YAŞAR UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

PHD THESIS

DESIGNING DYNAMIC AND SYNCHRONIZED INTERMODAL TRASPORTATION PLANS FOR CONTAINERS

SEL ÖZCAN TATARİ

THESIS ADVISOR: PROF. DR. DENİZ TÜRSEL ELİİYİ

PH.D. IN INDUSTRIAL ENGINEERING

PRESENTATION DATE: 03.12.2018

BORNOVA / İZMİR
DECEMBER 2018

We certify that, as the jury, we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of the Doctor of Philosophy.

Jury Members:	**Signature:**
Prof. Dr. Deniz TÜRSEL ELİİYİ Yasar University	....
Assoc. Prof. Dr. Yiğit KAZANÇOĞLU Yasar University	....
Assoc. Prof. Dr. Didem ÖZER ÇAYLAN Dokuz Eylul University	....
Asst. Prof. Dr. Adalet ÖNER Yasar University	....
Asst. Prof. Dr. Uğur ELİİYİ Dokuz Eylul University	....

Prof. Dr. Cüneyt GÜZELİŞ
Director of the Graduate School

ABSTRACT

DESIGNING DYNAMIC AND SYNCHRONIZED INTERMODAL TRASPORTATION PLANS FOR CONTAINERS

ÖZCAN TATARİ, Sel
PHD, in Industrial Engineering
Advisor: Prof. Dr. Deniz TÜRSEL ELİİYİ

December 2018

In this thesis, we present a mixed integer linear programming model for the operational level cargo allocation and vessel scheduling problem, where flow-dependent port-stay lengths, transit times and transshipment schedule synchronizations are considered. The proposed model aims to assign shipments to routes to minimize total tardiness, and construct vessel partial schedules for establishing coordination with port authorities to meet the berthing time windows. In addition to mathematical model, novel valid inequalities are proposed, and a benders decomposition algorithm is implemented. Algorithm performances are tested on real-life problem instances. The results show that benders decomposition with valid inequalities yields the best performance on the test instances. The thesis is further extended with the consideration of instant terminal port performances, and an integrated solution framework is proposed for this dynamic problem. The thesis study aims to contribute to both the practitioners and to the state-of-the-art literature.

Key Words: Liner shipping, Cargo allocation, Vessel scheduling, Transshipment problem, Benders Decomposition, Port performance.

ÖZ

KONTEYNER TAŞIMACILIĞINDA DİNAMİK VE SENKRONİZE İNTERMODAL TAŞIMA PLANLARI TASARIMI

ÖZCAN TATARİ, Sel
Doktora Tezi, Endüstri Mühendisliği
Danışman: Prof. Dr. Deniz TÜRSEL ELİİYİ
Aralık 2018

Bu tezde, operasyonel seviyede kargo yükü tahsisi ve gemi çizelgeleme problemi için karma bir tamsayılı doğrusal programlama modeli sunulmaktadır. Önerilen problemde elleçlenen konteyner miktarına bağlı liman kalış süreleri, transit süreler ve aktarma çizelge senkronizasyonları dikkate alınmaktadır. Matematiksel model toplam geç teslim edilen konteyner miktarını azaltmak için limanlara gönderileri bu doğrultuda tahsis ederken bir yandan da rıhtımda kalma zaman pencerelerini ayarlamak için liman yetkilileriyle koordinasyon kurmaya yönelik gemi çizelgelerini oluşturmayı hedeflemektedir. Matematiksel modele ek olarak probleme özgü geçerli eşitsizlikler önerilmiş olup Benders Ayrıştırma algoritması uygulanmıştır. Algoritma performansları gerçek test problemleri üzerinde incelenmiştir. Sonuçlar, geçerli eşitsizlikler ile zenginleştirilmiş Benders Ayrışması yönteminin en iyi performansı verdiğini göstermektedir. Buna ek olarak, gerçek hayat dinamiklerini yansıtmak amacıyla anlık liman performanslarının dikkate alındığı entegre bir çözüm yöntemi de önerilmiştir. Bu tez çalışması ile hem konteyner hat taşımacılığı firmalarına fayda sağlamak, hem de güncel literatüre katkıda bulunmak amaçlanmaktadır.

Anahtar Kelimeler: Hat taşımacılığı, Kargo yükü tahsisi, Gemi çizelgeleme, Aktarma problemi, Benders Ayrıştırması, Liman performansı.

ACKNOWLEDGEMENTS

First of all, I would like to thank my supervisor Prof. Dr. Deniz Türsel Eliiyi for her guidance and patience during this study. Her guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D study.

I would like to express my enduring love to my parents, who are always supportive, loving and caring to me in every possible way in my life. Last but not the least, I would like to thank to my husband and daughter for supporting me spiritually throughout writing this thesis and my life in general.

Sel Özcan Tatari

İzmir, 2018

TEXT OF OATH

I declare and honestly confirm that my study, titled “DESIGNING DYNAMIC AND SYNCHRONIZED INTERMODAL TRASPORTATION PLANS FOR CONTAINERS” and presented as a PhD Thesis, has been written without applying to any assistance inconsistent with scientific ethics and traditions. I declare, to the best of my knowledge and belief, that all content and ideas drawn directly or indirectly from external sources are indicated in the text and listed in the list of references.

Sel Özcan Tatari

Signature

…………………………………..

December 17, 2018

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SYMBOLS AND ABBREVIATIONS

ABBREVIATIONS:

LSC Liner Shipping Company

LSND Liner Shipping Network Design

SAVSP Shipment Assignment and Vessel Scheduling Problem

CTW Connecting Time Windows

LP Linear Programming.

IP Integer Programming

MILP Mixed Integer Linear Programming

MCNF Minimum Cost Network Flow

BD Benders Decomposition

PSP Primal Subproblem

DSP Dual Subproblem

UNCTAD United Nations Conference on Trade and Development

IPCTP Integrated Port Container Terminal Problem

O-D origin-destination

TEU twenty-foot equivalent unit

NDP Network Design Problem

CRP Cargo Routing Problem

FDP Fleet Deployment Problem

VSP Vessel Scheduling Problem

CHAPTER 1
INTRODUCTION

With over 80% of world merchandise trade being carried by sea, maritime transport is the backbone of international trade and globalization. As it is clearly stated in the recent review of maritime transport by the United Nations Conference on Trade and Development (UNCTAD), global seaborne shipments have increased by 4% in 2017, the fastest growth in five years, and the 2018-2023 projection indicates that this percentage will be around 3.8 per year. Furthermore, total containerized trade volumes are estimated at 148 million twenty-foot equivalent units (TEUs) (UNCTAD, 2018).

Among various seaborne transportation alternatives for containers, liner shipping is preferred mostly due to its cheaper freight rates, higher safety level and less environmental hazard (Gelareh and Pisinger, 2011). Liner shipping companies (LSCs) have fleets of different vessels deployed on services, where a predetermined number of homogeneous vessels are operating on each service at regular frequencies. Each service makes round trips and visits predetermined ports in a fixed sequence. In most cases, the shipment of a container through the liner shipping network may include the use of several services to reach from its origin to its destination port. As a result, many containers need to be transshipped. Due to the competitive business environment, the LSCs try to provide efficient and effective cargo routing solutions to their customers by improving their other management tools (e.g. service design and fleet deployment) (Wang et al. 2014).

When the route of a freight is planned, various critical factors need to be considered simultaneously. The most common objective is to perform the transportation with minimum cost. On the other hand, when the sole objective is the on-time delivery of the shipments, which is especially crucial for highly competitive businesses, objective becomes minimizing the *transit time*, i.e., the time it takes to travel from the origin to the destination. For example, for perishable or time-sensitive products having economic/technical depreciation (fashion, computers, etc.), where shorter transit times are strictly enforced, the second type of objective is more relevant (Guericke and Tierney, 2015; Notteboom, 2006; Vad Karsten et al., 2015).

In general, the LSCs prepare their routing plans in terms of shipments. A shipment is defined as a bundle of consolidated containers, each having the same characteristics such as the origin-destination (O-D) pair and the desired transit time. The route of a shipment may include either a sub-path of a single service or a combination of multiple sub-paths of multiple services requiring transshipments. As manufacturers aim to minimize their inventory holding costs, they prefer to send their just-finished products to an LSC as shipments. These shipments are stored temporarily by the LSC until the destined vessel for the container arrives at the origin port (Wang et al., 2014). In addition to the storage times of the shipments at the origin port, the waiting times at the transshipment ports required for connections to other services on the route should also be considered for routes having multiple sub-paths.

For products requiring short transit times, the LSC may prefer the route having the minimum transit time, and if this route includes transshipments, the synchronization of the connections, i.e., schedule coordination, plays a critical role on timely delivery. For instance, if the path of a shipment requires transshipment at port A from vessel i to vessel j, synchronization between the departure of vessel i from port A, and the arrival of vessel j to port A will yield minimum transshipment time. If synchronization is not achieved by the two vessels, the shipment should either be stored at port A until the next vessel of the same service as vessel j arrives, or the shipment should be rerouted. Both yield serious costs to the LSC.

On the other extreme, for shipments having loose deadlines, the LSC may prefer to send through alternative routes in order to increase its services' profit. These alternative routes usually include sub-paths of the underutilized vessels. Such shipments can be sent without any delay despite their waiting times at the origin and/or transshipment ports. In many cases, it is more profitable for the LSC to store or transfer such containers at the ports having low demurrage costs.

As stated in Gelareh et al. (2010), as many alliances were established in order to avoid underutilized vessels operating on transatlantic and transpacific routes, LSCs within these alliances started to determine the best ship size to deploy for these routes. As a result, the market shares of the smaller LSCs started to diminish. Therefore, the small

LSCs should seek alternative ways to increase their market shares. Although vessel scheduling is a tactical level decision, the number of TEUs loaded/unloaded have a huge impact on the port stay durations and influences the arrival/departure times of the vessels on an operational basis. Therefore, it is vital for especially small LSCs to link their shipment assignment and vessel scheduling decisions in a systematic way to increase its long-term profitability and schedule reliability.

In this study, we are motivated from our real-life case partner, the liner shipping agency in Izmir, Turkey, which mainly operates on the Mediterranean and the Black Sea. Our partner has relatively small market share and tries to improve its businesses. Firstly, we focus on a demand flow problem with transshipments on a given (fixed) liner shipping network, considering flow dependent port stay lengths and transit times, in addition to transshipments and arrival time constraints. With the proposed mathematical model, we aim to assign the shipments to the routes to decrease the total tardiness of the shipments, and to construct the partial schedules of the vessels to facilitate the LSC's coordination with the port authorities for the berthing time windows. Secondly, we propose an integrated solution approach which iteratively solves the proposed mathematical model with instant port container terminal performance information.

This thesis has the following differences from the existing literature:

1. Different than most existing studies, our study allows late arrivals of shipments to the destination port. As opposed to the studies of Wang et al. (2014) and (2016b), where the time spent at each port is assumed to be fixed, our proposed formulation calculates the actual schedule of the vessels on each service by computing the port durations of each vessel. We believe that taking non-fixed port stay durations will better reflect the practical dynamics of the liner shipping industry. This is realistic since the time spent at each port is a function of the number of TEUs handled, hence cannot be fixed through the whole planning horizon.

2. The real arrival and departure times can have minor deviations from the planned schedules due to the variable port durations and uncertainties at ports. Our formulation therefore provides more realistic estimations for

the arrival and departure times, which will help the information flow between the LSC and the port authorities prior to berthing.

3. We enforce every candidate shipment within the planning horizon to be sent via a candidate route. We assume that the LSC do not reject sending any shipment. In many real-life cases, due to fierce market competition among small LSCs and the low vessel utilizations, small LSCs prefer to accept all shipment requests. Hence, our problem environment differs from the existing cases in the literature.
4. This thesis study helps the decision maker to examine different solution alternatives for each shipment, which was not considered in the literature previously. Although the LSC may prefer to send the shipments via the route having the minimum transit time, storing some of the shipments at a transshipment port up to some days at no cost can decrease the total cost of the LSC in some cases. Such a strategy may also end up decreasing the total tardiness of the shipments planned. We allow such a flexibility in our solution approach.
5. We believe that our iterative search scheme, which incorporates instant container port terminal performance will provide a novel contribution integrating both the maritime and inland legs of the global supply chain.

With the above differences and contributions, this thesis aims to enrich to the shipment planning literature via the development of more realistic models and effective solutions. The remaining of the thesis document is organized as follows: Chapter 2 reviews the related literature. Chapter 3 introduces the notation used, explains the mathematical formulation in detail and defines the problem. The solution framework, path generation, strengthening constraints as well as Benders decomposition algorithm are presented in Chapter 4. Chapter 5 reports the computational experiments. Chapter 6 describes the integrated solution approach which iteratively solves the mathematical model addressed in Section 4.1 and then measures the instant port performances accordingly. Finally, conclusions and future research opportunities are discussed in Chapter 7.

CHAPTER 2
LITERATURE REVIEW

The decision-making problems on liner shipping can be categorized into three levels. At the strategic level, a liner shipping company makes long-term decisions such as ship fleet size and mix, strategic alliances, and network design. At the tactical level, frequency determination of services, fleet deployment, speed optimization and schedule construction are considered. At the operational level, cargo allocation (i.e., cargo assignment), cargo booking, and rescheduling decisions are taken into consideration. The complexity of decision-making further increases since the problems examined in different decisions levels are also interrelated within each other (Agarwal and Ergun, 2008; Christiansen, 2004).

It is necessary to state that the problems examined in different decisions levels are interrelated within each other. More specifically, a strategic level decision affects the operations in the tactical level, or vice versa (Agarwal and Ergun, 2008). The interrelations among different planning levels on liner shipping are depicted in Figure 2.1.

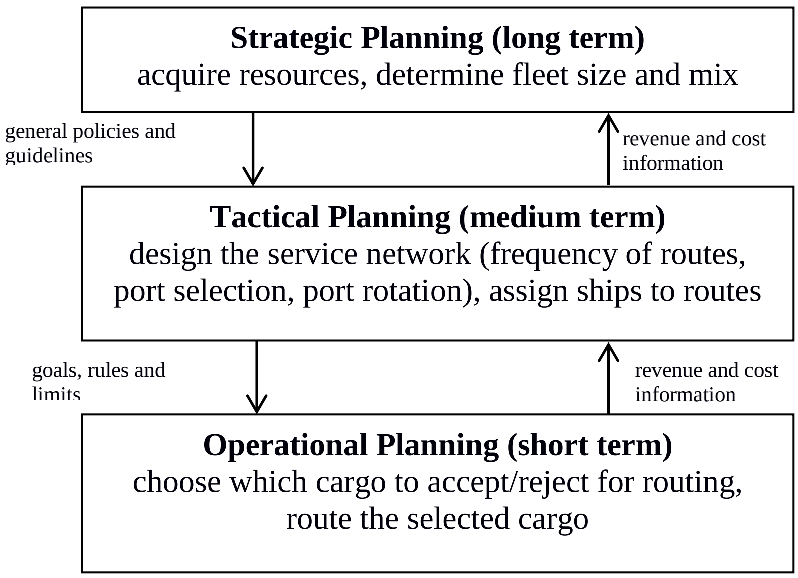


Figure 2.1. Different planning levels for liner shipping (Agarwal and Ergun, 2008).

Cargo routing/allocation problem seeks for the decision of which cargo to accept or reject for shipping and which path(s) to use to ship the selected cargo (Agarwal and Ergun, 2008). Cargo routing problems can be categorized as a subproblem of the cargo assignment problem and studied more as a tactical level problem. They are usually formulated as LP models and the number of containers is relaxed as a non-negative continuous decision variable. They are similar to multi-commodity network flow (MCNF) problems; the formulation of routing in the MCNF problem can be either origin-destination-based (O-D-based) link-flow (Agarwal and Ergun, 2008; Brouer et al., 2011) or alternatively as path-based flow, where each variable indicates a path for a certain commodity (Brouer et al., 2011, Wang and Meng, 2011, 2012; Wang et al., 2013; C.V. Karsten et al., 2015). In addition, Song and Dong (2012) explained how paths can be generated in the simple setting. Meng and Wang (2012) brought out proper methods for path generation.

Transshipment and transit time considerations are rarely studied in the cargo routing and assignment problems. To the best of our knowledge, the first study considering transshipment operations in liner shipping was by Agarwal and Ergun (2008). They proposed a MILP model to solve the integrated ship scheduling and cargo assignment problems simultaneously. Moreover, they presented a column generation-based heuristic and a two-phase Bender's decomposition-based algorithm to solve these two problems separately. However, the proposed model did not consider the transshipment costs while designing the service routes. They proved that the decision version of simultaneous SS and cargo assignment problems is NP-complete by reducing this problem into the well-known 0-1 Knapsack problem, which is also NP-complete (Agarwal and Ergun, 2008).

Álvarez (2009) considered the joint routing and fleet deployment problem by considering the revenues and operating expenses of the liner shipping company including the costs charged due to the transshipments at hubs. Their MILP model seeks for an answer to a tactical level decision of which service is to be served by which vessel. The main assumption in this study was that the liner company already owns its fleet and no strategic level decisions like purchasing a new vessel are allowed. The proposed solution approach exploits a generic interior point algorithm to solve MCNF sub-problems. A case study including 120 ports and 20 vessels of each five different vessel types was presented and the sensitivity of routing and vessel deployment policies to bunker prices was evaluated, as well.

Brouer et al. (2013) extended the problem studied in Álvarez (2009) by incorporating transshipments on the butterfly routes into the objective function without introducing increased complexity and (bi-)weekly frequency in the route generation. They constructed a benchmark suite for LSND problems, e.g., LINER-LIB problem instances including real-world data, and provided a heuristic approach for their model.

Gelareh et al. (2010) studied a hub and spoke network design problem for two liner shipping companies in a competitive environment. They considered the transit time restrictions as well, as the market share of a carrier was determined by both transit time and transportation cost. They proposed a novel MILP model for the LSND problem in a competitive environment. The authors claimed that the proposed model's flexibility allows extensions to fixed time services. In order to solve the problem faster, they used a Lagrangian decomposition method together with a primal bound generation procedure as a result, they eliminated several variables and reduced the problem size.

Reinhardt and Pisinger (2012) addressed an MILP model for the cargo routing problem considering transshipment operations, transshipment costs, a heterogeneous fleet mix and a mix of simple and butterfly routes, where multiple port calls of the same port on a single route was allowed. They developed a branch-and-cut algorithm to solve the problem by gradually adding the transshipment and connectivity constraints to the formulation in case of any violation. They highlighted that their study is the first attempt of an exact method applied to a cargo routing problem where transshipment is taken into account. They tested their algorithm on randomly generated test problems of size 5-15 ports with the forecast of 12 demands and concluded that their algorithm performs well on small test instances (up to 10 ports).

Bell et al. (2011) applied the frequency-based transit assignment model to containers with the objective of minimizing sailing time, container dwell time at the origin port and any intermediate transshipment ports, for liner services with a given frequency. Their model assumed that ships arrived at ports randomly, and that the dwell time at the origin port was half the average headway. Hence, the problem handled was rather at a tactical level than an operational level.

Song and Dong (2013) considered the liner service route design problem with the extensions of butterfly routes and empty container repositioning. They assumed a maximum duration between ports and integrate the route design problem with route-based speed optimization. Moreover, the transshipments of empty containers and their processing time at the ports on the long-haul services were considered in their study. However, the transshipments of laden containers were not allowed. A non-linear model followed by a three-stage solution method was proposed to solve the problem. At the first stage, they limited the route structure solution space by considering only specific types of route structure design issues. At the second stage, they proposed an efficient empty container repositioning algorithm. Lastly, at the third stage, the ship deployment with respect to the ship type, the number of ships and the sailing speed was optimized.

Guericke and Suhl (2013) solved the LSND problem with the integration of cargo routing and speed optimization where the objective is to maximize the total profit of the network. The authors proposed an evolutionary algorithm where a multi-commodity flow network with different layers and linearized bunker costs were exploited to calculate the overall network's profit. They considered transit times in a post-processing step after the cargo allocation. Moreover, they assumed that the hubs ports are where transshipment can take place were known in advance.

Mulder and Dekker (2014) solved the LSND problem through a genetic algorithm where fleet deployment, cargo routing, speed optimization, and empty container repositioning decisions were taken into account at the same time. The problem investigated in their study was to construct a service network and determine the routes used to transport cargo such that the profit is maximized given a certain demand matrix and cost/revenue data. In order to reduce the problem size, aggregation of ports as clusters was employed and initial route networks were obtained through an LP similar to MCNF problem. Transshipment operations were not considered in this study.

Plum et al. (2014) considered the transit time for the design of a single cyclic rotation of a service with up to 25 ports. The problem focused on how to transport a set of selected commodities on a generated single round trip to keep capacity constraints at each leg and the time duration constraints of the commodities. The problem was defined as single liner shipping service design, and an arc-flow and a path-flow MILP formulation together with a branch-and-cut-and-price algorithm were proposed. The pricing sub-problem in the branch-and-cut-and-price algorithm reduced to an elementary shortest path problem with resource constraints. Similar to the study of Reinhardt and Pisinger, the subtour elimination constraints were ignored initially and re-inserted if violation occurs (2012). The proposed algorithm performed well on problems with up to 25 ports. However, the model did not consider the time windows available for berthing.

Wang and Meng (2014) presented a non-linear mixed integer model for the network design problem by taking the transit times into account and proposed a column generation-based heuristic for solving the problem instance for a Europe-Asia network with 12 ports. They solve the network design problem with predetermined port rotations under consideration of transit times without considering transshipments. It is assumed that, for each origin–destination (O–D) port pair, there is a potential container shipment demand, and to fulfill the demand (or a portion of the demand) the real transit time must not be longer than the market level transit time or deadline.

Vad Karsten et al. (2015) formulated the cargo routing problem as a time-constrained MCNF problem by employing a maximum transit time for each commodity. They examined the trade-off between the reductions on bunker cost versus offering short transit times for commodities. Without considering the design of the network, only the cargo flow sub-problem is solved, which determines only how cargo should flow through the network. Moreover, constant transshipment times, transshipment costs and sea durations were assumed at each leg. The two-phase solution methodology includes the generation of the routes in the first phase and the decision of how much cargo should be transported through these routes in the second phase. After proposing both arc-flow and path-flow formulations, the authors indicated that the path-flow formulation performs faster than the arc-flow formulation in the delayed column-generation algorithm.

Guericke and Tierney (2015) studied the cargo routing problem with service levels and leg-based speed optimization. They allowed multiple port calls of vessels on a single service. Similar to the study of Vad Karsten et al. (2015), transshipment times, transshipment costs, transit time restrictions were also employed in this study. On the other hand, a path-flow formulation was utilized within the proposed MILP formulation. In order to decrease the complexity of the problem, the total number of available paths for each O-D pair was assumed to be limited. As they assume the half of the vessel is loaded and unloaded at each port visited, the studied problem is a tactical level problem.

Wang and Meng (2012) presented a liner ship route schedule design model, where the problem focuses on determining the arrival time of a ship at each port call on a route and on the sailing speed on each leg by considering the uncertainties during sailing and port calls. A mixed-integer non-linear stochastic programming model was developed with the objective of minimizing the ship and the expected bunker cost while satisfying a required transit time service level. The proposed model is a tactical level, and the authors assumed that a container route is either a part of one particular ship route or a combination of several ship routes for delivering containers from the original port to the destination port. An exact cutting-plane based solution algorithm was proposed as a solution methodology.

Wang et al. (2013) presented a mixed-integer non-linear non-convex optimization model to find the optimal ship schedule with transit-time-sensitive demand, which was assumed to be a decreasing continuous function of transit time. The model decides the number of ships deployed, leg-based sailing speeds of the ships and the volume of containers for each O-D pair. A branch-and-bound based holistic solution method was developed to solve the proposed model.

Wang et al. (2014) considered the liner shipping route schedule design problem and decided the arrival and departure times at each port call of the route. They assumed that the ports are available within the defined time windows in a week and while constructing the schedule of the vessels, these time windows are taken into account. Furthermore, a port can be visited at most twice in a week on the ship route, and a ship can only be served by one berth. These assumptions implicitly define the set of possible arrival days in a week at the port of call considering all the berths. However, as the schedule of a single service was optimized, transshipments were not considered in this study. The problem was formulated as a mixed-integer nonlinear nonconvex optimization model and an efficient holistic solution approach was proposed to reach global optimality.

Recently, Reinhardt et al. (2016) proposed a speed optimization problem on liner shipping by adjusting the berth times of vessels. Their study covers both transit time restrictions and transshipment times at ports. The proposed model should satisfy that overall transit times for the cargo is retained. They defined the minimum required time for transshipment of a container as Connecting Time Windows (CTW), usually measured from the departure of a vessel, from where the container is unloaded, to the arrival of the other vessel, to where the container is loaded. Moreover, they claimed that hot berthing can occur when the berth locations of the two vessels, where a container is transshipped from one vessel to another, are close to each other. Their mathematical formulation focused on the bunker cost minimization via penalties to limit the number of changes on port visit times. They approximated the cubic bunker consumption function via secant lines and solve large networks optimally in promising computation times.

A chance-constrained optimization model was developed by Wang et al. (2016a) which simultaneously attempts to determine the optimal fleet capacities, cargo allocation and vessel route schedules through schedule coordination. In addition to the weekly deterministic demand coming from the contracted customers, they considered the daily spot demand. A state-augmented shipping network is constructed, where each task on the container shipment activities is represented by a unique link. The authors emphasized the requirement of an efficient solution methodology for this problem which will be able to solve large networks in reasonable computation time.

Wang et al. (2016b) addressed a practical liner container assignment model where the demand is transit-time-sensitive. They generated a novel space-time network which incorporates the OD transit time and presented two novel LP formulations deciding on which demand proportion is satisfied with the objective of profit maximization. The authors proved that the LP formulations are solvable in polynomial time of the size of the liner shipping network. However, their problem is at the tactical level.

Ozcan and Eliiyi (2016) investigated the operational level cargo allocation and vessel scheduling problem which penalizes the positive/negative deviations from the vessels' schedules due to the undetermined port times and sailing times.

Recently, Öztürk et al. (2017) presented a mathematical formulation for transit shipment assignments to the trips outgoing from a transit container terminal. Different than most of the studies in the literature, they employed multiple objectives for the assignment of shipments. As their model is a variant of a well-known NP-hard problem, i.e., generalized assignment problem, they employed two problem-specific heuristic approaches and tested their performances on real-life test instances. Their heuristics are currently being used by a transit agency in real-life.

On the other hand, most research in liner shipping focuses on either the improvement of the maritime operations or the port operations. There is only a single study addressing the optimization of both operations at the same time (Tran et al., 2017), which designs an optimal shipping service considering related inland connections between hinterlands and ports. The mathematical formulation minimizes the total cost including ship costs, port costs inland/feeder transportation costs, inventory holding costs and CO_2 costs. As a solution method, the authors implemented a brute-force algorithm as well as a greedy algorithm. Their solution framework starts by generating all feasible voyages and then it constructs the service route by selecting the ports yielding minimum cost. However, the study is not at the operational level, the focus is not on improving integrated port operations.

In Table 2.1, we provide a literature overview in maritime shipping as a union of the studies concentrated on the VSP, and other maritime shipping problems with Transshipment, Transit Times and Schedule Coordination constraints in general. In the first column, the articles are sorted in chronological order. The second column refers to which maritime shipping problem(s) are studied, i.e. VSP, Network Design Problem (NDP), Cargo Routing Problem (CRP), Fleet Deployment Problem (FDP), or a combination. Columns 3, 4 and 5 indicate whether transshipment, transit times or schedule coordination is considered. We classify the articles as strategic, tactical and/or operational with respect to their decision-making level in the last column.

Table 2.1. Overview of literature in maritime shipping.

Study	*Problem*	*Transshipment*	*Transit Time*	*Schedule Coordination*	*Decision - making level*
Agarwal and Ergun (2008)	NDP	X	X		Tactical
Álvarez (2009)	NDP	X			Tactical
Gelareh et al. (2010)	NDP	X	X		Strategic/ Tactical
Bell et al. (2011)	CRP	X			Tactical
Reinhardt and Pisinger (2012)	NDP + FDP	X			Tactical
Wang and Meng (2012)	VSP				Tactical
Brouer et al. (2013)	NDP	X			Tactical
Wang et al. (2013)	CRP + VSP		X		Tactical
Song and Dong (2013)	CRP	X (Empty Containers only)			Tactical
Guericke and Suhl (2013)	NDP	X	X		Tactical
Mulder and Dekker (2014)	NDP + FDP + CRP		X		Tactical
Plum et al. (2014)	NDP		X		Tactical
Wang and Meng (2014)	NDP		X		Tactical
Wang et al. (2014)	VSP				Tactical/ Operational
Vad Karsten et al. (2015)	CRP	X	X		Operational
Guericke and Tierney (2015)	CRP	X	X		Tactical
Reinhardt et al. (2016)	SOP + VSP		X	X	Tactical
Wang et al. (2016b)	CRP	X	X		Tactical
Wang et al. (2016a)	FDP + VSP + CRP	X	X	X	Tactical (Demand Uncertainty)
Öztürk et al. (2017)	CRP	X	X		Operational
This thesis	VSP + CRP	X	X	X	Operational

As derived from Table 2.1, the maritime shipping problems studied in the literature are usually at the tactical level rather than operational, and they usually consider either transshipment operations and/or transit time restrictions. The schedule coordination restrictions are rarely mentioned in the literature, so far.

In this study, we address the joint CRP and VSP at the operational level regarding transshipment operations, transit times and schedule coordination restrictions. Our thesis study differs from the existing papers in the literature as it focuses on joint vessel scheduling and cargo allocation as a demand flow problem. In this respect, our thesis problem concurrently determines the port stay lengths while maintaining schedule coordination with the port authorities taking into account the instant port performances.

CHAPTER 3
THE SHIPMENT ASSIGNMENT AND VESSEL SCHEDULING PROBLEM

In this chapter, the assumptions and the formulation of the Shipment Assignment and Vessel Scheduling Problem (SAVSP) is explained in detail, and an illustrative example is provided for a better understanding. First, the problem environment and the notation are described.

We consider an LSC that operates mainly on the Mediterranean, West Africa and the Black sea, where all its services, denoted by the set *S*, provide a regular frequency, i.e. daily or weekly. The set V represents the vessels operating on services. We assume that a subset of homogeneous vessels, denoted by V_s are operating on each service *s,* $V_s \subseteq V$. The set P includes all feasible routes (path) and we define a subset P_b, $P_b \subseteq P$, for each shipment *b,* where the route $p \in P_b$ includes the ordered set of ports visited by vessel *v*. The first port on each route p represents the origin port of the corresponding shipment, whereas the last port indicates the destination port. In addition to the origin and destination ports, loading/unloading operations are performed at the transshipment ports. There is no capacity restriction on the arcs of each service. That is, no restriction exists on the amount of shipment transported through each leg.

In most liner shipping networks, a port may be visited several times through the vessel's round trip. In order to differentiate these unique vessel-port pairs, we replicate some ports depending on the total number of visits of the corresponding vessel during its round trip and generate a *modified* service network. For a better understanding, consider the illustrative example in Figure 3.1 (a). Assume that the shipment will be sent from the origin port *Thessaloniki, Greece* (GRSKG) to the transshipment port *Marport, Turkey* (TRMRP) using vessel 2, and then from TRMRP to the destination port *Koper, Slovenia* (SIKOP) using vessel 1. To represent the unique vessel-port subsets for route p, we replicate the ports in the visiting sequence and relabel them with unique numbers.

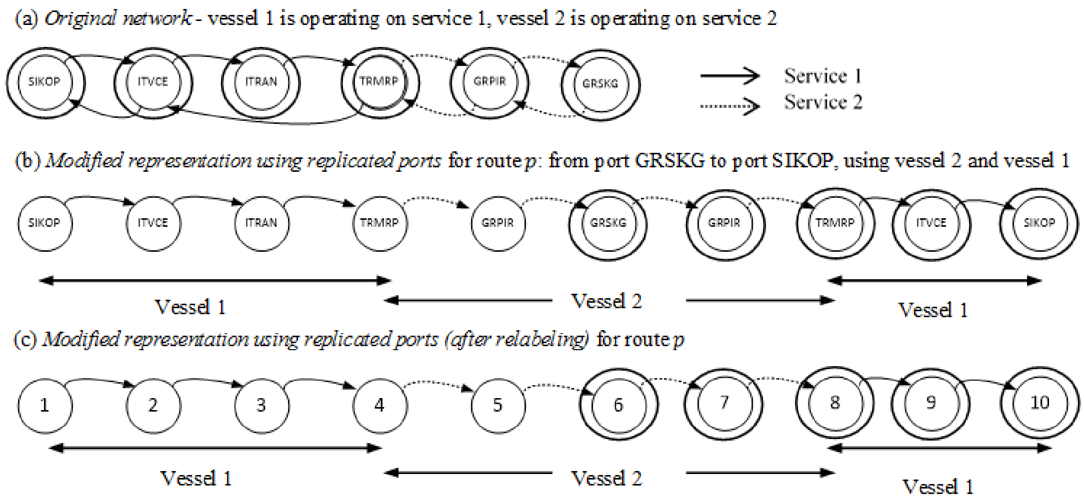
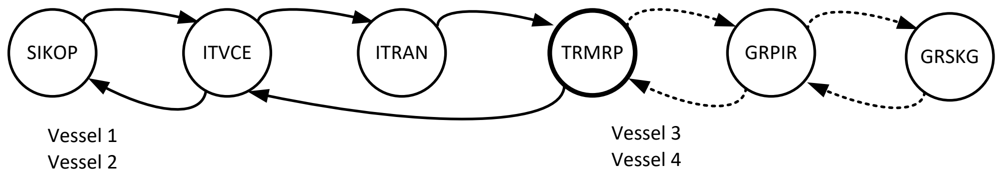


Figure 3.1. An illustrative example to represent the modified service network

As depicted in Figure 3.1 (b), the double-circled ports with the corresponding vessels are included in the subset for route p. Afterwards, the unique vessel-port pairs on the network are defined. The origin port GRSKG, for example, is represented with 6 in Figure 3.1 (c) and the ordered subset of vessel-port pairs for route p are constructed as $\{(2,6),(2,7),(2,8),(1,8),(1,9),(1,10)\}$ where, $(2,6)$ corresponds to vessel 2 and port 6 and $(2,7)$ corresponds to vessel 2 and port 7 etc. (see Figure 3.1 (c)).

3.1. An Illustrative Example

For a better understanding of how the mathematical model behaves, we provide an example in this section. Consider a LSC having two services, namely service 1 and service 2. Figure 3.2 depicts this small network. Service 1 has the port rotation of Koper-Venice-Ravenna-Istanbul-Venice-Koper, and service 2 has Istanbul-Piraeus-Thessaloniki-Piraeus-Istanbul. Assume that there is a fixed number of homogeneous vessels operating on each service and the services operate on a 3-day frequency. That is, the difference of the departure times of vessel 1 and 2 from Koper, and of vessel 3 and 4 from Istanbul is 72 hours. In a 15-day planning horizon, at least two homogeneous vessels operate on each service. Sea durations between ports are deterministic and fixed as 48 hours. Port stays of vessels depend on the number of TEUs loaded/unloaded; the departure time can be determined by adding the current port stay to the port arrival time, whereas the arrival time to the next port can be found by adding the leg duration to the departure time.



Service 1: SIKOP-ITVCE-ITRAN-TRMRP-ITVCE-SIKOP

Service 2: TRMRP-GRPIR-GRSKG-GRPIR-TRMRP

SIKOP: Koper, Slovenia
ITVCE: Venice, Italy
ITRAN: Ravenna, Italy
TRMRP: Istanbul, Turkey
GRPIR: Piraeus, Greece
GRSKG: Thessaloniki, Greece

List of upcoming two vessels operating on each service:
Service 1: Vessel 1 and 2
Service 2: Vessel 3 and 4

Route #	Port visit sequences and vessel information
1	SIKOP-ITVCE-ITRAN-TRMRP using vessel 1 and TRMRP-GRPIR-GRSKG using vessel 3
2	SIKOP-ITVCE-ITRAN-TRMRP using vessel 2 and TRMRP-GRPIR-GRSKG using vessel 4

Figure 3.2. An illustrative example of two services, four vessels and two routes

The liner shipping company should decide on which route the shipment will be sent to minimize the total tardiness of shipments. By selection of any route, the arrival and departure times of the vessel(s) operating on that route will also be determined. We elaborate on two alternative scenarios below to clarify the impact of port duration on tardiness.

In the first scenario, assume that the shipment from Koper to Thessaloniki has a quantity of 100 TEUs and a desired delivery time of 360 hours. For the sake of simplicity, consider the case where no loading/unloading operation takes place at the intermediate ports. Namely, the port stay durations at any port except the origin, destination or transshipment ports, is zero. Also assume a constant loading/unloading time for this shipment, e.g. 4 hours.

Figure 3.2 indicates two feasible routes, namely route 1 and 2, for this shipment. Assume that route 1 is selected. In order to provide on-time delivery, vessel 1 arrives to the origin port Koper at time zero. After 4 hours of loading it departs from that port, and after 48 hours of sailing the vessel arrives at port Venice at time 52. Following the same reasoning, the vessel arrives at port Ravenna and Istanbul at times 100 and 148, respectively. The unloading operation from vessel 1 starts immediately and finishes after 4 hours at time 152. Next, the loading of vessel 3 starts without delay and finishes at time 156. Vessel 3 departs from port Istanbul at time 156 and arrives at ports Piraeus and Thessaloniki at times 204 and 252, respectively. The total transit time is thus calculated as 256. As the transit time is less than the desired delivery time, the shipment is delivered on-time.

Due to the 3-days frequency enforcement of each service, the arrival and departure times of the unused vessels can be derived easily. For example, for vessel 2, the arrival time to port Koper will be 72, and the arrival time of vessel 4 to port Istanbul will be 320 (see Table 3.1).

Table 3.1. Arrival and departure times of vessels for route 1 (Scenario 1).

		Arrival times			
Route #		**Vessel 1**	**Vessel 2**	**Vessel 3**	**Vessel 4**
1	SIKOP	0	72		
	ITVCE	52	120		
	ITRAN	100	168		
	TRMRP	148	216	152	224
	GRPIR			204	272
	GRSKG			252	320
		Departure times			
Route #		**Vessel 1**	**Vessel 2**	**Vessel 3**	**Vessel 4**
1	SIKOP	4	72		
	ITVCE	52	120		
	ITRAN	100	168		
	TRMRP	152	216	156	224
	GRPIR			204	272
	GRSKG			**256**	320

As an alternative route for this shipment, assume that route 2 is selected (see Figure 3.2). In this case, the port stays of vessels 1 and 3 become zero, and the arrival and departure times of vessels 2 and 4 are summarized in Table 3.2. Due to the 3-days frequency, vessel 2 arrives at port Koper, and after 4 hours of loading time it departs from Koper at time 76. Since vessel 2 has no planned loading/unloading operation at ports Venice and Ravenna, the arrival and departure times are identical. Vessel 2 arrives at Istanbul at time 220 and finishes with the unloading operation at time 224, while vessel 4 arrives at the same time and departs at time 228 after 4 hours of loading. After 2 days at sea, vessel 4 arrives at Piraeus at time 276, and departs without delay. As a last leg, vessel 4 arrives at the destination port Thessaloniki at time 324, and the total transit time of this shipment is calculated as 328.

Table 3.2. Arrival and departure times of vessels for route 2 (Scenario 1).

		Arrival times			
Route #		**Vessel 1**	**Vessel 2**	**Vessel 3**	**Vessel 4**
2	SIKOP	0	72		
	ITVCE	48	124		
	ITRAN	96	172		
	TRMRP	144	220	144	224
	GRPIR			192	276
	GRSKG			240	324
		Departure times			
Route #		**Vessel 1**	**Vessel 2**	**Vessel 3**	**Vessel 4**
2	SIKOP	4	76		
	ITVCE	48	124		
	ITRAN	96	172		
	TRMRP	144	224	144	228
	GRPIR			192	276
	GRSKG			240	**328**

As in the solution of route 1, the shipment is not delayed when route 2 is selected. Hence, these two solutions are indistinguishable in terms of the tardiness objective.

In the second scenario we consider two shipments, the same shipment in the previous example and another shipment between the same O-D pair with a quantity of 400 TEUs and a desired delivery time of 305 hours. We assume a constant loading/unloading time at each origin, destination and transshipment port for shipments 1 and 2, as 4 and 16 hours respectively. When route 1 is used for both shipments, the transit time of the shipments will be 320 hours, since the total time spent at ports Koper, Istanbul and Thessaloniki is increased to 24 hours. It can be observed that the second shipment is delayed by 15 hours. The arrival and departure times of the vessels can be examined in Table 3.3.

Now consider the following solution, the shipment with the 360-hour deadline is sent through route 2 and the shipment with the 305-hour deadline is sent through route 1. Table 3.4 summarizes the arrival and departure times for this solution, and it can be seen that, with this second solution, the delay for both shipments is decreased to zero. The first shipment arrives at Istanbul at time 220 and should wait 28 hours for the arrival of vessel 4.

Table 3.3. Arrival and departure times of vessels for route 1 (Scenario 2).

		Arrival times			
Route #		**Vessel 1**	**Vessel 2**	**Vessel 3**	**Vessel 4**
1	SIKOP	0	72		
	ITVCE	68	120		
	ITRAN	116	168		
	TRMRP	164	216	184	256
	GRPIR			252	304
	GRSKG			300	352
		Departure times			
Route #		**Vessel 1**	**Vessel 2**	**Vessel 3**	**Vessel 4**
1	SIKOP	20	72		
	ITVCE	68	120		
	ITRAN	116	168		
	TRMRP	184	216	204	256
	GRPIR			252	304
	GRSKG			**320**	352

Table 3.4. Arrival and departure times of vessels for routes 1&2 (Scenario 2).

		Arrival times			
Route #		**Vessel 1**	**Vessel 2**	**Vessel 3**	**Vessel 4**
1&2	SIKOP	0	72		
	ITVCE	64	124		
	ITRAN	112	172		
	TRMRP	160	220	176	248
	GRPIR			240	300
	GRSKG			288	348
		Departure times			
Route #		**Vessel 1**	**Vessel 2**	**Vessel 3**	**Vessel 4**
1&2	SIKOP	16	76		
	ITVCE	64	124		
	ITRAN	112	172		
	TRMRP	176	224	192	252
	GRPIR			240	300
	GRSKG			**304**	**352**

The illustrative example in this section indicates how port durations affect the delay of shipments, and how a solution without any delay can be achieved regardless of the waiting times at the origin and transshipment ports. In many real-world cases, it is more profitable for the LSC to store containers at interim ports having low demurrage costs. Moreover, although vessel scheduling is a tactical level decision, the number of TEUs loaded/unloaded have a huge impact on the port durations and hence, it influences the arrival/departure times of the vessels on an operational basis. Therefore, it is significant for the LSC to take shipment assignment and vessel scheduling decisions in a systematic way so as to increase profitability and schedule reliability.

3.2. The Mathematical Model

The proposed formulation assigns the shipments to the routes to decrease total tardiness, while concurrently constructing the partial schedules of the vessels to facilitate the LSC's coordination with the port authorities for the berthing time windows. The sets and parameters used in the formulation are defined below.

Sets:

B: set of shipments $b = 1,2, \ldots$

P: set of routes for all shipments $p = 1,2, \ldots$

S: set of services $s = 1,2, \ldots$

N: set of ports $n = 1,2, \ldots$

V: set of vessels $v = 1,2, \ldots 3$

L: set of route legs $l = 1,2, \ldots$

P_b: set of candidate routes for shipment b, $P_b \subseteq P$

VN_p: set of vessel-port pairs visited through route p

$P_{(v,n)}$: set of routes including the vessel-port pair (v, n), $P_{(v,n)} \subseteq P$

L_p: set of legs on route p, $L_p \subseteq L$

TS_p: set of transshipment ports on route p, $TS_p \subseteq N$

B_v: set of shipments transported by vessel v, $B_v \subseteq B$

B_n: set of shipments loaded or unloaded at port n, $B_n \subseteq B$

V_s: set of vessels operating on service s, $V_s \subseteq V$

Parameters:

H_p: deadline of shipment b transported via route p, $b \in B, p \in P_b$

Y_p: latest possible arrival time of shipment b transported via route p, $b \in B, p \in P_b$

T_p: maximum required time between legs of route *b* to make the planned transshipment for shipment b, $b \in B, p \in P_b$

S_{pl}: sailing time for leg *l* of route p, $p \in P, l \in L_p$

$HT_p^{(v,n)}$: handling time of the shipment transported by vessel *v* to port *n* on route p, $p \in P, n \in N_p$

F_{v_1,v_2}: minimum required time for vessels v_1 and v_2 to maintain the frequency of service s, $v_1, v_2 \in V_s$

$dest_p$: destination port of route p, $p \in P$

ε: a positive number, close to zero

M: a very large positive number

Below are the decision variables of our mathematical model.

Decision Variables:

$a^{(v,n)}$: the arrival time of vessel *v* to port n, $p \in P_{(v,n)}$, $(v,n) \in VN_p$

$a_p^{(v,n)}$: the arrival time of vessel *v* to port *n* on route p, $p \in P_{(v,n)}$, $(v,n) \in VN_p$

$d_p^{(v,n)}$: the departure time of vessel *v* from port *n* on route p, $p \in P_{(v,n)}$, $(v,n) \in VN_p$

$a_p^{(v,[i])}$: the arrival time of vessel *v* to the *i*th port in the sequence of the service operating on route p, $p \in P_{(v,[i])}$, $(v,[i]) \in VN_p$

$d_p^{(v,[i])}$: the departure time of vessel v to the *i*th port in the sequence of the service operating on route p, $p \in P_{(v,[i])}$, $(v,[i]) \in VN_p$

q_p: delay of route p, $p \in P_b, b \in B$

$$x_p = \begin{cases} 1, & \text{if route } p \text{ is selected} \\ 0, & \text{otherwise} \end{cases} \quad , p \in P_b, b \in B$$

Based on the above definitions, the MILP model for the SAVSP is as follows:

$$min \sum_{b \in B} \sum_{p \in P_b} q_p + \sum_{(v,n) \in VN_p} \varepsilon a^{(v,n)} \quad (1)$$

subject to

$$\sum_{p \in P_b} x_p \geq 1 \qquad b \in B \quad (2)$$

$$d_p^{(v,[n])} + x_p S_{pl} \leq a_p^{(v,[n+1])} \quad p \in P, (v,[n]), (v,[n+1]) \in VN_p, l \in L_p \tag{3}$$

$$\sum_{b \in B} \sum_{p \in P_b} \left(HT_p^{(v,n)} x_p \right) + a_p^{(v,n)} \leq d_p^{(v,n)} + \left(1 - x_p\right) M \quad p \in P,\ (v,n) \in VN_p \tag{4}$$

$$\left(x_p\right) M \geq a_p^{(v,n)} \quad p \in P,\ (v,n) \in VN_p \tag{5}$$

$$Y_p - H_p x_p \leq q_p \quad p \in P \tag{6}$$

$$a_p^{(v_2,m)} - a_p^{(v_1,m)} \leq T_p x_p \quad p \in P,\ \left((v_1,m),(v_2,m) \in VN_p\right) \ni m \in TS_p \tag{7}$$

$$a_p^{(v_1,m)} - d_p^{(v_2,m)} \leq 0 \quad p \in P,\ \left((v_1,m),(v_2,m) \in VN_p\right) \ni m \in TS_p \tag{8}$$

$$a_p^{(v,n)} \leq a^{(v,n)} \quad p \in P,\ (v,n) \in VN_p \tag{9}$$

$$a^{(v_2,[0])} - a^{(v_1,[0])} \geq F_{v_1,v_2} \quad v_1, v_2 \in V_s, s \in S \tag{10}$$

$$a^{(v,n)}, a_p^{(v,n)}, d_p^{(v,n)} \geq 0 \quad p \in P,\ (v,n) \in VN_p \tag{11}$$

$$x_p \in \{0,1\},\ q_p \geq 0, \text{integer} \quad p \in P \tag{12}$$

The first term in the objective function (1) minimizes the total tardiness whereas the second term fine-tunes the optimal solution (vessel schedules) by forcing the arrival time of the vessel-port pair (v,n) to be equal to the maximum arrival time of the routes of the vessel-port pair (v,n). Constraint set (2) enforces that every shipment should be sent via a candidate route. Constraint set (3) controls the arrival and departure times of the vessels on every route. Constraint set (4) calculates the departure time of each vessel from each visited port. Constraint set (5) forces the arrival time of a vessel to any non-visited port to zero.

Constraint set (6) calculates the tardiness of each shipment. Since our model allows late arrivals of shipments, tardiness becomes positive once a shipment misses its deadline. The minimum connection time required for the synchronization of the transshipment is achieved through constraint set (7). If the arrival time of the loading vessel of the shipment is at least T_p time units larger than the arrival time of its discharging vessel, the synchronization of this transshipment is achieved. Our formulation guarantees this minimum time allowance if the selected route includes a transshipment. Constraint set (8) ensures that, if a transshipment decision is given, the arrival time of the discharging vessel to the transshipment port should be strictly less than the departure time of the loading vessel. Constraint set (9) determines the arrival time of a vessel to each port equals to the maximum arrival time of the shipments on that vessel. Constraint set (10) dictates the desired frequency for the operating vessels for each service. Finally, constraint sets (11)-(12) define the decision variables.

3.2.1. Calculating Partial Vessel Schedules and Port Stays

The largest arrival and departure times among the shipments on each route determine the arrival and departure times of the vessels operating on these routes. For a better understanding, consider the illustrative example in Figure 3.3 where the flow of vessels and shipments are represented on a timeline.

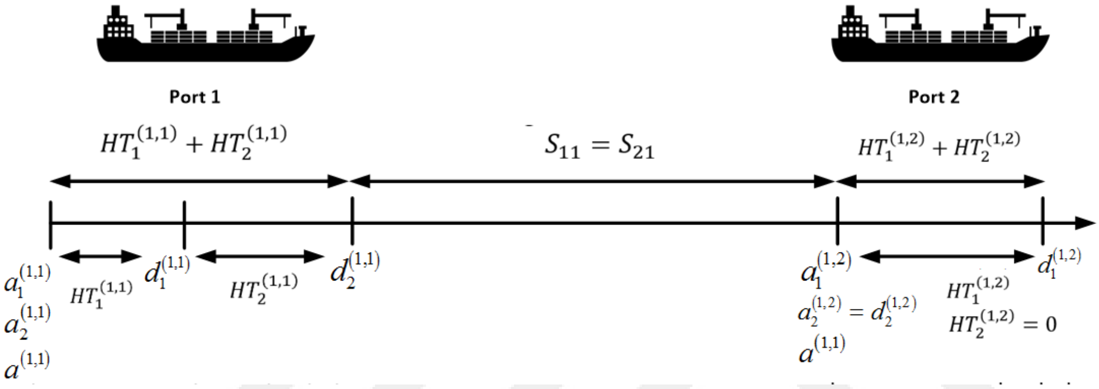


Figure 3.3. An example of the port stay, arrival and departure time calculations

We assume that there are two shipments, namely shipment 1 and 2, both have the origin as port 1, however their destination ports are different. Shipment 1 has to be unloaded at the next port call of vessel 1, i.e., at port 2, whereas shipment 2 will be dropped off at later port visits of vessel 1. Two shipments are planned to be sent through the routes 1 and 2, respectively.

Both routes use vessel 1 for transporting the two shipments. Both shipments are ready before the vessel arrives, hence our formulation yields, $a_1^{(1,1)} = a_2^{(1,1)} = 0$, leading the arrival time of the vessel $a^{(1,1)}$ to be set to zero. Since $HT_p^{(v,n)}$ indicates the total time spent on route p at port n visited by vessel v, the value of $HT_p^{(v,n)}$ has a positive value only for the ports the shipment is planned to be (un)loaded. Moreover, values of the decision variables $d_p^{(v,n)}$ depend on the arrival time of the vessel and the total time spent at the corresponding port. As the loading/unloading operations at a port can be performed sequentially, the largest among the departure times of the routes, determines the departure time of vessel v from port n. The arrival time of vessel v on route p to its next port call is determined by adding its departure time from the current port call to the constant sailing time for each leg l of route p, i.e., S_{pl}. For routes having the same vessel and port call, the sailing times should be equal to each other. Hence, $S_{11} = S_{21}$ (see Figure 3.3).

When vessel 1 arrives to port 2, the unloading operation of shipment 1 starts immediately. Since there is no unloading operation for shipment 2, $HT_2^{(1,2)}$ is zero, which yields to the arrival and departure times of route 2 for the vessel-port pair $(1,2)$ are equal to each other, i.e., $a_2^{(1,2)} = d_2^{(1,2)}$. As only shipment 1 will be unloaded at port 2, the corresponding port duration equals to $HT_1^{(1,2)}$ and vessel 1 will depart from port 2 after the unloading operation finishes i.e., at $d_1^{(1,2)}$. Accordingly, the partial vessel schedules are determined.

3.2.2. Synchronization of Transshipments and Fixed Service Frequency

Let T_p be the maximum required time between legs of route p to make the planned transshipment. Our formulation enforces that a minimum between the arrival times of the vessels are required in order to provide a successful transshipment within planned time interval. As an example, if the arrival time of the second vessel, onto which the shipment will be loaded, is at least T_p units larger than the arrival time of the first vessel, from which the shipment will be unloaded; then the synchronization of this transshipment becomes successful. Our formulation guarantees this minimum time requirement if the selected route includes a transshipment.

We define parameter F_{v_1,v_2} as the desired frequency of the service s, where the vessels v_1 and v_2 operating on this service, $v_1, v_2 \in V_s$. Consider the example illustrated in Figure 3.2. In this case, vessel 1 and 2 operate on service 1 and vessel 3 and 4 operate on service 2. Our formulation enforces that the vessel 2 should arrive to the first port call of service 1, F_{v_1,v_2} times later than the arrival time of vessel 1 to the same port. Similarly, the difference between the arrival times of vessels 3 and 4 to the first port call of service 2 should be $F_{3,4}$.

In the next chapter, we propose the solution methodology developed in this thesis for solving the modeled shipment assignment and vessel scheduling problem.

CHAPTER 4
SOLUTION METHODOLOGY

An exact solution algorithm with two phases is proposed for solving the SAVSP. In the first phase, all feasible routes are generated for each candidate O-D pair with a shipment during the planning horizon. A depth-first-search is implemented for this purpose for searching the routes between any given O-D pair. The generated routes are used to construct the set of routes P as inputs of the SAVSP model that is proposed in Chapter 3.

In the second phase, the model is solved and the best route for each shipment is determined. As the model includes a large number of variables, a group of tightening constraints are introduced to improve computational time performance. In addition, a Benders decomposition algorithm, which provides much faster and effective solutions, is developed for the problem. These are presented in the following sections.

4.1. Path Generation

In the path generation phase, we use a depth-first-search algorithm for searching paths between the given O-D pair. The nodes in the search tree keep respectively the port and the service information where each level of the tree indicates a transshipment operation. At the root node, the information of the origin port is stored. The *first level nodes* include the port-service pairs that can be reached from the origin port directly, i.e., using only a single service. If the destination port is reached at one of the *first level nodes*, a direct path between the O-D pair is obtained and the corresponding node is fathomed. A direct path enforces no transshipment decision, indicating that the shipment should be send between the O-D pair through a single vessel. In order to find the other candidate paths requiring transshipment(s), we continue on searching from the unfathomed *first level nodes* and construct the *second level nodes* by including ports that can be reached from the corresponding *first level node* using a single service. The algorithm terminates when the destination port is reached. In order to provide the search tree with a finite number of nodes, we assume that the search is limited among the eligible ports defined by the LSC, and the port visited through one of the previous nodes of the tree cannot be visited again, thereby eliminating cycles. We provide an illustrative example with Figure 4.1, which depicts a result of the search for possible paths from port 1 to port 4.

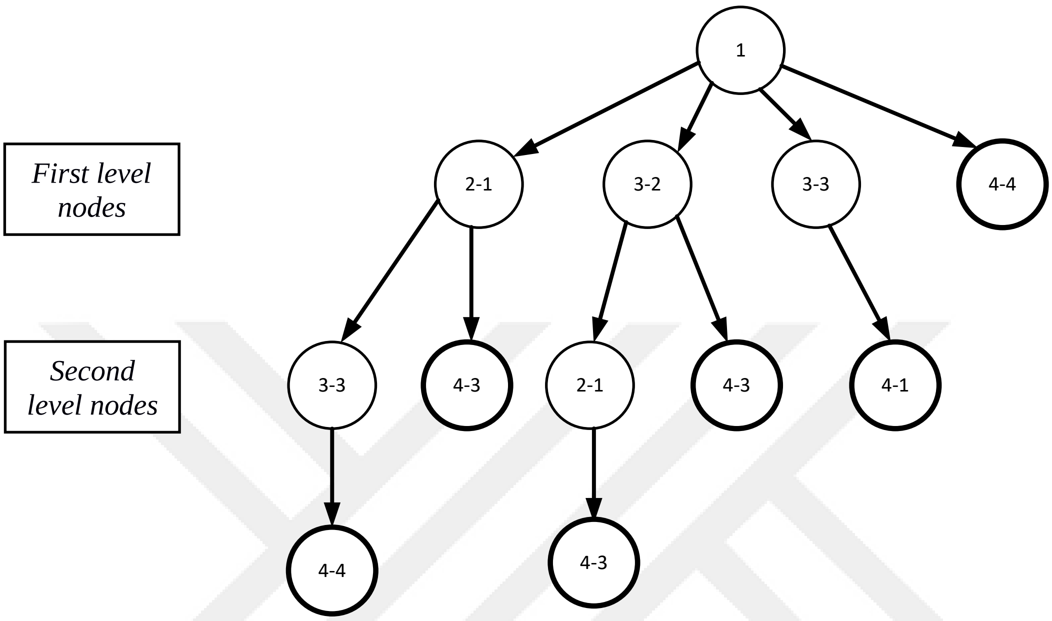


Figure 4.1. An example of the tree representation for path calculation.

In the figure, there is a single direct path from port 1 to port 4 via service 4, whereas paths between ports 1-4 including a single transshipment are:

- from port 1 to port 2 using service 1, from port 2 to port 4 using service 3,
- from port 1 to port 3 using service 2, from port 3 to port 4 using service 3,
- from port 1 to port 3 using service 3, from port 3 to port 4 using service 3.

Moreover, there are only two paths between 1-4, including two transshipments, which are:

- from port 1 to port 2 using service 1, from port 2 to port 3 using service 3, from port 3 to port 4 using service 4.
- from port 1 to port 3 using service 2, from port 3 to port 2 using service 1, from port 2 to port 4 using service 3.

For the real liner shipping problem considered in this thesis, all possible paths can easily be generated for the given enriched service network structure. Note that the number of routes is exponential in input size.

4.2. Introducing Bounds and Valid Inequalities for the SAVSP Model

Instead of assigning an arbitrary large value to parameter M defined by constraint sets (4) and (5), we introduce two tight upper bounds. $PS_{(v,n)}^{UB}$ is defined as an upper bound on the port stay length of vessel v at port n. The port stay length is maximized when the number of shipments (un)loaded at the same port from/to the same vessel is at its maximum. To calculate this upper bound for each vessel-port pair, the related port stays, i.e. $HT_p^{(v,n)}$, are added, assuming that all shipments in vessel v are handled at port n. This leads to an upper bound as:

$$PS_{(v,n)}^{UB} = \sum_{p:\{p \in P_b, b \in B_v\}} HT_p^{(v,n)}.$$

Another upper bound is introduced on the arrival time of vessel v to port n on route p as $A_{p,(v,n)}^{UB}$. There are two components affecting the value of $A_{p,(v,n)}^{UB}$. The first is the sailing time of vessel v until reaching port n, i.e., $\sum_{l=1}^{l=n-1} S_{pl}$, where consecutive $n-1$ legs need to be sailed until port n. The second component includes the upper bound on the total port stay length for $n-1$ port visits, as computed above. The second component is then expressed as $\sum_{k=1}^{n-1} PS_{(v,k)}^{UB}$. Hence, the upper bound becomes:

$$A_{p,(v,n)}^{UB} = \sum_{l=1}^{l=n-1} S_{pl} + \sum_{k=1}^{n-1} PS_{(v,k)}^{UB}.$$

Based on these bounds, we replace constraint sets (4) and (5) in the mathematical model with valid inequalities (4') and (5'), as below:

$$\sum_{b \in B} \sum_{p \in P_b} \left(HT_p^{(v,n)} x_p \right) + a_p^{(v,n)} \le d_p^{(v,n)} + PS_{(v,n)}^{UB} \left(1 - x_p \right)$$

$$p \in P,\ (v,n) \in VN_p \qquad (4')$$

$$A_{p,(v,n)}^{UB} \left(x_p \right) \ge a_p^{(v,n)} \qquad p \in P,\ (v,n) \in VN_p \qquad (5')$$

We also introduce $C_{p,(v,n)}^{LB}$ as a lower bound on the arrival time of vessel v to port n on the route p. This lower bound is composed of two parts: The first part includes the sailing time of vessel v until reaching port n, i.e., $\sum_{l=1}^{l=n-1} S_{pl}$. The second part is the lower bound on total port stay length for $n-1$ port visits. This latter part of $C_{p,(v,n)}^{LB}$ is computed by finding the minimum port stay on port n for every shipment as $\min_p\left\{HT_p^{(v,n)}\right\}$. Hence, vessel v will spend at least $(|B_v|)\min_p\left\{HT_p^{(v,n)}\right\}$ time units at port n, where $|B_v|$ represents the cardinality of set B_v. As a result, the lower bound $C_{p,(v,n)}^{LB}$ is computed as:

$$C_{p,(v,n)}^{LB} = \sum_{l=1}^{l=n-1} S_{pl} + |B_v| \sum_{k=1}^{n-1} \min_p \left\{HT_p^{(v,k)}\right\}.$$

Hence, the following tightening constraint (13) can be introduced into the model:

$$a_p^{(v,n)} \geq x_p C_{p,(v,n)}^{LB} \qquad p \in P,\ (v,n) \in VN_p \qquad (13)$$

4.3. Benders Decomposition Algorithm

As the number of paths and shipments are far above hundreds for the practical case, solving the MILP model in (1) – (12) becomes increasingly difficult due to the large number of decision variables and constraints in the problem. In pilot computational experiments, we have observed that the computation time grows exponentially with the number of constraints in the problem. However, we have observed that the MILP model in (1) – (12) can be decomposed in order to obtain a pair of problems (an LP subproblem and an IP master problem) that can be solved in a faster and more effective manner. In this section, we present a Benders decomposition algorithm for our model.

Benders decomposition (BD) (Benders, 1962) is an algorithm for solving MILPs with linking constraints and is preferred when the master problem has all the integer variables and it is difficult to treat them in subproblems (Agarwal and Ergun, 2008). When the integer variables are fixed, the original problem is decomposed into several LP subproblems, which iteratively generate optimality and/or feasibility cut(s) to the master problem. BD is usually preferred to reduce the number of variables at the expense of an increase in the number of constraints.

Cordeau, et al. applied BD to simultaneously solve the aircraft routing and crew scheduling problems (2001), while an alternative use of BD on power transmission network design problems is addressed by Binato et al. (2001). There are a few studies where BD is applied to the problems in liner shipping. As discussed earlier, Agarwal and Ergun presented a BD-based algorithm to solve the ship scheduling and cargo assignment problems simultaneously (2008). Moreover, Gelareh and Pisinger addressed BD approach to solve the liner shipping network design and fleet deployment problem simultaneously (2010). The interested reader is referred to a recent by study Rahmaniani et al. (2017) for a comprehensive literature review on BD.

For given nonnegative values of $\bar{x}_p$ and $\bar{q}_p$ ($p \in P_b, b \in B$) satisfying constraints (2) and (6), our model reduces to the following *primal subproblem* (PSP) including only the scheduling variables:

$$min \sum_{(v,n)\in VN_p} a^{(v,n)} \tag{14}$$

subject to

$$d_p^{(v,[n])} + \bar{x}_p S_{pl} \leq a_p^{(v,[n+1])} \qquad p \in P, (v,[n]), (v,[n+1]) \in VN_p, l \in L_p \tag{15}$$

$$\sum_{b\in B} \sum_{p\in P_b} \left(HT_p^{(v,n)} \bar{x}_p \right) + a_p^{(v,n)} \leq d_p^{(v,n)} + \left(1 - \bar{x}_p\right) M \qquad p \in P,\ (v,n) \in VN_p \tag{16}$$

$$\left(\bar{x}_p\right) M \geq a_p^{(v,n)} \qquad p \in P,\ (v,n) \in VN_p \tag{17}$$

$$a_p^{(v_2,m)} - a_p^{(v_1,m)} \leq T_p \bar{x}_p \qquad p \in P,\ \left((v_1,m),(v_2,m) \in VN_p\right) \ni m \in TS_p \tag{18}$$

$$a_p^{(v_1,m)} - d_p^{(v_2,m)} \leq 0 \qquad p \in P,\ \left((v_1,m),(v_2,m) \in VN_p\right) \ni m \in TS_p \tag{19}$$

$$a_p^{(v,n)} \leq a^{(v,n)} \qquad p \in P,\ (v,n) \in VN_p \tag{20}$$

$$a^{(v_2,[0])} - a^{(v_1,[0])} \geq F_{v_1,v_2} \qquad v_1, v_2 \in V_s, s \in S \tag{21}$$

$$a^{(v,n)}, a_p^{(v,n)}, d_p^{(v,n)} \geq 0 \qquad p \in P,\ (v,n) \in VN_p \tag{22}$$

Let $\alpha_p^{(v,n)},\ \beta_p^{(v,n)},\ \gamma_p^{(v,n)}, \pi_p^{(v,n)}, \phi_p^{(v,n)},\ \delta_p^{(v,n)}, \theta^{(v,n)} \geq 0\ \ p \in P, (v,n) \in VN_p$ be the dual variables associated with constraints (15) – (21), respectively. Then, the dual of (14) – (21) yields the following *dual subproblem* (DSP):

$$\max \sum_{b\in B} \sum_{(v,n)\in VN_p} \sum_{l\in L_p} \Bigg(\alpha_p^{(v,n)} \bar{x}_p S_{pl} + \beta_p^{(v,n)} \left((1-\bar{x}_p)M - \left(HT_p^{(v,n)} \bar{x}_p \right) \right) +$$

$$\left(\gamma_p^{(v,n)} M \bar{x}_p \right) - \left(\pi_p^{(v,n)} T_p \bar{x}_p \right) \Bigg) + \sum_{s\in S} \sum_{v:v1,v2\in V_s} (F_{v1,v2}) \theta^{(v,0)} \tag{23}$$

subject to

$$\alpha_p^{(v,[n+1])} - \beta_p^{(v,n)} - \gamma_p^{(v,n)} + \pi_p^{(u1,m)} - \pi_p^{(u2,m)} - \phi_p^{(u1,m)} - \delta_p^{(v,n)} \leq 0$$

$$p \in P, (v,n) \in VN_p \left((u_1,m), (u_2,m) \in VN_p \right) \ni m \in TS_p \tag{24}$$

$$-\alpha_p^{(v,[n])} - \beta_p^{(v,n)} + \phi_p^{(u,m)} \leq 0$$

$$p \in P, (v,n) \in VN_p \left((u,m) \in VN_p \right) \ni m \in TS_p \tag{25}$$

$$\delta_p^{(v,n)} + \theta^{(u2,[0])} - \theta^{(u1,[0])} \leq 1$$

$$p \in P, (v,n) \in VN_p, u_1, u_2 \in V_s, s \in S \tag{26}$$

For given values of $\bar{x}_p$ and $\bar{q}_p$, there is always a feasible schedule independent of the assignment of the shipments, as the null vector 0 satisfies constraints (23) – (26). Therefore, the PSP always yields bounded and feasible solutions. So, does the DSP, and therefore only optimality cuts will be added to the *Benders master problem* (BMP). The BMP is formulated as follows:

$$\min \sum_{b\in B} \sum_{p\in P_b} q_p \tag{27}$$

subject to

$$\sum_{p\in P_b} x_p \geq 1 \qquad b \in B \tag{28}$$

$$Y_p - H_p x_p \leq q_p \qquad p \in P \tag{29}$$

$$x_p \in \{0,1\},\ q_p \geq 0, \text{integer} \qquad p \in P \tag{30}$$

The general scheme of the BD algorithm is depicted in Figure 4.2. After partitioning the original model as BMP and DSP, the algorithm solves the two problems repeatedly. It first starts by solving the BMP to optimality, after which the lower bound of the original problem is updated with the objective function value of the master problem. Next, the DSP is solved by taking $\bar{x}_p$ and $\bar{q}_p$ values from the optimal solution of the BMP as input. At each iteration, a new constraint (Benders cut) extracted by the DSP is added to the BMP. Since there will always be a feasible subproblem, the sum of both the objective function value of the master problem and the objective function value of the subproblem provides an upper bound for the original problem. The algorithm terminates when the upper and lower bound of the original problem converge. The computational time performance gains of the BD algorithm are discussed in Section 5.

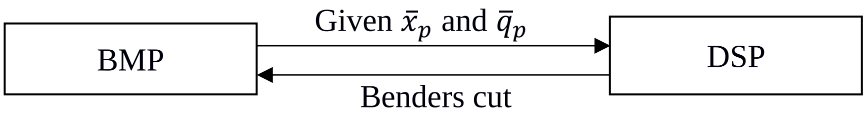


Figure 4.2. Benders Decomposition.

CHAPTER 5
COMPUTATIONAL STUDY

Various problem instances from a real shipping network are used to assess the performances of the solution methodology. We explain the characteristics of our test problems in Section 5.1. Section 5.2 reports the performance of the model in Section 3.1 obtained for different ε values, where we report the runtimes as well as percent improvements. We conduct further analyses with the same model to understand how the number of routes act under different cases and affects the solution quality. Thus, Section 5.3 discusses the performances of the case where we restrict the number of feasible routes for each problem and incorporate only the shortest n routes into the formulation. Additionally, the two variants of the Benders Decomposition algorithm via including and excluding the bounds and valid inequalities presented in the previous chapter are implemented, and their performances are compared in Section 5.4.

5.1. Characteristics of the Test Problems

Four different-sized networks of real data are used, provided by a shipping agency in Izmir, Turkey. The network sizes vary between 15 ports with 2 services and 39 ports with 8 services. The services have various frequencies, daily or weekly. The complete network of the LSC is illustrated on Figure 5.1.

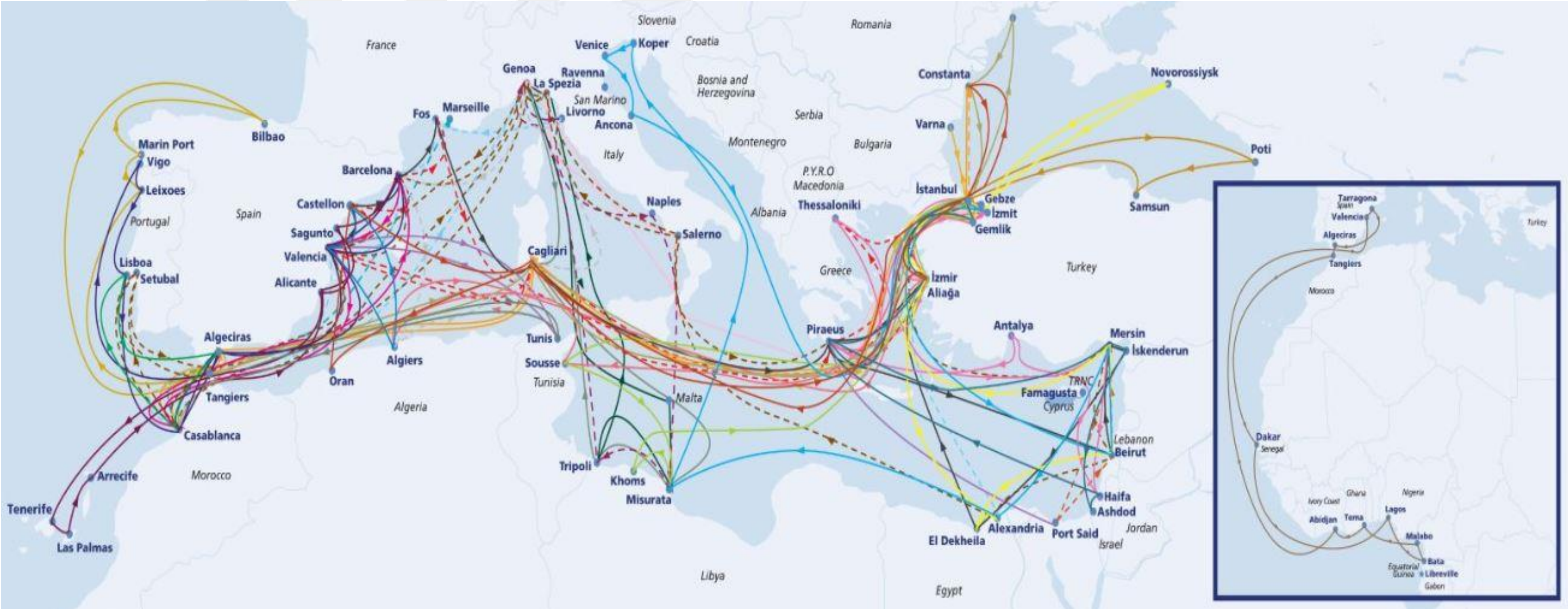


Figure 5.1. The complete network of the liner shipping agency (Arkasline, 2018)

The characteristics of the shipping network for each problem instance is summarized in Table 5.1. The main features of the test problems are defined as follows: The *number of unique ports* is the union of the ports within the considered subset of service, whereas the *total number of ports* includes the duplicated ports, as well. The *maximum number of ports visited on a single service* is an indicator on how the length of the voyages on each single service can vary. Although the *number of ports intersecting by any services* reflects the number of the transshipment alternatives within the considered services from one aspect, the *Average number of ports on services* together with the *total number of unique ports* and the *Number of unique ports* creates another point of view on the complexity of the paths generated.

Table 5.1. Characteristics of the test problems.

Problem	Number of unique ports	Total number of ports	Number of services	Maximum number of ports visited on a single service	Number of ports intersecting by any services	Average number of ports on services
1	15	29	2	10	2	8
2	18	35	3	9	1	8
3	29	80	5	12	1	10
4	39	118	8	14	1	11

For the sake of simplicity, we assume similar-sized shipments. The generated data reflect up to 80% of the real O-D pairs of the mentioned shipping agency. Hourly average container handlings of each port are obtained and used to represent distinct port performances. In addition, we take individual transit times of all shipments as the current transit times, as suggested by the agency. Based on obtained data, the maximum transshipment time is taken as a full 24-hour working day. Due to privacy reasons, we provide only the sizes of our problem instances, and summarize their characteristics in Table 5.2.

Table 5.2. Characteristics of the scenarios tested.

Problem	Scenario	Limit on the number of routes	Number of Transshipments	Number of different O-D pairs	% covered of the considered real network
1	1	-	1	10	20
1	2	-	1	20	60
1	3	-	1	50	100
2	1	20	2	10	30
2	2	20	2	20	65
2	3	20	2	50	100
3	1	20	2	10	20
3	2	20	2	20	35
3	3	20	2	50	100
4	1	20	2	10	20
4	2	20	2	20	60
4	3	20	2	50	100

For each test problem, we assessed our solution algorithms under three different scenarios. For Problem 1, there is no limitation on the number of routes. There are only two services operating on the network for this instance, hence at most one transshipment can take place. We consider different number of O-D pairs as 10, 20 and 50, which respectively mimic 20, 60 and 100% of the real shipping network for this problem. For the remaining problems, where the complexity of the network increases immensely, we limit the number of routes for each O-D pair to 20, which covers most dense and efficient routes on the network. On the other hand, the maximum number of transshipments is set to 2, as suggested by the agency.

All problems are solved using CPLEX 12.6 and the programming code is compiled by Java on a computer having a i7-5500 CPU @2.40Ghz processor and a 16.0 GB RAM. The characteristics of the test instances are listed in Table 5.3. The test instances include the information of the *Problem type*, *Scenario tested*, *The Total Number of different O-D pairs*, and *The Number of shipments for each O-D pair*, respectively.

Table 5.3. Characteristics of the test instances.

Test instance	Problem	Scenario	# of different O-D pairs	# of shipments for each O-D pair
1_1_10_1	1	1	10	1
1_1_10_2	1	1	10	2
2_1_10_1	2	1	10	1
2_1_10_2	2	1	10	2
3_1_10_1	3	1	10	1
3_1_10_2	3	1	10	2
4_1_10_1	4	1	10	1
4_1_10_2	4	1	10	2
1_2_20_1	1	2	20	1
1_2_20_2	1	2	20	2
2_2_20_1	2	2	20	1
2_2_20_2	2	2	20	2
3_2_20_1	3	2	20	1
3_2_20_2	3	2	20	2
4_2_20_1	4	2	20	1
4_2_20_2	4	2	20	2
1_3_50_1	1	3	50	1
1_3_50_2	1	3	50	2
2_3_50_1	2	3	50	1
2_3_50_2	2	3	50	2
3_3_50_1	3	3	50	1
3_3_50_2	3	3	50	2
4_3_50_1	4	3	50	1
4_3_50_2	4	3	50	2

5.2. Sensitivity Analysis for the ε Value

With the anticipation that the ε value has a huge impact on the computational time performance, the SAVSP model was experimented with two different ε values, namely $\varepsilon = 10^{-4}$ and $\varepsilon = 10^{-8}$. We report the runtimes of both cases, as well as the optimality gaps of two cases in Table 5.4.

Table 5.4. Performances of the problems tested with respect to different ε values.

Test instance	Number of feasible routes	runtime (s) with epsilon: 10^{-4}	gap (%) with epsilon: 10^{-4}	runtime (s) with epsilon: 10^{-8}	gap (%) with epsilon: 10^{-8}
1_1_10_1	60	2.89	0	1.7	0
1_1_10_2	120	81.72	0	4.47	0
2_1_10_1	174	24.39	0	1.84	0
2_1_10_2	348	4631.23	0	513.02	0
3_1_10_1	180	1003.97	0	385.09	0
3_1_10_2	360	3549.66	0	858.04	0
4_1_10_1	200	119.27	0	0.21	0
4_1_10_2	400	141506	0	102.15	0
1_2_20_1	102	32.44	0	5.12	0
1_2_20_2	204	2974.69	0	15.19	0
2_2_20_1	400	8554.34	0	901.16	0
2_2_20_2	800	26677.3	0	10126.8	0
3_2_20_1	400	12111.8	0	1323.32	0
3_2_20_2	800	136875	14.05*	10835.1	0
4_2_20_1	388	48160.8	0	366.63	0
4_2_20_2	776	181166	38.02*	11354.2	0
1_3_50_1	264	6551.89	0	628.15	0
1_3_50_2	528	172800	12.87*	103253	0
2_3_50_1	982	191264	27.22*	130135	0
2_3_50_2	1964	178335	45.33*	244175	0
3_3_50_1	748	197658	0	7010.54	0
3_3_50_2	1496	126888	32.91*	347017	0
4_3_50_1	986	72129.8	39.35*	422.05	0
4_3_50_2	1972	214574	57.20*	372590.4	0

*: out of memory

It is apparent from the table that the model with $\varepsilon = 10^{-8}$ outperforms the one with $\varepsilon = 10^{-4}$ in terms of runtime performance for every test problem. In addition, optimality is achieved for every test problem with $\varepsilon = 10^{-8}$. Due to these reasons, $\varepsilon = 10^{-8}$ is used for further computational analyses.

Note that, independent from the epsilon value, the runtime increases as the problem size grows due increase in the number of ports and routes, as expected. Except for the five large test problems, where both the number of different O-D pairs is set to 50 and the number of routes is high, the optimal solutions are achieved within practical time limits for the problem, namely under 3 hours.

5.3. Sensitivity Analysis on the Number of Routes

Once the optimal solutions are analyzed, it was observed that the shortest route alternatives yielded the best solutions for most O-D pairs. To verify this observation, further analyses were conducted to understand how the number of routes act under different cases and affect the solution quality. This kind of analysis also provides practical good solutions for large test problems that fail in terms of runtime performance.

For this purpose, the number of feasible routes is restricted for each problem and only the shortest *n* routes are incorporated into the formulation, where *n* varies depending on the problem size. The performances of case 1, where the number of routes is unrestricted, and case 2, where the *n* shortest routes are chosen, are compared. Table 5.5 summarizes the runtime performances and the solution qualities of each test instance. The third and fifth columns report the runtimes for case 1 and 2, respectively and the percent improvements on the runtime are computed and shown in the last column. We also report the optimality gaps once case 2 is selected, in the last column of Table 5.5.

Table 5.5. Runtime performances and the solution qualities of the test instances.

Test instance	# of feasible routes (case 1)	runtime (s) with epsilon: 10^{-8} (case 1)	# of feasible routes (case 2)	runtime (s) (case 2)	improvement on the runtime (%)	optimality gap (%)
1_1_10_1	60	1.7	20	1.63	4.12	78.12
1_1_10_1	60	1.7	50	0.89	47.65	0
1_1_10_2	120	4.47	50	0.33	92.62	65.08
1_1_10_2	120	4.47	100	1.17	73.83	33.7
2_1_10_1	174	1.84	20	0.32	82.61	72.1
2_1_10_1	174	1.84	50	1.25	32.07	72.1
2_1_10_2	348	513.02	100	56.97	88.9	36.83
2_1_10_2	348	513.02	250	196.17	61.76	35.8
3_1_10_1	180	385.09	20	30.38	92.11	44.74
3_1_10_1	180	385.09	50	138.16	64.12	0
3_1_10_2	360	858.04	100	142.34	83.41	56.5
3_1_10_2	360	858.04	250	385.24	55.1	29.11
4_1_10_1	200	0.21	50	0.04	80.95	33.05
4_1_10_1	200	0.21	100	0.17	19.05	30.18
4_1_10_2	400	102.15	100	10.2	90.01	58.1
4_1_10_2	400	102.15	250	56.03	45.15	22.34
1_2_20_1	102	5.12	20	0.1	100	98
1_2_20_1	102	5.12	50	1.28	75	3.11
1_2_20_2	204	15.19	20	0.15	99.01	76.53
1_2_20_2	204	15.19	50	2.18	85.65	1.81
2_2_20_1	400	901.16	50	17.98	98	56.54
2_2_20_1	400	901.16	100	38.05	95.78	29.3
2_2_20_2	800	10126.81	100	496.52	95.1	28.01
2_2_20_2	800	10126.81	250	4459.1	55.97	19.35
3_2_20_1	400	1323.32	50	101.7	92.31	38.26
3_2_20_1	400	1323.32	100	376.25	71.57	23.35
3_2_20_2	800	10835.08	100	784.46	92.76	31.45
3_2_20_2	800	10835.08	250	4512.7	58.35	17.7
4_2_20_1	388	366.63	100	108.33	70.45	18.34
4_2_20_1	388	366.63	250	294.05	19.8	11.66
4_2_20_2	776	11354.16	100	94.22	99.17	34.1
4_2_20_2	776	11354.16	250	5433.08	52.15	19.29

Table 5.5 (cont.) Runtime performances and the solution qualities of the test instances.

Test instance	# of feasible routes (case 1)	runtime (s) with epsilon: 10-8 (case 1)	# of feasible routes (case 2)	runtime (s) (case 2)	improvement on the runtime (%)	optimality gap (%)
1_3_50_1	264	628.15	20	0.25	99.96	91.42
1_3_50_1	264	628.15	100	2.38	99.62	67.72
1_3_50_1	264	628.15	170	122.07	80.57	43.07
1_3_50_2	528	103253.2	20	1.33	100	58.11
1_3_50_2	528	103253.2	100	219.57	99.79	41.09
1_3_50_2	528	103253.2	170	937.2	99.09	17.78
2_3_50_1	982	130135.1	50	270.66	99.79	66.18
2_3_50_1	982	130135.1	100	1651.7	98.73	48.14
2_3_50_1	982	130135.1	170	6377.53	95.1	21.37
2_3_50_2	1964	244174.6	100	568.81	99.75	36.49
2_3_50_2	1964	244174.6	250	5910.85	97.37	31.09
2_3_50_2	1964	244174.6	500	10343.6	95.4	15.71
3_3_50_1	748	7010.54	50	148.3	97.88	41.84
3_3_50_1	748	7010.54	100	677.51	90.34	25.72
3_3_50_1	748	7010.54	250	2915.3	58.42	22.34
3_3_50_2	1496	347016.7	100	1988.18	99.43	32.52
3_3_50_2	1496	347016.7	250	9572.93	97.24	25.14
3_3_50_2	1496	347016.7	500	15120.2	95.64	14.03
4_3_50_1	986	422.05	100	81.74	80.63	12.23
4_3_50_1	986	422.05	250	160.52	61.97	9.24
4_3_50_2	1972	372590.4	100	4419.15	98.81	53.86
4_3_50_2	1972	372590.4	250	9440.26	97.47	49.1
4_3_50_2	1972	372590.4	500	24941.7	93.31	23.65

We observe that the % optimality gap increases as the number of feasible routes decreases. Also, as depicted in Table 5.5, the number of different O-D pairs and the number of feasible routes have a significant impact on the solution quality. For small networks such as problem 1, for cases where the number of different O-D pairs is below 50, the % optimality gap dramatically increases once the number of feasible routes is restricted.

Note that, for the small network-low number of different O-D pairs combination, the solution time performances for the unlimited case are already reasonable from the operational decision-making point of view. This indicates that there is no need to limit the number of feasible routes and tighten the solution space. However, for the case where the number of different O-D pairs is 50, selecting the shortest *n* routes yields sufficiently good solutions with respect to the optimality gap. The largest contribution of the route restriction can be observed on the largest instance of problem 1, where a reasonable solution with a 17% optimality gap is obtained in 15 minutes (see Table 5.5).

For the moderate networks, i.e., problems 2 and 3, the gain from limiting the number of routes is considerable for the largest instances. The improvement on the runtime is around 95% for problem 2 with an optimality gap of 20%. Note that, no solution can be obtained for these test problem instances when the routes are unlimited. As expected, restricting the number of feasible routes performs better for the cases of large networks, i.e., problem 4. Although some large optimality gaps are observed, obtaining a feasible solution within reasonable/practical runtimes seems to be worth the restriction in the number of routes. However, the proposed solution algorithm still has high optimality gaps for cases having the highest number of feasible routes for each test problem.

5.4. Effects of Valid Inequalities and Benders Decomposition

We first observe the performance of the MILP defined in Chapter 3 in (1) – (12). Afterwards, the valid inequalities explained in Section 4.2 are implemented and the computational performance of the strengthened MILP is reported. As both MILP variants yield low performances within the operational time limits, the BD algorithm (discussed in Section 4.3) is also implemented.

All problems are solved using CPLEX 12.8.0, and the programming code is compiled by Java on a computer having a i7-5500 CPU @2.40Ghz processor and a 16.0 GB RAM. Details and examples of implementing BD in CPLEX can be found in Rudin (2016). Table 5.6 summarizes the computational time performances of all algorithms. All test instances are solved to optimality by all algorithms. The first column in Table 5.6 enumerates the tested instances. The second column shows the total number of feasible routes, i.e. the cardinality of set P. Column 3 presents the exact solution computation times (in seconds) for the SAVSP model in (1) – (12) ($MILP_N$). Columns

4 to 6 lists computation times for the model with the valid inequalities ($MILP_{VI}$), the BD algorithm without valid inequalities (BD_N), and the BD algorithm with the valid inequalities (BD_{VI}), respectively.

Table 5.6. Performances with Valid Inequalities and Benders Decomposition.

Test Instance	# of routes	$MILP_N$	$MILP_{VI}$	BD_N	BD_{VI}
1_1_10_1	60	1.7	1.12	1.15	0.95
1_1_10_2	120	4.47	1.09	1.08	0.82
2_1_10_1	174	1.84	1.06	1.11	0.91
2_1_10_2	348	513.02	329.07	94.8	63.25
3_1_10_1	180	385.09	196.63	74.6	32.2
3_1_10_2	360	858.04	451.16	240.33	81.97
4_1_10_1	200	0.21	0.17	0.09	0.08
4_1_10_2	400	102.15	72.03	17.67	9.54
1_2_20_1	102	5.12	1.13	1.14	0.88
1_2_20_2	204	15.19	5.37	2.04	1.15
2_2_20_1	400	901.16	380.46	237.07	83.2
2_2_20_2	800	10126.81	8843.72	7910.63	5360.6
3_2_20_1	400	1323.32	395.68	257.77	87.26
3_2_20_2	800	10835.08	8499.59	7735.06	5026.4
4_2_20_1	388	366.63	203.24	81.07	44.99
4_2_20_2	776	11354.16	8778.09	6991.72	5870.2
1_3_50_1	264	628.15	475.03	129.32	94.98
1_3_50_2	528	103253.2	100125.8	9942.22	8614.4
2_3_50_1	982	130135.1	100744.1	9975.06	9065.8
2_3_50_2	1964	244174.6	225093.33	24349.1	21976
3_3_50_1	748	7010.54	5411.52	5980.15	2186.1
3_3_50_2	1496	347016.7	29554.43	29171.6	26315
4_3_50_1	986	422.05	269.41	82.08	52.19
4_3_50_2	1972	372590.4	34407.8	33308.8	30542

Figure 5.2 visualizes the relative computational time deviations from the best, for all algorithms over the 24 test instances.

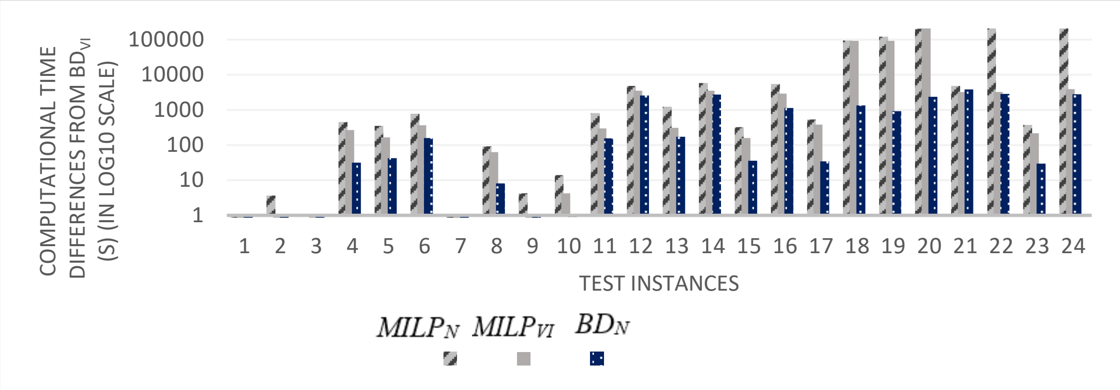


Figure 5.2. Relative performances of algorithms over the test instances.

As expected, the computation time increases as the problem size increases, in parallel with the increase in the number of routes. Although each solution approach has a different characteristic, they act similarly on a quarter of the test instances. For example, for the test instances 1, 2, 3, 7, 9 and 10, where the number of feasible routes is below 204, the computational time performance of the algorithms is comparable. On the other hand, $MILP_N$ has the worst performance over all instances, implying that both the valid inequalities in (4') (5') and (13), and the decomposition approach have a huge impact on the computational time performance.

The sole impact of the valid inequalities can be observed when $MILP_{VI}$ and $MILP_N$ are compared. $MILP_{VI}$ always outperforms $MILP_N$, meaning that the valid inequalities strengthen the formulation and tighten the polyhedron defined by the constraints (2) – (12). The major effect of the proposed valid inequalities can be observed in the largest test instances, namely instances 22 and 24, where the percent improvements on the runtime are 91% and 90%, respectively.

For the test instances 18 - 20, in which the number of different O-D pairs is set to 50, the impact of BD can be best observed. When $MILP_N$ and BD_N are compared, the percent improvements on the runtime for instances 18 – 20 are respectively as 90%, 92% and 90%. Similarly, the dominancy of BD_N over $MILP_{VI}$ for the same test instances is quantified with percent improvements on the runtime. 90%, 90% and 89%

decrease in runtime are achieved for the same test instances when BD_N is employed instead of $MILP_{VI}$. Note that BD_N always outperforms $MILP_{VI}$ in terms of computational time performance.

A clear dominance of the BD_{VI} over the remaining three algorithms prevails. BD_{VI} achieves optimality on 15 test instances in less than 100 seconds, and its performance on the average is around 4800 seconds. The reason of BD_{VI}'s high performance is mainly due to elimination of the unboundedness in the dual subproblems through the introduction of the valid inequalities, thereby eliminating the need to add feasibility cuts to the master problem.

We report the average performances in Table 5.7. Note that the decrease in solution time with the valid inequalities and BD can be observed even for the case of 10 different O-D pairs. The improvement is particularly pronounced for the 50 O-D pair instances. On the average, all instances are solved optimally around 14 hours once $MILP_N$ is employed. The average computational time decreases to nearly 6 hours with the $MILP_{VI}$. Once BD_N is used, the average solution time reduces to less than 2 hours. The best performance with an average of 1.5 hours is obtained with the BD_{VI}.

Table 5.7. Average computation time performances of each algorithm.

# of O-D pairs	$MILP_N$	$MILP_{VI}$	BD_N	BD_{VI}
10	233.32	131.54	53.85	23.72
20	4365.93	3388.41	2902.06	2059.33
50	150653.83	62010.18	14117.28	12355.79
Average (s)	*51751.03*	*21843.38*	*5691.07*	*4812.94*

Table 5.8 reports pairwise comparisons of each solution approach in terms of average solution times. The second and the last column in Table 5.8 indicate the sole performance of the proposed valid inequalities in (4') (5') and (13), whereas the average performance of the BD can be observed from columns 3 and 6. By comparison of columns (2) and (3), we can conclude that BD_N always outperforms $MILP_{VI}$. This implies that BD has relatively higher impact on computational time performance than the valid inequalities. Except the comparison of BD_N and BD_{VI}, the computational time gain is most pronounced for the largest number of O-D pairs, as expected.

Table 5.8. Average pairwise computation time comparisons of the algorithms tested.

# of O-D pairs	$MILP_N$ vs $MILP_{VI}$	$MILP_N$ vs BD_N	$MILP_N$ vs BD_{VI}	$MILP_{VI}$ vs BD_N	$MILP_{VI}$ vs BD_{VI}	BD_N vs BD_{VI}
10	43.62%	76.92%	89.84%	59.06%	81.97%	55.96%
20	22.39%	33.53%	52.83%	14.35%	39.22%	29.04%
50	58.84%	90.63%	91.80%	77.23%	80.07%	12.48%

We believe that our results indicate fruitful directions for managerial use. Our formulation will provide the flexibility to LSC to adjust the routes and select the best alternatives in a broader spectrum. The optimal solutions indicate that the shortest route is not the best route for many cases, hence it is relevant and necessary to consider all feasible route alternatives.

CHAPTER 6
SHIPMENT ASSIGNMENT AND VESSEL SCHEDULING PROBLEM WITH INSTANT PORT CONTAINER TERMINAL PERFORMANCES

In Chapter 6, we discuss a new solution approach to the SAVSP, where the instantaneous container port terminals are also taken into account. Section 6.1 includes the integrated solution framework of this idea, where the computational study of this framework is reported in Section 6.2.

6.1. The Integrated Solution Framework

In the SAVSP that was discussed in Chapter 4, it was assumed that the operation time at the ports are static, i.e., not changing over time. However, this assumption is unrealistic as port performances can significantly vary depending on the workload. Therefore, the static approach may yield over/or underestimated shipment assignments during the planning horizon. To incorporate this issue, we present in this section a solution framework that iteratively evaluates shipment assignments with instant port container terminal information, gathered by solving the Integrated Port Container Terminal Problem (IPCTP) proposed in Kizilay et al. (2018). Figure 6.1 illustrates the proposed solution framework.

The integrated solution framework starts by taking the information of the shipments such as the transit times, origin and destination ports. Subsequently, the feasible routes are generated as stated in Section 4.1. At the outset of the algorithm, the ports are assumed to have identical performances, i.e., the parameter $HT_p^{(v,n)}$, the handling time of the shipment in SAVSP is the same for every port n and it changes only with respect to shipment size.

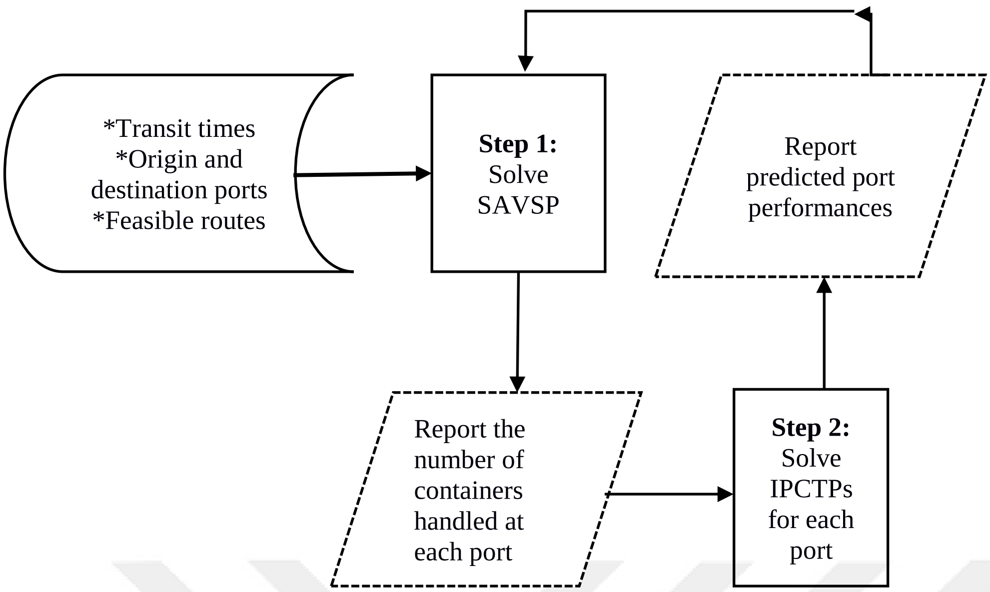


Figure 6.1. Integrated solution framework.

Next, the SAVSP is solved to optimality with this initial information. The optimal number of containers handled at each port is derived from the optimal solution of the SAVSP. Afterwards, the port container terminal performances are re-calculated by solving several IPCTPs simultaneously for each port. Each IPCTP takes the number of containers handled at each port as an input parameter and the optimal solution of each IPCTP reports the updated port performance for the corresponding port. Note that, IPCTP addresses the integrated problem of quay crane assignment, quay crane scheduling, yard location assignment, and vehicle dispatching operations at port container terminals proposing both mixed integer programming and constraint programming formulations to tackle this integrated problem (Kizilay et al., 2018). With the optimal solution of the IPCTP, the parameters of the SAVSP is updated and the model is re-solved with the new parameters.

The two problems are repeatedly solved until one of the terminating conditions is met. Explicitly, the algorithm stops once the difference of two consecutive optimal objective function values of the SAVSP is under 10%, or when 10 instances of SAVSPs are solved within the loop. The results are shown in Table 6.1.

Table 6.1. Detailed computation times of the integrated solution framework.

Test instance	# of feasible routes	Total Computation Time (min)	Number of iterations	Average Computation Time per iteration (min)
1_1_10_1	60	0.94	10	0.09
1_1_10_2	120	18.95	10	1.90
2_1_10_1	174	17.85	10	1.78
2_1_10_2	348	408.58	2	204.29
3_1_10_1	180	26.42	10	2.64
3_1_10_2	360	187.00	5	37.40
4_1_10_1	200	136.98	10	13.70
4_1_10_2	400	1555.45	4	388.86
1_2_20_1	102	9.36	10	0.94
1_2_20_2	204	1063.69	3	354.56
2_2_20_1	400	564.00	2	282.00
2_2_20_2	800	745.35	2	372.67
3_2_20_1	400	1442.00	4	360.50
3_2_20_2	800	704.59	2	352.30
4_2_20_1	388	804.94	2	402.47
4_2_20_2	776	1138.19	3	379.40
1_3_50_1	264	261.23	7	37.32
1_3_50_2	528	1935.36	3	645.12
2_3_50_1	982	720.58	2	360.29
2_3_50_2	1964	720.55	3	240.18
3_3_50_1	748	1035.79	3	345.26
3_3_50_2	1496	1353.99	3	451.33
4_3_50_1	986	1383.59	5	276.72
4_3_50_2	1972	1842.41	3	614.14

All test instances are solved to optimality using the integrated solution framework. The solutions of the IPCTP were previously reported in Kizilay et al. (2018) and used directly for the purposes of our computational experiment. Hence, on column 3 of Table 6.1, we report the total computation times for the SAVSP only. Column 4 indicates the total number of repeatedly solved SAVSPs within the solution framework. The average computation time per iteration are presented on the last column.

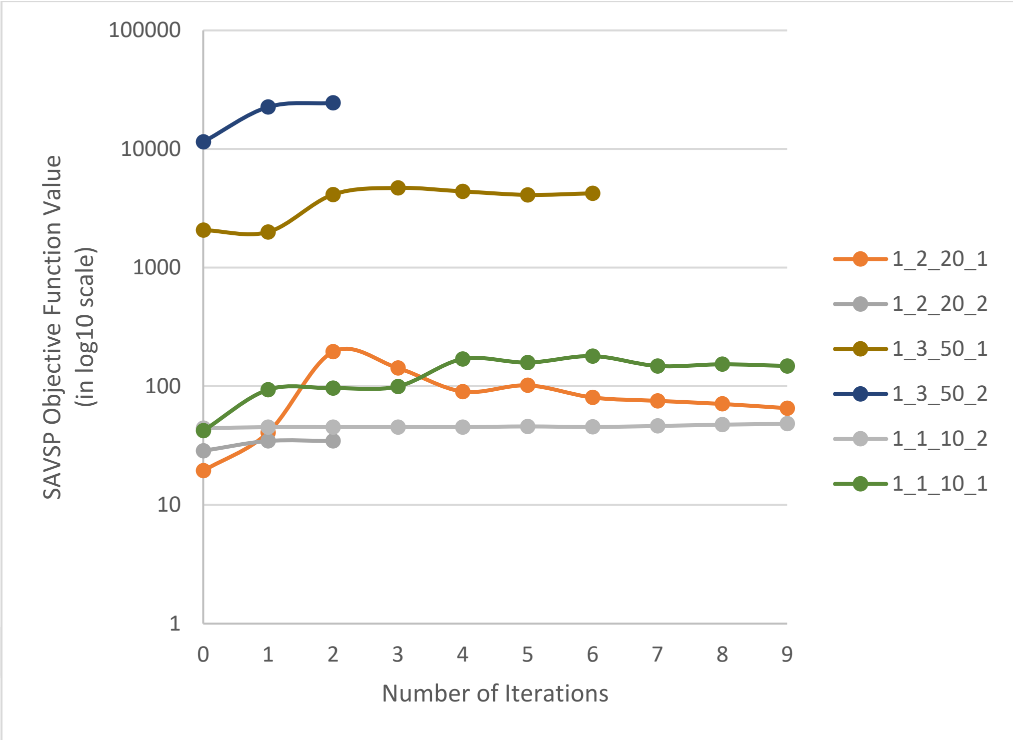
For small instances (1_1_10_1, 1_1_10_2, 2_1_10_1, 3_1_10_1, 4_1_10_1 and 1_2_20_1), the solution algorithm terminates under 27 minutes, performing the maximum number of iterations for each problem instance. Note that small instances are much more sensitive to port performance when compared to large instances. At the other extreme, for the 13 large test instances, although the average computation time per iteration is relatively large, the solution converges within 3 iterations.

We can conclude that, this fast convergence trait of our solution framework renders our solution framework as promising even for large instances. This solution effort is well-worth spending for the LSC in real life, as it helps the LSC to improve its operations with the consideration of more realistic port performance metrics.

We also illustrate how each problem class converge in Figures 6.2.(a)-(d), and how the computational time varies in Figures 6.3.(a)-(d). Note that, due to the skewness towards large values the logarithmic scale is used for the y-axis, as depicted in Figures 6.2.(a)-(d).

Figure 6.2.(a) reveals that, for Problem type 1, only half of the test instances, namely, 1_1_10_1, 1_1_10_2 and _1_2_20_1, converge at the end of 10 iterations. However, these test instances oscillate sharply till the 7th iteration, and start to converge afterwards. For these test instances, the changes in consecutive objective function values after the 7th iteration is slightly higher than 10%. Hence, we can conclude that terminating the algorithm even after the 7th iteration will yield high-quality solutions in shorter times. The same is true for test instance 4_1_10_1, as illustrated in Figure 6.2.(d).

Figures 6.3.(a)-(d) show the cumulative computation times for problem types 1, 2, 3, and 4, respectively. Note that, the cumulative computation time functions are concave for almost all of the test instances, as expected. The computational time at earlier iterations is usually larger, as the initial parameter set includes identical port performances, which yields numerous alternative solutions. Then, as the gap among alternative solutions increases through the differences of the port performances at each iteration, the computation times gradually decrease in the subsequent iterations.



,

Figure 6.2 (a) Convergence graph for the problem type 1.

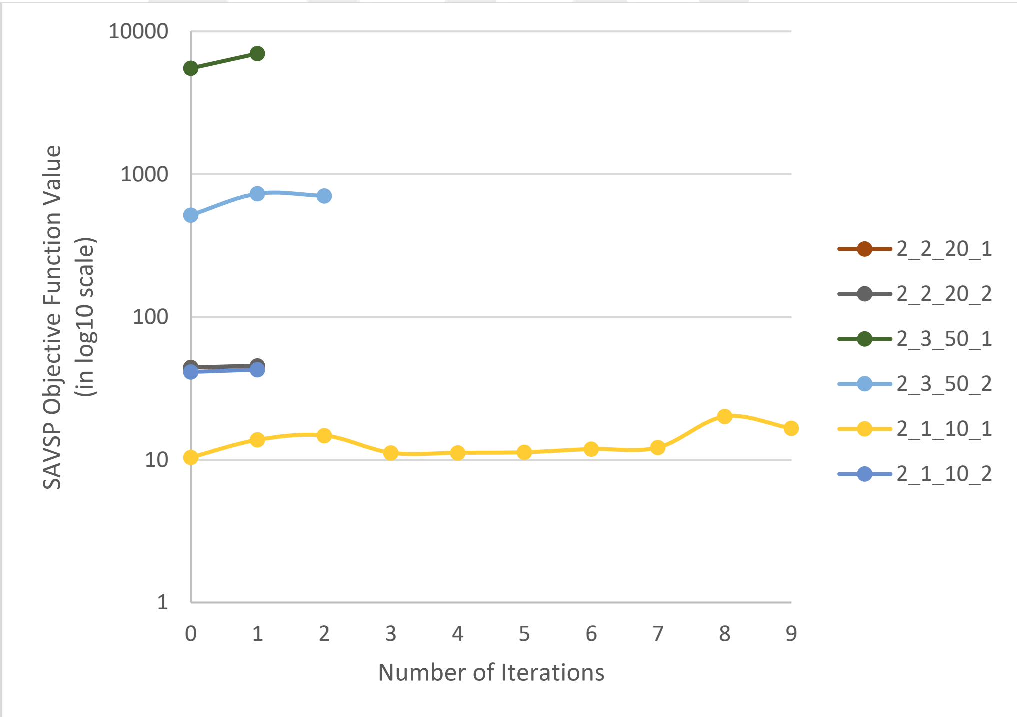


Figure 6.2 (b) Convergence graph for the problem type 2.

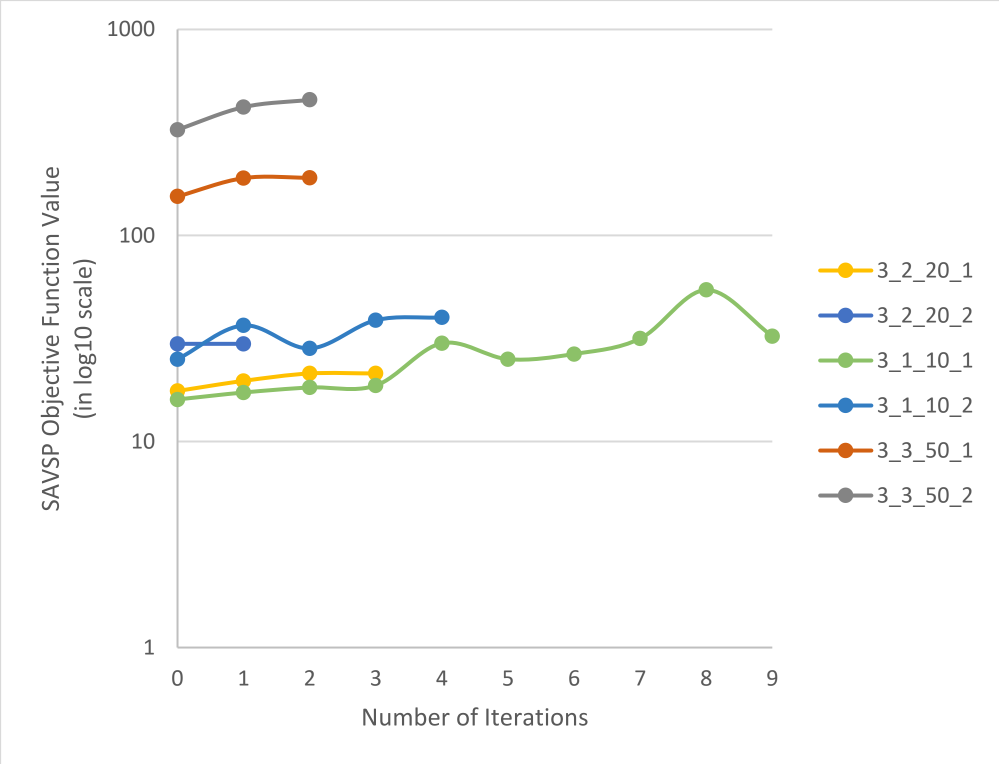


Figure 6.2 (c) Convergence graph for the problem type 3.

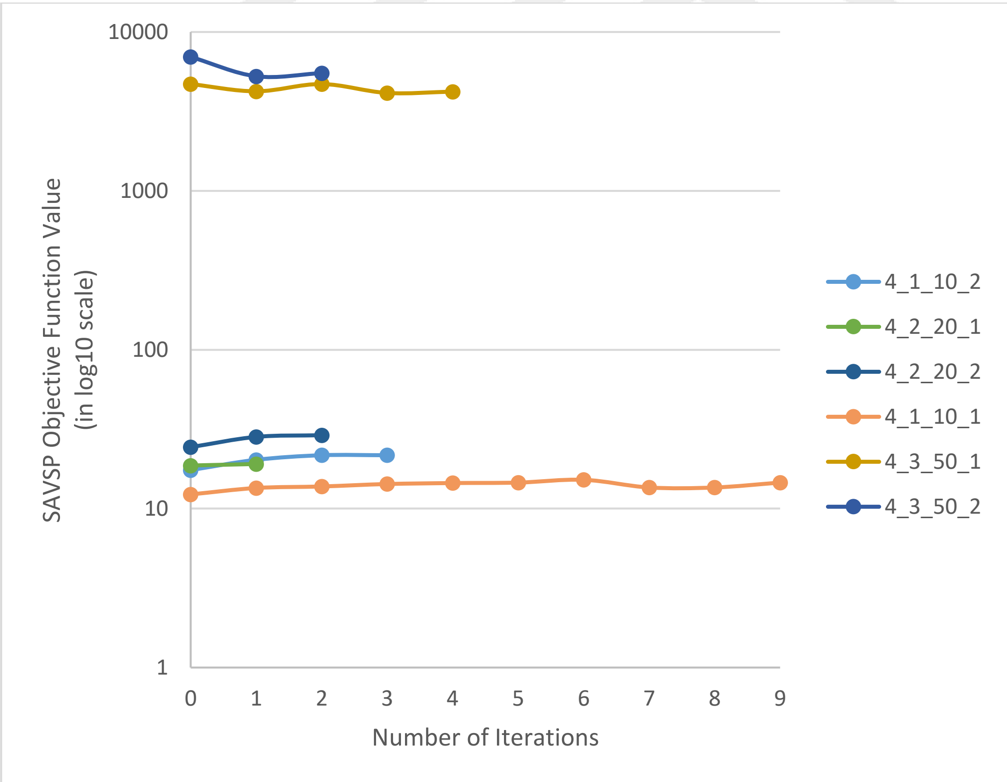


Figure 6.2 (d) Convergence graph for the problem type 4.

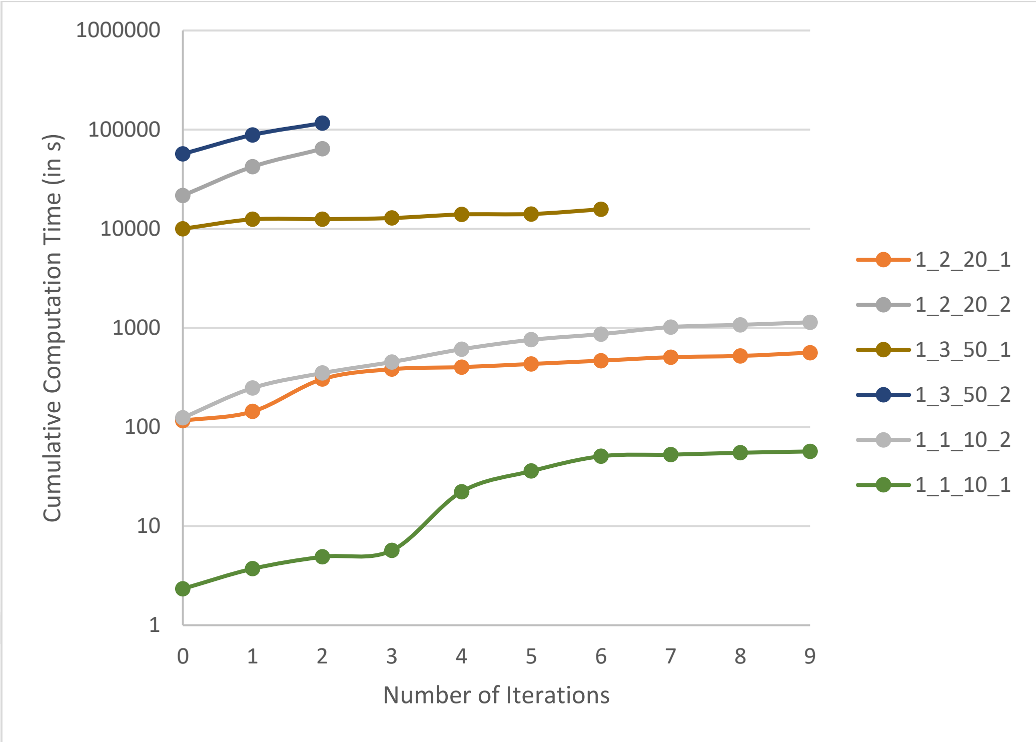


Figure 6.3 (a) Cumulative computation times for the problem type 1.

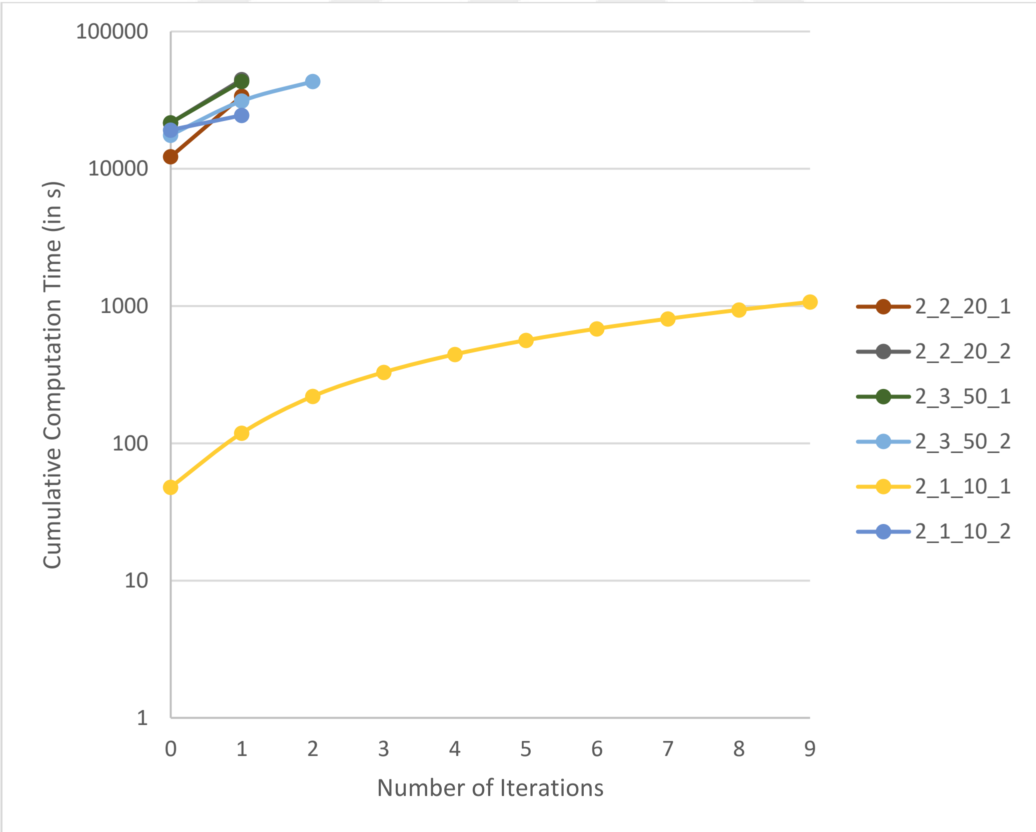


Figure 6.3 (b) Cumulative computation times for the problem type 2.

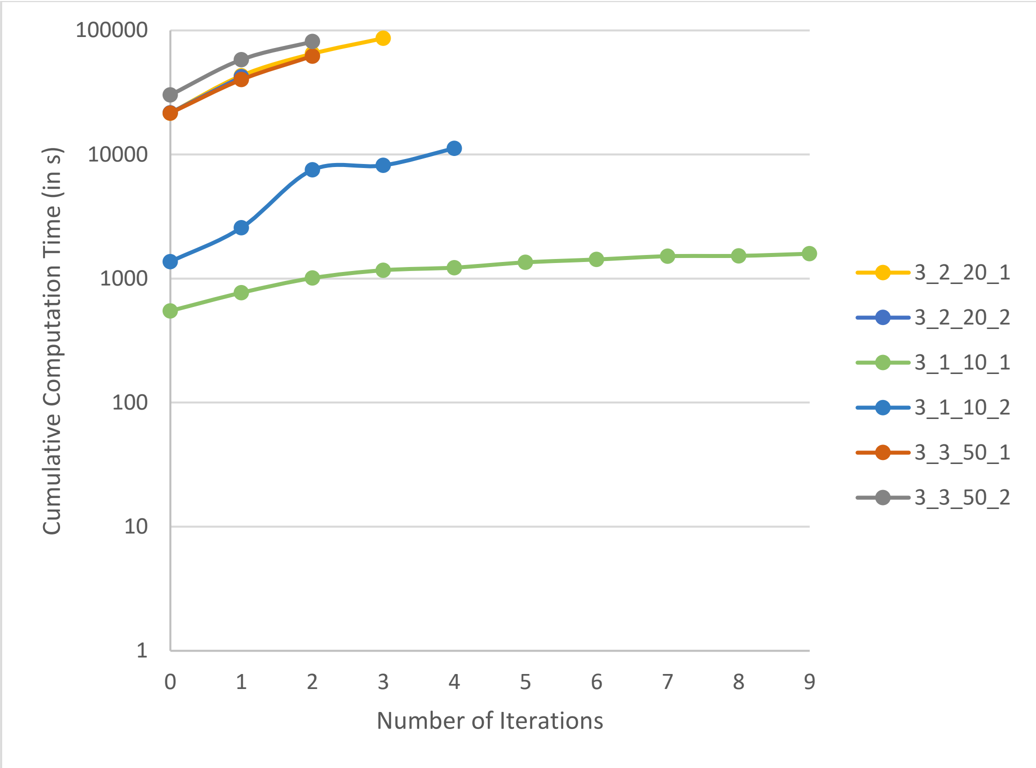


Figure 6.3 (c) Cumulative computation times for the problem type 3.

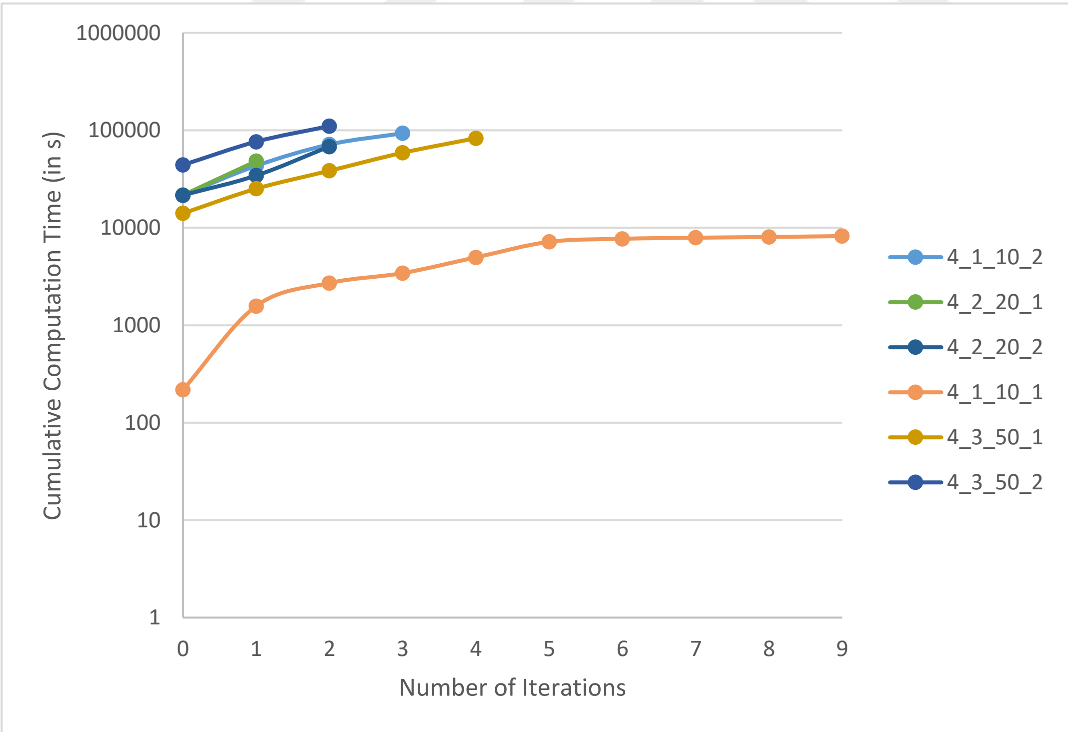


Figure 6.3 (d) Cumulative computation times for the problem type 4.

CHAPTER 7
CONCLUSIONS AND FUTURE RESEARCH

In this thesis, the shipment assignment and vessel scheduling problem motivated by a liner shipping agency in Izmir, Turkey is considered. The problem is formulated as a novel mixed integer linear programming model and solved by a two-stage algorithm. The algorithm generates all feasible routes for each shipment in the first stage, while in the second stage the proposed model is solved optimally by taking the routes generated as input. Valid inequalities are proposed to restrict the feasible region and increase the computational time performance. We also implemented a Benders decomposition approach for the problem.

The performances of all algorithms are tested on 24 problem instances, which are generated in line with the real practice based on past data of the liner shipping agency. The computational results indicate that the Benders decomposition algorithm including the proposed valid inequalities yields the best computational time performance, solving 21 of the test instances to optimality within 2.5 hours. The algorithm also achieves optimality in less than 100 seconds on 15 of the test instances, promising fast optimal solutions for real instances of the problem.

The thesis has two major practical contributions. Our MILP model allows the agency to determine which shipment will be sent through which route. Concurrently, our formulation determines the arrival and departure times of the vessels while maintaining a desired service frequency. We believe that our approach can handle the stochasticity of the port stays and sailing times by adjusting the arrival and departure times of the vessels and may provide an insight to the practitioners to redesign their routes for every O-D pair, as the solution suggests alternative routes for every shipment.

In addition to the shipment assignment and vessel scheduling problem, we present an integrated solution approach which considers the instant port performances and concurrently seeks for the best assignment of the shipments to the routes. Our solution algorithm iteratively solves two separate problems, the shipment assignment and vessel scheduling problem provided in Section 4.1 and the integrated port container terminal problem by Kizilay et al. (2018). The optimal solution of the first problem determines the updated workloads at each port, leading in the re-optimization of the integrated port operations according to the new workloads. The proposed solution framework is tested for the 24 instances, and the results indicate that small test problems converge around 10 iterations whereas the large cases converge quickly, i.e., within 2 or 3 iterations. As a first step, we suggest embedding all optimal solutions of SAVSP found so far to the LSC's database for further use. We believe that the LSC can use this approach in practice and this will prevent the LSC solving the same problems repeatedly.

An interesting extension of our current work is to enhance the proposed formulation by including speed optimization as well as bunker consumption decisions. Liner shipping companies are dealing with high fuel prices as well as concerns related with the greenhouse gas emissions and carbon footprint. The sailing speed of a vessel has a remarkable effect on the bunker consumption and the bunker cost accounts for a large proportion of the total operation cost of the vessel (Ronen, 1993). The bunker cost is estimated to be more than 60% of the total operating cost of a liner shipping company (Golias et al., 2009). Hence, slow steaming is preferred as it yields a reduction in the bunker consumption, i.e., paying less on bunker cost. On the other extreme, slow steaming may lead an increase in the delivery times, resulting in unattractive service times for the customers (Rheinhardt et al., 2016). As a future study, we plan to investigate the problem of reducing the bunker consumption as well as maintaining the service quality, with the consideration of vessel capacities and utilizations.

In our current work, stated in Section 4.1, we assume constant sea durations at each leg. However, this assumption is not realistic as the sailing times are highly sensitive to whether conditions, for example, and this may yield miscalculations in vessel schedules and transit times. We plan to incorporate leg-by-leg speed optimization decisions into our current SAVSP, explained in Section 4.1. To do so, we will first start working on how the cost function will be estimated. Afterwards, we plan to express the objective function in equation (1) in Section 4.1, in terms of cost. Also, total operating costs of port terminals, including handling and transshipment costs, will also be included into the objective function. We believe that this approach will provide realistic calculations for the liner shipping company with the consideration of uncertainty in vessel schedules and port stay durations.

A natural extension of the integrated solution approach is to examine the performance of the algorithm as well as the solution qualities by changing the terminating conditions. We plan to further monitor the cumulative computation time performances for convergence, by restricting the difference of the two consecutive optimal solution values of the shipment assignment and vessel scheduling problem is less than 5% and 15%.

An alternative approach to the integrated solution framework can be to reformulate the problem through stochastic programming; where “act of god”, i.e., uncertainties such as whether conditions, congestions at ports or port strikes, which slash the container traffic; can be treated explicitly. To incorporate such uncertainties, we plan to address a the two-stage (planning and recourse) stochastic programming shipment assignment and vessel scheduling model. In the first stage, the model is solved according to the realized port performances. The solution is then revised in the later stage. For example, in the second stage, the liner shipping company may reassign the shipments at intermediate ports in the route, when a delay is observed in the first stage. At the recourse level, the model will consider the shipment assignments made in the first stage, its realizations and the instant port performances at the adjustment point. The adjustment points will be the candidate transshipment ports that are going to be visited by the vessels through their upcoming port visits.

Another future research topic can be developing sophisticated heuristic algorithms to provide high quality solutions faster than the discrete optimization techniques applied in this thesis study. The iterative solution scheme discussed in Chapter 6 resulted in high computation times for large test problems. Note that, as discussed in Chapter 5, the sole SAVSP yields low computation time performance in some test instances. The IPCTP even cannot find the optimal solution for several instances (Kizilay et al., 2018). Due to the high problem complexity of both SAVSP and IPCTP, a realistic approach for practical use is to utilize heuristic approaches, providing near-optimal solutions in short computation time. Hence, as a first step, we plan to employ an assortment of local search heuristics, iterated greedy, variable neighborhood search, and/or adaptive large neighborhood search, for both problems. If these heuristics do not provide efficient solutions, we aim to employ population-based heuristic algorithms such as genetic algorithm or differential evolution and further enrich them by embedding the local search heuristics.

We believe that the contribution of this thesis study is twofold. Our first contribution to the practitioners becomes apparent when the liner shipping company's main objective is to maximize customer satisfaction by providing on-time delivery and to maintain schedule reliability. The liner shipping company can treat each shipment independently by selecting the best route, to do so, the liner shipping company further investigate all possible alternative routes, including the unforeseen ones, for each shipment. This thesis study also fills a significant gap in the maritime shipping literature. To the best of our knowledge, current studies do not consider the operational shipment assignment and the vessel scheduling problem. In addition, there is no existing study focusing both the integrated maritime and landside operations concurrently. Therefore, there is vast opportunity for extending the developed mathematical model and the solution methodology with the consideration of other maritime planning problems, such as berthing and stowage planning.

REFERENCES

Agarwal, R., & Ergun, Ö. (2008). Ship scheduling and network design for cargo routing in liner shipping. *Transportation Science*, 42(2), 175-196.

Álvarez, J. F. (2009). Joint routing and deployment of a fleet of container vessels. *Maritime Economics & Logistics*, 11(2), 186-208.

ArkaslineCompany website (2018). Retrieved 13 November 2018, from http://www.arkasline.com.tr/hatlar_ve_programlar.html

Bell, M. G., Liu, X., Angeloudis, P., Fonzone, A., & Hosseinloo, S. H. (2011). A frequency-based maritime container assignment model. *Transportation Research Part B: Methodological*, 45(8), 1152-1161.

Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. In *Numerische mathematik*, 4(1), 238-252.

Binato, S., Pereira, M. V. F., & Granville, S. (2001). A new Benders decomposition approach to solve power transmission network design problems. *IEEE Transactions on Power Systems*, 16(2), 235-240.

Brouer, B. D., Pisinger, D., & Spoorendonk, S. (2011). Liner shipping cargo allocation with repositioning of empty containers. *INFOR: Information Systems and Operational Research*, 49(2), 109-124.

Brouer, B. D., Dirksen, J., Pisinger, D., Plum, C. E., & Vaaben, B. (2013). The Vessel Schedule Recovery Problem (VSRP)–A MIP model for handling disruptions in liner shipping. *European Journal of Operational Research*, 224(2), 362-374.

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

Cordeau, J. F., Stojković, G., Soumis, F., & Desrosiers, J. (2001). Benders decomposition for simultaneous aircraft routing and crew scheduling. *Transportation Science*, 35(4), 375-388.

Gelareh, S., Nickel, S., & Pisinger, D. (2010). Liner shipping hub network design in a competitive environment. *Transportation Research Part E: Logistics and Transportation Review*, 46(6), 991-1004.

Gelareh, S., & Pisinger, D. (2011). Fleet deployment, network design and hub location of liner shipping companies. *Transportation Research Part E: Logistics and Transportation Review*, 47(6), 947-964.

Golias, M. M., Saharidis, G. K., Boile, M., Theofanis, S., & Ierapetritou, M. G. (2009). The berth allocation problem: Optimizing vessel arrival time. *Maritime Economics & Logistics*, 11(4), 358-377.

Guericke, S., & Suhl, L. (2013). Liner network design under consideration of transit times and partner networks. In *International Conference on Logistics and Maritime Systems*.

Guericke, S., & Tierney, K. (2015). Liner shipping cargo allocation with service levels and speed optimization. *Transportation Research Part E: Logistics and Transportation Review*, 84, 40-60.

Karsten, C. V., Pisinger, D., Ropke, S., & Brouer, B. D. (2015). The time constrained multi-commodity network flow problem and its application to liner shipping network design. *Transportation Research Part E: Logistics and Transportation Review*, 76, 122-138.

Kizilay, D., Eliiyi, D. T., & Van Hentenryck, P. (2018). Constraint and mathematical programming models for integrated port container terminal operations. In *International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research* (344-360). Springer, Cham.

Meng, Q., & Wang, S. (2012). Liner ship fleet deployment with week-dependent container shipment demand. *European Journal of Operational Research*, 222(2), 241-252.

Mulder, J., & Dekker, R. (2014). Methods for strategic liner shipping network design. *European Journal of Operational Research*, 235(2), 367-377.

Notteboom, T. E. (2006). The time factor in liner shipping services. *Maritime Economics & Logistics*, 8(1), 19-39.

Ozcan, S., Eliiyi, D. T., (2016). Cargo Allocation and Vessel Scheduling Problem with Synchronization of Transshipments. In *Conference on Operations Research – Annual Conference of the German Operations Research Society (GOR), Book of Abstracts* (78). Springer.

Öztürk, C. Sargut, F.Z., Örnek, M.,A., Eliiyi D. T., (2017). Optimisation and heuristic approaches for assigning inbound containers to outbound carriers. *Maritime Policy & Management*, 44(7), 825-836.

Plum, C. E., Pisinger, D., Salazar-González, J. J., & Sigurd, M. M. (2014). Single liner shipping service design. *Computers & Operations Research*, 45, 1-6.

Rahmaniani, R., Crainic, T. G., Gendreau, M., & Rei, W. (2017). The Benders decomposition algorithm: A literature review. *European Journal of Operational Research,* 259(3), 801-817.

Reinhardt, L. B., & Pisinger, D. (2012). A branch and cut algorithm for the container shipping network design problem. *Flexible Services and Manufacturing Journal*, 24(3), 349-374.

Reinhardt, L. B., Plum, C. E., Pisinger, D., Sigurd, M. M., & Vial, G. T. (2016). The liner shipping berth scheduling problem with transit times. *Transportation Research Part E: Logistics and Transportation Review*, 86, 116-128.

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

Rudin, P. (2016). OR in an OB World. Retrieved 13 November 2018, from https://orinanobworld.blogspot.com.tr/2016/12/support-for-benders-decomposition-in.html

Song, D. P., & Dong, J. X. (2012). Cargo routing and empty container repositioning in multiple shipping service routes. *Transportation Research Part B: Methodological*, 46(10), 1556-1575.

Song, D. P., & Dong, J. X. (2013). Long-haul liner service route design with ship deployment and empty container repositioning. *Transportation Research Part B: Methodological*, 55, 188-211.

Tran, N. K., Haasis, H. D., & Buer, T. (2017). Container shipping route design incorporating the costs of shipping, inland/feeder transport, inventory and CO2 emission. *Maritime Economics & Logistics*, 19(4), 667-694.

Wang, S., & Meng, Q. (2011). Schedule design and container routing in liner shipping. *Transportation Research Record: Journal of the Transportation Research Board*, (2222), 25-33.

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

Wang, S., & Meng, Q. (2014). Liner shipping network design with deadlines. *Computers & Operations Research*, 41, 140-149.

Wang, S., Meng, Q., & Liu, Z. (2013). Containership scheduling with transit-time-sensitive container shipment demand. *Transportation Research Part B: Methodological*, 54, 68-83.

Wang, H., Wang, S., & Meng, Q. (2014). Simultaneous optimization of schedule coordination and cargo allocation for liner container shipping networks. *Transportation Research Part E: Logistics and Transportation Review*, 70, 261-273.

Wang, H., Zhang, X., & Wang, S. (2016a). A joint optimization model for liner container cargo assignment problem using state-augmented shipping network framework. *Transportation Research Part C: Emerging Technologies*, 68, 425-446.

Wang, S., Meng, Q., & Lee, C. Y. (2016b). Liner container assignment model with transit-time-sensitive container shipment demand and its applications. *Transportation Research Part B: Methodological*, 90, 135-155.

UNCTAD Review of Maritime Transport (2018). Retrieved 13 November 2018, from https://unctad.org/en/PublicationsLibrary/rmt2018_en.pdf