



Tramp Ship Routing and Scheduling - Models, Methods and Opportunities

Vilhelmsen, Charlotte; Larsen, Jesper; Lusby, Richard Martin

Publication date:
2015

Document Version
Peer reviewed version

[Link back to DTU Orbit](#)

Citation (APA):
Vilhelmsen, C., Larsen, J., & Lusby, R. M. (2015). *Tramp Ship Routing and Scheduling - Models, Methods and Opportunities*. DTU Management Engineering.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Tramp Ship Routing and Scheduling - Models, Methods and Opportunities

Charlotte Vilhelmsen Jesper Larsen Richard M. Lusby

Department of Management Engineering, Technical University of Denmark
Produktionstorvet, Building 426, 2800 Kgs. Lyngby, Denmark
chaan@dtu.dk, jesla@dtu.dk, rmlu@dtu.dk

November 13, 2015

Abstract

In tramp shipping, ships operate much like taxies, following the available demand. This contrasts liner shipping where vessels operate more like busses on a fixed route network according to a published timetable. Tramp operators can enter into long term contracts and thereby determine some of their demand in advance. However, the detailed requirements of these contract cargoes can be subject to ongoing changes, e.g. the destination port can be altered. For tramp operators, a main concern is therefore the efficient and continuous planning of routes and schedules for the individual ships. Due to mergers, pooling, and collaboration efforts between shipping companies, the fleet sizes have grown to a point where manual planning is no longer adequate in a market with tough competition and low freight rates.

The aim of this paper is to provide a comprehensive introduction to tramp ship routing and scheduling. This includes a review on existing literature, modelling approaches, solution methods as well as an analysis of the current status and future opportunities of research within tramp ship routing and scheduling. We argue that rather than developing new solution methods for the basic routing and scheduling problem, focus should now be on extending this basic problem to include additional real-world complexities and develop suitable solution methods for those extensions. Such extensions will enable more tramp operators to benefit from the solution methods while simultaneously creating new opportunities for operators already benefitting from existing methods.

1 Introduction

In 2013, global seaborne trade was estimated to have reached nearly 9.6 billion tons (UNCTAD, 2014). This translates into well over a tonne of cargo for every single individual on the planet, every single year, or equivalently around 90% of world trade by volume (ICS). World trade therefore depends on the international shipping industry's efficiency and competitive freight rates. Furthermore, even though international shipping is the most carbon efficient mode of transport, the CO₂ emissions from the industry as a whole are still estimated to be around 2.2% of current global emissions (ICS) adding further incentive to improve efficiency within the industry. For several years now, the maritime sector has been subject to low freight rates caused by surplus fleet capacity and the weak economy. The combination of low freight rates and high bunker oil prices has made it difficult for many operators to produce earnings sufficient to cover just their minimum operating costs. The shipping industry itself, therefore, has as much incentive as any to

improve their cost-effectiveness and recent years have shown increased exploration of strategies in this direction, e.g. slow steaming, which refers to the practice of operating ships at reduced speeds in order to save fuel. At the same time, the maritime sector also has an incentive to take their CO_2 emissions into consideration, both due to political pressure and to the fact that the industry itself is bound to be affected by the impacts of climate change, such as rising sea levels and more extreme weather. From all sides there is therefore an interest in research to increase efficiency within maritime transportation.

Within commercial shipping it is common to distinguish between three different basic operating modes although they need not be mutually exclusive: *liner*, *tramp* and *industrial* (Lawrence, 1972). Within liner shipping, which is primarily characterised by container shipping, vessels operate much like busses on a fixed route network according to a published timetable. In contrast, tramp ships operate much more like taxis following the available cargoes. Many tramp operators do however know some of the demand in advance as they can enter into long term agreements called *contracts of affreightment* (CoAs). Such contracts state that the tramp operator is obliged to transport specified quantities of cargo between specified ports at a given rate during a specified time period. In addition to these contract cargoes, a tramp operator then tries to maximise profit from optional cargoes called *spot cargoes*. In industrial shipping, the ship operator is also the cargo owner and the objective is therefore to carry all the predefined cargoes at the minimum cost. Tramp and industrial shipping are primarily characterised by tankers and dry bulk carriers.

UNCTAD (2014) reports that although it is estimated that more than half of seaborne trade in dollar terms is containerised, this segment only accounts for 16% of global seaborne trade by volume. Furthermore, container ships only account for 13% of the world fleet deadweight tonnage while bulk carriers and oil tankers are responsible for respectively 43% and 29%. Similarly, UNCTAD (2013) reports that in 2012, bulk commodities accounted for nearly three quarters of the total ton-miles performed that year. Tramp and industrial operators are therefore accountable for a massive part of the global fleet as well as the total ton-miles performed each year, and hence even small improvements in efficiency within these operating modes can be expected to have great impact from both an economic and an environmental aspect.

As such, tramp shipping is not characterised by large economies of scale. Therefore, this shipping mode is generally not difficult to enter and has previously been comprised of many small operators. Perhaps this is the reason why research within tramp shipping has previously lagged far behind that of industrial shipping (Christiansen et al., 2004). However, recent trends of mergers, pooling, and collaboration efforts between shipping companies have increased fleet sizes to a point where manual planning is no longer adequate (Christiansen et al., 2004). A further motivation for research within tramp shipping is that many companies previously involved in industrial shipping have now outsourced their transportation to independent shipping companies while some have even chosen to branch out and become more involved in the spot market in order to better utilise their existing fleet (Christiansen et al., 2004). Both these situations have created a shift from industrial to tramp shipping which, in combination with the increased need to minimise manual planning, has been reflected in the growing literature on research within tramp shipping and also motivated the tramp ship focus of this paper.

Within tramp shipping, and for that matter also industrial shipping, the main focus in tactical planning, and to some extent also operational planning, is routing and scheduling the existing fleet. This paper deals with tactical routing and scheduling where high-level routes and schedules are constructed. More detailed plans are made at the operational level once berth availability, weather conditions etc. are known. For tramp operators the tactical routing and scheduling problem boils down to determining which spot cargoes to transport, assigning all contract cargoes as well as chosen spot cargoes to specific ships while simultaneously finding the sequence and timing of port calls for all ships. The need to decide which spot cargoes to transport can greatly affect the requirements for solution methods. This is for instance the case if a broker calls the tramp operator with a specific cargo request and wants to know almost immediately whether or not the tramp operator is able and willing to transport the given cargo. This need to provide a yes/no answer very quickly means that solution methods with short running time are required. This is in contrast to most other tactical planning problems where running time is rarely an issue.

The tough competition between operators in today's low market adds pressure to devise the most efficient fleet schedules to properly utilise the existing fleet. However, as already mentioned,

recent years have shown an increase in fleet sizes to a point where the construction of efficient schedules is both very time consuming and extremely difficult even for the most experienced planner. In fact, large pool managers can be responsible for routing and scheduling more than 100 vessels worldwide. At the same time, uncertainty plays a big part in maritime optimisation where planners face a constantly changing environment with large daily variations in demand and many unforeseen events. Therefore, it is often necessary to re-plan routes and schedules continuously to accommodate new cargoes and changes to existing plans. This is even more relevant for tramp operators than for industrial operators due to the interaction with the spot market.

Hence, there is a need for an automated approach to this dynamic and ongoing planning problem that can both aid the construction of efficient schedules and enable fast changes to existing schedules in case of new or changed customer demands. Some commercial tools for optimisation of vessel fleet scheduling have been developed. Even so, many tramp operators still use experienced planners to manually route and schedule their fleets. We reflect on some of the reasons for this later in this paper; however, for now we note that further work is needed in this area. The main goal of this paper is therefore to provide an introduction to the general area of tramp ship routing and scheduling to facilitate further research on this topic.

Note that from an operations research point of view, the industrial ship routing and scheduling problem (ISRSP) can be viewed as a special case of the tramp ship routing and scheduling problem (TSRSP) where there are no spot cargoes. In such a case, the profit maximising tramp objective can be replaced by a cost minimisation objective as used in industrial shipping. Furthermore, compared to the fixed number of cargoes to carry in industrial shipping, the addition of spot cargoes in tramp shipping also yields greater flexibility suggesting greater financial impact from schedule optimisation within tramp shipping than within industrial shipping. The fact that the TSRSP can be viewed as a generalisation of the ISRSP, and that the financial impact from tramp ship routing and scheduling can be expected to be greater than that of industrial ship routing and scheduling, yields further motivation for focusing on tramp shipping as opposed to industrial shipping.

The remainder of the paper is organised as follows. In Section 2 we review literature on the TSRSP. In Section 3 we discuss in further details the main characteristics of the basic TSRSP and also provide examples of mathematical formulations for this problem. We also relate the TSRSP to vehicle routing problems and discuss some distinctions between these two related problems. In Section 4 we review the current status of research and applications within tramp ship routing and scheduling and use this to reflect on the possible future research directions. We argue that, rather than developing new solution methods for the basic TSRSP, the main research focus should now be to extend this basic problem to include additional real-world complexities. Section 5 contains a description of the common methods and tools used for solving tramp ship routing and scheduling problems and also relates these methods and tools to literature. Finally, in Section 6 we arrive at a conclusion and reflect on the directions for future research within tramp ship routing and scheduling.

2 Literature Review

As mentioned in Section 1, the three different basic operating modes, liner, tramp, and industrial, are not mutually exclusive. Especially tramp and industrial shipping are, as mentioned, very closely related, and to some extent, the TSRSP can be viewed as a generalisation of the ISRSP. Thereby, it can be difficult as well as meaningless to separate the literature on these two problems into two distinct groups. Rather, most work on tramp ship routing and scheduling can be considered relevant for research within industrial ship routing and scheduling, and vice versa. With research in these two fields dating as far back as the 1950s, and adding to this the literature on problems that constitute a mixture of the different operating modes, the total amount of literature relevant for tramp ship routing and scheduling is quite extensive. It is therefore not our aim to provide a comprehensive review of all literature relevant for the TSRSP. For this, we instead refer the reader to the four review papers Ronen (1983), Ronen (1993), Christiansen et al. (2004) and Christiansen et al. (2013), which, on a decade basis, have provided the research community with comprehensive reviews on the latest research on ship routing and scheduling within all operating modes. Here,

we instead review the very recent work and limit ourselves to work specifically on tramp ship routing and scheduling, though we recognize the relevance of work on problems from the two other operating modes and on problems that constitute a mixture of the different ship operating modes.

Even though the latest review paper was published in 2013, several papers related to tramp ship routing and scheduling have been published since then. A very welcomed addition to the literature on the TSRSP can be found in Hemmati et al. (2014) where they present a wide range of benchmark instances as well as an instance generator for both industrial and tramp ship routing and scheduling problems. The development of benchmark data for the tramp shipping community, just as we see it for vehicle routing and as has recently been developed for the liner shipping community (Brouer et al., 2014) and for single-product maritime inventory routing problems (Papageorgiou et al., 2014), can enable more research within tramp ship routing and scheduling. Benchmark data can also facilitate easy comparison of solution methods developed for similar problems. This could for instance allow methods developed for vehicle routing to be easily tested in a tramp ship setting even if vehicle routing researchers have no desire to devote a considerable amount of time to both gather and generate data that accurately reflects the situation in the shipping industry. Initial results using both a commercial mixed-integer programming solver and an adaptive large neighborhood search heuristic are provided for the benchmark instances in order to provide an idea of the difficulty of each instance.

Cóccola et al. (2015) present a novel column generation approach in which the conventional dynamic programming route-generator is replaced by a continuous time Mixed Integer Linear Programming (MILP) subproblem. In each iteration, multiple columns are generated in the subproblem by solving a continuous time precedence based MILP formulation. Computational results derive from five data instances based on real data from a chemical shipping company. On relatively small or tightly constrained problems the column generation approach from Cóccola et al. (2015) is able to find the optimal solution in a very short time. For more complex instances, their approach generates good feasible solutions in reasonable time. Thereby, their method outperforms both an exact optimisation model and some heuristic solution methods previously reported in the literature.

Fagerholt and Ronen (2013) note that most research within this area has focused on solving simplified versions of reality. They present and consolidate results for three practical problems that each extend these simplified problem versions by adding further complexities and opportunities. The first extension considers flexible cargo quantities, the second allows split cargoes (i.e each cargo is allowed to be split among several ships), while the third includes sailing speed optimisation. Their results show that by using advanced heuristics, these extensions can, despite the increased problem complexity, be solved to achieve significantly better solutions. We discuss these findings further in Section 4.2.

In line with the findings from Fagerholt and Ronen (2013), Stålhane et al. (2012) consider the TSRSP with split cargoes. They present a new path flow formulation and devise a Branch-Price-and-Cut procedure to solve the problem. Their column generation based solution method relies on a new subproblem that combines an elementary shortest path problem with resource constraints and a multi-dimensional knapsack problem used to determine the optimal cargo quantities. In order to solve this special subproblem, the authors develop a dynamic programming algorithm as well as a new dominance criterion for dominating partial paths. They further present some new valid inequalities for this TSRSP with split loads and complement these by adaptations of already existing inequalities from the literature. Their computational results show that their solution method outperforms existing methods for instances with a long planning horizon, and where either the average cargo size is large or the time windows are narrow, or both.

Castillo-Villar et al. (2014) present a Variable Neighborhood Search based heuristic procedure for solving the TSRSP with discretised time windows. They ignore spot cargoes and therefore seek to minimise cost rather than to maximise profit. The discretisation approach allows them to incorporate several practical extensions and they, also in line with the findings from Fagerholt and Ronen (2013), specifically investigate the inclusion of variable speed. Computational results on generated instances show that the heuristic is able to find reasonably good solutions within reasonable computation time. It is, however, only tested on smaller instances.

Magirou et al. (2015) also consider the topic of speed optimisation within the tramp shipping industry, though for a single vessel ignoring the interactions between vessels in the fleet. The

authors start by showing that dynamic programming can be used to determine the optimal speed of a single vessel facing a known, repeating sequence of voyages assuming that the freight rates between all origin destination pairs as well as fuel prices are known with certainty and that there are no constraints on vessel speed. It is then shown that this model can be extended to voyage selection on a graph of ports and that this voyage-speed selection principle developed for the deterministic case holds when stochastic freight rates are considered. In the stochastic setting two cases are considered. The first considers the situation in which the freight rate random variables are independent from one voyage to the next and whose distributions are the same for each port of origin, while the second investigates what happens when the freight rates depend on the state of the market, which is modelled as a Markovian random variable. The authors adapt standard solution methods for Markovian decision processes and provide a comparison between models with a discounted net revenue criterion and those with an average non-discounted profit.

Norstad et al. (2015) present a problem that they themselves characterize as a fleet deployment problem somewhere between liner and tramp shipping. However, their context is very similar to the TSRSP and their mathematical formulation of the problem is very similar to the models for the TSRSP presented in Section 3.2. Therefore, we venture to include the work from Norstad et al. (2015) in this tramp specific review. Norstad et al. (2015) add voyage separation requirements (VSRs) to the routing and scheduling problem in order to enforce a minimum time spread between similar voyages. This is done in order to ensure that consecutive voyages are performed fairly evenly spread in time to reduce inventory costs for the charterer. This is one way of balancing the conflicting objectives of profit maximisation for the ship operator and minimisation of inventory costs for the charterer. In this respect, the incorporation of VSRs correspond to a crude way of viewing the TSRSP in the broader context of the supply chain. They present an arc flow formulation that is solved directly using commercial software. They also present a path flow formulation, which is solved by a priori path generation (APPG) and a commercial solver for the final problem. Their computational results show that both formulations can be used to solve smaller instances, while the path flow formulation can also be used to solve problems of more realistic sizes. However, neither of these methods are applicable for larger and more complex problem instances. Results also show that the inclusion of VSRs can significantly improve the spread of the voyages and at only marginal profit reductions.

Similar to Norstad et al. (2015), Vilhelmsen and Lusby (2014) consider the incorporation of VSRs into the basic TSRSP. They present a new mixed integer programming formulation for this problem and develop a new, exact solution method for it. This method is a Branch-and-Price procedure based on a Dantzig-Wolfe decomposition of the original formulation. In the master problem, the VSRs are relaxed along with the binary variable restrictions. They present a new tailor made time window branching scheme that can restore feasibility with respect to the VSRs, and to some extent also restore integrality. Since this branching scheme is not complete with respect to integrality, they complement it by constraint branching, which utilises the strong integer properties of the master problem constraint matrix, to efficiently eliminate fractionality. They use a dynamic programming algorithm to solve the subproblems. Computational results show that the developed algorithm is able to find very good if not optimal solutions extremely fast, although one instance requires longer time. A comparison of their method to the APPG method from Norstad et al. (2015) shows that for all but one instance, their optimal solutions are obtained in the same or shorter time than what the APPG method uses.

Stålhane et al. (2014) also relate the TSRSP to the broader context of the supply chain. They do this by combining traditional tramp shipping with a vendor managed inventory (VMI) service in an attempt to challenge the traditional contracts of affreightment. The authors present an arc flow formulation for this problem as well as a path flow formulation. The path flow formulation is solved by a hybrid method that combines a priori path generation of all feasible routes with Branch-and-Price to generate schedules for these routes. Larger instances are solved using a heuristic version of path generation. Computational results show that the profit and efficiency of the supply chain can be significantly increased by using vendor managed inventory services compared to the traditional contracts of affreightment. Computational results also show that the heuristic can significantly speed up computation time compared to the exact method, and at only a small reduction in solution quality. However, even with the heuristic approach they are only able to solve small sized instances where only few VMI services are introduced and running times increase drastically from

seconds to days with increases in problem size or in the number of VMIs.

Hemmati et al. (2015) continue the work from Stålhane et al. (2014) in order to develop a more efficient method for the same problem, i.e. for a TSRSP with inventory constraints. The authors present a new heuristic method that works in two phases by first converting the inventory constraints into cargoes, and then solving the resulting TSRSP with an adaptive large neighborhood search heuristic. This procedure continues iteratively by changing the cargoes derived from the inventory constraints and resolving the resulting TSRSP. Computational results show that the heuristic can solve realistically sized instances of the TSRSP with inventory constraints in reasonable time, and that when the number of inventory pairs is large, the heuristic is much faster than the methods from Stålhane et al. (2014) and generally finds better solutions. The authors also introduce new benchmark instances for this problem of more realistic sizes than those solved in Stålhane et al. (2014). On these new and more realistically sized instances, results show that the value of introducing these VMIs is not as large as previously indicated.

Vilhelmsen et al. (2014b) explore the effects of incorporating bunker planning into the basic TSRSP with full shiploads. They provide a description as well as a mixed integer programming formulation for this new problem. The model extends standard tramp formulations by incorporating variations in bunker prices, port costs incurred when bunkering, as well as time consumption of bunkering. They devise a solution method based on column generation with dynamic programming to solve the subproblems. The subproblems are solved heuristically by discretising the continuous bunker purchase variables. Based on industry data, they develop instance generators that independently generate cargoes and bunker prices. Computational results show that the integrated planning approach can increase profits, and that the decision of which cargoes to carry, and on which ships, is affected by the bunker integration and by changes in the bunker prices.

Meng et al. (2015) also consider the joint problem of routing, scheduling and bunkering a tramp fleet carrying full shiploads. In contrast to Vilhelmsen et al. (2014b), they assume that speed and costs are independent of the ship load and that ships are not allowed to detour to purchase bunker. Also, while the bunker prices considered in Vilhelmsen et al. (2014b) fluctuate over time, Meng et al. (2015) assume that bunker prices are fixed over the entire planning horizon. This assumption of fixed prices allows them to develop an efficient method for determining the optimal bunkering plan for a given route. They present a tailored Branch-and-Price method to solve the problem and embed the whole thing in a rolling horizon manner that incorporates an estimated opportunity cost of each ship. Computational results from Meng et al. (2015) on generated instances also show that the integrated planning approach can increase profits.

We note that the first review paper (Ronen, 1983) contains only two references on tramp shipping, while the second (Ronen, 1993) does not contain any. The third review paper (Christiansen et al., 2004) lists five new tramp references, though some of these are for problems with a mix of tramp and industrial shipping. Finally, the fourth review paper (Christiansen et al., 2013) lists around 30 papers related to the TSRSP, though some of these are specifically for industrial shipping. These numbers show a clear trend of increased research interest within the TSRSP, and we note that we above listed 12 references on the TSRSP from just within the last few years.

3 The Tramp Ship Routing and Scheduling Problem

In this section we use Section 3.1 to describe the main characteristics of the TSRSP. We also discuss some of the operator specific characteristics of the problem and give some pointers to both recent and earlier work that together cover the different combinations of operator specific characteristics. In Section 3.2 we present two mathematical models for two specific versions of the TSRSP, while we use Section 3.3 to reflect on the similarities and distinctions between the TSRSP and vehicle routing problems.

3.1 Problem Description

The fleet size and mix is determined at the strategic planning level. In the TSRSP we therefore assume a fixed heterogeneous fleet comprised of ships of different sizes, load capacities, bunker consumptions, speeds, and other characteristics. Since ships operate around the clock, some ships

can be occupied with prior tasks when planning starts, so each ship is further characterized by the time it is available for service and the location at which it is when it becomes available. The characteristics of each ship determine which cargoes, ports and canals it is compatible with, e.g. the draft of a ship can prohibit it from entering a shallow port and thereby make the ship incompatible with all cargoes being either loaded or unloaded in this specific port. For some ships there can also be maintenance requirements during the planning horizon, and these must be respected in the scheduling process.

During the planning horizon, the tramp operator is obliged to transport a given list of contracted cargoes and can then turn to the spot market to derive additional revenue from spot cargoes if fleet capacity allows it, and if it is profitable. Each cargo is mainly characterized by the quantity to be transported, the revenue obtained from transporting it, and the loading and unloading port. There is also a ship specific service time in port for loading and unloading, and a time window giving the earliest and latest start for loading. In some cases there is also a time window for unloading.

Since we consider a fixed fleet, we can disregard the fixed setup costs and focus on the variable operating costs which consist mainly of ship dependent fuel and port costs. Other costs can be relevant depending on the specific operator.

The objective of the TSRSP is then to create a profit maximizing set of fleet schedules, one for each ship in the fleet, where a schedule is a sequence and timing of port calls representing cargo loading and unloading. The optimal solution therefore combines interdependent decisions on which optional cargoes to carry, the assignment of cargoes to ships, and the optimal sequence and timing of port calls for each ship.

The above problem description is generic, and further operator specific characteristics are needed to properly model and solve the problem. Below we state some key operator characteristics.

- **Full shiploads or multiple cargoes (mixed loads):** Full shiploads correspond to the case where a ship can at most carry one cargo at a time, while multiple cargoes corresponds to the case where each ship can carry multiple cargoes at once. The full shipload case yields a much simpler mathematical formulation since the tasks of loading and unloading each cargo can be aggregated into one task since no other cargoes can be handled in between. This also means that capacity constraints as well as precedence and coupling constraints are implicitly handled in the preprocessing phase.
- **Fixed or flexible cargo sizes:** Quite naturally, the fixed cargo size case refers to the case where each cargo size is a fixed quantity whereas the flexible case refers to the situation where each cargo size is given in an interval. This means that in the flexible cargo case, the extra complexity of deciding the specific amount of each cargo to transport is added to the problem. Cargo revenues are, in the flexible case, defined as an amount per unit transported. Note that in the full shipload case, the same model can be used for both fixed and flexible cargo quantities since the specific cargo quantity transported by each ship can be determined from ship capacity during preprocessing along with the corresponding revenue.
- **Mixable or multiple (non-mixable) products:** In the case of mixable products, different cargoes are compatible with each other whether they consist of the same product or of multiple mixable ones. In either case, different cargoes can be loaded onboard a ship with no consideration to the type of products already onboard. In the case of multiple (non-mixable) products, different cargoes are not necessarily compatible and must be transported in different tanks onboard the ship in order to be onboard simultaneously. Thereby, additional complexity is added to the problem since the specific allocation of each cargo to tanks must also be decided. The multiple product case also leads to further problem characteristics as the tanks can be of either fixed or flexible sizes.
- **Disallowing or allowing spot vessels:** If the capacity of the fixed fleet is not sufficient to carry the contracted cargoes, it can be allowed to charter in outside spot vessels to transport these cargoes at a specified cost. There can even be cases where fleet capacity is actually sufficient, but where it is simply profitable to charter in outside vessels to transport some contract cargoes thereby freeing up fleet capacity to instead transport spot cargoes.

A typical example of the full shipload case is the transportation of crude oil. In the literature, examples of TSRSPs for full shiploads can be found in Appelgren (1969, 1971), Norstad et al. (2015) and Vilhelmsen et al. (2014b).

Multiple cargoes, on the other hand, are common within dry bulk shipping and also for transportation of chemicals and refined oil products. The multiple cargo case can be found in e.g. Korsvik et al. (2010), Andersson et al. (2011b) and C occola et al. (2015). Note that fleet size can only be used as a crude estimate of problem complexity since the planning problem for a large fleet sailing full shiploads can easily be less complex than that for a smaller fleet carrying multiple cargoes.

Flexible cargo sizes are often seen within transportation of bulk products. When such products are transported on a recurrent basis, as under a CoA, flexibility is often given with respect to the exact amount of cargo to transport each time. This results in what is known as a *More Or Less Owner's Option* contract in which there is a target amount to transport and a certain flexibility to vary from this target, e.g. $\pm 10\%$. Note though that the ship operator is of course paid per unit delivered and not according to the target amount. In particular, transportation of liquid products more or less implies flexible cargo sizes in order to prevent sloshing in partially empty tanks during sailing, and also to ensure ship stability. Although flexible cargo sizes relate to most operators transporting bulk products, and in particular liquid products, this problem characteristic is neglected in most work on the TSRSP. We do however find a few examples on flexible cargoes, e.g. Br onmo et al. (2007b), Br onmo et al. (2010) and Korsvik and Fagerholt (2010).

Within transportation of liquid products we also find transportation of chemicals, and this segment is a typical example of the multiple product case. A chemical tanker can have as many as 50 different tanks and hazardous materials regulations play a major role when allocating the products to the different tanks. E.g. products in neighboring tanks must be non-reactive and incompatible products must not succeed each other in a tank unless it is cleaned. The added complexity of the multiple (non-mixable) products is also rarely considered in the literature. A few examples of work in this area can be found in Fagerholt and Christiansen (2000a), Kobayashi and Kubo (2010), Oh and Karimi (2008) and Neo et al. (2006), although the latter only considers one ship. The work presented in Vilhelmsen et al. (2014a) also relates to the multiple (non-mixable) products case, though it focuses on the solution of a subproblem that can facilitate the incorporation of the tank allocation aspect in the TSRSP.

As far as we know, the use of spot vessels is not typical for any specific shipping segment but is a much more operator specific problem characteristic. Furthermore, we note that the inclusion of spot vessels only results in minor changes to the mathematical formulation of the problem, and most often do not complicate the solution procedure further. Therefore, the inclusion of this characteristic does not on its own constitute an interesting research area. Rather, it is included in a wide variety of papers, depending on the real life application considered in those papers. We note that spot vessels are included in the work presented in Vilhelmsen and Lusby (2014), where their presence does, however, complicate the solution procedure.

3.2 Mathematical Formulation

The existing research on tramp ship routing and scheduling covers a broad range of problem types using different combinations of the above operator specific characteristics. As can be seen in Christiansen et al. (2007), the different combinations lead to associated mathematical formulations differing in both size and complexity. It is not the aim of this section to present all the different mathematical formulations derived from these main characteristics. Nor is it the aim to repeat the specific mathematical formulations presented in Christiansen et al. (2007). However, it is the aim to provide the reader with a basic understanding of the structure of the TSRSP. Therefore, we here present two arc flow formulations for two relatively simple versions of the TSRSP. The first formulation is for full shiploads and in this case it does not really matter whether we assume fixed or flexible cargo sizes, or if we assume mixable or multiple products. In order to provide an idea of the diversity of TSRSP models and of the effect of changing a single operator characteristic, we also present the mathematical formulation of the TSRSP with multiple cargoes of fixed size and mixable products. In both of these formulations we exclude the use of spot vessels. However, in Section 3.2.3 we briefly describe how to incorporate such vessels into the two previous formulations.

3.2.1 Formulation with full shiploads

Let \mathcal{V} be the set of ships, and let \mathcal{N} denote the set of cargoes to be transported during the planning horizon. We partition the cargo set \mathcal{N} into the two smaller and disjoint sets, \mathcal{N}_C and \mathcal{N}_O , containing, respectively, the contracted cargoes and the optional spot cargoes.

In order to define the problem on ship specific graphs, we define an origin node and a destination node for each ship $v \in \mathcal{V}$ and denote these $o(v)$ and $d(v)$ respectively. The origin node corresponds to the geographical location of the ship when planning starts, while the destination node is artificial and simply corresponds to the geographical location of ship v at the end of the planning horizon. We can also represent each cargo $i \in \mathcal{N}$ as a network node, and this node corresponds to the full transportation of cargo i , i.e. to both the loading port and the unloading port of cargo i . Due to port and cargo compatibility, capacity and time requirements, as well as other restrictions, not all ships can transport all cargoes. Therefore, we define \mathcal{N}^v as the set of nodes that ship v can visit, i.e. the nodes $o(v)$ and $d(v)$ as well as all nodes corresponding to cargoes, that can be transported by ship v . We further define ship specific arc sets, \mathcal{A}^v , containing all arcs $(i, j) \in \{\mathcal{N}^v \times \mathcal{N}^v\}$ that are feasible for ship v to traverse. If ship v uses an arc $(i, j) \in \{\mathcal{N}^v \times \mathcal{N}^v\}$ it corresponds to ship v transporting cargo i just before transporting cargo j . Note that \mathcal{A}^v also contains the arc $(o(v), d(v))$ corresponding to ship v being idle during the entire planning horizon.

For each cargo $i \in \mathcal{N}^v$ we have a time window $[a_i^v, b_i^v]$ describing the earliest and latest time to start service for this cargo, when transported by ship v . For $o(v)$ this window is collapsed into the time, ship v is available for service. For any arc $(i, j) \in \mathcal{A}^v$, T_{ij}^v denotes the fixed time from arrival at the loading port of cargo i to the arrival at the loading port of cargo j and includes any port time for loading and unloading cargo i as well as sailing time for transportation of cargo i and ballast sailing time from the unloading port of cargo i to the loading port of cargo j . For each arc $(i, j) \in \mathcal{A}^v$ we also have a ship specific profit, P_{ij}^v , which takes into account the revenue incurred from transportation of cargo i , the ship dependent port and sailing costs from transportation of cargo i on ship v , and finally the cost of traveling ballast with ship v from the unloading port of cargo i to the loading port of cargo j .

For the mathematical formulation we define binary flow variables x_{ij}^v for $v \in \mathcal{V}$, $(i, j) \in \mathcal{A}^v$ that are equal to 1, if ship v transports cargo i just before transporting cargo j , and 0 otherwise. The start time for service for each cargo is also variable; hence we define time variables t_i^v for each $v \in \mathcal{V}$ and $i \in \mathcal{N}^v$.

We can now give an arc flow formulation of this simple version of the TSRSP:

$$\max \sum_{v \in \mathcal{V}} \sum_{(i,j) \in \mathcal{A}^v} P_{ij}^v x_{ij}^v \quad (1)$$

s.t.

$$\sum_{v \in \mathcal{V}} \sum_{j \in \mathcal{N}^v} x_{ij}^v = 1, \quad \forall i \in \mathcal{N}_C, \quad (2)$$

$$\sum_{v \in \mathcal{V}} \sum_{j \in \mathcal{N}^v} x_{ij}^v \leq 1, \quad \forall i \in \mathcal{N}_O, \quad (3)$$

$$\sum_{j \in \mathcal{N}^v} x_{o(v)j}^v = 1, \quad \forall v \in \mathcal{V}, \quad (4)$$

$$\sum_{i \in \mathcal{N}^v} x_{ij}^v - \sum_{i \in \mathcal{N}^v} x_{ji}^v = 0, \quad \forall v \in \mathcal{V}, j \in \mathcal{N}^v \setminus \{o(v), d(v)\}, \quad (5)$$

$$\sum_{i \in \mathcal{N}^v} x_{id(v)}^v = 1, \quad \forall v \in \mathcal{V}, \quad (6)$$

$$x_{ij}^v (t_i^v + T_{ij}^v - t_j^v) \leq 0, \quad \forall v \in \mathcal{V}, (i, j) \in \mathcal{A}^v, \quad (7)$$

$$a_i^v \leq t_i^v \leq b_i^v, \quad \forall v \in \mathcal{V}, i \in \mathcal{N}^v, \quad (8)$$

$$x_{ij}^v \in \{0, 1\}, \quad \forall v \in \mathcal{V}, (i, j) \in \mathcal{A}^v. \quad (9)$$

The objective function (1) maximises the joint profit from all ships in the fleet. Constraints (2) and (3) ensure that all contract cargoes are transported by exactly one ship, and that all spot cargoes are transported by at most one ship. Constraints (4) and (6) together with the flow conservation

constraints in (5) ensure that each ship is assigned a schedule starting at the origin node and ending at the destination node. Constraints (7) ensure that if ship v transports cargo i directly before cargo j , the time for start of service for cargo j cannot begin before service start time for cargo i plus port and travel time for transportation of cargo i plus ballast travel time with ship v from the unloading port of cargo i to the loading port of cargo j . Since waiting time is allowed, the constraints have an inequality sign. In constraints (8), the service start time for ship v for cargo i , t_i^v , is forced to be within its time window, thereby also ensuring that no ship can start its schedule before it is available for service. Note that if ship v does not transport cargo i , the time variable t_i^v has no effect and is in some sense artificial. Finally, the flow variables are restricted to be binary in (9).

We note that constraints (7) are nonlinear and should in fact be:

$$x_{ij}^v = 1 \quad \Rightarrow \quad t_i^v + T_{ij}^v \leq t_j^v \quad \forall v \in \mathcal{V}, (i, j) \in \mathcal{A}^v. \quad (10)$$

However, as long as we require the x_{ij}^v variables to be binary, we can easily linearise these constraints by following the approach presented in Desrosiers et al. (1995). This approach introduces a large constant M_{ij}^v for each ship $v \in \mathcal{V}$ and each arc $(i, j) \in \mathcal{A}^v$. This constant, M_{ij}^v , must be at least as large as the largest value that $t_i^v + T_{ij}^v - t_j^v$ can take, i.e. $M_{ij}^v \geq b_i^v + T_{ij}^v - a_j^v$. The linearised constraints then become

$$t_i^v + T_{ij}^v - t_j^v \leq M_{ij}^v(1 - x_{ij}^v), \quad \forall v \in \mathcal{V}, (i, j) \in \mathcal{A}^v.$$

Thereby, the arc flow formulation (1)-(9) can easily be linearised. However, this is only relevant if we want to solve the arc flow formulation by use of standard commercial optimisation software for mixed integer linear programming, and for most real life sized instances, this approach will be too time consuming.

3.2.2 Formulation with multiple cargoes of fixed size

In the full shipload case each cargo i could be represented by a single node in the underlying network corresponding to loading, transporting and unloading the cargo. However, the transition from full shiploads to multiple cargoes means that other cargoes can potentially be loaded or unloaded in between loading and unloading of cargo i . Therefore, in the multiple cargo case each cargo i is represented by two nodes in the underlying network, namely a node i corresponding to the loading port of cargo i and a node $N + i$ corresponding to the unloading port of cargo i , where N is the number of cargoes. The nodes corresponding to loading ports are denoted \mathcal{N}_P , i.e. pickups, while the nodes corresponding to unloading ports are denoted \mathcal{N}_D , i.e. discharges/deliveries. This means that $\mathcal{N}_P = \{1, \dots, N\}$, $\mathcal{N}_D = \{N + 1, \dots, 2N\}$ and now \mathcal{N} denotes the full cargo node set, i.e. $\mathcal{N} = \mathcal{N}_P \cup \mathcal{N}_D$. Similar to the full shipload case, we partition the loading nodes into two disjoint sets for contract and optional cargoes respectively, i.e. $\mathcal{N}_P = \mathcal{N}_C \cup \mathcal{N}_O$. The ship set \mathcal{V} as well as the artificial origin and destination nodes remain as in the full shipload case. Likewise, \mathcal{N}^v still contains the nodes that ship v can visit, though this node set now counts both loading and unloading nodes as well as the origin and destination nodes. The ship specific loading nodes, $\mathcal{N}_P \cap \mathcal{N}^v$, are denoted \mathcal{N}_P^v while the corresponding ship specific unloading nodes, $\mathcal{N}_D \cap \mathcal{N}^v$, are denoted \mathcal{N}_D^v . \mathcal{A}^v now contains the arcs that ship v can traverse while adhering to precedence constraints for loading and unloading of the same cargo as well as capacity, time and other relevant constraints.

In this model we can allow a ship specific time window on both loading and unloading by associating a time window $[a_i^v, b_i^v]$ with all nodes $i \in \mathcal{N}^v$, $v \in \mathcal{V}$, i.e. on both loading and unloading nodes. T_{ij}^v , $v \in \mathcal{V}$, $(i, j) \in \mathcal{A}^v$ now denotes the fixed time from arrival at the port associated with node i to the arrival at the port associated with node j with ship v . With each cargo i we now associate a revenue R_i from transporting it. Furthermore, with each arc $(i, j) \in \mathcal{A}^v$ we associate the variable cost C_{ij}^v corresponding to port costs at the port of node i and travel costs when sailing directly from the port of node i to the port of node j with ship $v \in \mathcal{V}$. Since multiple cargoes can now be onboard the same ship simultaneously, we must add capacity constraints to the model. For this we denote the fixed cargo quantity of cargo i by Q_i and the capacity of ship $v \in \mathcal{V}$ by V_{CAP}^v .

The variables are similar to before, though x_{ij}^v , $v \in \mathcal{V}$, $(i, j) \in \mathcal{A}^v$ is now 1 if ship v visits the port of node i right before visiting the port of node j , and 0 otherwise. Similarly, t_i^v , $v \in \mathcal{V}$, $i \in \mathcal{N}^v$

denotes the start time for service at the port associated with node i . To keep track of the total load onboard each ship we use variables $l_i^v, v \in \mathcal{V}, i \in \mathcal{N}^v \setminus \{d(v)\}$ which denote the total load onboard ship v just after completing service at node i . We assume that the ship is empty when planning starts.

We can now present the arc flow formulation for the TSRSP with multiple cargoes of fixed size and no spot vessels.

$$\max \sum_{v \in \mathcal{V}} \sum_{i \in \mathcal{N}_P^v} R_i x_{ij}^v - \sum_{v \in \mathcal{V}} \sum_{(i,j) \in \mathcal{A}^v} C_{ij}^v x_{ij}^v \quad (11)$$

s.t.

$$\sum_{v \in \mathcal{V}} \sum_{j \in \mathcal{N}^v} x_{ij}^v = 1, \quad \forall i \in \mathcal{N}_C, \quad (12)$$

$$\sum_{v \in \mathcal{V}} \sum_{j \in \mathcal{N}^v} x_{ij}^v \leq 1, \quad \forall i \in \mathcal{N}_O, \quad (13)$$

$$\sum_{j \in \mathcal{N}_P^v \cup \{d(v)\}} x_{o(v)j}^v = 1, \quad \forall v \in \mathcal{V}, \quad (14)$$

$$\sum_{i \in \mathcal{N}^v} x_{ij}^v - \sum_{i \in \mathcal{N}^v} x_{ji}^v = 0, \quad \forall v \in \mathcal{V}, j \in \mathcal{N}^v \setminus \{o(v), d(v)\}, \quad (15)$$

$$\sum_{i \in \mathcal{N}_D^v \cup \{o(v)\}} x_{id(v)}^v = 1, \quad \forall v \in \mathcal{V}, \quad (16)$$

$$x_{ij}^v (t_i^v + T_{ij}^v - t_j^v) \leq 0, \quad \forall v \in \mathcal{V}, (i, j) \in \mathcal{A}^v, \quad (17)$$

$$a_i^v \leq t_i^v \leq b_i^v, \quad \forall v \in \mathcal{V}, i \in \mathcal{N}^v, \quad (18)$$

$$x_{ij}^v (l_i^v + Q_j - l_j^v) = 0, \quad \forall v \in \mathcal{V}, (i, j) \in \mathcal{A}^v \mid j \in \mathcal{N}_P^v, \quad (19)$$

$$x_{i, N+j}^v (l_i^v - Q_j - l_{N+j}^v) = 0, \quad \forall v \in \mathcal{V}, (i, N+j) \in \mathcal{A}^v \mid j \in \mathcal{N}_P^v, \quad (20)$$

$$l_{o(v)}^v = 0, \quad \forall v \in \mathcal{V}, \quad (21)$$

$$\sum_{j \in \mathcal{N}^v} Q_i x_{ij}^v \leq l_i^v \leq \sum_{j \in \mathcal{N}^v} V_{\text{CAP}}^v x_{ij}^v, \quad \forall v \in \mathcal{V}, i \in \mathcal{N}_P^v, \quad (22)$$

$$0 \leq l_{N+i}^v \leq \sum_{j \in \mathcal{N}^v} (V_{\text{CAP}}^v - Q_i) x_{N+i,j}^v, \quad \forall v \in \mathcal{V}, i \in \mathcal{N}_P^v, \quad (23)$$

$$t_i^v + T_{i, N+i}^v - t_{N+i}^v \leq 0, \quad \forall v \in \mathcal{V}, i \in \mathcal{N}_P^v, \quad (24)$$

$$\sum_{j \in \mathcal{N}^v} x_{ij}^v - \sum_{j \in \mathcal{N}^v} x_{j, N+i}^v = 0, \quad \forall v \in \mathcal{V}, i \in \mathcal{N}_P^v, \quad (25)$$

$$x_{ij}^v \in \{0, 1\}, \quad \forall v \in \mathcal{V}, (i, j) \in \mathcal{A}^v. \quad (26)$$

The objective function (11) again maximises the joint profit from all ships in the fleet while constraints (12) and (13) ensure that all contract cargoes are transported by exactly one ship, and that all spot cargoes are transported by at most one ship. Constraints (14) and (16) together with the flow conservation constraints in (15) still ensure that each ship is assigned a schedule starting at the origin node and ending at the destination node, though now it is required that the first task in a schedule is a pickup while the last task must be a discharge. Constraints (17) ensure that if ship v visits node i directly before node j , the time for start of service at the port of node j cannot begin before service start time at the port of node i plus port time at the port of node i plus travel time from the port of node i to the port of node j with ship v . Again, we have an inequality sign since waiting time is allowed. In constraints (18), the service start time for ship v at the port of node i , t_i^v , is forced to be within its time window, thereby also ensuring that no ship can start its schedule before it is available for service. Note that now we also have time windows on ports corresponding to unloading. Similar to before, if ship v does not transport cargo i , the time variables t_i^v and t_{i+N}^v have no effect. Constraints (19) ensure the correct relationship between the binary flow variables and the ship load at each of the loading ports while constraints (20) do the same for the unloading ports. As mentioned, we assume that the ship is empty when planning starts and this is captured in constraints (21). The feasible interval for the ship load at

each loading port is given by constraints (22) while constraints (23) give the same for the unloading ports. Constraints (24) ensure that a cargo cannot be unloaded before it has been loaded, and together with the coupling constraints (25) they ensure that the same ship performs both tasks. Note that constraints (23) can actually be omitted from the formulation due to constraints (22), (24) and (25). Finally, in constraints (26) the flow variables are, as before, restricted to be binary.

The nonlinear constraints (17), (19) and (20) can be linearised as described for the full shipload formulation. However, we again note that for most real life sized instances this is not relevant since the use of standard commercial optimisation software to solve the full formulation will be too time consuming.

3.2.3 Allowing the use of spot vessels

If it is allowed to charter in outside spot vessels to transport some of the contract cargoes, the above formulations can easily be modified to take this into account. For these modifications we let P_i^s denote the profit obtained from transporting cargo $i \in \mathcal{N}_C$ with a spot vessel. Note that this profit may very well be negative. Also, we introduce binary variables s_i , $i \in \mathcal{N}_C$ that are equal to 1 if cargo i is transported by a spot vessel, and equal to 0 otherwise.

In the full shipload formulation from Section 3.2.1 the objective function (1) and constraints (2) must be modified to the following:

$$\max \sum_{v \in \mathcal{V}} \sum_{(i,j) \in \mathcal{A}^v} P_{ij}^v x_{ij}^v + \sum_{i \in \mathcal{N}_C} P_i^s s_i \quad (27)$$

s.t.

$$\sum_{v \in \mathcal{V}} \sum_{j \in \mathcal{N}^v} x_{ij}^v + s_i = 1, \quad \forall i \in \mathcal{N}_C, \quad (28)$$

$$s_i \in \{0, 1\}, \quad \forall i \in \mathcal{N}_C. \quad (29)$$

The objective function (27) still maximises profit, though now for both the fixed fleet and for the spot vessels used. Constraints (28) ensure that each contract cargo is transported by either a ship in the fixed fleet or by a spot vessel. Finally, constraints (29) enforce binary restrictions on the spot vessel variables.

In the formulation with multiple cargoes from Section 3.2.2 the objective function (11) and constraints (12) must similarly be modified to the following:

$$\max \sum_{v \in \mathcal{V}} \sum_{i \in \mathcal{N}_P^v} R_i x_{ij}^v - \sum_{v \in \mathcal{V}} \sum_{(i,j) \in \mathcal{A}^v} C_{ij}^v x_{ij}^v + \sum_{i \in \mathcal{N}_C} P_i^s s_i \quad (30)$$

s.t.

$$\sum_{v \in \mathcal{V}} \sum_{j \in \mathcal{N}^v} x_{ij}^v + s_i = 1, \quad \forall i \in \mathcal{N}_C, \quad (31)$$

$$s_i \in \{0, 1\}, \quad \forall i \in \mathcal{N}_C. \quad (32)$$

3.3 Ship Routing vs. Vehicle Routing

As can be noticed from the problem description in Section 3.1 and seen from the mathematical formulations in Section 3.2, the TSRSP is closely related to the Vehicle Routing Problem (VRP) and its many variants for which we refer the reader to Toth and Vigo (2002). Looking through literature, it is also quite clear that modelling approaches as well as solution methods for these two problems share great similarities, and that researchers within each of the two transportation modes have benefited from research within the other transportation mode. Literature also shows that research on the TSRSP has lagged far behind that of the VRP, though recent years have seen a huge increase in the amount of literature on the TSRSP. Therefore, the exchange of research ideas has most likely been far from equally balanced between the two research communities. However, we note that one of the most successful solution methods from the vehicle routing literature is based on decomposition and dynamic column generation, and the application of this method for a

pickup and delivery problem with time windows was first studied by Appelgren (1969) for a tramp ship routing and scheduling application.

Even though the TSRSP is closely related to the VRP and its many variants, there are however important differences that facilitate the development of industry specific solution methods. Ronen (1983, 2002) elaborate on the operational differences between ships and trucks while Christiansen et al. (2004) add to this list of differences as well as reflect on similarities and differences between ships and trains, and ships and aircrafts. Rather than repeating it here, we refer the reader to Ronen (1983, 2002) and Christiansen et al. (2004) for an extensive discussion on the differences between maritime routing and scheduling problems, and those of other transportation modes. Below we list a few of these differences including one specific for the tramp shipping industry.

- **Continuous operation:** As opposed to trucks, ships operate around the clock. This means that ship schedules usually do not have periods of idleness to absorb any delays. It also means that ships do not only have different starting positions but also different starting times, as some ships can be occupied with prior tasks when planning begins.
- **No common depot:** Even in multi-depot versions of the VRP, vehicles must return to their home depot, whereas ships do not have to return to their starting point.
- **Compatibility issues:** In shipping, compatibility issues can arise between ships and cargoes, ships and ports, and ships and canals due to equipment, capacity, draft, size etc. Even the flag of the ship can prevent it from entering ports in countries that have political or cultural issues with the nation corresponding to the flag of the ship. Even more complicating, the draft of the ship is affected by the current load of the ship, and since this affects its compatibility with both ports and canals, the compatibility between ship and cargoes, ports and canals can be affected by the current onboard cargo.
- **Optional cargoes:** Specific for the TSRSP, the distinction between contract cargoes and optional spot cargoes leads to a priority on cargoes not used in standard vehicle routing problems where all customers must be serviced at minimum cost. In contrast, the tramp objective is to maximize profit as in the less known Pickup and Delivery Selection Problem, see Schönberger et al. (2003).

4 Status and Perspectives of the TSRSP

In this section we first review the current status of the TSRSP in Section 4.1, and afterwards discuss a direction for current and future research within the area in Section 4.2. Finally, in Section 4.3 we present some literature consistent with and relevant for this research direction.

4.1 Current Status of the TSRSP

As noted in Section 2, recent years have shown a dramatic increase in the amount of research not just within maritime transportation in general but also specifically within tramp ship routing and scheduling. Complemented by the simultaneous improvements in both software and hardware, this has made the development of both heuristic and exact methods reach a level of efficiency that renders many realistic sized instances of the TSRSP now solvable within reasonable time.

Even so, many tramp operators still use experienced planners to manually route and schedule their fleets despite the increased need for planning aid from increased fleet sizes. It is therefore only natural to reflect on the reasons for this. Below we list some of the reasons we believe are responsible for this remaining gap between research and implementation.

- **Conservative industry:** The shipping industry has a long and proud tradition of using planners with practical experience from a background in shipping rather than a more academic background. Therefore, most planners are very skeptical towards optimisation-based tools not to mention reluctant to facilitate development of systems that could eventually make the company less dependent on them. Fagerholt (2004) presents some experience from the development and implementation of a decision support system for vessel fleet scheduling,

and also describes the struggle to convince the planners of the value of such a system. At the same time, the conservative nature of the industry has made many shipping companies unwilling to share data for research projects, and without real life data, the gap between research and implementation naturally remains.

- **Industry under pressure:** Previously the research and technology were not at an adequate level to facilitate implementation of optimisation-based tools, and now that they are adequate, the industry is under such pressure that it is hard to find resources for anything that is not immediately income generating. Certainly, it can be hard to justify the allocation of resources to a research project that might in the future turn out to be valuable, when the company is at the same time struggling to create revenues matching just their minimum operating costs.
- **Inability to plan ahead:** Uncertainty is known to play a big part in maritime optimisation where planners face a constantly changing environment with large daily variations in demand and many unforeseen events caused, among other things, by changing weather conditions. This uncertainty coupled with the long voyages spanning several days and sometimes even weeks, makes it hard to plan far ahead. The financial impact, not to mention loss of goodwill, from not adhering to agreed cargo transportation is simply too big to allow planning far ahead. For some operators the uncertainty is so high that they cannot plan even a single voyage ahead. Instead they simply wait for a ship to become idle and then assign it the currently best transportation request. This can for instance be the case in transportation of refined oil products where the unloading port and date can be unknown right up until actual unloading. A loaded ship can simply wait at sea until oil prices in ports reach a satisfactory level. This can make the combinatorial puzzle, of finding the best match of ships to cargoes, seem simple and make it even harder for conservative planners to see the potential in optimisation-based tools.
- **Dynamic nature of problem:** The fact, that the basic TSRSP can be solved, is naturally a requirement for the development of planning tools in this area. However, the basic TSRSP is a static and simplified version of reality. As already mentioned, there is great uncertainty in maritime transportation; hence, it can be necessary to continuously solve a modified version of the problem. This means that the fleet information must continuously be updated as well as cargo information. Note that even things such as the changing tidal conditions can affect the solution to the problem and create cause for reoptimisation. Similarly, the planning tool must somehow be linked to information from the spot market and continuously update these data. Overall, this means that the optimisation component itself will in many cases be just a very small part of the full planning tool and hence, implementation will lag behind that of research within methods for solving the TSRSP.
- **Modelling issues:** Most mathematical models are simplified versions of reality, but in shipping this is even more so. Aside from operational considerations completely ignored in the modelling process, those considerations that have been included in the models have been greatly simplified. In shipping there are a lot of constraints that are simply so difficult, if not impossible, to accurately model that simplification is the only option. To give a few examples we note that time windows are in reality often soft since extensions can be negotiated at a certain cost. However, this is not always possible; hence, modelling time windows as soft is not accurate either. Similarly, the constraints describing ship-port compatibility are affected by both the current load of the ship and the tidal conditions, and these constraints can even in some cases be of a rather subjective nature since a risk-willing captain can agree to sail a ship into a shallow port even if the draft of the ship should not allow this.
- **Simplified problem:** Adding to the above, the basic TSRSP is a simplified version of reality. In most cases, the problem must be extended to incorporate operator specific complexities in order for the solutions to be applicable in real life. Conversely, planners must find ways to modify solutions from the simplified problem in order to take into account the additional real-world aspects and complexities ignored in the simplified problem.

4.2 Current and Future Research Direction for the TSRSP

The first two items, and to some extent also the third, on the above list do not as such suggest further research within tramp ship routing and scheduling, but rather that researchers take on a more consultant like role in order to better sell their ideas to the industry. At the same time it is probably a matter of giving the shipping industry a little more time to get to a point where they are both interested in and able to accept optimisation-based tools. In fact, the shipping industry has started to change in this direction by gradually employing more planners with more academic background and less practical experience. The third item on the list also require changes from within the industry itself. Operators will be able to plan further ahead if they are given more flexibility to cope with the high uncertainty, e.g. send a different ship than originally agreed, arrive later or earlier than planned etc.

The fourth and fifth item on the list focus mainly on the implementation side. They advocate the development of decision support tools that have access to all necessary and continuously updated data, and which allow much interaction between the user and the planning tool to facilitate the construction of several good solutions as opposed to just one solution that is, at least on paper, optimal. The development of decision support tools is certainly interesting and relevant work. However, actual implementation is not a very generic process as it will no doubt rely on the IT systems currently used by the considered ship operator as well as other corporate characteristics. In this paper we will therefore not go further into details with the implementation side of the TSRSP though we both acknowledge and advocate for relevant work on implementation.

The last item on the above list advocates further research in the direction of development of new mathematical models and solution methods for tramp ship routing and scheduling. However, this research should focus on extensions of the basic problem described in Section 3 and we note that solution methods and technology are now advanced enough to facilitate the incorporation of additional real-world complexities. The ability to solve such extended TSRSPs will hopefully also facilitate more implementations since the corresponding extended models will fit the reality of a broader range of tramp operators. Furthermore, even operators previously able to benefit from solution methods for the basic TSRSP can benefit from these extensions as they can create new opportunities for profit maximisation as well as increased customer satisfaction. To give an example, we note that the basic TSRSP assumes that each ship sails at fixed speed while determination of the actual speed along each voyage leg is left for operational planning. Recent years have, however, shown quite a bit of research where the variable speed aspect is incorporated into the basic TSRSP and, not surprisingly, results show that operators can increase profits from such an approach, see e.g. Norstad et al. (2011). Below is a quote from Fagerholt and Ronen (2013) supporting the general idea of further research within extensions of the basic TSRSP:

” This demonstrates that it is much more important to model and solve the right problem, considering the opportunities that often arise in practical problems, than to strive for optimal solutions to simplified versions of the problem.”

In general, the development of research within tramp ship routing and scheduling over recent years is also consistent with the notion of extending the basic TSRSP to include additional complexities. In Section 4.3 we aim at scoping the basic TSRSP and present examples from literature covering a broad range of extensions of this basic TSRSP.

4.3 Scoping the Basic TSRSP and its Extensions

Naturally, there will be quite different perceptions of what constitutes the ‘basic TSRSP’ and what qualifies as ‘additional complexities’. Here, we let the existing literature on tramp ship routing and scheduling guide our definition of the basic problem. We do this by first noting that, although the multiple cargo case is much more complex than the full shipload case, plenty of literature deal with multiple cargoes. We already mentioned several examples in Section 2, and adding to this we note that, among many others, Norstad et al. (2011) also manage to incorporate further complexities in the multiple cargo problem as they in this paper allow variable speed. Therefore, although the multiple cargo case does add complexity compared to the full shipload case, in our definition of the ‘basic TSRSP’ we allow for both full shiploads and multiple cargoes. However, as

we noted in Section 2, flexible cargo sizes are neglected in most work on the TSRSP. This stands in contrast to the fact that flexible cargo sizes are relevant for most tramp operators transporting bulk products, and in particular liquid products. Therefore, we consider this problem characteristic as an ‘additional complexity’. Similarly, we noted in Section 2 that the multiple (non-mixable) product case is rarely considered in the existing literature on the TSRSP; hence, we do not consider this problem characteristic as part of the basic TSRSP. Finally, we noted in Section 2 that the inclusion of spot vessels does not as such complicate the solution procedure, and that this problem characteristic is included in a wide variety of papers on the TSRSP. Accordingly, in our definition of the basic TSRSP we allow the use of spot vessels, though they are certainly not required to be used.

Any complicating problem characteristic not already mentioned or implicitly excluded by the mathematical formulations of the problem in Section 3.2 (e.g. variable speed), can, in our opinion, be considered an ‘additional complexity’. Thereby, the mathematical formulations presented in Sections 3.2.1, 3.2.2 and 3.2.3 actually cover our definition of the basic TSRSP. Below we list some examples of such additional complexities as well as literature on these examples.

We have already mentioned both flexible cargo sizes and multiple (non-mixable) products as additional complexities and provided examples of work on these topics in Section 3. Again, we note that when routing and scheduling fleets that sail full shiploads, the flexible cargo size assumption can be implicitly included during preprocessing.

As mentioned, an extension of the basic TSRSP, which has received a lot of attention in recent years, is the incorporation of variable speed. This recent interest in speed optimisation has been motivated by both steep bunker fuel price increases as well as an increased focus on the environmental impact of maritime transportation (and transportation in general). Examples of routing and scheduling combined with speed optimisation can be found in Norstad et al. (2011), Gatica and Miranda (2011) and Magirou et al. (2015).

Another topic that has recently received increased attention due to the increase in fuel prices, is the aspect of bunkering, i.e. refueling. The main attention on this topic has been directed towards liner shipping, see e.g. Plum et al. (2015) who present an overview of formulations, solution methods as well as results on this topic. In the majority of work on bunkering, the problem is solved for a fixed route, and this assumption is certainly obvious within liner shipping. However, since tramp ships do not sail according to fixed route networks, for tramp operators it seems natural to incorporate bunker planning in the routing and scheduling phase and this is done in Vilhelmsen et al. (2014b) and Meng et al. (2015).

Recent years have also shown an increase in literature involving several parts of the supply chain. The majority of this work is related to industrial shipping, where the supply chain aspect fits naturally, since the industrial ship operator is also the cargo owner; hence, he/she is also responsible for any inventory management at the ends of the maritime transportation legs. Such situations leads to a specific class of problems called *maritime inventory routing problems*. We refer the reader to Christiansen et al. (2013) for a thorough introduction as well as literature review on such problems and to Andersson et al. (2010) for a comprehensive literature review as well as a description of the industrial aspects of combined inventory management and routing in both maritime and road-based transportation. The inventory aspect is less obvious within tramp shipping. Even so, we note that Stålhane et al. (2014) and Hemmati et al. (2015), as already mentioned, combine tramp shipping with a vendor managed inventory service. In this work the tramp operator has both mandatory cargoes that must be transported, optional cargoes that can be transported, as well as inventory pairs that may be serviced a number of times to keep inventories within their limits. Also mentioned previously, Norstad et al. (2015) add voyage separation requirements to the routing and scheduling problem in order to reduce inventory costs for the charterer, and Vilhelmsen and Lusby (2014) consider basically the same problem.

Although we can find other examples of additional complexities added to the basic TSRSP, we end this section by one last example, namely the split load case, where each cargo can be split among several ships. This extension is considered in Korsvik et al. (2011), Stålhane et al. (2012) and Andersson et al. (2011a).

5 Solution Methods for the TSRSP

Most work on the TSRSP and its extensions contains a problem specific arc flow formulation or refer to similar work, that contains such a formulation. Most of these formulations can, just as the ones presented in Section 3.2, easily be linearised, and, in theory, solved by use of standard commercial optimisation software for mixed integer linear programming. An example of this approach can be found in Norstad et al. (2015). However, as already mentioned in Section 3.2, and also concluded in Norstad et al. (2015), for most real life sized instances, this approach will be too time consuming. In this section we therefore discuss some methods and tools commonly used to solve the TSRSP and extensions of it. We also relate these methods and tools to literature within the area.

As can be seen from Section 2, the methods used to solve the TSRSP and its extensions are quite diverse, ranging from simple heuristics to exact methods. Naturally, the choice of method must depend on the complexity of the problem at hand as well as requirements for solution quality and computation time. Even though literature contains a broad range of both problems and methods, one method seems to receive much more attention than any of the others: Column generation. We therefore dedicate Section 5.1 to a generic description of the exact column generation approach for the TSRSP. Section 5.2, on the other hand, focuses on heuristic solution approaches for the TSRSP.

5.1 Column Generation

The popularity of column generation is in line with research within many other fields. We note that both crew scheduling and vehicle routing problems are very often solved using this approach, and that the method has achieved great success within these areas and many others, see e.g. Butchers et al. (2001) and Toth and Vigo (2002). Furthermore, Desaulniers et al. (2005) devoted two whole chapters to column generation in maritime problems. Looking through the literature, it seems that this approach has become more popular within recent years where focus has been on TSRSP extensions. This is on one hand a little surprising since the added complexity contained in these extensions should indicate the use of heuristic methods to achieve reasonable running times. On the other hand, the increase in software and hardware has allowed more focus on exact methods, and one of the main advantages of the column generation approach is that it allows complicating constraints to be handled in ship specific subproblems which generate the columns. Furthermore, it should be noted that even though this is an exact method, it can easily be modified in a heuristic direction as we discuss in Section 5.2.1. No matter the heuristic or exact nature of the method, the frequent use of the column generation approach for a broad range of complex instances, not just within shipping but also within many other research areas, demonstrates the great advantage and flexibility of this method that allows easy incorporation of complicating constraints.

5.1.1 Master Problem

Since most of the TSRSP constraints relate to a specific ship, it seems natural to decompose the problem and use column generation. Whether we use a priori column generation or dynamic/delayed column generation, the master problem of such a decomposition is the same. It is a path formulation containing only the constraints which couple the ships together and ensure that each ship follows exactly one path. In the basic TSRSP, the coupling constraints correspond to constraints (2) and (3), or equivalently (12) and (13), though these must be expressed by new path flow variables while a similar reformulation of the original objective function (1) or (11) must be performed.

To arrive at this new path formulation for the basic TSRSP as well as many of its extensions, we let \mathcal{R}^v denote the set of all feasible schedules for ship v . We denote the profit of a schedule by p_r^v for $r \in \mathcal{R}^v$ and define a binary schedule variable λ_r^v that is equal to 1 if ship v sails schedule r , and 0 otherwise. The profit p_r^v is calculated based on information from the underlying schedule, which holds all necessary information, i.e. the ship it is constructed for, the cargoes carried, and the timing of port calls during the schedule. Finally, we let the parameter a_{ir}^v be equal to 1 if ship v carries cargo i in schedule r , and 0 otherwise. As earlier, \mathcal{N} denotes the set of cargoes to be transported and can be partitioned into \mathcal{N}_C and \mathcal{N}_O containing, respectively, the set of contract cargoes and the set of optional cargoes.

The master problem is now given by the following path flow reformulation of the original arc flow model:

$$\max \sum_{v \in \mathcal{V}} \sum_{r \in \mathcal{R}^v} p_r^v \lambda_r^v \quad (33)$$

s.t.

$$\sum_{v \in \mathcal{V}} \sum_{r \in \mathcal{R}^v} a_{ir}^v \lambda_r^v = 1, \quad \forall i \in \mathcal{N}_C, \quad (34)$$

$$\sum_{v \in \mathcal{V}} \sum_{r \in \mathcal{R}^v} a_{ir}^v \lambda_r^v \leq 1, \quad \forall i \in \mathcal{N}_O, \quad (35)$$

$$\sum_{r \in \mathcal{R}^v} \lambda_r^v = 1, \quad \forall v \in \mathcal{V}, \quad (36)$$

$$\lambda_r^v \in \{0, 1\}, \quad \forall v \in \mathcal{V}, r \in \mathcal{R}^v. \quad (37)$$

The objective function (33) maximises profit from chosen schedules. Constraints (34) and (35) are the path flow reformulations of constraints (2) and (3), respectively, or equivalently (12) and (13). The convexity constraints (36) and binary restrictions on the schedule variables in (37) ensure that each ship is assigned exactly one schedule. Note that if the problem is to be solved by Branch-and-Bound, it is helpful to insert explicit slack variables in constraints (35). If such slack variables are added to constraints (35), we recognise the master problem as a Set Partitioning Problem in which the columns correspond to feasible ship schedules.

For the TSRSP, and most of its extensions, there is no need to include all feasible schedules for each ship in the master problem. Instead we can limit the column set to contain the profit maximising schedule for each combination of ship and cargo set. There can, however, be situations where additional coupling constraints in the master problem means that the profit maximising schedule for one ship and cargo set is not necessarily profit maximising when taking the entire fleet into account. This is the case in Vilhelmsen and Lusby (2014) where they have temporal dependencies between the schedules for different ships. These dependencies require the enumeration of all feasible schedules rather than just all profit maximising schedules.

Accordingly, two schedules in the master problem might correspond to the same route but differ in other aspects, e.g. the timing of port calls. For some applications, non-integer solutions to the master problem can be considered feasible as long as the (fractionally) selected schedules for a specific ship all correspond to the same geographical route. Constraints (36) ensure that the schedule for each ship is a convex combination of different schedules for the same ship while constraints (37) must be rewritten to ensure that the schedules included in the convex combination correspond to the same geographical route. This means that the binary restrictions in (37) can be replaced by restrictions of the form

$$\{r : \lambda_r^v > 0\} \quad \text{correspond to equal geographical routes,} \quad \forall v \in \mathcal{V}. \quad (38)$$

If convex schedule combinations are used, a greater part of the solution space can be spanned without generating as many columns as would be required if not utilising such convex combinations. This approach is used in e.g. Christiansen and Nygreen (1998) for a ship routing problem with inventory constraints and Vilhelmsen and Lusby (2014) also allow such convex schedule combinations. However, there can be many situations where convex schedule combinations do not qualify as actual schedules, primarily due to cost calculations. One example is when the TSRSP is extended to allow variable speed as in Norstad et al. (2011). Since the fuel consumption function is a convex function of speed, once we include variable speed we can no longer allow convex schedule combinations, since the convex combination of schedule costs would overestimate the actual cost of the schedule. Therefore, we must select exactly one schedule for each ship. Another example is in the work presented in Vilhelmsen et al. (2014b) where the TSRSP is extended to include bunker planning. In this case the schedules also contain information on bunker stops, and each of these stops corresponds to a predefined price for a specific bunker option at a given time. Assume now that two schedules for the same ship each contain a stop at a bunker option, B , though at different times denoted t_1^B and t_2^B with different bunker prices, P_1^B and P_2^B , respectively. The convex combination of these two schedules will have an arrival time somewhere in the interval (t_1^B, t_2^B) ,

assuming that $t_1^B \leq t_2^B$. Due to price fluctuations, there is no guarantee that the bunker price in this interval corresponds to the convex combination of the bunker prices P_1^B and P_2^B . Therefore, in this situation as well, we must select one single schedule for each ship and so constraints (38) cannot be used.

Before going into details with the actual column generation procedure, we want to briefly discuss the underlying structure of the master problem and the implied integer properties from this structure. To ease notation, we let V denote the number of vessels, $|\mathcal{V}|$, and by A^v we denote the $|\mathcal{N}| \times |\mathcal{R}^v|$ submatrix containing the schedules for ship $v \in \mathcal{V}$, i.e. the a_{ir}^v entries. If we insert slack variables, $s_1, \dots, s_{|\mathcal{N}_O|}$, in constraints (35), then the constraint matrix looks as in Figure 1, where zero entries are ignored. The convexity constraints (36) are generalised upper

	$\lambda_1^1 \quad \dots \quad \lambda_{ \mathcal{R}^1 }^1 \quad \dots \quad \lambda_1^V \quad \dots \quad \lambda_{ \mathcal{R}^V }^V \quad s_1 \quad \dots \quad s_{ \mathcal{N}_O }$	
1		}
⋮	$A^1 \quad \dots \quad A^V$	
$ \mathcal{N} $	1 ⋮ 1	
1		
⋮	⋮	
V	1 ⋮ 1	

Figure 1: The master problem constraint matrix

bound constraints, and because of these, the submatrix for each ship is perfect. This means that fractional solutions can never occur solely within one of the individual ship submatrices. Fractional solutions can, however, appear across submatrices for different ships. Thereby, the LP solution can only be fractional if two or more ships are competing for the same cargo. We refer the reader to Padberg (1973) and Conforti et al. (2001) for a discussion on perfect matrices and their properties.

Note that for problems with this integer property, once we have eliminated fractionalities across submatrices for different ships, the solution will never contain more than one schedule for each ship; accordingly, there is no point in checking for “equal geographical routes” as in constraints (38), unless further complexities are added to the problem.

5.1.2 A Priori Column Generation

Due to the uncertainty involved in maritime transportation as well as the long voyages, many operators are prohibited from planning more than a few voyages ahead for each ship. This is especially true for the full shipload case, and for such operators, depending on the constraints for ship-cargo compatibility, the number of feasible combinations of ships to cargo sets can be small enough to allow a priori full enumeration of all these combinations. For problems where we can limit the column set to contain just the profit maximising schedule for each combination of ship and cargo set, this means that all the columns in the master problem can be a priori generated. Examples of tramp related work using the a priori column generation approach can be found in Kim and Lee (1997), Brønmo et al. (2007b), and Andersson et al. (2011b).

For each feasible cargo set for each ship, we therefore need to determine the profit maximising schedule, i.e the order and timing of port visits, that yields the best profit for this cargo set. Depending on the TSRSP considered, it can be necessary to simultaneously determine other schedule details, e.g. load quantities in case of flexible cargo sizes. Naturally, only feasible schedules can be considered; hence, the timing of port visits must adhere to cargo time windows, ship capacity must be respected and so on. Generally speaking, all constraints from the original arc flow formulation that corresponds to the considered ship, must be adhered to in this column generation process.

The column generation in the a priori approach therefore consists of repeatedly determining the optimal route and schedule for a given ship and a given cargo set. This can be done e.g. by simply enumerating all feasible routes for the given cargo set or by dynamic programming. Some instances of the TSRSP and its extensions will, however, be too large to allow full enumeration of all combinations of ships to cargo sets. In such cases it is possible to instead use dynamic column generation as described below. The enumeration approach is used in e.g. Brønmo et al. (2007b) while the dynamic programming approach, on the other hand, is used in e.g. Fagerholt and Christiansen (2000b).

5.1.3 Dynamic Column Generation

As already mentioned, some instances of the TSRSP and its extensions will be too large to allow full enumeration of all combinations of ships to cargo sets, and some situations require the enumeration of all feasible schedules rather than just all profit maximising schedules. Therefore, for several reasons, there can be situations where the a priori column generation approach is prohibited. In such situations, dynamic column generation can be used, and this is the approach used in e.g. Stålhane et al. (2012), Vilhelmsen et al. (2014b) and Cóccola et al. (2015). We refer the reader to Desaulniers et al. (2005) for a detailed description of this method.

We also note that both Appलगren (1969) and Brønmo et al. (2010) have found dynamic column generation more efficient than a priori generation for situations where a priori generation was actually possible. They find the dynamic approach both faster and more flexible as it allows them to deal with larger and more loosely constrained instances than the a priori approach.

In the dynamic column generation approach, the master problem initially only contains a small subset, \mathcal{R}'^v , of the feasible schedules for each ship $v \in \mathcal{V}$, and new columns that have the potential to improve the current master problem solution are then generated iteratively. The new columns are generated in subproblems, or pricing problems, derived from a Dantzig-Wolfe decomposition, and there are as many independent subproblems as there are ships. Each subproblem corresponds to a specific ship and contains all constraints related to this ship from the original formulation. In each iteration we solve the *restricted master problem* (RMP), which is the linear relaxation of the original master problem though with only its current restricted column set. The dual variables from this solution are then used to guide the generation of promising columns in the subproblems, which are added to the master problem, whereafter a new iteration begins. This process continues until no further promising columns can be generated. At that point, the current solution to the RMP is optimal for the linear relaxation of the original master problem with all feasible schedules in the column set.

The purpose of the subproblems is to generate ship specific schedules that can potentially improve the current solution to the RMP. This in turn corresponds to finding feasible schedules with positive reduced costs in the current RMP solution. Since only feasible schedules should be generated, the subproblem must contain all relevant constraints for the corresponding ship, e.g. time windows and capacity constraints.

To illustrate the structure of these subproblems, we let σ_i be the dual variables for constraints (34) and (35). The variables corresponding to (34) are free of sign while the variables corresponding to (35) must be nonnegative. Furthermore, we let ω_v be the dual for constraint (36) which is also free of sign. Finally, we define $\pi_i = \sigma_i$ for all $i \in \mathcal{N}$ and $\pi_{o(v)} = \omega_v$ corresponding to the origin node. If we first restrict our attention to the simple full shipload TSRSP formulation presented in Section 3.2.1, the corresponding subproblem for ship v is given by:

$$\max \sum_{(i,j) \in \mathcal{A}^v} (P_{ij}^v - \pi_i) x_{ij}^v \quad (39)$$

s.t.

$$(4) - (9). \quad (40)$$

The objective function (39) maximises the reduced cost with respect to the current dual variables, and the constraints (4)-(9) ensure that only feasible schedules are considered. If the optimal schedule has a positive reduced cost, it has the potential to improve the current RMP; hence, it will be represented by a new column in the RMP.

For the multiple cargo formulation from Section 3.2.2, the subproblem for ship v is given by:

$$\max \sum_{i \in \mathcal{N}_p^v} R_i x_{ij}^v - \sum_{(i,j) \in \mathcal{A}^v} (C_{ij}^v + \pi_i) x_{ij}^v \quad (41)$$

s.t.

$$(14) - (26). \quad (42)$$

The objective function (41) again maximises the reduced cost with respect to the current dual variables, and the constraints (14)-(26) again ensure that only feasible schedules are considered.

Each subproblem can be modelled as an elementary shortest path problem with time windows (ESPPTW). The ESPPTW has been proven \mathcal{NP} -hard in the strong sense by Dror (1994); hence, no pseudo-polynomial algorithms are likely to exist for this problem. When more complex TSRSPs are considered, the subproblems are most often more general elementary shortest path problems with resource constraints (ESPPRC) which are also \mathcal{NP} -hard in the strong sense since they generalise the ESPPTW. This is for instance the situation for the multiple cargo case, where, in addition to time, ship capacity is also a constraining resource. The ESPPRC, and thereby also the ESPPTW, entails finding an elementary shortest path between two nodes while satisfying the resource constraints, e.g. time windows and capacity constraints.

In maritime transportation, and in particular within deep sea shipping, voyages travel times are often so long that few, if any, time feasible cycles can be expected in the subproblem networks. Therefore, it is quite common to relax the subproblem to allow non elementary paths, since the regular shortest path problem with resource constraints (SPPRC) can be solved in pseudo polynomial time (see e.g. Desrochers and Soumis (1988); Irnich and Desaulniers (2005)). This relaxation is used in e.g. Brønmo et al. (2010) and Vilhelmsen et al. (2014b). Note though that if time-feasible cycles do in fact exist, schedules can be generated where the same cargo is transported more than once. In such cases, the solution method must be modified to handle these cycles.

The SPPRC is typically solved by dynamic programming algorithms on the underlying ship specific networks, and this is also the approach used in both Brønmo et al. (2010) and Vilhelmsen et al. (2014b). We refer the reader to Desaulniers et al. (1998), Irnich and Desaulniers (2005) and Irnich (2008) for a thorough introduction to the SPPRC, the related dynamic programming algorithms, and several associated concepts.

5.1.4 The Column Generation Scheme

When a priori column generation has been applied, the number of columns in the master problem is often sufficiently small to allow the master problem to be solved directly by commercial optimisation software for mixed integer programs and thereby obtain an optimal integral solution. This approach is used in e.g. Fagerholt and Christiansen (2000b) and Fagerholt (2001). When the column count becomes too large for this approach, we must instead solve the linear relaxation of the master problem, and then apply a Branch-and-Bound procedure if the optimal solution is fractional.

When columns have been generated dynamically, once the column generation phase is completed, we have an optimal solution to the linear relaxation of the original master problem in (33)-(37). If this solution is fractional, the column generation procedure must also be embedded in a Branch-and-Bound framework where new columns are generated in each node of the search tree. In this case, the entire procedure is called a Branch-and-Price procedure, since new columns are priced out in each node of the tree.

The strong integer property of the constraint matrix means that the linear relaxation of the master problem will provide very tight upper bounds. This property is verified in Vilhelmsen et al. (2014b), where they incorporate bunker planning in the TSRSP. There, they experience relatively few fractional solutions, and for those fractional solutions they do encounter, the IP gap is very small. Note however that when extending the basic TSRSP, the additional complexities added to the basic model can easily compromise the strong integer property of the constraint matrix. Examples of this can be found in the work on split loads by Stålhane et al. (2012), on vendor managed inventory by Stålhane et al. (2014) and on voyage separation requirements by Vilhelmsen and Lusby (2014). In each of these cases, the considered extension creates an interdependence between schedules for different ships such that these new constraints must remain in the master problem rather than being added to the subproblems.

5.1.5 Branching

When in fact branching is required, the branching scheme is very often tailor made to the specific application considered. However, one generic method that seems to be frequently used for tramp shipping, as well as other scheduling problem such as e.g. crew scheduling and vehicle routing problems, is that of constraint branching, and we discuss this approach below. Since constraint branching is not a complete branching strategy, this approach must often be complemented by other branching schemes. Another generic branching approach used within tramp shipping as well as other areas, is that of time window branching which we also discuss below.

Constraint branching was introduced by Ryan and Foster (1981), and is an efficient branching scheme developed for solving set partitioning and set packing problems. Within tramp shipping, this branching approach exploits the underlying structure of the constraint matrix by noting that within any optimal fractional solution, there must be a cargo $i \in \mathcal{N}$ that is transported (fractionally) by several ships. Since, by definition, each spot vessel (slack variables can also be viewed as a form of spot vessel schedules) can only transport one cargo, the spot vessels cannot compete with each other for cargoes. Therefore, at least one of the ships currently competing for cargo i must be a non-spot vessel and we denote this ship k . For each ship $v \in \mathcal{V}$ and cargo $i \in \mathcal{N}$ we introduce the sum

$$S_i^v = \sum_{r \in \mathcal{R}'^v} a_{ir}^v \lambda_r^v.$$

If the current solution is fractional, there must exist a cargo i and a regular ship $k \in \mathcal{V}$ for which $0 < S_i^k < 1$. In an integral solution, the cargo can only be transported by one ship and so ship k can either transport or *not* transport cargo i . The branching strategy is therefore to construct a 1-branch where ship k is forced to transport cargo i and a 0-branch where ship k is not allowed to transport cargo i . In the 1-branch this means that $a_{ir}^k = 1$ should hold for all $r \in \mathcal{R}'^k$. Thereby, we can remove all schedules for ship k that do not include cargo i and all schedules for other ships that do include cargo i , either explicitly or implicitly by simply setting their upper bounds equal to zero. This also means removing any spot vessel schedule for cargo i . Furthermore, we remove cargo i in all subproblem networks not corresponding to ship k . Without further modifications it is possible for the subproblem of ship k to construct new schedules that do not include cargo i . Different ways of overcoming this problem exist, the simplest of which simply relies on the dual variables to eventually enforce the construction of schedules for ship k that include cargo i . In the 0-branch, we instead have $a_{ir}^k = 0$ for all $r \in \mathcal{R}'^k$. This is the opposite process of the 1-branch, and so we instead remove all schedules for k that *do* include cargo i . Furthermore, we remove cargo i in the subproblem network corresponding to ship k while we cannot make any changes in the networks corresponding to other ships.

To ease notation, we let S_i^S denote the similar sum for spot vessels (including slack variables). When $0 < S_i^S < 1$, we can then use the same branching approach as just described for regular ships though the corresponding updates of the RMP and subproblems must of course be modified accordingly.

Competition for a cargo i requires at least two different ships. Thereby, in a fractional solution there must be at least two distinct ships (one of them possibly a spot vessel), k_1 and k_2 , for which $S_i^{k_1}$ and $S_i^{k_2}$ are fractional. Therefore, in any fractional solution we have at least two candidate cargo-ship pairs for branching and we must select one of these. The approach for selecting the branching pair varies in the literature as basically two different opinions on this selection exist:

Most integer: Select the vessel cargo pair with largest S_i^k . In other words, from the set of all candidate pairs \mathcal{C} , where

$$\mathcal{C} = \{(i, k) \in \mathcal{N}_C \cup \mathcal{N}_O \times \mathcal{V} \cup \{S\} : 0 < S_i^k < 1\},$$

we select

$$(i, k)^* = \arg \max_{(i, k) \in \mathcal{C}} \left\{ S_i^k \right\}.$$

The premise is that a value close to 1 is evidence to suggest that ship k most likely transports cargo i in an optimal integer solution. If the largest S_i^k is, however, much smaller than 1, it seems more intuitive to enforce a 0-branch by picking pair (i', k') with the smallest S_i^k . A small value of S_i^k indicates that the ship is unlikely to transport cargo i . One typically sets a threshold value for the largest S_i^k , e.g. 0.8, prior to solving the problem. If the largest value is at least the threshold, a 1-branch is enforced, while a 0-branch is enforced otherwise.

Most fractional: If S_i^k is close to 1, the solution favours ship k for transporting cargo i ; hence, forcing ship k to transport the cargo, as we do in the 1-branch, might not change the solution much and thereby not the upper bound either. In the 0-branch, where we force ship k not to transport the cargo, we can on the other hand expect to see a greater impact on the upper bound. This could potentially create an unbalanced search tree. The situation is similar, though reversed, if S_i^k is close to 0. To maintain a balanced search tree we might therefore prefer branching on candidates that are as fractional as possible, i.e. as close to $1/2$. In such cases we would hence select the following candidate pair to branch on:

$$(i, k)^* = \arg \min_{(i,k) \in \mathcal{C}} \left\{ \left| S_i^k - \frac{1}{2} \right| \right\}.$$

Examples of work on the TSRSP using constraint branching can be found in Stålhane et al. (2012) and Vilhelmsen and Lusby (2014).

Time window branching was originally presented by Gélinas et al. (1995) as a branching scheme to restore integrality. As mentioned in Section 5.1.3, for some applications and solution methods it is possible to generate cyclic schedules, i.e. schedules where the same cargo is transported more than once. The presence of such schedules in the master problem will compromise the strong integer property of the constraint matrix and therefore also cause trouble for the constraint branching scheme described above. In such situations time window branching complements constraint branching very nicely, as it can be used to eliminate cycles, though the approach can be useful for much more than just that.

The basic concept in time window branching is to split a given time window into two smaller time windows that each correspond to a new problem, i.e. to a new branch in the branch-and-bound search tree. The key to success with this branching scheme is to select a good time window to split on, and furthermore to select a good split time for this specific time window. This split time must be chosen in such a way that the current solution becomes infeasible in each of the two new problems, i.e. in a way that makes at least one currently chosen schedule infeasible in each new branch.

We illustrate this process in Figure 2 for the time window $[a_i, b_i]$ corresponding to a cargo i which is currently transported by two schedules, or twice in a cycle by the same schedule. The start times of these two visits are denoted T_i^1 and T_i^2 , respectively. The grey boxes in Figure 2 correspond to the so called *feasibility interval* of each of these two visits. These intervals, $[T_i^1, u_1]$ and $[T_i^2, u_2]$, contain all the start times that will allow the corresponding schedule to still remain feasible. This will most often correspond to redistributing waiting time; hence, we have depicted these feasibility intervals as extensions of the current visit times. Since these two intervals are disjoint, we can choose a split time within $(u_1, T_i^2]$, say t_s , and create one branch where the time window for cargo i is $[a_i, t_s - \epsilon]$ and the second visit is infeasible, and one branch where the time window is $[t_s, b_i]$ where the first visit is infeasible. Here, $\epsilon > 0$ is a very small tolerance.

Note that split times within one of the feasibility intervals would also render one visit infeasible in each branch. However, during the next iteration of schedule generation, the visit, for which we selected a split time within its feasibility interval, could be regenerated though only a little later in time. After branching, all previously generated schedules violating these new time windows, are removed from the master problem in each new branch and the corresponding subproblems are updated to reflect the new time windows.

How to choose the specific time window to branch on and also what exact split time to use within this time window, can depend on the considered application though a generic and detailed approach can be found in Gélinas et al. (1995). They propose three strategies for choosing which time window to branch on; namely focusing on elimination of important cycles, focusing on the

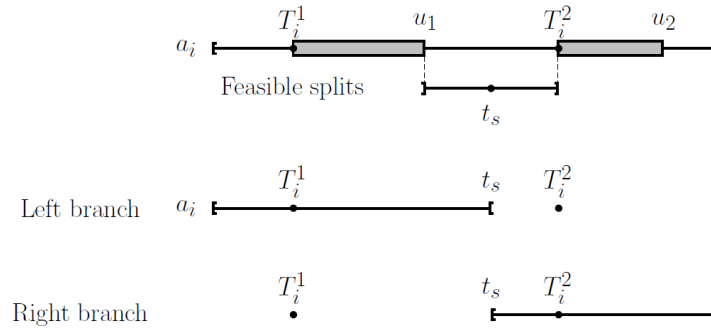


Figure 2: Time window branching due to fractionality

total number of visits made to each node in the underlying ship-cargo network, and finally, focusing on the flow values. Computational results show that focusing on the flow values is the most efficient strategy though slight improvements can be made by combining all three strategies. Examples of work on the TSRSP using time window branching can be found in Brønmo et al. (2010) and Vilhelmsen and Lusby (2014).

5.2 Heuristic Solution Approaches

The performance of exact approaches typically deteriorates as the problem size grows. Formulations can very quickly become intractable when considering large problem instances. This usually means that only small instances, or somewhat simplified versions of reality, are solved to optimality with exact frameworks in reasonable time. To combat problems that give rise to intractable MILP formulations as well as those that result in excessive running times for exact procedures, heuristic approaches are often employed. Heuristic methods are usually based on rules of thumb and attempt to construct “good” solutions. They do not necessarily provide the optimal solution nor do they even provide a quality measure on the distance from optimality, unlike pure mathematical programming techniques. In many cases they are, however, preferred as they can quickly provide good solutions, and can be more flexible when incorporating additional real-world complexities. This naturally leads back to the discussion in Section 4.2 on whether to solve simplified problems to optimality or allow the inclusion of additional real-world complexities, possibly at the expense of optimality. The fact that heuristics are often able to quickly find good solutions typically outweighs the fact that they do sacrifice optimality. In this section we review heuristic approaches within the field of TSRSPs. The discussion is broken into two distinct parts, and each is considered in turn. The first is dedicated to heuristic modifications of column generation based approaches, while the second considers a variety of more general metaheuristic techniques that have been developed. Metaheuristics are problem independent algorithms that guide a subordinate heuristic by intelligently searching through the solution space (see e.g. Osman and Laporte (1996)).

5.2.1 Heuristic Column Generation Procedures

It is somewhat natural to consider heuristic modifications of the described column generation procedures for TSRSPs. Column generation is, after all, the best performing exact method. The purpose when heuristically modifying a column generation procedure is to speed up the algorithm without sacrificing too much optimality.

Examples of tramp related work that utilise heuristic modifications of the a priori column generation can be found in Fagerholt (2001), where the possible times for start of service are discretised, and in Fagerholt and Christiansen (2000a), where the number of a priori generated schedules is reduced by introducing heuristic rules regarding the capacity utilisation of the ships. Stålhane et al. (2014) also present heuristic modifications of a priori column generation by imposing limits on the inventory deliveries and the amount of spot cargoes a ship can transport. We note, though, that this is actually a hybrid approach that also utilises dynamic column generation.

The dynamic column generation approach can also be heuristically modified and in the literature we find several examples of this. In Brønmo et al. (2010), where flexible cargo sizes are allowed, the authors use a two-stage approach for the subproblems. In this approach the otherwise continuous cargo quantity variables are discretised, since this allows them to follow a more standard dynamic programming algorithm for solving the subproblem. Afterwards, for the p best solutions to the discretised subproblems, the authors find the optimal continuous load quantities for each of these p routes by solving an LP. In Vilhelmsen et al. (2014b), where the integration of bunker planning in the TSRSP is considered, a similar approach by discretising the otherwise continuous bunker purchase variables is followed. Stålhane et al. (2012) also use a heuristic modification of dynamic column generation as they only consider a subset of the actual arcs in their subproblem networks and limit the length of the paths generated.

5.2.2 Metaheuristics

The field of metaheuristics encompasses a wide range of algorithms, which can be used to intelligently search the solution space without the tractability issues of large mathematical programming formulations. Examples of well known metaheuristics include, but are not limited to, local search procedures, tabu search, and genetic algorithms. They are iterative generation processes that intelligently guide the search to promising parts of the solution space. Due to its complexity, it is not surprising that several examples of metaheuristics can be found in the TSRSP literature.

Local search procedures consider the immediate neighbours of a potential solution in the hope of locating and moving to an improved solution. Generally speaking, a neighbour of a given solution is almost identical to the solution, differing in only one or two components. Starting from an initial solution a local search heuristic iteratively considers a neighbouring solution of the current solution and accepts the neighbouring solution if it yields a better objective function. Iterations are performed as long as a terminating criterion (e.g. time limit or iteration count) is not satisfied. An application of local search within tramp ship routing and scheduling can be found in Brønmo et al. (2007a) in which the authors propose a multi-start local search heuristic for the multiple cargo TSRSP. Rather than starting with just one initial solution, a multi-start local search procedure provides several initial solutions, each of which is considered for improvement, in the hope of diversifying the search. In Brønmo et al. (2007a) a set of initial solutions is obtained using a procedure with random and deterministic elements to assign cargoes to ships. The best solutions of these are improved using a so-called quick version of their local search, which considers three neighbourhoods: an intra ship schedule operator, and two inter schedule operators. With the intra schedule operator, a cargo is selected and removed from the schedule it is in. The cargo is then reinserted into the same schedule in the best possible location. The first of the two inter schedule operators is the same, with the exception that the removed cargo is moved to its best location across all other ship schedules. The second inter schedule operator is of a similar flavour whereby two cargoes are selected and inserted into each of the other's schedules. Each neighbourhood is used with a certain frequency. An attempt is then made to further improve the best solutions from the quick version by running an extended version of the local search. Five neighbourhoods are used in this stage. Three of these are identical to those of the quick version, while the remaining two are extensions of two of these. The first is an intra schedule operator that re-sequences two cargoes, while the second is an inter schedule operator where three cargoes i , j , and k , are selected and removed from ships u , v , and w . The cargoes are then reinserted into ships v , w , and u . The performance of the metaheuristic is compared with that of a priori column generation approach. Computational results show that the multi-start local search procedure returns optimal or near-optimal solutions within reasonable time for real-life instances. Norstad et al. (2011) extend the framework of Brønmo et al. (2007a) to consider variable speed. The algorithm is structurally the same; however, an additional step which optimizes the speed of the ship is conducted whenever a new ship schedule enters the solution.

Tabu search is a well known metaheuristic in Operations Research literature and was first proposed in Glover (1986). This is essentially an improved local search procedure, differing in two main aspects. First, moves to previously visited solutions for a certain number of iterations are forbidden (i.e. tabooed). However, moves to neighbouring solutions with a worse objective are permitted. Both extensions are an attempt to diversify the search. Accepting worse solutions from

time to time also guards against getting stuck in a local optimum. Examples of tabu search procedures within tramp ship routing and scheduling can be found in Korsvik et al. (2010), Fagerholt et al. (2011), and Korsvik and Fagerholt (2010).

Korsvik et al. (2010) propose a tabu search approach for the multiple cargo problem considered in Brønmo et al. (2007a). The authors generate an initial solution by first sorting unassigned cargoes in order of the start of the time window for the loading port. Cargoes are then sequentially assigned to the ship that results in minimum cost. Furthermore, the loading and unloading ports are inserted in the best positions in the ship's partial schedule. The neighbourhood defined for the improvement phase is quite simple and similar to the first of the two inter schedule operators of Brønmo et al. (2007a); the authors select a cargo, remove it from the schedule it was in and move it to a different ship's schedule. When a cargo is removed from a schedule it is forbidden from being assigned to the ship it was removed from for a certain number of iterations. Inserting a cargo into a schedule is done at minimum cost using simple insertions, i.e. the ordering of the port calls already in the schedule remains unchanged. To improve the running times a neighbourhood reduction technique is employed. Furthermore, moves to infeasible solutions are permitted, and the algorithm runs for a specified number of iterations. Periodic schedule re-optimization is performed at regular intervals to re-optimize the order of the port calls within a ship schedule. The authors compare the results of their method with those of the multi-start local search heuristic described above. Computational results suggest that the tabu search algorithm performs better than the multi-start local search heuristic, especially for large and tightly constrained problems. Fagerholt et al. (2011) consider project shipping with cargo stowage and cargo coupling constraints, a special real-life TSRSP, and propose a modified version of the tabu search algorithm in Korsvik et al. (2010) to solve the problem. The structure of the tabu search procedure is largely unchanged. Again, computational results show that the tabu search heuristic yields optimal or near-optimal solutions within reasonable time.

Korsvik and Fagerholt (2010) present a tabu search algorithm for solving the TSRSP with flexible cargo sizes. The algorithm is very similar in structure to that of Korsvik et al. (2010). However, there are some differences. First, the authors use a randomly generated initial solution. Second, moves to infeasible solutions are not permitted. Instead, the authors introduce a second neighbourhood for the improvement phase. The additional neighbourhood is also similar to one of the neighbourhoods proposed in Brønmo et al. (2007a) and focuses on swapping two cargoes (from different ships) between the ships. Again, the cargoes are inserted into the ship schedules using simple insertion strategies, which leave the order of the previous cargoes unchanged. For a given ship schedule a fast quantity optimization heuristic is then used to determine its optimal cargo quantities. Finally, the algorithm terminates when a set number of iterations have elapsed. Computational results show that the tabu search procedure provides optimal or near-optimal solutions within reasonable time for real-life instances.

A slightly different variant of local search, known as Variable Neighbourhood Search (VNS), has also been proposed by Malliappi et al. (2011) for the multiple cargo TSRSP. VNS was first proposed by Mladenović and Hansen (1997) and is a metaheuristic approach for solving combinatorial optimization problems. As the name suggests, the algorithm utilises local search, but with the addition of a perturbation phase to allow the algorithm to search distant neighbours of the incumbent solution. Within each neighbourhood local search is applied to find a local optimum. Like all other metaheuristic approaches, VNS must be initialized with a feasible solution. For this, Malliappi et al. (2011) consider the cargoes in non-increasing order of the quantity to be carried. The cargoes are sequentially added to ships, being inserted in the best possible position for the first feasible ship. If no feasible assignment exists for a given cargo it is assigned to a dummy ship. Once an initial solution is obtained the two level VNS algorithm is activated. The outer level determines in which neighbourhood to randomly generate a neighbour of the incumbent solution. For this six different neighbourhoods are proposed. Three of which are based on selecting a different number of ships and deleting $\delta\%$ of the assigned cargoes on each. The other three operators re-sequence a cargo within a schedule, reassign a cargo to a different schedule, or swap two cargoes (from different ships) between the ships. The purpose of the outer level is to "shake" the solution to a distant neighbourhood. The so-called shaking neighbourhoods are considered in turn and for each the local search neighbourhoods of Brønmo et al. (2007a) are iteratively considered to find a local optimum. The authors compare their method to both the tabu search heuristic of Korsvik et al.

(2010) and the multi-start local search method of Brønmo et al. (2007a), though with their own implementation of these two methods. Computational results from modified benchmark instances from land transportation show that the VNS, on average, performs best.

Large Neighbourhood Search (LNS) is another well known technique in the metaheuristic field. First proposed by Shaw (1997), this procedure is an extension of the local search strategy in which the neighbourhood can be extremely large. Unlike, the traditional local search procedure, neighbouring solutions need not be structurally similar and can, in fact, be very different. Such large neighbourhoods can assist in diversifying the search. An example of LNS in tramp ship routing and scheduling can be found in Korsvik et al. (2011). The authors consider ship routing allowing split loads. An initial solution is constructed by randomly selecting a cargo one by one from a list of unassigned cargoes and assigning it a random ship until all cargoes have been assigned. A simple insertion strategy, like that of Brønmo et al. (2007a), is used to find the best places within the schedule for the loading and unloading ports. No split loads are considered initially. This initial solution is improved using a local search heuristic that considers four different neighbourhoods. Two of these consider reassigning cargoes; one reassigns a given cargo to another ship, while the other swaps two cargoes between two different ships. The remaining two consider the splitting/merging of cargoes. The improvement phase consists of successive iterations of a so-called destroy-and-repair algorithm. That is, at every iteration the solution is destroyed by removing a randomly selected number of cargoes. The solution is then repaired using several fast insertion heuristics, and subsequently optimised using the four neighbourhood local search heuristic described above. Results indicate that the authors' approach is able to provide good solutions to real-life instances in reasonable time.

The Adaptive Large Neighbourhood Search Procedure of Ropke and Pisinger (2006) extends the LNS to multiple neighbourhoods. Adaptive Large Neighbourhood Search (ALNS) procedures typically have several large neighbourhoods defined and each is assigned a weight reflecting the performance of the neighbourhood. Over time the weights are updated to reflect the relative success of each of the neighbourhoods and essentially ensure the algorithm searches in areas where it has had greatest success. ALNS procedures within tramp ship routing and scheduling can be found in Hemmati et al. (2014) and Hemmati et al. (2015). Hemmati et al. (2014) apply an ALNS to a benchmark suite of tramp shipping instances and report high quality results. The algorithm starts with an initial solution in which no cargoes are assigned to ships. In the subsequent improvement phase, the algorithm chooses between three different destroy (or removal) heuristics followed by a choice of two insertion heuristics. The authors consider a random removal, where q randomly chosen cargoes are removed, the so-called Shaw removal in which q similar cargoes are removed, and a worst removal, in which the q cargoes with highest cost are removed. Regarding the insertion heuristics, a greedy insertion based heuristic as well as a regret based heuristic are used. Initially all destroy and repair methods are equally likely to be chosen; however, over time the selection weights are adjusted to ensure better performing heuristics are chosen. Hemmati et al. (2015) describe an application of this ALNS procedure to solve the vendor managed inventory problem introduced in Stålhane et al. (2014). Results indicate that the metaheuristic approach is, for the most part, competitive with an exact procedure in terms of solution quality, but superior to a heuristic column generation approach.

Finally, genetic algorithms have been used to solve a wide variety of optimisation problems. A genetic algorithm (GA) is an optimisation method for solving a wide range of problems constrained as well as unconstrained. It is based on a natural selection process that mimics biological evolution. A set of individual feasible solutions are denoted a population and this population is then modified in each of the iterations. In each iteration of the genetic algorithm it randomly selects individuals from the current population and uses them as parents to produce the children for the next generation. Applications of GAs within tramp ship routing and scheduling can be found in Lin and Liu (2011), Kang et al. (2012), and Jung et al. (2011). Lin and Liu (2011) propose a GA for the multiple cargo TSRSP. The proposed GA outperforms a mathematical formulation, which is solved directly by a commercial solver. The MILP formulation used, however, is different to that which is used in the above references. Kang et al. (2012), and Jung et al. (2011) focus on a quite specific TSRSP involving the transportation of cars.

6 Concluding Remarks

In this paper we have provided a thorough introduction to the general area of tramp ship routing and scheduling. We have reviewed the most recent literature on tramp ship routing and scheduling and provided the reader with a general understanding of the problem, as well as modeling approaches for it. We have also provided an analysis of both the current status and the future direction of research within the area and provided the reader with a general knowledge on solution methods for tramp ship routing and scheduling problems. Thereby, this paper can be used to provide researchers, new to the field, with a comprehensive understanding of this research topic.

We have argued that, in order for the shipping industry to fully benefit from the research already conducted on the TSRSP, attention should now be given to actual implementation. We have also argued that one of the main directions of current and future research lies in the development of models and methods for extensions of the basic TSRSP rather than in the development of new methods for the basic problem. Such extensions will enable more tramp operators to benefit from the solution methods, while simultaneously creating new opportunities for operators already benefitting from existing methods.

Although some TSRSP extensions, such as variable speed, are quite generic, a lot of extensions will be operator specific. Therefore, it is hard to point in any one direction for future research with respect to extensions. With this being said, we do however want to mention two interesting research topics that, in our opinion, deserve great attention in future research within tramp shipping:

- We find that the static and deterministic approach in most work on tramp shipping constitutes a huge simplification for most operators. In fact, uncertainty plays a big part in maritime optimization where planners face a constantly changing environment with many unforeseen events and large daily variations in demand. Weather and currents have a huge impact on sailing plans but in addition to this it is quite common within tramp shipping that the discharge port is to some extent unknown even at loading time. Furthermore, port conditions such as strikes and lack of available berths can delay loading and unloading, and new cargoes are continuously revealed. Naturally, one must walk before one can run, but now that the basic TSRSP can be solved efficiently, it seems obvious to expand the research to include both dynamic and stochastic models. Adding to this the open-ended nature of the TSRSP, it seems only natural to embed solution methods into a rolling horizon approach (see e.g. Sethi and Sorger (1991)), e.g. as in Meng et al. (2015), and/or to incorporate a value of time to account for the occupation of ships, e.g. as in Appelgren (1969). Both these approaches should then incorporate any information on future demand. The dynamic and stochastic nature of the problem also calls for considerations to solution robustness and persistence, i.e. consistency between planning and realization, so that the whole fleet is not rescheduled due to a small delay with one ship.
- As already mentioned, the work presented in Norstad et al. (2015), Vilhelmsen and Lusby (2014), Stålhane et al. (2014) and Hemmati et al. (2015) relates the TSRSP to the broader context of the supply chain. In industrial shipping, where the ship operator is also the cargo owner, the supply chain aspect seems obvious and much literature in this area also includes this aspect, see Christiansen et al. (2013). Even though this aspect seems less obvious in a tramp shipping setting, it is still a very interesting topic for further research, and we see two main reasons for this. First, in a market with tough competition and freight rates already critically low, it seems obvious for tramp operators to instead improve their customer service through inventory management services in order to attract more business. Second, optimising over the entire supply chain, or at least parts of it, rather than just the isolated TSRSP, will no doubt enable overall cost reductions. With the right collaboration setup, some part of these cost reductions will reflect back on the tramp operator and allow an increase in profit. Furthermore, we note that including information from the supply chain will also give a better understanding of future demand and thereby assist in resolving the issue from above regarding dynamic and stochastic modelling approaches.

Acknowledgements

The research presented in this paper has been partly funded by The Danish Maritime Fund and we gratefully acknowledge their financial support.

References

- H. Andersson, A. Hoff, M. Christiansen, G. Hasle, and A. Lokketangen. Industrial aspects and literature survey: Combined inventory management and routing. *Computers & Operations Research*, 37(9):1515–1536, 2010.
- H. Andersson, M. Christiansen, and K. Fagerholt. The maritime pickup and delivery problem with time windows and split loads. *INFOR*, 49(2):79–91, 2011a.
- H. Andersson, J.M. Duesund, and K. Fagerholt. Ship routing and scheduling with cargo coupling and synchronization constraints. *Computers & Industrial Engineering*, 61(4):1107–1116, 2011b.
- L.H. Appelgren. A column generation algorithm for a ship scheduling problem. *Transportation Science*, 3:53–68, 1969.
- L.H. Appelgren. Integer programming methods for a vessel scheduling problem. *Transportation Science*, 5:64–78, 1971.
- G. Brønmo, M. Christiansen, K. Fagerholt, and B. Nygreen. A multi-start local search heuristic for ship scheduling - a computational study. *Computers & Operations Research*, 34:900–917, 2007a.
- G. Brønmo, M. Christiansen, and B. Nygreen. Ship routing and scheduling with flexible cargo sizes. *Journal of the Operational Research Society*, 58(9):1167–1177, 2007b.
- G. Brønmo, B. Nygreen, and J. Lysgaard. Column generation approaches to ship scheduling with flexible cargo sizes. *European Journal of Operational Research*, 200(1):139–150, 2010.
- B.D. Brouer, J.F. Alvarez, C.E.M. Plum, D. Pisinger, and M.M. Sigurd. A base integer programming model and benchmark suite for liner-shipping network design. *Transportation Science*, 48(2):281–312, 2014.
- E.R. Butchers, P.R. Day, A.P. Goldie, S. Miller, J.A. Meyer, D.M. Ryan, A.C. Scott, and C.A. Wallace. Optimized crew scheduling at air new zealand. *Interfaces*, 31(1):30–56, 2001.
- K.K. Castillo-Villar, R.G. González-Ramírez, P.M. González, and N.R. Smith. A heuristic procedure for a ship routing and scheduling problem with variable speed and discretized time windows. *Mathematical Problems in Engineering*, 2014.
- M. Christiansen and B. Nygreen. A method for solving ship routing problems with inventory constraints. *Annals of Operations Research*, 81:357–378, 1998.
- M. Christiansen, K. Fagerholt, and D. Ronen. Ship routing and scheduling: Status and perspectives (review). *Transportation Science*, 38(1):1–18, 2004.
- M. Christiansen, K. Fagerholt, B. Nygreen, and D. Ronen. Maritime transportation. In C. Barnhart and G. Laporte, editors, *Transport. Handbooks in Operations Research and Management Science*, vol. 14, chapter 4, pages 189–284. Elsevier, North-Holland, Amsterdam, 2007.
- M. Christiansen, K. Fagerholt, B. Nygreen, and D. Ronen. Ship routing and scheduling in the new millennium (review). *European Journal of Operational Research*, 228(3):467–483, 2013.
- M.E. Cóccola, R. Dondo, and C.A. Méndez. A MILP-based column generation strategy for managing large-scale maritime distribution problems. *Computers & Chemical Engineering*, 72:350–362, 2015.
- M. Conforti, G. Cornuéjols, A. Kapoor, and K. Vušković. Perfect, ideal and balanced matrices. *European Journal of Operational Research*, 133(3):455–461, 2001.

- G. Desaulniers, J. Desrosiers, I. Ioachim, M.M. Solomon, F. Soumis, and D. Villeneuve. A unified framework for deterministic time constrained vehicle routing and crew scheduling problems. In T. Crainic and G. Laporte, editors, *Fleet Management and Logistics*, chapter 3, pages 57 – 94. Kluwer Academic Publishers, 1998.
- G. Desaulniers, J. Desrosiers, and M.M. Solomon, editors. *Column Generation*. Springer, New York, 2005.
- M. Desrochers and F. Soumis. A reoptimization algorithm for the shortest path problem with time windows. *European Journal of Operational Research*, 35:242 – 254, 1988.
- J. Desrosiers, Y. Dumas, M.M. Solomon, and F. Soumis. Time constrained routing and scheduling. In M.O. Ball, T.L. Magnanti, C.L. Monma, and G.L. Nemhauser, editors, *Network Routing. Handbooks in Operations Research and Management Science*, vol. 8, chapter 2, pages 35–139. North-Holland, Amsterdam, 1995.
- M. Dror. Note on the complexity of the shortest path models for column generation in VRPTW. *Operations Research*, 42(5):977–978, 1994.
- K. Fagerholt. Ship scheduling with soft time windows: An optimisation based approach. *European Journal of Operational Research*, 131(3):559–571, 2001.
- K. Fagerholt. A computer-based decision support system for vessel fleet scheduling - experience and future research. *Decision Support Systems*, 37(1):35–47, 2004.
- K. Fagerholt and M. Christiansen. A combined ship scheduling and allocation problem. *Journal of the Operational Research Society*, 51:834 – 842, 2000a.
- K. Fagerholt and M. Christiansen. A travelling salesman problem with allocation, time window and precedence constraints - an application to ship scheduling. *International Transactions in Operational Research*, 7(3):231 – 244, 2000b.
- K. Fagerholt and D. Ronen. Bulk ship routing and scheduling: Solving practical problems may provide better results. *Maritime Policy & Management*, 40(1):48–64, 2013.
- K. Fagerholt, L.M. Hvattum, T.A.V. Johnsen, and J.E. Korsvik. Routing and scheduling in project shipping. *Annals of Operations Research*, pages 1–15, 2011.
- R.A. Gatica and P.A. Miranda. Special issue on latin-american research: A time based discretization approach for ship routing and scheduling with variable speed. *Networks and Spatial Economics*, 11(3):465–485, 2011.
- S. Gélinas, M. Desrochers, J. Desrosiers, and M.M. Solomon. A new branching strategy for time constrained routing problems with application to backhauling. *Annals of Operations Research*, 61:91–109, 1995.
- F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13(5):533–549, 1986.
- A. Hemmati, L.M. Hvattum, K. Fagerholt, and I. Norstad. Benchmark suite for industrial and tramp ship routing and scheduling problems. *INFOR*, 52(1):28–38, 2014.
- A. Hemmati, M. Stålhane, L.M. Hvattum, and H. Andersson. An effective heuristic for solving a combined cargo and inventory routing problem in tramp shipping. *Computers & Operations Research*, 64:274 – 282, 2015.
- ICS. International Chamber of Shipping Website (accessed 14.08.2015), <http://www.ics-shipping.org/shipping-facts/shipping-and-world-trade>.
- S. Irnich. Resource extension functions: properties, inversion, and generalization to segments. *OR Spectrum*, 30(1):113–148, 2008.

- S. Irnich and G. Desaulniers. Shortest path problems with resource constraints. In G. Desaulniers, J. Desrosiers, and M.M. Solomon, editors, *Column Generation*, chapter 2, pages 33 – 66. Springer, 2005.
- J.N. Jung, M.N. Kang, H.R. Choi, H.S. Kim, B.J. Park, and C.H. Park. Development of a genetic algorithm for the maritime transportation planning of car carriers. *Dynamics in logistics*, 6: 481–488, 2011.
- M.H. Kang, H.R. Choi, H.S. Kim, and B.J. Park. Development of a maritime transportation planning support system for car carriers based on genetic algorithm. *Applied Intelligence*, 36(3): 585–604, 2012.
- S.-H. Kim and K.-K. Lee. An optimization-based decision support system for ship scheduling. *Computers & Industrial Engineering*, 33(3-4):689–692, 1997.
- K. Kobayashi and M. Kubo. Optimization of oil tanker schedules by decomposition, column generation, and time-space network techniques. *Japan Journal of Industrial and Applied Mathematics*, 27(1):161–173, 2010.
- J.E. Korsvik and K. Fagerholt. A tabu search heuristic for ship routing and scheduling with flexible cargo quantities. *Journal of Heuristics*, 16(2):117–137, 2010.
- J.E. Korsvik, K. Fagerholt, and G. Laporte. A tabu search heuristic for ship routing and scheduling. *Journal of the Operational Research Society*, 61(4):594–603, 2010.
- J.E. Korsvik, K. Fagerholt, and G. Laporte. A large neighbourhood search heuristic for ship routing and scheduling with split loads. *Computers & Operations Research*, 38(2):474–483, 2011.
- S.A. Lawrence. *International Sea Transport: The Years Ahead*, pages 1–316. Lexington Books, Lexington, MA., 1972.
- D.-Y. Lin and H.-Y. Liu. Combined ship allocation, routing and freight assignment in tramp shipping. *Transportation Research Part E: Logistics and Transportation Review*, 47(4):414–431, 2011.
- E.F. Magirou, H.N. Psaraftis, and T. Bouritas. The economic speed of an oceangoing vessel in a dynamic setting. *Transportation Research Part B: Methodological*, 76:48–67, 2015.
- F. Malliappi, J.A. Bennell, and C.N. Potts. A variable neighborhood search heuristic for tramp ship scheduling. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6971 LNCS:273–285, 2011.
- Q. Meng, S. Wang, and C.-Y. Lee. A tailored branch-and-price approach for a joint tramp ship routing and bunkering problem. *Transportation Research Part B: Methodological*, 72:1 – 19, 2015.
- N. Mladenović and P. Hansen. Variable neighborhood search. *Computers & Operations Research*, 24(11):1097–1100, 1997.
- K.-H. Neo, H.-C. Oh, and I.A. Karimi. Routing and cargo allocation planning of a parcel tanker. In *16th European Symposium on Computer Aided Process Engineering and 9th International Symposium on Process Systems Engineering*, pages 1985 – 1990, 2006.
- I. Norstad, K. Fagerholt, and G. Laporte. Tramp ship routing and scheduling with speed optimization. *Transportation Research Part C: Emerging Technologies*, 19(5):853–865, 2011.
- I. Norstad, K. Fagerholt, L.M. Hvattum, H.S. Arnulf, and A. Bjørkli. Maritime fleet deployment with voyage separation requirements. *Flexible Services and Manufacturing Journal*, 27(2-3): 180–199, 2015.

- H.-C. Oh and I.A. Karimi. Routing and scheduling of parcel tankers: a novel solution approach. In A. Bruzzone, F. Longo, Y. Merkuriev, G. Mirabello, and M.A. Piera, editors, *The 11th International Workshop on Harbor Maritime Multimodal Logistics Modeling and Simulation*, pages 98 – 103, September 2008.
- I. H. Osman and G Laporte. Metaheuristics: A bibliography. *Annals of Operations Research*, 63: 513–623, 1996.
- M. Padberg. On the facial structure of set packing polyhedra. *Mathematical Programming*, 5(1): 199–215, 1973.
- D.J. Papageorgiou, G.L. Nemhauser, J. Sokol, M.-S. Cheon, and A.B. Keha. MIRPLib - A library of maritime inventory routing problem instances: Survey, core model, and benchmark results. *European Journal of Operational Research*, 235(2):350 – 366, 2014.
- C.E.M. Plum, D. Pisinger, and P.N. Jensen. Bunker purchasing in liner shipping. In Chung-Yee Lee and Qiang Meng, editors, *Handbook of Ocean Container Transport Logistics*, International Series in Operations Research and Management Science, pages 251–278. Springer, 2015.
- D. Ronen. Cargo ships routing and scheduling: Survey of models and problems. *European Journal of Operational Research*, 12(2):119–126, 1983.
- D. Ronen. Ship scheduling: The last decade. *European Journal of Operational Research*, 71(3): 325–333, 1993.
- D. Ronen. Marine inventory routing: Shipments planning. *Journal of the Operational Research Society*, 53(1):108–114, 2002.
- S. Ropke and D. Pisinger. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *TRANSPORTATION SCIENCE*, 40(4):455–472, 2006.
- D.M. Ryan and B. Foster. An integer programming approach to scheduling. *Computer Scheduling of Public Transport. Urban Passenger Vehicle and Crew Scheduling. Proceedings of an International Workshop*, pages 269–280, 1981.
- J. Schönberger, H. Kopfer, and D.C. Mattfeld. A combined approach to solve the pickup and delivery selection problem. In U. Leopold-Wildburger, F. Rendl, and G. Wäscher, editors, *Operations Research Proceedings 2002*, pages 150–155. Springer, 2003. Selected Papers of the International Conference on Operations Research (SOR 2002), Klagenfurt, September 2-5, 2002.
- S. Sethi and G. Sorger. A theory of rolling horizon decision making. *Annals of Operations Research*, 29:387–416, 1991.
- P. Shaw. A new local search algorithm providing high quality solutions to vehicle routing problems, 1997.
- M. Stålhane, H. Andersson, M. Christiansen, J.-F. Cordeau, and G. Desaulniers. A branch-price-and-cut method for a ship routing and scheduling problem with split loads. *Computers & Operations Research*, 39(12):3361–3375, 2012.
- M. Stålhane, H. Andersson, M. Christiansen, and K. Fagerholt. Vendor managed inventory in tramp shipping. *Omega (United Kingdom)*, 47:60–72, 2014.
- P. Toth and D. Vigo. *The vehicle routing problem*. SIAM, 2002.
- UNCTAD. Review of maritime transport 2013
. <http://unctad.org/en/PublicationsLibrary/rmt2013en.pdf>, 2013.
- UNCTAD. Review of maritime transport 2014
. <http://unctad.org/en/PublicationsLibrary/rmt2014en.pdf>, 2014.

-
- C. Vilhelmsen and R.M. Lusby. Tramp ship routing and scheduling with voyage separation requirements. *Technical Report, Department of Management Engineering, Technical University of Denmark*, 2014.
- C. Vilhelmsen, J. Larsen, and R.M. Lusby. A heuristic and hybrid method for the tank allocation problem in maritime bulk shipping. *Technical Report, Department of Management Engineering, Technical University of Denmark*, 2014a.
- C. Vilhelmsen, R.M. Lusby, and J. Larsen. Tramp ship routing and scheduling with integrated bunker optimization. *Euro Journal of Transportation and Logistics*, 3(2):143–175, 2014b.