route structure, including all port-to-port legs, the region and the ship type, then trips will no longer be indistinguishable in the branch-and-bound tree. To achieve this result, we move the variables defining which legs to use, in what order to use them and with which ship type into a Dantzig-Wolfe CG subproblem.

In addition to applying the Dantzig-Wolfe Decomposition, we will also look at further reformulating the master problem using Benders Decomposition. The variables governing the amounts of material to carry are not required to be integer, and thus can be placed into a Benders subproblem. The potential benefit of using this approach is that some of the variables and constraints from the CG master problem that are indexed by route can be moved into the Benders subproblem, allowing for fewer updates needing to be made to the master problem at each CG iteration.

To allow the master MIP to still capture the flow of material to some extent and thus choose appropriate routes, a new variable $\lambda_{mpq}$ is created to allow the problem to assign a flow of material to each leg, up to capacities depending on which routes are being used along that leg, without needing to track how much material is carried on each particular route or in which holds on a ship. If the material flow assigned by the master problem is not feasible, then a suitable cut will be generated by the Benders subproblem.

6.2 Notation

New notation required for the decomposition approach is as follows:

New set

\[
K := \text{set of defined routes.}
\]
Column generation subproblem decision variables

\[ f_{vl} := \text{binary: 1 if loading port } l \in L \text{ is the first port visited, 0 otherwise;} \]
\[ x_{vpq} := \text{binary: 1 if trip includes the leg from port } p \in P \text{ to } q \in P, 0 \text{ otherwise}. \]

Master problem decision variables

\[ \gamma_k := \text{integer: number of times route } k \in K \text{ is used;} \]
\[ \lambda_{mpq} := \text{real: amount of material } m \in M \text{ to transfer between ports } p \in P \text{ and } q \in P; \]
\[ n_{k mh} := \text{integer: number of holds of type } h \in H \text{ for material } m \in M \text{ on ships using route } k \in K; \]
\[ o_{mpq} := \text{real: number of trucks shipping material } m \in M \text{ from port } p \in D \text{ to port } q \in D; \]
\[ o^+ := \text{real: number of trucks used over the soft truck limit.} \]

Column generation constants in master problem

\[ \pi_{kpq} := 1 \text{ if route } k \in K \text{ includes the leg from port } p \in P \text{ to } q \in P, 0 \text{ otherwise;} \]
\[ \pi(v) := 1 \text{ if ship type } v \in V \text{ is used on route } k \in K, 0 \text{ otherwise.} \]

Benders subproblem decision variables

\[ y_{kmp} := \text{real: material } m \in M \text{ to ship from port } p \in P \text{ to } q \in P \text{ using route } k \in K; \]

Master problem constants in Benders subproblem

\[ \pi_k, \lambda_{mpq}, n_{k mh}, o_{mpq}, \text{ and } o^+ \text{ represent fixed solutions to } \gamma_k, \lambda_{mpq}, n_{k mh}, o_{mpq}, \text{ and } o^+, \text{ respectively.} \]

6.3 Master problem formulation

The master problem objective function (6.3.1) includes the costs of shipping via truck from the original formulation, plus the costs for the routes generated by the CG subproblem, \( \sum_{k \in K} [RC]_k \gamma_k \), and the variable \( \eta \), which in conjunction with the constraints generated by the Benders subproblem accounts for the material loading and discharging costs.

\[
\min_{\eta, \gamma, \lambda, \eta, o, o^+} \eta + \sum_{k \in K} [RC]_k \gamma_k + \sum_{p \in D} \sum_{q \in D} \sum_{m \in M} [MGC]_{pq}[TCap]o_{mpq} \\
+ o^+ [ETF] \max_{p \in D, q \in D} [MGC]_{qp}[TCap] \tag{6.3.1}
\]

The constraints are as follows:

\[
\sum_{k \in K} [LC]_{\pi(v)k} \gamma_k \pi_{kpq} - \sum_{m \in M} \lambda_{mpq} \geq 0 \quad \forall p, q \in P \tag{6.3.2}
\]
(6.3.2) Similar to (4.3.2), but for all routes $k \in K$ combined. The total amount of material taken on a leg from port $p \in P$ to $q \in P$ must be less than or equal to the capacity of that leg for the ship type $v(k) \in V$.

$$\sum_{p \in P} \lambda_{mlp} - \sum_{p \in P} \lambda_{mpl} = [B]_{ml} \quad \forall l \in L, \quad \forall m \in M, \quad \forall k \in K$$  \hspace{1cm} (6.3.3)

(6.3.3) Similar to (4.3.3). The amount of material $m \in M$ to buy from port $l \in L$ must be equal to the amount of material $m \in M$ shipped away from that port $l \in L$ less the amount of material $m \in M$ shipped into that port $l$.

$$\sum_{p \in P} \lambda_{mpd} - \sum_{p \in P} \lambda_{mpd} + [TCap] \sum_{p \in D} o_{mpd} - [TCap] \sum_{p \in D} o_{mpd} = [S]_{md} \quad \forall d \in D, \quad \forall m \in M, \quad \forall k \in K$$  \hspace{1cm} (6.3.4)

(6.3.4) Similar to (4.3.4). The amount of material $m \in M$ to supply to port $d \in D$ must be equal to the amount of material $m \in M$ shipped into that port $d \in D$ less the amount of material $m \in M$ shipped away from that port $d \in D$ (by ship or by truck).

$$\sum_{l \in L} \sum_{d \in D} \lambda_{mld} = \sum_{d \in D} [S]_{md} \quad \forall m \in M$$  \hspace{1cm} (6.3.5)

(6.3.5) The total amount of material $m \in M$ to supply to all discharging ports $d \in D$ must be equal to the total amount of material $m \in M$ taken over the bridge in the same trip.

$$\sum_{h \in H} \sum_{k \in K} [SHC]_{\pi(k)h} n_{kmh} \pi_{kpq} - \lambda_{mpq} \geq 0 \quad \forall m \in M, \quad \forall p, q \in P$$  \hspace{1cm} (6.3.6)

(6.3.6) Similar to (4.3.6), but for all routes $k \in K$ combined. The amount of material $m \in M$ taken from port $p \in P$ to $q \in P$ must be less than or equal to the combined capacity of all holds assigned to that material on routes $k \in K$ covering that leg.

$$\sum_{h \in H} \sum_{k \in K} [SHC]_{\pi(k)h} [SHN]_{\pi(k)h} \gamma_k \pi_{kpq} - \sum_{m \in M} \lambda_{mpq} \geq 0 \quad \forall p, q \in P$$  \hspace{1cm} (6.3.7)

(6.3.7) The total amount of material taken from port $p \in P$ to $q \in P$ must be less than or equal to the combined capacity of all holds on routes $k \in K$ covering that leg.

$$[SHN]_{\pi(k)h} \gamma_k - \sum_{m \in M} n_{kmh} \geq 0 \quad \forall k \in K, \quad \forall h \in H$$  \hspace{1cm} (6.3.8)

(6.3.8) Similar to (4.3.7). The number of holds of type $h \in H$ on route $k \in K$ that are assigned to any materials must be less than or equal to the number of holds of type $h \in H$ available on that route.
\[ o^+ - \sum_{m \in M} \sum_{p \in D} \sum_{q \in D} o_{mpq} \geq -[\text{TLim}] \] (6.3.9)

(6.3.9) Truck limit constraint, the same as (4.3.8).

\[ \sum_{m \in M} n_{kmh} \geq \sum_{m \in M} n_{kmj} \quad \forall k \in K, \quad \forall h, j \in H : [\text{SHC}]_{\pi(k)h} > [\text{SHC}]_{\pi(k)j} \quad \text{and} \quad [\text{SHN}]_{\pi(k)h} > [\text{SHN}]_{\pi(k)j} \] (6.3.10)

\[ [\text{SHN}]_{\pi(k)j} \gamma_k - \sum_{m \in M} n_{kmj} \geq [\text{SHN}]_{\pi(k)h} \gamma_k - \sum_{m \in M} n_{kmh} \quad \forall k \in K, \quad \forall h, j \in H : \] (6.3.11)

(6.3.10) - (6.3.11) Similar to (5.1.2) - (5.1.3). Symmetry breaking for material-to-hold assignment on each route.

\[ - \sum_{k \in K : \pi(k) = v} \gamma_k \geq - \min \left\{ [\text{LPUB}]_v, \sum_{r \in R} [\text{NWUB}]_{vr}, \left[ \frac{\sum_{m \in M} \sum_{p \in L} [B]_{mp}}{[\text{MSL}]_v} \right] \right\} \] (6.3.12)

\[ \sum_{k \in K} \gamma_k \geq \max \left\{ \left[ \frac{\sum_{m \in M} \sum_{p \in L} [B]_{mp}}{\max_v \left[ \text{TSC}_v \right]} \right], \left[ \frac{\text{count} \left( m : \sum_{p \in L} [B]_{mp} > 0 \right)}{\max_v \left( \sum_{h \in H} [\text{SHN}]_{vh} \right)} \right], \sum_{r \in R} [\text{NWLB}]_r, \sum_{r \in R} [\text{LBH}]_r \right\} \] (6.3.13)

(6.3.12) - (6.3.13) From the bounds and limits created in sections 5.2 and 5.3.

\[ \eta \geq \min_{v \in V} \left\{ \sum_{p \in L} \sum_{q \in P} \sum_{m \in M} [\text{RMC}]_{vp} \lambda_{mpq} - \sum_{p \in L} \sum_{q \in P} \sum_{m \in M} [\text{RMC}]_{vp} \lambda_{mpq} \right\} + \sum_{p \in D} \sum_{q \in P} \sum_{m \in M} [\text{RMC}]_{vp} \lambda_{mpq} - \sum_{p \in D} \sum_{q \in P} \sum_{m \in M} [\text{RMC}]_{vp} \lambda_{mpq} \] (6.3.14)

(6.3.14) The materials loading and discharging costs, calculated from the variable \( y \) in the Benders subproblem, can be bounded from below by calculating the costs of shipping materials between ports as governed by \( \lambda \) when using the cheapest ship type \( v \in V \).

\[ \eta \geq \sum_{m \in M} \sum_{p \in P} \sum_{q \in P} d_{mpq}^{6.5.2(f)} \lambda_{mpq} - \sum_{k \in K} \sum_{p \in P} \sum_{q \in P} d_{kpq}^{6.5.3(f)} [\text{LC}]_{\pi(k)pq} \gamma_k \] (6.3.15)

\[ - \sum_{k \in K} \sum_{p \in P} \sum_{m \in M} \sum_{h \in H} d_{kmp}^{6.5.4(f)} [\text{SHC}]_{\pi(h)h} n_{kmh} \quad \forall f \in F \]
\[
0 \geq \sum_{m \in M} \sum_{p \in P} \sum_{q \in P} d_{mpq}^{6.5.2(g)} \lambda_{mpq} - \sum_{k \in K} \sum_{p \in P} \sum_{q \in P} d_{kmpq}^{6.5.3(g)} [LC]_{\tau(k) pq} \gamma_k \\
- \sum_{k \in K} \sum_{p \in P} \sum_{m \in M} \sum_{h \in H} d_{kmp}^{6.5.4(g)} [SHC]_{\tau(k) m} n_{kh} \quad \forall g \in G
\] (6.3.16)

(6.3.15) - (6.3.16) Constraints generated by the Benders subproblem: (6.3.15) to bound the material loading and discharging costs, and (6.3.16) to cut off solutions that are infeasible in the Benders subproblem. \(d_{n(f)}\) is the dual of constraint \(n\) in the Benders subproblem for extreme point \(f\) in its set of extreme points \(F\), and \(d_{n(g)}\) is the dual of constraint \(n\) in the Benders subproblem for extreme ray \(g\) in its set of extreme rays \(G\).

Each time a new route is added to the master MIP by the CG subproblem, any existing constraints generated by Benders Decomposition must be removed, as they applied only to that version of the master problem in which the new route did not exist.

### 6.4 Column generation formulation

The objective function (6.4.1) for the column generation subproblem is the reduced cost of the route to be generated. At each iteration we seek the route with the most negative reduced cost, until no more negative-cost routes exist. The reduced cost is defined by the cost of the route, that is, the costs of starting at the first port and sailing on each leg, less the dual values of the route constraints in the master problem, with \(d_{n}\) the dual value of constraint \(n\).

Note that an existing route in the master problem will have a more detailed reduced cost formulation than a route that is not yet included, as the dual values for many more constraints need to be considered. Also note that no dual for constraint (6.3.8) is included in the reduced cost calculation, despite that constraint containing the route use variable \(\gamma_k\), because its contribution is taken care of by constraint (6.3.7).

\[
\min_{f, x} \sum_{v \in V} \sum_{l \in L} \left[ PSC \right]_{vl f lk} + \sum_{p \in P} \sum_{q \in P} \left[ SEC \right]_{vpq x_{pqk}} \\
- \sum_{p \in P} \sum_{q \in P} \sum_{v \in V} d_{pq}^{6.5.2} x_{pqvl} [LC]_{vpq} - \sum_{p \in P} \sum_{q \in P} \sum_{v \in V} d_{pq}^{6.3.7} x_{pqvl} [TSC]_{v}
\] (6.4.1)

The constraints are as follows:

\[
\sum_{p \in L} f_{vp} - 1 = 0
\] (6.4.2)

(6.4.2) Similar to (4.3.11). Exactly one loading port \(l \in L\) may be the first port.

\[
\sum_{l \in L_r} f_{vl} - x_{vpq} \geq 0 \quad \forall v \in V, \quad \forall p \in L_r, \quad \forall q \in L_r \cup D, \quad \forall r \in R
\] (6.4.3)

(6.4.3) There may be only one ship type \(v \in V\) and region \(r \in R\), and thus a leg \(p \in L\) to \(q \in P\).
may only be used if there is a first port with the same ship type and region.

\[ \sum_{q \in L} f_{vl} - \sum_{p \in L} \sum_{q \in D} x_{vpq} = 0 \quad \forall v \in V \] (6.4.4)

(6.4.4) Similar to (4.3.13). Only one leg travels over the bridge between the loading and discharging regions.

\[ \sum_{v \in V} \sum_{q \in P} x_{vqp} - \sum_{v \in V} \sum_{q \in D} x_{vpq} \geq 0 \quad \forall p \in D \] (6.4.5)

(6.4.5) Similar to (4.3.14). A discharging port \( d \in D \) can only be departed if it has also previously been entered from another port.

\[ \sum_{v \in V} f_{vp} + \sum_{v \in V} \sum_{q \in L} x_{vqp} - \sum_{v \in V} \sum_{q \in P} x_{vqp} = 0 \quad \forall p \in L \] (6.4.6)

(6.4.6) Similar to (4.3.15). A loading port \( l \in L \) that is departed must also be entered, unless it is the first port visited.

\[ -\sum_{v \in V} f_{vp} - \sum_{v \in V} \sum_{q \in L} x_{vqp} \geq -1 \quad \forall p \in L \] (6.4.7)

(6.4.7) Similar to (4.3.16). A loading port \( l \in L \) that is the first port cannot also be entered from another port.

\[ -\sum_{q \in P} \sum_{v \in V} x_{vqp} \geq -1 \quad \forall p \in P \] (6.4.8)

(6.4.8) Similar to (4.3.17). A port \( p \in P \) may be entered from another port at most once.

\[ -\sum_{v \in V} \sum_{p \in L} \sum_{q \in P} x_{vpq} \geq -[\text{MaxLP}] \] (6.4.9)

\[ -\sum_{v \in V} \sum_{p \in P} \sum_{q \in D} x_{vpq} \geq -[\text{MaxDP}] \] (6.4.10)

(6.4.9) - (6.4.10) Similar to (4.3.18) - (4.3.19). The number of loading ports \( p \in L \) departed or discharging ports \( p \in D \) entered must be less than or equal to the maximum number of loading/discharging ports that are allowed to be visited.

### 6.5 Benders Decomposition formulation

The objective function (6.5.1) for the Benders subproblem captures the material loading and discharging costs of the problem.
\[
\min \sum_{k \in K} \sum_{p \in L} \sum_{q \in P} \sum_{m \in M} [RMC]_{vp} y_{kmp} x_{kpq} - \sum_{k \in K} \sum_{p \in D} \sum_{q \in P} \sum_{m \in M} [RMC]_{vp} y_{kmp} x_{kpq} + \sum_{k \in K} \sum_{p \in D} \sum_{q \in P} \sum_{m \in M} [RMC]_{vp} y_{kmp} x_{kpq} - \sum_{k \in K} \sum_{p \in L} \sum_{q \in P} \sum_{m \in M} [RMC]_{vp} y_{kmp} x_{kpq} \]

The constraints are as follows:

\[
\sum_{k \in K} y_{kmp} x_{kpq} \geq \lambda_{mpq} \forall m \in M, \ \forall p, q \in P \quad (6.5.2)
\]

(6.5.2) The amount of material \( m \in M \) taken from port \( p \in P \) to \( q \in P \) over all routes \( k \in K \) must be equal to the amount of that material flowing along that leg as defined in the master problem.

\[
- \sum_{m \in M} y_{kmp} x_{kpq} \geq -[LC]_{\tau(k)pq} \forall k \in K, \ \forall p, q \in P
\]

(6.5.3) Similar to (4.3.2). The total amount of material taken from port \( p \in P \) to \( q \in P \) on route \( k \in K \) must be less than or equal to the total capacity of that leg over all trips on that route.

\[
- \sum_{l \in L} \sum_{d \in D} y_{kmld} x_{kd} \geq -[SHC]_{\pi(h)kmh} \forall k \in K, \ \forall m \in M \quad (6.5.4)
\]

(6.5.4) Similar to (4.3.6). The amount of material \( m \in M \) shipped from port \( p \in P \) on route \( k \in K \) must be less than or equal to the total capacity of the number of holds assigned to that material over all ships travelling on that route.

\[
y_{kmp} - \sum_{q \in P} y_{kmq} x_{kqp} \geq 0 \ \forall k \in K, \ \forall p \in L, \ \forall m \in M \quad (6.5.5)
\]

(6.5.5) The amount of material \( m \in M \) shipped away from loading port \( p \in L \) on route \( k \in K \) must be greater than or equal to the amount of that material shipped into that port on that route.

\[
\sum_{q \in P} y_{kmq} x_{kqp} - y_{kmp} \geq 0 \ \forall k \in K, \ \forall p \in D, \ \forall m \in M \quad (6.5.6)
\]

(6.5.6) The amount of material \( m \in M \) shipped into discharging port \( p \in D \) on route \( k \in K \) must be greater than or equal to the amount of that material shipped away from that port on that route.
Implementing the decomposition approach described is quite a large undertaking, with many things to be considered, and only an initial investigation of the approach was able to be conducted in this work. In this section we will describe ideas for future work in addition to what has been attempted for this project.

7.1 Ordering of master and subproblems

One very important consideration for this particular approach, the combination of both Benders Decomposition and column generation (CG), is in which order to apply the two methods.

A master problem with Benders constraints included should cause CG to provide better routes than a master problem without, but in the same way, a master problem solved with more route variables has a greater chance of being optimal, causing the Benders subproblem to add constraints applicable to the optimal solution instead of constraints applicable to a sub-optimal solution.

As well as the issue of in which order to apply the methods, we need to consider how many times to run each, that is, how many columns to add by CG before applying Benders Decomposition, and how many constraints to add in Benders Decomposition before applying CG.

After some time spent in this approach, it is possible that certain route variables can be identified as not appearing in the basis for any optimal solutions, and that these variables could be removed. In removing route variables, there is a chance that the CG subproblem will add them again in the next iteration, unless it is specifically constrained so it does not.

These considerations give us many options for how to apply the approach, and an investigation into which of the possibilities will perform best is left to further research. In this work, we run the approach as shown in the flow chart in Figure 7.1.

First, Benders Decomposition is applied until all cuts are generated. Then, CG is run for a maximum of 10 routes. If no routes with negative reduced cost are found by CG, then we stop and output the master MIP solution. If any routes are added by CG, then we return to Benders Decomposition and continue.

7.2 Feasible route generation

To begin the CG process, it may help to begin with some existing routes that will allow for an initial feasible solution to the master problem. A very simple way of doing this is to define as a route each leg between a loading port and a discharging port for each ship type, but these routes alone are unlikely to give particularly good solutions.

---

8Constraints that cut off sub-optimal solutions but not the optimal solution are unnecessary, in the same way that variables that will not be basic in the optimal solution are unnecessary. These facts are, of course, the entire point of Benders Decomposition and Column Generation.

9The choice of 10 rows per CG iteration was made following a very brief testing phase that has not been recorded.
Figure 7.1: Flow chart of decomposition approach applied
A fairly naïve generating heuristic for starting routes was implemented by considering for each ship type \( v \in V \), for each material \( m \in M \) individually, and also for all materials mixed, the routes generated using a network flow based approach similar to \( \langle \text{tnw} \rangle \) and \( \langle \text{lbnw} \rangle \). However, in this implementation of the network flow based algorithm, the leg capacities are as with the general problem, \( [\text{LC}]_{vpq} \), and instead of maximizing the total material flow, the objective function attempts to capture some of the cost of the route as well as possible with the existence of the bridge node:

\[
\begin{align*}
\min_{y, z} & \quad \sum_{p \in P} \sum_{q \in Q} [\text{SEC}]_{pq} y_{pq} + \sum_{p \in L} \sum_{q \in D} [\text{SEC}]_{pq} \left( y_{p(BRIDGE)} + y_{q(BRIDGE)} \right) - M \sum_{p \in L} z_{p(BRIDGE)} \\
\text{s.t.} & \quad z_{pq} \leq y_{pq}[\text{LC}]_{pq} \quad \forall p, q \in Q \\
& \quad \sum_{q \in Q} y_{pq} \leq 1 \quad \forall p \in Q \\
& \quad \sum_{q \in Q} z_{qp} - \sum_{q \in Q} z_{pq} \leq [\text{NWS}]_p \quad \forall p \in D \\
& \quad \sum_{q \in Q} z_{qp} - \sum_{q \in Q} z_{pq} \geq 0 \quad \forall p \in D \\
& \quad \sum_{p \in L} \sum_{q \in Q} y_{pq} \leq [\text{MaxLP}] \\
& \quad \sum_{p \in D} \sum_{q \in Q} y_{pq} \leq [\text{MaxDP}] \\
\end{align*}
\]  

(7.2.1)

where \( \pi \) is the set ship type for this instance of the problem, binary variable \( y_{pq} = 1 \) if we use leg \( pq \), and \( z_{pq} \) is the amount of material to carry on leg \( pq \). \( [\text{NWB}]_p \) and \( [\text{NWS}]_p \) are the defined buy and supply amounts respectively for port \( p \) in this instance of the problem (either for an individual material type or all material mixed), the set \( Q \) includes all ports plus the bridge node, and \( M \) is a very large number (“big-M”) used to encourage the flow of material across the bridge.

The reason flow needs to be encouraged, instead of enforced by exact buy and supply amounts, is that only one shipload is run through the network in each iteration. Therefore, it is not feasible to expect that the demands can be satisfied completely in each run.

Another possible method for generating initial routes would be to run the original arc-formulation MIP for several seconds, and then to save the routes used by any integer feasible solutions found within that time. This was not attempted for this project, and could be considered for future work.

### 7.3 Cycle elimination in CG subproblem

As there is no material flow in the CG subproblem problem, it does not detect sub-tour cycles. After each new route is found, we apply the following network flow LP to check that no cycle exists. Upon finding a cycle, we return a constraint to the CG subproblem to eliminate it.

Let \( \kappa_{pq} \) indicate the position of a leg in the route. Let \( \pi_{pq} \) be fixed to \( \max_{v \in V} x_{vpq} \), and \( \mathbf{f}_p \) be
fixed to \(\max_{v \in V} f_{vp}\) from the CG subproblem solution.

In the first constraint, we initialize \(\kappa = 1\) for the first leg, and in the next constraint force it to increment for each leg in the route thereafter.

\[
\min_{\kappa} \sum_{p \in P} \sum_{q \in P} \kappa_{pq}
\]

s.t.

\[
\sum_{p \in P} \sum_{q \in P} \kappa_{pq} = 1
\]

\[
\sum_{q \in P} \kappa_{pq} \geq \sum_{q \in P} \kappa_{qp} + 1 \quad \forall p \in P : \overline{f}_p \neq 1, \sum_{q \in P} \overline{x}_{pq} = 1
\]

(7.3.1)

If 7.3.1 is infeasible, then the route is not a simple path, and the following constraint is returned to the CG subproblem:

\[
\sum_{p \in P} \sum_{q \in P} x_{vpq} \leq \sum_{p \in P} \sum_{q \in P} \overline{x}_{pq} - 1 \quad \forall v \in V
\]

(7.3.2)

The cycle elimination process is repeated until 7.3.1 is feasible and thus no cycle exists.

We may note that these cycles could be eliminated within the CG subproblem by the addition of a variable to track the joining of legs, but this set partitioning approach was preferred so that the CG subproblem could be kept simpler, and in hope that the cycles would not occur frequently.

7.4 Bounding \(\eta\) in the master MIP: \(\langle \eta \rangle_{\text{bound}}\)

When the only type of ship used in the master MIP solution is the cheapest ship type to operate, then the bound imposed by (6.3.14) forces \(\eta\) to its true value for the materials loading and discharging costs. However, when instead the most expensive of the two ship types is used in the solution, (6.3.14) provides only a weak bound for \(\eta\).

A reformulation of the decomposition approach to improve this bound for \(\eta\) was attempted by adding to the material flow variable, \(\lambda_{mpq}\), information on what type of ship should be used for the flow, making \(\lambda_{mpqv}\). When this is known, the value for \(\eta\) can be determined precisely, and then the only cuts returned by the Benders subproblem are those corresponding to extreme rays that enforce feasibility.

The model for this revised formulation is detailed in Section D of the Appendix.

Another method for bounding \(\eta\) in the master MIP was considered but not implemented. The idea behind this method is to choose the ship type by which to calculate the materials cost by investigating the ship types for routes passing through each leg. If all routes used in the solution on a leg \(pq\) have the same ship type \(v\), then we know that all material flowing along this leg should be costed to ship type \(v\).
It is possible that the time spent for the additional work required to calculate this cost would outweigh any time saved by bounding $\eta$ better in this manner.

7.5 Fixing in the master MIP: $\langle rfix \rangle$

A considerable amount of time may be spent solving the master MIP at each iteration of Benders Decomposition, so it was thought worthwhile to attempt to improve the solution time using a fixing technique.

In contrast to $\langle fix \rangle$, in this fixing approach we don’t use the fixed MIP to bound the objective function of the MIP, but instead produce a fixed MIP that if feasible will give an optimal solution.

The idea behind this technique is to create a relaxed problem, $\text{MIP}^{\text{H-RELAX}}$, in which the integer requirement for the material-to-hold assignment variables is relaxed, but the rest of the problem remains the same. Upon finding a solution to $\text{MIP}^{\text{H-RELAX}}$, we fix all the route and leg variables from this solution for a new problem, $\text{MIP}^{\text{H-FIX}}$, to see if there is a feasible way to assign materials to holds with these variables fixed. If $\text{MIP}^{\text{H-FIX}}$ is feasible, then we know its objective function value is optimal. A feasible $\text{MIP}^{\text{H-FIX}}$ will have the same objective function value as $\text{MIP}^{\text{H-RELAX}}$, because the hold assignment variables do not affect the cost.

We fix variables $\gamma_k$ and $\lambda_{mpq}$ in the following way in $\text{MIP}^{\text{H-FIX}}$:

- $\gamma_k^{\text{H-FIX}} := \gamma_k^{\text{H-RELAX}}$
- $\lambda_{mpq}^{\text{H-FIX}} := \lambda_{mpq}^{\text{H-RELAX}}$.

Whether or not $\text{MIP}^{\text{H-FIX}}$ is feasible, the solution to $\text{MIP}^{\text{H-RELAX}}$ is used as a lower bound to the master MIP, since it is a relaxation of that problem.

7.6 Discussion of results

In Table C.2 of the Appendix, we show the results of this decomposition approach for the same problem instances discussed in Section 5.5 from Table C.1.

While the results of this approach have not been as positive as desired, and do not conclusively demonstrate it to be superior to the original formulation, we should recall that this was an initial investigation, and that there is potential for further refinement to the method to improve its performance dramatically.

Perhaps the most notable issue with the results, aside from solution times, is that no method for any problem set converges to an optimal solution. The variance between the decomposition solutions and the original formulation solutions ranges from 1% to 15%. These variances can be better understood if we recall that we are running the LP-relaxation of the master MIP to obtain our columns, and that branch-and-price has not been applied.

Failure to reach an optimal solution using CG without the application of branch-and-price occurs because the column that prices in most favourably for the LP-relaxation is not necessarily the column that would most improve the MIP.
In terms of speed, the results are not consistently performing either better or worse than the original formulation. The running time also does not appear to be proportional to the size of the variance from the original formulation solution. We note that the problem set that had been considered the hardest to solve, P12–15, converged within the 300 second time limit to an objective value just 5.2% from its best known solution, and yet the second hardest problem set, P14–17, did not converge at all.

In the basic Reformulation approach, problem set P14–17 had only two rows added via Benders Decomposition, which shows that each master MIP ran on average for 100–150 seconds. This is a disappointing result, a subproblem running for longer than we would like for the entire problem, and it is not clear how many times the subproblem will need to be run before finally converging to a solution.

The initial LP bound also does not correspond to the strength of the solution, as was expected it might. However, further investigation of results may be required for the LP bound at each run of the master MIP, as it is possible that the LP bound becomes weaker as more rows and columns are added. The results have not been analysed to this level detail for this work.

It is worth mentioning that the requirement that the Benders constraints be removed from the master problem after each iteration of CG had not been considered in the initial planning stage. As much of the total solution time is spent in solving the master MIP during Benders Decomposition, the addition of more rather than fewer columns during each iteration of the CG process may improve the running time.

The two improvement methods, $\langle \eta \text{bound} \rangle$ and $\langle r\text{fix} \rangle$, have not shown any consistent improvement in comparison to the Reformulation. For two of the hard problems, P2–6 and P1–4, $\langle \eta \text{bound} \rangle$ produced a better solution to the Reformulation, but took substantially more time to do so. Where $\langle \eta \text{bound} \rangle$ gives the same solution as the Reformulation, it does so using fewer iterations of Benders Decomposition, but each iteration in general takes longer to run due to the additional variables and constraints used in the $\langle \eta \text{bound} \rangle$ formulation.

That $\langle r\text{fix} \rangle$ is producing different solutions to the Reformulation is surprising, as we would expect the results to be the same, differing only in the time spent to solve. The fact that they are different is explained by the existence of multiple optimal solutions for many of the master MIPs, due to the remaining symmetry in the material-to-hold assignment. For some reason, it seems that the application of our fixing heuristic leads us to a different optimal MIP solution than that reached without the heuristic.

The poor solution times for $\langle r\text{fix} \rangle$ stem both from the hold relaxed MIP, $\text{MIP}^{\text{H-RELAX}}$, not running fast enough to bring any improvement to the total time, and that the fixed problem, $\text{MIP}^{\text{H-FIX}}$, is often infeasible, such that the time spent in fixing is effectively wasted.

To improve the performance of the decomposition approach, the implementation of branch-and-price to obtain better solutions should be investigated. However, it is conceivable that the branch-and-price process would slow down the approach considerably. During branch-and-price, as described in Section 2.3, we branch in the B&B tree and run CG again following each
convergence of CG, which could amount to substantially more time spent generating columns than our method of applying CG only for the LP-relaxation. We should also recall the difficulties caused in branch-and-price by the fact that constraints generated in the master problem during B&B need to be taken into account by each of the CG subproblems.

The results suggest that much of the difficulty existing in this approach stems from the MIP being still too large to solve quickly, and that the bound from its LP-relaxation is not strong enough to generate the columns for an optimal MIP solution. An idea that may deal with these concerns is to further decompose the problem by moving the material-to-hold assignment component of the master problem into the CG subproblem, leading to a partitioning approach similar to that used by Fagerholt and Christiansen in [26].

This revised decomposition would simplify the MIP by removing some of its variables, and as the material-to-hold assignment was the cause of some symmetry, we may also have a stronger bound from the LP-relaxation by removing this component of the problem. Should the bound from the LP-relaxation using this revised decomposition be strong enough to generate good columns, it may not then be necessary to apply branch-and-price. However, if the route variables now contain the hold composition information, there would instead be symmetry among the routes.

Another risk associated with this decomposition revision is that there are many more potential route variables to be generated, and it is possible that too many columns will be generated for the master problem to solve quickly before a solution is found.

Should branch-and-price or the revised decomposition approach mentioned above prove to be too slow to solve, the development of a better route generation heuristic to be used in place of the CG subproblem should be considered. If a route generation heuristic can be devised that is fast to run and has a high probability of creating the columns required for an optimal or near-optimal solution, then that heuristic could potentially be used in place of this Stage 2 problem to create the routes for use in the Stage 3 problem.

8 Conclusion

We have investigated techniques to improve a mixed integer programming model for the problem at hand, and have found the best results to be produced by a combination of methods including symmetry breaking, trip bounding and objective function bounding. This combination, called $\langle sb \rangle + \langle sbh \rangle + \langle tl \rangle + \langle lb \rangle + \langle fix \rangle$ in the table of results, performed faster than the original formulation in all attempted problem sets where it solved to completion, solving in just 10% of the original formulation time in its best case and 60% in its worst case for a hard problem.

We also investigated an attempt at reformulating the problem using a combination of Benders Decomposition and Dantzig-Wolfe Decomposition with column generation. The performance results for this approach are at this stage inconclusive, and it is recommended that further research be conducted.

Suggestions for future research are made for both the original formulation and the decomposition formulation. For the original formulation, research into improving the lower and upper bounds of
the objective function, the investigation of valid inequalities, and the acceptance of sub-optimal solutions in a reduced solution time are recommended for future work. For the decomposition formulation, research into branch-and-price, development of better route generation heuristics, and a further decomposition of the problem similar to Fagerholt and Christiansen [26] such that the material-to-hold assignment is solved by the column generation subproblem are suggested.
Bibliography


APPENDIX

A  List of Figures

2.1 Example of a constrained polytope in 2-dimensional space . . . . . . . . . . . . . 8
2.2 Example of a the convex hull of an integer-feasible region . . . . . . . . . . . . . 9
2.3 Example of Branch and Bound (B&B) . . . . . . . . . . . . . . . . . . . . . . . . 11
4.1 Example of regional division . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18
4.2 Example of a trip including trucks . . . . . . . . . . . . . . . . . . . . . . . . . . 19
5.1 Example of network (tnw) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29
7.1 Flow chart of decomposition approach applied . . . . . . . . . . . . . . . . . . . . 43

B  List of Tables

C.1 MIP Improvement Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 56
C.2 Reformulation Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 65

C  Computational results

All models and heuristics for this study were written in XPress-Mosel and implemented using
XPress-MP\textsuperscript{10}, one of the leading commercial MIP solvers. The default settings of XPress-MP
for searching the B&B tree were used.

Over 2,000 lines of XPress-Mosel code were used for the original formulation, and the decompo-
sition formulation consisted of several files of 200 to 1,000 lines each. Due to the space on paper
such a number of lines would fill, the XPress-Mosel code has not been included as text in this
thesis.

Computations were completed on a Dell Inc. OptiPlex GX280 with Intel Pentium 4 CPU 3GHz
and 1024 MB RAM, running Microsoft Windows XP Professional v 5.1.2600.

The names of the algorithms and solution methods in the tables of results are as defined in
the section headings for those methods, e.g. ⟨sb⟩ is symmetry breaking. “Base” refers to the
original MIP model before attempted improvements, and “Reformulation” refers to the model
reformulated by decomposition methods. Combinations of methods in the tables are listed with
a plus symbol between them, e.g. ⟨sb⟩+⟨fix⟩ is a combination of symmetry breaking and fixing.

\textsuperscript{10}http://www.dashoptimization.com/
C.1 Results of MIP improvement methods

Results for the improvement methods of Section 5 are shown for 8 problem sets, each representing a different aggregation of time-periods.

The problem status listed is for within 300 seconds of running time\textsuperscript{11}.

The problems are presented in order of perceived difficulty level, from easiest to hardest. We have defined an easy problem to be one that the Base model could solve within the 300 second time limit, a hard problem to be one that the Base model could not solve within 300 seconds, and a very hard problem to be one that none of the attempted methods could solve within 300 seconds.

The headings for Table C.1 are as follows:

**Problem:** The problem set attempted, e.g. the aggregation of time-periods 1 to 4 is presented as P1–4. The difficulty level and the optimal solution\textsuperscript{12} are also presented in this column;

**Method:** Method attempted, e.g. symmetry breaking \langle sb \rangle or fixing \langle fix \rangle;

**Status:** Status of the problem at time of finish or end of time limit: either Optimal or Unfinished (timed-out);

**Time (s):** Time spent in seconds solving for this problem set and method, inclusive of any heuristics run\textsuperscript{13};

**Nodes:** Number of nodes in the B&B tree at time of finish or end of time limit;

**Sol Node:** The node at which the best solution for this problem set and method was found\textsuperscript{14};

**Gap %:** The percentage by which the lower bound\textsuperscript{15} at end of time limit is lower than the best solution for this problem-set and method;

**Var %:** Variance, the percentage by which the best solution for this problem set and method is greater than the optimal or best known solution for this problem set;

**Rows:** Number of rows (constraints) in the MIP;

**Cols:** Number of columns (variables) in the MIP;

**Fix %:** Percentage by which the upper bound given by the fixing heuristic is greater than the optimal or best known solution for this problem set.

Let \langle tl \rangle = \langle tl\textsubscript{sd} \rangle+\langle tl\textsubscript{nw} \rangle and \langle lb \rangle = \langle lb\textsubscript{mh} \rangle+\langle lb\textsubscript{nw} \rangle.

\textsuperscript{11}The actual stopping time varies as it is determined at an appropriate point by the solver once the imposed time limit has passed.

\textsuperscript{12}If a proven optimal solution could not be found, the best known solution is presented instead. In this case, the best known solution will be used in place of the optimal solution for any relevant calculations, such as variance.

\textsuperscript{13}Note that the 300 second time-limit within XPress-MP applies to each distinct problem, and not to the total running time, so that the total time spent using a method that involved running several heuristics before the MIP may be significantly above the time limit.

\textsuperscript{14}The nodes are numbered in order of when visited during the B&B search.

\textsuperscript{15}That is, the lower bound used for pruning in the B&B tree.
Table C.1: Results of MIP Improvement Methods

<table>
<thead>
<tr>
<th>Problem</th>
<th>Method</th>
<th>Status</th>
<th>Time (s)</th>
<th>Nodes</th>
<th>Sol Node</th>
<th>Gap (%)</th>
<th>Var (%)</th>
<th>Rows</th>
<th>Cols</th>
<th>Fix (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P13–14</td>
<td>Base</td>
<td>Optimal</td>
<td>22</td>
<td>1,911</td>
<td>75</td>
<td>-</td>
<td>-</td>
<td>905</td>
<td>1,352</td>
<td>-</td>
</tr>
<tr>
<td>Difficulty:</td>
<td>⟨sb⟩ + ⟨sbh⟩ + ⟨tl⟩ + ⟨lb⟩ + ⟨fix⟩</td>
<td>Optimal</td>
<td>10</td>
<td>81</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>932</td>
<td>1,358</td>
<td>131.5</td>
</tr>
<tr>
<td>Easy</td>
<td>⟨sb⟩ + ⟨tl⟩ + ⟨lb⟩ + ⟨fix⟩</td>
<td>Optimal</td>
<td>8</td>
<td>55</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>918</td>
<td>1,358</td>
<td>131.5</td>
</tr>
<tr>
<td>Optimal</td>
<td>⟨sb⟩ + ⟨sbh⟩ + ⟨fix⟩</td>
<td>Optimal</td>
<td>16</td>
<td>907</td>
<td>527</td>
<td>-</td>
<td>-</td>
<td>921</td>
<td>1,352</td>
<td>-</td>
</tr>
<tr>
<td>Solution:</td>
<td>⟨sb⟩ + ⟨fix⟩</td>
<td>Optimal</td>
<td>10</td>
<td>311</td>
<td>140</td>
<td>-</td>
<td>-</td>
<td>907</td>
<td>1,352</td>
<td>42.1</td>
</tr>
<tr>
<td>4,714,181</td>
<td>⟨sb⟩ + ⟨sh⟩</td>
<td>Optimal</td>
<td>20</td>
<td>1,645</td>
<td>549</td>
<td>-</td>
<td>-</td>
<td>921</td>
<td>1,352</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>⟨fix⟩</td>
<td>Optimal</td>
<td>27</td>
<td>2,596</td>
<td>912</td>
<td>-</td>
<td>-</td>
<td>907</td>
<td>1,352</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>⟨sb⟩ + ⟨sh⟩</td>
<td>Optimal</td>
<td>10</td>
<td>485</td>
<td>205</td>
<td>-</td>
<td>-</td>
<td>905</td>
<td>1,352</td>
<td>41.8</td>
</tr>
<tr>
<td></td>
<td>⟨sb⟩ + ⟨tl⟩ + ⟨lb⟩</td>
<td>Optimal</td>
<td>26</td>
<td>1,987</td>
<td>728</td>
<td>-</td>
<td>-</td>
<td>918</td>
<td>1,358</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>⟨sb⟩ + ⟨lb⟩</td>
<td>Optimal</td>
<td>23</td>
<td>1,985</td>
<td>816</td>
<td>-</td>
<td>-</td>
<td>917</td>
<td>1,358</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>⟨sb⟩ + ⟨tl⟩</td>
<td>Optimal</td>
<td>27</td>
<td>2,275</td>
<td>1,192</td>
<td>-</td>
<td>-</td>
<td>908</td>
<td>1,352</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>⟨tl⟩ + ⟨lb⟩</td>
<td>Optimal</td>
<td>21</td>
<td>1,901</td>
<td>340</td>
<td>-</td>
<td>-</td>
<td>916</td>
<td>1,358</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>⟨lnw⟩</td>
<td>Optimal</td>
<td>5</td>
<td>293</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>907</td>
<td>1,352</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>⟨lbmh⟩</td>
<td>Optimal</td>
<td>19</td>
<td>1,901</td>
<td>660</td>
<td>-</td>
<td>-</td>
<td>913</td>
<td>1,358</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>⟨tlsd⟩</td>
<td>Optimal</td>
<td>39</td>
<td>3,743</td>
<td>240</td>
<td>-</td>
<td>-</td>
<td>906</td>
<td>1,352</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>⟨thw⟩</td>
<td>Optimal</td>
<td>23</td>
<td>2,449</td>
<td>44</td>
<td>-</td>
<td>-</td>
<td>905</td>
<td>1,352</td>
<td>-</td>
</tr>
</tbody>
</table>

Continued on Next Page...
<table>
<thead>
<tr>
<th>Problem</th>
<th>Method</th>
<th>Status</th>
<th>Time (s)</th>
<th>Nodes</th>
<th>Sol Node</th>
<th>Gap (%)</th>
<th>Var (%)</th>
<th>Rows</th>
<th>Cols</th>
<th>Fix (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P9–13</td>
<td>Base</td>
<td>Optimal</td>
<td>93</td>
<td>3,439</td>
<td>1,560</td>
<td>-</td>
<td>-</td>
<td>1,116</td>
<td>2,065</td>
<td>-</td>
</tr>
<tr>
<td>Difficulty:</td>
<td>⟨sb⟩+(sbh)+(tl)+(lb)+(fix)</td>
<td>Optimal</td>
<td>9</td>
<td>15</td>
<td>8</td>
<td>-</td>
<td>-</td>
<td>1,143</td>
<td>2,071</td>
<td>280.0</td>
</tr>
<tr>
<td>Easy</td>
<td>⟨sb⟩+(tl)+(lb)+(fix)</td>
<td>Optimal</td>
<td>9</td>
<td>7</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>1,129</td>
<td>2,071</td>
<td>280.0</td>
</tr>
<tr>
<td>Optimal Solution:</td>
<td>⟨sb⟩+(sbh)+(fix)</td>
<td>Optimal</td>
<td>24</td>
<td>293</td>
<td>293</td>
<td>-</td>
<td>-</td>
<td>1,132</td>
<td>2,065</td>
<td>59.8</td>
</tr>
<tr>
<td>⟨sb⟩</td>
<td>Optimal</td>
<td>43</td>
<td>1,679</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1,118</td>
<td>2,065</td>
<td>-</td>
</tr>
<tr>
<td>⟨fix⟩</td>
<td>Optimal</td>
<td>629</td>
<td>458</td>
<td>32</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1,116</td>
<td>2,065</td>
<td>59.9</td>
</tr>
<tr>
<td>⟨sb⟩+(tl)+(lb)</td>
<td>Optimal</td>
<td>79</td>
<td>3,143</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1,129</td>
<td>2,071</td>
<td>-</td>
</tr>
<tr>
<td>⟨sb⟩+(lb)</td>
<td>Optimal</td>
<td>89</td>
<td>3,281</td>
<td>230</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1,128</td>
<td>2,071</td>
<td>-</td>
</tr>
<tr>
<td>⟨sb⟩+(tl)</td>
<td>Optimal</td>
<td>55</td>
<td>2,119</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1,119</td>
<td>2,065</td>
<td>-</td>
</tr>
<tr>
<td>⟨tl⟩+(lb)</td>
<td>Optimal</td>
<td>98</td>
<td>4,075</td>
<td>525</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1,127</td>
<td>2,071</td>
<td>-</td>
</tr>
<tr>
<td>⟨lbw⟩</td>
<td>Optimal</td>
<td>84</td>
<td>3,329</td>
<td>513</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1,118</td>
<td>2,065</td>
<td>-</td>
</tr>
<tr>
<td>⟨lbmh⟩</td>
<td>Optimal</td>
<td>74</td>
<td>2,281</td>
<td>372</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1,124</td>
<td>2,071</td>
<td>-</td>
</tr>
<tr>
<td>⟨tlsd⟩</td>
<td>Optimal</td>
<td>102</td>
<td>4,225</td>
<td>1,046</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1,117</td>
<td>2,065</td>
<td>-</td>
</tr>
<tr>
<td>⟨tlnw⟩</td>
<td>Optimal</td>
<td>82</td>
<td>3,727</td>
<td>3,339</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1,116</td>
<td>2,065</td>
<td>-</td>
</tr>
</tbody>
</table>

Continued on Next Page...
<table>
<thead>
<tr>
<th>Problem</th>
<th>Method</th>
<th>Status</th>
<th>Time (s)</th>
<th>Nodes</th>
<th>Sol Node</th>
<th>Gap (%)</th>
<th>Var (%)</th>
<th>Rows</th>
<th>Cols</th>
<th>Fix (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P2-6</td>
<td>Base</td>
<td>Unfinished</td>
<td>328</td>
<td>21,557</td>
<td>3,896</td>
<td>3.0</td>
<td>0.1</td>
<td>1,462</td>
<td>2,042</td>
<td>-</td>
</tr>
<tr>
<td>Difficulty:</td>
<td>⟨sb⟩+⟨sbh⟩+(tl)+(lb)+(fix)</td>
<td>Optimal</td>
<td>34</td>
<td>1,845</td>
<td>763</td>
<td>-</td>
<td>-</td>
<td>1,501</td>
<td>2,048</td>
<td>42.6</td>
</tr>
<tr>
<td>Hard</td>
<td>⟨sb⟩+(tl)+(lb)+(fix)</td>
<td>Optimal</td>
<td>42</td>
<td>1,901</td>
<td>526</td>
<td>-</td>
<td>-</td>
<td>1,479</td>
<td>2,048</td>
<td>76.7</td>
</tr>
<tr>
<td>Optimal</td>
<td>⟨sb⟩+⟨sbh⟩+(fix)</td>
<td>Optimal</td>
<td>48</td>
<td>2,571</td>
<td>1,063</td>
<td>-</td>
<td>-</td>
<td>1,490</td>
<td>2,042</td>
<td>56.6</td>
</tr>
<tr>
<td>Solution:</td>
<td>⟨sb⟩+(fix)</td>
<td>Optimal</td>
<td>48</td>
<td>2,613</td>
<td>592</td>
<td>-</td>
<td>-</td>
<td>1,468</td>
<td>2,042</td>
<td>79.0</td>
</tr>
<tr>
<td>7,218,930</td>
<td>⟨sb⟩+(sbh)</td>
<td>Optimal</td>
<td>324</td>
<td>21,573</td>
<td>16,436</td>
<td>-</td>
<td>-</td>
<td>1,490</td>
<td>2,042</td>
<td>-</td>
</tr>
<tr>
<td>⟨sb⟩</td>
<td>Optimal</td>
<td>205</td>
<td>14,391</td>
<td>10,713</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1,468</td>
<td>2,042</td>
<td>-</td>
</tr>
<tr>
<td>⟨fix⟩</td>
<td>Optimal</td>
<td>245</td>
<td>14,039</td>
<td>3,659</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1,462</td>
<td>2,042</td>
<td>40.3</td>
</tr>
<tr>
<td>⟨sb⟩+(tl)+(lb)</td>
<td>Unfinished</td>
<td>359</td>
<td>18,197</td>
<td>6,766</td>
<td>0.3</td>
<td>0.0</td>
<td>1,479</td>
<td>2,048</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>⟨sb⟩+(lb)</td>
<td>Optimal</td>
<td>159</td>
<td>8,647</td>
<td>6,611</td>
<td>-</td>
<td>-</td>
<td>1,478</td>
<td>2,048</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>⟨sb⟩+(tl)</td>
<td>Optimal</td>
<td>253</td>
<td>12,849</td>
<td>4,011</td>
<td>-</td>
<td>-</td>
<td>1,469</td>
<td>2,042</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>⟨tl⟩+(lb)</td>
<td>Unfinished</td>
<td>330</td>
<td>21,720</td>
<td>3,308</td>
<td>2.4</td>
<td>0.0</td>
<td>1,473</td>
<td>2,048</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>⟨lbw⟩</td>
<td>Optimal</td>
<td>305</td>
<td>23,171</td>
<td>3,093</td>
<td>-</td>
<td>-</td>
<td>1,464</td>
<td>2,042</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>⟨lbmh⟩</td>
<td>Unfinished</td>
<td>354</td>
<td>18,991</td>
<td>3,810</td>
<td>2.5</td>
<td>0.0</td>
<td>1,470</td>
<td>2,048</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>⟨tlsd⟩</td>
<td>Unfinished</td>
<td>327</td>
<td>23,050</td>
<td>6,120</td>
<td>2.7</td>
<td>0.0</td>
<td>1,463</td>
<td>2,042</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>⟨thw⟩</td>
<td>Unfinished</td>
<td>328</td>
<td>21,715</td>
<td>9,831</td>
<td>2.8</td>
<td>0.0</td>
<td>1,462</td>
<td>2,042</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

Continued on Next Page...
Table C.1 – Continued

<table>
<thead>
<tr>
<th>Problem</th>
<th>Method</th>
<th>Status</th>
<th>Time (s)</th>
<th>Nodes</th>
<th>Sol Node</th>
<th>Gap (%)</th>
<th>Var (%)</th>
<th>Rows</th>
<th>Cols</th>
<th>Fix (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P13–15</td>
<td>Base</td>
<td>Unfinished</td>
<td>332</td>
<td>31,286</td>
<td>512</td>
<td>0.5</td>
<td>0.0</td>
<td>1,528</td>
<td>1,957</td>
<td>-</td>
</tr>
<tr>
<td>Difficulty:</td>
<td>⟨sb⟩+(sbh)+(tl)+(lb)+(fix)</td>
<td>Optimal</td>
<td>200</td>
<td>18,806</td>
<td>1,733</td>
<td>-</td>
<td>-</td>
<td>1,574</td>
<td>1,966</td>
<td>76.3</td>
</tr>
<tr>
<td>Hard</td>
<td>⟨sb⟩+(tl)+(lb)+(fix)</td>
<td>Optimal</td>
<td>207</td>
<td>19,988</td>
<td>1,222</td>
<td>-</td>
<td>-</td>
<td>1,549</td>
<td>1,966</td>
<td>50.0</td>
</tr>
<tr>
<td>Optimal</td>
<td>⟨sb⟩+(sbh)+(fix)</td>
<td>Optimal</td>
<td>180</td>
<td>17,732</td>
<td>277</td>
<td>-</td>
<td>-</td>
<td>1,558</td>
<td>1,957</td>
<td>63.6</td>
</tr>
<tr>
<td>Solution:</td>
<td>⟨sb⟩+(fix)</td>
<td>Optimal</td>
<td>227</td>
<td>25,841</td>
<td>1,679</td>
<td>-</td>
<td>-</td>
<td>1,533</td>
<td>1,957</td>
<td>48.5</td>
</tr>
<tr>
<td>8,174,249</td>
<td>⟨sb⟩+(sbh)</td>
<td>Optimal</td>
<td>213</td>
<td>15,712</td>
<td>8,632</td>
<td>-</td>
<td>-</td>
<td>1,558</td>
<td>1,957</td>
<td>-</td>
</tr>
<tr>
<td>⟨sb⟩</td>
<td>Optimal</td>
<td>219</td>
<td>14,625</td>
<td>104</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1,533</td>
<td>1,957</td>
<td>-</td>
</tr>
<tr>
<td>⟨fix⟩</td>
<td>Unfinished</td>
<td>331</td>
<td>35,099</td>
<td>2,696</td>
<td>0.5</td>
<td>0.0</td>
<td>1,528</td>
<td>1,957</td>
<td>50.0</td>
<td>-</td>
</tr>
<tr>
<td>⟨sb⟩+(tl)+(lb)</td>
<td>Optimal</td>
<td>209</td>
<td>15,314</td>
<td>583</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1,549</td>
<td>1,966</td>
<td>-</td>
</tr>
<tr>
<td>⟨sb⟩+(lb)</td>
<td>Optimal</td>
<td>194</td>
<td>15,883</td>
<td>1,106</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1,548</td>
<td>1,966</td>
<td>-</td>
</tr>
<tr>
<td>⟨sb⟩+(tl)</td>
<td>Optimal</td>
<td>313</td>
<td>25,533</td>
<td>25,364</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1,534</td>
<td>1,957</td>
<td>-</td>
</tr>
<tr>
<td>⟨tl⟩+(lb)</td>
<td>Unfinished</td>
<td>336</td>
<td>39,381</td>
<td>6,745</td>
<td>0.5</td>
<td>0.0</td>
<td>1,544</td>
<td>1,966</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>⟨lbw⟩</td>
<td>Unfinished</td>
<td>335</td>
<td>28,676</td>
<td>6,082</td>
<td>0.5</td>
<td>0.0</td>
<td>1,531</td>
<td>1,957</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>⟨lbmh⟩</td>
<td>Unfinished</td>
<td>331</td>
<td>30,951</td>
<td>3,789</td>
<td>0.5</td>
<td>0.0</td>
<td>1,540</td>
<td>1,966</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>⟨tlsd⟩</td>
<td>Unfinished</td>
<td>331</td>
<td>28,038</td>
<td>6,273</td>
<td>0.5</td>
<td>0.0</td>
<td>1,529</td>
<td>1,957</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>⟨tlhw⟩</td>
<td>Unfinished</td>
<td>330</td>
<td>34,825</td>
<td>2,813</td>
<td>0.5</td>
<td>0.0</td>
<td>1,528</td>
<td>1,957</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Continued on Next Page...
Table C.1 – Continued

<table>
<thead>
<tr>
<th>Problem</th>
<th>Method</th>
<th>Status</th>
<th>Time (s)</th>
<th>Nodes</th>
<th>Sol Node</th>
<th>Gap (%)</th>
<th>Var (%)</th>
<th>Rows</th>
<th>Cols</th>
<th>Fix (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P14–16</td>
<td>Base</td>
<td>Unfinished</td>
<td>350</td>
<td>41,130</td>
<td>2,851</td>
<td>0.2</td>
<td>0.0</td>
<td>1,563</td>
<td>2,159</td>
<td>-</td>
</tr>
<tr>
<td>Difficulty: ⟨sb⟩+⟨sbh⟩+(tl)+⟨lb⟩+(fix)</td>
<td>Optimal</td>
<td>92</td>
<td>7,400</td>
<td>1,287</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1,606</td>
<td>2,165</td>
<td>115.4</td>
</tr>
<tr>
<td>Hard</td>
<td>⟨sb⟩+⟨tl⟩+⟨lb⟩+(fix)</td>
<td>Unfinished</td>
<td>351</td>
<td>62,668</td>
<td>4,682</td>
<td>0.2</td>
<td>0.0</td>
<td>1,581</td>
<td>2,165</td>
<td>67.4</td>
</tr>
<tr>
<td>Optimal</td>
<td>⟨sb⟩+⟨sbh⟩+(fix)</td>
<td>Optimal</td>
<td>75</td>
<td>5,543</td>
<td>375</td>
<td>-</td>
<td>-</td>
<td>1,595</td>
<td>2,159</td>
<td>50.6</td>
</tr>
<tr>
<td>Solution: 7,240,233</td>
<td>⟨sb⟩+⟨fix⟩</td>
<td>Optimal</td>
<td>66</td>
<td>7,772</td>
<td>1,015</td>
<td>-</td>
<td>-</td>
<td>1,570</td>
<td>2,159</td>
<td>50.4</td>
</tr>
<tr>
<td>⟨sb⟩</td>
<td>Unfinished</td>
<td>340</td>
<td>33,896</td>
<td>1,088</td>
<td>0.2</td>
<td>0.0</td>
<td>1,570</td>
<td>2,159</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>⟨fix⟩</td>
<td>Unfinished</td>
<td>337</td>
<td>27,190</td>
<td>725</td>
<td>0.3</td>
<td>0.0</td>
<td>1,563</td>
<td>2,159</td>
<td>86.2</td>
<td></td>
</tr>
<tr>
<td>⟨sb⟩+(tl)+⟨lb⟩</td>
<td>Optimal</td>
<td>60</td>
<td>5,306</td>
<td>2,179</td>
<td>-</td>
<td>-</td>
<td>1,581</td>
<td>2,165</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>⟨sb⟩+⟨lb⟩</td>
<td>Unfinished</td>
<td>329</td>
<td>69,776</td>
<td>1,598</td>
<td>0.2</td>
<td>0.0</td>
<td>1,580</td>
<td>2,165</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>⟨sb⟩+(tl)</td>
<td>Optimal</td>
<td>76</td>
<td>7,155</td>
<td>1,376</td>
<td>-</td>
<td>-</td>
<td>1,571</td>
<td>2,159</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>⟨tl⟩+(lb)</td>
<td>Unfinished</td>
<td>338</td>
<td>57,627</td>
<td>536</td>
<td>0.2</td>
<td>0.0</td>
<td>1,574</td>
<td>2,165</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>⟨lbnw⟩</td>
<td>Unfinished</td>
<td>329</td>
<td>54,695</td>
<td>571</td>
<td>0.2</td>
<td>0.0</td>
<td>1,565</td>
<td>2,159</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>⟨lbmh⟩</td>
<td>Unfinished</td>
<td>413</td>
<td>40,839</td>
<td>2,731</td>
<td>0.2</td>
<td>0.0</td>
<td>1,571</td>
<td>2,165</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>⟨tlsh⟩</td>
<td>Unfinished</td>
<td>333</td>
<td>51,598</td>
<td>856</td>
<td>0.2</td>
<td>0.0</td>
<td>1,564</td>
<td>2,159</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>⟨tlhw⟩</td>
<td>Unfinished</td>
<td>331</td>
<td>48,549</td>
<td>457</td>
<td>0.2</td>
<td>0.0</td>
<td>1,563</td>
<td>2,159</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

Continued on Next Page...
Table C.1 – Continued

<table>
<thead>
<tr>
<th>Problem</th>
<th>Method</th>
<th>Status</th>
<th>Time (s)</th>
<th>Nodes</th>
<th>Sol Node</th>
<th>Gap (%)</th>
<th>Var (%)</th>
<th>Rows</th>
<th>Cols</th>
<th>Fix (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1–4</td>
<td>Base</td>
<td>Unfinished</td>
<td>334</td>
<td>20,661</td>
<td>3,942</td>
<td>3.3</td>
<td>0.0</td>
<td>1,641</td>
<td>2,343</td>
<td>-</td>
</tr>
<tr>
<td>Difficulty:</td>
<td>⟨sb⟩+⟨sbh⟩+(tl)+⟨lb⟩+(fix)</td>
<td>Optimal</td>
<td>91</td>
<td>3,968</td>
<td>671</td>
<td>-</td>
<td>-</td>
<td>1,687</td>
<td>2,352</td>
<td>212.8</td>
</tr>
<tr>
<td>Hard</td>
<td>⟨sb⟩+(tl)+⟨lb⟩+(fix)</td>
<td>Optimal</td>
<td>63</td>
<td>3,317</td>
<td>1,306</td>
<td>-</td>
<td>-</td>
<td>1,662</td>
<td>2,352</td>
<td>212.8</td>
</tr>
<tr>
<td>Optimal</td>
<td>⟨sb⟩+⟨sbh⟩+(fix)</td>
<td>Unfinished</td>
<td>341</td>
<td>15,256</td>
<td>3,426</td>
<td>0.9</td>
<td>0.0</td>
<td>1,671</td>
<td>2,343</td>
<td>112.4</td>
</tr>
<tr>
<td>Solution:</td>
<td>⟨sb⟩+⟨fix⟩</td>
<td>Unfinished</td>
<td>337</td>
<td>15,272</td>
<td>13,825</td>
<td>4.0</td>
<td>0.1</td>
<td>1,646</td>
<td>2,343</td>
<td>105.1</td>
</tr>
<tr>
<td>7,941,491</td>
<td>⟨sb⟩+⟨sbh⟩</td>
<td>Unfinished</td>
<td>336</td>
<td>16,852</td>
<td>2,624</td>
<td>8.2</td>
<td>0.1</td>
<td>1,671</td>
<td>2,343</td>
<td>-</td>
</tr>
<tr>
<td>⟨sb⟩</td>
<td>Unfinished</td>
<td>335</td>
<td>17,953</td>
<td>1,667</td>
<td>7.2</td>
<td>0.5</td>
<td>1,646</td>
<td>2,343</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>⟨fix⟩</td>
<td>Optimal</td>
<td>144</td>
<td>6,993</td>
<td>511</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1,641</td>
<td>2,343</td>
<td>113.8</td>
</tr>
<tr>
<td>⟨sb⟩+(tl)+⟨lb⟩</td>
<td>Unfinished</td>
<td>341</td>
<td>20,036</td>
<td>2,189</td>
<td>3.0</td>
<td>0.2</td>
<td>1,662</td>
<td>2,352</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>⟨sb⟩+⟨lb⟩</td>
<td>Unfinished</td>
<td>415</td>
<td>14,810</td>
<td>13,611</td>
<td>3.2</td>
<td>0.0</td>
<td>1,661</td>
<td>2,352</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>⟨sb⟩+(tl)</td>
<td>Unfinished</td>
<td>337</td>
<td>21,523</td>
<td>19,375</td>
<td>3.5</td>
<td>0.0</td>
<td>1,647</td>
<td>2,343</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>⟨tl⟩+(lb)</td>
<td>Unfinished</td>
<td>341</td>
<td>18,864</td>
<td>8,654</td>
<td>3.5</td>
<td>0.1</td>
<td>1,657</td>
<td>2,352</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>⟨lbnw⟩</td>
<td>Unfinished</td>
<td>334</td>
<td>20,965</td>
<td>7,170</td>
<td>3.5</td>
<td>0.0</td>
<td>1,644</td>
<td>2,343</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>⟨lbmh⟩</td>
<td>Unfinished</td>
<td>340</td>
<td>16,571</td>
<td>1</td>
<td>8.4</td>
<td>0.1</td>
<td>1,653</td>
<td>2,352</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>⟨tlsd⟩</td>
<td>Unfinished</td>
<td>337</td>
<td>15,932</td>
<td>674</td>
<td>3.5</td>
<td>0.2</td>
<td>1,642</td>
<td>2,343</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>⟨tlnw⟩</td>
<td>Unfinished</td>
<td>335</td>
<td>20,375</td>
<td>4,999</td>
<td>4.0</td>
<td>0.1</td>
<td>1,641</td>
<td>2,343</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

Continued on Next Page...
<table>
<thead>
<tr>
<th>Problem</th>
<th>Method</th>
<th>Status</th>
<th>Time (s)</th>
<th>Nodes</th>
<th>Sol Node</th>
<th>Gap (%)</th>
<th>Var (%)</th>
<th>Rows</th>
<th>Cols</th>
<th>Fix (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P14–17</td>
<td>Base</td>
<td>Unfinished</td>
<td>321</td>
<td>8,656</td>
<td>502</td>
<td>6.9</td>
<td>0.7</td>
<td>2,351</td>
<td>3,442</td>
<td>-</td>
</tr>
<tr>
<td>Difficulty:</td>
<td>⟨sb⟩+(⟨sbh⟩+(⟨tl⟩+(⟨lb⟩)+(⟨fix⟩))</td>
<td>Unfinished</td>
<td>438</td>
<td>10,535</td>
<td>8,952</td>
<td>2.3</td>
<td>0.0</td>
<td>2,406</td>
<td>3,448</td>
<td>121.2</td>
</tr>
<tr>
<td>Very Hard</td>
<td>⟨sb⟩+(⟨tl⟩+(⟨lb⟩)+(⟨fix⟩))</td>
<td>Unfinished</td>
<td>359</td>
<td>9,503</td>
<td>2,648</td>
<td>2.7</td>
<td>0.2</td>
<td>2,373</td>
<td>3,448</td>
<td>58.8</td>
</tr>
<tr>
<td>Best Known</td>
<td>⟨sb⟩+(⟨sbh⟩)+(⟨fix⟩)</td>
<td>Unfinished</td>
<td>665</td>
<td>7,803</td>
<td>6,822</td>
<td>2.9</td>
<td>0.2</td>
<td>2,395</td>
<td>3,442</td>
<td>- Inf</td>
</tr>
<tr>
<td>Solution:</td>
<td>⟨sb⟩+(⟨fix⟩)</td>
<td>Unfinished</td>
<td>345</td>
<td>8,367</td>
<td>5,846</td>
<td>2.8</td>
<td>0.2</td>
<td>2,362</td>
<td>3,442</td>
<td>121.2</td>
</tr>
<tr>
<td>10,506,522</td>
<td>⟨sb⟩+(⟨sbh⟩)</td>
<td>Unfinished</td>
<td>351</td>
<td>9,236</td>
<td>5,478</td>
<td>3.4</td>
<td>0.5</td>
<td>2,395</td>
<td>3,442</td>
<td>-</td>
</tr>
<tr>
<td>(found by:</td>
<td>⟨sb⟩)</td>
<td>Unfinished</td>
<td>418</td>
<td>7,569</td>
<td>7,095</td>
<td>3.6</td>
<td>0.6</td>
<td>2,362</td>
<td>3,442</td>
<td>-</td>
</tr>
<tr>
<td>⟨sb⟩+(⟨fix⟩)</td>
<td>⟨fix⟩</td>
<td>Unfinished</td>
<td>338</td>
<td>7,219</td>
<td>6,183</td>
<td>5.4</td>
<td>0.3</td>
<td>2,351</td>
<td>3,442</td>
<td>65.7</td>
</tr>
<tr>
<td>in 5,321 secs,</td>
<td>⟨sb⟩+(⟨tl⟩+(⟨lb⟩))</td>
<td>Unfinished</td>
<td>355</td>
<td>11,428</td>
<td>2,291</td>
<td>2.7</td>
<td>0.2</td>
<td>2,373</td>
<td>3,448</td>
<td>-</td>
</tr>
<tr>
<td>1.4% gap)</td>
<td>⟨sb⟩+(⟨lb⟩)</td>
<td>Unfinished</td>
<td>347</td>
<td>8,319</td>
<td>3,196</td>
<td>3.1</td>
<td>0.3</td>
<td>2,372</td>
<td>3,448</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>⟨sb⟩+(⟨tl⟩)</td>
<td>Unfinished</td>
<td>355</td>
<td>9,960</td>
<td>8,288</td>
<td>3.0</td>
<td>0.0</td>
<td>2,363</td>
<td>3,442</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>⟨tl⟩+(⟨lb⟩)</td>
<td>Unfinished</td>
<td>358</td>
<td>10,195</td>
<td>9,666</td>
<td>2.9</td>
<td>0.3</td>
<td>2,362</td>
<td>3,448</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>⟨lbw⟩</td>
<td>Unfinished</td>
<td>352</td>
<td>11,315</td>
<td>4,348</td>
<td>3.2</td>
<td>0.6</td>
<td>2,353</td>
<td>3,442</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>⟨lbmh⟩</td>
<td>Unfinished</td>
<td>348</td>
<td>8,660</td>
<td>1,542</td>
<td>6.4</td>
<td>0.3</td>
<td>2,359</td>
<td>3,448</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>⟨tlsd⟩</td>
<td>Unfinished</td>
<td>348</td>
<td>9,141</td>
<td>1,597</td>
<td>6.5</td>
<td>0.3</td>
<td>2,352</td>
<td>3,442</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>⟨thw⟩</td>
<td>Unfinished</td>
<td>336</td>
<td>6,688</td>
<td>4,660</td>
<td>6.8</td>
<td>0.5</td>
<td>2,351</td>
<td>3,442</td>
<td>-</td>
</tr>
</tbody>
</table>

Continued on Next Page...
<table>
<thead>
<tr>
<th>Problem</th>
<th>Method</th>
<th>Status</th>
<th>Time (s)</th>
<th>Nodes</th>
<th>Sol Node</th>
<th>Gap (%)</th>
<th>Var (%)</th>
<th>Rows</th>
<th>Cols</th>
<th>Fix (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P12–15</td>
<td>Base</td>
<td>Unfinished</td>
<td>376</td>
<td>3,464</td>
<td>1,064</td>
<td>10.4</td>
<td>2.4</td>
<td>2,816</td>
<td>6,087</td>
<td>-</td>
</tr>
<tr>
<td>Difficulty:</td>
<td>⟨sb⟩+⟨sbh⟩+(tl)+(lb)+(fix)</td>
<td>Unfinished</td>
<td>319</td>
<td>1,372</td>
<td>1,001</td>
<td>4.9</td>
<td>1.7</td>
<td>2,872</td>
<td>6,096</td>
<td>103.7</td>
</tr>
<tr>
<td>Very Hard</td>
<td>⟨sb⟩+(tl)+(lb)+(fix)</td>
<td>Unfinished</td>
<td>325</td>
<td>1,551</td>
<td>1,251</td>
<td>4.5</td>
<td>1.1</td>
<td>2,840</td>
<td>6,096</td>
<td>60.0</td>
</tr>
<tr>
<td>Best Known</td>
<td>⟨sb⟩+(sbh)+(fix)</td>
<td>Unfinished</td>
<td>313</td>
<td>1,395</td>
<td>1,231</td>
<td>4.8</td>
<td>1.7</td>
<td>2,856</td>
<td>6,087</td>
<td>74.7</td>
</tr>
<tr>
<td>Solution:</td>
<td>⟨sb⟩+(fix)</td>
<td>Unfinished</td>
<td>311</td>
<td>1,324</td>
<td>1,058</td>
<td>3.7</td>
<td>0.7</td>
<td>2,824</td>
<td>6,087</td>
<td>104.6</td>
</tr>
<tr>
<td>9,740,791</td>
<td>⟨sb⟩+(sbh)</td>
<td>Unfinished</td>
<td>314</td>
<td>2,677</td>
<td>2,650</td>
<td>5.8</td>
<td>1.2</td>
<td>2,856</td>
<td>6,087</td>
<td>-</td>
</tr>
<tr>
<td>(found by:</td>
<td>⟨sb⟩</td>
<td>Unfinished</td>
<td>314</td>
<td>2,904</td>
<td>2,821</td>
<td>7.8</td>
<td>3.0</td>
<td>2,824</td>
<td>6,087</td>
<td>-</td>
</tr>
<tr>
<td>⟨sb⟩+(fix)</td>
<td>⟨fix⟩</td>
<td>Unfinished</td>
<td>323</td>
<td>1,009</td>
<td>501</td>
<td>8.3</td>
<td>1.3</td>
<td>2,816</td>
<td>6,087</td>
<td>84.2</td>
</tr>
<tr>
<td>in 10,494 secs, 1.4% gap)</td>
<td>⟨sb⟩+(tl)+(lb)</td>
<td>Unfinished</td>
<td>324</td>
<td>3,635</td>
<td>1,085</td>
<td>6.7</td>
<td>2.1</td>
<td>2,840</td>
<td>6,096</td>
<td>-</td>
</tr>
<tr>
<td>⟨sb⟩+(lb)</td>
<td>Unfinished</td>
<td>322</td>
<td>3,982</td>
<td>3,790</td>
<td>6.3</td>
<td>1.7</td>
<td>2,839</td>
<td>6,096</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>⟨sb⟩+(tl)</td>
<td>Unfinished</td>
<td>319</td>
<td>2,916</td>
<td>1,545</td>
<td>6.4</td>
<td>1.8</td>
<td>2,825</td>
<td>6,087</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>⟨tl⟩+(lb)</td>
<td>Unfinished</td>
<td>333</td>
<td>5,585</td>
<td>4,438</td>
<td>6.1</td>
<td>1.1</td>
<td>2,832</td>
<td>6,096</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>⟨lbw⟩</td>
<td>Unfinished</td>
<td>331</td>
<td>6,373</td>
<td>5,510</td>
<td>5.9</td>
<td>1.1</td>
<td>2,819</td>
<td>6,087</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>⟨lbmh⟩</td>
<td>Unfinished</td>
<td>319</td>
<td>3,618</td>
<td>2,736</td>
<td>9.1</td>
<td>1.2</td>
<td>2,828</td>
<td>6,096</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>⟨tksd⟩</td>
<td>Unfinished</td>
<td>315</td>
<td>2,914</td>
<td>2,021</td>
<td>10.0</td>
<td>1.2</td>
<td>2,817</td>
<td>6,087</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>⟨thnw⟩</td>
<td>Unfinished</td>
<td>318</td>
<td>3,047</td>
<td>2,019</td>
<td>10.4</td>
<td>1.4</td>
<td>2,816</td>
<td>6,087</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>
C.2 Results of decomposition approach

Results for the decomposition approach in Section 6 and the improvement methods in Section 7 are shown for the same 8 problem sets given in Table C.1, and presented in the same order. The problem status listed is for within 300 seconds of total running time\textsuperscript{16}.

The headings for Table C.2 are as follows:

**Problem**: The problem set attempted, its difficulty level, and the time taken to find a solution by the original formulation using method \( \langle \text{sb}\rangle + \langle \text{sbh}\rangle + \langle \text{tl}\rangle + \langle \text{lb}\rangle + \langle \text{fix}\rangle \);

**Method**: Method attempted;

**Status**: Status of the problem at time of finish or end of time limit: either Finished or Unfinished (timed-out);

**Time (s)**: Time spent in seconds solving for this problem set and method, inclusive of any heuristics run;

**Sol Time (s)**: Time in seconds at which the reported solution was found, if the problem did not finish (with remaining time spent in Benders Decomposition);

**BD Rows**: Total number of constraints added to the master problem by Benders Decomposition (including those that are later deleted);

**CG Cols**: Total number of variables added to the master problem by column generation;

**LP Bound %**: The percentage by which the solution to the LP-relaxation of the initial master problem is lower than its MIP solution;

**Var %**: Variance, the percentage by which the best solution for this problem set and method is greater than the optimal or best known solution for this problem set\textsuperscript{17}.

\textsuperscript{16}For the decomposition approach, the total running time is checked at several intervals in the code.

\textsuperscript{17}If the first iteration of Benders Decomposition does not finish within the time limit, then no feasible solution was found and the variance is not reported.
Table C.2: Results of decomposition approach

<table>
<thead>
<tr>
<th>Problem</th>
<th>Method</th>
<th>Status</th>
<th>Time (s)</th>
<th>Sol Time (s)</th>
<th>BD Rows</th>
<th>CG Cols</th>
<th>LP Bound (%)</th>
<th>Var (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P13–14 (Easy)</td>
<td>Reformulation</td>
<td>Finished</td>
<td>8</td>
<td>-</td>
<td>2</td>
<td>0</td>
<td>4.5</td>
<td>5.8</td>
</tr>
<tr>
<td>Original Time (s):</td>
<td>⟨η bound⟩</td>
<td>Finished</td>
<td>5</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>4.5</td>
<td>5.8</td>
</tr>
<tr>
<td>10</td>
<td>⟨rfix⟩</td>
<td>Finished</td>
<td>7</td>
<td>-</td>
<td>4</td>
<td>0</td>
<td>4.5</td>
<td>5.8</td>
</tr>
<tr>
<td>P9–13 (Easy)</td>
<td>Reformulation</td>
<td>Finished</td>
<td>9</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>1.7</td>
<td>1.1</td>
</tr>
<tr>
<td>Original Time (s):</td>
<td>⟨η bound⟩</td>
<td>Finished</td>
<td>7</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>1.7</td>
<td>1.1</td>
</tr>
<tr>
<td>9</td>
<td>⟨rfix⟩</td>
<td>Finished</td>
<td>8</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>1.7</td>
<td>1.1</td>
</tr>
<tr>
<td>P2–6 (Hard)</td>
<td>Reformulation</td>
<td>Finished</td>
<td>76</td>
<td>-</td>
<td>14</td>
<td>23</td>
<td>14.9</td>
<td>3.8</td>
</tr>
<tr>
<td>Original Time (s):</td>
<td>⟨η bound⟩</td>
<td>Finished</td>
<td>145</td>
<td>-</td>
<td>16</td>
<td>27</td>
<td>14.9</td>
<td>3.4</td>
</tr>
<tr>
<td>34</td>
<td>⟨rfix⟩</td>
<td>Finished</td>
<td>83</td>
<td>-</td>
<td>11</td>
<td>24</td>
<td>14.9</td>
<td>4.0</td>
</tr>
<tr>
<td>P13–15 (Hard)</td>
<td>Reformulation</td>
<td>Finished</td>
<td>16</td>
<td>-</td>
<td>6</td>
<td>1</td>
<td>7.9</td>
<td>10.0</td>
</tr>
<tr>
<td>Original Time (s):</td>
<td>⟨η bound⟩</td>
<td>Finished</td>
<td>12</td>
<td>-</td>
<td>3</td>
<td>0</td>
<td>7.9</td>
<td>10.0</td>
</tr>
<tr>
<td>200</td>
<td>⟨rfix⟩</td>
<td>Finished</td>
<td>22</td>
<td>-</td>
<td>5</td>
<td>1</td>
<td>7.9</td>
<td>10.0</td>
</tr>
<tr>
<td>P14–16 (Hard)</td>
<td>Reformulation</td>
<td>Finished</td>
<td>117</td>
<td>-</td>
<td>14</td>
<td>3</td>
<td>4.8</td>
<td>11.2</td>
</tr>
<tr>
<td>Original Time (s):</td>
<td>⟨η bound⟩</td>
<td>Finished</td>
<td>181</td>
<td>-</td>
<td>12</td>
<td>4</td>
<td>4.0</td>
<td>11.2</td>
</tr>
<tr>
<td>92</td>
<td>⟨rfix⟩</td>
<td>Unfinished</td>
<td>303</td>
<td>303</td>
<td>12</td>
<td>3</td>
<td>4.8</td>
<td>11.2</td>
</tr>
<tr>
<td>P1–4 (Hard)</td>
<td>Reformulation</td>
<td>Finished</td>
<td>234</td>
<td>-</td>
<td>12</td>
<td>29</td>
<td>15.9</td>
<td>10.4</td>
</tr>
<tr>
<td>Original Time (s):</td>
<td>⟨η bound⟩</td>
<td>Unfinished</td>
<td>318</td>
<td>318</td>
<td>4</td>
<td>37</td>
<td>16.6</td>
<td>7.8</td>
</tr>
<tr>
<td>91</td>
<td>⟨rfix⟩</td>
<td>Unfinished</td>
<td>315</td>
<td>60</td>
<td>13</td>
<td>20</td>
<td>15.9</td>
<td>14.9</td>
</tr>
</tbody>
</table>

Continued on Next Page...
## Table C.2 – Continued

<table>
<thead>
<tr>
<th>Problem</th>
<th>Method</th>
<th>Status</th>
<th>Time (s)</th>
<th>Sol Time (s)</th>
<th>BD Rows</th>
<th>CG Cols</th>
<th>LP Bound (%)</th>
<th>Var (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P14–17 (Very Hard)</td>
<td>Reformulation</td>
<td>Unfinished</td>
<td>359</td>
<td>-</td>
<td>2</td>
<td>0</td>
<td>7.7</td>
<td>-</td>
</tr>
<tr>
<td>Original Time (s):</td>
<td>⟨η bound⟩</td>
<td>Unfinished</td>
<td>604</td>
<td>-</td>
<td>3</td>
<td>0</td>
<td>7.7</td>
<td>-</td>
</tr>
<tr>
<td>300+</td>
<td>⟨r fix⟩</td>
<td>Unfinished</td>
<td>353</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>7.7</td>
<td>-</td>
</tr>
<tr>
<td>P12–15 (Very Hard)</td>
<td>Reformulation</td>
<td>Finished</td>
<td>227</td>
<td>-</td>
<td>18</td>
<td>2</td>
<td>7.1</td>
<td>5.2</td>
</tr>
<tr>
<td>Original Time (s):</td>
<td>⟨η bound⟩</td>
<td>Finished</td>
<td>160</td>
<td>-</td>
<td>9</td>
<td>6</td>
<td>7.1</td>
<td>5.2</td>
</tr>
<tr>
<td>300+</td>
<td>⟨r fix⟩</td>
<td>Unfinished</td>
<td>330</td>
<td>293</td>
<td>10</td>
<td>3</td>
<td>6.7</td>
<td>5.2</td>
</tr>
</tbody>
</table>
D Revised reformulation for $\eta$ bound

This section contains the revised reformulation model described in Section 7.4.

The key constraint for this formulation is equation (D.2.8), which does not just form a lower bound for $\eta$ but defines it exactly. Only extreme ray constraints then need to be generated by the Benders subproblem.

Only the master problem and Benders Decomposition subproblem are changed in this revision.

D.1 Notation

Revised master problem decision variables

$$\lambda_{mpqv} := \text{real: amount of material } m \in M \text{ to transfer between ports } p \in P \text{ and } q \in P \text{ using ships of type } v \in V;$$

Revised master problem constants in Benders subproblem

$\bar{\lambda}_{mpqv}$ represents fixed solutions to $\lambda_{mpqv}$.

D.2 Revised MIP reformulation

Master problem

$$\begin{align*}
\min_{\eta, \gamma, \lambda, n, o, o^+} & \quad \eta + \sum_{k \in K} [RC]_k \gamma_k + \sum_{p \in D} \sum_{q \in D} \sum_{m \in M} [MGC]_{pq} [TCap] o_{mpq} \\
& \quad + o^+ [ETF] \max_{p \in D, q \in D} [MGC]_{qp} [TCap] \\
\text{s.t.} & \quad \sum_{k \in K; \pi(k) = v} [LC]_{vpq} \gamma_k \pi_{kpq} - \sum_{m \in M} \lambda_{mpqv} \geq 0 \quad \forall p, q \in P, \quad \forall v \in V \quad (D.2.2) \\
& \quad \sum_{p \in P} \sum_{v \in V} \lambda_{mldv} - \sum_{p \in P} \sum_{v \in V} \lambda_{mplv} = [B]_{ml} \quad \forall l \in L, \quad \forall m \in M, \quad \forall k \in K \quad (D.2.3) \\
& \quad \sum_{p \in P} \sum_{v \in V} \lambda_{mpdv} - \sum_{p \in P} \sum_{v \in V} \lambda_{mdpv} + [TCap] \sum_{p \in D} o_{mpd} - [TCap] \sum_{p \in D} o_{mdp} = [S]_{md} \quad \forall d \in D, \quad \forall m \in M, \quad \forall k \in K \quad (D.2.4) \\
& \quad \sum_{l \in L} \sum_{d \in D} \sum_{v \in V} \lambda_{mldv} = \sum_{d \in D} [S]_{md} \quad \forall m \in M \quad (D.2.5)
\end{align*}$$
\[
\sum_{h \in H} \sum_{k \in K: \tau(k) = v} \text{[SHC]}_{vh} \eta_{kmh} x_{kpq} - \lambda_{mpq} \geq 0 \quad \forall m \in M, \quad \forall p, q \in P, \quad \forall v \in V 
\tag{D.2.6}
\]

\[
\sum_{h \in H} \sum_{k \in K: \tau(k) = v} \text{[SHC]}_{vh} \text{[SHN]}_{vh} y_{kmh} x_{kpq} - \sum_{m \in M} \lambda_{mpq} \geq 0 \quad \forall p, q \in P, \quad \forall v \in V 
\tag{D.2.7}
\]

\[
\eta \geq \sum_{m \in M} \sum_{p \in L} \sum_{q \in P} \sum_{v \in V} \text{[RMC]}_{vp} \lambda_{mpq} - \sum_{m \in M} \sum_{p \in L} \sum_{q \in P} \sum_{v \in V} \text{[RMC]}_{vp} \lambda_{mpq} 
\]

\[
+ \sum_{m \in M} \sum_{p \in D} \sum_{q \in P} \sum_{v \in V} \text{[RMC]}_{vp} \lambda_{mpq} - \sum_{m \in M} \sum_{p \in D} \sum_{q \in P} \sum_{v \in V} \text{[RMC]}_{vp} \lambda_{mpq} 
\tag{D.2.8}
\]

\[
\eta \geq \sum_{m \in M} \sum_{p \in P} \sum_{q \in P} \sum_{v \in V} d_{mpqv}^{6.3.11(f)} \lambda_{mpq} - \sum_{k \in K} \sum_{p \in P} \sum_{q \in P} d_{kpq}^{6.5.3(f)} \text{[LC]}_{\tau(k) pq} \gamma_{k} 
\]

\[
- \sum_{k \in K} \sum_{p \in P} \sum_{m \in M} \sum_{h \in H} d_{kmhp}^{6.4.4(f)} \text{[SHC]}_{\tau(k) h} \eta_{kmh} \quad \forall f \in F 
\tag{D.2.9}
\]

\[
0 \geq \sum_{m \in M} \sum_{p \in P} \sum_{q \in P} \sum_{v \in V} d_{mpqv}^{6.3.11(g)} \lambda_{mpq} - \sum_{k \in K} \sum_{p \in P} \sum_{q \in P} d_{kpq}^{6.5.3(g)} \text{[LC]}_{\tau(k) pq} \gamma_{k} 
\]

\[
- \sum_{k \in K} \sum_{p \in P} \sum_{m \in M} \sum_{h \in H} d_{kmhp}^{6.4.4(g)} \text{[SHC]}_{\tau(k) h} \eta_{kmh} \quad \forall g \in G 
\tag{D.2.10}
\]

Include the following equations in their original form: (6.3.8), (6.3.9), (6.3.11), (6.3.10), (6.3.12) and (6.3.13).

**Benders Decomposition subproblem**

Objective function in original form: (6.5.1).

s.t.

\[
\sum_{k \in K} y_{kpq} x_{kpq} \geq \sum_{v \in V} \lambda_{mpq} \quad \forall m \in M, \quad \forall p, q \in P 
\tag{D.2.11}
\]

Include the following equations in their original form: (6.5.3), (6.5.4), (6.5.5) and (6.5.6).