Inventory Constrained Maritime Routing and Scheduling for Multi-Commodity Liquid Bulk

A Thesis
Presented to
The Academic Faculty

by

Seung-June Hwang

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Industrial and Systems Engineering
Georgia Institute of Technology
May 2005

Inventory Constrained Maritime Routing and Scheduling for Multi-Commodity Liquid Bulk

Approved by:

Dr. Faiz Al-Khayyal, Advisor
School of Industrial & Systems Engineering
Georgia Institute of Technology

Dr. Chelsea (Chip) C. White III
School of Industrial & Systems Engineering
Georgia Institute of Technology

Dr. Earl Barnes
School of Industrial & Systems Engineering
Georgia Institute of Technology

Dr. I. A. Karimi
Department of Chemical & Biomolecular Engineering
University of Singapore

Dr. Shabbir Ahmed
School of Industrial & Systems Engineering
Georgia Institute of Technology

Date Approved: April 14, 2005

PREFACE

This thesis is for the dissertation of my Ph.D. study at the Department of Industrial and Systems Engineering at Georgia Institute of Technology. It serves as documentation of my work during the study, which has been made from fall 2000 until spring 2005. The study has been supported in part by The Logistics Institute – Asia Pacific, a collaboration between the Georgia Institute of Technology and the National University of Singapore.

This thesis consists of three chapters with conclusions and appendices. Each chapter contains the paper that is submitted to or intended for an international journal or proceedings. Therefore, each chapter is self explanatory without the cross references from other chapters.

The first chapter gives an overview of the maritime shipping industry and provides a focused survey on the problems of ship routing and scheduling of bulk materials. The second chapter focuses on the formulation of a model for finding a minimum cost routing in a network for a heterogeneous fleet of ships engaged in pickup and delivery of several liquid bulk cargos subject to the inventory level of each product in each port being maintained between certain levels. The third chapter shows this combined multi-ship pickup-delivery problem can be decomposed into several subproblems by dualizing coupling constraints and suggests a solution method using Lagrangian relaxation and a randomized greedy heuristic approach. Appendices consist of theoretical results regarding linear relaxation techniques, a glossary of notation, etc.

ACKNOWLEDGEMENTS

I would like to thank my advisor Faiz Al-Khayyal for all of his help and understanding during my entire graduate school experience. I would like to thank Chelsea (Chip) C. White III, Earl Barnes, Shabbir Ahmed and I.A. Karimi, for serving on my thesis committee. I would like to thank all my friends who I met at Georgia Tech. Finally and most importantly, I am grateful to my wife Sinwook for her continual support and assistance. In addition, I appreciate the support and sacrifice of my whole family from my mom and dad to my son Alex.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Vehicle routing and scheduling has been studied extensively in the context of truck transport and, to a lesser extent, in the context of rail transport. However, relatively little work has been done on ship routing and scheduling even though approximately 90% of the volume and 70% of the value of all goods transported worldwide is carried by sea.

In the first chapter, we survey the literature on maritime transport of liquid bulk products with an eye on challenges that lend themselves to solution by operations research methodology. The survey is by no means exhaustive and is intended as an introduction to the subject for researchers new to the field. Only brief synopses of the articles are used to capture the flavor of the type of problems that arise and the models and solution strategies that have been proposed to deal with them. While the paper focuses on routing and scheduling problems, other important problems are also identified.

In the second chapter, we formulate a model for finding a minimum cost routing in a network for a heterogeneous fleet of ships engaged in pickup and delivery of several liquid bulk cargos. The problem is frequently encountered by maritime chemical transport companies, including oil companies serving an archipelago of islands. The products are assumed to require dedicated compartments in the ship. The problem is to decide how much of each product should be carried by each ship from supply ports to demand ports, subject to the inventory level of each product in each port being maintained between certain levels that are set by the production rates, the consumption rates, and the storage capacities of the various products in each port. This important and challenging inventory constrained multi-ship pickup-delivery problem is formulated as a mixed-integer nonlinear program. We show that the model can be reformulated as an equivalent mixed-integer linear program with special structure. Over 100 test problems are randomly generated and solved using CPLEX 7.5. The results of our numerical experiments illuminate where problem structure can be exploited in order to solve larger instances of the model.

The third chapter of this thesis deals with solution algorithms that take advantage of model properties. We show that the mixed-integer linear program can be decomposed into several subproblems by dualizing coupling constraints. We solved this minimization problem by the Lagrangian Relaxation method to get a better lower bound and to measure the quality of solutions obtained from two suggested randomized greedy heuristic methods.

We conducted numerical studies to establish the goodness of our combined Lagrangian Relaxation/Heuristic approach. Test results show an average duality gap of 26.8% and an average optimality gap of 12.5% on small sized problems. More importantly, our solution times are, on average, three orders of magnitude faster than getting a first feasible solution by CPLEX when using the default options of the solver.

CHAPTER I

MARITIME TRANSPORT OF BULK MATERIALS: AN OPERATIONS RESEARCH PERSPECTIVE

1.1 Introduction

Many industries face the daily challenge of determining the flows in supply chain networks in order to meet customer demand at minimum cost. Such problems have been studied extensively for road and rail networks and for the intermodal hubs that connect them. These studies have provided solutions that have become the logistical frameworks for the advancement of intercontinental trade. However, globalization of trade has placed a heavy burden on maritime shipping. In the face of this growth, many of the world's major ports have installed modern equipment to load, unload and store the ever increasing volumes of goods passing through them. To secure even more efficiency for modern ports (and simultaneously decrease the expenses of shipping companies), optimal routing and scheduling of maritime fleets is needed. This chapter focuses on the state of the art of modeling and on solving problems related to the maritime transport of liquid bulk materials. These problems are much less understood than container transport problems.

We have concentrated on the shipment of liquid bulk materials because container shipping is more closely related to vehicle routing in terms of both the types of challenges that arise and the mathematical methodologies that are employed to model and solve them. Moreover, bulk transportation constitutes more than 80% of both global waterborne trade and fleet size compared with 10% for container shipping [55]. Also, bulk transportation accounts for up to 80% of the total ton-miles by water, according to Ballou [3]. By comparison, the freight moving in container ships is far less in tonnage, but it is much more efficiently processed. This is because the use of containers reduces handling time, allows intermodal transfer, and reduces loss and damage to the goods.

Transportation is a significant fraction of the economies of most developed nations. It accounts for approximately 15% of the U.S. gross national product. Worldwide, approximately 90% of the volume and 70% of the value of all goods is transported by sea (see Transportation Science [39]). Given the (relatively) long delivery times associated with marine transport, the impact of inefficiencies in maritime shipping can easily be magnified throughout the supply chain. As globalization of multinational enterprises increases, the demand for maritime transport capacity could eventually outstrip supply, a situation that would lead to increased prices for finished goods. It would take several years for additional capacity to be built, and by then the marketplace could change again. The economic significance of such potential problems has motivated both private companies and academic researchers to pursue the use of modern decision support systems to arrive at better utilization of existing resources and to make better planning decisions. A recent issue of the Journal *Transportation Science* [39] on Maritime Transportation focused on this problem area and published only four papers. These papers represent only a small sampling of the many challenges confronting companies operating at different locations in the supply chain.

In this chapter, we give an overview of the maritime shipping industry and provide a focused survey of the problems of ship routing and scheduling of bulk materials. In particular, we are interested in issues and models that have been successfully tackled via Operations Research (OR) methodology. In recent years, OR has had much success in solving a growing array of complex decision problems confronting managers of large organizations that require the efficient use of materials, equipment, and human resources. In the areas of logistics and supply chains, OR analysts determine the optimal means of coordinating diverse elements of an enterprise in order to achieve specified goals by applying mathematical principles to organizational problems.

One of the most successful applications of Operations Research has been in vehicle routing. This problem calls for determining the most efficient use (either in the sense of cost minimization or profit maximization) of a fleet of vehicles that must make a number of stops to pick up and/or deliver passengers or products. Most of the major trucking companies in the United States currently have implemented OR techniques to manage their fleet assignment and vehicle routing. The *container* maritime transportation industry has benefited from this early work because container ports are hubs of the intermodal networks that transfer containers from sea to land. Consequently, many container shipping problems have been looked at to some degree because of their similarity to rail-truck intermodal networks on land. Lagging far behind are issues relating to *bulk* (liquid and dry) maritime shipping.

Maritime transportation of bulk materials is of increasing importance to the island nations of Pacific Asia because of their growing interdependence as a result of globalization and the limited transportation capacities available by truck, rail and pipeline. It is safe to say that the rapid economic growth of Pacific Asia nations can be sustained only if logistics systems for bulk cargo keep pace with increasing demand. Table 1 from the reference [55] shows that 12 (*italicized*) of the top 25 ports in the world are in the Pacific Asia region, but they receive more than 60% of worldwide port calls. This significance may increase as the region's economy expands. According to [54], more than half of the world's supertankers pass through the South East China sea from the Middle East to countries with large energy appetites such as Japan, South Korea, and China. Additionally, many major oil companies have refining centers in this region, most notably in Singapore. To fuel anticipated economic growth, significant increases in maritime transported liquid bulk petrochemical products must be accommodated within the existing transportation networks. This fuels the need to apply operations research methodology to the situation in order to find ways to more efficiently utilize existing systems and to make better strategic decisions on capacity expansions.

With annual growth in maritime shipping being measured in the billions of U.S. dollars, maritime transportation companies can expect large gains from improving the routing and scheduling of their ships. According to Chajakis [13], a 7% reduction in the costs of logistics in the refinery industry increases annual profits by 23%. Moreover, such a reduction can be attained easily through intelligent scheduling and the use of modeling tools without recourse to large amounts of capital investment. However, the OR literature shows that relatively few research and implementation studies have been done on maritime industries in comparison with the number done on the other transportation modes of air, rail and

Table 1: Top 25 World Ports(italicized Asian Ports) by Calls, 1997

Ports	Calls	Percentile	Ports	Calls	Percentile
1 *Singapore*	45,816	16.14%	15 Felixtowe	7,266	2.56%
2 *Hong Kong*	31,352	11.05%	16 Priraeus	7,023	2.47%
3 Rotterdam	15,852	5.59%	17 Houston	6,803	2.40%
4 Antwerp	14,265	5.03%	18 New Orleans	6,762	2.38%
5 *Kaohsiung*	13,402	4.72%	19 Barcelona	6,649	2.34%
6 *Yokohama*	13,043	4.60%	20 London	6,649	2.34%
7 *Busan*	11,958	4.21%	21 *Shanghai*	6,376	2.25%
8 Hamburg	11,704	4.12%	22 Le Havre	5,960	2.10%
9 *Nagoya*	10,274	3.62%	23 *Tokyo*	5,937	2.09%
10 Europort	10,048	3.54%	24 Genova	5,612	1.98%
11 *Kobe*	9,772	3.44%	25 Los Angeles	5,585	1.97%
12 *Port Kelang*	9,683	3.41%	Total	283,627	
13 *Jakarta*	8,351	2.94%	All ports	1,298,757	
14 *Osaka*	7,658	2.70%	Top 25 (percent)		22.1 %

motor vehicles. To keep the cost of consumer products low, it is essential that maritime transportation companies operate efficiently by determining routes and schedules that minimize total distribution costs while satisfying various requirements such as ship capacity, time windows on pick-up and/or delivery, timely availability of ships, etc.

The purpose of this chapter is to help the maritime transportation industry better understand the potential impact of applying OR techniques to its business. Also, we hope to stimulate increased academic research by surveying the open literature on maritime transportation, classifying the models that have been developed, and summarizing proposed solution techniques.

The remainder of this chapter is organized as follows. In section 1.2, we review the important items that should be considered when operations research methods are used to model ship routing and scheduling. In section 1.3, we present the typical types of ship routing and/or scheduling problems that arise in the bulk shipping industries.

1.2 The Maritime Transportation Industry

It is helpful to consider the principal components of maritime transportation as listed in the Table 2 in order to understand the routing and scheduling problem and to help formulate models for finding the efficient use of a fleet. The principal components are modified from Fisher and Rosenwein [27] by considering additional parameters such as, but not limited to heterogeneous types of ships, multicompartment ships, and the storage capacities of ports. As can be seen, the components listed are merely the major problem parameters and constraints that need to be considered. The presence or absence of any component in a model is determined by a specific situation that may dictate the need for additional constraints not listed in Table 2, such as minimum load and unload amounts or the need, especially for chemical products, for setup times to load and unload different cargoes. It is safe to say that most real ship routing problems need to consider most of the components, thus making the problem very complex.

Cargo can be categorized from an OR point of view as either discrete or continuous. Discrete cargoes are itemized by containers and transported by container ships. Each cargo is specified by load port, delivery port, time window, etc. Continuous cargoes are referred to as *bulk*. Typically they are divided into *dry bulk* and *liquid bulk*. Most (but not all) of the (petro)chemical products are categorized as liquid bulk and are carried in special vessels whose specifications are determined by the characteristics of the chemicals involved. For example, Liquified Natural Gas (LNG) carriers should be designed to maintain high pressure. For some chemicals, storage-vessels should be designed to guard against chemical reactions with the tank. Therefore, liquid bulk is usually carried by dedicated ships or in designated compartments of multicompartment ships. Dry bulk commonly consists of *break bulk* materials such as steel, logs, lumber, wood pulp, paper, etc., and the term general dry bulk covers such cargoes as grain, coal, fertilizer, coke, sulfur, etc. While dry bulk deals with discrete objects, shipping decisions are based on weight, volume, area, etc.; i.e., continuous measures.

Ports usually offer facilities for storage of specified cargoes, railroad switching services, materials handling equipment such as pipelines, heavy lift cargo handlers, and other general

Table 2: Principal Components of Maritime Transportation.

Major categories	Components
Cargoes	types of cargo
	quantity of each
	load ports for each type cargo
	delivery ports for each type cargo
	time-window constraints on load and delivery times
Ports	number of ports
	navigable water depth
	distance between ports
	loading/discharging duration for each cargo type
	storage capacity for each cargo type
	facilities needs to load or unload for specific cargo
Ships	capacity
	compartments
	number of ships
	types (heterogeneous or homogeneous)
	limitation on ports or canals
	maximum speed
	location at the start of scheduling horizon
	time of availability
Costs	spot charter rates
	port and canal dues
	idle ship and demurrage charges
	operating costs for ships in fleet
	- crew
	- bunker fuel
	- flushing between loads
	- maintenance and repair
	- port charges

services such as bunker fuel. Because different ports offer different levels of such infrastructure, the business requirements of maritime companies frequently predetermine the ports that should be visited in a routing problem. The problem remains of how to best — with "best" usually equating to least cost — sequence the visits within permitted time windows while satisfying demand.

Several types of water carriers are used. In general, they can be classified as either *barges*, or *bulk carriers* (liquid or dry), or *container ships*, or special purpose carriers such as car carriers, Liquified Natural Gas (*LNG*) *carriers* and *passenger ships*. Barges have standardized dimensions of either 26 by 175 feet or 35 by 195 feet and tow up to 40,000 tons (See Ballou [3]). Bulk carriers are of two types, one for transportation of bulk liquids such as crude oil, chemicals and petrochemicals, and the other for transporting dry bulk such as coal, sand, grain, etc. Depending on the purpose of the ship, it can have several compartments, including dedicated compartments for bulk liquids, and/or movable compartment partitions for dry bulk carriers that add the flexibilty of changing compartment sizes to accommodate varying cargo dimensions. A typical (average-sized) bulk liquid carrier for crude oil has a 200,000m^3 cargo hold capacity, while a dry bulk carrier's cargo hold is about 100,000m^3 (See [51]).

Several types of vessels are shown below with graphics from the reference [51]. General bulk carriers (see Figure 1) can transport several types of cargo at the same time, without restrictions on specific cargoes. Ore, bulk and oil (OBO) carriers (see Figure 2) are dedicated carriers called OBO's. Some of them are designed for carrying only ore and these are called ore carriers (see Figure 3). General cargo ships (see Figure 4) carry a variety of packaged freight of any kind, heavy or light, liquid or solid. Most of these are equipped with stevedoring lifts. Oil tankers (see Figure 5) are designed for carrying crude oil for relatively long distances. Typically, these are very large and cannot navigate canals and certain waterways. Chemical tank carriers are related ships designed for transporting (petro)chemical products. *Liquified Petroleum Gas* (LPG) and Liquified Natural Gas (LNG) carriers (see Figure 6) have compartments that maintain high pressure and low temperature in order to transport large volumes of the products in stable states. Container carriers (see Figure 7) haul containers inside the hull and atop of the deck.

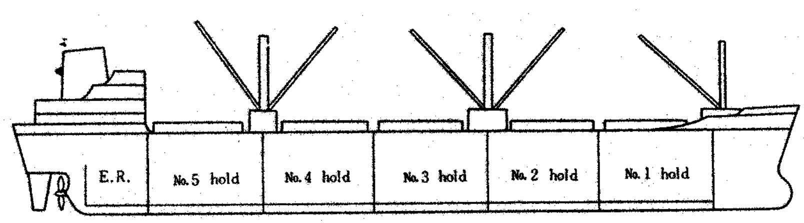

Figure 1: General Bulk carrier

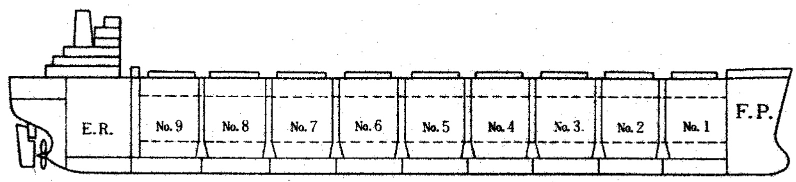

Figure 2: Ore, bulk and oil (OBO) carrier

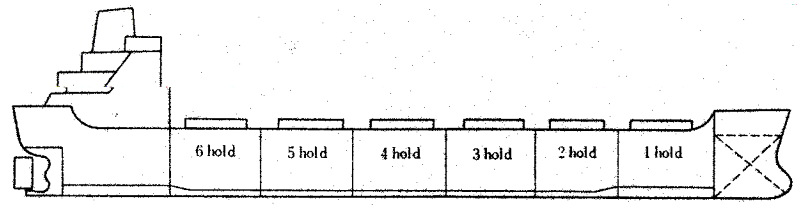

Figure 3: Ore carrier

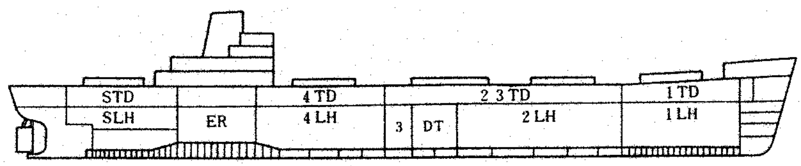

Figure 4: General cargo ship

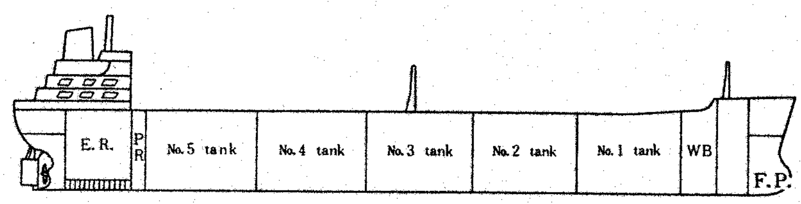

Figure 5: Oil tanker

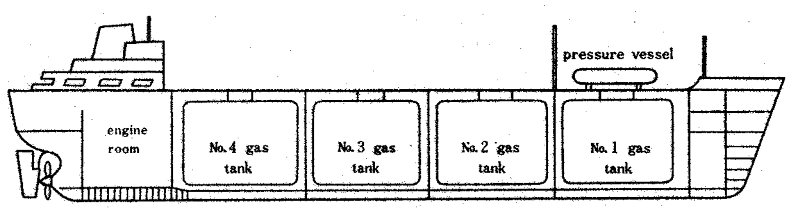

Figure 6: LPG, LNG carrier

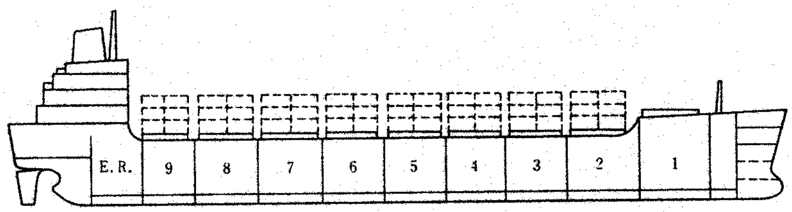

Figure 7: Container carrier

Table 3: Average Freight Ton-Mile Transportation Price by Mode 96

Mode	*Price (cents/ton-mile)*	*Scaled percentage*
Truck	25.08	100
Rail	2.50	10
Pipe	1.40	6
Water	0.73	3
Air	58.75	234

Low cost is one of the distinctive characteristics of water transportation. The costs in Table 3 (see Ballou [3]) are averages that result from the ratio of freight revenue generated by a mode to the total ton-miles shipped in 1996. The third column of the table gives the cost of the different modes relative to truck transport. With truck ton-mile transportation cost representing 100%, water transport costs 3% of the cost of truck transport, followed by 6% for pipeline transport and 10% for rail transport. However, while the ton-mile cost is low, other associated expenses such as terminal costs, including harbor fees, and the cost for loading and unloading cargo are relatively high. In spite of these high terminal costs, the low "line-haul cost" ensures that the marginal ton-mile cost drops significantly as distance and shipment size increases; thus the need for ever-larger ships. Consequently, in large enough vessels, water is the least-expensive transportation mode for bulk commodities in substantial volume over long distances.

New technological advances, such as GPS (global positioning systems) and GIS (geographic information systems), provide new tools for dynamically improving ship routing and scheduling. Although not in general use in the maritime industry, systems are available that allow ships at sea to interact in real time with the main office and receive instructions about any modifications in their route and schedules that take advantage of favorable currents or revised delivery agreements. The instructions are communicated by a main computer that regularly receives information from all the ships in a company's fleet as well as from ports about any docking delays. The computer also solves new (re-routing) problems that may be presented by the information arriving either from the ships or from ports. Such capabilities are possible and would give a maritime enterprise a significant edge over its competition. To date, however, few maritime enterprises have embraced quantitative decision tools to help with operational planning. This failure puts the industry far behind the trucking and airline industries, which are realizing enormous gains from their implementation of similar tools.

1.3 Maritime Logistics Problems

This section gives a brief overview of several maritime logistics problems, including brief synopses of the literature on attempts at tackling related problems. We begin by pointing out the major differences between routing and scheduling problems for land networks and for water networks. First, ports (nodes in the transportation network) are usually multi-functional, serving as supply points, demand points, distribution centers and intermodal centers, all depending on the cargoes, ships, and harbor facilities. Second, compared with vehicle routing, relatively few ships and ports are involved. Third, ships tend to travel long distances at relatively low speeds, and they are able to travel 24 hours a day for weeks at a time. Therefore, maritime transportation usually poses no critical planning horizon. Generally, an optimal solution (route and schedule) varies according to the changes in the planning horizon, and random effects increase as the planning horizon lengthens. It is standard to set the planning horizon according to the fiscal period of a business. Alternatively, one can consider a meaningful period of time or implement a rolling horizon concept (see Sherali et al [49]). Moreover, different ships on the same route differ in performance. Another difference between ship and vehicle scheduling is that in ship scheduling, fleet size is a much more important factor. This is because the cost of operating one additional ship has a much larger impact on the solution than would one more vehicle.

The earliest work on ship routing and scheduling has been surveyed by Ronen [43, 45]. Ronen [43], categorizes 1970's and earlier ship routing problems into the three categories of liner, tramp, and industrial operations. The liner operational problem is analogous to the ship routing model in our categorization scheme. Tramp operation is the problem of assigning an optimal sequence of cargoes to each vessel in a given fleet, which falls into the categorization of ship routing and scheduling in our schema. Industrial operations are problems in which the owner of the cargo controls the ship, and this also falls into our categorization of ship routing and scheduling. Ten years later, Ronen [45] updated his survey to cover the 1980's models. This time the author categorized the problem into four major categories of fleet sizing, inventory routing, optimal cruising speed, and ship scheduling. In contrast to Ronen [45], our survey is more focused on the routing itself rather than on fleet sizing and speed of the ship, although those problems are defined herein to illustrate the breadth of the challenges facing the industry.

Bulk shipping is set apart by the continuous nature of the product. The decision variable of quantity load and/or unload is continuous and seems easier to solve compared with problems with integer variables. However, products involved with bulk shipping have more restrictions, such as the requirement that a product be handled separately, which means each ship must have a dedicated compartment or dedicated ships in the case of liquid bulks. Therefore, usually we need to consider assignment of the product into the compartment, and this complicates the problem. Typical of the nature of liquid products, most crude oil carriers must return empty from the destination to the origin.

In the remainder of this section, maritime transportation logistics problems of bulk materials are classified into four categories of *Ship Routing, Ship Routing and Scheduling, Inventory Routing* and other combined and complex models. Ship Routing problems involve decisions on the sequence of ports to visit for each fleet of ships on a fixed route. Ship routing and scheduling problems consider the distribution problem in a case in which sets of cargoes are specified by loading, discharging port, and time, whereas Inventory routing is constrained to maintain local inventory of the product. Typical examples of enterprises confronted with these problems are briefly overviewed, and the recent literature in each problem area is surveyed.

1.3.1 Ship Routing

The objective of ship routing is to maximize profit by determining the optimal sequence of ports of call for each ship, the number of trips each ship makes in a planning horizon and the amount of cargo transported between any two ports by each ship. In the operation of charters, the goal is to find the best route and to evaluate the profit potential for each ship in order to determine if chartering additional vessels makes sense. Shipping companies operating fleets of ships for general cargo transportation on a particular trade route fixed by two end ports must deal with this kind of problem. For example, a company operating a regular route with two ships for pickup and delivery of cargoes will want to determine the best sequence of ports for each ship (see Figure 1.3.1).

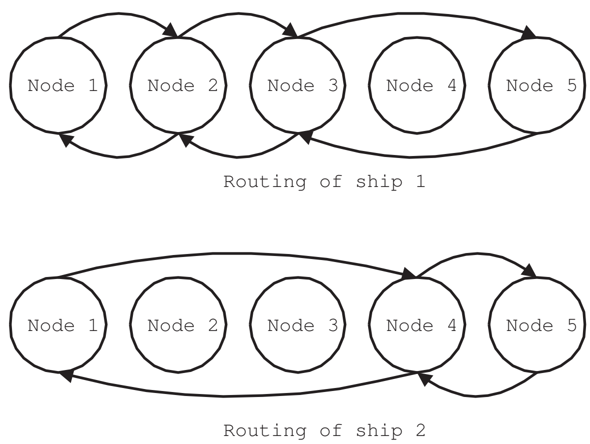

Figure 8: A possible route of two ships for fixed route

Periods of excess demand and/or capacity create a need for chartering decisions. Faced with high demand, a company may want to charter one or more ships for a certain period. In this case, knowing the profit potential of all candidate ships is crucial to a decision designed to meet all of the forecasted demand. On the other hand, a business downturn that creates excess capacity presents opportunities to fill the space using modern techniques of yield management that were pioneered by the airline industry.

1.3.1.1 *Literature Survey*

Rana and Vickson [41] formulated a mathematical programming model for optimally routing a chartered container ship that helps to manage the decision of whether to charter a container ship, and, if so, which size/type of ship is most appropriate in order to maximize profit. The model determines the optimal route, the number of containers transported between ports, and the number of trips the ship makes during the charter period. The problem is formulated as a nonlinear integer programming problem which is converted into to a linear mixed-integer programming problem by fixing an integer variable to a constant and solving it several times by changing the value of this integer variable. The mixed-integer problem is solved by the Benders decomposition method by solving the cargo allocation subproblem and the integer network subproblem.

In a later work, Rana and Vickson [42] considered the scenario of a maritime company whose fleet of ships must service a network of ports. This differs from the above model suggested by Rana and Vickson [41] by considering multiple ships, and its objective is to find the best route for each of the ships. The nonlinear integer programming formulation is solved by the Lagrangian relaxation method. The Lagrangian relaxation problem is separated into nonlinear integer programming subproblems and each subproblem is decomposed further into several linear mixed-integer programming problems and solved independently.

Fagerholt [21] considers the problem of finding an optimal number of fleets and routes for a liner service. Their approach solves the problem by generating feasible routes using a dynamic programming algorithm for each ship and then applying a partitioning formulation to obtain an optimal solution.

Boffey et. al [9] studied heuristic algorithms to determine good routes for the container ships so as to increase the revenue of a liner service on the North Atlantic. Nemhauser and Yu [34] studied the common carrier transportation system to determine the number and the starting time of services so that the total profit is maximized.

1.3.2 Ship Routing and Scheduling

The objective of determining optimal routes and schedules is to minimize the cost of ship operation within a planning horizon under the condition that all cargoes are transported to their destination within time windows. Usually, there exists a single source of supply for each product type, and each cargo consists of the same product with discharging locations and time windows.

1.3.2.1 Literature Survey

Psaraftis et al [40] suggest an optimal polynomial time algorithm for a single ship routing and scheduling problem with time windows in a case in which the shoreline is a straight line. It also presents heuristics for a general problem.

The paper of Brown et al [12] presents and solves a crude oil tanker routing and scheduling problem. Each tanker is assigned to a single origin and to a single destination with full shipload. All tankers are assumed to be of the same size and to have a single compartment. Each cargo has loading and discharging dates and ports. To obtain an optimal solution for this routing and scheduling problem, all feasible schedules are first generated for each cargo. Second, an optimal speed is selected for each cargo schedule. Finally with these all feasible schedules, this method solves a set partitioning problems to determine the least expensive schedule for each cargo.

The paper of Fisher and Rosenwein [27] considers the efficient scheduling of ships engaged in pickup and delivery of bulk cargoes. Each cargo consists of a designated quantity of a product to be lifted from one or more load ports to one or more destination ports with time windows. This algorithm first generates a candidate schedule for each ship that contains all feasible solutions. This guarantees optimality, or alternatively can be heuristically limited to contain only those schedules likely to be in an optimal solution according to the size of the problem and the computational time requirement. Choosing one optimal solution within the candidate schedule is formulated as a set-packing problem and solved by a dual method of the lagrangian relaxation algorithm. A similar study by Kim and Lee [29] describes a decision support system formulated as a set packing problem.

The paper of Sherali et al [49] present a solution for routing and scheduling oil tankers transporting multiple cargoes that are compartmentalized with loading and unloading time windows. Each voyage has a single origin and destination and each cargo is a full shipload. It considers a pre-determined penalty that is incurred when shipments are not delivered within time windows. Taking a different approach from those proposed by Brown et al [12] and by Fisher and Rosenwein [27], Sherali et al [49] do not generate a feasible schedule but instead incorporate the process of selecting a feasible schedule within the mixed-integer programming model itself. The formulation is enhanced by inclusion of valid inequalities (see Nemhauser and Wolsey [33]) and by incorporating a rolling horizon concept. The problem is solved by the branch and bound method to obtain a near optimal 5% gap of the lower bound of the solution.

The paper of Ronen [44] addresses the problem of finding the route and schedule that minimizes the cost of travel and port charges. This paper considers a fleet of ships of different sizes (capacities) that deliver bulk or semi-bulk cargoes from a single origin to many destinations and then return. Time windows are not considered.

The paper of Ronen [44] addresses the problem to find the route and schedule that minimizes the cost of traveling and port charges. The set of ships with different sizes (capacities) deliver a set of different bulk or semi-bulk from the single origin to many of their destinations and return back to the single origin without considering time windows. Their approach assesses the utility of three algorithms: a single-step cost minimization heuristic solution and a biased random generator of schedules that selects the least expensive schedule out of the many generated; and an optimizing algorithm based on a mixed binary nonlinear formulation suitable only for small problems. Later work done by Cho and Perakis [14] reformulates Ronen's [44] nonlinear, mixed-integer program into a linear one by eliminating the nonlinearities of the original model and reducing the integer variables.

The paper of Papadakis and Perakis [35] discuses the problem of a fleet of ships carrying a specific amount of bulk cargo to several destination ports within a specified time window. Each vessel of the fleet may load at any origin, and unload at the destination and return to the same origin. It formulates the problem to minimize the operating cost by considering speed selection and vessel allocation problems at the same time. The solution strategy incorporates methods that under certain conditions allow the decoupling of the vessel allocation and speed selection problems. In a general case, it arrives at a solution through a Lagrangian algorithm.

The purpose of the papers of Fagerholt and Christiansen [24, 25] is to solve a ship scheduling and allocation problem in a situation in which a fleet of ships is engaged in the pickup and delivery of various dry bulk products within specified time windows. Each ship is equipped with flexible cargo holds to separate different types of dry bulk cargo. The formulation of their ship scheduling and allocation problem is based on knowledge of a candidate schedule for each ship and for each cargo. To generate the candidate schedule for the formulation, a traveling salesman problem with allocation, time window and precedence constraints should be solved as a subproblem. This subproblem is solved as a shortest path problem on a graph whose nodes are the states representing the set of nodes sequenced in the path, the last visited node in the path and the accumulated cargo allocation when leaving the last visited node. The arcs of the graph represent transitions from one state to another. The optimal solution is achieved by a set partitioning approach consisting of two phases. In the first phase, numerous candidate schedules are generated by a forward dynamic programming algorithm that extends an existing schedule by adding one more cargo at a time. This generates candidate schedules and ship scheduling. In the second phase, the formulation of a set partitioning problem is solved for an optimal solution by using candidate schedules generated in the first phase.

The paper of Fagerholt [23] deals with a topic similar to that of Fagerholt and Christiansen [25], but it considers more of the issue of soft time windows by allowing controlled time window violations at an appropriate penalty cost for some customers and by searching for possibilities that significantly reduce the transportation cost. It separates time windows for each cargo into two categories of inter- and outer-time windows. It starts by generating feasible candidates for each ship and calculates the corresponding operations cost including the inconvenience (penalty) cost. At the next step, the algorithm solves a set partitioning problem.

Many navy applications require the solution of routing and scheduling problems. Naval applications have specialized objectives such as maximizing the utility of a fleet of vessels as opposed to the objective of minimizing the cost of operating the vessels in general. Related papers include Cline et al [19], Brown et al [11, 10], and Darby et al [20].

In general, ship routing and scheduling problems are formulated as set partitioning problems with preprocessing used to generate candidate schedules for each ship. In contrast, Sherali et al [49] use a different approach that incorporates mixed-integer programming with a branch and bound method. Ronen [44], and Papadakis and Perakis [35] also differ in their solution approaches by formulating the problem as a nonlinear program and solving it by a Lagrangian algorithm. In all of the cited works on ship routing and scheduling, the satisfaction of time windows creates most of the computational complexity.

There are differences between each paper in modeling considerations. Fagerholt and Christiansen [24], [25] and Fagerholt [23] consider multiple compartments with flexible cargo holds, but Brown et al [12] consider only one product that represents a full shipload, an approach that simplifies the problem. Fisher and Rosenwein [27] take an approach in which cargoes can be less than a full shipload. However, in Fisher and Rosenwein [27] cargoes can be loaded after every remaining cargo has been unloaded, while Fagerholt and Christiansen [24], [25] permit loading before other cargoes are unloaded. In another variation, Brown et al [12] and Papadakis and Perakis [35] consider the selection of optimal speed. Bausch et al [5] and Sherali et al [49] deal with fixed multiple cargo space while Fagerholt and Christiansen [24], [25] consider flexible cargo holds. Sherali et al [49], and Fagerholt [23] introduce into consideration flexible time windows with a penalty concept.

1.3.3 Inventory Routing

The inventory routing problem is a distribution problem in which each customer maintains a local inventory of a product. Some nodes consume a certain amount of product daily, and others produce a certain amount of product each day. The objective is to minimize delivery costs while attempting to ensure that no customer runs out of the commodity, and no producer has to stop production because of limited storage capacity. This type of problem is typically faced by major oil companies that directly control fleets of ships or sometimes use spot chartered vessels for the transport of raw materials used in their business. Generally, the ship owner transports the cargo (usually chemical products) so as not to be short of the resources needed to operate. Sometimes in chemical industries, continuous production is required because of the huge setup costs to restart production. In such cases, proper inventory maintenance is critical.

1.3.3.1 Related surveys

Miller [32] described a fleet scheduling and inventory resupply problem faced by an international chemical company that transfers multiple chemicals from one point to multiple destinations with a requirement to maintain a specified level of inventory.

Christiansen [15] presents a two-pronged problem that combines inventory management and routing with time-window constraints. A fleet of ships transports a single bulk product between production and consumption harbors. The quantities loaded and discharged are determined by the production rates of the harbors, stock levels, and the actual ship visiting the harbor. The paper formulates the problem into network flow models with consideration of loading and discharging conditions, time constraints and inventory levels at the harbor. The solution is determined by the Dantzig-Wolf decomposition method in which ship routing and inventory management are decomposed into subproblems. Each subproblem is formulated as a shortest path problem and solved by a dynamic programming algorithm explained in Christiansen and Nygreen [18]. By solving dynamic programming problems for each ship and harbor, this method generates paths for each ship, including information about the geographical route, the load quantity, and the start time at each harbor arrival. Similar paths also are generated for each harbor, including information about the number of arrivals at the harbor, the load quantity, and the start time at each harbor arrival. The best columns that correspond to the ship route and the harbor visit sequence are generated by solving subproblems. The complexity of the problem depends on the number of possible routes for each ship. The paper by Christiansen and Nygreen [17], introduces the idea of reducing the size of the time windows and decreasing the route possibilities, which is actually a preprocessing phase to solve the master problem posed by Christiansen [15].

The paper of Larson [30] presents a problem of transporting sludge from 14 wastewater treatment plants in New York city to disposal sites 106 miles off shore. Each plant has an average sludge production rate and a holding capacity. First the author calculates the safety stock level and the maximum deterministic time interval, accounting for the probabilistic behavior of inflowing sludge, that a ship has to pump out sludge so that storage tank capacity is not exceeded at any plant. Second, using the time interval determined in the preceding stage, candidate vessel tours are generated. The next step is to select, by a heuristic algorithm, the best tour generated.

The nature of inventory management forces us to analyze stochastic system behavior. The general approach is to consider the time interval in which each facility should be loaded or unloaded considering the safety stock level. Without interrupting continuous production of each facility, candidate schedules of vessels are generated and the best route is selected.

1.3.4 Other Combined and Complex Models

In the real world, ship routing and scheduling problems can be partial or total combinations of the models suggested in the previous sections. In either combination, the scale and complexity of the problem increases dramatically and becomes harder to solve. This is especially true when other decision variables such as ship speed, multiple compartments, or special parameters such as ocean currents and weather conditions become factors.

According to Ronen [45], reducing speed by 20% will reduce fuel consumption by about 50%, and from 20% to 60% of daily operating cost is dependent on fuel cost. As the price of fuel increases, this area of study becomes particularly important. As we can imagine, speed and fuel consumption are nonlinearly dependent and make the problem harder if combined with other decision factors. Related studies include Berford [7], Papadakis and Perakis [35], Perakis [36], Perakis and Papadakis [37], Perakis and Papadakis [38], Brown et al. [12], Bausch et al. [4], Schrady and Wadsworth [47].

The amount of research done on multicompartment ship transportation is very limited. This is because container ships do not need to consider multicompartments, and a typical type of bulk material such as crude oil is transported usually by one ship. Another reason is because of the difficulty of designing a multicompartment ship unless decisions are made at the shipbuilding stage about what bulk material each compartment will be used for. However, some multicompartment ships are used in non-ocean going areas such as on the Great Lakes, in the Philippines, and on the Mississippi River. Because of the nature of bulk liquids, compartments typically are dedicated to a specific product. This compartment dedication is not typical when the cargoes are dry bulk. That is why some of the ships for transportation of dry bulk have flexible compartments with multiple bulkheads positions. The study for the fixed compartment ship routing and scheduling is done by Bausch et al. [5]. Fagerholt and Christiansen [24, 25], and Fagerholt [22] studied the ship routing and scheduling problem for multiple products with flexible compartments.

Other researchers, including Williams [57], have used heuristic algorithms to tackle the problem of replenishment at sea. Lo and McCord [31] used technological advances in satellite altimetry to factor ocean currents into their study of transportation by ship. Wang and Chretienne [56] used dynamic programming to suggest a heuristic approach to situations in which weather forecasts are only available for a limited time ahead of a ship's travel.

In this dissertation, we introduce the problem of designing a minimum cost routing schedule for a heterogeneous fleet of ships engaged in pickup and delivery of various liquid bulk cargoes. Each port is defined as either a supply or demand port, according to the chemical involved, and production and consumption rates and storage capacities are specified. Each ship has dedicated compartments for multiple products. This problem falls into the categorization of inventory routing. It is formulated as a mixed-integer nonlinear program and transformed into a mixed-integer linear program. This problem extends the bulk-shipping model of Christiansen [15] and Christiansen and Nygreen [17, 18] in several ways. Specifically, we consider pickup and delivery of *multiple commodities* using ships having *multiple dedicated compartments*. Additionally, we allow more than one ship to be docked in a harbor at the same time. The papers by Fagerholt and Christiansen [24, 25] deal with multiple products and compartments. However, these papers assume flexible cargo holds that are suitable only for dry bulk products and not for liquid bulk products in dedicated compartments such as those considered in this dissertation.

CHAPTER II

APPLICATIONS AND MODEL

2.1 Introduction

This chapter addresses the problem of determining a minimum cost routing schedule for a heterogeneous fleet of ships engaged in pickup and delivery of various bulk liquid cargos across a set of supply and demand harbors with specified product availabilities and needs, respectively. Due to the nature of the products, it is impossible to carry more than two products without being separated into dedicated compartments of the ships. The optimal routing schedule should specify how much of each product to carry from which port to which port, at what time, and on which ship, subject to the conditions that all ports must have sufficient product for consumption, and the stock levels of the products cannot exceed the inventory capacity of that port.

This problem is motivated by a real logistics problem faced by an oil company in Asia Pacific serving an archipelago of islands. This company has a fleet of tankers and barges that transport petrochemical products between various plants and has many storage terminals and direct customers. Since plants and customers are dispersed over many islands, and since there is no terrestrial transportation infrastructure, such as a pipeline network connecting the islands, it is necessary to carry all inter-island supply and demand by ships. Each island has a different production and consumption rate for specified products, and the inter-island transport schedule should be such that proper stock levels for the petrochemicals are maintained at each island during the planning horizon. The problem is further complicated by the fact that the ships are able to carry a number of different products at the same time, and since some of these products cannot mix, these need to be carried in separate dedicated compartments. Figure 2.1 illustrates the problem for eight harbors, four products, and three ships in the Philippines.

In this chapter, we first identify the most important logistics considerations for this

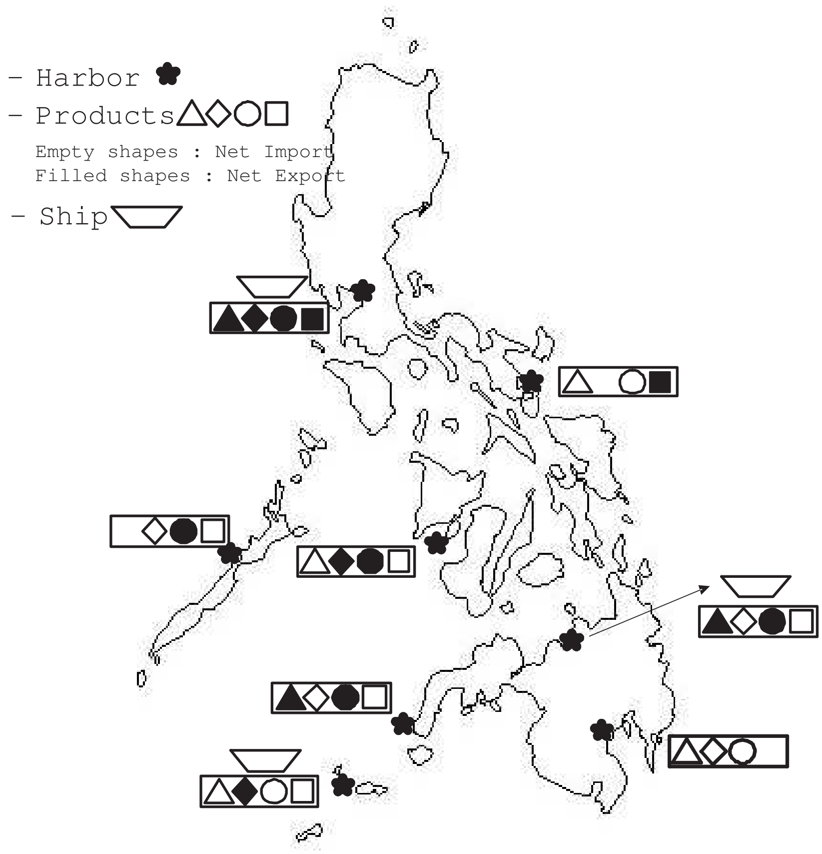

Figure 9: A 4 commodity problem with 8 harbors and 3 ships

difficult ship-routing problem. Next, using a network flow model, we formulate the problem as a combined multi-ship pickup-delivery problem. The interaction between multiple ships arriving at the same destination makes the formulation highly nonlinear. We use novel linearization schemes to develop an *equivalent* mixed-integer linear programming reformulation for the problem.

The remainder of this chapter is organized as follows. In Section 2.2, we review the existing literature on ship routing problems. Section 2.3 describes the critical characteristics of the problem under consideration. Section 2.4 develops an optimization model for the problem, and Section 2.5 presents equivalent linear reformulations of the nonlinear constraints in this model. Section 2.6.1 shows an illustrative example with the optimal solution obtained by a commercial optimization solver. Finally, Section A.4 offers some concluding remarks. Two appendices include a notational summary and the proof of a key result.

2.2 Maritime Routing and Scheduling Literature

Operations Research has long recognized the need for systematic mathematical techniques for the optimal routing and scheduling of vehicles to meet the needs of a dispersed set of customers. Models and solution algorithms for these so-called *vehicle routing* (cf., [53, 26]) problems have revolutionized the operations of trucking industries. However, even though approximately 90% of the volume and 70% of the value of all goods transported worldwide is carried by sea [39], until recently, relatively little work has been done on optimization based routing and scheduling of ships. In this section, we review some of the existing work done in this area.

Miller [32] described a fleet scheduling and inventory resupply problem faced ay an international chemical company that transfers multiple chemicals from one origin to multi destinations under the condition that certain inventory level is maintained.

Ronen [44] addresses a problem of scheduling the shipment of large quantities of a bulk or semi-bulk commodity from one origin area to many destination ports. A set of ships with different capacities deliver a set of different shipments to their destinations and return back to the single origin. The model doesn't consider any time window constraints. The paper suggests and compares three different algorithms: a single step cost minimization heuristic, a biased random search that chooses the cheapest schedule out of many generated schedules, and an optimizing algorithm based on a mixed binary nonlinear formulation.

Brown et al. [12] discuss the problem of routing and scheduling crude oil tankers. The problem is faced by a major oil company which controls a fleet of several dozen crude oil tankers of similar sizes, and uses them to ship crude oil from the Middle East to Europe and North America. A voyage usually has a single loading port and a single discharging port and the cargo is a full shipload. The paper explicitly considered constraints on loading/discharging durations (time windows) for each port. The authors suggested an enumerative solution method where all feasible schedules are generated, and the cheapest schedule is selected.

Baush et al. [5] discuss the distribution of multiple liquid bulk products among plants, distribution centers, and industrial customers by using vessels equipped with multiple compartments during the planning horizon of 2-3 weeks. Each cargo consists of earliest loading date and location, and latest discharging date and location. Authors generate all feasible schedules for all vessels and choose best schedule for each vessel to minimize the cost of schedule which includes idle cost of vessels and spot charter cost during the planning horizon.

Fisher and Rosenwein [27] considered a bulk shipping problem, where each cargo consists of a designated quantity of a product to be lifted from one or more load ports and delivered to one or more destination ports within specified time windows. The solution algorithm proposed in this paper first generates a menu for each ship that contains all feasible solutions that guarantee optimality or alternatively can be heuristically limited to contain only those schedules likely to be in an optimal solution. One optimal solution is then chosen from the menu by formulating a set-packing problem and solving it using Lagrangian relaxation.

Papadakis and Perakis [35] discuss the problem of a fleet of ships carrying a specific amount of bulk cargo from several destination ports during a specified time interval. Each vessel in the fleet may load at an origin, unload at a destination and return to its origin. The problem considers only one type of cargo. In addition to ship routing and pickup, the paper also considers optimal speed selection for the ships. The solution method is based on decoupling the speed selection problem from the vessel allocation problem using Lagrangian relaxation.

Christiansen [15] presents a combined inventory management problem and ship routing problem with time windows. A fleet of ships transport a single product between production and consumption harbors. The quantities loaded and discharged are determined by the production rates of the harbors, possible stock levels, and the actual ship visiting the harbor. The author combines a Dantzig-Wolf decomposition approach with branch-and-bound to solve the problem.

Christiansen and Nygreen [18] consider the same problem as in [15]. In this paper, the authors used a path flow formulation to generate paths for each ship including information about the geographical route, the load quantity, and the start time for each harbor arrival. The method also generates paths for each harbor including information about the number of arrivals to the harbor, the load quantity, and the start time for each harbor arrival. The path generation problems are used as subproblems in a column generation scheme to solve the overall planning problem. The authors also consider methods for reducing the width of the time windows to reduce the number of feasible paths generated [17].

Fagerholt and Christiansen [24] consider a combined multi-ship pick up and delivery problem with time windows and multi compartments for *dry* bulk. Each ship in the fleet is equipped with a flexible cargo hold that can be partitioned into several small holds in a given number of ways. Consequently, multiple products can be delivered by the same ship at the same time. A set partitioning approach with two phases is proposed as a solution method. In the first phase, a number of candidate schedules for allocation of cargos to the ships' cargo holds is generated. In the second phase the total transportation cost is minimized by solving a set partitioning problem where the columns correspond to the candidate schedule generated in the first phase. Fagerholt and Christiansen [25] consider a special type traveling salesman problem with allocation, time windows and precedence constraints. This problem occurs as a subproblem of the model in [24].

Ronen [46] addresses inventory routing problem faced by producers of multiple liquid bulk products. The objective is to minimize the cost of shipping while ensuring the stock level requirements of the producing origins and consuming destinations. Author segmented the planning horizon on daily basis and decide the time, quantity, origin and destination for each product to deliver.

A recent review of ship routing and scheduling by Christiansen et al. [16] includes some papers above ([44], [12], [5], [15], [18], [24] [46].) The authors survey the literature for various models that have been developed and solved catagrized by type of operations (tramp, liner, industrial, millitary, etc.)

Recently, Jetlund and Karimi [28] consider the maximum profit scheduling for a fleet of ships delivers multiple liquid bulk cargoes. Each cargo needs to be delivered from the pick-up port to discharge port with time windows. Author formulate a mixed-integer linear programming problem and suggest a heuristic method with the result of profit increase compared to the schedule actually used by a multi national shipping company in the Asia Pacific region.

2.3 Multi-Commodity Bulk Shipping

In this section we describe some of the critical characteristics of the multi-commodity bulk shipping problem under consideration.

We consider a heterogeneous fleet of ships and barges. The ships have dedicated multiple compartments to be able to carry different products simultaneously. The ships in the fleet differ by size, number of compartments, and the set of products they can carry.

The fleet is used to distribute multiple liquid bulk products amongst geographically dispersed ports. Each port is either a producer or a consumer for a certain commodity, and the average production and consumption rate for each commodity is known. A ship loads a product from a producing port or harbor, and unloads it at a consuming harbor. Partial loading of the ship is allowed.

The loading and unloading of a ship at a harbor is carried out in one of the piers or jetties. It is assumed that each harbor has enough piers to accommodate all incoming ships. There is no dedicated pier for any cargo type. However it is impossible to *simultaneously* load or unload different products onto a ship at a pier. Furthermore, more than two ships *cannot* be simultaneously loading and/or unloading the same product.

Under the above conditions, our problem is to determine which product is to be loaded into (or unloaded from) which compartment of which ship, the quantity to be loaded/unloaded, the time period of loading/unloading, and to schedule the arrivals and departures of the ships so as to maintain the inventory levels between operating bounds during the planing horizon. The overall plan should minimize the total daily cost of the ships: fuel costs, port and canal dues, and loading and unloading charges over a finite planning horizon. Our model considers a captive fleet of ships and does not consider chartering vessels.

At the beginning of planning horizon T_i, it is assumed that the starting position and the cargo for each ship is known. Finally, it is also assumed that each ship starts from some harbor in the beginning of the planning horizon and finishes in some harbor at the end of the planning horizon. However, in practice, we would deal with the situation that ships are enroute at the start and end of a planning horizon by implementing a rolling horizon concept as follows. At the start of the first planning horizon T_1, we generate dummy harbors (with no supply or demand) at the position of the ships in the middle of the sea. The problem is solved based on the current information (e.g., location of ships, inventory levels of products, etc.). At a specified time *prior* to the end of T_1, we begin a new planning horizon T_2 with routing decisions *from* all enroute dummy harbors replacing the destinations determined by the previous planning horizon T_1.

The above ship routing problem is quite similar to the bulk-shipping problems considered by Christiansen [15] and Christiansen and Nygreen [17, 18]. As in these papers, we consider inventory constrained scheduling of a heterogeneous fleet of ships, where there is no central source of supply. On the other hand, the major difference is that we consider pickup and delivery of *multiple commodities* using ships with *multiple dedicated compartments.* Additionally, we allow more than one ship to be docked in the same harbor at any given time. The papers by Fagerholt and Christiansen [25, 24] deal with multiple products and compartments. However, these papers assume flexible cargo holds that are suitable only for dry bulk products and not for liquid bulk products in dedicated compartments such as those considered in this paper. These difference increase the complexity and difficulty of the problem considered herein.

2.4 Model Formulation

In this section, we describe a mathematical model for the problem under consideration. The model is developed along the lines of Christiansen [15], but with significant modifications to account for multiple commodities, dedicated ship compartments and multi-ship port calls with overlapping docking times. In the following formulation, the decision variables are written in lower case letters and the parameters and sets are written in upper case letters. To keep the notation relatively simple, we assume that all ships are in ports at both the start and finish of the planning horizon; i.e., no ship is enroute at the beginning and end of the model's scheduling period.

2.4.1 Routing Constraints

The routing constraints define and link the sequence of arrivals and departures of the various ships to and from various harbors. Let V denote the set of all ships. Following Christiansen [15], let us define a network whose nodes are labelled (i, m), where i denotes a harbor, and m is the arrival number at that harbor within the planning horizon. For example, node $(2, 1)$ denotes the first arrival to harbor 2. We shall refer to such a pair as a *position*. Figure 10 is an example network for 2 ships, and 3 harbors each having 2 positions. Let H_T be the set of all harbors, and let M_i be the set of possible arrival numbers

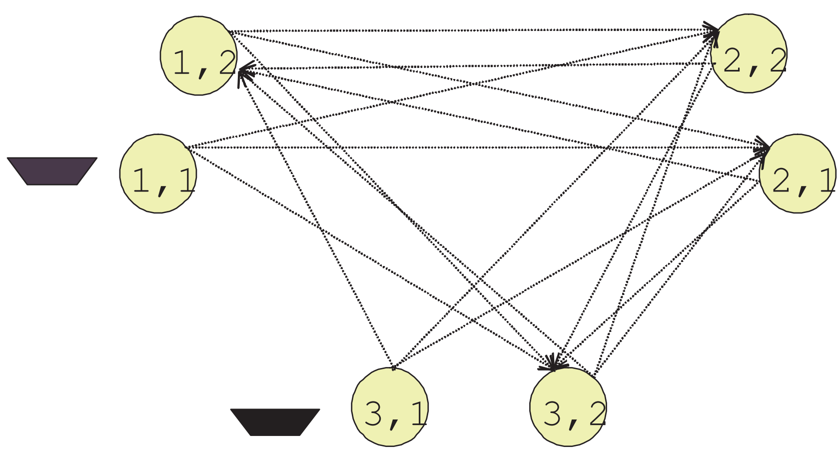

Figure 10: Network model of 2 ships, 3 harbors with 2 positions for each harbor

$\{1, 2, \ldots, \mu_i\}$ at harbor i where μ_i is a specified number of arrivals to harbor i. We let S_T denote the set of all feasible positions (harbor-arrival pairs); i.e., $S_T = \{(i, m) \mid i \in H_T, m \in M_i\}$. Let $(i_v, m_v) \in S_T$ denote the *initial position* of ship v. For example, if ship v is initially located at harbor i, then $i_v = i$. If harbor i initially has only one ship v, then $m_v = 1$. In cases when there are ρ_i ships starting at harbor i, then arbitrarily set an arrival sequence number $m_v \in \{1, 2, \ldots, \rho_i\}$ for each ship $v \in \{v_1, v_2, \ldots, v_{\rho_i}\}$. Let $S_0 := \{(i_v, m_v) | v \in V\}$ be the set of initial positions for all ships. Then $S_N := S_T \backslash S_0$ is the set of all possible positions that ships can occupy *after* leaving their starting positions.

For $i \neq j$, for all $(i, m) \in S_N \cup \{(i_v, m_v)\}$, and $(j, n) \in S_N$, we set the binary variable

x_{imjnv} equal to 1 if ship $v \in V$ has a route segment that includes harbor i as the m-th arrival followed *immediately* by a visit to harbor j as the n-th arrival.

2.4.1.1 Initial Position Constraints

The following constraints enforce the requirement that each ship v must depart from its initial position:

$$\sum_{(j,n)\in S_N} x_{i_v m_v jnv} = 1, \quad \text{for every } v \in V. \tag{C1}$$

To allow for ships to remain unused in a harbor for the entire planning horizon, we introduce the binary variable z_{imv} to equal 1 if ship v ends its route as the m-th arrival to harbor i. The constraint

$$\sum_{(j,n)\in S_N} x_{i_v m_v jnv} + z_{i_v m_v v} = 1, \quad \text{for every } v \in V.$$

ensure that ship v will not depart its initial position whenever $z_{i_v m_v v} = 1$.

2.4.1.2 Flow Conservation Constraints

Flow conservation constraints ensure that the m-th arrival to harbor i should either leave harbor i or end its route there. The flow conservation constraints

$$\sum_{(j,n)\in S_T} x_{jnimv} - \sum_{(j,n)\in S_N} x_{imjnv} - z_{imv} = 0, \quad \text{for every } (v,i,m) \in V \times S_N \tag{C2}$$

guarantee that $z_{imv} = 0$ if (i,m) is an intermediate position (non-initial position) and must equal to 1 if it is the final position of ship v's schedule. This is because for each (v,i,m), at most one ship v can occupy position (i,m) by the forthcoming constraints (C4); thus, $\sum_{(j,n)\in S_T} x_{jnimv} \le 1$. If there is such an arrival, then it must depart unless harbor i is the terminal point of the ship's journey during the planning horizon. In the latter case, $\sum_{(j,n)\in S_N} x_{imjnv} = 0$ so that constraint (C2) enforces $z_{imv} = 1$.

2.4.1.3 Route Finishing Constraints

To simplify our notation, we will assume that at the beginning and end of planning horizon, all ships are in port and not enroute to some destination. Terminating journeys at a port can be achieved by imposing the constraints

$$\sum_{(i,m)\in S_N} z_{imv} = 1, \quad \text{for each } v \in V. \tag{C3}$$

When ships are allowed to stay at their initial positions, we need to replace S_N by $S_N \cup \{(i_v, m_v)\}$ in constraint (C3).

2.4.1.4 One Time Visit Constraints

These constraints ensure each harbor-arrival pair (i, m) is visited at most once. Let the binary variable y_{im} equal to 1 when position (i, m) is *not* visited. Then,

$$\sum_{v\in V} \sum_{(j,n)\in S_T} x_{jnimv} + y_{im} = 1, \quad \text{for every } (i, m) \in S_N. \tag{C4}$$

ensure that at most one ship can be the m-th arrival to harbor i and y_{im} must be 1 when position (i, m) is not visited.

2.4.1.5 Arrival Sequence Constraints

Since it is not known a *priori* how many visits will be made to each harbor during a planning horizon, it is necessary to create enough positions (i, m) to allow as many visits as needed for an optimal solution. Clearly, not all positions in every harbor will be utilized. However, if harbor i does not have the $(m - 1)$-th arrival, then it cannot have the m-th arrival; conversely, if there is an m-th arrival, there must have been an $(m - 1)$-th arrival. This property can be expressed by the constraints

$$y_{im} - y_{i(m-1)} \geq 0, \quad \text{for every } (i, m) \in S_N. \tag{C5}$$

2.4.2 Constraints for Loading and Discharging

Constraints are needed to connect the quantities of various products to be loaded and unloaded at the various harbors to the capacities of the ships visiting these harbors. We introduce the following three sets of variables: q_{imvk}, which correspond to the quantity of product k loaded onto or unloaded from ship v at position (i, m); l_{imvk}, which correspond to the quantity of product k onboard ship v as it departs from position (i, m); and o_{imvk} is a binary variable indicating whether product k is loaded onto (or unloaded from) ship

v at position (i, m). The following sets of parameters will also be used: J_{ik} is equal to $+1$ (respectively, -1) if harbor i is a producer (respectively, consumer) of product k, and 0 otherwise; Q_{vk} is the quantity of product k loaded onto ship v at the start of planning horizon; CAP_{vk} is the capacity of the compartment onboard ship v dedicated to carry product k. The set of all products that ship v can carry is denoted by K_v.

2.4.2.1 Ship Load Constraints

If a ship v travels from position (i, m) to position (j, n), then the quantity l_{jnvk} of product k onboard at departure from (j, n) should equal the quantity l_{imvk} onboard at departure from (i, m) plus, if $J_{jk} = +1$, (respectively, minus, if $J_{jk} = -1$) the quantity q_{jnvk} loaded (respectively, unloaded) at (j, n). But this will only happen if ship v travels from (i, m) to (j, n); i.e., if $x_{imjnv} = 1$. Therefore, the loading constraints can be expressed as

$$x_{imjnv}[l_{imvk} + J_{jk} q_{jnvk} - l_{jnvk}] = 0,$$
$$\text{for each } v \in V \text{ and every } (i, m, j, n, k) \in A_v \times K_v \tag{1}$$

where $A_v := \{(i, m, j, n) | i \neq j, (i, m) \in S_N \cup S_0^v, (j, n) \in S_N\}$ is the set of all feasible arcs for ship v in the network. The above constraints are nonlinear, but we will derive an *equivalent* linear system in Section 2.5.

2.4.2.2 Initial Ship Load Constraints

The amount $l_{i_v m_v vk}$ of product k onboard ship v at departure from the initial position (i_v, m_v) should be equal to the initial quantity Q_{vk} onboard plus if $J_{i_v k} = +1$ (respectively, minus if $J_{i_v k} = -1$) the quantity $q_{i_v m_v vk}$ loaded (respectively, unloaded) at the initial position. Thus,

$$Q_{vk} + J_{i_v k} q_{i_v m_v vk} - l_{i_v m_v vk} = 0, \quad \text{for each } v \in V \text{ and every } k \in K_v. \tag{C6}$$

2.4.2.3 Compartment Capacity Constraints

The amount l_{imvk} of product k onboard ship v at departure from position (i, m) cannot exceed the capacity CAP_{vk} of the compartment dedicated for product k. However, this will only meaningful if ship v visits (i, m); i.e., $\sum_{(j,n)\in S_T} x_{jnimv} = 1$, otherwise the quantity $l_{imvk} = 0$. Therefore, the compartment capacity constraints can be expressed as

$$l_{imvk} \leq \sum_{(j,n)\in S_T} CAP_{vk} x_{jnimv}, \quad \text{for each } v \in V \text{ and every} (k, i, m) \in K_v \times S_N. \qquad \text{(C7)}$$

2.4.2.4 Servicing Product Constraints

Introduce the variable o_{imvk} to indicate when product k is serviced at position (i, m) by ship v. We want o_{imvk} to be 1 if q_{imvk} is positive, otherwise it should be 0. That is, we want to ensure that the quantity q_{imvk} of product k loaded onto ship v at position (i, m) cannot exceed the capacity CAP_{vk} of the compartment of ship v dedicated for product k. This is expressed as

$$q_{imvk} \leq CAP_{vk} o_{imvk}, \quad \text{for each } v \in V \text{ and every } (k, i, m) \in K_v \times S_T. \qquad \text{(C8)}$$

2.4.3 Constraints for Time Aspects

Constraints are needed to define the arrival and departure times of the m-th arrival to harbor i. The variables used are: t_{im} is the time of the m-th arrival to harbor i; and t_{Eim} is the departure time of the m-th arrival to harbor i that is the service ending time at position (i, m). The following parameters are also used: TQ_{ik} is the time required to load (unload) one unit amount of product k at harbor i; W_i is the set-up time required to service a product at harbor i (for notational simplicity, we assume fixed set-up times for a port no matter what products are being serviced); and T_{ijv} is the time required by ship v to sail from harbor i to harbor j plus the set-up time required at harbor j immediately preceding loading and unloading service times.

2.4.3.1 Service Time Sequence Constraints

Clearly, the m-th visit should occur after the $(m-1)$-th visit. That is,

$$t_{im} - t_{i(m-1)} \geq 0, \quad \text{for every } (i, m) \in S_N. \qquad \text{(C9)}$$

2.4.3.2 Service Finishing Time Constraints

The time of departure for the m-th arrival to harbor i (namely, t_{Eim}) equals the m-th arrival time (t_{im}) plus the time required to service all products ($\sum_{v\in V}\sum_{k\in K_v} TQ_{ik}q_{imvk}$) plus the fixed setup time W_i (incurred $\sum_{v\in V}\sum_{k\in K_v} o_{imvk}$ times) for switching from one product to another. This constraint is given by:

$$t_{im} + \sum_{v\in V}\sum_{k\in K_v} TQ_{ik}q_{imvk} + W_i \sum_{v\in V}\sum_{k\in K_v} o_{imvk} - t_{Eim} = 0, \quad \text{for every } (i,m) \in S_T. \tag{C10}$$

2.4.3.3 Route and Schedule Compatibility Constraints

If ship v travels from position (i,m) to (j,n) — that is, $x_{imjnv} = 1$ — then the arrival time t_{jn} at (j,n) is the sum of the departure time t_{Eim} from (i,m) and the travel time T_{ijv} from harbor i to harbor j by ship v. Thus,

$$x_{imjnv}[t_{Eim} + T_{ijv} - t_{jn}] \le 0, \quad \text{for each } v \in V \text{ and every } (i,m,j,n) \in A_v. \tag{2}$$

Notice that these constraints are only valid when the positions (i,m) and (j,n) are directly connected by ship v; i.e., when $x_{imjnv} = 1$. Constraints (2) are nonlinear, but we shall present *equivalent* linear reformulations (in Section 2.5) that are derived from global optimization theory.

2.4.4 Constraints for the Inventories

Inventory constraints connect the required stock levels at the harbors to the quantities loaded onto and unloaded from the visiting ships. The following variables are used: s_{imk} is the stock level of product k in harbor i at the time of the m-th arrival; s_{Eimk} is the stock level of product k in harbor i when the m-th ship departs; and p_{im} is a binary variable which is equal to zero if the m-th and $(m-1)$-th arrivals to harbor i overlap; i.e., the m-th ship arrives before the $(m-1)$-th ship departs harbor i. The set K_i^H represents the set of products that harbor i produces and consumes. The parameters used here are as follows: J_{ik} is set equal to $+1$ (respectively, -1) if harbor i is a producer (respectively, consumer) of product k; $R_{ik} > 0$ is the production (if $J_{ik} = +1$) or consumption (if $J_{ik} = -1$) *rate* of product k in harbor i; T is the length of the planning horizon; S_{MNik} is the minimum allowable stock level of product k at harbor i (safety stock); and S_{MXik} is the maximum allowable stock level of product k at harbor i (production/deliveries must stop when this level is reached).

2.4.4.1 Initial Inventory Constraints

We classify the harbors into two groups: those that have ships and those that do not at the start of the planning horizon. For those harbors that do not have ships (namely, $H_N := H_T \backslash \{j \mid (j,m) \in S_0\}$) the stock level s_{i1k} of product k in harbor i at the time of the first ship arrival is the amount IS_{ik} of product k in harbor i at the start of the planning horizon plus the amount produced when $J_{ik} = +1$ (or minus the amount consumed when $J_{ik} = -1$) until the arrival t_{i1} of the first ship; i.e.,

$$s_{i1k} = IS_{ik} + J_{ik} R_{ik} t_{i1}, \quad \text{for every } (i,k) \in H_N \times K_i^H. \tag{C11}$$

For those harbors that do have ships at the start of the planning horizon, $t_{i1} = 0$ so that $s_{i1k} = IS_{ik}$.

2.4.4.2 Inventory Level Constraints

Constraints are needed to measure the inventory level of product k at harbor i when the m-th arrival departs. (Recall that we allow the simultaneous servicing of multiple ships in the same harbor.) Suppose now that there are two ships in harbor i at the same time. Although the notation (i,m) determines which of the ships arrived first, it is not clear which ship leaves first. This can cause difficulties in modeling the inventory constraints. To tackle this issue, we make the simplifying assumption that the second ship entering harbor i will load or unload its quantity of product k with complete knowledge of how much of the same product the first ship will be loading or unloading. So, even when the first ship completes its service later than the second ship, the stock levels s_{im} and s_{Eim} will always be within their bounds (see constraints (C15) and (4)).

For product k in harbor i, if ship v is the m-th arrival, then the stock level s_{Eimk} equals the level s_{imk} before ship v arrives less the amount q_{imvk} loaded if $J_{ik} = +1$ (or plus the amount q_{imvk} unloaded if $J_{ik} = -1$) plus the amount produced (if $J_{ik} = +1$) while ship v is being loaded (or minus the amount consumed (when $J_{ik} = -1$) while ship v is unloading) at the rate R_{ik} during the time period $t_{Eim} - t_{im}$. These inventory constraints can be expressed as follows:

$$s_{imk} - \sum_{v \in V} J_{ik} q_{imvk} + J_{ik} R_{ik}(t_{Eim} - t_{im}) - s_{Eimk} = 0,$$

$$\text{for every } (i, m, k) \in S_T \times K_i^H. \qquad \text{(C12)}$$

2.4.4.3 Stock Level Constraints

Constraints are needed to ensure that the stock levels of a product are consistent between successive arrivals to a harbor. If only a *single* ship is allowed in a harbor at any time during the planning horizon, the constraints can be simply stated as:

$$s_{Ei(m-1)k} + J_{ik} R_{ik}(t_{im} - t_{Ei(m-1)}) - s_{imk} = 0, \quad \text{for every } (i, m, k) \in S_N \times K_i^H.$$

Now suppose that there are two ships in harbor i, which arrived as the $(m-1)$-th and m-th ship. It could easily be the case that the m-th ship starts servicing a product k, before the $(m-1)$-th ship begins its servicing of the same product. However, in our model, we make the simplifying assumption that the m-th ship will load or unload product k only after the $(m-1)$-th ship has completed its loading or unloading of the same product. The two ship constraint becomes:

$$s_{Ei(m-1)k} + J_{ik} R_{ik}[t_{im} - t_{Ei(m-1)}]p_{im} = s_{imk}, \quad \text{for every } (i, m, k) \in S_N \times K_i^H. \qquad (3)$$

Here, p_{im} is 0 if there are two or more ships in harbor i during the m-th arrival. Thus, if there are two ships, constraint (3) sets $s_{Ei(m-1)k} = s_{imk}$ so that overlapping does not cause problems. The following constraints force p_{im} to take on the right 0 or 1 value:

$$t_{im} - t_{Ei(m-1)} \geq [p_{im} - 1]T, \quad \text{for every } (i, m) \in S_N, \qquad \text{(C13)}$$

$$[t_{im} - t_{Ei(m-1)}] \leq T p_{im}, \quad \text{for every } (i, m) \in S_N, \qquad \text{(C14)}$$

We only need constraints for two ships because, by assumption, the ships will have products serviced consecutively in the order they arrive. The above constraints enforce p_{im} to be equal to 0 if $[t_{im} - t_{Ei(m-1)}] < 0$ (overlapping) and equal to 1 if $[t_{im} - t_{Ei(m-1)}] \geq 0$. Appealing to well-known results from global optimization theory, *equivalent* linear representations of nonlinear constraints (3) are presented in Section 2.5.

2.4.4.4 Stock Level Bounds

At any position (i, m), the stock level of product k should be within the prescribed levels at the beginning and ending of service. Thus,

$$S_{MNik} \leq s_{imk} \leq S_{MXik}, \quad \text{for every } (i, m, k) \in S_T \times K_i^H, \tag{C15}$$

$$S_{MNik} \leq s_{Eimk} + J_{ik} R_{ik} (T - t_{Eim})(y_{i(m+1)} - y_{im}) \leq S_{MXik}, \tag{4}$$
$$\text{for every } (i, m, k) \in S_T \times K_i^H.$$

Constraint (4) considers the stock level of product k not only at the end of each service but also at the end of the planning horizon. It has the term $(y_{i(m+1)} - y_{im})$ which is 1 if (i, m) is the last position for harbor i; otherwise, 0. Recall that $y_{im} = 1$ if position (i, m) is not visited; otherwise, 0. Therefore, the term $R_{ik}(T - t_{Eim})(y_{i(m+1)} - y_{im})$ is only activated when (i, m) is the last position for harbor i. Equivalent linear representations of nonlinear constraints (4) are also presented in Section 2.5. Notice that $y_{im} = 1$ implies $y_{i(m+1)} = 1$ because of the arrival sequence constraint (C5).

2.4.5 Objective Function

The objective of our ship routing and scheduling model is to minimize total operating costs over the planning horizon. The key cost components are the traveling costs, which include fuel and ship operating costs, and the loading/unloading costs, which include port operations, duties, etc. The parameter C_{ijv} denotes the total traveling cost for a ship v from harbor i to harbor j, and C_{Wik} is the fixed cost of loading or unloading product k at harbor i. The cost function of the problem can then be expressed as follows:

$$\sum_{v \in V} \sum_{(i,m,j,n) \in A_v} C_{ijv} x_{imjnv} + \sum_{(i,m) \in S_T} \sum_{v \in V} \sum_{k \in K_v} C_{Wik} o_{imvk}. \tag{O}$$

The optimization model for our problem is to find $(x, y, z, l, q, o, t, t_E, s, s_E, p)$ that minimize (O) subject to linear constraints (C1) through (C15) and nonlinear constraints (1) through (4) as well as variable bounds on (l, q, t, t_E, s, s_E) and binary integrality restrictions on (x, y, z, o, p). We will define equivalent linear representations for constraints (1) through (4) to yield a mixed-integer linear programming formulation for our model.

2.5 Linear Reformulation

In this section, we linearize the nonlinear terms and reformulate the problem into an *equivalent* mixed-integer linear program.

2.5.1 Linearizing Ship Load Constraints

The feasible region defined by ship load constraints (1) has the following general nonlinear structure:

$$\{(x, y) \mid xf(y) = 0,\ x \in \{0, 1\},\ y \in \mathbb{Y}\}, \tag{5}$$

where $f(\cdot)$ is a function with domain $\mathbb{Y}$. Specifically, setting $x := x_{imjnv}$, $y := (l_{imvk}, l_{jnvk}, q_{jnvk})$, and $f(y) := l_{imvk} + J_{jk}q_{jnvk} - l_{jnvk}$ in (15) yields constraint (1).

The constraint set given by (15) has a simpler characterization. First we need the following result.

Proposition 2.5.1. *Consider the set $S := \{(x, y) \mid xf(y) = 0, x \in \{0, 1\}, y \in \mathbb{Y}\}$, where $\{f(y) \mid y \in \mathbb{Y}\}$ is compact; i.e, there exist bounds $[L, U]$ such that $L \leq f(y) \leq U$ for all $y \in \mathbb{Y}$. Then, set S is equivalent to:*

$$S' := \{(x, y) \mid L(1 - x) \leq f(y) \leq U(1 - x),\ x \in \{0, 1\},\ y \in \mathbb{Y}\}.$$

Proof. The proof of the above result is straightforward and omitted. □

For constraint (1), $f(y) := l_{imvk} + J_{jk}q_{jnvk} - l_{jnvk}$ is linear and $-CAP_{vk}$ and CAP_{vk} are valid lower and upper bounds. Using Proposition 2.5.1, we can then replace (1) with the equivalent linear constraints:

$$l_{imvk} + J_{jk}q_{jnvk} - l_{jnvk} + CAP_{vk}x_{imjnv} \leq CAP_{vk}, \tag{C16}$$

$$\text{for every } v \in V, \text{ and every } (i, m, j, n, k) \in A_v \times K_v,$$

$$l_{imvk} + J_{jk}q_{jnvk} - l_{jnvk} - CAP_{vk}x_{imjnv} \geq -CAP_{vk}, \tag{C17}$$

$$\text{for every } v \in V, \text{ and every } (i, m, j, n, k) \in A_v \times K_v.$$

2.5.2 Linearizing Route and Schedule Compatibility Constraints

Note that the route and schedule compatibility constraints (2) also have the same structure as (15). Here, setting $x := x_{imjnv}$, $y := (t_{Eim}, t_{jn})$, and $f(y) := t_{Eim} + T_{ijv} - t_{jn}$ in (15) gives constraint (2). In this case the upper bound on $f(y)$ is $2T$. Notice that (2) are inequality constraints. Using Proposition 2.5.1, we can replace (2) with the equivalent linear constraint:

$$t_{Eim} + T_{ijv} - t_{jn} + 2Tx_{imjnv} \leq 2T, \tag{C18}$$

$$\text{for every } v \in V, \text{ and every } (i, m, j, n) \in A_v.$$

2.5.3 Linearizing Stock Level Constraints

The stock level constraints (3) given by:

$$s_{Ei(m-1)k} + J_{ik}R_{ik}[t_{im} - t_{Ei(m-1)}]p_{im} = s_{imk}, \quad \text{for every } (i, m, k) \in S_N \times K_i^H$$

are linearized using the convex envelope of bilinear forms (see Al-Khayyal and Falk [2], Al-Khayyal [1], Sherali and Alameddine [50], Sherali [48], and Tawarmalani and Sahinidis [52]).

The linearization process is accomplished in the following way. First, derive bounds for $(t_{im} - t_{Ei(m-1)})$. Noting that either a service time or the time between a departure and the next arrival can be as large as the entire planning horizon with the other quantity being small, we conclude that $-T \leq (t_{im} - t_{Ei(m-1)}) \leq T$. Next introduce a new variable w_{im} in place of $[t_{im} - t_{Ei(m-1)}]p_{im}$, and replace (3) by the linear system of equations and inequalities (cf., [2], [1]):

$$s_{Ei(m-1)k} + J_{ik}R_{ik}w_{im} = s_{imk}, \qquad \text{for every } (i,m,k) \in S_N \times K_i^H, \tag{6}$$

$$w_{im} \geq -Tp_{im}, \qquad \text{for every } (i,m) \in S_N, \tag{7}$$

$$w_{im} \geq t_{im} - t_{Ei(m-1)} + Tp_{im} - T, \qquad \text{for every } (i,m) \in S_N, \tag{8}$$

$$w_{im} \leq t_{im} - t_{Ei(m-1)} - Tp_{im} + T, \qquad \text{for every } (i,m) \in S_N, \tag{9}$$

$$w_{im} \leq Tp_{im}, \qquad \text{for every } (i,m) \in S_N, \tag{10}$$

$$p_{im} \in \{0,1\}, \qquad \text{for every } (i,m) \in S_N. \tag{11}$$

Remark. The projection of the set defined by (6) through (11) onto the vector space determined by constraints (3) is a polyhedral outer approximation of the constraint region (3). This result follows from Proposition A.3.2 by taking $x := p_{im}$ and $f(y) := t_{im} - t_{Ei(m-1)}$ together with $[L,U] = [-T,T], \{l,u\} = \{0,1\}$ and $[a,b] = [-\infty,\infty]$. While (6) through (11) represent a polyhedral relaxation of (3), we show in Theorem 1 that, under optimization, our reformulation is exact; i.e., the optimal solution with linear constraints (6) through (11) is also optimal for the nonlinear model having constraints (3).

Alternatively, instead of linearizing $[t_{im} - t_{Ei(m-1)}]p_{im}$ with one variable w_{im}, we can consider linearizing the two terms $t_{im}p_{im}$ and $t_{Ei(m-1)}p_{im}$ separately by introducing two sets of variables, w^1_{im} and w^2_{im}, respectively, in the following way. Both t_{im} and t_{Eim} are bounded below by 0 and above by T. Using these bounds, analogous to (6) through (11), we can replace (3) by the system of linear equations and inequalities:

$$s_{Ei(m-1)k} + J_{ik}R_{ik}[w^1_{im} - w^2_{im}] = s_{imk}, \qquad \text{for every } (i,m,k) \in S_N \times K_i^H \tag{C19.a}$$

$$w^1_{im} \geq 0, \qquad \text{for every } (i,m) \in S_N, \tag{C19.b}$$

$$w^1_{im} \geq t_{im} + Tp_{im} - T, \qquad \text{for every } (i,m) \in S_N, \tag{C19.c}$$

$$w^1_{im} \leq t_{im}, \qquad \text{for every } (i,m) \in S_N, \tag{C19.d}$$

$$w^1_{im} \leq Tp_{im}, \qquad \text{for every } (i,m) \in S_N, \tag{C19.e}$$

$$w^2_{im} \geq 0, \qquad \text{for every } (i,m) \in S_N, \tag{C19.f}$$

$$w^2_{im} \geq t_{Ei(m-1)} + Tp_{im} - T, \qquad \text{for every } (i,m) \in S_N, \tag{C19.g}$$

$$w^2_{im} \leq t_{Ei(m-1)}, \qquad \text{for every } (i,m) \in S_N, \tag{C19.h}$$

$$w^2_{im} \leq Tp_{im}, \qquad \text{for every } (i,m) \in S_N. \tag{C19.i}$$

$$p_{im} \in \{0,1\}, \qquad \text{for every } (i,m) \in S_N. \tag{C19.j}$$

Analogous to the first alternative, applying Proposition A.3.2 twice to (3) yields the linear relaxations (c1.a) through (C19.j) which are exact under optimization by Theorem 1.

By virtue of Proposition 2.5.2, the reformulation obtained by linearizing terms $t_{im}p_{im}$ and $t_{Ei(m-1)}p_{im}$ separately using two variables is tighter than that obtained by linearizing $[t_{im} - t_{Ei(m-1)}]p_{im}$ using a single variable. While the two reformulations are equivalent when the integrality restriction on p_{im} is included (c.f., Theorem 1), the tighter reformulation is preferable from a computational viewpoint when the integrality restriction is relaxed. Note that the two reformulations define feasible sets in higher dimensions than the region defined by (3). When comparing tightness of relaxations we will always be looking at the projection of each relaxation onto the space of original variables; i.e., the space defined by (3).

Denote the continuous relaxation of (C19.j) as ($\overline{\text{C19.j}}$) and (11) as ($\overline{11}$); i.e., both ($\overline{\text{C19.j}}$) and ($\overline{11}$) label the conditions $0 \leq p_{im} \leq 1$ for every $(i,m) \in S_N$.

Proposition 2.5.2. *For each point feasible to (c1.a) through (c1.i) and ($\overline{\text{C19.j}}$), there is a corresponding point feasible to the continuous relaxation of (6) through (10)and ($\overline{11}$).*

Proof. This result is stated and proved in a more general setting in Appendix A, Proposition A.3.6. □

Remark. Let S_1 be the projection of the set defined by (c1.a) through (c1.i) and ($\overline{\text{C19.j}}$) onto the vector space of the feasible set of the general model in Section 2.4 given by (C1) through (C15) and (1) through (4). Now let S_2 be the projection of the continuous relaxation of (6) through (10) and ($\overline{11}$) onto the same vector space. Then, by Proposition 2.5.2, we have $S_1 \subseteq S_2$; i.e., (C19) yields a tighter relaxation of (3).

We next show why our linear reformulations are exact under optimization. First, we need the following results.

Proposition 2.5.3. *Consider the nonlinear feasible region P_1, where $L \leq U$ and $l \leq u$, and the relaxation P_2 defined as*

$$\begin{aligned} P_1 := &\ \{\ (x, y) \mid a \leq xf(y) \leq b,\ L \leq f(y) \leq U,\ x \in \{l, u\}\ \} \\ P_2 := &\ \{\ (x, y, z) \mid a \leq z \leq b,\ L \leq f(y) \leq U,\ x \in \{l, u\}, \\ &\qquad\qquad\qquad z \geq lf(y) + Lx - Ll,\ z \geq uf(y) + Ux - Uu, \\ &\qquad\qquad\qquad z \leq uf(y) + Lx - Lu,\ z \leq lf(y) + Ux - Ul\ \}. \end{aligned}$$

If $(x, y, z) \in P_2$, then $z = xf(y)$ and $(x, y) \in P_1$.

Proof. There are two cases

Case 1. ($x = l$) We have from P_2

$$\begin{aligned} z &\geq lf(y) + Ll - Ll \Rightarrow z \geq lf(y), \\ z &\geq uf(y) + Ul - Uu \Rightarrow z - uf(y) \geq U(l - u) \\ z &\leq uf(y) + Ll - Lu \Rightarrow z - uf(y) \leq L(l - u) \\ z &\leq lf(y) + Ul - Ul \Rightarrow z \leq lf(y). \end{aligned}$$

Thus, $z = lf(y)$ and $U(l - u) \leq z - uf(y) \leq L(l - u) \Rightarrow L \leq f(y) \leq U$ because $l \leq u$ and $z - uf(y) = f(y)(l - u)$.

Case 2. ($x = u$) We have from P_2

$$\begin{aligned} z &\geq lf(y) + Lu - Ll \Rightarrow z - lf(y) \geq L(u - l), \\ z &\geq uf(y) + Uu - Uu \Rightarrow z \geq uf(y), \\ z &\leq uf(y) + Lu - Lu \Rightarrow z \leq uf(y), \\ z &\leq lf(y) + Uu - Ul \Rightarrow z - lf(y) \leq U(u - l). \end{aligned}$$

Thus, $z = uf(y)$ and $L \leq f(y) \leq U$. Therefore, if $(x, y, z) \in P_2$, then $z = xf(y)$ and $(x, y) \in P_1$. $\square$

If $f(y)$ is discrete and x is continuous, we have the analogous statement.

Proposition 2.5.4. *For given $L \leq U$ and $l \leq u$, define the sets*

$$
\begin{aligned}
P_1' := \ & \{\ (x,y) \mid a \leq xf(y) \leq b,\ l \leq x \leq u,\ f(y) \in \{f(L), f(U)\}\ \} \\
P_2' := \ & \{\ (x,y,z) \mid a \leq z \leq b,\ l \leq x \leq u,\ f(y) \in \{f(L), f(U)\}, \\
& \qquad z \geq f(L)x + lf(y) - lf(L),\ z \geq f(U)x + uf(y) - uf(U), \\
& \qquad z \leq f(U)x + lf(y) - lf(U),\ z \leq f(L)x + uf(y) - uf(L)\ \}.
\end{aligned}
$$

If $(x, y, z) \in P_2'$, then $z = xf(y)$ and $(x, y) \in P_1'$.

We can now state the main result.

Theorem 1. *Let (P) denote an optimization problem which has terms $xf(y)$, where $(x, y) \in P_1 \cup P_1'$, and let (P_R) denote the corresponding relaxed problem obtained by replacing $xf(y)$ with z, P_1 with P_2, and P_1' with P_2'. Then the (x, y) component of the optimal solution of problem (P_R) is optimal for problem (P).*

Proof. Follows from Propositions A.3.2, and A.3.3. $\square$

Remark. The relaxation of (P) given by (P_R) is exact in the sense that it will always produce an optimal solution for (P). More precisely, if (x^*, y^*, z^*) solves (P_R), then (x^*, y^*) solves (P).

2.5.4 Linearizing Stock Level Bounds Constraints

The stock level constraints (4) can be rewritten as

$$
\begin{aligned}
& S_{MNik} \leq s_{Eimk} + J_{ik}R_{ik}T(y_{i(m+1)} - y_{im}) \\
& -J_{ik}R_{ik}t_{Eim}y_{i(m+1)} + J_{ik}R_{ik}t_{Eim}y_{im} \leq S_{MXik}, \qquad \text{for every } (i, m, k) \in S_T \times K_i^H.
\end{aligned}
$$

We shall linearize this constraint using the reformulation technique explained in Section 2.5.3. Using Proposition 2.5.1, we linearize the terms $t_{Eim}y_{i(m+1)}$ and $t_{Eim}y_{im}$ by introducing the two sets of variables u_{im}^1 and u_{im}^2, respectively, together with the bounds $0 \leq t_{im}, t_{Eim} \leq T$. It follows that constraints (4) are equivalent (under optimization) to the system of linear inequalities:

$$S_{MNik} \leq s_{Eimk} + J_{ik}R_{ik}(T)(y_{i(m+1)} - y_{im}) - J_{ik}R_{ik}u_{im}^1 + J_{ik}R_{ik}u_{im}^2 \leq S_{MXik},$$
$$\text{for every } (i,m,k) \in S_T \times K_i^H, \tag{C20.a}$$

$$u_{im}^1 \geq 0, \qquad \text{for every } (i,m) \in S_T, \tag{C20.b}$$

$$u_{im}^1 \geq t_{Eim} + y_{i(m+1)} - T, \qquad \text{for every } (i,m) \in S_T, \tag{C20.c}$$

$$u_{im}^1 \leq t_{Eim}, \qquad \text{for every } (i,m) \in S_T, \tag{C20.d}$$

$$u_{im}^1 \leq y_{i(m+1)}, \qquad \text{for every } (i,m) \in S_T, \tag{C20.e}$$

$$u_{im}^2 \geq 0, \qquad \text{for every } (i,m) \in S_T, \tag{C20.f}$$

$$u_{im}^2 \geq t_{Eim} + y_{im} - T, \qquad \text{for every } (i,m) \in S_T, \tag{C20.g}$$

$$u_{im}^2 \leq t_{Eim}, \qquad \text{for every } (i,m) \in S_T, \tag{C20.h}$$

$$u_{im}^2 \leq y_{im}, \qquad \text{for every } (i,m) \in S_T, \tag{C20.i}$$

$$y_{im} \in \{0,1\} \qquad \text{for every } (i,m) \in S_T. \tag{C20.j}$$

2.6 mixed-integer Linear Programming Formulation

Combining all linear reformulations of the nonlinear constraints (1) - (4) with the linear constraints (C1) - (C15) yields the mixed-integer linear program

$$\min_{\substack{(x,y,z,l,q,o,t,t_E \\ s,s_E,p,w^1,w^2,u^1,u^2)}} \quad \text{Objective function (O)}$$

$$\begin{aligned}
\text{subject to} \quad & \text{Constraints (C1) through (C20.j)}, \\
& t_{im} \leq T, \quad \text{for every } (i,m) \in S_T, \\
& t_{Eim} \leq T, \quad \text{for every } (i,m) \in S_T, \\
& l, q, s, s_E, w^1, w^2, u^1, u^2 \quad \text{nonnegative vectors} \\
& \qquad\qquad \text{(possibly with given upper bounds)}, \\
& x, y, z, o, p \quad \text{binary vectors.}
\end{aligned}$$

Below, we solve a small illustrative example and vary some model parameters to gain some insights into developing solution algorithms that exploit problem structure.

2.6.1 Example

Consider the case of 2 ships ($V = \{1, 2\}$) carrying 2 products ($K_v = \{1, 2\}, \forall v \in V$) between 3 harbors ($H_T = \{1, 2, 3\}$), with each harbor handling both products (i.e., $K_i^H = \{1, 2\}$ for $i = 1, 2, 3$). Furthermore, harbor 1 consumes product 1 ($J_{11} = -1$) and produces product 2 ($J_{12} = +1$), while harbors 2 and 3 both consume product 2 ($J_{i2} = -1$, for $i = 2, 3$) and produce product 1 ($J_{i1} = +1$ for $i = 2, 3$). We want to find the optimal ship routing for a 2 day planning horizon ($T = 2$). Assume that ship 1 is initially located in harbor 1 ($(i_1, m_1) = (1, 1)$) with capacities $CAP_{11} = 10$ and $CAP_{12} = 10$ for products 1 and 2, respectively. Further assume that ship 2 is initially located in harbor 3 ($(i_2, m_2) = (3, 1)$) with capacities $CAP_{21} = 10$ and $CAP_{22} = 25$ for products 1 and 2, respectively. Finally, assume that the compartments of both ships are initially empty ($Q_{vk} = 0,\ \forall v \in V, k \in K_v$). Initial inventory levels of all products ($IS_{ik},\ \forall i \in H_T, k \in K_i^H$) at each harbor are given in Table 4, while the production rates ($R_{ik},\ \forall i \in H_T, k \in K_i^H$) of all products at each harbor are listed in Table 5, where positive rates are production and negative rates are consumption. Assume that it takes 0.3 days to travel from one harbor to each of the

Table 4: Initial inventory levels IS_{ik} for product k in harbor i

IS_{11}	IS_{12}	IS_{21}	IS_{22}	IS_{31}	IS_{32}
10	15	5	15	10	15

Table 5: Daily rates R_{ik} for product k in harbor i

$J_{11}R_{11}$	$J_{12}R_{12}$	$J_{21}R_{21}$	$J_{22}R_{22}$	$J_{31}R_{31}$	$J_{32}R_{32}$
-10	20	5	-10	5	-15

others for each ship ($T_{ijv} = 0.3,\ \forall i, j \in H_T, i \neq j, v \in V$); however, the cost for traveling between the harbors is different for each ship. It costs \$1 for ship 1 to travel between any two harbors ($C_{ij1} = 1,\ \forall i, j \in H_T, i \neq j$) and it costs \$1.5 per trip to operate ship 2

$(C_{ij2} = 1.5,\ \forall i, j \in H_T, i \neq j)$. The unit cost of loading or unloading any product at any harbor is taken as \$0.5 $(C_{Wik} = 0.5,\ \forall i \in H_T, k \in K_i^H)$. The time it takes to service one unit of any product is assumed to be 0.01 days $(TQ_{ik} = 0.01,\ \forall i \in H_T, k \in K_i^H)$, and set up times are taken as 0 $(W_i = 0, \forall i \in H_T)$. Figure 11 shows a feasible route for this example,

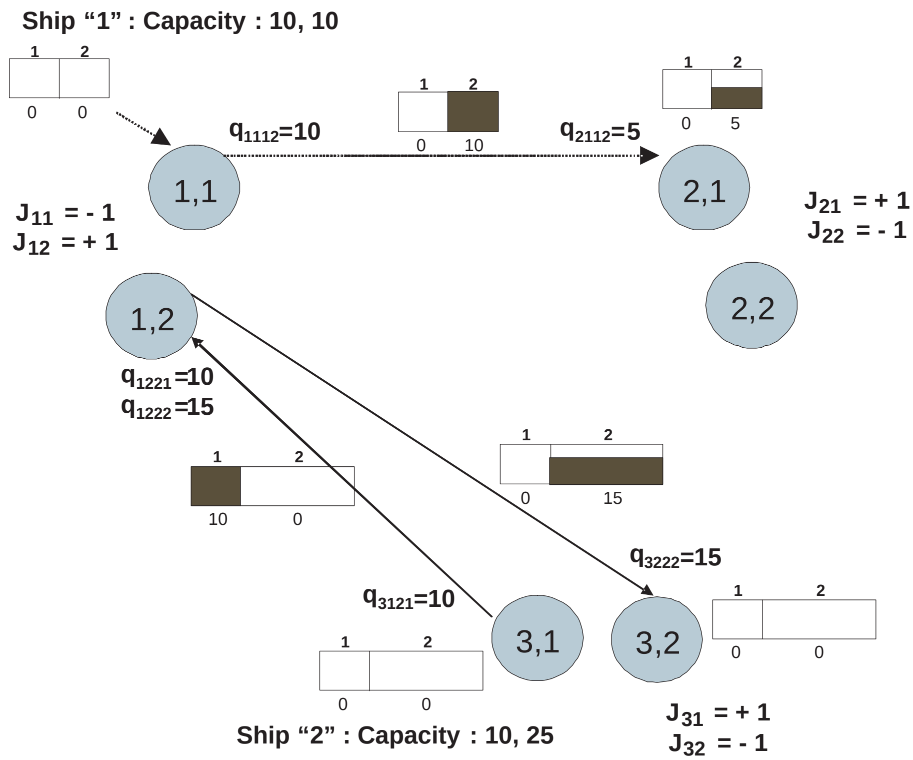

Figure 11: A feasible route

where each node represents a position (defined as pairs of harbor and arrival numbers).

The feasible schedule illustrated in Figure 11 determines the inventory levels over time of all products at each harbor, and this is displayed in Figure 12 for our planning horizon of 2 days. In each chart, the solid line represents the change of inventory level as ships load or unload, while the dashed line represents the change of inventory if no loading or unloading occurs. For example, the chart for product 1 in harbor 1 shows that the initial inventory level starts from 10, and is consumed at the rate of 10. If no ship arrives (represented by a dashed line) before time 1.0, the stock is depleted by the end of the first day. However, before time 0.5, our feasible solution has the first ship unloading 10 units of product 1 so

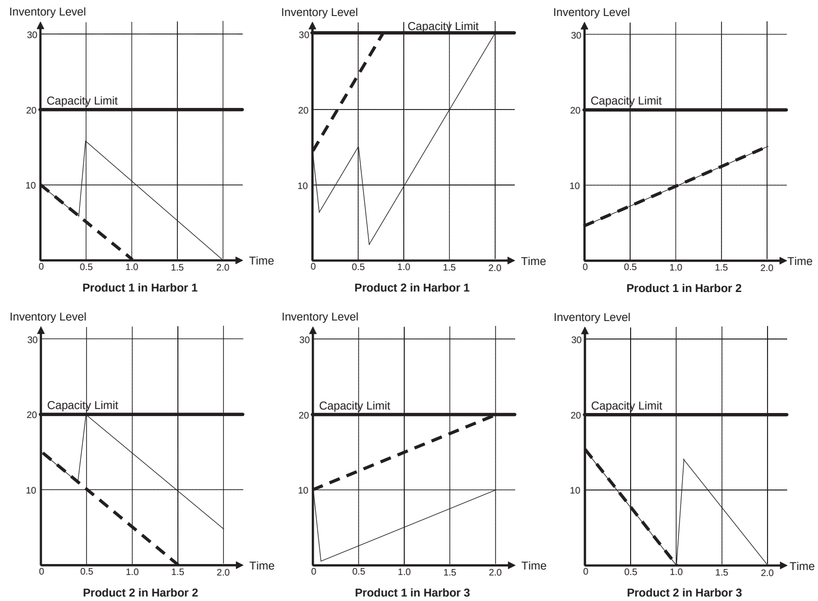

Figure 12: Examples of movement of inventory levels of products at harbors

that the stock level is maintained between its upper (capacity limit) and lower levels, 20 and 0, respectively, during the planning horizon.

Notice that the inventory level of product 1 in harbor 2 increases because our feasible solution does not call for loading this product by any ship during the planning horizon. As can be seen, inventory levels of all products in all ports are maintained between their upper and lower levels.

The mixed-integer linear program for this problem has 384 constraints and 155 variables including 61 binary variables. It is solved optimally by ILOG CPLEX 7.500 in 0.02 seconds on a four-CPU Sun E450 server machine using only the default options of the solver. The optimal solution turns out to be the feasible solution displayed in Figure 11. The total cost is \$7, consisting of \$4 for travel costs for the single trip of ship 1 and the two trips of ship 2, and \$3 for loading and unloading (ship 1 loads and unloads product 1, and ship 2 loads and unloads both products 1 and 2 for a total of six service calls costing \$0.5 each).

For the purpose of this illustrative example, we are not interested in devising the best CPLEX solution strategy. Rather, we seek to use the exact CPLEX solutions to uncover the problem parameters most sensitive to scaling, and to use this knowledge in developing a solution strategy for solving larger instances of the model. This will be presented in the next chapter.

2.6.2 Computing Time

Since it is not known *a priori* how many visits will be made to each harbor during a planning horizon, it is necessary to create enough positions (i, m) to allow as many visits as needed for an optimal solution. However, the number of binary variables and the number of constraints in the model grow exponentially with the number of positions (harbor-arrival pairs), while an insufficient number of positions may lead to an infeasible problem; i.e., a model with no feasible solutions. In this section, we first investigate the impact of the number of positions on computing time, and later show the relationship between the number of positions and the solution quality.

To measure the increase in solution time as a function of the number of positions in the model, a preliminary computational experiment was conducted on one-hundred randomly generated test problems. For our experiment, we chose ten different settings of the triple $(|H_T|, |V|, |K|)$, as listed in Table 6, and generated ten test problems for each setting.

Table 6: Ten test configurations (number of harbors $|H_T|$, ships $|V|$, and products $|K|$)

$(3, 2, 2)$	$(3, 2, 3)$	$(3, 3, 2)$	$(3, 3, 3)$	$(4, 2, 2)$
$(4, 2, 3)$	$(4, 3, 2)$	$(4, 3, 3)$	$(4, 4, 2)$	$(4, 4, 3)$

For our test problems the $|H_T|$ harbors are first randomly located on the plane within a box. For each $i \in H_T$, the location of port i, denoted by (a_i, b_i), is randomly generated by taking $a_i, b_i \sim \mathrm{U}[0, 10]$, where $\mathrm{U}[\alpha, \beta]$ denotes the uniform distribution over the interval $[\alpha, \beta]$. For simplicity, the distance between harbors i and j is assumed to be the Euclidian metric

$$\|(a_i, b_i) - (a_j, b_j)\| = \sqrt{(a_i - a_j)^2 + (b_i - b_j)^2}.$$

To differentiate between our vessels, we generate a weighting factor $w_v \sim \mathrm{U}[0.5, 1]$ for each v, that influences the travel cost and travel time. In particular, travel cost (C_{ijv}) for vessel

v is taken to be proportional to travel distance with constant of proportionality w_v. On the other hand, as travel costs go up, we would expect travel times (T_{ijv}) to go down. This is accomplished by taking simplifying assumption that travel time to be proportional to travel distance with constant of proportionality $1/w_v$. The values for each problem's parameters were generated in accordance with Table 7. The ranges for the uniform distributions and

Table 7: Generation of parameters for test problems

Parameter	Distribution / Value
T: Planning horizon	10
C_{ijv}: Cost of travel form port i to j by ship v	$w_v \times \sqrt{(a_i - a_j)^2 + (b_i - b_j)^2}$
T_{ijv}: Travel time between ports i and j by ship v	$T/5 + 0.4\sqrt{(a_i - a_j)^2 + (b_i - b_j)^2}/w_v$
CAP_{vk}: Capacity of product k on ship v	U$[20, 70]$
Q_{vk}: Initial quantity of product k on ship v	$CAP_{vk}\times$ U$[0, 1]$
C_{Wik}: Fixed cost to service product k in port i	U$[5, 10]$
J_{ik}: $+1$, if port i produces product k, -1, otherwise	either $+1$ or -1 with probability 1/2
R_{ik}: Rate of production or consumption of product k in port i	U$[1, 6]$
S_{MXik}: Minimum stock level of product k in port i	U$[20, 70]$
S_{MNik}: Maximum stock level of product k in port i	0
IS_{ik}: Initial stock level of product k in port i	$S_{MXik} \times$ U$[0.3, 0.7]$
TQ_{ik}: Time to load/unload product k in port i	U$[0, 0.03]$
W_i: Set-up time to change-over products in port i	U$[0, 0.1]$

scaling factors for travel times were selected to create nontrivial problems.

To complete the specification of our test problems, we need to fix the number of possible visits μ_i for harbor i. We need to choose μ_i large enough to admit an optimal solution, but not too large as to require long solution times.

To that end, we first determine the *minimum* number of visits to each harbor within the planning horizon. This minimum, m_i, for harbor i can be calculated by considering the length of the planning horizon (T), the maximum and the minimum stock levels (respectively, S_{MXik} and S_{MNik}) of product k in harbor i, the capacity (CAP_{vk}) of each product k on ship v, the initial inventory level (IS_{ik}) for each product k in harbor i, and the production/consumption rate (R_{ik}) of each product k in harbor i. For each harbor i that produces or consumes product k, this quantity is given by

$$m_i = \max_{k \in K_i^H} m_{ik},$$

where

$$m_{ik} = \begin{cases} \left\lceil \frac{T \times R_{ik} + (IS_{ik} - S_{MXik})}{max_{v \in V}\{CAP_{vk}\}} \right\rceil & \text{if } J_{ik} = +1 \\ \\ \left\lceil \frac{T \times R_{ik} + (S_{MNik} - IS_{ik})}{max_{v \in V}\{CAP_{vk}\}} \right\rceil & \text{if } J_{ik} = -1 \end{cases}$$

is the minimum number of loadings (if $J_{ik} = +1$) or unloadings (if $J_{ik} = -1$) of product k in harbor i within the planning horizon.

The minimum (un)loadings m_{ik} are determined based on the assumption that the ship with the largest capacity for product k is the only one visiting harbor i. So we calculate how many times it needs to visit based on rate R_{ik}, stock level IS_{ik}, maximum harbor capacity S_{MXik}, and minimum harbor capacity S_{MNik}. For $J_{ik} = +1$, we assume that the vessel with the largest capacity for product k loads at harbor i when the inventory level is at S_{MXik}. Starting from level IS_{ik}, it takes $(S_{MXik} - IS_{ik})/R_{ik}$ time units for the storage tanks to reach this level. With

$$T - \left(\frac{S_{MXik} - IS_{ik}}{R_{ik}} \right) = \frac{TR_{ik} + (IS_{ik} - S_{MXik})}{R_{ik}}$$

time units remaining in the planning horizon, we again assume that the largest capacity vessel reloads when inventory has reached S_{MXik}. The time it takes to reach that level, starting from $S_{MXik} - \max_{v \in V}\{CAP_{vk}\}$, is

$$\frac{\max_{v \in V}\{CAP_{vk}\}}{R_{ik}}.$$

Therefore, after the first visit, this ship will need to reload

$$\frac{\frac{TR_{ik} + (IS_{ik} - S_{MXik})}{R_{ik}}}{\frac{\max_{v \in V}\{CAP_{vk}\}}{R_{ik}}} = \frac{TR_{ik} + (IS_{ik} - S_{MXik})}{max_{v \in V}\{CAP_{vk}\}}$$

more times. By rounding up any fractional values for this quantity, we capture the first visit to yield the minimum number of visits m_{ik} for product k in harbor i. A similar argument holds for the case $J_{ik} = -1$ of harbors i consuming product k, except now the largest capacity ship is unloading instead of loading.

To the minimum number of visits m_i, we add $m' \in \{1, 2, 3, 4\}$. Thus, each harbor has $m_i + m'$ positions. Also, for each harbor i, we fix the variable $y_{in} = 0, \forall n \leq m_i$, so that the harbor is visited at least m_i times; otherwise, the problem would be infeasible.

Each test problem was solved four times by taking $m' \in \{1,2,3,4\}$ in order to observe the impact on solution time of growth in the number of positions in the model. The results for the different settings were similar. The case $(|H_T|, |V|, |K|) = (3,2,2)$ is depicted in Figure 13. The fitted curve was developed using exponential regression. The other cases

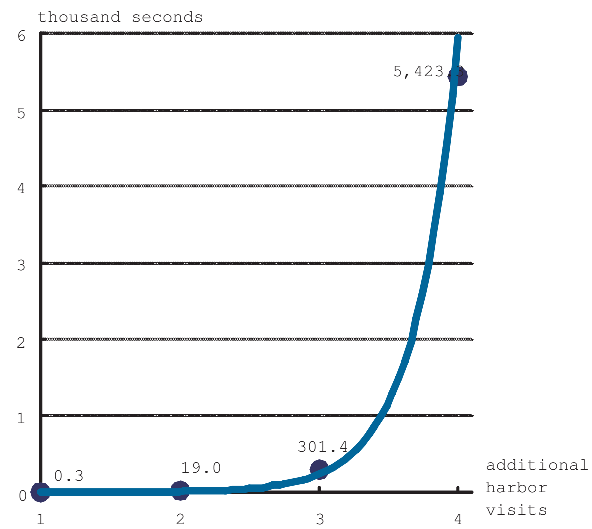

Figure 13: Average solution times for ten $(|H_T|, |V|, |K|) = (3,2,2)$ test problems as function of m'

are listed in Table 8 and shows similar exponential growth characteristics. These results

Table 8: Average computing times (seconds) for each configuration

Additional Harbor Visitis	Configurations					
	(3,2,3)	(3,3,2)	(3,3,3)	(4,2,2)	(4,2,3)	(4,3,2)
1	12.0	0.1	1.8	5.3	2.7	389.3
2	135.4	16.9	1535.3	205.8	491.5	9083.7
3	392.2	282.4	29579.4	1722.2	15812.4	>>15801.4†
4	953.0	2901.4	71675.8	14974.8	>>29328.4*	>>67257.1 ‡

clearly suggest that it is advantageous to limit the number of possible positions in the model for each port. The asterisk(*) indicates the average computing time for 4 problems with 6 problems not terminating before the time limit of 4.5E+5 seconds. The dagger(†) indicates the average computing time for 4 problems with 6 problems still running at time 4.5E+5 seconds. The double dagger(‡) indicates the average computing time for 2 problems with 8 problems exceeding the time limit of 4.5E+5 seconds.

In the foregoing, we constructed different test problems by varying the number of possible visits $\mu_i = m_i + m'$ for $m' \in \{1, 2, 3, 4\}$. While solution times are faster for smaller μ_i, it is possible for the problem to be infeasible if μ_i is too small for some i. This is illustrated in Table 9 which lists the results of test runs on four problems of different sizes taken from our one-hundred problem test bed.

Table 9: Optimal costs and computing times (seconds) as number of possible harbor visits increases

Additional Harbor Visits (m')	**Problems**							
	1 (3,2,3)		**2** (3,2,2)		**3** (3,3,2)		**4** (3,3,3)	
	Cost	Time	Cost	Time	Cost	Time	Cost	Time
1	inf	$< .01$	**36.2**	**0.1**	70.9	0.4	inf	$< .01$
2	**77.4**	**0.1**	36.2	0.6	**64.3**	**139.8**	**98.8**	**8938.7**
3	77.4	0.5	36.2	21.9	64.3	2480.8	**76.2**	**2.1E+5**
4	77.4	2.1	36.2	371.5	64.3	25284.4	**75.3**	**>4.5E+5**

For each problem, the objective cost and solution time is listed and the triple $(\cdot, \cdot, \cdot)$ denotes the problem settings $(|H_T|, |V|, |K|)$. While problems 2 and 3 only require $m' = 1$ to be feasible, that is not the case for problems 1 and 4. Moreover, problem 4 requires $m' \geq 4$ in order to find an optimal solution. We should point out that we are adding the same number of additional visits m' to each harbor, so it is conceivable that the times for problem 4 can be reduced by allowing the additive amount to vary; i.e., by taking $\mu_i = m_i + m'_i$ for $m'_i \in \{1, 2, 3, 4\}$.

2.7 Concluding Remarks

In this chapter we have developed a comprehensive mathematical model for planning the sailing routes and loading/unloading schedules for a fleet of ships carrying liquid bulk cargos across a network of harbors during a specified planning horizon. The objective is to minimize the sum of the travel costs and the fixed costs incurred when products are loaded or unloaded. More precisely, our model is to optimize (O) subject to constraints (C1) through (C20). The model differs from existing work in this area in that it considers ships with multiple compartments that are dedicated to carrying different cargo types. Furthermore, our model allows the simultaneous servicing of multiple ships at a harbor. We resolved some inherent nonlinearities in the problem by using some novel linearizing schemes from global optimization theory. We illustrated the model on a small example that was solved using a commercial solver for mixed-integer linear programming. Numerical experiments with this solver demonstrate the need for specialized algorithms that exploit the structure inherent in the model. In particular, exponential growth in the solution times as the number of harbor visits increases is not surprising. However, the results in Table 9 suggest that a solution scheme that starts with a small number of possible visits and selectively increases this quantity should lead to a robust procedure that can solve larger problems than currently possible. This will be the topic of the next chapter.

CHAPTER III

SOLUTION STRATEGY

3.1 Introduction

In this chapter, we first present a brief summary of the optimization model discussed in the previous chapter focusing on the structure of the problem. Using the structure of the problem, we decomposed the problem into ship sub-problems and harbor sub-problems by dualizing the coupling constraints. Ship sub-problems and harbor sub-problems are reduced to network flow problems and small sized integer programming problems, respectively.

We examine a method for solving large-scale linear integer programming problems via Lagrangian Relaxation. We introduce an iterative scheme to update the Lagrange multipliers in order to increase the lower bound on the problem. Because of the duality gap this method only produces a lower bound on the optimal objective value. We introduce two randomized greedy heuristic methods and use the lower bound obtained by Lagrangian relaxation to measure the goodness of the (primal feasible) heuristic solution.

3.1.1 Problem Assumptions

The motivating application is an oil company serving an archipelago of islands in Asia Pacific. The problem consists of a fleet of ships that delivers chemical products to terminals and direct customers nationwide in the Philippines. The Philippines consist of islands and it is cost-effective to distribute chemical commodities by ship. Each harbor (island) has storage tanks for specific commodities. Each harbor has its own production and consumption rate for a specific commodity and this determines the harbor as being either a *producing harbor* or a *consumption harbor* for that commodity. We have a heterogeneous fleet of vessels equipped with commodity dedicated multi-compartments. Our objective is to minimize the cost of operating ships to satisfy the stock level of each product in each harbor which must be sufficient to meet demand, and the stock level cannot exceed the inventory capacity of that harbor within the planning period. The assumptions of the models are:

- Operation
  1. Shipping multiple bulk commodities from producing harbors (exporting harbors) to the consuming harbors (importing harbors) defined by each commodity.
  2. Each ship starts and finishes its route at a harbor at the start and end of the planning period (i.e., ships cannot be at sea when the planning period begins and ends).
  3. Fixed planning period.
- The ships
  1. Heterogeneous types of ships in terms of size, number of compartments, available commodities, cost of operation, and speed.
  2. Commodity dedicated multi compartments for each ship.
  3. The location of the ship at the start of the planning period is known.
  4. Ship's keel may preclude entry to certain harbors.
- The harbors
  1. Known consumption and production rate for each commodity for each harbor.
  2. Known inventory level for each commodity at the start of the planning period.
  3. Multiple ships can load or unload at the same time at the same harbor.
  4. Set-up time needed for each commodity to be loaded or unloaded.
  5. Limited inventory capacity for each commodity.
  6. Navigable depth.
- The commodities
  1. Commodities can be loaded and/or unloaded partially by traveling through the harbors.

2. The amount of each commodity in each ship is known at the starting time of the planning period.

- The costs

1. Daily cost of the ships.

2. Bunker fuel.

3. Harbor and canal dues.

4. Loading and unloading charges.

3.1.2 Mathematical Model

The problem is to minimize the operating cost while satisfying four groups of constraints on binary flow through a network, ship loading and discharging, time restrictions, and inventory levels. We define the *state* of the transportation system as being specified by (i, m) where i is the physical harbor and m is the arrival number in that harbor i. We formulate this problem as a mixed-integer problem in which each state is indicated by a node and the arc flow variable is defined by x_{imjnv} which takes on the value 1 if the states (i, m) and (j, n) are directly connected by ship v, and the value 0 otherwise. The complete set of notation is defined in Appendix A. Also a detailed description is given in Chapter II.

Figure 14 shows an example of routes between states where two ships are visiting three harbors when ship 1 is located at harbor 1 and ship 2 is located at harbor 3 at the start time of the planning period. According to the definition of flow variable x_{imjnv}, Figure 14 indicates the specific arc flow variables; for example, $x_{11221} = 1$ and $x_{11211} = 0$. Notice that the state $(2, 3)$ is not visited by any ship. This is because we generate the possible set of arrivals for each harbor before we solve the problem.

3.1.2.1 Objective Function

We want to minimize the cost of operation. It consists of the traveling cost C_{ijv} incurred each time ship v moves between harbors i and j, and the loading and unloading cost C_{Wik} incurred if product k is loaded or unloaded at harbor i. Variable o_{imvk} is 1 if product k is

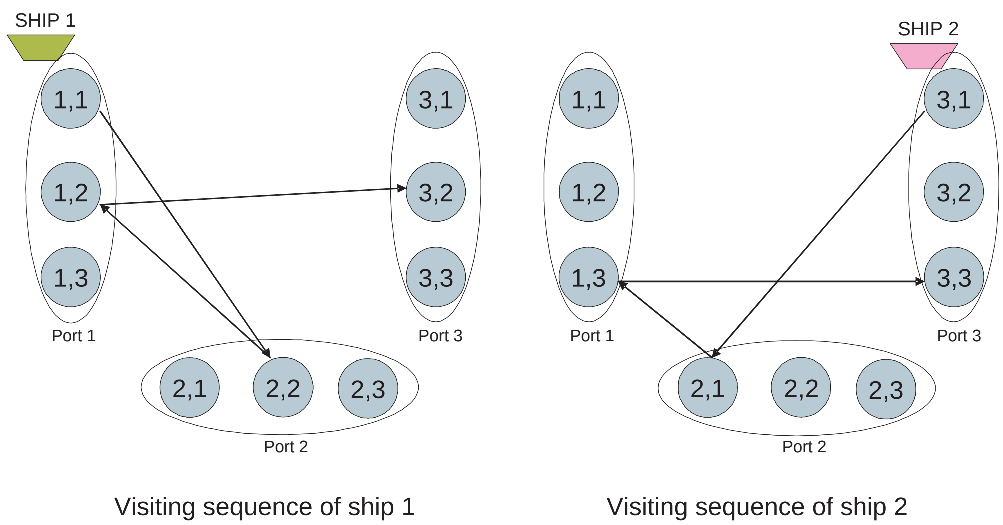

Figure 14: An example of a possible sequence of visits for two ships with 3 harbors.

serviced (loaded or unloaded) at state (i,m) by ship v. The objective is to minimize the function

$$\sum_{v\in V}\sum_{(i,m,j,n)\in A_v} C_{ijv}x_{imjnv} + \sum_{i\in H_T}\sum_{m\in M_i}\sum_{v\in V}\sum_{k\in K_v} C_{Wik}o_{imvk}. \tag{O}$$

Set V is the set of all ships indexed by v, H_T is the set of total harbors, M_i is the set of arrival numbers generated for harbor i, and K_v is the set of products that ship v can carry. The set A_v is the collection of all feasible arcs for ship v expressed as (i,m,j,n). According to the example in Figure 14, arc $[(1,1),(3,2)]$ is not an element of the set A_2 because ship 2 cannot start its route at state $(1,1)$ which is occupied by the starting state for ship 1. The set $A_v, \forall v \in V$, can be determined in the preprocessing step. If ship v cannot navigate harbor i for some physical reason, such as navigable depth of the harbor i, or ship v does not carry a commodity that harbor i handles, then we do not need to consider arcs whose head or tail is state $(i,m), \forall m \in M_i$, where M_i is the set of possible arrivals to harbor i.

3.1.2.2 Routing Constraints

This group of constraints keep track of the route of ships in the network formulation. Let $S_T := \{(i,m) : m \in M_i, \text{ for } i \in H_T\}$ be the total set of states (i,m) and $S_0 := \{(i_v,m_v) : v \in V\}$ be the set of initial states (i_v,m_v) of all ships $v \in V$, then define the set of non-initial states as $S_N := S_T \backslash S_0$. In the case when there are ρ_i ships starting at harbor i, then arbitrarily set the arrival sequence number $m_v \in \{1, 2, \ldots, \rho_i\}$ for each ship $v \in \{v_1, v_2, \ldots, v_{\rho_i}\}$. The routing constraints are

$$\sum_{(j,n)\in S_T} x_{jnimv} - \sum_{(j,n)\in S_N} x_{imjnv} - z_{imv} = 0, \quad \text{for every } (v,i,m) \in V \times S_T, \tag{12}$$

$$\sum_{(i,m)\in S_T} z_{imv} = 1, \quad \text{for each } v \in V, \tag{13}$$

$$\sum_{v\in V} \sum_{(j,n)\in S_T} x_{jnimv} + y_{im} = 1, \quad \text{for every } (i,m) \in S_T, \tag{14}$$

$$y_{im} - y_{i(m-1)} \geq 0, \quad \text{for every } (i,m) \in S_N. \tag{15}$$

Flow conservation constraints (12) ensure that the m-th arrival to harbor i should either leave harbor i or end its route there. Variable z_{imv} is equal to 1 if ship v ends its route at the state (i,m), otherwise 0. Therefore, **route finishing constraints** (13) ensure that any ship should finish its route at some state (i,m). In the example in Figure 14, $z_{321} = z_{332} = 1$ and the others are 0. **One time visit constraints** (14) ensure that every state should be visited at most once. Variable y_{im} is a binary variable which is 1 if the state (i,m) is **not** visited, otherwise 0, so that $y_{23} = 1$ and the others are all 0 for the example in Figure 14. By the **arrival sequence constraints** (15), we can specify how many times the harbor i is visited.

3.1.2.3 Constraints for Loading and Discharging

This group of constraints keep track of the amount onboard for each commodity. The set K_v is the set of products that ship v carries, while K_i^H is the set of products that harbor i handles. The constraints are

$$\begin{gathered} l_{imvk} + J_{jk} q_{jnvk} - l_{jnvk} + CAP_{v_k} x_{imjnv} \leq CAP_{v_k}, \\ \text{for every } v \in V, \text{ and every } (i,m,j,n,k) \in A_v \times K_v, \end{gathered} \tag{16}$$

$$\begin{gathered} l_{imvk} + J_{jk} q_{jnvk} - l_{jnvk} - CAP_{v_k} x_{imjnv} \geq -CAP_{v_k}, \\ \text{for every } v \in V, \text{ and every } (i,m,j,n,k) \in A_v \times K_v, \end{gathered} \tag{17}$$

$$Q_{vk} + J_{ik} q_{i_v m_v vk} - l_{i_v m_v vk} = 0, \quad \text{for each } v \in V \text{ and every } k \in K_v, \tag{18}$$

$$l_{imvk} \leq \sum_{(j,n) \in S_T} CAP_{v_k} x_{jnimv}, \quad \text{for each } v \in V \text{ and every} (k, i, m) \in K_v \times S_N, \tag{19}$$

$$q_{imvk} \leq CAP_{vk} o_{imvk}, \quad \text{for each } v \in V \text{ and every } (k, i, m) \in K_v \times S_T. \tag{20}$$

Ship load constraints (16) and (17) consider the case when ship v journeys from state (i, m) to state (j, n). Then l_{jnvk}, the amount of product k onboard of ship v after finishing service at state (j, n), will be equal to the sum of l_{imvk}, the amount of commodity k onboard before state (j, n) is serviced, and q_{jnvk}, the amount of product k loaded or unloaded at state (j, n) by ship v. Parameter J_{ik} is 1 if harbor i is producing product k, otherwise -1. **Initial Ship load constraints** (18) show that the amount $l_{i_v m_v vk}$ of product k onboard ship v at departure from the initial position (i_v, m_v) should be equal to the initial quantity Q_{vk} onboard plus, if $J_{i_v k} = +1$ (respectively, minus if $J_{i_v k} = -1$), the quantity $q_{i_v m_v vk}$ loaded (respectively, unloaded) at the initial state. **Compartment capacity constraints** (19) guarantee that the amount of each commodity onboard after servicing a state is less than or equal to the ship's compartment capacity for each commodity. **Servicing product constraints** (20) ensure that the quantity q_{imvk} of product k loaded onto ship v at position (i, m) cannot exceed the capacity CAP_{vk} of the compartment of ship v dedicated for product k.

3.1.2.4 Constraints for Time Aspects

This group of constraints are for the relationships between the service time and the travel time between harbors. Variables t_{im} and t_{Eim} represent the times to start and finish service, respectively, at state (i, m) during a planning horizon of length T. The time aspect constraints are

$$t_{im} - t_{i(m-1)} \geq 0, \quad \text{for every } (i, m) \in S_N, \tag{21}$$

$$t_{im} + \sum_{v \in V} \sum_{k \in K_v} TQ_{ik} q_{imvk} + W_i \sum_{v \in V} \sum_{k \in K_v} o_{imvk} - t_{Eim} = 0, \quad \text{for every } (i, m) \in S_T, \tag{22}$$

$$t_{Eim} + T_{Sijv} - t_{jn} + 2Tx_{imjnv} \leq 2T,$$
$$\text{for every } v \in V, \text{ and every } (i,m,j,n) \in A_v. \tag{23}$$

Service time sequence constraints (21) enforces the requirement that the m-th arrival should occur after the $(m-1)$-th arrival. **Service finishing time constraints** (22) identify the time to finish service at each state. At the state (i, m), service finishing time t_{Eim} equals service starting time t_{im} plus the time required to service the m-th ship in harbor i. The quantity TQ_{ik} is the time it takes to service a unit amount of product k at harbor i, and W_i is the set-up time to start service. It is assumed that the setup time is the same for any product at a harbor. **Route and schedule compatibility constraints** (23) check the service starting time at state (j, n). If ship v travels from position (i, m) to (j, n) — that is, $x_{imjnv} = 1$ — then the arrival time t_{jn} at (j, n) is the sum of the departure time t_{Eim} from (i, m) and the travel time T_{ijv} from harbor i to harbor j by ship v.

3.1.2.5 Constraints for the Inventories

This group of constraints ensure that the stock level is within the physical capacity limits of the harbor. The following variables are used: s_{imk} is the stock level of product k in harbor i at the time of the m-th arrival; s_{Eimk} is the stock level of product k in harbor i when the m-th ship departs; and p_{im} is a binary variable which is equal to zero if the m-th and $(m-1)$-th arrivals to harbor i overlap; i.e., when the m-th ship arrives before the $(m-1)$-th ship departs harbor i. The parameters used here are as follows: J_{ik} is set equal to $+1$ (respectively, -1) if harbor i is a producer (respectively, consumer) of product k; $R_{ik} > 0$ is the production (if $J_{ik} = +1$) or consumption (if $J_{ik} = -1$) *rate* of product k in harbor i; S_{MNik} is the minimum allowable stock level of product k at harbor i (safety stock); and S_{MXik} is the maximum allowable stock level of product k at harbor i (production/deliveries must stop when this level is reached). The inventory constraints are

$$s_{i1k} = IS_{ik} + J_{ik}R_{ik}t_{i1}, \quad \text{for every } (i,k) \in H_N \times K_i^H, \tag{24}$$

$$s_{imk} - \sum_{v \in V} J_{ik}q_{imvk} + J_{ik}R_{ik}(t_{Eim} - t_{im}) - s_{Eimk} = 0,$$
$$\text{for every } (i,m,k) \in S_T \times K_i^H, \tag{25}$$

$$t_{im} - t_{Ei(m-1)} \geq [p_{im} - 1]T, \quad \text{for every } (i,m) \in S_N, \tag{26}$$

$$[t_{im} - t_{Ei(m-1)}] \leq Tp_{im}, \quad \text{for every } (i,m) \in S_N, \tag{27}$$

$$s_{Ei(m-1)k} + J_{ik}R_{ik}[w^1_{im} - w^2_{im}] = s_{imk}, \quad \text{for every } (i,m,k) \in S_N \times K^H_i, \tag{c1.a}$$

$$w^1_{im} \geq 0, \quad \text{for every } (i,m) \in S_N, \tag{c1.b}$$

$$w^1_{im} \geq t_{im} + Tp_{im} - T, \quad \text{for every } (i,m) \in S_N, \tag{c1.c}$$

$$w^1_{im} \leq t_{im}, \quad \text{for every } (i,m) \in S_N, \tag{c1.d}$$

$$w^1_{im} \leq Tp_{im}, \quad \text{for every } (i,m) \in S_N, \tag{c1.e}$$

$$w^2_{im} \geq 0, \quad \text{for every } (i,m) \in S_N, \tag{c1.f}$$

$$w^2_{im} \geq t_{Ei(m-1)} + Tp_{im} - T, \quad \text{for every } (i,m) \in S_N, \tag{c1.g}$$

$$w^2_{im} \leq t_{Ei(m-1)}, \quad \text{for every } (i,m) \in S_N, \tag{c1.h}$$

$$w^2_{im} \leq Tp_{im}, \quad \text{for every } (i,m) \in S_N. \tag{c1.i}$$

$$S_{MNik} \leq s_{imk} \leq S_{MXik}, \quad \text{for every } (i,m,k) \in S_T \times K^H_i, \tag{28}$$

$$S_{MNik} \leq s_{Eimk} + J_{ik}R_{ik}T(y_{i(m+1)} - y_{im}) - J_{ik}R_{ik}(v^1_{im} - v^2_{im}) \leq S_{MXik}, \quad \text{for every } (i,m,k) \in S_T \times K^H_i, \tag{c2.a}$$

$$v^1_{im} \geq 0, \quad \text{for every } (i,m) \in S_T, \tag{c2.b}$$

$$v^1_{im} \geq t_{Eim} + y_{i(m+1)} - T, \quad \text{for every } (i,m) \in S_T, \tag{c2.c}$$

$$v^1_{im} \leq t_{Eim}, \quad \text{for every } (i,m) \in S_T, \tag{c2.d}$$

$$v^1_{im} \leq y_{i(m+1)}, \quad \text{for every } (i,m) \in S_T, \tag{c2.e}$$

$$v^2_{im} \geq 0, \quad \text{for every } (i,m) \in S_T, \tag{c2.f}$$

$$v^2_{im} \geq t_{Eim} + y_{im} - T, \quad \text{for every } (i,m) \in S_T, \tag{c2.g}$$

$$v^2_{im} \leq t_{Eim}, \quad \text{for every } (i,m) \in S_T, \tag{c2.h}$$

$$v^2_{im} \leq y_{im}, \quad \text{for every } (i,m) \in S_T. \tag{c2.i}$$

Initial inventory constraints (24) stipulate that the stock level s_{i1k} of product k in harbor i at the time of the first ship arrival is the amount IS_{ik} of product k in harbor i at the start of the planning horizon plus the amount produced, when $J_{ik} = +1$ (or minus the amount consumed when $J_{ik} = -1$), until the arrival t_{i1} of the first ship. The harbors have ships at the start of the planning horizon, $t_{i1} = 0$ so that $s_{i1k} = IS_{ik}$. **Inventory level constraints** (25) calculate the s_{Eimk}, the stock level of product k at the end of service at state (i, m). For product k in harbor i, if ship v is the m-th arrival, then the stock level s_{Eimk} equals the level s_{imk} before ship v arrives less the amount q_{imvk} loaded if $J_{ik} = +1$ (or plus the amount q_{imvk} unloaded if $J_{ik} = -1$) plus the amount produced (if $J_{ik} = +1$) while ship v is being loaded (or minus the amount consumed (when $J_{ik} = -1$) while ship v is unloading) at the rate R_{ik} during the time period $t_{Eim} - t_{im}$. **Stock level constraints** (c1.a)-(c1.i) represent an *equivalent* linearized formulation (see, Al-Khayyal and Hwang [**?**]) of the constraints

$$s_{Ei(m-1)k} + J_{ik}R_{ik}[t_{im} - t_{Ei(m-1)}]p_{im} = s_{imk}, \quad \text{for every } (i, m, k) \in S_N \times K_i^H$$

that ensure the stock levels of a product are consistent between successive arrivals to harbor i. Here, p_{im} is 0 if there are two or more ships in harbor i during the m-th arrival. Thus, if there are two ships, the above equation sets $s_{Ei(m-1)k} = s_{imk}$ so that overlapping does not cause conflicts. Constraints (26) and (27) force p_{im} to take right 0 or 1 value. **Stock level bounds constraints** (28) and (c2.a)-(c2.i) guarantee that the stock level of products should be between specified minimum and maximum stock levels at the beginning and end of service. Constraints (c2.a)-(c2.i) represent an *equivalent* linearized formulation (see, Chapter II) of the constraint

$$S_{MNik} \leq s_{Eimk} + R_{ik}(T - t_{Eim})(y_{i(m+1)} - y_{im}) \leq S_{MXik},$$
$$\text{for every } (i, m, k) \in S_T \times K_i^H.$$

If the stock level for a product at both service starting time and finishing time is within its bounds, then the stock level will be between the minimum and maximum stock levels during the entire planning period. This is because the stock level at the end of the m-th service and at the start of the $m+1$-th service for a specific harbor must always be within the minimum and maximum bounds under the assumption that the rate of consumption and production is constant.

3.1.2.6 Variables

The decision variables in the mixed-integer linear program are

$$\langle x_{imjnv}, o_{imvk}, z_{imv}, y_{im}, p_{im} \rangle \in \{0,1\}^{\eta}, \tag{29}$$

$$\langle q_{imvk}, l_{imvk}, t_{im}, t_{Eim}, s_{imk}, s_{Eimk}, w^r_{im}, v^r_{im} \rangle \in [\nu, \mu] \tag{30}$$

where η denotes the number of binary variables and $[\nu, \mu]$ denotes the hyper-rectangle defined by a lower bound vector ν and an upper bound vector μ for all continuous variables. (In (29) and (30), we use braces "$\langle$" and "$\rangle$" to represent a column vector with sub-vectors defined by all elements of one variable listed before the next variable is listed.) All of the continuous variables are bounded because they are related to the quantity of the products, and to time, which are physically bounded by the ship and inventory capacity and the planning period. Also, the integer variables are all binary so that every variable in our model is bounded. Therefore, our problem has a compact feasible set.

3.1.3 Size of the Problem

This problem has a very large number of constraints and variables. As the length of the planning period increases, both the number of constraints and the number of variables increase dramatically, because we need to consider many more possible visits that each harbor may process. This will increase the cardinality of the set M_i, the set of arrival numbers at harbor i. An example of 5 ships carrying 5 commodities to 10 harbors is illustrated in the Figure 15 as the arrival number is increased. It shows that the problem size, in terms of number of variables and constraints, increases exponentially when the planning period is expanded.

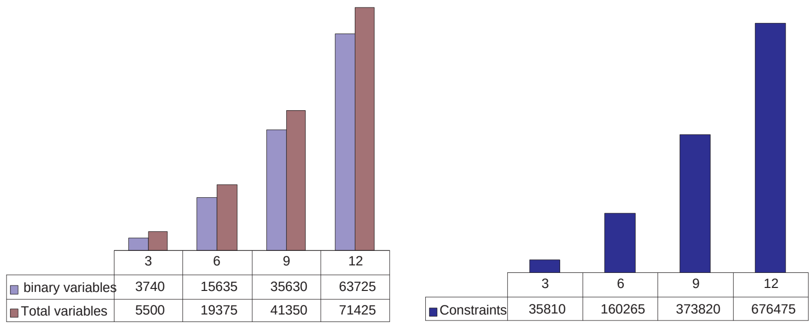

Figure 15: Number of variables and constraints when each port is restricted to 3 through 12 possible arrivals during the planning period.

3.2 Structure of the Problem

The constraints of the original problem can be decomposed into two polyhedra with coupling constraints. One polyhedron is defined by ship related constraints and the other one is defined by harbor related constraints. Each polyhedron has the special structure that allows it to be decoupled into several sub-polyhedra.

3.2.1 Decomposition of the Problem

In this chapter, we show how to decompose the constraints into three sets of constraints (namely, ship, harbor, and coupling constraints) by introducing new variables and constraints. The number of constraints in each polyhedron is still large. Specifically, the ship polyhedron is decoupled into $|V|$ sub-polyhedra, and the harbor polyhedron is decoupled into $|H|$ sub-polyhedra, where $|V|$ and $|H|$ are the number of ships and harbors, respectively.

Notice that constraints (12), (13), and (16)-(20) are defined for each ship $v \in V$ and all the variables in these constraints are defined for each ship v. We call these ship constraints. Constraints (15), (21), (24), (26), (27), (c1.a)-(c1.i), (28), and (c2.a)-(c2.i) are defined for each state (i, m) and variables in those constraints have index (i, m) without index v. Therefore, those constraints are related only to each state defined. The remaining constraints (14), (22), (23), and (25) define the relationship between ships and states. These constraints are composed of ship related variables (which have index v) and harbor related variables (which do not have index v). These are called coupling constraints.

Now define the new variable q_{imk} for the amount of product k transferred to the state (i, m). Also, let the binary variable o_{imk} be 0 if no product k transactions are made at the state (i, m); then

$$\sum_{v \in V} q_{imvk} = q_{imk}, \ \forall\ k \in K_i^H,\ (i, m) \in S_T, \tag{31}$$

$$\sum_{v \in V} o_{imvk} = o_{imk}, \ \forall\ k \in K_i^H,\ (i, m) \in S_T. \tag{32}$$

This is because the state (i, m) — corresponding to a node in our network formulation — can be visited at most once by one of the ships $v \in V$. By using these relationships, we can replace (22), and (25) with

$$t_{im} + \sum_{k \in K_i^H} TQ_{ik} q_{imk} + W_i \sum_{k \in K_i^H} o_{imk} - t_{Eim} = 0, \forall\ (i, m) \in S_T, \tag{33}$$

and

$$s_{imk} - J_{ik} q_{imk} + R_{ik} J_{ik} (t_{Eim} - t_{im}) - s_{Eimk} = 0, \forall\ k \in K_i^H,\ (i, m) \in S_T, \tag{34}$$

respectively.

This transformation allows the constraints (31) and (32) to be coupling constraints instead of (22) and (25). Then we have constraints (33) and (34) as harbor related constraints.

Here, the objective is to minimize the function (O) and the constraints are given by (12) through (30), with (22) and (25) replaced by (33) and (34), respectively, after adding constraints (31) and (32). Therefore, we have four types of coupling constraints (14), (23), (31) and (32) with ship constraints (12), (13), (16)-(20) and harbor constraints (15), (21), (24), (26), (27), (c1.a)-(c1.i), (28), (c2.a)-(c2.i), (33) and (34).

After augmenting the inequality constraints with slack and surplus variables, and relaxing the binary variables, the relaxation of the shipping model we want to solve has the general structure (See Table 15 in Appendix C)

$$
\begin{array}{llr}
(P) & \min\limits_{x,y} & cx \\
& \text{Subject to} & D_1x + D_2y = b^0 \\
& & F_1x \qquad\quad = b^1, \\
& & F_2y = b^2, \\
& & x \in [\nu_x, \mu_x], \\
& & y \in [\nu_y, \mu_y].
\end{array}
$$

where c is the objective coefficient row vector, and $[\nu_x, \mu_x]$ and $[\nu_y, \mu_y]$ denote hyper-rectangles defined by lower bound vectors ν_x and ν_y, and upper bound vectors μ_x and μ_y. Additionally, D_1, D_2, F_1, and F_2 are real matrices and x and y are column vectors where[1]

$$x := \langle x_{imjnv}, z_{imv}, o_{imvk}, q_{imvk}, l_{imvk}, t_{imv}, t_{Eimv} \rangle$$

$$y := \langle y_{im}, o_{imk}, p_{im}, q_{imk}, s_{imk}, s_{Eimk}, w^r_{im}, v^r_{im}, t_{im}, t_{Eim} \rangle$$

range over the values defined as follows: for each $v \in V$, $(i, m, j, n) \in A_v$, $k \in K_i^H \cap K_v$ and $(i, m) \in S_T$, and for each $k \in K_i^H$, position $(i, m) \in S_T$. Notice that all the elements of variable x have index v but y do not.

As noted above, even relaxing the binary variables to the interval $[0, 1]$ will yield a large-scale linear program with a technology matrix

$$\begin{bmatrix} D_1 & D_2 \\ F_1 & 0 \\ 0 & F_2 \end{bmatrix}$$

which is not totally-unimodular (Tables 15, 16 and 17 in Appendix C show the sparsity of

[1]Recall, the braces $\langle \alpha_{ij}, \beta_{kl} \rangle$ are to be read that all components of column vector α are listed before those of β.

the sub-matrices). The dimension of each component matrix can be expressed as follows;

$$\begin{aligned}
D_1 &: \quad o(|S_T|^2|V|) \times o(|S_T|^2|V|), \\
D_2 &: \quad o(|S_T|^2|V|) \times o(|S_T||K|), \\
F_1 &: \quad o(|S_T|^2|V||K|) \times o(|S_T|^2|V|), \\
F_2 &: \quad o(|S_T||K|) \times o(|S_T||K|)
\end{aligned}$$

where $|S_T|$, $|V|$, and $|K|$ are the cardinalities of the set of states, ships, and commodities, respectively.

We can rewrite problem (P) in concise form as

$$\begin{aligned}
\min_{x,y} \quad & cx \\
\text{s.t.} \quad & D_1 x + D_2 y = b^0 \\
& x \in \mathcal{P}^1, \\
& y \in \mathcal{P}^2.
\end{aligned}$$

Ship polyhedron $\mathcal{P}^1$ and *harbor polyhedron* $\mathcal{P}^2$ are defined as

$$\mathcal{P}^1 := \{x | F_1 x = b^1, x \in [\nu_x, \mu_x]\}$$

and

$$\mathcal{P}^2 := \{y | F_2 y = b^2, y \in [\nu_y, \nu_y]\}$$

The original vector x can be expressed as the partitioned vector[2] $x := (x_d, x_c)$, where the discrete (binary) vector is given by $x_d := \langle x_{imjnv}, z_{imv}, o_{imvk} \rangle$ and the nonnegative continuous vector is given by $x_c := \langle q_{imvk},\, l_{imvk}, t_{imv}, t_{Eimv} \rangle$. The ship polyhedron can be written as

$$\overline{\mathcal{P}}^1 := \{x | F_1 x = b^1, x_d \in \{0, 1\}, x_c \in [\nu_{x_c}, \mu_{x_c}]\}.$$

where $[\nu_{x_c}, \mu_{x_c}]$ denotes a hyper-rectangle defined by a lower bound vector ν_{x_c}, and an upper bound vector μ_{x_c} for all continuous variables x_c.

[2]For convenience, given column vectors a, b, c, we suppress the transpose superscript "T" and write $a = (b, c)$ instead of $a^T = (b^T, c^T)$ when there is no ambiguity.

Similarly, vector y can be expressed as $y := (y_d, y_c)$, where binary vector $y_d := \langle y_{im}, o_{imk}, p_{im} \rangle$, and nonnegative continuous vector $y_c := \langle q_{imk}, s_{imk}, s_{Eimk}, w^r_{im}, v^r_{im}, t_{im}, t_{Eim} \rangle$ are partitioned to give the harbor polyhedron

$$\overline{\mathcal{P}}^2 := \{y | F_2 y = b^2, y_d \in \{0,1\}, y_c \in [\nu_{y_c}, \mu_{y_c}]\},$$

where $[\nu_{y_c}, \mu_{y_c}]$ denotes the bounding hyper-rectangle for continuous vector y_c.

Polyhedron $\mathcal{P}^1$ can be decomposed into $|V|$ *ship-sub-polyhedra*, one for each ship $v \in V$ because every constraint is a restriction on a single ship. Table 16 in Appendix C shows the staircase structure of the matrix F'_1 defining polyhedron $\mathcal{P}^1$, after rearranging the rows and columns of the matrix F_1. Similarly, polyhedron $\mathcal{P}^2$ can be decomposed into $|H|$ *harbor-sub-polyhedra*, one for each harbor $i \in H$ because each constraint defining $\mathcal{P}^2$ involves only a single harbor. Table 17 in Appendix C shows the staircase structure of the matrix F'_2 defining polyhedron $\mathcal{P}^2$, obtained by rearranging the rows and columns of the matrix F_2. Therefore, we can rewrite $\mathcal{P}^1$ and $\mathcal{P}^2$ after rearranging x, y, b^1 and b^2 into $\hat{x}$, $\hat{y}$, $\hat{b}^1$ and $\hat{b}^2$, and defining the matrix decomposition

$$F'_1 = \begin{bmatrix} F_{11} & 0 & 0 & 0 \\ 0 & F_{12} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & F_{1|V|} \end{bmatrix} \text{ and } F'_2 = \begin{bmatrix} F_{21} & 0 & 0 & 0 \\ 0 & F_{22} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & F_{2|H|} \end{bmatrix},$$

as $\mathcal{P}^1 := \{\hat{x} | F'_1 \hat{x} = \hat{b}^1, \hat{x} \in [\hat{\nu}_x, \hat{\mu}_x]\}$ and $\mathcal{P}^2 := \{\hat{y} | F'_2 \hat{y} = \hat{b}^2, \hat{y} \in [\hat{\nu}_y, \hat{\mu}_y]\}$. Thus, the constraints in $\mathcal{P}^1$ and $\mathcal{P}^2$ can be written as

$$\begin{aligned} & F_{11} x^1 = b^{11}, \\ & F_{12} x^2 = b^{12}, \\ & \quad \vdots \\ & F_{1|V|} x^{|v|} = b^{1|V|}, \\ & x^v \in [\nu_{x^v}, \mu_{x^v}], \forall v \in V, \end{aligned}$$

and

$$
\begin{aligned}
&F_{21}y^1 = b^{21},\\
&F_{22}y^2 = b^{22},\\
&\quad\vdots\\
&F_{2|H|}y^{|H|} = b^{2|H|},\\
&y^i \in [\nu_{y^i}, \mu_{y^i}], \forall i \in H,
\end{aligned}
$$

respectively. Here for $v \in V$, each sub-vector x^v of $\hat{x} = (x^1, x^2, \dots, x^{|V|})$ consists of components

$$x^v := \langle x_{imjnv}, z_{imv}, o_{imvk}, q_{imvk}, l_{imvk}, t_{imv}, t_{Eimv}\rangle, \forall (i,m) \in S_T, (i,m,j,v) \in A_v, k \in K_i^H.$$

Similarly, for $i \in H$, sub-vector y^i of $\hat{y} := (y^1, y^2, \dots, y^{|H|})$ consists of components

$$y^i := \langle y_{im}, o_{imk}, p_{im}, q_{imk}, s_{imk}, s_{Eimk}, w^r_{im}, v^r_{im}, t_{im}, t_{Eim}\rangle, \forall (i,m) \in S_T.$$

Define sub-polyhedra

$$\mathcal{P}^1_v := \{x^v | F_{1v}x^v = b^{1v}, x^v \in [\nu_{x^v}, \mu_{x^v}]\}, \forall v \in V$$

and

$$\mathcal{P}^2_i := \{y^i | F_{2i}y^i = b^{2i}, y^i \in [\nu_{y^i}, \mu_{y^i}]\}, \forall i \in H.$$

Let D'_1 and D'_2 be the coupling matrices rearranged from D_1 and D_2 corresponding to the vectors $\hat{x}$ and $\hat{y}$, respectively, and let $\hat{b}^0$ and $\hat{c}$ be the vectors rearranged from b^0 and c, respectively. Then Problem (P) can be rewritten as

$$
\begin{aligned}
(P) \quad \min_{\hat{x},\hat{y}} \quad & \hat{c}\hat{x}\\
\text{s.t.} \quad & D'_1\hat{x} + D'_2\hat{y} = \hat{b}^0\\
& x^v \in \mathcal{P}^1_v, \qquad \forall v \in V\\
& y^i \in \mathcal{P}^2_i, \qquad \forall i \in H.
\end{aligned}
$$

Below we show that, for each $v \in V$, ship polyhedron $\mathcal{P}^1_v$ has network structure with side constraints. For each $i \in H$, the special property of harbor polyhedron $\bar{\mathcal{P}}^2_i$ is that it has a small number of binary variables $\langle y_{im}, o_{imk}, p_{im}\rangle$.

3.2.2 Structure of Ship Sub-Polyhedra

For each $v \in V$ in our model, vector x^v consists of both continuous and discrete variables. By partitioning x^v in a certain way, we can uncover hidden totally unimodular constraint sub-matrices.

The ship sub-polyhedron $\mathcal{P}_v^1$ is defined by constraints $F_{1v}x^v = b^{1v}, x^v \in [\nu_{x^v}, \mu_{x^v}]$ which can be partitioned as

$$F_{1v}x^v = \begin{bmatrix} F_{1v}^{11} & 0 \\ F_{1v}^{21} & F_{1v}^{22} \end{bmatrix} \begin{bmatrix} u^v \\ w^v \end{bmatrix} = \begin{bmatrix} b_1^{1v} \\ b_2^{1v} \end{bmatrix} = b^{1v}$$

where $u^v := \langle x_{imjnv}, z_{imv} \rangle$ and $w^v := \langle o_{imvk}, q_{imvk}, l_{imvk}, t_{imv}, t_{Eimv} \rangle$, and the matrix F_{1v}^{11} is totally unimodular.

Now we can rewrite the linear relaxation of our model as

$$\begin{aligned}
(P) \qquad z^* = \quad & \min_{u^v, w^v, y^i} && \sum_{v \in V} c_u^v u^v + \sum_{v \in V} c_w^v w^v && \\
& \text{Subject to} && \sum_{v \in V} (D_{1v}^1 u^v + D_{1v}^2 w^v) + \sum_{i \in H} D_{2i} y^i = b^0, && \\
& && F_{1v}^{11} u^v = b_1^{1v}, && \forall v \in V, \\
& && F_{1v}^{21} u^v + F_{1v}^{22} w^v = b_2^{1v}, && \forall v \in V, \\
& && F_2 y^i = b^{2i}, && \forall i \in H, \\
& && u^v \in [0,1], && \forall v \in V, \\
& && w^v \in [\nu_{w^v}, \mu_{w^v}], && \forall v \in V, \\
& && y^i \in [\nu_{y^i}, \mu_{y^i}], && \forall i \in H,
\end{aligned} \tag{35}$$

where c_u^v and c_w^v are partitioned sub-vectors of objective coefficient vector $c = (c_u, c_w)$ associated with sub-vector $x^v = (u^v, w^v)$. Accordingly, matrices D_{1v}^1, D_{1v}^2 and D_{2i} are partitioned in the following way

$$D_1'\hat{x} + D_2'\hat{y} = \hat{b}^0 =$$

$$[D_{11}^1, D_{11}^2, \ldots D_{1|V|}^1, D_{1|V|}^2] \begin{bmatrix} u^1 \\ w^1 \\ \vdots \\ u^{|V|} \\ w^{|V|} \end{bmatrix} + [D_{21}, \ldots, D_{2|H|}] \begin{bmatrix} y^1 \\ \vdots \\ y^{|H|} \end{bmatrix}.$$

3.2.3 Structure of Harbor-Sub-Polyhedra

For each $i \in H$, polyhedron $\overline{\mathcal{P}}_i^2 := \{y^i | F_{2i} y^i = b^{2i}, y_d^i \in \{0,1\}, y_c^i \in [\nu_{y_c^i}, \mu_{y_c^i}]\}$ has $o(|M_i||K_i^H|)$ binary variables $\langle y_{i'm}, o_{i'mk}, p_{i'm} \rangle$ where $|M_i|$ and $|K_i^H|$ are the cardinalities of the arrival numbers and products, respectively. Therefore, we can implement the binary branch and bound process to solve quickly the harbor sub-problem.

3.3 Solution by Lagrangian Relaxation Method

In this section we briefly summarize our solution strategy based on the Lagrangian Relaxation Method [33]. The foregoing showed how our relaxed model can be decomposed into $|V| + |H|$ ship and harbor polyhedra, each living in its own space, in addition to a set of coupling constraints. By dualizing the coupling constraints of our model, as well as the side constraints within all ship polyhedra, the Lagrangian dual problem decomposes into $|V| + |H|$ ship and harbor subproblems. Each subproblem has a nice structure that can be solved quickly.

3.3.1 Lagrangian Relaxation Problem

The general idea is to dualize the coupling constraints $\sum_{v \in V}(D_{1v}^1 u^v + D_{1v}^2 w^v) + \sum_{i \in H} D_{2i} y^i = b^0$, and the side constraints $F_{1v}^{21} u^v + F_{1v}^{22} w^v = b_2^{1v}, \forall v \in V$; thereby, yielding a Lagrangian relaxation which is decoupled in u, w and y. Recall, $x = (u, w) \in \mathcal{P}^1$ and $y \in \mathcal{P}^2$, where $u = (u^1, \ldots, u^{|V|})$ and $w = (w^1, \ldots, w^{|V|})$. It is convenient to rewrite problem (P) in the more compact notation

$$
\begin{array}{llll}
(P) & z^* = \min\limits_{u,w,y} & c_u u + c_w w \\
& \text{s.t.} & u \in \Upsilon, \\
& & y \in \mathcal{P}^2, \\
& & (u,w,y) \in C, \\
& & w \in [\nu_w, \mu_w],
\end{array}
$$

where the set $C := \{(u,w,y) | \sum_{v\in V}(D^1_{1v}u^v + D^2_{1v}w^v) + \sum_{i\in H} D_{2i}y^i = b^0, F^{21}_{1v}u^v + F^{22}_{1v}w^v = b^{1v}_2, \forall v \in V\}$ denotes the hyperplane defined by coupling constraints (14), (23), (31), and (32) (see Table 15 in Appendix C), and side constraints (16) - (20) (see Table 16 in Appendix C). Set $\Upsilon := \Upsilon^1 \times \cdots \times \Upsilon^{|V|}$, where $\Upsilon^v := \{u^v | F^{11}_{1v}u^v = b^{1v}_1, u^v \in [\nu_{u^v}, \mu_{u^v}]\}, \forall v \in V$.

For notational simplicity, let us rewrite the sets $C := \{u | A_1 u + A_2 w + A_3 y = a^0\}$ and $\Upsilon := \{u | Eu = a^1, u \in [0,1]\}$ and $\mathcal{P}^2 := \{y | F_2 y = b^2, y \in [\nu_y, \mu_y]\}$. Then problem (P) has the structure

$$
\begin{array}{llll}
(P) & z^* = \min\limits_{u,w,y} & c_u u + c_w w \\
& \text{Subject to} & A_1 u + A_2 w + A_3 y = a^0, \\
& & Eu \qquad\qquad\quad = a^1, \\
& & \qquad\qquad F_2 y = b^2, \\
& & u \in [0,1], w \in [\nu_w, \mu_w], y \in [\nu_y, \mu_y],
\end{array}
$$

where $a^0 = (b^0, b^1_2)$ for $b^1_2 = \langle b^{1v}_2 \rangle_{v\in V}$, $a^1 = b^1_1$ for $b^1_1 = \langle b^{1v}_1 \rangle_{v\in V}$, and $b^2 = \langle b^{2i} \rangle_{i\in H}$.

By dualizing the coupling and side constraints $(u,w,y) \in C$ with Lagrange multiplier row vector λ, we can compute, for each fixed value of $\lambda \in \mathbb{R}^\tau$, the Lagrangian dual objective function by solving the linear program

$$
\begin{array}{lll}
L(\lambda) = \lambda a^0 + \min\limits_{u,w,y} & (c_u - \lambda A_1)u + (c_w - \lambda A_2)w - \lambda A_3 y \\
\text{s.t.} & Eu \qquad\qquad\qquad\qquad = a^1, \\
& \qquad\qquad\qquad\qquad F_2 y = b^2, \\
& u \in [0,1], w \in [\nu_w, \mu_w], y \in [\nu_y, \mu_y].
\end{array}
\tag{36}
$$

where τ is the number of rows of A_1. This problem can be decomposed as $L(\lambda) = \lambda a^0 + S(\lambda) + H(\lambda)$, where

$$
\begin{aligned}
S(\lambda) = \min_{u,w} \quad & (c_u - \lambda A_1)u + (c_w - \lambda A_2)w \\
\text{s.t.} \quad & Eu = a^1, \\
& u \in [0,1], \quad w \in [\nu_w, \mu_w],
\end{aligned} \tag{37}
$$

and

$$
\begin{aligned}
H(\lambda) = \min_{y} \quad & -\lambda A_3 y \\
\text{s.t.} \quad & F_2 y = b^2, \\
& y \in [\nu_y, \mu_y].
\end{aligned} \tag{38}
$$

In the notation of this section, our original problem (before relaxation) is the mixed-integer linear program (MILP)

$$
\begin{aligned}
(IP) \qquad z^* = \min_{u,w,y} \quad & c_u u + c_w w \\
\text{Subject to} \quad & A_1 u + A_2 w + A_3 y = a^0, \\
& Eu = a^1, \\
& F_2 y = b^2, \\
& u, w_d, y_d \in \{0,1\}, \\
& w_c \in [\nu_{w_c}, \mu_{w_c}], y_c \in [\nu_{y_c}, \mu_{y_c}].
\end{aligned}
$$

Imposing the binary restriction on the Lagrangian dual $L(\lambda)$ gives another (in fact tighter) Lagrangian dual objective as the solution of the MILP

$$
\begin{aligned}
L_{IP}(\lambda) = \lambda a^0 + \min_{u,w,y} \quad & (c_u - \lambda A_1)u + (c_w - \lambda A_2)w - \lambda A_3 y \\
\text{s.t.} \quad & Eu = a^1, \\
& F_2 y = b^2, \\
& u, w_d, y_d \in \{0,1\}, \\
& w_c \in [\nu_{w_c}, \mu_{w_c}], y_c \in [\nu_{y_c}, \mu_{y_c}].
\end{aligned} \tag{39}
$$

While $L_{IP}(\lambda)$ is the optimal value of a parameterized mixed-integer linear program, we will later refer to the above MILP as *problem* $L_{IP}(\lambda)$.

Problems $S(\lambda)$ and $H(\lambda)$ can be decoupled into smaller sized subproblems. As we have seen in Section 3.2, we can decouple each problem into several subproblems. Restricting (37) and (38) to binary variables gives,

$$
\begin{aligned}
S_{IP}(\lambda) = \min_{u,w} \quad & (c_u - \lambda A_1)u + (C_w - \lambda A_2)w \\
\text{s.t.} \quad & Eu = a^1, \\
& u, w_d \in \{0,1\}, w_c \in [\nu_{w_c}, \mu_{w_c}],
\end{aligned}
$$

and

$$
\begin{aligned}
H_{IP}(\lambda) = \min_{y} \quad & -\lambda A_3 y \\
\text{s.t.} \quad & F_2 y = b^2, \\
& y_d \in \{0,1\}, y_c \in [\nu_{y_c}, \mu_{y_c}],
\end{aligned}
$$

where vector $u := \langle x_{imjnv}, z_{imv} \rangle$ is binary and w is expressed as $w := (w_d, w_c)$ where the indices of discrete (binary) vector $w_d := \langle o_{imvk} \rangle$ and the nonnegative continuous vector $w_c := \langle q_{imvk}, l_{imvk} \rangle$. Vector y is expressed as $y := (y_d, y_c)$ where the indices of (binary) vector $y_d := \langle y_{im}, o_{imk}, p_{im} \rangle$ and the nonnegative vector $y_c := \langle s_{imk}, s_{Eimk}, q_{imk}, w^r_{im}, v^r_{im}, t_{im}, t_{Eim} \rangle$. In summary, $L_{IP}(\lambda) = \lambda a^0 + S_{IP}(\lambda) + H_{IP}(\lambda)$ and we want to find λ^* that solves the dual problem $\max_{\lambda \in \mathbb{R}^\tau} L_{IP}(\lambda)$.

It is well known [33] that if the solution (u^*, w^*, y^*) of $L_{IP}(\lambda)$ satisfies the dualized coupling and side constraints, then it is a solution of the original problem (IP); otherwise, we have found a better lower bound than that obtained from the LP relaxation of the original problem. That is, $\max_{\lambda \in \mathbb{R}^\tau} L_{IP}(\lambda) \geq \max_{\lambda \in \mathbb{R}^\tau} L(\lambda)$.

Let $\upsilon_1, \upsilon_2, \ldots, \upsilon_N$ denote the vertices of polyhedron

$$
\mathcal{Q} := \{(u, w, y) | u \in \Upsilon, y \in \mathcal{P}^2, w \in [\nu_w, \mu_w]\}.
$$

Suppose it is bounded, then $z = (u, w, y) \in \mathcal{Q}$ can be written as

$$
z = \alpha_1 \upsilon_1 + \alpha_2 \upsilon_2 + \cdots + \alpha_N \upsilon_N, \text{ where, } \sum_{i=1}^{N} \alpha_i = 1, \alpha_i \geq 0, \forall i \in \{1, 2, \ldots, N\}.
$$

We can rewrite

$$L(\lambda) = \min_{1 \le i \le N} \phi(\upsilon_i, \lambda),$$

where, $\phi(\upsilon_i, \lambda) = \lambda a^0 + [c_u - \lambda A_1, c_w - \lambda A_2 - \lambda A_3]\upsilon_i$. For fixed υ_i, the objective ϕ is a linear function of λ. As λ varies, the optimal vertex υ_i changes and this implies that $L(\lambda)$ is a piecewise linear concave function, since $L(\lambda)$ is the pointwise minimum of N linear functions.

Similarly, let $\bar{\upsilon}_1, \bar{\upsilon}_2, \ldots, \bar{\upsilon}_M$ denote the vertices of the convex hull of points in

$$\overline{\mathcal{Q}} := \{(u, w, y) | u \in \Upsilon, y \in \bar{\mathcal{P}}^2, w_d \in \{0, 1\}, w_c \in [\nu_{w_c}, \mu_{w_c}]\}$$

and write z as

$$z = \beta_1 \bar{\upsilon}_1 + \beta_2 \bar{\upsilon}_2 + \cdots + \beta_M \bar{\upsilon}_M, \text{ where } \sum_{i=1}^{M} \beta_i = 1, \beta_i \ge 0, \forall i \in \{1, 2, \ldots, M\}.$$

Then we can write

$$L_{IP}(\lambda) = \min_{1 \le i \le M} \phi(\bar{\upsilon}_i, \lambda). \tag{40}$$

Let $\bar{z}^*$ denote the optimal cost of the original problem (IP) and let its linear relaxation (P) have optimal objective value z^*. Let λ^* be an optimal solution of the dual $\max_\lambda L(\lambda)$ and let $\bar{\lambda}^*$ be an optimal solution to the dual $\max_\lambda L_{IP}(\lambda)$. Then, it is well known that

$$\bar{z}^* \ge L_{IP}(\bar{\lambda}^*) \ge L_{IP}(\lambda^*) \ge L(\lambda^*) = z^*.$$

The first inequality holds because an optimal solution to (IP) is feasible to $L_{IP}(\lambda)$ for all λ. The others readily follow by definition.

Figure 16 shows the piecewise linear concave function $L_{IP}(\lambda)$ and $L(\lambda)$.

3.3.2 Dual Ascent Method

Our objective is to find an optimal λ^* that maximizes $L_{IP}(\lambda)$. We applied the Dual Ascent method [8] to obtain λ^*. As we have seen, $L_{IP}(\lambda)$ is a piecewise linear concave function of λ. By using the basic property of concave functions, we verify optimality of the current λ^k at each iteration k. If the current λ^k is not optimal, we derive an improving direction, and choose the step size along the improving direction.

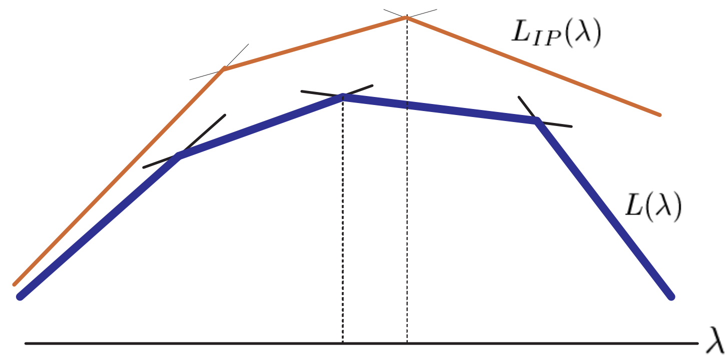

Figure 16: Domination of piecewise linear concave function $L_{IP}(\lambda)$ over $L(\lambda)$

3.3.2.1 Optimality Condition

For a smooth concave function f, which generally means a twice continuously differentiable, concave function, x^* is optimal if and only if its gradient vanishes at x^*. However, the optimal value function $L_{IP}(\lambda)$ is not differentiable at certain points as shown in Figure 16. If the minimum in (39) is *not* obtained *uniquely* for some $\lambda = \hat{\lambda}$, then $L_{IP}(\lambda)$ is not differentiable at $\hat{\lambda}$.

Let $V(\lambda) = \{\bar{v}_i | L_{IP}(\lambda) = \phi(\bar{v}_i, \lambda), i = 1, 2, \dots, M\}$ and let $I(\lambda) = \{i | L_{IP}(\lambda) = \phi(\bar{v}_i, \lambda), i = 1, 2, \dots, M\}$. Here, $V(\lambda)$ is the set of all extreme points that solve the MILP (40), and $I(\lambda)$ is the index set for the points in $V(\lambda)$. For $(u^i, w^i, y^i) \in V(\lambda)$, it can be shown (Chapter 6, [6]) that the vector $\sum_{i \in I(\lambda)} \alpha_i (a^0 - A_1 u^i - A_2 w^i - A_3 y^i) \in \partial L_{IP}(\lambda)$ for all $\alpha_i \geq 0$ such that $\sum_{i \in I(\lambda)} \alpha_i = 1$; that is, for every $(u^i, w^i, y^i) \in V(\lambda)$, the vector $a^0 - A_1 u^i - A_2 w^i - A_3 y^i$ and all convex combinations are subgradients of L_{IP} at λ, where $\partial L_{IP}(\lambda)$ denotes the subdifferential of L_{IP} at λ.

By the concave and piecewise linear properties of $L_{IP}(\lambda)$, the multiplier λ^* is an optimal solution to $\max_\lambda L_{IP}(\lambda)$ if and only if Lagrange multiplier λ^* satisfies the first order optimality condition $0 \in \partial L_{IP}(\lambda^*)$.

Next we present our development of computational methods for maximizing this piecewise linear concave function. For given λ, at each iteration the method finds a vertex in $V(\lambda)$ and checks for the optimality condition $0 \in \partial L_{IP}(\lambda)$. As more vertices of $V(\lambda)$ are found, eventually we will have enough to verify optimality or conclude λ is not optimal.

Note that $0 \in \partial L_{IP}(\lambda)$ can be determined without finding all vertices in $V(\lambda)$ if the right ones are found early. When λ is not optimal, the method determines an improving search direction which is followed for an optimal step size to fix the value of λ for the next iteration.

3.3.2.2 Improving Direction

At iteration k, we have for λ^k, a set of extreme points $V_k(\lambda^k) \subseteq V(\lambda^k)$ that solve $L_{IP}(\lambda^k)$, with associated index set $I_k(\lambda^k) \subseteq I(\lambda^k)$. Note that $|V(\lambda^k)| = |I(\lambda^k)| = 1$ if problem $L_{IP}(\lambda^k)$ has a *unique* solution.

We want to know if $0 \in \partial L_{IP}(\lambda^k)$. This is equivalent to verifying the existence of α_i such that

$$\sum_{i \in I_k(\lambda^k)} \alpha_i(a^0 - A_1u^i - A_2w^i - A_3y^i) = 0, \quad \sum_{i \in I_k(\lambda^k)} \alpha_i = 1, \ \alpha_i \geq 0, \forall i \in I_k(\lambda^k).$$

We can check the condition by solving the Phase I linear problem $< PH1(k) >$

$$\begin{aligned}
\min \quad & \sum_{j=1}^{\tau+1} s_j \\
\text{s.t.} \quad & (A_1u^1 + A_2w^1 + A_3y^1)\alpha_1 + \cdots + (A_1u^n + A_2w^n + A_3y^n)\alpha_n + Is = a^0, \\
& \alpha_1 + \cdots + \alpha_n + s_{\tau+1} = 1, \\
& \alpha_i \geq 0, \quad \forall i \in \{1, 2, \ldots, n\}, \\
& s_j \geq 0, \quad \forall j \in \{1, 2, \ldots, \tau + 1\},
\end{aligned} \tag{41}$$

where $n := |I_k(\lambda^k)|$ and τ is the number of rows of matrix A_1 (i.e., the number of coupling and side constraints that were dualized).

Iteration k

Solve $< PH1(k) >$. There are three possible outcomes.

Case I: optimal cost 0.

This means $0 \in \partial L_{IP}(\lambda^k)$. Therefore, λ^k is optimal to $\max_\lambda L_{IP}(\lambda)$

Case II: optimal cost is positive.

Then an improving search direction is found as follows.

Let (ρ^k, ρ_0^k) be dual optimal to $< PH1(k) >$, then,

$$\rho^k(A_1u^i + A_2w^i + A_3y^i) + \rho_0^k \le 0, \forall i \in I_k(\lambda^k). \tag{42}$$

$$\rho^k a^0 + \rho_0^k > 0 \Rightarrow -\rho^k a^0 < \rho_0^k. \tag{43}$$

By (42), and (43), we have

$$\rho^k(a^0 - A_1u^i - A_2w^i - A_3y^i) > 0, \forall i \in I_k(\lambda^k). \tag{44}$$

Thus, $\rho^k g^i > 0, \forall i \in I_k(\lambda^k)$, where $g^i := a^0 - A_1u^i - A_2w^i - A_3y^i$ is a subgradient of $L_{IP}(\lambda)$ at $\lambda = \lambda^k$.

Suppose that we have *all* of the optimal vertices for problem $L_{IP}(\lambda)$. If $\rho g^i > 0$ for all $i \in I(\lambda^k)$, then the direction ρ is an improving direction [6], because, for every convex combinations, say g, of $g^i, \forall i \in I(\lambda^k)$, we must have $\rho g > 0$. However, we only know that $\rho g^i > 0, \forall i \in I_k(\lambda^k) \subseteq I(\lambda^k)$.

Therefore, we want to check that the direction ρ^k is an improving direction for λ^k by verifying that the directional derivative of $L_{IP}(\lambda)$ at λ^k in direction ρ^k is positive; i.e.,

$$\lim_{t \downarrow 0} \frac{L_{IP}(\lambda^k + t\rho^k) - L_{IP}(\lambda^k)}{t} > 0. \tag{45}$$

In practice, we used small positive t to solve $L_{IP}(\lambda^k + t\rho^k)$ and verified that $L_{IP}(\lambda^k + t\rho^k) > L_{IP}(\lambda^k)$.

Now Case II has two subcases:

Case II-1: Condition (45) holds (ρ^k is an ascent direction).

Find t_k that solves the step size problem

$$\max_{t \ge 0} L_{IP}(\lambda^k + t\rho^k). \tag{46}$$

Define $\lambda^{k+1} = \lambda^k + t_k\rho^k$.

Let $(u^{k+1}, w^{k+1}, y^{k+1})$ be an optimal solution of MILP

$$\min\{(c_u - \lambda^{k+1}A_1)u + (c_w - \lambda^{k+1}A_2)w - \lambda^{k+1}A_3y | (u, w, y) \in \overline{\mathcal{Q}}\}. \tag{47}$$

Define $V_{k+1}(\lambda^{k+1}) = \{(u^{k+1}, w^{k+1}, y^{k+1})\}$.

Set $k \leftarrow k + 1$ and repeat iteration k.

Case II-2: Condition (45) does not hold (ρ^k is a non-ascent direction).

This case can only happen if $|V(\lambda^k)| > 1$. Then, a new optimal solution $(u^s, w^s, y^s) \notin V_k(\lambda^k)$ of $L_{IP}(\lambda^k)$ is found (See Proposition 3.3.1) by pivoting.

Update $V_k(\lambda^k) \leftarrow V_k(\lambda^k) \cup \{(u^s, w^s, y^s)\}$. This amounts to adding a new column

$$\begin{bmatrix} A_1 u^s + A_2 w^s + A_3 y^s \\ 1 \end{bmatrix}$$

to problem $< PH1(k) >$.

Return to the beginning of iteration k and solve problem $< PH1(k) >$.

The following result is well known (e.g., Theorem 6.3.4 in [6]). We include a statement and proof in our notation for completeness.

Proposition 3.3.1. *Suppose $L_{IP}(\lambda^k + t_s\rho^k) \leq L_{IP}(\lambda^k)$, for $t_s > 0$ sufficiently small and ρ^k is an optimal dual sub-vector associated with constraints (41) of problem $< PH1(k) >$. Let the vertex (u^s, w^s, y^s) be an optimal solution of $L_{IP}(\lambda^k + t_s\rho^k)$. Then $(u^s, w^s, y^s) \in V(\lambda^k)\backslash V_k(\lambda^k)$.*

Proof. We first show that $(u^s, w^s, y^s) \in V(\lambda^k)$. By contradiction, suppose $(u^s, w^s, y^s) \notin V(\lambda^k)$. For small $t_s > 0$ we are given that

$$\begin{aligned}
& L_{IP}(\lambda^k + t_s\rho^k) \\
= \;& (\lambda^k + t_s\rho^k)a^0 + (c_u - (\lambda^k + t_s\rho^k)A_1)u^s + (c_w - (\lambda^k + t_s\rho^k)A_2)w^s - (\lambda^k + t_s\rho^k)A_3 y^s \\
= \;& \lambda^k a^0 + (c_u - \lambda^k A_1)u^s + (c_w - \lambda^k A_2)w^s - \lambda^k A_3 y^s + t_s\rho^k(a^0 - A_1u^s - A_2w^s - A_3y^s) \\
\leq \;& L_{IP}(\lambda_k).
\end{aligned}$$

Moreover, $L_{IP}(\lambda_k) < \lambda^k a^0 + (c_u - \lambda^k A_1)u^s + (c_w - \lambda^k A_2)w^s$ by the assumption that $(u^s, w^s, y^s) \notin V(\lambda^k)$. Let

$$\delta := \lambda^k a^0 + (c_u - \lambda^k A_1)u^s + (c_w - \lambda^k A_2)w^s - L_{IP}(\lambda_k).$$

Then

$$L_{IP}(\lambda^k + t_s\rho^k) = L_{IP}(\lambda_k) + \delta + t_s\rho^k(a^0 - A_1u^s - A_2w^s - A_3y^s).$$

Since $\delta > 0$, then there exists a $t_s > 0$, sufficiently small, such that $L_{IP}(\lambda^k + t_s\rho^k) > L_{IP}(\lambda_k)$, which contradicts our assumption that $L_{IP}(\lambda^k + t_s\rho^k) \leq L_{IP}(\lambda^k)$ for $t_s > 0$ sufficiently small. Therefore, the vertex $(u^s, w^s, y^s) \in V(\lambda^k)$.

We now show that $(u^s, w^s, y^s) \notin V_k(\lambda^k)$. By the above argument, we know that $(u^s, w^s, y^s) \in V(\lambda^k)$. Then,

$$\begin{aligned}
& L_{IP}(\lambda^k + t_s\rho^k) \\
= \; & \lambda^k a^0 + (c_u - \lambda^k A_1)u^s + (c_w - \lambda^k A_2)w^s - \lambda^k A_3 y^s + t_s\rho^k(a^0 - A_1u^s - A_2w^s - A_3y^s) \\
= \; & L_{IP}(\lambda_k) + t_s\rho^k(a^0 - A_1u^s - A_2w^s - A_3y^s) \\
\leq \; & L_{IP}(\lambda_k).
\end{aligned}$$

This implies $\rho^k(a^0 - A_1u^s - A_2w^s - A_3y^s) \leq 0$. However, we have from (44) that $\rho^k g^i > 0$ for all $i \in I_k(\lambda^k)$, where $g^i := a^0 - A_1u^i - A_2w^i - A_3y^i$. Therefore, $s \notin I_k(\lambda^k)$ so that $(u^s, w^s, y^s) \in V(\lambda^k)\backslash V_k(\lambda^k)$; that is, the vertex (u^s, w^s, y^s) is a new optimal solution of $L_{IP}(\lambda^k)$. $\square$

As we have seen, at each step we found an improving direction. Let λ^0 be our first guess for the Lagrange multiplier, then it is clear that

$$L_{IP}(\lambda^0) < L_{IP}(\lambda^1) < L_{IP}(\lambda^2) < \cdots .$$

Under the assumption of a bounded feasible set, $L_{IP}(\lambda)$ is bounded by the optimal cost of the original problem, so that $\{L_{IP}(\lambda^k)\}$ is an increasing bounded sequence. Therefore, it converges to some limit point. Proposition 3.3.3 in Section 3.3.2.3 will establish that, in fact, it converges to $\max_\lambda L_{IP}(\lambda)$.

Summary of Dual Ascent approach:

Algorithm 3.3.2. (*Dual-Ascent-Direction*)

Given $\varepsilon > 0$ *small* ;

Set $V_0(\lambda^0) \leftarrow \emptyset$;

Choose $\lambda^0 \in \mathbb{R}^\tau$ *arbitrarily* ;

Solve $L_{IP}(\lambda^0)$ — *Let* (u^0, w^0, y^0) *be an optimal solution* ;

$V_0(\lambda^0) \leftarrow V_0(\lambda^0) \cup \{(u^0, w^0, y^0)\}$;

Set $k \leftarrow 0$, $\Psi \leftarrow 1$;

while $(\Psi \neq 0)$

Given $V_k(\lambda^k)$, *construct* $< PH1(k) >$;

Solve $< PH1(k) >$;

Set $\rho^k \leftarrow$ *dual optimal sub-vector of (41)* ;

Set $\Psi \leftarrow$ *optimal cost of* $< PH1(k) >$;

if $(\Psi = 0)$

terminate the algorithm, λ^k *maximizes* $L_{IP}(\lambda)$;

else

Solve $L_{IP}(\lambda^k + \varepsilon\rho^k)$ — *Let* (u^s, w^s, y^s) *be an optimal solution* ;

if $L_{IP}(\lambda^k + \varepsilon\rho^k) > L_{IP}(\lambda^k)$

Determine step size t_k *from (46)* ;

Set $\lambda^{k+1} \leftarrow \lambda^k + t_k\rho^k$, *and find vertex set* $V_{k+1}(\lambda^{k+1})$ *from (47)* ;

$k \leftarrow k + 1$;

else

$V_k(\lambda^k) \leftarrow V_k(\lambda^k) \cup \{(u^s, w^s, y^s)\}$;

end while

end of Algorithm.

Remark. The initial choice of Ψ is arbitrary so long as $\Psi \neq 0$. While λ^0 may be chosen arbitrarily, we will use the dual optimal sub-vector of the corresponding LP relaxation. In practice, as a stopping rule, we used $\Psi < \delta$ for specified $\delta > 0$.

3.3.2.3 Step Size Rule

At the k-th step, with given Lagrange multiplier λ^k and improving direction ρ^k, we want to find the optimal step size t_k by solving

$$\max_{t \geq 0} \Phi_k(t), \tag{48}$$

where $\Phi_k(t) := L_{IP}(\lambda^k + t\rho^k)$. Recall that $L_{IP}(\lambda^k + t\rho^k)$ is a piecewise linear concave function of t and (48) is a one-dimensional optimization problem, as shown in the Figure 17.

One way of solving the step size problem is to use a modified Bolzano's bisection method (see Bazaraa et al. [6]). The modification addresses how to proceed at points where Φ_k is nondifferentiable. We first capture t_k in an *interval of uncertainty* for which $\partial\Phi_k(t)$ is positive for some t_p and negative for some $t_n > t_p$. At each iteration, solve $L_{IP}(\lambda^k + \bar{t}\rho^k)$ where $\bar{t}$ is the midpoint of the current interval of uncertainty. If $\partial\Phi_k(\bar{t}) > 0$ then $t_k \geq \bar{t}$; else $t_k \leq \bar{t}$. We only obtain one of the elements of $\partial\Phi_k(t)$ at $\bar{t}$ by solving $L_{IP}(\lambda^k + \bar{t}\rho^k)$. However, taking any one element of $\partial\Phi_k(\bar{t})$ yields the same result that one-half the interval of uncertainty is discarded at each iteration. If an exact solution is sought, the method is guaranteed to converge in the limit point. We will present next a procedure which converges finitely to the exact solution, but the rate of reduction in the interval of uncertainty is not fixed at each iteration.

As shown in Section 3.3.2.2, firstly we choose $t > 0$ small to verify that ρ^k is an improving direction. Let $\gamma_k(t) \in \partial\Phi_k(t)$ denote a subgradient of $\Phi_k(t)$. Choose $t_p > 0$ such that $\gamma_k(t_p) = \rho^k(a^0 - A_1u^1 - A_2w^1 - A_3y^1) > 0$ where (u^1, w^1, y^1) solves $L_{IP}(\lambda^k + t_p\rho^k)$. Now choose $t_n > t_p$ so that it is large enough to have its $\gamma_k(t_n) < 0$. At t_p and t_n, we know their subgradients, so that we can calculate the point $\hat{t}$ where two lines, with slopes $\gamma_k(t_p)$ and $\gamma_k(t_n)$, intersect. Now solve $L_{IP}(\lambda^k + \hat{t}\rho^k)$. Suppose $\gamma_k(\hat{t}) = \rho^k(a^0 - A_1u^2 - A_2w^2 - A_3y^2) > 0$. Then change t_p to $\hat{t}$, otherwise t_n to $\hat{t}$. Figure 17 shows the movement of t_p as $t_1, t_2, \ldots$ and movement of t_n as $\bar{t}_1, \bar{t}_2, \ldots$. This procedure terminates when $t_p = t_n$.

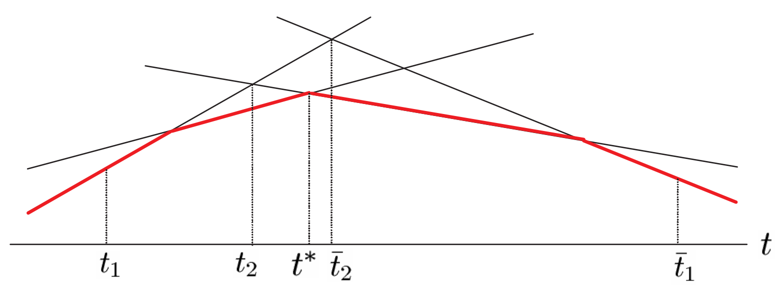

Figure 17: Choice of t to maximize one dimensional function $L_{IP}(\lambda^k + t\rho)$

As we stated in Section 3.3.2.2, the Dual Ascent approach converges. We want to show that the sequence $\{L_{IP}(\lambda^k)\}$ converges to the optimal Lagrangian dual objective value. It is well known that Lagrangian relaxation methods converge to dual bounds. for completeness, we include our proof below based on our search directions coupled with exact line searches.

Proposition 3.3.3. *Let* (ρ^k, ρ_0^k) *be dual optimal for* $< PH1(k) >$, $t_k^* \in arg\max\{L_{IP}(\lambda^k + t\rho^k), t \geq 0\}$, *and* $\lambda^{k+1} = \lambda^k + t_k^* \rho^k$. *The sequence* $\{L_{IP}(\lambda^k)\} \rightarrow \max_\lambda L_{IP}(\lambda)$.

Proof. If the sequence $\{L_{IP}(\lambda^k)\}$ is finite, then it must terminate at some iteration K for which $< PH1(K) >$ has optimal cost 0. By concavity of $L_{IP}(\lambda)$, we must have $L_{IP}(\lambda^K) = \max_\lambda L_{IP}(\lambda)$.

Now consider the case that $\{L_{IP}(\lambda^k)\}$ is an infinite sequence. For contradiction, suppose that $\{L_{IP}(\lambda^k)\}$ converges to $L < \max_\lambda L_{IP}(\lambda)$. Let $\bar{\lambda}$ satisfy $L < L_{IP}(\bar{\lambda}) \leq \max_\lambda L_{IP}(\lambda)$. For $\varsigma > 0$ sufficiently small, the vector $\bar{\rho}^k := \varsigma(\bar{\lambda} - \lambda^k)$ satisfies $L_{IP}(\lambda^k + \bar{\rho}^k) > L_{IP}(\lambda^k)$ at any iteration k. However, by the definition of t_k^* and the contradiction assumption, the direction $\bar{\lambda} - \lambda^k$ was not generated by $< PH1(k) >$ at any iteration.

Let us define

$$\bar{\rho}_0^k := \min_{i \in I_k(\lambda^k)} \{-\bar{\rho}^k(A_1 u^i + A_2 w^i + A_3 y^i)\}$$

then,

$$\bar{\rho}_0^k \leq -\bar{\rho}^k(A_1 u^i + A_2 w^i + A_3 y^i), \forall i \in I_k(\lambda^k),$$

so that

$$\bar{\rho}^k(A_1 u^i + A_2 w^i + A_3 y^i) + \bar{\rho}_0^k \leq 0, \forall i \in I_k(\lambda^k).$$

This implies that $(\bar{\rho}^k, \bar{\rho}_0^k)$ is dual feasible to $< PH1(k) >$.

As (ρ^k, ρ_0^k) is dual optimal solution for $< PH1(k) >$, we have

$$\bar{\rho}^k a^0 + \bar{\rho}_0^k \geq \rho^k a^0 + \rho_0^k = \theta > 0$$

where θ is the optimal cost of $< PH1(k) >$. However, we can choose $\varsigma > 0$ sufficiently small that satisfies

$$\bar{\rho}^k a^0 + \bar{\rho}_0^k = \varsigma(\bar{\lambda} - \lambda^k)a^0 + \min_{i \in I_k(\lambda^k)} \{-\varsigma(\bar{\lambda} - \lambda^k)(A_1 u^i + A_2 w^i + A_3 y^i)\} < \theta$$

which is a contradiction that (ρ^k, ρ_0^k) is dual optimal for $< PH1(k) >$. Therefore, $\{L_{IP}(\lambda^k)\}$ converges to $\max_\lambda L_{IP}(\lambda)$. □

3.4 Solution Strategy by Heuristic Method

In practical applications, problems as large and complex as ours take far too long to solve for an optimal solution. Even the method above yields only a bound on the optimal objective value because of the duality gap. While the general approach can be imbedded in a branch-and-bound technique to successively improve the dual bounds, this would be prohibitively expensive. We must therefore be able to compute primal feasible solutions and use the dual bound as a worst-case measure of how far a primal feasible point is away from optimality. Here we suggest heuristic methods that are fast and find a solution with known worst case optimality gap since a lower bound is obtained by the Lagrangian relaxation method.

The heuristic methods presented below are based on the following observation. In generating a sequence of cost effective greedy moves for each ship that satisfy harbor requirements, there are imbedded decision factors that can be randomized at each stage of the process and that produce a different feasible solution according to the random number generated. We can run these heuristics as many times as desired and then choose the best solution among the random trials.

3.4.1 Harbor-First Heuristic

To simplify our notation, we will describe the steps of an iterative process while suppressing the iteration counter. The steps in one iteration begin with first finding the harbor which most urgently needs service for one of the products it either supplies or demands. For the selected urgent harbor, we identify the set of ships that can provide service to that harbor. Among the set of ships, choose the one which is the most cost effective, and service the maximum possible quantities of all products.

3.4.1.1 Harbor Selection

We first select the harbor which most urgently needs to be serviced. For harbor i, urgent time U_i is defined as

$$U_i = CHT_i + \min_{k \in K_i^H} U_{ik} \tag{49}$$

where

$$U_{ik} = \begin{cases} \frac{S_{MXik} - CS_{ik}}{R_{ik}} & \text{if } J_{ik} = +1 \\ \frac{CS_{ik} - S_{MNik}}{R_{ik}} & \text{if } J_{ik} = -1 \end{cases}$$

is the urgent time of product k at harbor i. Here, CHT_i is the current time of harbor i which is initialized as 0 and updated whenever harbor i is serviced. Additionally, CS_{ik} is the current stock level (at time CHT_i) of product k at harbor i which is initialized as IS_{ik}, the initial stock level of product k at harbor i. The time U_{ik} is the time that the stock level of product k reaches the minimum (S_{MXik} if $J_{ik} = -1$) or maximum (S_{MNik} if $J_{ik} = +1$) stock level bound based on the production/consumption rate (R_{ik}) of product k in harbor i. Figure 18 shows the relationship between CS_{ik}, CHT_i, and U_{ik}.

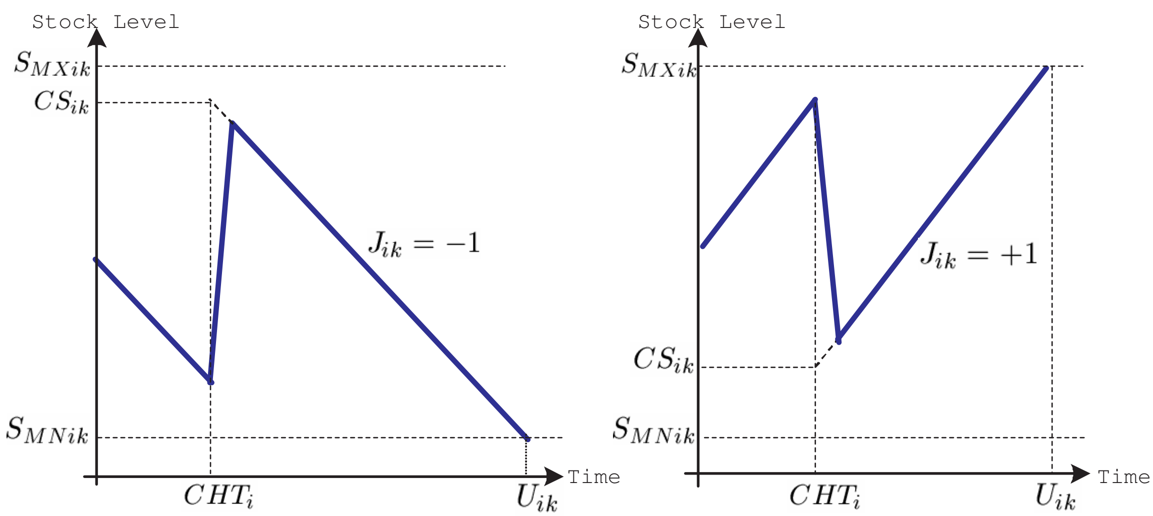

Figure 18: Calculation of the urgent time of product k at harbor i.

Each port should be visited by some ship that can service its needs before time $U_i < T$. (Recall, T is the length of the planning period of the model.) The procedure terminates when $U_i \geq T, \forall i \in H$.

3.4.1.2 Ship Selection

Suppose harbor i is the most urgent. Identify the ships that are able to reach harbor i from their current positions CP_v to the urgent port i; that is, the travel time T_{CP_viv} from current position CP_v to urgent harbor i by ship v should allow it to reach harbor i before urgent time U_i. This condition is satisfied when

$$CT_v + T_{CP_v iv} \leq U_i \tag{50}$$

where CT_v is defined as ship v's current time, which must be the earliest time that ship v can start moving.

Let UK_i denote a product that determines the urgent time U_i of harbor i; i.e., $UK_i \in \operatorname{argmin}\{U_{ik} : k \in K_i^H\}$. Among ships that satisfy condition (50), choose the subset that have a sufficient amount of product UK_i (if $J_{iUK_i} = -1$) or sufficient space to download product UK_i (if $J_{iUK_i} = +1$). We refer to this subset of ships as *candidate ships*. Among the candidate ships, we consider two criteria to choose which ship to send to the urgent harbor.

First, we can choose the ship which is the most cost effective one; namely, the candidate ship v with the least travel cost $C_{CP_v iv}$ from current position CP_v to urgent port i. A second criterion is to choose the ship which can service the largest quantity ($\sum_{k \in K_i^H} SQ_{ivk}$). Here the service quantity SQ_{ivk} can be calculated as described in Section 3.4.1.3 below.

These two different ship selection rules can be combined by constructing a weighted function $f_s(v)$ as

$$f_s(v) := \omega_t C_{CP_v iv} + \frac{\omega_q}{\sum_{k \in K_i^H} SQ_{ivk}} \tag{51}$$

where ω_t, and ω_q are, respectively, weights for the travel cost and for the reciprocal of the total amount of all products serviced by ship v.

We choose the ship v that determines the minimum $f_s(v)$ among the candidate ships. Notice that $f_s(v)$ is small if the travel cost $C_{CP_v iv}$ from current position CP_v to urgent harbor i is small and the total quantity $\sum_{k \in K_i^H} SQ_{ivk}$ to be serviced by ship v is large.

The random generation of weights ω_t and ω_q results in a different choice of ship to send to the urgent harbor for each random pair. However, because the scale of $C_{CP_v iv}$ and $\sum_{k \in K_i^H} SQ_{ivk}$ are different, we impose the relationship between weights

$$\omega_t + \frac{\omega_q}{\Omega} = 1, \tag{52}$$

where $\Omega := (\min_{v \in V_c}\{C_{CP_v iv}\})(\max_{v \in V_c}\{\sum_{k \in K_i^H} SQ_{ivk}\})$ and V_c is the set of candidate ships. For example, suppose ships 1 and 2 are in V_c. For ships 1 and 2, take the travel costs $C_{CP_v iv}$ from the current position to harbor i to be 20 and 10, respectively, and let the total service quantities $\sum_{k \in K_i^H} SQ_{ivk}$ be 100 and 50, respectively. If $\omega_t = 0.5$, then $\omega_q = 500$, because $\Omega = 1000$. For $\omega_t = 0.5$, we have $f_s(v) = 15$ for each ship 1 and 2. If we take $\omega_t < 0.5$, then $f_s(1) < f_s(2)$, so that we put more weight on quantities to be serviced. Therefore, for ω_t small, the choice of ship is likely to depend on ship v's travel cost $C_{CP_v iv}$ rather than the quantity $\sum_{k \in K_i^H} SQ_{ivk}$ that will be serviced at urgent port i.

3.4.1.3 Service quantity

Suppose ship v is chosen to service urgent harbor i. Let SQ_{ivk} be the maximum possible service quantity based on ship v's stock level of product k and the stock level of product k at harbor i. First, consider the harbor i stock level of product k at the time when ship v arrives. It can be calculated as

$$TempST_{ik} = CS_{ik} + J_{ik}R_{ik}(CT_v + T_{CP_v iv} - CHT_i), \tag{53}$$

because $CT_v + T_{CP_v iv} - CHT_i$ is the time when ship v arrives at port i. Then the maximum possible service quantity can be calculated as

$$SQ_{ivk} = \begin{cases} \min\{TempST_{ik} - S_{MNik}, CAP_{vk} - CQ_{vk}\} & \text{if } J_{ik} = +1 \\ \min\{S_{MXik} - TempST_{ik}, CQ_{vk}\} & \text{if } J_{ik} = -1 \end{cases} \tag{54}$$

where the current product quantity level CQ_{vk} is the quantity of product k onboard ship v at current time CT_v. It is initialized as $CQ_{vk} = Q_{vk}$, the initial quantity of product k on ship v at the start of the planning period.

3.4.1.4 Update Ship and Harbor Status

Suppose ship v services urgent harbor i in the amount SQ_{ivk} as determined by (54), and the stock level CS_{ik} of product k in port i at the end of service is updated according to (53). Then

- Harbor i was serviced at the time when ship v arrives at port i, so that the current time of harbor i is updated to

$$CHT_i \leftarrow CT_v + T_{CP_v iv}$$

- The current position of ship v is updated to

$$CP_v \leftarrow i$$

- The earliest time that ship v can depart for any new service is the time at which it completes its service at port i. Therefore, current time CT_v of ship v is updated to

$$CT_v \leftarrow CHT_i + \sum_{\{k \in K_i^H | SQ_{ivk} \neq 0\}} (W_i + TQ_{ik} SQ_{ivk})$$

- The quantity CQ_{ik} of product k onboard ship v after completing harbor i service is updated to.

$$CQ_{vk} \leftarrow CQ_{vk} - J_{ik} SQ_{ivk}.$$

Repeat all three steps of *harbor selection*, *ship selection* and *update ship and harbor status* until $U_i \geq T, \forall i \in H$.

Remark. For given weights (w_t, w_q) at each iteration, this method is not guaranteed to produce a feasible schedule. However, by randomizing the weights for each iteration and repeating the process multiple times, we successfully generated feasible schedules for all problems tested.

3.4.1.5 An Example

We illustrate the heuristic with the example in Figure 19. At the initial step, we assume that every ship should service as much as possible in its initial position. we can interpret this as a snapshot of a system in progress with initial time axis shifted to the present.

In Figure 19, we denote port i's status by the current time and stock level information; namely, $CHT_i : (CS_{i1}, CS_{i2}, UT_{i1}, UT_{i2})$. Ship v's status is denoted by the current time and the quantity onboard; that is, $CT_v : (CQ_{v1}, CQ_{v2})$. Ship movements are expressed as solid

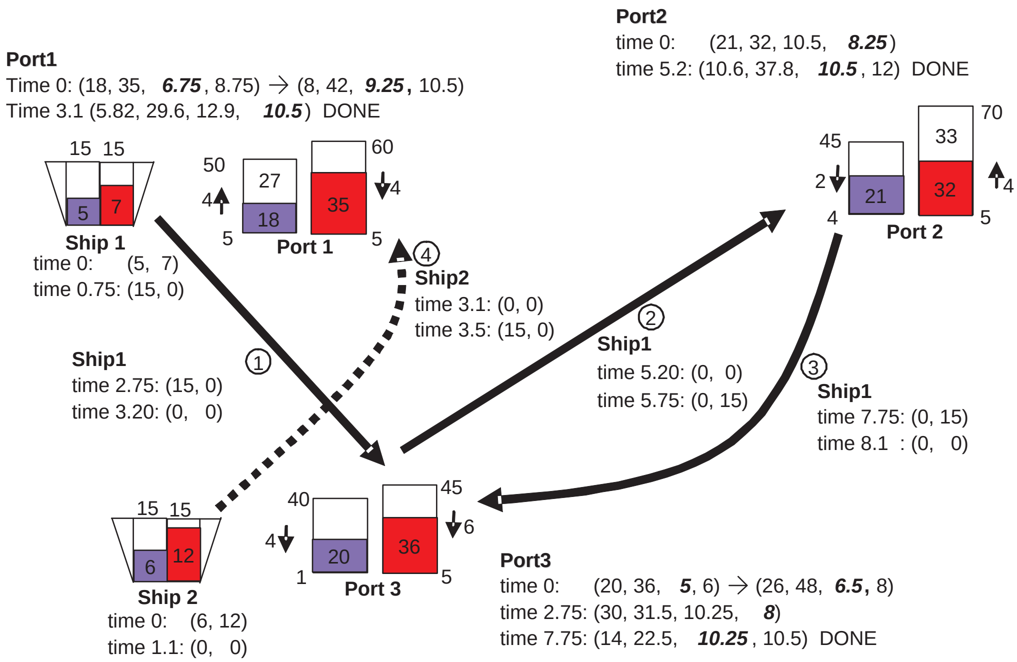

Figure 19: An example of Harbor-First Heuristic with 3 ports, 2 ships and 2 products with a planning horizon of 10 time units.

lines and dashed lines for ships 1 and 2, respectively. Moves at each iteration are marked as circled iteration numbers on the arcs. For example, at the first step, we decided to send ship 1 from the initial position to port 3 because port 3's urgent time is 5, and it is the most urgent among all other harbor urgent times. Furthermore, ship 1 is the only available ship that can be chosen among the set of ships.

3.4.2 Ship-First Heuristic

Here we consider an alternative heuristic method. We generate a sequence of ships in a random order. For the first ship, say v, in the sequence, we choose a harbor based on service urgency and the quantities that the first ship can service. This choice can be done using the same criterion function as (51), except here the argument is harbors since the ship has already been selected

$$f_H(i) := \omega_t C_{CP_v iv} + \frac{\omega_q}{\sum_{k \in K_i^H} SQ_{ivk}}. \tag{55}$$

We choose the harbor i that determines the minimum $f_H(i)$ among the *candidate harbors* whose urgent time U_i, calculated in (49), is less than the planning period and satisfies condition (50) for ship v. The random fixing of weights ω_t and ω_q leads to different choices of harbors.

After sending the first ship to the harbor that determines the minimum of $f_H(i)$ defined by (55), we eliminate the first ship in the sequence. This is one step of the heuristic. At each step, the urgency of harbors, current stock levels and time for harbors and ships are updated in exactly the same way as done in the previous Section 3.4.1. This algorithm continues until every harbor's urgent time is greater than the length of the planning horizon.

Remark. As with the first heuristic, feasible solutions are not guaranteed but our computational experience shows that the randomization process helps in generating feasible schedules after several trials.

3.4.2.1 An Example

Figure 20 shows the result of the Ship-First Heuristic. We assume that every ship services

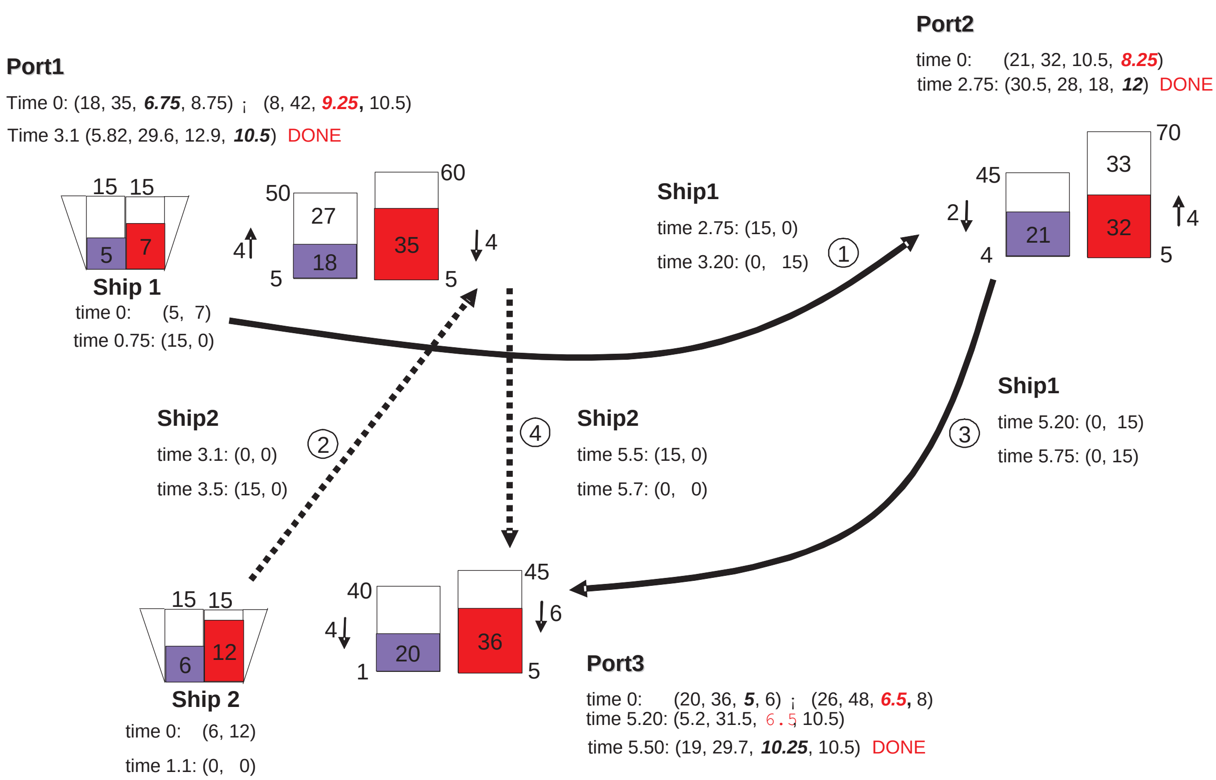

Figure 20: An example of Ship-First Heuristic (under ship sequence 1, 2, 1, 2) with 3 ports, 2 ships and 2 products with a planning horizon of 10 time units.

as much as possible in the initial position. We use the sequence of ships $\{1, 2, 1, 2, \ldots\}$ and choose the harbor to service by neglecting the costs of travel; i.e., we only consider the quantities to be serviced. The solution is totally different from the one obtained by our Harbor-First Heuristic (see Figure 19).

3.5 Computational Results

In this section, we apply the Dual Ascent method to four test problems taken from our test bed, generated in Section 2.6.2, assuming four additional harbor visits. Table 10 shows how close the lower bound, obtained by the Dual Ascent method, is to the optimal cost obtained by CPLEX using the default options of the solver. For each problem, we list the optimal objective cost and the solution time required by CPLEX, and the lower bound and the solution time taken by the Dual Ascent method. The triple $(\cdot, \cdot, \cdot)$ denotes the problem setting $(|H_T|, |V|, |K|)$. The duality gap (percentile) is calculated as

$$\frac{\text{Optimal cost} - \text{Lower bound}}{\text{Optimal cost}} \times 100.$$

Table 10: Quality of the lower bounds and computing times (seconds) by Dual Ascent Method

Test Problem	CPLEX		Dual Ascent		Duality Gap (%)
	Optimal	Solution Time	Lower Bound	Solution Time	
(3,3,2)	64.3	25284	47.1	271	26.8
(3,3,3)	75.3	450815	43.7	395	42.0
(4,2,2)	39.0	13327	25.3	252	35.1
(4,2,3)	50.2	28648	27.9	312	44.4

For the Dual Ascent method, we choose the initial guess λ^0 as the dual optimal subvector corresponding to the coupling and side constraints of the original linear programming relaxation. We chose the step size $t = 0.001$ to verify the condition (45) which is checked by $L_{IP}(\lambda^k + t\rho^k) > L_{IP}(\lambda^k)$.

To get a feasible solution for these four test problems, we applied both the Harbor-First and Ship-First heuristic methods. Table 11 shows the objective costs of the solutions obtained by these two heuristic methods along with the solution times. The optimality gap is calculated based on the better solution from the Harbor-First and Ship-First heuristics compared to the optimal cost from CPLEX. We ran these randomized greedy methods 5000 times for each heuristic method. The optimality gap (percentile) is calculated as

Table 11: Quality of the solution from the heuristic methods and computing times (seconds)

Test Problem	Harbor-First		Ship-First		Optimality Gap (%)
	Cost	Solution Time	Cost	Solution Time	
(3,3,2)	101.2	31.2	73.5	27.4	12.5
(3,3,3)	96.5	34.2	84.7	29.2	11.1
(4,2,2)	90.2	30.4	51.3	26.5	24.0
(4,2,3)	54.7	35.7	67.3	31.0	8.2

$$\frac{\text{Minimum cost of two heuristics} - \text{Optimal cost}}{\text{Minimum cost of two heuristics}} \times 100.$$

Our combined Dual Ascent/Heuristic approach produced an average duality gap of 37% and average optimality gap of 13.9%. More importantly, our solution times are on average three orders of magnitude faster than getting the exact solution by CPLEX. As an example, the case $(3, 3, 2)$ takes a total of 329.6 seconds to find a lower bound and a feasible solution with 26.8 % duality gap and 12.5% optimality gap, while CPLEX takes 25,284 seconds to find an optimal solution.

To test our method on bigger problems, we used the method in Section 2.6.2 to generate ten more test problems with 6 harbors, 4 ships, 3 products, and allowing 3 additional visits for each harbor. We first solved these mixed-integer linear programming problems by CPLEX using only the default options of the solver. We wait until the first integer feasible solution is found and record the solution time and the lower bound on the optimal objective value that CPLEX has calculated at this point. Next, we applied the Dual Ascent method and recorded the solution time and the lower bound obtained. We compare these results in Table 12, where the LP Relaxation column gives the optimal objective values of (35). The results in Table 12 show, on average, that the lower bounds obtained by the Dual Ascent method are worse than CPLEX's results; however, their average computing time was 0.28% of CPLEX's average time.

Table 12: Lower bounds and computing times (seconds) by CPLEX and the Dual Ascent Method

Test Problem	LP Relaxation	CPLEX		Dual Ascent	
		Lower Bound	Solution Time	Lower Bound	Solution Time
1	21.0	94.1	244,908	32.0	471
2	12.2	60.1	287,003	28.6	442
3	52.3	87.2	114,494	61.3	387
4	35.7	53.7	89,371	50.3	520
5	27.9	45.8	109,823	42.7	382
6	36.8	83.1	215,744	67.5	551
7	86.3	121.4	175,428	101.4	541
8	39.1	57.8	57,487	56.5	334
9	28.7	44.9	58,332	39.9	442
10	24.5	68.7	185,424	59.5	342
Average	36.5	71.7	153,801	54.0	441

The Dual Ascent method produces a lower bound on the optimal MILP. With a feasible solution, this bound can be used to establish posterior error bounds on how close that feasible solution is to optimality. This is very important since for most heuristically obtained solutions there is no mechanism for determining how good that solution is. Table 13 shows the objective cost obtained from our two heuristic methods, where the second column is the objective value of the first incumbent found by CPLEX and the last two columns are the minimum of the upper bounds produced by the two heuristics and the total time taken by the two heuristics. We ran each heuristic method 5000 times and chose the best solution among those random trials.

Test results in Table 13 show, on average, that the upper bounds obtained by our two heuristic methods are better than CPLEX's results. Additionally, our heuristic methods gave smaller upper bounds for all but three (problems 7, 8 and 10) out of ten cases. Moreover, in our experiments, CPLEX never found a feasible solution to problems bigger than the size of our test problems (namely; 6 harbors, 4 ships, 3 products and 3 additional harbor visits) within our pre-set time limit of 500K seconds.

A comparison of the worst case analysis by CPLEX and the Dual Ascent with heuristics method for these ten problems is shown in Table 14. On average, CPLEX spends 153,801

Table 13: Objective costs of feasible solutions and their computing times (seconds) by CPLEX and two heuristic methods

Test Problem	CPLEX Upper Bound	Harbor-First Upper Bound	Harbor-First Solution Time	Ship-First Upper Bound	Ship-First Solution Time	Best Upper Bound	Heuristic Solution Time
1	172.9	253.1	72.8	158.4	49.8	158.4	122.6
2	251.4	232.3	73.5	342.8	48.9	232.3	122.4
3	210.9	179.3	71.2	198.7	50.1	179.3	121.3
4	421.0	197.5	75.3	212.7	49.2	197.5	124.5
5	251.0	184.7	75.3	314.3	47.8	184.7	123.1
6	248.4	352.4	74.5	198.5	46.7	198.5	121.2
7	261.7	336.4	72.1	385.4	49.5	336.4	121.6
8	167.5	245.8	75.3	198.4	48.3	198.4	123.6
9	195.7	185.7	74.6	275.6	50.1	185.5	124.7
10	157.6	374.5	71.5	167.4	49.5	167.4	121
Average	233.8	254.1	73.61	245.2	49.0	203.8	122.6

Table 14: Worst case analysis by CPLEX and Dual Ascent Method with computing times (seconds)

Test Problem	CPLEX Worst Case (%)	CPLEX Solution Time	Dual Ascent/Heuristics Worst Case (%)	Dual Ascent/Heuristics Solution Time
1	45.6	244,908	79.7	593.6
2	76.1	287,003	87.6	564.4
3	58.6	114,494	65.7	508.3
4	87.2	89,371	74.5	644.5
5	81.8	109,823	76.9	505.1
6	66.5	215,744	65.9	672.2
7	53.6	175,428	69.9	662.6
8	65.5	57,487	71.5	457.6
9	77.1	58,332	78.5	566.7
10	56.4	185,424	64.4	463.0
Average	66.8	153,801	73.4	563.8

seconds to get a solution with an average posterior bound gap of 66.8% while the Dual Ascent method with two heuristics spends 563 seconds with an average posterior bound gap of 73.5%. However the Dual Ascent with heuristics takes only 0.4% of the solution time required by CPLEX.

3.6 Summary and Concluding Remarks

This research deals with chemical transport problems involving maritime pick up from and delivery to storage tanks that are continuously filled and drained. More specifically, we developed decision technology to determine the efficient use of multi-compartment bulk ships to transport chemical products while ensuring continuous production with no stock-outs, so that the inventory level of chemical products in storage tanks are maintained between prescribed upper and lower stock levels during the planning horizon. Due to the nature of the products, it is impossible to carry more than two products without these being separated into dedicated compartments of the ships. We need to decide how much of each product to carry, on which ship, subject to the conditions that all harbors must have sufficient product to meet demand, and the stock levels of the products cannot exceed the inventory capacity of that harbor.

We have formulated this ship-routing problem as a combined multi-ship pickup-delivery problem with inventory constraints. The original problem is a large-scale non-convex mixed-integer programming problem. All non-convexities involved weighted sums of products of two variables, one of which is binary and the other is continuous but bounded. We have shown that the structure gives rise to an equivalent large-scale linear mixed-integer programming problem (MILP).

We studied the underlying structure of the MILP and investigated several possible solution approaches. As a solution strategy for this large scale MILP with special structure, the Lagrangian relaxation method was used to find a bound (because of the duality gap) on the optimal objective value. Lagrangian relaxation takes advantage of the constraint structure of the model. By dualizing the coupling and side constraints, we generate a master problem for a given set of Lagrange multipliers. The master problem itself can be decomposed into several more tractable subproblems. One set of subproblems can be solved using network flow technology and the other set of subproblems can be solved by integer programming technology as the number of binary variables is relatively small. Using the solutions from the subproblems, we can update the Lagrange multipliers and solve a new master problem, and so on. Lagrange multipliers are updated in such a way that the optimal cost of the master (lower bounding) problem keeps increasing. To do this, we first find an improving direction from the current Lagrange multiplier vector and solve the optimal step size along that direction. With updated Lagrange multipliers the master problem is solved again, and so on until no improving direction exists.

To get a feasible solution, we devised heuristic methods that are fast and find a good solution. We generate a sequence of greedy moves for each ship that satisfy harbor requirements and that are as cost effective as possible. These greedy heuristics imbed decision factors that can be randomized at each stage of the process, and that produce a different feasible solution according to the random number generated. We can run these heuristics as many times as desired and then choose the best solution among the random trials.

Details of the Lagrangian relaxation method were presented and convergence to the dual solution was established. However, because of the duality gap, a primal feasible solution is generally not available. Heuristic methods for finding primal feasible solutions were also developed. Numerical experiments on small test problems indicate that the heuristics are very effective at finding good solutions quickly.

We conducted numerical studies to establish the goodness of the heuristic solution, on average, when compared to the dual bounds. This gives a worst case analysis since the dual bounds manifest a duality gap with respect to the primal optimal objective value. However, in theory at least, the dual bounds can be tightened by performing branch and bound on the primal binary variables and solving the subproblems by the Lagrangian relaxation method. One way of performing the branch and bound process is as follows. Assume that the Dual Ascent method terminates with δ-optimal objective value of problem $< PH1(k) >$ in step k. Solving the $< PH1(k) >$ produces optimal coefficients α_i that satisfy the constraints $(A_1u^1 + A_2w^1 + A_3y^1)\alpha_1 + \cdots + (A_1u^n + A_2w^n + A_3y^n)\alpha_n = a^0$ (See (41)) of problem $< PH1(k) >$. The convex combination of the solutions of the master problem (i.e., $(\alpha_1u^1 + \cdots + \alpha_nu^n,\ \alpha_1w^1 + \cdots + \alpha_nw^n,\ \alpha_1y^1 + \cdots + \alpha_ny^n)$) can be fractional valued for components that are required to be binary. We can branch on those variables as 0 or 1. At the termination of this branch and bound process, one has a convex combination of the solutions of the master problem which is feasible to the original problem because constraints (41) are satisfied and all mixed-integer decision vectors meet their integrality requirements. A few steps of this approach can be used to tighten the error bound but this would not be an efficient approach to solve our maritime routing and scheduling problem.

APPENDIX A

NOTE ON CONVEX UNDERESTIMATES OF SUMS OF PRODUCTS OF LINEAR FUNCTIONS

In this appendix we describe a technique for reformulating some structured nonlinear programs into linear programs by introducing additional variables. Specifically, we show that a nonlinear mixed-integer program can be reformulated into an equivalent mixed-integer linear program under certain conditions. In some cases, the reformulation can be tightened by judiciously choosing the nonlinear terms to be linearized.

A.1 Introduction

The purpose of this appendix is to derive some results of convex underestimates of sums of products of linear functions that are useful in certain applications. This appendix demonstrates straightforward extensions and results of Al-Khayyal and Falk [2] (See also Al-Khayyal [1]) and later extended by Sherali and Alameddine [50] and Sherali and Adams [48], as well as by Tawarmalani and Sahinidis [52]. We will derive the basic ideas of this useful relaxation technique and illustrate some extensions in Section A.2. In Section A.3, we apply this result to general linear functions. We also show that a particular type of non-linear mixed-integer program can be reformulated into an equivalent mixed-integer linear program under certain conditions, thereby making problem solving relatively easy. Also it compares two alternative relaxation methods and shows one is better than the other in the sense of tighter relaxation.

A.2 Simplification and extensions

In this section we will discuss the linearization of the feasible region defined by a special nonlinear equation formed by the product of variables. We will preliminarily start to investigate the linearization technique for the nonlinear form of the product of two variables xy and extend it to the form of $\frac{x}{y}$.

A.2.1 Relaxation of product of two variables

Consider the compact set B defined by $x, y \in \mathbb{R}$

$$B := \{(x, y)|L_x \leq x \leq U_x, L_y \leq y \leq U_y\}.$$

It is formed by the four constraints

$$\text{i) } x - L_x \geq 0 \qquad \text{ii) } U_x - x \geq 0$$

$$\text{i') } y - L_y \geq 0 \qquad \text{ii') } U_y - y \geq 0.$$

By writing the four ways of multiplying these nonnegative quantities, we obtain the set C_I of *implied constraints* which is defined as

$$C_I := \{(x, y)|xy \geq L_y x + L_x y - L_x L_y, \quad xy \geq U_y x + U_x y - U_x U_y$$
$$xy \leq U_y x + L_x y - L_x U_y, \quad xy \leq L_y x + U_x y - U_x L_y\}.$$

Specifically, we have

$$
\begin{aligned}
& x - L_x \geq 0, \quad y - L_y \geq 0, \\
\Rightarrow \quad & (x - L_x)(y - L_y) \geq 0, \\
\Rightarrow \quad & xy \geq L_y x + L_x y - L_x L_y \\[1em]
& U_x - x \geq 0, \quad U_y - y \geq 0, \\
\Rightarrow \quad & (U_x - x)(U_y - y) \geq 0, \\
\Rightarrow \quad & xy \geq U_y x + U_x y - U_x U_y \\[1em]
& x - L_x \geq 0, \quad U_y - y \geq 0, \\
\Rightarrow \quad & (x - L_x)(U_y - y) \geq 0, \\
\Rightarrow \quad & xy \leq U_y x + L_x y - L_x U_y \\[1em]
& y - L_y \geq 0, \quad U_x - x \geq 0, \\
\Rightarrow \quad & (y - L_y)(U_x - x) \geq 0, \\
\Rightarrow \quad & xy \leq L_y x + U_x y - U_x L_y.
\end{aligned}
$$

Notice that $B \subseteq C_I$. Now for notational simplicity, introduce the notation

$$[L,U] := \{(x,y)|L_x \le x \le U_x, L_y \le y \le U_y\}, \text{ where } L := \begin{bmatrix} L_x \\ L_y \end{bmatrix}, U := \begin{bmatrix} U_x \\ U_y \end{bmatrix}.$$

Then $B = \{(x,y)|(x,y) \in [L,U]\}$. Now consider the set B' in higher dimensional space

$$B' := \{(x,y,z)|(x,y) \in [L,U], z = xy\} \bigcap \{(x,y,z)|(x,y) \in C_I, z = xy\}.$$

Then, set B' has constraints as follows,

$$\begin{aligned} z &= xy, \\ L_x \le x &\le U_x, \\ L_y \le y &\le U_y, \\ z &\ge L_y x + L_x y - L_x L_y, \\ z &\ge U_y x + U_x y - U_x U_y, \\ z &\le U_y x + L_x y - L_x U_y, \\ z &\le L_y x + U_x y - U_x L_y. \end{aligned}$$

Notice that the projection B' onto the x-y plane is exactly B itself. By eliminating constraint $z = xy$ in B', we obtain the relaxation $\tilde{B}'$ of B'

$$\tilde{B}' := \{(x,y,z)|(x,y) \in [L,U]\} \bigcap \{(x,y,z)|(x,y) \in C_I, z = xy\}.$$

Then, $B' \subseteq \tilde{B}'$. Now, the projection of $\tilde{B}'$ onto the x-y plane is

$$\text{Proj}_{\mathbb{R}^2} \tilde{B}' := \{(x,y)|(x,y,z) \in \tilde{B}' \text{ for all } z\}.$$

Then, it is clear that $B \subseteq \text{Proj}_{\mathbb{R}^2} \tilde{B}'$ which means that for all $(x,y) \in B$, there exist z such that $(x,y,z) \in \tilde{B}'$.

Notice that $\tilde{B}' \subseteq \{(x,y,z)|(x,y) \in C_I, z = xy\}$ so $\tilde{B}'$ has constraints defining C_I which are obtained by substituting $z = xy$, that is

$$z \ge \max\{L_y x + L_x y - L_x L_y \ , \ U_y x + U_x y - U_x U_y\}, \tag{56}$$

$$z \le \min\{U_y x + L_x y - L_x U_y \ , \ L_y x + U_x y - U_x L_y\}. \tag{57}$$

Then, (56) and (57) represent convex and concave envelope of xy over $[L, U]$ respectively. So, whenever xy appears in a problem with bounded variables, we can linearize xy by introducing the new variable $z = xy$ and adding constraints C_I with $z = xy$.

As an example, consider feasible region S with $[L, U] = [0, e]$, where $e = (1, 1)^T$, and the additional constraint $xy \leq \frac{1}{2}$. Then we can express $\tilde{S}'$ as

$$z \geq \max\{0, x + y - 1\}, \quad z \leq \min\{x, y\}, \quad z \leq \frac{1}{2}.$$

Now Figure 21 shows the feasible regions S and $\tilde{S}'$. Set S is a two-dimensional space in the x-y plane. Set $\tilde{S}'$ is in the 3-dimensional space of (x, y, z). As shown in the Figure 21, for

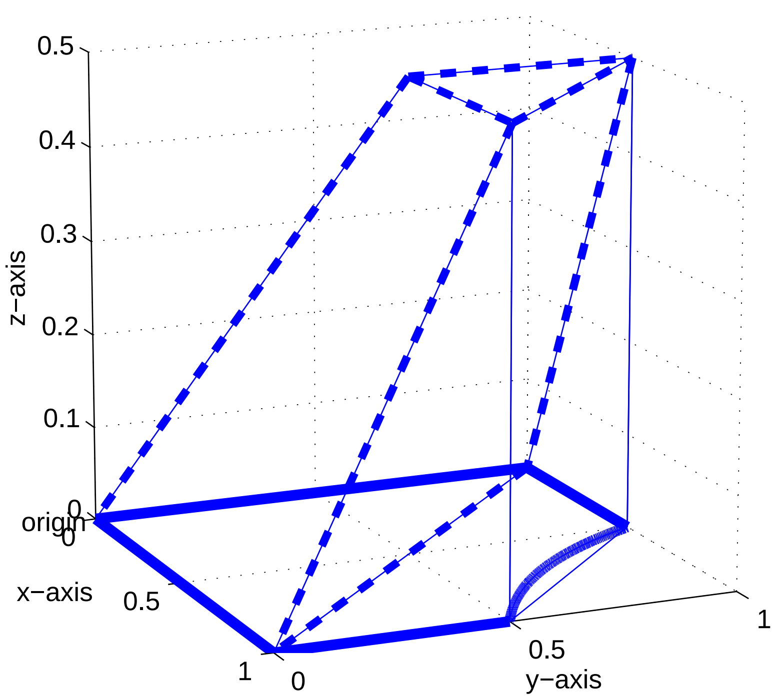

Figure 21: Example of nonlinear feasible region and its convex relaxation

every point $(x, y) \in S$, there exist z such that $(x, y, z) \in \tilde{S}'$. The projection of $\tilde{S}'$ onto the x-y plane is shown in Figure 22. It illustrates why the feasible region S is a subset of the convex set $\mathrm{Proj}_{\mathbb{R}^2}\tilde{B}'$.

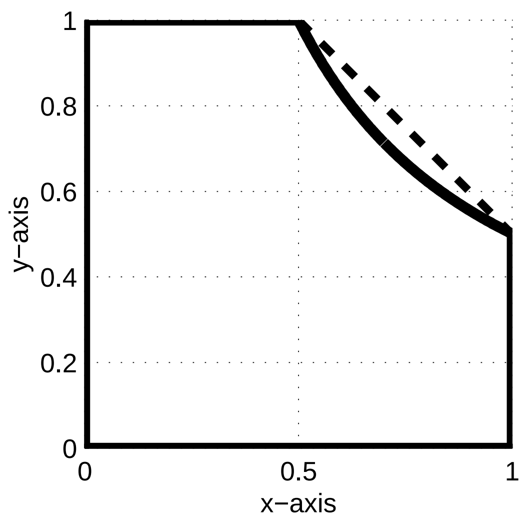

Figure 22: Original nonlinear feasible region and projected region

A.2.2 Extension to the form of $\frac{x}{y}$

Consider feasible region F generated by $x, y \in \mathbb{R}$ for $0 < L_y \leq U_y$, and suppose there exist additional constraints with the term $\frac{x}{y}$. Then, the hyperrectangle

$$L_x \leq x \leq U_x,$$
$$L_y \leq y \leq U_y$$

is equivalent to

$$L_x \leq x \leq U_x,$$
$$\frac{1}{U_y} \leq \frac{1}{y} \leq \frac{1}{L_y}.$$

So we can apply the same result of (56) and (57) as follows

$$L_x \leq x \leq U_x, \quad L_y \leq y \leq U_y,$$
$$z \geq \frac{x}{U_y} + \frac{L_x}{y} - \frac{L_x}{U_y}, \quad z \geq \frac{x}{L_y} + \frac{U_x}{y} - \frac{U_x}{L_y},$$
$$z \leq \frac{x}{L_y} + \frac{L_x}{y} - \frac{L_x}{L_y}, \quad z \leq \frac{x}{U_y} + \frac{U_x}{y} - \frac{U_x}{U_y}.$$

By letting $\frac{1}{y} = w$ we can rewrite it as

$$
\begin{aligned}
yw &= 1, \\
L_x \leq x &\leq U_x, \\
L_y \leq y &\leq U_y, \\
z &\geq \frac{x}{U_y} + L_x w - \frac{L_x}{U_y}, \\
z &\geq \frac{x}{L_y} + U_x w - \frac{U_x}{L_y}, \\
z &\leq \frac{x}{L_y} + L_x w - \frac{L_x}{L_y}, \\
z &\leq \frac{x}{U_y} + U_x w - \frac{U_x}{U_y}.
\end{aligned}
$$

Then, substituting $z' = wy$ and applying the same method with bounds on $w \in [\frac{1}{U_y}, \frac{1}{L_y}]$, we get

$$
\begin{aligned}
z' &= 1, \\
L_x \leq x &\leq U_x, \\
L_y \leq y &\leq U_y, \\
z &\geq \frac{x}{U_y} + L_x w - \frac{L_x}{U_y}, \\
z &\geq \frac{x}{L_y} + U_x w - \frac{U_x}{L_y}, \\
z &\leq \frac{x}{L_y} + L_x w - \frac{L_x}{L_y}, \\
z &\leq \frac{x}{U_y} + U_x w - \frac{U_x}{U_y}, \\
z' &\geq L_y w + \frac{y}{U_y} - \frac{L_y}{U_y}, \\
z' &\geq U_y w + \frac{y}{L_y} - \frac{U_y}{L_y}, \\
z' &\leq U_y w + \frac{y}{U_y} - 1, \\
z' &\leq L_y w + \frac{y}{L_y} - 1.
\end{aligned}
$$

Substituting $z' = 1$ gives

$$
\begin{aligned}
L_x \le x &\le U_x, \\
L_y \le y &\le U_y, \\
z \ge \frac{x}{U_y} + L_x w &- \frac{L_x}{U_y}, \\
z \ge \frac{x}{L_y} + U_x w &- \frac{U_x}{L_y}, \\
z \le \frac{x}{L_y} + L_x w &- \frac{L_x}{L_y}, \\
z \le \frac{x}{U_y} + U_x w &- \frac{U_x}{U_y}, \\
1 \ge L_y w + \frac{y}{U_y} &- \frac{L_y}{U_y}, \\
1 \ge U_y w + \frac{y}{L_y} &- \frac{U_y}{L_y}, \\
2 \le U_y w &+ \frac{y}{U_y}, \\
2 \le L_y w &+ \frac{y}{L_y}.
\end{aligned}
$$

For example, suppose we have constraints with a term $\frac{x}{y}$ in a compact set defined by

$$
\begin{aligned}
0 \le x \le 1, \\
1 \le y \le 2.
\end{aligned}
$$

Applying the result we get the convex relaxation as follows

$$
\begin{aligned}
x - 2z &\le 0, \\
x + w - z &\le 1, \\
x - z &\ge 0, \\
x + 2w - 2z &\ge 1, \\
2w + y &\le 3, \\
4w + y &\ge 4, \\
w + y &\ge 2, \\
0 \le x &\le 1, \\
1 \le y &\le 2.
\end{aligned}
$$

If we have constraint $\frac{x}{y} \leq 1$ and objective function to maximize $x + y$ then the optimal solution to the original problem is $(x^*, y^*) = (1, 2)$ and the optimal solution to the relaxed problem is determined at $(x^*, y^*, z^*, w^*) = (1, 2, \frac{1}{2}, \frac{1}{2})$.

A.3 Product of Linear Functions

In Section A.2, we have seen the relaxation technique for the form of product of variables. In this section we will investigate the product form of linear functions as a generalization of the result from Section A.2. Furthermore, we will show that some cases of relaxation give exact reformulation and others give tighter relaxations.

A.3.1 Product of linear functions

Given $f(x)$ and $g(y)$ are linear functions of vectors x and y. Consider the feasible region F generated by the following constraints

$$\begin{aligned} f(x)g(y) &\leq U, \\ L_f \leq f(x) &\leq U_f, \\ L_g \leq g(y) &\leq U_g. \end{aligned}$$

Consider the linearization of $f(x)g(y)$ by substituting $f(x)g(y) = z$, define the feasible region F' as

$$\begin{aligned} z &\leq U, \\ L_f \leq f(x) &\leq U_f, \\ L_g \leq g(y) &\leq U_g, \\ z &\geq L_g f(x) + L_f g(y) - L_f L_g, \\ z &\geq U_g f(x) + U_f g(y) - U_f U_g, \\ z &\leq U_g f(x) + L_f g(y) - L_f U_g, \\ z &\leq L_g f(x) + U_f g(y) - U_f L_g. \end{aligned}$$

Then F' is a relaxation of F and it is convex. Since

$$\begin{aligned}
& f(x) - L_f \geq 0, \quad g(y) - L_g \geq 0 \\
\Rightarrow \quad & (f(x) - L_f)(g(y) - L_g) \geq 0 \\
\Rightarrow \quad & f(x)g(y) \geq L_g f(x) + L_f g(y) - L_f L_g, \qquad (58)
\end{aligned}$$

$$\begin{aligned}
& U_f - f(x) \geq 0, \quad U_g - g(y) \geq 0 \\
\Rightarrow \quad & (U_f - f(x))(U_g - g(y)) \geq 0 \\
\Rightarrow \quad & f(x)g(y) \geq U_g f(x) + U_f g(y) - U_f U_g, \qquad (59)
\end{aligned}$$

$$\begin{aligned}
& f(x) - L_f \geq 0, \quad U_g - g(y) \geq 0 \\
\Rightarrow \quad & (f(x) - L_f)(U_g - g(y)) \geq 0 \\
\Rightarrow \quad & f(x)g(y) \leq U_g f(x) + L_f g(y) - L_f U_g, \qquad (60)
\end{aligned}$$

$$\begin{aligned}
& g(y) - L_g \geq 0, \quad U_f - f(x) \geq 0 \\
\Rightarrow \quad & (g(y) - L_g)(U_f - f(x)) \geq 0 \\
\Rightarrow \quad & f(x)g(y) \leq L_g f(x) + U_f g(y) - U_f L_g. \qquad (61)
\end{aligned}$$

From equation (58) and (59), we have

$$f(x)g(y) \geq \max\{L_g f(x) + L_f g(y) - L_f L_g \ , \ U_g f(x) + U_f g(y) - U_f U_g\}, \qquad (62)$$

and from equation (60) and (61) we get

$$f(x)g(y) \leq \min\{U_g f(x) + L_f g(y) - L_f U_g \ , \ L_g f(x) + U_f g(y) - L_g U_f\}. \qquad (63)$$

Combining (62) and (63), we see that the product $f(x)g(y)$ of two linear functions is bounded below by a piecewise linear convex function and bounded above by a piecewise linear concave function. So, if we replace the product form of $f(x)g(y)$ by z then we can rewrite the expression (58) through (61) as follows

$$z \geq L_g f(x) + L_f g(y) - L_f L_g,$$

$$z \geq U_g f(x) + U_f g(y) - U_f U_g,$$

$$z \leq U_g f(x) + L_f g(y) - L_f U_g,$$

$$z \leq L_g f(x) + U_f g(y) - U_f L_g.$$

It is clear that linearization of the product of functions can also be extended to the case of $\frac{f(x)}{g(x)}$. The basic idea of Section A.2 can be applied directly to such an extension and the desired relaxation can easily found.

A.3.2 Exactness of convex relaxation

Now we want to investigate exactness of this relaxation under certain conditions. The following Proposition A.3.1 is a well known result that commonly arises in optimization problems in the situation when something happens ($x = 1$) then another condition should follow that is represented by $f(y) = 0$.

Proposition A.3.1. *Consider the set* $S := \{(x, y) \mid xf(y) = 0, x \in \{0, 1\}, y \in \mathbb{Y}\}$*, where* $\{f(y) \mid y \in \mathbb{Y}\}$ *is compact; i.e, there exist bounds* $[L, U]$ *such that* $L \leq f(y) \leq U$ *for all* $y \in \mathbb{Y}$*. Then, set* S *is equivalent to :*

$$S' := \{(x, y) \mid L(1 - x) \leq f(y) \leq U(1 - x),\ x \in \{0, 1\},\ y \in \mathbb{Y}\}.$$

Proof. Suppose $x = 1$, then $f(y) = 0$ for both set S and S'. If $x = 0$ then any y satisfies $L \leq f(y) \leq U$ is in the set S and S'. □

The set S' can be derived exactly from applying the relaxation technique of Section A.3.1. Now we will state a more general result of exactness which follows readily from the foregoing technique.

Proposition A.3.2. *Consider the following nonlinear feasible region* P_1*, where* $L \leq U$ *and*

$l \leq u$, and the relaxation defined by P_2.

$$
\begin{aligned}
P_1 :=\ & \{\ (x,y) \mid a \leq xf(y) \leq b,\ L \leq f(y) \leq U,\ x \in \{l,u\}\ \} \\
P_2 :=\ & \{\ (x,y,z) \mid a \leq z \leq b,\ L \leq f(y) \leq U,\ x \in \{l,u\}, \\
& \qquad\qquad z \geq lf(y) + Lx - Ll,\ z \geq uf(y) + Ux - Uu, \\
& \qquad\qquad z \leq uf(y) + Lx - Lu,\ z \leq lf(y) + Ux - Ul\ \}.
\end{aligned}
$$

If $(x,y,z) \in P_2$, then $z = xf(y)$ and $(x,y) \in P_1$.

Proof. We can divide P_2 into two cases that is $x = l$ and $x = u$. If $x = l$ then the last four equations in P_2 are

$$
\begin{aligned}
z &\geq lf(y) + Ll - Ll, \Rightarrow z \geq lf(y), \\
z &\geq uf(y) + Ul - Uu, \Rightarrow z - uf(y) \geq U(l-u), \\
z &\leq uf(y) + Ll - Lu, \Rightarrow z - uf(y) \leq L(l-u), \\
z &\leq lf(y) + Ul - Ul, \Rightarrow z \leq lf(y).
\end{aligned}
$$

Then $z = lf(y)$ and $U(l-u) \leq z - uf(y) \leq L(l-u) \Rightarrow L \leq f(y) \leq U$ because $l \leq u$ and $z - uf(y) = f(y)(l-u)$. Now if $x = u$ then

$$
\begin{aligned}
z &\geq lf(y) + Lu - Ll, \Rightarrow z - lf(y) \geq L(u-l), \\
z &\geq uf(y) + Uu - Uu, \Rightarrow z \geq uf(y), \\
z &\leq uf(y) + Lu - Lu, \Rightarrow z \leq uf(y), \\
z &\leq lf(y) + Uu - Ul, \Rightarrow z - lf(y) \leq U(u-l)
\end{aligned}
$$

gives $z = uf(y)$ and $L \leq f(y) \leq U$. Therefore, if $(x,y,z) \in P_2$, then $z = xf(y)$ and $(x,y) \in P_1$. □

If $f(y)$ is discrete and x continuous, we have the analogous result.

Proposition A.3.3. *For given $L \le U$ and $l \le u$, define the sets*

$$P_1' := \ \{ \ (x,y) \mid a \le xf(y) \le b, \ l \le x \le u, \ f(y) \in \{f(L), f(U)\} \ \}$$

$$\begin{aligned} P_2' := \ \{ \ (x,y,z) \mid a \le z \le b, \ l \le x \le u, \ f(y) \in \{f(L), f(U)\}, \\ z \ge f(L)x + lf(y) - lf(L), \ z \ge f(U)x + uf(y) - uf(U), \\ z \le f(U)x + lf(y) - lf(U), \ z \le f(L)x + uf(y) - uf(L) \ \}. \end{aligned}$$

If $(x,y,z) \in P_2'$, then $z = xf(y)$ and $(x,y) \in P_1'$.

Taking Propositions A.3.2 and A.3.3 together, we have the following

Theorem A.3.4. *Consider an optimization problem (P) which has terms $xf(y)$, where (x,y) is constrained to be in either P_1 or P_1', and the corresponding relaxed problem (P_R) obtained by replacing $xf(y)$ with z and respectively, P_1 with P_2 (P_1' with P_2'). It follows that the (x,y) component of the optimal solution of problem (P_R) is optimal for problem (P).*

Remark. Thus the relaxation of (P) given by (P_R) is exact in the sense that it will always produce an optimal solution for the original problem.

Corollary A.3.5. *Consider optimization problem $(\bar{P})$ which has terms $xf(y)$, where (x,y) is constrained to be in either $\bar{P}_1$ or $\bar{P}_1{}'$ whose binary terms in P_1 and P_1' are linearly relaxed, and the corresponding relaxed problem $(\bar{P}_R)$ obtained by replacing $xf(y)$ with z and respectively, $\bar{P}_1$ with $\bar{P}_2$ ($\bar{P}_1{}'$ with $\bar{P}_2{}'$). If optimal solution (x^*, y^*, z^*) to $(\bar{P}_R)$ satisfies $x^* \in \{l, u\}$ or $f(y^*) \in \{F(L), F(U)\}$ then (x^*, y^*) is optimal for problem $(\bar{P})$.*

Remark. Thus the relaxation of $(\bar{P})$ given by $(\bar{P}_R)$ is exact in the sense that if any one variable of an optimal solution of $(\bar{P}_R)$ is at the boundary point, then it will always produce an optimal solution for the original problem.

A.3.3 Tighter relaxation

Now we want to show that one of two alternatives of relaxation is tighter than the other. Both ways are exact when one of the variable or function is binary by the exactness shown above.

Proposition A.3.6. *Consider the following nonlinear feasible region*

$$S := \{(x, y, z) \quad | \quad (f_1(x) - f_2(y))g(z) \le U, \quad L_{f_1} \le f_1(x) \le U_{f_1},$$
$$L_{f_2} \le f_2(y) \le U_{f_2}, \quad L_g \le g(z) \le U_g\}.$$

Let S_1 be the projection of the reformulation onto the space of S obtained by linearizing $(f_1(x) - f_2(y))g(z)$ by a single variable, and let S_2 be projection of the reformulation onto the space of S obtained by linearizing $f_1(x)g(z)$ and $f_2(y)g(z)$ using two separate variables. Then S_2 is a tighter reformulation than S_1, i.e. $S_2 \subset S_1$.

Proof. Consider the nonlinear function

$$(f_1(x) - f_2(y))g(z)$$

over a domain such that each component function has known lower and upper bounds over its domain or subset of interest; i.e.,

$$L_{f_1} \le f_1(x) \le U_{f_1},$$
$$L_{f_2} \le f_2(y) \le U_{f_2},$$
$$L_g \le g(z) \le U_g.$$

Define $w = f_1(x) - f_2(y)$ which has bounds

$$L_{f_1} - U_{f_2} \le w \le U_{f_1} - L_{f_2}.$$

Let $u = wg(z)$, then

$$u \ge (L_{f_1} - U_{f_2})g(z) + L_g[f_1(x) - f_2(y)] - L_g(L_{f_1} - U_{f_2}) = \alpha,$$
$$u \ge (U_{f_1} - L_{f_2})g(z) + U_g[f_1(x) - f_2(y)] - U_g(U_{f_1} - L_{f_2}) = \beta,$$
$$u \le (U_{f_1} - L_{f_2})g(z) + L_g[f_1(x) - f_2(y)] - L_g(U_{f_1} - L_{f_2}) = \gamma,$$
$$u \le (L_{f_1} - U_{f_2})g(z) + U_g[f_1(x) - f_2(y)] - U_g(L_{f_1} - U_{f_2}) = \delta.$$

The latter four inequalities can be summarized as

$$max\{\alpha, \beta\} \le u \le min\{\gamma, \delta\}. \tag{64}$$

Now, let $v_1 = f_1(x)g(z)$ and $v_2 = f_2(y)g(z)$. Then

$$v_1 \geq L_g f_1(x) + L_{f_1} g(z) - L_{f_1} L_g = \alpha_1,$$

$$v_1 \geq U_{f_1} g(z) + U_g f_1(x) - U_{f_1} U_g = \beta_1,$$

$$v_1 \leq U_g f_1(x) + L_{f_1} g(z) - L_{f_1} U_g = \gamma_1,$$

$$v_1 \leq L_g f_1(x) + U_{f_1} g(z) - L_g U_{f_1} = \delta_1,$$

$$v_2 \geq L_g f_2(y) + L_{f_2} g(z) - L_{f_2} L_g = \alpha_2,$$

$$v_2 \geq U_{f_2} g(z) + U_g f_2(y) - U_{f_2} U_g = \beta_2,$$

$$v_2 \leq U_g f_2(y) + L_{f_2} g(z) - L_{f_2} U_g = \gamma_2,$$

$$v_2 \leq L_g f_2(y) + U_{f_2} g(z) - L_g U_{f_2} = \delta_2.$$

We can summarize the latter eight inequalities as

$$max\{\alpha_1, \beta_1\} \leq v_1 \leq min\{\gamma_1, \delta_1\}, \tag{65}$$

$$max\{\alpha_2, \beta_2\} \leq v_2 \leq min\{\gamma_2, \delta_2\}. \tag{66}$$

Since $u = v_1 - v_2$, the bounds on $v_1 - v_2$ can be determined from (65) and (66) as

$$max\{\alpha_1, \beta_1\} - min\{\gamma_2, \delta_2\} \leq v_1 - v_2 \leq min\{\gamma_1, \delta_1\} - max\{\alpha_2, \beta_2\}. \tag{67}$$

Moreover, since $\alpha = \alpha_1 - \delta_2$, $\beta = \beta_1 - \gamma_2$, $\gamma = \delta_1 - \alpha_2$, and $\delta = \gamma_1 - \beta_2$, the bounds given by (64) can be written as

$$max\{\alpha_1 - \delta_2, \beta_1 - \gamma_2\} \leq v_1 - v_2 \leq min\{\delta_1 - \alpha_2, \gamma_1 - \beta_2\}. \tag{68}$$

We will now show that the lower bounds specified by (67) are always greater than or equal to the lower bounds determined by (68) and the upper bounds of (67) are always less than or equal to the upper bounds of (68); i.e., the bounds on $v_1 - v_2$ given by (67) are tighter than those given by (68).

For the lower bound, there are two cases to consider.

case 1 $\alpha_1 - \delta_2 \geq \beta_1 - \gamma_2$. Consider the four subcases

$$
\begin{array}{ll}
(i) & \alpha_1 \geq \beta_1,\ \gamma_2 \geq \delta_2, \\
(ii) & \alpha_1 \leq \beta_1,\ \gamma_2 \geq \delta_2, \\
(iii) & \alpha_1 \geq \beta_1,\ \gamma_2 \leq \delta_2, \\
(iv) & \alpha_1 \leq \beta_1,\ \gamma_2 \leq \delta_2.
\end{array}
$$

For each of these subcases, it follows that

$$\max\{\alpha_1, \beta_1\} - \min\{\gamma_2, \delta_2\} \geq \alpha_1 - \delta_2 = \max\{\alpha_1 - \delta_2, \beta_1 - \gamma_2\}.$$

case 2 $\beta_1 - \gamma_2 \geq \alpha_1 - \delta_2$. For the same foregoing subcases (i) through (iv), it follows that

$$\max\{\alpha_1, \beta_1\} - \min\{\gamma_2, \delta_2\} \geq \beta_1 - \gamma_2 = \max\{\alpha_1 - \delta_2, \beta_1 - \gamma_2\}.$$

An analogous argument is applied for the upper bound as follows.

case 1 $\delta_1 - \alpha_2 \leq \gamma_1 - \beta_2$. Consider the four subcases

$$
\begin{array}{ll}
(\bar{i}) & \delta_1 \geq \gamma_1,\ \alpha_2 \geq \beta_2, \\
(\overline{ii}) & \delta_1 \leq \gamma_1,\ \alpha_2 \geq \beta_2, \\
(\overline{iii}) & \delta_1 \geq \gamma_1,\ \alpha_2 \leq \beta_2, \\
(\overline{iv}) & \delta_1 \leq \gamma_1,\ \alpha_2 \leq \beta_2.
\end{array}
$$

For each of these subcases, it follows that

$$\min\{\gamma_1, \delta_1\} - \max\{\alpha_2, \beta_2\} \leq \delta_1 - \alpha_2 = \min\{\delta_1 - \alpha_2, \gamma_1 - \beta_2\}.$$

case 2 $\gamma_1 - \beta_2 \leq \delta_1 - \alpha_2$. For the same foregoing subcases $(\bar{i})$ through $(\overline{iv})$, it follows that

$$\min\{\gamma_1, \delta_1\} - \max\{\alpha_2, \beta_2\} \leq \delta_1 - \alpha_2 = \min\{\delta_1 - \alpha_2, \gamma_1 - \beta_2\}.$$

$\square$

A.4 Concluding remarks

We have discussed a useful convex relaxation technique which can be applied to various extensions to the form of product of variables and/or functions. Also we found that if one of the variable or the function is optimal at the boundary point under the convex relaxation, then the projection of such an optimal point onto the domain of the original problem produces an optimal solution to the original problem. In the sense of better bound, we suggested a tighter relaxation of the product form by introducing more variables.

APPENDIX B

GLOSSARY OF NOTATION

B.1 Variables

B.1.1 Variables for Network Flows

- x_{imjnv} : Arc flow variable is 1 if harbor arrivals (i, m) and (j, n) are directly connected in ship v's route; otherwise, 0.

- z_{imv} : Route end indicator variable is 1 if (i, m) is the end of the route for ship v; otherwise, 0.

- y_{im} : Slack variable is 1 if (i, m) is not visited; otherwise, 0.

B.1.2 Variables for Loading and Unloading

- l_{imvk} : Load onboard in the compartment for product k of ship v when leaving (i, m).

- q_{imvk} : Quantity of product k loaded into or unloaded from ship v's in position (i, m).

B.1.3 Variables for Time Aspect

- o_{imvk} : Binary variable is 1 if product k is loaded or discharged at harbor arrival (i, m) by ship v; otherwise, 0.

- t_{Eim} : Ending service time at (i, m).

B.1.4 Variables for Inventories

- s_{imk} : Stock level of product k in harbor i when service starts at (i, m). Also we know the value of s_{imk} for all $(i, m) \in S_F$.

- s_{Eimk} : Stock level of product k in harbor i when service finishes at (i, m).

B.1.5 Variables for Stock Levels

- p_{im}: Binary variable is 0 of there are two or more ships at harbor i during the the m-th arrival; otherwise, 0.

B.2 Sets

B.2.1 Sets for Network Flows

- S_T : Set of all harbor arrivals (i, m) for $i \in H_T$ and $m \in M_i$.

- H_T : Set of total harbors.

- M_i : Set of arrival numbers at harbor i.

- S_0 : Set of initial positions $\{(i_v, m_v)|v \in V\}$. If more than one ship starts from the same harbor, then they are assigned a departure sequence number m_v; otherwise, $m_v = 1$.

- V : Set of available ships indexed by v.

- H_v : Set of harbors that can be visited by ship v.

B.2.2 Sets for Loading and Unloading

- A_v : Set of all feasible arcs for ship v.

- K : Set of products.

- K_v : Set of products that ship v can carry.

- K_i^H : Set of products that harbor i handles.

B.3 Parameters

B.3.1 Parameters for Network Flows

- i_v : Starting harbor of vessel v.

- m_v : Assigned arrival sequence number for vessel v in harbor i_v.

B.3.2 Parameters for Loading and Unloading

- J_{jk} : Indicator variable is 1 (if product k is loaded at harbor j), 0 (if product k passes through harbor j), or -1 (if product k is unloaded at harbor j).

- Q_{vk} : Quantity of product k on ship v at start of planning horizon.

- CAP_{vk} : Capacity of the compartment for product k in ship v.

B.3.3 Parameters for Time Aspect

- TQ_{ik} : Time required to load a unit of product k at harbor i.

- W_i : Setup time to change products for loading and unloading at harbor i.

- T_{ijv} : Sailing time from harbor i to harbor j.

B.3.4 Parameters for Inventories

- IS_{ik} : Initial stock level of product k at harbor i.

- R_{ik} : The consumption or production rate for product k in harbor i.

- S_{MNik} : Minimum stock level at harbor i.

- S_{MXik} : Maximum stock level at harbor i.

- T : Length of planning period.

B.3.5 Parameters for Objective function

- C_{ijv} : Cost for ship v to sail from harbor i to harbor j.

- C_{Wik} : Loading and unloading charges incurred at harbor i for product k.

APPENDIX C

STRUCTURE OF THE PROBLEM

Table 15: Staircase structure of the problem with four coupling constraints and two sub polyhedron $\mathcal{P}^1$ and $\mathcal{P}^2$.

	x_{iminv}	z_{imv}	o_{imvk}	l_{imvk}	q_{imvk}	y_{im}	o_{imk}	p_{im}	s_{imk}	s_{Eimk}	q_{imk}	w^r_{im}	v^r_{im}	t_{im}	t_{Eim}
(14)	✓					✓									
(23)	✓													✓	✓
(31)					✓						✓				
(32)			✓				✓								
(12)	✓	✓													
(13)		✓													
(16)	✓			✓	✓										
(17)	✓			✓	✓										
(18)				✓	✓										
(19)	✓			✓											
(20)			✓		✓										
(15)						✓									
(21)														✓	
(24)									✓					✓	
(26)								✓						✓	✓
(27)								✓						✓	✓
(c1.a)-(c1.i)								✓	✓	✓		✓		✓	✓
(28)									✓						
(c2.a)-(c2.i)						✓				✓			✓		✓
(33)						✓	✓				✓			✓	✓
(34)									✓	✓	✓			✓	✓

Table 16: Staircase structure of the matrix defining polyhedron $\mathcal{P}^1$ (ship constraints) after rearranging rows and columns.

	x_{iminv_1}	z_{imv_1}	o_{imv_1k}	l_{imv_1k}	q_{imv_1k}	x_{iminv_2}	z_{imv_2}	o_{imv_2k}	l_{imv_2k}	q_{imv_2k}
$(12)_{v_1}$	✓	✓								
$(13)_{v_1}$		✓								
$(16)_{v_1}$	✓			✓	✓					
$(17)_{v_1}$	✓			✓	✓					
$(18)_{v_1}$				✓	✓					
$(19)_{v_1}$	✓			✓						
$(20)_{v_1}$			✓		✓					
$(12)_{v_2}$						✓	✓			
$(13)_{v_2}$							✓			
$(16)_{v_2}$						✓			✓	✓
$(17)_{v_2}$						✓			✓	✓
$(18)_{v_2}$									✓	✓
$(19)_{v_2}$						✓			✓	
$(20)_{v_2}$								✓		✓

Table 17: Staircase structure of the matrix defining polyhedron $\mathcal{P}^2$ (harbor constraints) after rearranging rows and columns.

	y_{i_1m}	o_{i_1mk}	p_{i_1m}	s_{i_1mk}	s_{Ei_1mk}	q_{i_1mk}	$w^r_{i_1m}$	$v^r_{i_1m}$	t_{i_1m}	t_{Ei_1m}	y_{i_2m}	o_{i_2mk}	p_{i_2m}	s_{i_2mk}	s_{Ei_2mk}	q_{i_2mk}	$w^r_{i_2m}$	$v^r_{i_2m}$	t_{i_2m}	t_{Ei_2m}
$(15)_{i_1}$	✓																			
$(21)_{i_1}$									✓											
$(24)_{i_1}$				✓					✓											
$(26)_{i_1}$			✓						✓	✓										
$(27)_{i_1}$			✓						✓	✓										
$(c1)_{i_1}$			✓	✓	✓		✓		✓	✓										
$(28)_{i_1}$				✓																
$(c2)_{i_1}$	✓				✓			✓		✓										
$(33)_{i_1}$	✓	✓				✓			✓	✓										
$(34)_{i_1}$				✓	✓	✓			✓	✓										
$(15)_{i_2}$											✓									
$(21)_{i_2}$																			✓	
$(24)_{i_2}$														✓					✓	
$(26)_{i_2}$													✓						✓	✓
$(27)_{i_2}$													✓						✓	✓
$(c1)_{i_2}$													✓	✓	✓		✓		✓	✓
$(28)_{i_2}$														✓						
$(c2)_{i_2}$											✓				✓			✓		✓
$(33)_{i_2}$											✓	✓				✓			✓	✓
$(34)_{i_2}$														✓	✓	✓			✓	✓

REFERENCES

[1] Al-Khayyal, F., "Generalized bilinear programming : Part 1. models, applications and linear programming relaxation," *European Journal of Operational Research*, vol. 60, pp. 309–314, 1992.

[2] Al-Khayyal, F. and Falk, J., "Jointly constrained biconvex programming," *Mathematics of Operations Research*, vol. 8, pp. 273–286, 1983.

[3] Ballou, R., *Business Logistics Management, 4th.* Upper Saddle River, New Jersey: Prentice Hall, 1998.

[4] Bausch, D., Brown, G., and Ronen, D., *Elastic Set Partitioning - A Powerful Tool for Scheduling Transportation of Oil and Gas.* Advances in Operations Research in the Oil and Gas Industry, M. Breton and G. Zaccour (Eds.), Paris: Tecnip, 1991.

[5] Bausch, D., Brown, G., and Ronen, D., "Scheduling short term marine transport of bulk products," *Maritime Policy and Management*, vol. 25, pp. 335–348, 1998.

[6] Bazaraa, M. S., Sherali, H., and Shetty, C., *Nonlinear programming : Theory and algorithms, 2nd.* New York: Wiley, 1993.

[7] Benford, H., "A simple approach to fleet deployment," *Maritime Policy and Management*, vol. 8, pp. 223–228, 1981.

[8] Bertsimas, D. and Tsitsiklis, J., *Introduction to Linear Optimization.* Belmont, MA: Athena Scientific, 1997.

[9] Boffey, T., Edmond, E., Hinxman, A., and Pulsglove, C., "Two approaches to scheduling container ships with an application to the north atlantic route," *Journal of the Operational Research Society*, vol. 30, pp. 413–425, 1979.

[10] Brown, G., Dell, R., and Farmer, R., "Scheduling cost guard district cutters," *Interfaces*, vol. 26, pp. 59–72, 1996.

[11] Brown, G., Goodman, C., and Wood, R., "Annual scheduling of atlantic fleet naval combatants," *Operations Research*, vol. 38, pp. 249–259, 1990.

[12] Brown, G., Graves, G., and Ronen, D., "Scheduling ocean transportation of crude oil," *Management Science*, vol. 33, pp. 335–346, 1987.

[13] Chajakis, E., *Management Science for Marine Petroleum Logistics.* Decision Making: Recent Developments and Worldwide Applications, S. H. Zanakis, G. Doukidis and C. Zopounidis (Eds.), Dordrecht: Kluwer Academic Publishers, 2000.

[14] Cho, S., , and Perakis, A., "An improved formulation for bulk cargo ship scheduling with a single loading port," *Maritime Policy Management*, vol. 28, pp. 339–345, 2001.

[15] Christiansen, M., "Decomposition of a combined inventory and time constrained ship routing problem," *Transportation Science*, vol. 33, pp. 3–16, 1998.

[16] Christiansen, M., Fagerholt, K., and Ronen, D., "Ship routing and scheduling: Status and perspectives," *Transportation Science*, vol. 38, pp. 1–18, 2004.

[17] Christiansen, M. and Nygreen, B., "A method for solving ship routing problems with inventory constrains," *Annals of Operations Research*, vol. 81, pp. 357–378, 1998.

[18] Christiansen, M. and Nygreen, B., "Modelling path flows for a combined ship routing and inventory management problem," *Annals of Operations Research*, vol. 82, pp. 391–421, 1998.

[19] Cline, A., Kong, D., and Meyering, J., "Routing and scheduling cost guard buoy tenders," *Interfaces*, vol. 22, pp. 56–72, 1992.

[20] Darby, D., R.F. Fink, G. M., and Smith, J., "An intellegent system for us coast guard cutter scheduling," *European Journal of Operational Research*, vol. 87, pp. 574–585, 1995.

[21] Fagerholt, K., "Optimal fleet design in a ship routing problem," *International Transactions in Operational Research*, vol. 6, pp. 453–464, 1999.

[22] Fagerholt, K., "A simulation study on the design of flexible cargo holds in small sized bulk ships," *Maritime Policy and Management*, vol. 26, pp. 105–109, 1999.

[23] Fagerholt, K., "Ship scheduling with soft time windows : An optimization based approach," *European Journal of Operational Research*, vol. 131, pp. 559–571, 2001.

[24] Fagerholt, K. and Christiansen, M., "A combined ship scheduling and allocation problem," *Journal of the Operational Research Society*, vol. 51, pp. 834–842, 2000.

[25] Fagerholt, K. and Christiansen, M., "A traveling salesman problem with allocation, time window and precedence constraints - an application to ship scheduling," *International Transactions in Operational Research*, vol. 7, pp. 231–244, 2000.

[26] Fisher, M., *Vehicle routing.* Handbooks in Operations Research and Management Science Vol 8, Network Routing, M.O. Ball, T.L. Magnati, G.L. Nemhauser (Eds.), Amsterdam: Elsevier Science Publishers, 1995.

[27] Fisher, M. and Rosenwein, M., "An interactive optimization system for bulk-cargo ship scheduling," *Naval Research Logistics*, vol. 36, pp. 27–42, 1989.

[28] Jetlund, A. and Karimi, I., "Improving the logistics of multi-compartment chemical tankers," *Computers and chemical Engineering*, vol. 28, pp. 1267–1283, 2004.

[29] Kim, S. and Lee, K., "An optimization-based decision support system for ship scheduling," *Computers and Industrial Engineering*, vol. 33, pp. 689–692, 1997.

[30] Larson, R., "Transporting sludge to the 106-mile site: An inventory/routing model for fleet sizing and logistics system design," *Transportation Science*, vol. 22, pp. 186–198, 1988.

[31] McCord, H., Lee, Y., and Lo, H., "Ship routing through altimetry-derived ocean currents," *Transportation Science*, vol. 33, pp. 49–67, 1999.

[32] Miller, M., "An interactive, computer-aided ship scheduling system," *European Journal of Operational Research*, vol. 32, pp. 363–379, 1987.

[33] Nemhauser, G. and Wolsey, L., *Integer and Combinatorial Optimization.* New York, NY: John Wiley & Sons, 1998.

[34] Nemhauser, G. and Yu, P., "A problem in bulk service scheduling," *Operations Research*, vol. 20, pp. 813–819, 1972.

[35] Papadakis, N. and Perakis, A., "A nonlinear approach to the multiorigin, multi-destination fleet deployment problem," *Naval Research Logistics*, vol. 36, pp. 515–528, 1989.

[36] Perakis, A., "A second look at fleet deployment," *Maritime Policy and Management*, vol. 12, pp. 209–214, 1985.

[37] Perakis, A. and Papadakis, N., "Fleet deployment optimization models. part 1," *Maritime Policy and Management*, vol. 14, pp. 127–144, 1987.

[38] Perakis, A. and Papadakis, N., "Fleet deployment optimization models. part 2," *Maritime Policy and Management*, vol. 14, pp. 145–155, 1987.

[39] Psaraftis, H., "Foreword to the focused issue on maritime transportation," *Transportation Science*, vol. 33, 1999.

[40] Psaraftis, H., Solomon, M., Magnati, T., and Kim, T., "Routing and scheduling on a shoreline with release times," *Management Science*, vol. 36, pp. 212–223, 1990.

[41] Rana, K. and Vickson, R., "A model and solution algorithms for optimal routing of a time-chartered containership," *Transportation Science*, vol. 22, pp. 83–94, 1988.

[42] Rana, K. and Vickson, R., "Routing container ships using lagrangean relaxation and decomposition," *Transportation Science*, vol. 25, pp. 201–214, 1991.

[43] Ronen, D., "Cargo ship routing and scheduling : Servey of models and problems," *European Journal of Operational Research*, vol. 12, pp. 119–126, 1983.

[44] Ronen, D., "Short-term scheduling of vessels for shipping bulk or semi-bulk commodities originating in a single area," *Operations Research*, vol. 34, pp. 164–173, 1986.

[45] Ronen, D., "Ship scheduling : The last decade," *European Journal of Operational Research*, vol. 71, pp. 325–333, 1993.

[46] Ronen, D., "Marine inventory routing: shipments planning," *Journal of the Operational Research Society*, vol. 53, pp. 108–144, 2002.

[47] Schrady, D. and Wadsworth, D., "Naval combat logistics support system," *Journal of the Operational Research Society*, vol. 42, pp. 941–948, 1991.

[48] Sherali, H. and Adams, W., *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems.* Boston: Kluwer Academic, 1998.

[49] Sherali, H., Al-Yakoob, S., and Hassan, M., "Fleet management models and algorithms for an oil-tanker routing and scheduling problem," *IIE Transactions*, vol. 31, pp. 395–406, 1999.

[50] Sherali, H. and Alameddine, A., "An explicit characterization of the convex envelope of a bivariate bilinear function over special polytopes," *Annals of Operations Research*, vol. 25, pp. 197–210, 1990.

[51] Society, K. V. A., *Introduction to Vessel and Marine Engineering 6th*. Seoul: Dongmyung, 1998.

[52] Tawarmalani, M. and Sahinidis, N. V., *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications*. Boston, MA: Kluwer Academic, 2002.

[53] Toth, P. and Vogo, D., *The vehicle routing problem*. Society for Industrial and Applied Mathematic, 2002.

[54] United States Energy Information Administration, *South China Sea Full Report*, Mar. 2003.

[55] U.S. Department of Transportation, Bureau of Transportation Statistics, Maritime Administration, U.S. Coast Guard, Washington, DC, *Maritime Trade and Transportation 1999*, 1999.

[56] Wang, C. and Chretienne, P., "Maritime routing-problems and dynamic programing," *Rairo-Recherche Operationalle-Operations Research*, vol. 27, pp. 61–76, 1993.

[57] Williams, T., "Heuristic scheduling of ship replenishment at sea," *Transportation Science*, vol. 43, pp. 11–18, 1992.