Optimisation of Vendor-Managed Inventory Systems

by

Tareq Ahmed Al-Ameri

May 2005

A thesis submitted for the degree of Doctor of Philosophy of the University of London and for the Diploma of Membership of the Imperial College London

Centre for Process Systems Engineering
Department of Chemical Engineering and Chemical Technology
Imperial College London
London SW7 2BY, UK

To my two fathers,

إلى روح فقيد الوطن الغالي المغفور له بإذن الله تعالى... الشيخ زايد بن سلطان آل نهيان (رحمه الله)

إلى والدي العزيز... أحمد محمد سعيد العامري

ABSTRACT

Vendor managed inventory (VMI) has evolved as a means for manufacturers to outsource non-core processes such as administration and inventory management. In VMI systems, the vendor takes the responsibility of replenishing manufacturers' sites. As a result, the vendor can plan its production and distribution strategy to maintain adequate levels of material at manufacturers' sites. On the other hand, the manufacturer can concentrate on core processes resulting in fixed cost minimisation and better customer service.

The research project aims at designing a dynamic VMI system. In this system, the entire supply chain performance is optimised in terms of production planning at vendor's site, distribution strategy, and inventory management at manufacturer's site. We also explore some of the complications involved in setting up such a system.

The VMI system is modelled as a mixed-integer linear program (MILP) using discrete-time representation. The mathematical representation follows the resource-task network (RTN) formulation. To address the complexity of the problem, different optimisation-based solution algorithms are proposed and compared in terms of solution quality and CPU time.

First, the problem is solved directly using an exact detailed model. Secondly, an iterative procedure combines a novel aggregate model with the detailed model to provide aggregate pre-matches for the detailed binary variables. Finally, a novel rolling horizon approach that simultaneously combines the aggregate and the detailed models is designed to solve the problem. The entire VMI system is tested with real-life data taken from gas and oil companies' industrial case studies.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to H.H. Sheikh Khalifa Bin Zayed Al-Nahyan, President of the United Arab Emirates and Chairman of the Supreme Petroleum Council, and H.H. Sheikh Mohammed Bin Zayed Al-Nahyan, Crown Prince of Abu Dhabi and member of the Supreme Petroleum Council, for establishing the Petroleum Institute in Abu Dhabi and for their unlimited support to UAE nationals.

I would like to thank my sponsor, Abu Dhabi National Oil Company (ADNOC) for the financial support. Special thanks to H.E. Yousef Omair bin Yousef, Secretary General of the Supreme Petroleum Council and ADNOC CEO, for giving me the opportunity to be a member of the prestigious Imperial College Institution.

I would like to thank my supervisors at the Centre for Process Systems Engineering, Prof. Nilay Shah and Dr. Lazaros Papageorgiou for their professional guidance throughout my PhD research.

I would also like to thank my ADNOC colleagues at Imperial, Ali, Khaled, and Mohamed for their help during my transfer and PhD examinations. Many thanks to all good friends that I have made in the Centre.

I would like to express my appreciation to my wife for her continuing support throughout my studies.

Finally, I want to express my respect and gratitude to my family, especially my parents for their unwavering financial and moral support during my undergraduate as well as my postgraduate studies.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Vendor Managed Inventory

Lieberman and Hillier (2001) define supply chains as:

" ... *A network of facilities that procure raw materials, transform them into intermediate goods and then final products, and finally deliver the products to customers through a distribution system that includes an inventory system. Thus, it spans procurement, manufacturing, and distribution, with effective inventory management* ... "

From this simple definition we can see that the survival of members (or companies) in this network depends highly on the co-operation and co-ordination between them. However, usually this is not the case and instead of co-ordination, companies in the supply chain compete against each other. Therefore, each company tries to achieve cost reductions at the expense of other companies in their supply chain. Consequently, the supply chain suffers the increased costs that ultimately affect all members. Only recently have companies started to realise that they need to compete as supply chains instead of competing as single entities (Christopher, 1998).

Moreover, achieving effective inventory management is a key element of optimising supply chain performance. Actually, because of the high associated costs, one objective of supply chain management is to minimise inventory buffers between different parts of the chain. Inventory minimisation can be achieved by proper information sharing about demand forecasts and inventory levels. This is the concept of VMI, which is the subject of this project.

1.1.1 Traditional systems

For simplicity, we can generally divide the inventory management methodologies into two main categories. The first category is the *reactive* methodology which is the traditional policy used in supply chains now. The reactive policy gets its name from responding (reacting) to externally generated orders. Much research has been conducted for reactive systems to find the optimal ordering policy. By optimal policy, we mean the frequency of orders and quantities ordered. The second category is the *proactive* policy, which is the new evolving methodology replicated in VMI systems. The proactive policy gets its name by co-ordinating its plan to take action without receiving externally generated orders.

Imagine a supply chain with only one manufacturer that receives raw materials from only one vendor (supplier). Traditionally, the manufacturer is responsible for monitoring inventory and placing purchase orders when material stocks fall below a certain level. When orders are received, the vendor reacts by dispatching material towards the manufacturer. After some delay in processing and distribution (i.e. *lead time*), the material arrives at manufacturer's sites using the available mode of transportation. In this system, both the vendor and the manufacturer act independently to minimise their costs, which might ultimately affect their supply chain performance.

The manufacturer's goal in traditional (reactive) systems is:

- To find the optimal order cycle to avoid stock-outs
- To find the optimal order quantity that minimises holding costs

The vendor's goal in traditional (reactive) systems is:

- To find the optimal production plan and inventory policy (at the vendor's site) to cover all orders
- To respond to orders in an efficient way that minimises the transportation costs and lead times of delivering materials to the manufacturer
- To try and anticipate orders and build up stocks.

Traditional systems can lead to inefficiencies both for vendors and manufacturers:

- The vendor lacks important information about order requirements received from the manufacturer. As a result, the vendor will tend to keep more inventories at its site to cover those orders, which results in high holding costs.
- The vendor might alter its production plans regularly to adapt to some short-term orders and demand variability.
- The manufacturer utilises some of its resources for non-core processes such as inventory management and administration. As a result, the total costs can increase and production quality might be compromised.
- According to a survey by the European logistics association, the rate of late and incomplete deliveries is approximately 10% (Kaipia *et al.*, 2002). Hence, the manufacturer's customer service is usually affected by lead-time variations for different vendors. As a result, the manufacturer might lose the loyalty of some customers.

1.1.2 Definition of VMI

In a VMI strategy, the vendor is responsible for proactively maintaining adequate levels of materials at the manufacturer's site. Hence, no pressure is put on vendors to respond to orders accurately. The VMI contract (Taras, 2001) between the vendor and the manufacturer ensures that:

- The manufacturer will share demand forecasts and sales information with the vendor. This can be achieved by sending the manufacturer's point of sale (POS) data to the vendor.
- The vendor will have access to the manufacturer's inventory system. Electronic Data Interchange (EDI) is employed for any data exchange between the two parties.
- The vendor is responsible for replenishing the manufacturer's site with material. A mutually agreed-upon policy is set-up for the replenishment policy (usually an upper and lower bound on inventory level). Then, the manufacturer has to trust the vendor to take over its inventory management according to that policy.

Therefore, no orders are received anymore in a VMI strategy. Instead, the vendor is responsible for maintaining inventory levels at the manufacturer's site within the agreed-upon bounds. Some researchers (Christopher, 1998) refer to VMI as Co-managed inventory due to the necessary co-operation in those systems. For the purpose of this thesis, we will only use the term "VMI".

VMI strategies necessitate the use of Information Technology (IT) for their application. Such IT tools are computer-based distribution optimisers as well as material tracking software. In addition, information sharing utilises the Internet and EDI systems. Fortunately, some of these tools are commonly used in many companies. Therefore, the amount of investment (from the vendor's perspective) in the implementation of VMI systems is relatively low. However, the manufacturer is required to invest in its IT infrastructure to make production schedules and inventory levels more transparent to the vendor (Kuk, 2004).

The VMI scope considered in our work is modified slightly to meet industry requirements. In our VMI system, we deal with continuous products (oil, gas …etc.) rather than discrete products (stationary, parts ...etc.). Hence, holding costs are neglected at customer sites as long as material is always available to meet customer demands and does not exceed storage capacity. A contract between the vendor and the customer guarantees that material will always be available to cover the forecasted demands. Demand forecasting here is the job of the customer, and the vendor agrees only to meet these forecasts for a long period of time (usually a whole year). A penalty cost for not meeting demand forecasts is agreed-upon between the two parties.

1.1.3 Benefits of VMI

It is obvious that VMI is a promising way of managing supply chains. The resulting co-operative system increases the competitiveness on the supply chain scale rather than the single company scale. Hence, both vendors and manufacturers can potentially benefit from VMI strategies.

Potential benefits for the vendor include the following:

- The vendor can co-ordinate its production plans and distribution policies (ahead of time) according to well-known demand forecasts, which results in fewer production schedule alterations and lower transportation costs.
- Instead of keeping safety stocks, the vendor will tend to keep minimum inventory levels which can result in lower stock holding costs at the vendor's site.

Potential benefits for manufacturers include the following:

- By outsourcing non-core processes such as administration and inventory management, the fixed costs associated with such processes are eliminated.
- The manufacturer can concentrate on core processes, which results in better production quality and customer service.
- Inventory levels at manufacturers' sites are minimised and stock outs should decline substantially which result in significant savings.

1.1.4 Conclusion

It is obvious now that VMI is composed of three parts undertaken by the vendor. Those parts are production planning, distribution, and inventory management. A successful VMI strategy optimally integrates those parts. A potentially better overall supply chain performance is attained when VMI is implemented. A detailed comparison between traditional and VMI systems is given in (Toni and Zamolo, 2005).

In our work, production planning refers to the process of optimising the daily production rates of each product at each plant. However, some references (Martin, 1995) refer to the same concept as the "Master Production Schedule" (MPS). Throughout this thesis, we only refer to those rates as production plans.

A substantial part of the modelling effort is spent on the distribution component of VMI systems. Therefore, an extensive introduction and literature review is mainly focused on routing and scheduling problems.

1.2 Transportation Problems

Many applications involve the question "How to optimally transport goods?" Our work involves the optimisation of the distribution system of supply chains using VMI strategies. Moreover, distribution costs represent approximately 10% of the revenues of firms. In addition, distribution costs account for more than 45% of the logistics costs (Laporte and Osman, 1995). Hence, a brief introduction to transportation problems is provided in this section. The simple transportation problem in this section is meant to introduce the routing problems presented in the next section.

Table 1.1: General formulation for transportation tables

		Destinations					
		D_1	D_2	...	D_{n-1}	D_n	Supply
	S_1	c_{11}	c_{12}		$c_{1,n-1}$	c_{1n}	Sup_1
	S_2	c_{21}	c_{22}		$c_{2,n-1}$	c_{2n}	Sup_2
	.						.
Sources	.						.
	.						.
	S_{m-1}	$c_{m-1,1}$	$c_{m-1,2}$		$c_{m-1,n-1}$	$c_{m-1,n}$	Sup_{m-1}
	S_m	c_{m1}	c_{m2}		$c_{m,n-1}$	c_{mn}	Sup_m
	Demand	Dem_1	Dem_2	...	Dem_{n-1}	Dem_n	

Table 1.1 above shows a typical transportation problem table (Lieberman and Hillier, 2001). Product is transported from m different sources/suppliers to n different destinations/customers. Each source S_i, $i \in [1,2...m]$, has a certain supply Sup_i while each destination D_j, $j \in [1,2...n]$, has a certain demand Dem_j. A cost c_{ij} is associated with transporting product(s) from source i to destination j. Nevertheless, some routes between sources and destinations are not allowed. Such routes are given a cost of M (very large number) in the transportation table. Hence, a transportation problem will have m supply availability constraints and n demand requirement constraints. Solving

transportation problems involves finding the optimal quantities x_{ij} transported from sources i to destinations j at costs c_{ij}.

1.2.1 Network representation of the transportation problem

A transportation problem similar to the one shown in Table 1.1 can be represented using network models. Figure 1.1 shows a network representation of a transportation problem with m sources and n destinations. In Figure 1.1, the sources and destinations represent the *nodes* of the network. The possible supply routes between sources and destinations represent the *arcs* of the network. Considering demand to be positive, then supply will have a negative value (or the other way around). In our work, the convention of treating demand as a positive value is enforced.

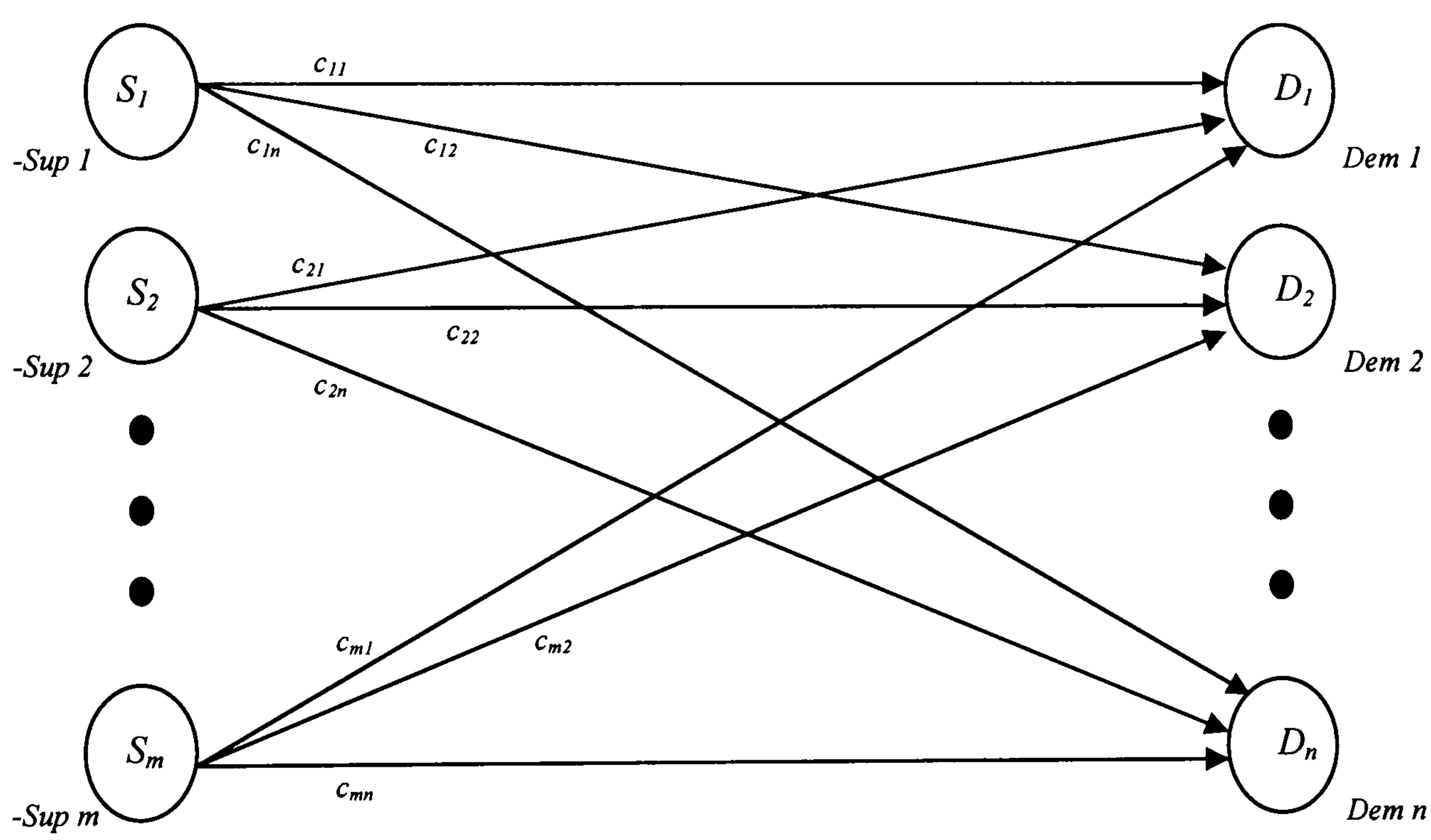

Figure 1.1: Network representation of the transportation problem

A general Linear Programming (LP) formulation takes the form:

Minimise cx

subject to

$$Ax=b$$

$$x \geq 0$$

Where $\boldsymbol{c}$ is a vector of the objective function coefficients, $\boldsymbol{x}$ is a vector of all decision variables, A is the coefficient constraint matrix and $\boldsymbol{b}$ is the right hand side. Following a similar structure, $\boldsymbol{c}$ will represent a vector of the costs c_{ij} and $\boldsymbol{b}$ will represent demand requirements and supply availabilities. As a result, a transportation problem will have the following LP formulation:

Minimise $$Z = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij} X_{ij} \quad (1.1)$$

Subject to

$$\sum_{j=1}^{n} X_{ij} \le Sup_i \qquad \textit{for } i = 1,2,\ldots,m \quad (1.2)$$

$$\sum_{i=1}^{m} X_{ij} = Dem_j \qquad \textit{for } j = 1,2,\ldots,n \quad (1.3)$$

$$X_{ij} \ge 0 \qquad \textit{for all } i \textit{ and } j \quad (1.4)$$

The objective (1.1) minimises the total transportation costs. Constraints (1.2) and (1.3) represent supply availabilities and demand requirements, respectively. Constraints (1.4) ensure positive values for transported quantities. As the formulation shows, the transportation problem has m supply constraints and n demand constraints. The resulting overall constraint coefficients matrix (A) has a total of (m+n) constraints.

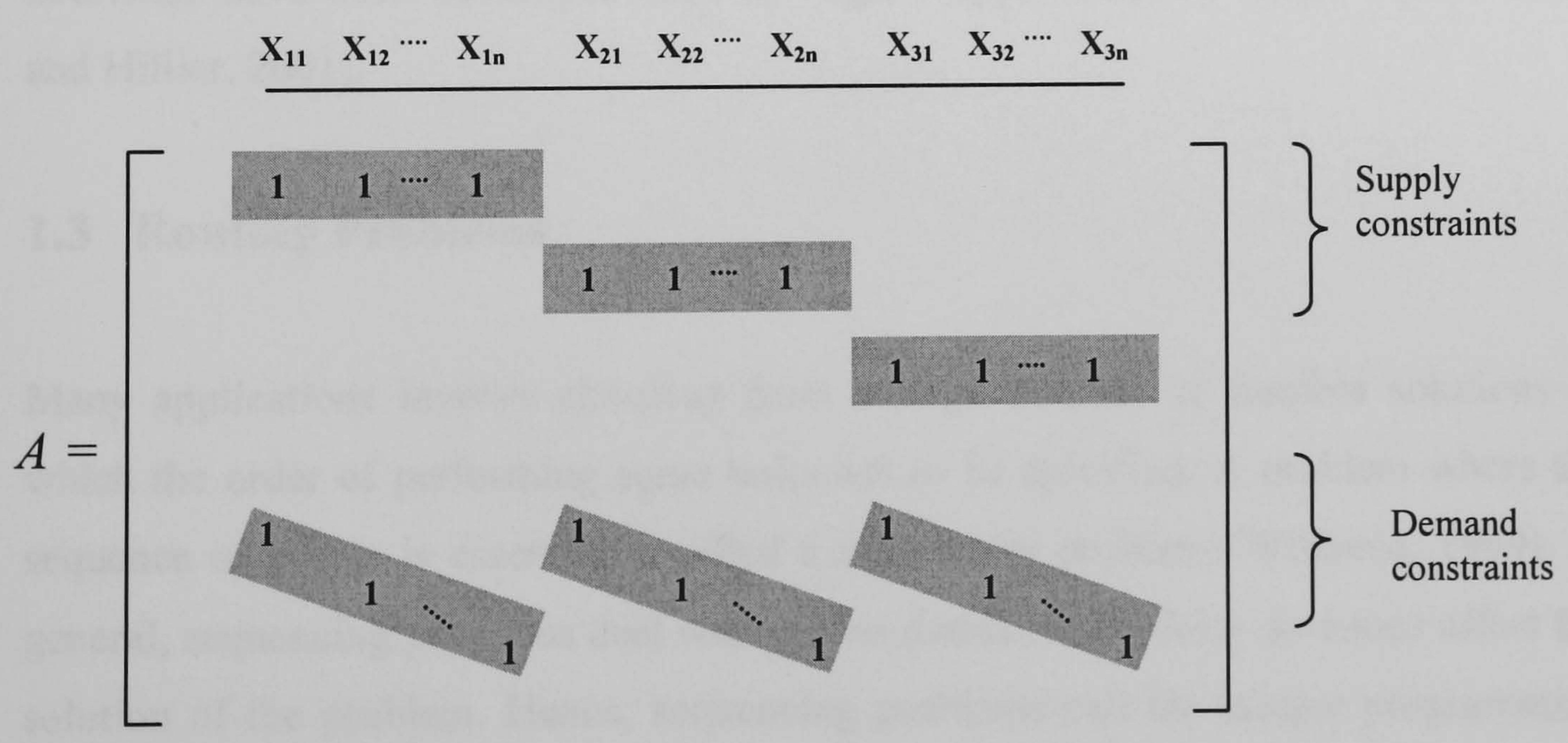

Figure 1.2: Constraint coefficients matrix for the transportation problem

Mathematically speaking, transportation problems have a very distinctive structure where A takes the general form shown in Figure 1.2. If all demand requirements and supply availabilities (Dem_j and Sup_i respectively) are integral, then solving the transportation problem as an LP problem will lead to an integer optimal solution(Williams, 1999; Lieberman and Hillier, 2001). The theoretical reason is explained by Williams (1999). For every cost vector $\boldsymbol{c}$ and integer right hand side vector $\boldsymbol{b}$, A is a *unimodular* matrix. The property of unimodularity assures that the solution variables x_{ij} will have integer values.

1.2.2 Heuristics for transportation problems

Many heuristics are widely used in the area of operations research to solve transportation problems. The most popular heuristic for transportation problems is the North West Corner Rule (NWCR). In this heuristic, the demands of different locations are satisfied using the nearest available supply. This heuristic gets its name from the position of the starting allocation of supply to demand. Supply allocation starts at the uppermost left cell in the transportation table (i.e. northwest corner cell). Usually the results obtained from this method are used as an initial solution to a problem where further optimisation is needed, or as a preliminary solution where only feasible solutions are required. Since NWCR is not a cost-based heuristic, other cost-based heuristics have been developed such as *Vogel's approximation* method (Lieberman and Hillier, 2001).

1.3 Routing Problems

Many applications involve choosing from a large number of feasible solutions in which the order of performing some tasks has to be specified. A problem where the sequence of events is essential is called a *sequencing* problem (Williams, 1999). In general, sequencing problems deal with *yes-no* questions in which decisions affect the solution of the problem. Hence, sequencing problems call for integer programming (IP) methods in order for these decisions to be incorporated in models.

The transportation problem (similar to the form shown in the previous section) is a special case of a sequencing problem. A transportation problem involves the decision of "what supplier should service each customer with a certain demand using the available supply"? Therefore, a decision has to be made about the supplier-customer combinations in addition to the quantities transported. It should be enforced that each customer receives a discrete (integer) quantity of the product. Consequently, we end up with an MILP problem, which can be difficult to solve. However, the special structure of the transportation problem explained above ensures that all solutions will take integer values for integer supply availabilities and demand requirements. The LP solution to the transportation problem can be viewed as a solution to an IP problem. In other words, if the variable x_{ij} takes a nonzero value, then the answer to the question "Is source i supplying destination j?" will be "*yes*" and vice versa. The value of x_{ij} in the final solution represents the quantity transported from i to j.

Imagine a transportation problem in which suppliers have to provide customers with a product. However, in addition to the feasible routes between suppliers and customers, feasible routes exist between customers themselves. A network representation of such a problem with two suppliers (S_1 and S_2) and 3 customers (D_1, D_2, D_3) is shown in Figure 1.3.

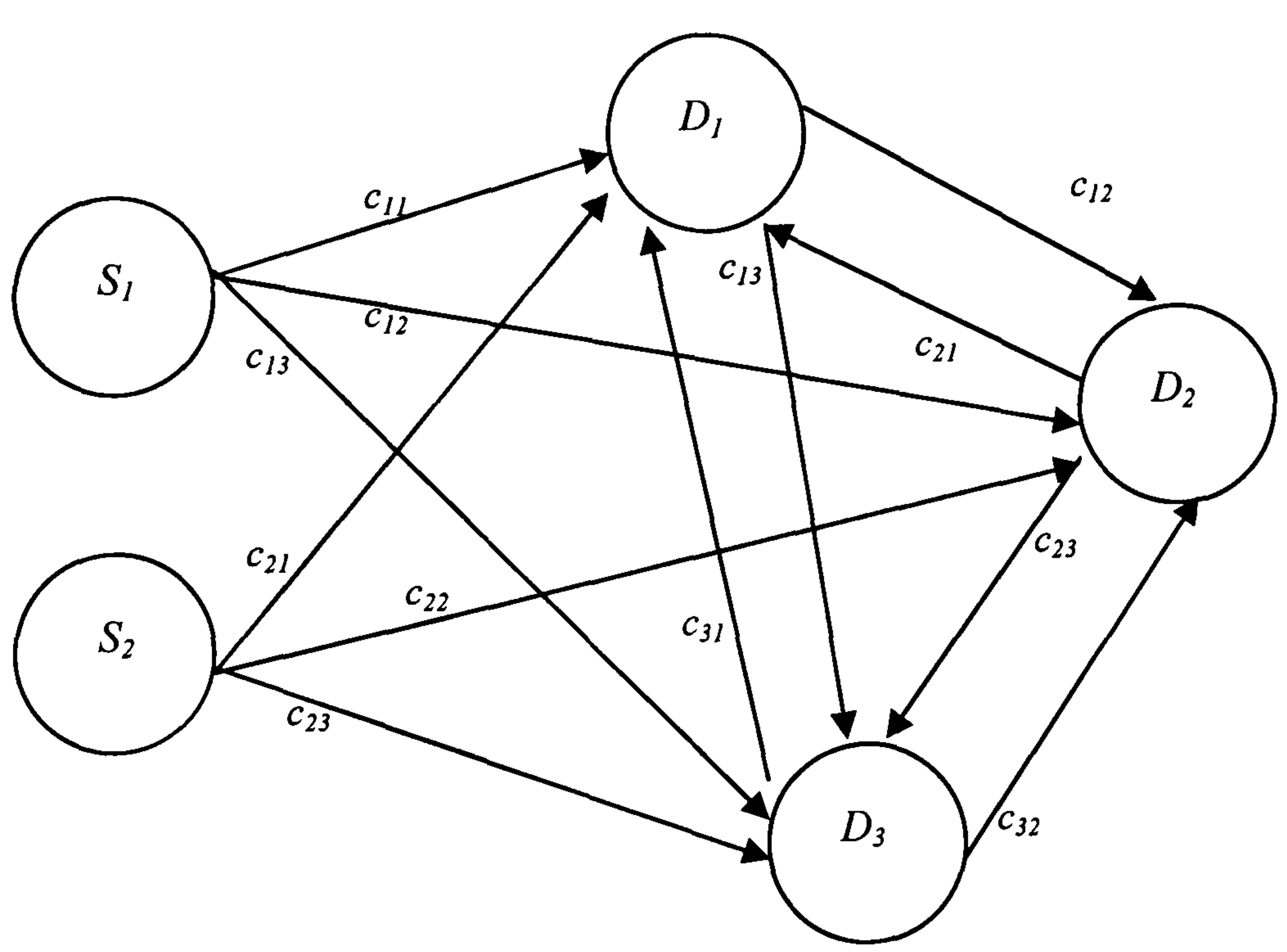

Figure 1.3: Network representation with feasible customer-customer routes

For example, a decision of how to transport a product from S_1 to D_2 involves finding out the optimal route between these nodes. Five possibilities are available: route S_1-D_2, route S_1-D_1-D_2, route S_1-D_3-D_2, route S_1-D_1-D_3-D_2, and S_1-D_3-D_1-D_2. A total transportation cost is associated with each route. Based on the routing decisions, the quantities transported can be obtained. These quantities involve unloading at many different customer locations. Therefore, the only way to solve this kind of problem is via different formulations of the problem in which the optimal route (sequence) is chosen. It is obvious that solving a totally connected network problem with n nodes involves choosing from a large number of feasible routes (n^2-n). As the complexity of the problem increases, a feasible rather than optimal solution can be enough.

In our work, we are interested in sequencing problems to solve transportation problems. A transportation problem with a large number of feasible routes/sequences is called a *routing* problem. The name comes from finding the optimal route among many feasible routes. One of the earliest and most important routing problems is the Travelling Salesman Problem (TSP).

1.3.1 The travelling salesman problem

The TSP has emerged as one of the most challenging routing problems in operations research. The idea behind the TSP is simple. A salesman has to start from home and visit a set of cities and return back home using the minimum distance (cost). The simplicity of the idea does not fairly reflect the complexity of the formulation or the solution process.

1.3.1.1 Definition of the TSP

Using graph theory, the TSP can be defined as follows (Laporte, 1992): For a graph $G=(Y,A)$, define a set of Y vertices and a set of A arcs between those vertices. Let c_{ij} be the travel cost between vertices i and j. The objective of the TSP is finding the least-cost route that passes through each node only once.

1.3.1.2 Formulation of the TSP

There are many formulations to solve TSP problems in which IP is utilised. In general, all formulations apply the same constraints. The first and trivial constraint is the number of visitations to each node (one visit only). The second constraint is the elimination of sub-tours. Sub-tours are tours on subsets of less than n nodes. The major differences between TSP formulations appear in the sub-tour elimination constraints. Here we only show the earliest TSP mathematical formulation by Dantzig *et al.* (1954):

Let $$x_{ij} = \begin{cases} 1 & \textit{if route between } i \textit{ and } j \textit{ is taken} \\ 0 & \textit{otherwise} \end{cases}$$

Minimise $$Z = \sum_{i \neq j}^{n} c_{ij} x_{ij} \tag{1.5}$$

Subject to

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad \textit{for } i = 1,2,\ldots,n \tag{1.6}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad \textit{for } j = 1,2,\ldots,n \tag{1.7}$$

$$\sum_{i,j \in S}^{n} x_{ij} \leq |S| - 1 \qquad \textit{for } S \subset V, 2 \leq |S| \leq n-2 \tag{1.8}$$

$$x_{ij} \in \{0,1\} \qquad \textit{for all } i \textit{ and } j,\ i \neq j \tag{1.9}$$

Total travel cost (or distance) is minimised by the objective function in (1.5). Constraints (1.6) ensure that every node is entered only once. Constraints (1.7) ensure that every node is exited only once. Constraints (1.8) ensure the elimination of sub-tours, because having a sub-tour on a subset S and $|S|$ arcs and nodes, will violate these constraints. Note that constraint (1.8) prohibits sub-tours over *n-2* nodes since sub-tours over one node and *n-1* nodes are handled by constraints (1.6) and (1.7). Constraints (1.9) represent the binary conditions on x_{ij}.

1.3.1.3 Applicability of the TSP to our work

The TSP has many applications in operations research (OR) and its logic can be applied to many other non-routing problems such as *paper cutting* and *job sequencing*. A number of exact and heuristic algorithms have been utilised to solve the TSP. A survey of several algorithms used to solve the TSP is given by Laporte (1992). A more recent survey can also be found in (Burkard *et al.*, 1998).

As far as our work is concerned, the interest in the TSP arises because of the following:

- The TSP is used as an introduction to other routing problems such as vehicle routing problems and ship scheduling.
- The aggregate model in our work incorporates some TSP structures such as sub-tour elimination constraints (see Chapter 4).
- In ship scheduling problems, TSP problems with extra constraints are solved to generate candidate schedules. Those TSP-generated schedules for each ship are further optimised within a ship scheduling context to find exact timings as well as optimum speeds. Christiansen and Fagerholt (2001) discuss in detail how a travelling salesman problem with capacity constraints, time windows, and precedence is solved in a case of ship scheduling.

In general, the TSP is considered to be the mother of all routing problems. All other routing problems (such as vehicle routing problems) can be considered to be TSP problems with extra constraints. It is important to note that the TSP is only a routing problem. In other words, the time dimension is not considered in such problems. Laporte and Osman (1995) give a bibliography with a classification of all different routing problems.

1.3.2 Vehicle routing problems

The vehicle routing problem (VRP) is the problem of organising a fleet of vehicles to deliver some product(s) from depot(s) to customers using a minimum cost.

Table 1.2 shows a classification of the VRP by Assad (1988) and Desrochers *et al.* (1990).

1.3.2.1 Definition of the VRP

Using graph theory, the basic VRP can be defined as follows (Jalisi, 2000): For a graph $G=(Y,A)$, define a set of Y vertices and a set of A arcs between those vertices. Let $Y=\{y_i | i=1,...,n\}$ be a set of n nodes where $i=1$ refers to the depot node and $i=2...n$ refer to customer nodes. Let $V=\{v_k | k=1,...,m\}$ be a set of m vehicles. Let c_{ij} be the travel cost between nodes y_i and y_j. Each customer y_i has a certain demand requirement of q_i. Each vehicle v_k has a capacity of Q_k. The objective of the VRP is finding the least cost route for vehicles so that all customers' demand is satisfied. As mentioned above, the VRP can be considered to be a special case of the TSP. If $(m=1)$ and $(Q_1 \geq \sum_{i=2}^{n} y_i q_i)$, then the VRP reduces to the TSP. Similar to the TSP, the time dimension is not considered in the VRP. Instead, a total demand per customer is satisfied for each routing problem.

Table 1.2: Classification of the vehicle routing problem

<table>
<tr><td>Commodity</td><td colspan="2">1) Single commodity
2) Multiple commodities</td></tr>
<tr><td>Objective</td><td colspan="2">1) Minimise distance
2) Minimise number of vehicles
3) Minimise total transportation cost</td></tr>
<tr><td>Nature of demand</td><td colspan="2">1) Pure pickups
2) Pure deliveries
3) Mixed</td></tr>
<tr><td>Information on Demand</td><td colspan="2">1) Stochastic
2) Deterministic</td></tr>
<tr><td rowspan="3">Vehicles</td><td>Fleet size</td><td>1) Fixed
2) Variable</td></tr>
<tr><td>Capacity constraints</td><td>1) Identical
2) Different</td></tr>
<tr><td>Route duration</td><td>1) No durations
2) Identical durations
3) Different durations</td></tr>
</table>

1.3.2.2 Formulation of the VRP

The VRP has many different formulations (Jalisi, 2000). Here, we present the formulation related to the TSP.

Let $x_{ijk} = \begin{cases} 1 & \textit{if vehicle k visits customer } x_j \textit{ after customer } x_i \\ 0 & \textit{otherwise} \end{cases}$

Minimise
$$Z = \sum_{i \neq j}^{n} \sum_{k}^{m} c_{ij} x_{ijk} \tag{1.10}$$

Subject to

$$\sum_{i=1}^{n} \sum_{k=1}^{m} x_{ijk} = 1 \qquad \textit{for } j = 1,2,\dots,n \tag{1.11}$$

$$\sum_{i=1}^{n} x_{ipk} - \sum_{j=1}^{n} x_{pjk} = 0 \qquad \textit{for } k = 1,\dots,m \;\&\; p=1,\dots,n \tag{1.12}$$

$$\sum_{i \neq j}^{n} q_i x_{ijk} \leq Q_k \qquad \textit{for } k = 1,2,\dots,m \tag{1.13}$$

$$\sum_{j=2}^{n} x_{1jk} = 1 \qquad \textit{for } k = 1,2,\dots,m \tag{1.14}$$

$$y_i - y_j + n \sum_{k=1}^{m} x_{ijk} \leq n-1 \qquad \textit{for } i \neq j \;\&\; i,j=2,\dots,n \tag{1.15}$$

$$x_{ijk} \in \{0,1\} \qquad \textit{for all } i, j, \textit{ and } k \tag{1.16}$$

The objective in (1.10) is the minimisation of the total transportation cost. Constraints (1.11) ensure that every customer is visited only once. Constraints (1.12) ensure that each node is entered and exited once. Constraints (1.13) ensure that vehicle capacities are not exceeded. Constraints (1.14) ensure that every vehicle is used only once. Constraints (1.15) ensure sub-tour elimination (similar to the TSP) while y_i and y_j are arbitrary real numbers used to force each route to pass through the depot. Constraints (1.16) represent the binary conditions on x_{ij}.

1.3.2.3 Applicability of the VRP to our work

It should be noted that the formulation and definition provided above is for the *basic* VRP. Some extra constraints are added to the basic VRP because of their importance in practice. Some of these practical issues are inventory management and time constraints. However, the basic VRP has received much more attention in the literature than other practical VRPs (Laporte and Osman, 1995). Many different exact and approximate solutions are available for the VRP. Because of its difficulty, heuristic algorithms are mainly used to solve the VRP problem. Most of these heuristics are two-phase methods (Bramel and Simchi-Levi, 1995). Two-phase methods are generally divided into two types. The first type is *cluster first-route second* where customers are clustered into groups, and then an optimal route is obtained for each cluster. The second type is *route first-cluster second* where a TSP is solved for all customers, and then customers are clustered into groups.

We are not interested in the solution procedure of the VRP. For our work, the interest in the VRP arises because of the following:

- Although we use a different formulation, the aggregate model of the ship-scheduling problem presented in our work can be considered as a routing problem with some extra practical constraints (journey times and capacities).
- Although we use ships instead of land-based vehicles, a practical VRP called the *Inventory Routing Problem* (IRP), has a great resemblance to the ship-scheduling problem of our work in terms of the objectives and the formulations.

The IRP is a VRP where customers have different daily consumption rates of the delivered product. The consumption rate of customers results in a daily demand requirement for each customer. In addition to minimising the transportation costs, the objective of maintaining certain levels of inventory at customers' sites is enforced. A literature review of this particular problem is presented in Chapter 2. A fairly recent review on dynamic routing and inventory problems is provided by Ukovich *et al.* (1998).

1.3.3 Ship scheduling problems

The VMI system is explored within a ship-scheduling context instead of land-based vehicles. As opposed to VRP, our VMI system includes the combined routing and scheduling of ships (i.e. the time element is considered). In other words, products have to be received at specific time periods (or time windows), while in VRP, a feasible solution is just a route that visits all nodes regardless of time. The time element adds a certain complexity to the VMI problem which reduces the feasible region. Therefore, we explore optimisation-based techniques to solve highly-constrained VMI problems while heuristics (which tend to work better in less-constrained problems) are mainly used to solve routing problems.

The shipped products in many ship-scheduling problems are usually quantified in cargoes, where each cargo has a specific pickup/delivery dates and specific loading/unloading ports. In our work, we do not deal with cargoes since the vendor is responsible for deciding how much of the products are to be delivered and when. In addition, we consider bulk shipping of continuous products (*e.g.* oil products) as opposed to discrete products (*e.g.* parts). Moreover, delivery is scheduled for time periods rather than time windows because the customer has a continuous (but time-varying) level of demand. A comprehensive literature review of ship scheduling problems is provided in Chapter 2. The mathematical formulations of the problem are explained in Chapter 3.

1.4 Project Aims

The PhD project aims at designing an overall dynamic VMI system. In this system, an integrated production planning, distribution, and inventory management methodology is developed for the vendor. To achieve that aim, the project plan is to use different optimisation-based techniques to simulate and optimise the replenishment policy for VMI systems. Then, the performance of VMI systems is tested with different industrial case studies.

A potential future objective is to design a Computerised Decision Support Tool (CDST) for *Industrial* VMI systems. The user is asked to input all the necessary data into the CDST. In return, the CDST outputs all required information using the proper type of display (tables, charts, graphics ...etc.).

Input to CDST:

- Production sites, maximum production rates, cost of production per product, sites' storage capacities and required ship loading times at those sites
- Customer locations, location storage capacities, and required ship unloading times at those locations
- Ships available for transport, ship storage capacities, and their initial position
- Valid journeys between sites and locations, journey durations, and costs for each ship
- Starting and ending dates for the period of optimisation
- Daily forecasted demand of each product at each customer location for the period of optimisation
- Initial inventories at locations and ships in addition to the desired final inventories
- Market selling price for each product

Output of CDST:

- Optimal average daily production rate of each product at each production site during the scheduling period
- Optimal schedule of journeys for each ship
- Optimal loading/unloading rate of material during the scheduling period
- Inventories at all ships, sites, and locations at any day during the scheduling period

1.5 Problem Statement

In our work, the entire VMI system in terms of production, transportation, and inventory is optimised.

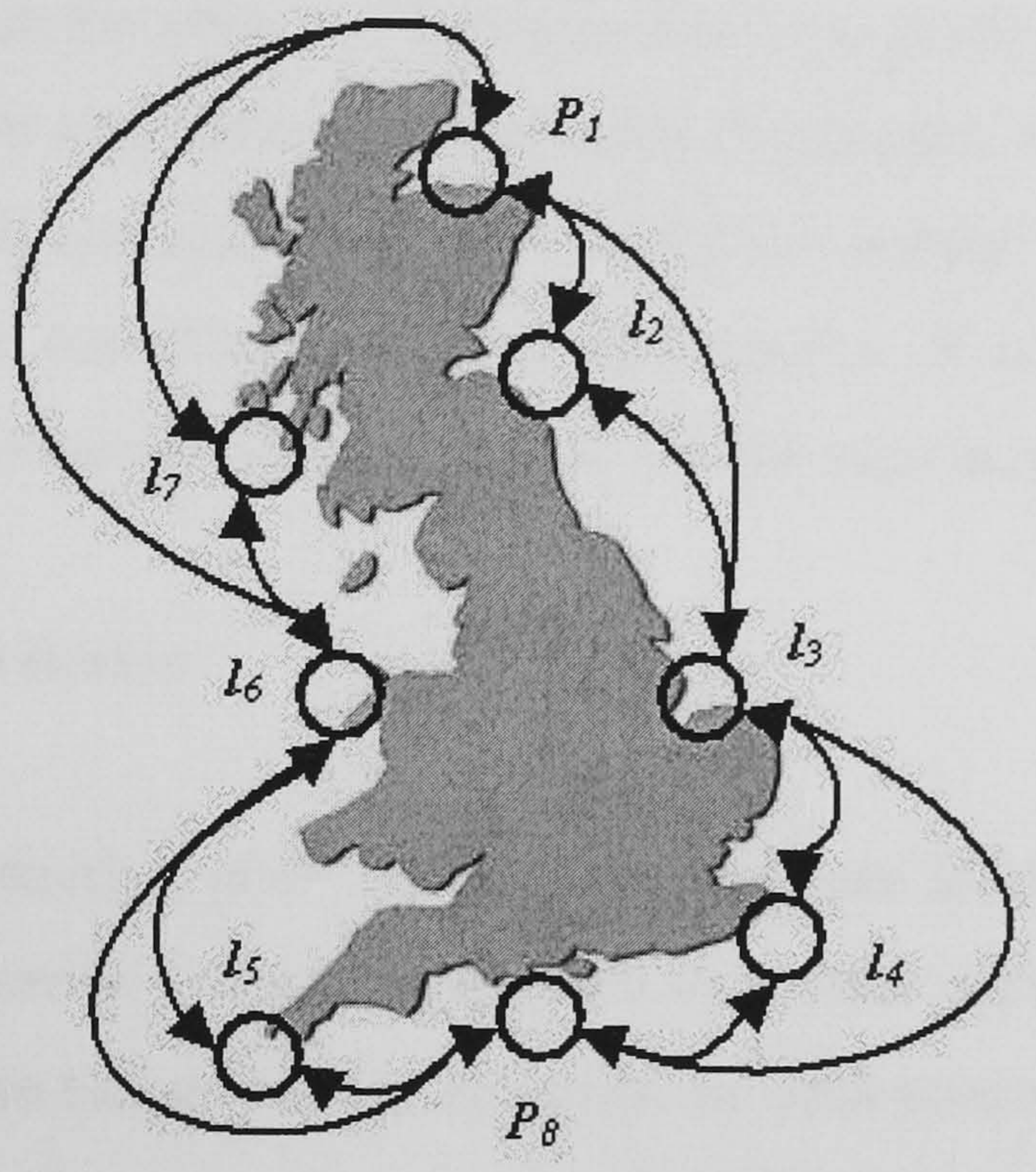

Figure 1.4: Production plants and customer locations

The vendor is responsible for transporting a number of products from the production plants(s) to geographically-dispersed customer locations. Forecasted daily/weekly demands at customer locations have to be satisfied without exceeding the customer storage capacities. No orders are received and the vendor must decide when to replenish each customer and by how much. The production rates of each product at each production plant must not be exceeded. A fleet of ships owned by the vendor is used to transport the product(s). All ships in the fleet have a certain fixed capacity that must not be exceeded. In addition, the problem incorporates any factors that affect the entire schedule such as plant shut downs and ship maintenance. Figure 1.4 shows an example of how the production plants (p_1 *and* p_8) and the customer locations ($l_2 \ldots l_7$) are distributed on the UK map. Arrows on the map show the allowed ship journeys. The detailed VMI problem specifications and required data are described below.

Objective

In order to avoid infeasibilities, demand requirements are treated as soft constraints. Consequently, the objective becomes maximising the total profit rather than minimising the total cost. A penalty for unfulfilled demand requirements for each customer is included in the objective function. The total profit is calculated by taking the difference between total sales and total costs. Total sales are represented by total demand satisfied at all customer locations. Total costs consist of transportation costs and penalty costs for unfulfilled demand requirements. If total sales data are not available, the objective becomes minimisation of total costs instead.

Plants/locations information

Storage tanks at production plants as well as customer locations are dedicated to specific products to avoid mixing. In addition to storage capacities, the production rates of each product in tonnes per day are given for each production plant. Moreover, the initial inventories of each product at every plant/location are given.

Ships information

All ships in the fleet have flexible compartments. As a result, any combination of the products can be carried onboard as long as the total ship storage capacity is not exceeded. In addition, every ship's initial position is known at the beginning of the optimisation. Initially, a ship can either be in a particular port or on its way to one. Moreover, the initial inventories of each product on each ship are given.

Products information

The production costs of all products are given for each plant. In addition, the selling prices of each product are given for every customer.

Journey information

Each ship is assumed to be suitable for certain journeys because of some limitations such as depth restrictions. At the start of the horizon, all journeys must originate from their initial positions. For every ship, journey durations as well as journey costs are provided. Shipping costs are based on sailing time and ship type (size, bunker, ...etc.). Product transhipment (transferring product between ships) is not allowed because of the high costs of transferring bulk commodities. There are no limitations on the customer locations that a ship can visit although these limitations are common in industrial operations. In addition, a ship can unload at different customer locations. As for production plant(s), no unloading is allowed. Restricting the feasible journeys does not necessarily prohibit a ship from visiting any port. Instead, the route will be different while the overall solution is approximately the same. This fact can always be true because the journey costs in the VMI problem satisfy the triangle inequality $c_{ij} + c_{jk} \geq c_{ik}$ (Laporte, 1992). In general, if a ship does not load/unload at a location, then the conclusion is that ship did not enter the port and it passed through that port on its way to another port.

Loading/unloading information

For each plant/location, the pumping rate is fixed. Consequently, the total quantity loaded/unloaded per discrete period can be calculated.

Supply/demand information

Daily or weekly demand forecasts of each product are satisfied at fixed time periods rather than time windows. For every customer location, once product is available and needed, we assume that demand is depleted instantaneously. Penalties for not meeting a customer's demand are incurred in the objective function. In addition, to VMI customers, some external (non-VMI) demand can be included in the problem. We assume that external demand (if any) is taken directly from the vendor's storage tank.

Extra specifications

The VMI problem can easily adapt to any additional problem-specific restrictions. Such restrictions include ship maintenance and plant shut-downs.

1.6 Problem Complexity

An optimisation problem is said to belong to class *P* if it is tractable on a deterministic machine. *NP* is the set of all problems that are tractable on non-deterministic machines (Garey and Johnson, 1979). A problem is *NP-complete* if a quick algorithm to solve this problem can be used to solve all other NP-complete problems quickly. *NP-hard* problems are at least as hard as NP-complete problems.

All nontrivial routing and scheduling problems are NP-hard in general (Bramel and Simchi-Levi, 1995). On the same notion, it is very essential to emphasize the level of difficulty of the industrial VMI systems we deal with in our work. In reactive (order-driven) systems, the vendor responds to orders received from the customer. Although these traditional systems may be unbeneficial to vendors, the resulting problems are relatively easier mathematically because many of the key decisions have been taken (albeit sub-optimally) by the customer. On the other hand, VMI systems are beneficial to the vendor and mathematically difficult to solve. In VMI systems, time becomes much more important because customers are using material all the time and they need regular replenishments. The vendor looks at the entire demand forecasts of the all customers rather than some fixed orders. Consequently, the optimisation of these systems involves the entire space of decisions rather than a subset of it. In addition, linking routing and scheduling decisions of multiple products with production plans and inventory management makes the VMI problem even more difficult. Only consideration of physical real-life constraints can reduce the level of difficulty for VMI problems. Such constraints include port restrictions. In our work, we first try to mathematically model these systems using a dynamic process-scheduling system. Then, we propose efficient solution approaches to solve the resulting model. We are aiming to derive the maximum benefit from the VMI strategy. Therefore, we raise the challenge by exploring mainly optimisation-based algorithms. Nevertheless, general heuristic algorithms are used to show the benefits of optimisation.

1.7 Thesis Outline

Chapter 2 provides a relevant literature review. First, a summary of VMI-related work is presented. Then, the inventory routing problem is discussed as a special VMI problem. Finally, the ship-scheduling problem is extensively reviewed since it represents the mode of transportation in the VMI system considered here.

Chapter 3 describes the mathematical formulation of VMI systems. A detailed MILP model is explained first. Then, a direct solution approach using the detailed model is introduced.

Chapter 4 describes the aggregate-time mathematical formulation of VMI systems. An aggregate MILP model is explained first. Then, an iterative solution approach using the aggregate and detailed models is introduced.

Chapter 5 discusses the rolling horizon approach tailored for our VMI problem. First, background about the rolling horizon is presented. Secondly, mathematical modelling of the VMI system in a rolling horizon framework is explained. Finally, a hybrid rolling horizon solution approach is introduced.

Chapter 6 uses an illustrative VMI example problem to evaluate the solution approaches described in the previous chapters. First, two general industrial heuristics are introduced. Secondly, solutions obtained by optimisation-based approaches are compared to those obtained using industrial heuristics.

Chapter 7 presents a realistic evaluation of the VMI system using industrial case studies from different companies. For each case study, the problem is first solved using both heuristics and optimisation-based approaches. Then, potential improvements to the current problems are discussed and final recommendations are presented.

Chapter 8 presents some concluding remarks on the VMI problem and the mathematical approaches used. Finally, future work areas for the VMI problem are provided.

Chapter 2

Literature Review

The PhD project includes three main parts. The first part is production planning, the second part is distribution planning, and the third part is inventory management. The VMI system in our research is a dynamic integration of all these parts. There exists some work that combines production and distribution (Cohen and Lee, 1988). Moreover, there exists some work that combines distribution and inventory planning (Dror and Ball, 1987). To our knowledge, there is no previous work in the literature which views VMI systems as a dynamic integration of production planning, distribution, and inventory management.

VMI was popularised in the 1980's by Wal-Mart and Procter & Gamble in the US (Waller *et al.*, 1999). Therefore, VMI is considered a new area since all VMI work started to be published in the late 1990's. However, the main aim of our project is the optimisation of VMI systems (production planning, distribution, and inventory management). In Section 2.1, a review of existing VMI-related work is presented and compared to our project. Since the mode of transportation for our VMI distribution system is shipping, a comprehensive literature review is presented for ship scheduling in Section 2.3. As for the VRP, we introduce a brief literature review on the special case of the IRP because of its direct relevance to VMI objectives.

Although production planning is a major part of VMI systems, it is not the driving force compared to demand, nor should it be. In our work, production plans are represented in terms of the daily production rates at all plants. Those rates are subject to many constraints such as plant storage and production capabilities. For simplicity, we assume that raw materials are always available at the vendor's plants. No literature review is presented for production planning problems. Such problems can be explored further in terms of raw material availabilities and material requirements planning (MRP). However, such considerations will extend the VMI supply chain to include raw material suppliers. This aspect of production planning is out of the scope of this thesis and included as a future area of research.

2.1 Vendor Managed Inventory

Recall that in Chapter 1, we categorised inventory management methodologies into two main categories (Section 1.1.1). The first category is the traditional (reactive) strategy and the second category is the proactive (VMI) strategy. In traditional strategies, pure inventory problems are solved without comprehensive consideration of other aspects. The objective of pure inventory problems is to minimise the holding cost over a long time horizon. In such problems, distribution is incorporated only in terms of a fixed cost for deliveries. Many inventory models are formulated to find the optimal frequency and order quantity. One such important model is the *economic order quantity* model (Christopher, 1998). Other models try to minimise the stocks by increasing the order frequency and decreasing the order quantity. Such systems are widely used in production assembly lines where materials are ordered "Just In Time" to reduce inventory buffers (Beasley, 2003). Extensive work on pure inventory models exists in the literature. All these pure inventory models are classified under the traditional reactive (order-based) system. No pure inventory models are included in our literature review.

On the other hand, the VMI strategy spans inventory management to include production and distribution. Therefore, VMI is a strategy for managing supply chains. In Section 2.1.1 a literature review on existing published VMI work is presented. In Section 2.1.2, the existing work in the literature is compared to the VMI problem in this project.

2.1.1 Selected literature on VMI

Waller *et al.* (1999) evaluate the effect of VMI systems on a certain supply chain of one vendor and multiple distribution centres some of which are non-VMI retailers. The performance measure in their model is the level of inventory at distribution centres. A simulation is conducted on scenarios with different order frequencies and different demand variability. The daily demand is normally distributed and the fixed mean is relative to the centre's capacity. Daily supply is assumed to cover all demand requirements. However, if the amount of orders exceeds the supply on any day, a first-come first-served basis is used. No dynamic transportation is modelled in the simulation. Randomly generated transportation times are only considered from the vendor to the centres. The model is implemented using the SIMAN programming language. The model is tested on real-life data from Hewlett Packard. Then, the simulation studies the effects on non-VMI customers. Results show that the better co-ordination of manufacturing in the VMI system also benefits other non-VMI retailers. Finally, different production capacities are tested for the same demand requirements. Results show a reduction in production backlogging for the vendor due to elimination of large, infrequent orders. As for VMI distribution centres, great reductions in inventory levels result for all demand variability cases.

Achabal *et al.* (2000) design a VMI decision support system (DSS). The DSS includes a specific demand-forecasting model and inventory-management model for each retailer. No distribution model is included in the DSS. The forecasting model uses sales data and promotion policies to provide demand forecasts. Two types of forecasts are generated. First, weekly demand forecasts are generated and passed to the inventory model to determine the proper stock levels. Second, aggregate long-term forecasts are generated and passed to the manufacturer for production planning. Forecasting integrates normal sales, seasonal variations, and retailer-based effects. Demand is normally distributed, and target service levels equal the probability that demand is less that the target inventory. The target levels (minimum inventory and customer service levels) are agreed upon by the vendor and each retailer. A major apparel manufacturer implements the DSS with more than 30 retailers. Although inventory turnover slightly declined, the service level significantly increased after using the VMI DSS.

Cetinkaya and Lee (2000) solve a VMI problem analytically. One vendor supplies a product to a group of retailers with random demand. The source of supply for the vendor is the manufacturer. In other words, the vendor places orders to receive the product from the manufacturer where VMI system is applied only between the vendor and the retailers. Retailers are clustered in different geographical regions such that their demand can be consolidated in one larger load rather than smaller multiple loads. The total costs incurred include procurement, inventory holding, waiting costs for received but not delivered orders, and transportation costs from the vendor to the region of retailers with no lead times. As opposed to other inventory models, transportation costs consist of a fixed dispatching cost and a unit-dependent cost rather than just fixed delivery costs. The vendor accumulates orders from retailers until the beginning of the consolidation cycle rather than directly delivering the product. The manufacturer replenishes the vendor with product instantly based on the accumulated orders by retailers and the on-hand inventory level at the vendor's site. The renewal reward theorem is used to evaluate the expected average cost for the case of Poisson demand. The resulting expected average cost is a function of the target inventory level at vendor's site and the consolidation cycle time. The minimum average total cost is then explicitly used to find the optimal order quantity and consolidation cycle. A heuristic approach is used to solve for those optimal values by approximating the exact formula. Calculations show that the order quantity reduces to the economic order quantity formula if no shipment consolidation is considered.

Axsater (2001) solves the same problem presented by Cetinkaya and Lee (2000). However, the author presents an exact rather than an approximate optimisation algorithm of the problem for a range of discrete values for the consolidation cycle time. The exact algorithm outperforms the approximate algorithm of Cetinkaya and Lee (2000) in terms of cost reductions.

Kaipia *et al.* (2002) analyse the time benefits of VMI systems using the entire product scale rather than the stock keeping unit (SKU) scale. The analysis objective is to eliminate the ordering process and giving that time to the supplier to better organise replenishments. Two extreme cases are considered; one is order-based while the other is order-free (VMI). Demand data of multiple products are collected for both cases. In order-based systems, demand is represented by purchase orders. In VMI systems, demand is approximated from consumption rates or sales data. Further parameters of demand deviations and response times are then calculated for both cases. The potential time saving of VMI systems is calculated by taking the difference between the response times. The ratio between the mean absolute deviations of demand is calculated to quantify the *bullwhip effect* (demand amplification in the supply chain). The analysis is applied to three real life case studies. For every case, the order-based case is compared to the VMI system. Then, the power is shifted from the vendor to the customer by switching from VMI systems to Just-In-Time (JIT) system. Time benefits are gained by switching from order-based systems to VMI systems. In addition, the VMI system outperforms the JIT system because of the flexibility that enables the vendor to better plan operations in VMI systems.

Dong and Xu (2002) evaluate the short-term and long-term effect of VMI on a supply chain of one manufacturer (supplier) and one buyer (customer). The Economic Order Quantity (EOQ) is utilised to describe the buyer's inventory system and the supplier's profit by obtaining the optimal purchase price and purchase quantity. Distribution is included in the order fixed set-up costs. Note here that the supplier generates orders for the buyer. The authors analytically prove that VMI reduces the total inventory costs of the system (buyer and supplier). Furthermore, the authors show that only the buyer's short-term profits increase after VMI implementation. In the long-term, the buyer's profit is still consistent with the short-term result. However, the supplier benefits in the long-term as opposed to short-term results. The long-term profit of the supplier is directly related to the long-term profit of the buyer because of the increased volume of sales.

Disney and Towill (2002) link the Automatic Pipeline, Inventory and Order Based Production Control System (APIOBPCS) with VMI. APIOBPCS has been used before for production scheduling at particular manufacturer. However, the authors innovate this system and couple it with a VMI supply chain. The coupled system is used with a two-echelon supply chain of one manufacturer (vendor) and one distributor (customer). The distributor uses sales data to provide future demand forecasting and the re-order point level and passes this information to the manufacturer. Then, the manufacturer uses these data to set the production target levels. Note that the system contains only production and inventory as the two main components of the system. Distribution is accounted for in this system only in terms of delivery lead-time. A closed-loop block diagram representation of the VMI-APIOBPCS with a sampling time of 1 is presented. First, difference equations for production, inventory and forecasting are generated. Distribution is represented in the loop as a discrete time delay to account for delivery lead-time. All difference functions are then transformed to the z-domain to yield the system's discrete transfer function. The input is this system is the consumption rate, while the output is the inventory level. The performance of VMI-APIOBPCS system is analysed in the time domain by taking the inverse z-transform of the transfer function. The initial value theorem is used to find the initial conditions of the system. Furthermore, the stability of the VMI-APIOBPCS is studied and the stability conditions of the system are finally determined. The main parameters that affect the stability of the system are related to production rate. Note here that the authors only aim at finding the stable region of the system. In other words, no optimisation of VMI-APIOBPCS is presented.

In addition to their work in testing the stability of the VMI-APIOBPCS, Disney and Towill (2002) present a related paper that optimises the VMI-APIOBPCS described above by minimising the total cost. The total cost is the summation of the distributor's inventory cost metric, the system inventory cost metric, and the production adaptation cost metric. The objective function is represented by the reciprocal of the Euclidean distance from zero in three-dimensional space. Hence, the problem becomes a maximisation rather than a minimisation process. Since adapting different production rates can affect the inventory level due to bullwhip effect, different ratios of production costs to inventory costs are tested. A search algorithm over a user-defined range of decision parameters is then used to find the best set of parameters. One of these parameters is the ratio of safety stock to consumption rate. This ratio is used to evaluate the trade-off between transportation costs (implied by batch sizes) and inventory holding costs. Furthermore, results show that the bullwhip effect is reduced if the delivery lead-time decreases.

2.1.2 A comparison between our work and VMI in the literature

It is obvious that VMI attracted much attention lately because of its potential benefits both for vendors and customers. Researchers are always keen to evaluate these benefits. However, the approach and context of such evaluations is usually different. We believe that present research only covers some aspects of the VMI problem. A classification of differences between the VMI system in our work and literature is provided below.

Evaluation approach

Different aspects of the VMI problem are tackled to study its performance. Some work replicates pure inventory models objectives by obtaining the order frequency and quantity. Benefits are assessed analytically in terms of the optimal order quantities and shipment cycle (Cetinkaya and Lee, 2000) or purchase prices (Dong and Xu, 2002). Other work views the VMI system within production control systems (Disney and Towill, 2002) and the stability of the system is the major objective. Available simulations of the VMI system measure the performance by the level of inventory (Waller *et al.*, 1999). The decision support system designed by Achabal *et al.*, (2000) is heavily dependent on forecasting. Others (Kaipia *et al.*, 2002) explore only the benefits of order time elimination. In most published VMI work, very little attention is given to the transportation component of the supply chain (Disney *et al.*, 2003) or on the time-phased replenishment of customers. In our work, a full optimisation methodology is suggested to evaluate the benefits of VMI systems. We dynamically optimise the VMI system by integrating all of its components (production planning, distribution and inventory).

Distribution optimisation

Most of the previous work deals with only one customer. In such cases, delivery costs are fixed and included in the set-up costs. For models with multiple customers, distribution factors are considered in terms of lead-times. In our work, distribution is included where the vendor has to plan the routing of shipments to minimise the transportation costs.

Type of products

Most of the research deals with discrete items rather than continuous types of materials. In our work, we focus on process industries that deal with continuous products such as industrial gases and oil products.

Order generation

Some work replicates traditional systems within VMI context. In such work, the vendor generates orders (Dong and Xu, 2002). Although some benefits are realised by the total reduction of inventory in the system, potential benefits are not totally achieved. The time benefit of order elimination is properly explained by Achabal *et al.* (2000). In our work, we eliminate the order process and simultaneous production, distribution, and inventory management is passed to the vendor. In the next section, we present a selected literature review on the IRP.

2.2 Inventory Routing Problems

The IRP represents one of the most important practical problems of the VRP. Although much less published research can be found about the IRP compared to the VRP, an abundance of formulations and applications of the IRP can be found in the literature.

The IRP is defined in Chapter 1 as a special VRP. The main resemblance of the IRP to our problem is the objective. Daily customer demand should be satisfied while ensuring that the customer's storage capacities are not exceeded. It is obvious that the IRP is a much harder problem compared to the VRP. Therefore, in attempting to solve these problems, some researchers add many restrictions. Some restrictions can produce solutions within a certain gap from the exact optimal solution. The decision of the trade-off between solution quality and reasonable running time is left to the expert. For instance, Gallego and Simchi-Levi (1990) prove that a direct policy (where each customer is served by only one vehicle) produces solutions within 6% margin of optimality.

Note that only deterministic IRPs are considered in the literature review for our work. Some of the articles reviewed have some stochastic analyses. Stochastic demands as well as uncertainty in general are discussed as a future work area in Chapter 8. In Section 2.2.1, we present a brief review of the deterministic IRP in the literature. In Section 2.2.2, a comparison between our work and the literature in the field of the IRP in presented.

2.2.1 Selected literature on IRP

To our knowledge, Ukovich *et al.* (1998) present the most recent review on the IRP in which the authors refer to the IRP as the dynamic routing-and-inventory problem (DRAI). The authors provide a classification of all DRAI in the literature. According to their classification, solution methods are divided into two main approaches. The first approach is *frequency domain* where the optimal solution involves finding the best replenishment frequencies. The second approach is *time domain* where the optimal solution involves finding replenishment schedules. The authors provide a literature review of DRAI problems with both approaches. Note, that the second approach (time domain) is the more relevant approach to our work. However, in our literature review we view articles with both approaches.

Dror and Michael Ball (1987) present a heating-oil distribution problem in which they reformulate a long-term IRP to a reduced version that deals with short-term decisions. Customers are divided into subsets, and the problem is solved for each period (i.e. a time domain approach is used). The trade-off between early and later replenishments is considered when demand is stochastic. Early replenishment has a probability of another replenishment sooner and hence more costs. On the other hand, later replenishment has a probability of a stock out. The authors define the expected cost function for a single customer case. Then, the single-customer case is extended to cover multiple customers (i.e. IRP) by incorporation of transportation costs. The TSP is solved as a sub-problem within the IRP to find the least cost route over a set of customers. The model is solved using a three-phase heuristic. The first step involves assigning a customer to each time period (as an LP problem). The second step involves solving a VRP for each day. The final step involves improving the solution by exchanging some of the combinations generated in the previous two steps. At the end, the proposed algorithm is compared to the manual one using a real-life problem of heating-oil distribution. A 50% increase in performance (units/hour) is obtained with this algorithm compared to the manual system.

Bramel and Simchi-Levi (1995) present a location-based heuristic (LBH) that formulates a routing problem as a location problem. The heuristic divides the region of all customers to small squares. Each square has a set of customers with a total demand less than the capacity of the vehicle. A TSP is solved for each square to minimise the total transportation costs for that particular vehicle and set of customers. The capacitated concentrator location problem (CCLP) is used to cluster customers into minimum-tour groups. The LBH is used to solve a VRP and IRP. While requiring similar CPU times, the LBH provides better solutions for the VRP than many other heuristics. On the other hand, applying the LBH to the IRP involves extra assumptions/constraints. In addition to transportation costs, the total costs involve a fixed cost for each placed order and holding costs. There is an infinite storage capacity at customers' sites. Demands are treated as hard constraints. The main assumption is that customers are divided into subsets. If a customer in any subset is served, then all customers in that subset are also served. The authors justify this assumption from retailers' and drivers' perspectives. From retailer's point of view, orders are received at constant regular times (i.e. frequency approach is used). As for drivers, it is reasonable to have drivers specialise in certain routes rather than learning all possible routes. The optimal *cycle time* (time between deliveries) is found using the *economic order quantity* formula. A solution of the IRP is presented with different fixed-cost values with uniformly distributed demands (1,10). The solution shows that in one-third of the cases, the vehicle load is not exhausted.

Campbell *et al.* (1997) view the IRP within a VMI context. The authors use a two-phase algorithm based on cluster first-route second to solve the IRP. Customers are first clustered in two cost-compatible groups. Then, the authors develop an integer programming approach to solve the routing problem of the IRP. In their modelling, the authors make certain assumptions. First, each cluster is served by only one vehicle. Second, when a customer in a cluster is visited, then all customers in that cluster are visited. This assumption emphasizes the importance of consumption rate compatibility of customers in each cluster. However, the authors do not explain the clustering method. They refer to the second cost estimate as the method of partitioning customers into clusters.

Viswanathan and Mathur (1997) present a multi-product IRP where the objective is to obtain the optimal long-term replenishment policy for products (i.e. a frequency approach is used). The authors implement a stationary nested joint replenishment policy (SNJRP) heuristic. A *stationary* policy involves finding equal replenishment intervals. A *nested* policy involves having multiple replenishment intervals for some products relative to other products with less replenishment intervals. SNJRP starts by calculating the replenishment intervals of customers independently using the *power-of-two* rule (two times the base period). The customer with the lowest interval is added to the cluster and assigned a certain weight relative to the route. Customers with greater intervals are then added to the cluster (nested) until the capacity of the vehicle is exhausted. Then new clusters are constructed with new customers in the same manner. The optimisation process involves re-evaluation of some customers' intervals based on other customers' capacities and time intervals. SNJRP can produce a savings of 10% on the total costs compared to the independent replenishment policies.

Bard *et al.* (1998) present the first work on IRP that introduce satellite facilities (IRPSF). *Satellite facilities* are geographically distributed depots that can serve as re-supply points for vehicles. Similar to all VRPs, vehicles have to go back to the central warehouse by the end of the day. In other words, no overnight stays are allowed at satellite facilities. The authors use a similar decomposition technique proposed by Dror and Ball (1987) to reduce the long-term problem into a short-term problem (i.e. a time domain approach is used). The general decomposition approach starts with identifying customers whose replenishment dates are within the planning horizon. A trade-off between assigning customers to their replenishment dates and cost of otherwise is evaluated. For the customer-day combination, a VRP problem with satellite facilities is solved to minimise the total transportation cost. Finally, the solution is improved by exchanging customers between certain routes. The step of solving the VRPSF is compared with three different existing VRP heuristic algorithms. The decomposition approach is tested with different mean and variance values for consumption rates with a fixed unloading time. Results are compared for each VRP heuristic used. Note here that the initial step of assigning customers to days is the inventory part of the IRP. The rest is just solving the VRP for each day. The latter step involves most of the work in the method.

Rusdiansyah and Tsao (2005) combine IRP and VMI in a vending machine supply chain over a planning horizon of six days. The objective function is to minimise the holding costs and average travelling costs. Every retailer (customer) is allowed a certain number of visits in the planning horizon within specific time windows. A heuristic algorithm containing two phases (initialisation and improvement) is used to solve the problem. In the initialisation phase, a visit frequency is determined for each retailer. In the improvement phase, visit-day combination interchanges are performed using a Tabu Search (TS) algorithm. The mathematical model was coded using the C-language. The model was then tested using real data examples. Examples show that combining inventory and travel costs in one model can result in huge savings.

2.2.2 Comparison between our work and IRP in the literature

It is obvious that there exists some resemblance between the ship-scheduling problem considered in our VMI system and the so-called IRP. The main common area is inventory management. In both problems, customers' locations have certain capacities and demand must be satisfied while not exceeding those capacities. However, since the mode of transportation is different (ships in our case), the problem formulation and solution procedures are different. Differences between ship routing and land-based vehicle routing are discussed by Ronen (1983). We try to classify the differences between our problem and the IRP as follows:

Central supply

In the IRP, there is always a central supplier/warehouse. Sometimes the IRP is referred to as *one-warehouse multi-retailer* distribution system. The existence of satellite facilities in some problems (Bard *et al.*, 1998) does not undermine the role of the central warehouse where all vehicles have to start from and return to. In our problem, there is no central supplier and ships can end up in any port.

Number of customers/fleet size

IRPs deal with a large number of customers (usually hundreds). The fleet size is also greater in case of the IRP. In our problem, and similar to all shipping problems, we have fewer customers and a much smaller fleet size.

Costs involved

In the IRP, costs include transportation costs, fixed cost per delivery, and inventory holding costs. In our model, we only consider transportation costs with no fixed delivery costs. Because we deal with industrial VMI systems, material availability is the main priority. Hence, we do not include any holding costs.

Time horizon

The optimisation process in the IRP involves long-term decisions such as annual plans (sometimes over an infinite time horizon). Consequently, optimality is measured according to the average of customer satisfaction over the time horizon. In the ship scheduling problem, we deal with short-to-intermediate-term scheduling. Hence, the service level is evaluated per customer.

Demand information

In the IRP customers have certain consumption rates. This consumption rate can be translated into demand in our case. However, in the IRP, consumption rates are usually treated as hard constraints. In our problem, we deal with demand as soft constraints but demands can vary significantly between time periods. Moreover, demand requirements in our problem are not necessarily fixed for each customer as opposed to consumption rates in most IRPs.

Discharge quantities

IRPs deal with small loads compared to the capacity of the vehicle. Usually, it is specified that a set of customers be replenished only by one vehicle. This vehicle is assigned to an optimal route that visits all these customers. In our problem we deal with bulk shipments, where load quantities are of the same order of the ship capacity (most of the time full loads are delivered to customers). Moreover, any customer in our problem can be visited by more than one ship.

Solution approaches

Most of the solution methods are two-phase approaches (usually cluster first-route second). With a large number of customers, these approaches seem reasonable and more efficient. The optimisation then involves how customers are clustered in an efficient way so their demand consumption is compatible. Hence, when a vehicle visits the cluster, it usually delivers to all customers in that cluster. The second phase of the optimisation is a TSP for that particular vehicle to replenish that group of customers. In our VMI system, we formulate the ship-scheduling problem as an RTN. Then, the problem is solved using MILP optimisation-based algorithms. In the next section, we present a selected literature review on the Ship Routing and Scheduling Problem (SRS).

2.3 Ship Routing and Scheduling

The world's population is increasing while resources remain scarce. Consequently, the world economy is moving toward international trade where shipping is the most important transportation method. In addition, ships cost millions of dollars and their operations (fuel consumption and servicing/maintenance) cost thousands of dollars. Therefore, optimising ship routing and scheduling can result in very significant savings. Moreover, important capital investment decisions (i.e. ship procurements and sizing) can be based on the routing-and-scheduling aspect of shipping.

The SRS problem has been an interesting area of research for the last century. Although the idea seems simple and straightforward, solving SRS problems is mathematically complex in general. Hence, the quality of the mathematical formulation can be of great importance to solving these problems. Amazingly, in spite of the potential savings and the challenging nature, the SRS problem has received relatively little attention compared to other vehicle routing problems. Because the mode of transportation in our VMI system is shipping, we present an extensive review of the SRS problem in Section 2.3.1. In Section 2.3.2, a comparison between our work and the literature in the field of SRS in presented.

2.3.1 Selected literature on SRS

(Ronen (1983) in his comprehensive review of SRS problems explains the reasons for the low attention the SRS problem received compared to the VRP. The main reasons are the structure of SRS problems and the underlying uncertainty due to weather or mechanical difficulties associated with ships. He also describes the range of ship scheduling as short-term (days to weeks), medium-term (weeks to months), and long-term (months to years). In this review, Ronen categorises the modes of operations for ships into three main categories: liner (similar to bus routes), tramp (similar to taxi-cabs) and industrial (similar to a private fleet of trucks). Our problem (like most of industry operations) falls under the third category since the vendor (supplier) owns the fleet of ships and utilises this fleet to meet the requirements of its customers at a minimum cost. Chronologically, the review of Ronen (1983) is the earliest contribution to our literature survey of SRS. Similar to the classification of the VRP provided in Section 1.3.2, Ronen introduces a classification for SRS problems (mode, time windows, ports, products, ships, demands, ... etc.). Finally, the presentation of the differences between the VRP and the SRS problem is of great importance to our work since our problem captures similar features of both problems. Briefly, the main differences mentioned by Ronen (1983) are:

1) Great variations between ships in capacity, speed, and operating costs.
2) The mode of operation (liner, tramp, or industrial) dictates the scheduling environment.
3) The higher uncertainty associated with ships due to longer journey times.
4) The limitations of drivers' circumstances that usually forces vehicles to return to their point of origin.
5) The flexibility of continuous operation of ships (days or weeks) compared to vehicles.
6) Ship schedules are more flexible compared to those of vehicles. Therefore, ship schedules can be changed easily.

Ronen (1986) presents an SRS problem using heuristics and non-linear optimisation. He uses three algorithms to solve a ship-scheduling problem. Those algorithms are: The exact algorithm, single step cost minimization algorithm, and a random generation algorithm. Each ship is assigned to only one route that includes certain ports due to port entry restrictions. The objective is to minimise the total shipping cost that includes route costs, unloading costs, and port charges. The problem is structured as mixed-integer non-linear problem. Non-linearity arises from including costs per unit shipped on ships. The results of the three solution algorithms are compared to the general rule of sending the largest ships to farthest ports (based on the existing system). The exact algorithm used the context of the transportation problem to find the optimal solution. The random generator algorithm chooses a ship randomly using a uniform distribution. Randomly, ports are chosen until the load of the ship is exhausted or there are no remaining ports. Finally, the algorithm calculates the optimum route for each ship based on its designated ports. On the other hand, the single step cost minimization heuristic algorithm uses the tonnage-mileage relationship as a basis. Since the exact algorithm consumes a lot of computational time depending on the size of the problem, the random generator and the heuristic proved to produce near optimal (and sometimes optimal) results depending on the size of the problem. In all cases the "industry rule" produced the worst solution.

Ronen *et al.* (1987) present a crude oil transportation problem. The cost of scheduling includes opportunity costs, fuel consumption, port entry charges, and charter ship costs. The problem is structured as a set-partitioning problem (SPP) in the sense that only feasible independent schedules are generated. However, the authors suggested and used an updated SPP that he called the Elastic Set Partitioning Problem (ESPP). In (ESPP) some constraints can be violated at some penalty in the objective function in order to avoid infeasibilities. The procedure included the generation of all feasible schedules for all ships, the cost calculations associated with each schedule, and finally the optimisation process to choose the optimal set of schedules. The schedule generator takes into account all mentioned cost/port restrictions and specifies the cruising speed based on that schedule. Then the cost calculator ties all these cost/restrictions and comes up with the total cost for that schedule on that ship. Finally, the optimisation procedure solves the problem to optimality by satisfying all cargo and ship constraints (i.e. each ship is assigned to one schedule only, and each cargo can be used only once). As a result, the optimisation problem is an MILP problem. The model is tested using operational data with different test conditions (cargoes, time horizon, ports ...etc.). The problem is solved in relatively short CPU times.

To our knowledge, Miller (1987) presents the first work that considers integration of distribution and inventory. Miller uses a node-arc formulation to generate all feasible schedules for ships. This formulation contains four different types of nodes for every period over the time horizon in order to capture every single detail of the simulation. Arcs represent journeys between locations in the network. In addition to cost information, the generated schedules include information about inventory at end terminals. The solution procedure consists of four steps. The first step uses a heuristic approach to generate feasible schedules without applying any constraints. The second step uses generated routes to apply the constraints and produce any violations. The third step is a report generator for the scheduler to make any corrections/improvements to schedules in the fourth and last step. In addition to user interventions, the system makes its own improvements towards optimisation. Tested with a real life problem for shipping multi-chemical products, the four-step user-interactive procedure proved to be a good tool for obtaining near-optimal results.

Fisher and Rosenwein (1989) present one of the earliest research papers on SRS with time windows. The objective is to minimise the total transportation cost (charter costs and fleet operating costs). In addition to the usual constraints (capacity, loading/unloading, port restrictions), a cargo sequence constraint is added because some products cannot be lifted using the same compartments. For each ship, a maximum cruising speed is specified. Using cruising speeds and distances between ports, arrival times at those ports can be calculated. The solution method consists of two phases. First, all feasible schedules are generated where a possibility of all feasible schedules as well as a possibility of eliminating some schedules is given. Then a set packing formulation is used to solve the problem (i.e. choosing the optimal set of schedules) using Lagrangian LP-relaxation. Finally, interactive graphical reports are generated to allow clear vision and some manual corrections to schedules. The model was tested with a real life problem for the US navy. The system shows a significant improvement over manual scheduling.

Kao and Lee (1996) tackle a ship-scheduling problem in order to minimise the idle time for ships inside loading/unloading ports (demurrage). Ships are treated as jobs while docks are treated as loading/unloading machines. Here, the port does the scheduling in order to minimise the demurrage costs that the port has to pay to ship owners. Therefore, the available information to the port authorities is the ship arrival times and due dates (the date in which loading/unloading has to finish). Since machines are set-up in a parallel manner, different machines can be used at the same time. Constraints include starting a job after its ready time and before the end of the time horizon In addition to other feasibility constraints such as assuring that the number of jobs does not exceed the number of machines. To avoid infeasibilities, jobs are allowed to start after their due date, however, this will incur cost penalties. The problem is structured as only a binary integer programming model (i.e. only decision variables are to be obtained). A real-world example is applied to test the efficiency of the model using a uniform inter-arrival time distribution for ships and medium-term (90 days) time horizon for fifteen ships. Applying this model to the problem decreased the idleness of the docks from 30% to 10%.

In addition to a brief review of ship-scheduling systems including early and recent literature, Kim and Lee (1997) present a decision support system (DSS) for ship scheduling. The authors state the major difference between liner and industrial operations mentioned by Ronen (1983). In liner operations, carriers have fixed costs for transportations. On the other hand, industrial carriers base their operations on demand and supply information (which is the case with our problem in our work). The authors use a set-packing formulation to solve a ship-scheduling problem to design a DSS. However, the generation of schedules is unique. The new procedure generates a feasible graph of all possible schedules. Then, all feasible schedules are generated from that graph. These feasible schedules are then inserted into the LINDO optimiser to find the optimal solution. The DSS is tested using a real-life scenario. The DSS produced good results in short CPU times with some visual output for optimal schedules.

To our knowledge, Nygreen and Christiansen (1999) present the first recent article that combines ship routing with inventory constraints. They present an SRS with inventory considerations of the visited harbours. The problem is structured as a "path flow formulation" and uses "decomposition approach" to solve the problem. The LP-relaxation uses "column generation" where columns represent routes of ships and sequences for harbours. The technique used for solving the ship-scheduling problem is a combination of the multi-vehicle pickup and delivery problem with time windows and a multi-inventory problem. The procedure uses a Dantzig-Wolfe decomposition algorithm to solve sub-problems for each ship as well as each harbour. The sub-problems are integrated to solve the master problem. Each harbour has stock limits and rates of production/consumption of the shipped product. The initial state of ships can be anywhere including harbours and the sea. The cost includes fuel usage as well as harbour entry charges. The problem resembles the multi-pickup and multi-delivery vehicle routing with time windows in the transportation part of the problem. It is important to note that time windows are associated with harbours as a whole. Consequently, visiting different nodes within a harbour should be within the time window of that harbour. Sub-problems are solved independently (both using the shortest path problem to find roust/sequences) and more constraints are introduced in the master problem. Since the problem includes column generation, the authors have written their branch and bound algorithm and used the SCICONIC subroutine library for solving the master matrix with some modification for column generation. It is worth mentioning that solving a problem with this size is very time-consuming, therefore, the authors have implemented some techniques to reduce the number of possibilities by reducing time windows and reducing the number of routes. The procedure is tested with a problem of shipping ammonia in Northern Europe with three ships and eleven harbours for a time horizon of a month. Results for two different time windows are presented. Nygreen and Christiansen (1999) present a similar work to their contribution in 1998 with emphasis on the path flow formulation and solving the LP-relaxations of the model using the simplex method.

To our knowledge, Fagerholt (2000) presents the first SRS article to treat time windows as soft constraints. As a result, each customer has an inner and an outer time window. A special cost function is enforced for any inner window violations. The cost for not meeting that time constraint is added to the objective function. Therefore, the objective becomes minimizing the total cost of transportation (fuel usage, port charges, and charter ships) and "inconvenience" costs (incurred by not meeting the customer time). In addition, each customer is associated with a cost factor that is added specially for that particular customer. Hence, the importance of each customer in the model is quantified in the overall model. In common with other research, the solution method involves two phases. First, all feasible schedules are generated. Second, the master problem is solved using a set-portioning formulation. The schedule generation procedure uses heuristics to eliminate some schedules (based on waiting time and capacity utilisation). Then, the travelling salesman problem with time windows and capacity constraints is applied to generate the most promising schedules. Transportation costs are related to cruising speed, which is related to fuel consumption. On the other hand, inconvenience costs are related to violations of time windows. Therefore, the set-portioning optimisation process involves finding the starting time as well as the times of all stops in order to minimise the total cost. The master problem is run using General Algebraic Modelling System (GAMS) as the model builder and the CPLEX package as the solver. The schedule generator is written in Pascal. The model is tested with different real-life cases and different cost functions (linear, quadratic, ...etc.). Although they differ depending on the cost function used, results in terms of deviation from the hard time window case show that a substantial savings in transportation costs can be achieved if customers allow some flexibility in the their time windows.

Christiansen and Fagerholt (2000) present the first SRS problem with cargo flexibility for multi product shipments. Ship scheduling with multiple products has been described in the literature. However, Christiansen and Fagerholt (2000) include cargo allocation as part of the optimisation procedure. The problem is divided into two sub-problems; the first sub problem solves the scheduling problem in a similar way to multi pickup multi delivery routing with time windows. The second sub problem is solved as a multi allocation problem (mixing of products is not allowed). The solution procedure uses two phases like all previous articles. In the first phase, all feasible schedules are generated along with different allocations in all ships. This step is performed using the travelling salesman problem with allocation, time windows and precedence constraints (TSP-ATWPC). The second phase involves solving the problem as a set-partitioning problem (each ship sails on one schedule) to find the optimal schedule for each ship. The schedule generation is done in a systematic way. The generation algorithm was coded in Pascal while GAMS/CPLEX combination is used for optimisation. The model is tested with a problem of shipping fertilizers using different values of an allowance factor (which accounts for acceptable waiting time/capacity utilisation). With low values of the allowance factor (less number of schedules), the problem solved faster although it sometimes eliminated the optimal solution. On the other hand, for high values of the allowance factor (all feasible schedules), CPU times increased slightly while optimal solutions are obtained.

Christiansen and Fagerholt (2001) explain in detail the TSP-ATWPC used to generate the schedules in their previous article. The TSP is cited for different problems in the literature even with time windows and precedence, however this is the first article to consider allocation in the TSP problem. The TSP-ATWPC is a sub-problem to generate a limited set of most promising feasible schedules for the main scheduling problem. In addition to usual TSP, allocation is added to force feasible cargo set-ups inside compartments. Time windows are added to arrange visitation within specified time windows, and precedence constraints force loading ports to be visited prior to unloading ports. Since TSP-TWPC is discussed previously in the literature, the allocation problem is emphasised by Christiansen and Fagerholt (2001). Due to different allocation possibilities, the number of schedules will increase. However, some restrictions are imposed to reduce the number of possibilities of cargoes such as having a utilisation factor for each ship. This utilisation factor has to exceed some certain amount in order for multiple cargo-combinations to be generated for that ship. A dynamic programming model is used to solve the TSP-ATWPC in which the objective of minimizing the total waiting time is used as an approximation of reducing the total costs in the master problem. The TSP-ATWPC is written in Pascal. The procedure produced different results with respect to different examples by varying the utilisation factor between the two extremes (0 and 1). The solution of the set-partitioning problem is shown by Christiansen and Fagerholt (2000).

After evaluating the use of soft time windows, Fagerholt (2001) solves a similar SRS problem. In addition to soft time windows, the problem is more complex than that of Christiansen and Fagerholt (2000) due to multiple compartments (i.e. ships can lift more than one cargo). Moreover, an extra shortest-path formulation algorithm is used to further optimise sequences generated by TSP-ATWPC. The solution procedure consists of two phases. The first phase is the generation of schedules. The schedule generation phase involves three steps. In the first step, feasible schedules that satisfy some heuristics (time/capacity utilisations) are generated. Next, a TSP-TWPC is solved for every sequence to choose most promising ones. Finally, for every fixed sequence of visits, a shortest path formulation is used to optimise the schedule for that sequence in terms of the total costs where specifying the cruising speed affects the fuel usage and arrival times. Consequently, operating costs and inconvenience costs are calculated. The second phase of the solution procedure is solving the problem using set-portioning formulation to find the best schedule for each ship. In the second phase, schedules are available and optimisation process assures starting of service and the times of each stop. The schedule generator (including heuristics, TSP-CTWP, and schedule optimisation) is written in Pascal. The MILP is written in GAMS using the CPLEX solver. The model is tested with a real-life bulk-shipping problem in North Europe using different cases (in terms of nodes, time horizon, and time discretisation, cost functions...etc.). Significant savings in transportation costs are achieved when using soft time windows.

Christiansen and Fagerholt (2002) represent the first work to include penalty costs due to arriving outside operating hours in a port. The authors present a problem of ship scheduling where ports have restrictions (in addition to fees and depths) on operating hours. Therefore, a ship entering a port outside the operating hours will result in the ship being idle until the start of service at that port. Operating hours can extend the time window of loading/unloading in those ports. Assuming that operating hours are relatively short compared to the time window, the time window can be viewed as multiple time windows. The authors define a "risky" arrival (such as arriving near a weekend) for which a penalty cost in incurred. The penalty cost as well as the length of the risky interval associated with risky arrivals is based on the planner's experience. The ships have fixed speeds because fixed speeds will lead to dependent evaluation of risky arrivals. In other words, ships will not have the luxury of changing speed to avoid risky arrivals. Instead, avoiding risky arrivals will be the job of the scheduler. The solution procedure involves the generation of all feasible schedules in a systematic way. Heuristics are applied to eliminate some schedules due to time/capacity utilisation. Then a travelling salesman problem with capacity, penalised multiple time windows, and precedence constraints (TSP-CPmTWPC) is solved for each schedule to produce the optimal schedule for each set of nodes. The TSP-CPmTWPC is solved using dynamic programming. No further sequence optimisation is used for schedules generated by TSP-CPmTWPC. All costs associated with these schedules are calculated including penalising for idle times. Finally, a set-partitioning formulation is used to solve the problem to optimality. The schedule generator (including heuristics, TSP-CPmTWPC) is written in Pascal. The MILP is written in GAMS using the CPLEX solver. The model is tested with a real-life bulk-shipping problem in north Europe with different cases (in terms of nodes, cargoes, time horizon, different costs and so on). The results showed an increasing avoidance of risky arrivals as the penalty cost increases. However, due to introduction of penalty costs (for arriving at risky times), the transportation costs increased compared to the available manual costs because of the different (more expensive) routing incurred.

Jetlund and Karimi (2004) solve a short-term ship-scheduling problem of multiple chemical products. To simplify the problem the authors make several assumptions. One assumption assures that a ship cannot visit a port more than once during the planned schedule. The problem is constructed as an MILP deterministic model with possible reruns to include any new information. A new heuristic approach that solves the main problem for every flexible-compartment ship is proposed. The heuristic assigns cargoes to ships permanently during the planning horizon. Assigning cargoes to ships is performed by calculating the maximum marginal profit of any cargo on any ship. Hence, the problem is solved ship by ship to maximise the ship's profit. Runs were performed using the CPLEX solver on GAMS. The heuristic decision support system (DSS) was tested using real data from an international chemical shipping company. The DSS showed a 33 % increase in the company's profit compared to their actual schedule.

Cheng and Duran (2004) develop a DSS to improve the logistics of a world-wide crude oil transportation supply chain. The supply chain consists of one central supply location and four main demand regions. The DSS combines discrete even simulation and stochastic optimal control. The discrete event simulation model is used to represent the tanker queuing system. The stochastic optimal control model is used evaluate the impact of decision-making on the state of the system (i.e. inventories and ship locations). After building the integrated DSS, multiple runs (scenarios) were performed to evaluate the system's performance. Although, results show that the DSS show expected behaviour, the authors are suggesting that more runs should be performed for further validation.

Karimi *et al.* (2005) describe an interesting ship-scheduling problem of tank containers. Tank containers are used on ships, trucks, or trains as opposed to drums. Tank containers are fixed in size, environment-friendly, and cleanable modes that can be used with different transport methods besides shipping. An empty tank container is moved first from a container company to a chemical company's site. The container is loaded with a chemical product at the chemical company's site. Then, the product is sent to a particular destination to be unloaded. Finally, the containers are cleaned and repositioned to their origin or other sites that need them. The transportation network consists of depots, ports and sites. A site receives containers and loads them. A depot receives empty containers, cleans them, and sends them to sites or other depots. In general, a land route takes the containers from their origin to sea ports. Shipping is then used to transport the loaded containers to another sea port. Finally, a land route is used take the emptied containers back to their origin or to other sites. The objective minimises the total cost of container movements (land and sea based). Many assumptions are made to the initial problem such that ships have unlimited capacities (i.e. every ship can carry any number of containers). Hence, no distinction is made between land and sea based transportation methods. A two-phase approach is used to solve the initial container movement problem. First, an "event-based" model is used to generate possible movements based on the order list. Second, a linear programming formulation is used to choose the set of events that minimise the total cost. After testing the initial model with an illustrative example, a new model with more realistic extensions is presented. Such extensions include limited ship capacities and time windows as opposed to just in time (JIT) strategy. In both models, ship schedules are know in advance, the model only specifies which container should go on which ship.

(Christiansen *et al.*, 2004) presents one of the most recent review papers on the ship scheduling and routing problem. This review includes the latest papers on ship routing and scheduling. The review is divided into two main parts; strategic planning and tactical decisions. Strategic planning involves the design and sizing of the fleet. Tactical decisions deals with the optimisation of the flee operations. The different modes of operations (liner, tramp, and industrial) are discussed under the tactical decisions. Finally, the review explores some new trends and potential research areas in the shipping industry.

2.3.2 Comparison between our work and SRS in the literature

Our VMI problem is structured using the RTN formulation (explained in Chapter 3). The resulting MILP problem is then optimised to minimise the total transportation cost. Ships and customers capacity limitations are treated as hard constraints. However, customers' daily/weekly demand is treated as soft constraints. A penalty is imposed for each unit of customer's unmet demand requirements. As a result, the minimisation problem is transferred into a maximisation of profit with a penalty of unmet demands. The main differences between our VMI ship-scheduling problem and other literature work are classified as follows:

VMI context

The basic and conceptual difference between our work and previous literature work is the context of VMI systems. In VMI, customers demand requirements are dealt with in proactive manner. As opposed to all previous articles, demands of customers are known and all SRS models are "order-driven". In most SRS problems in the literature, cargoes have certain pickup date and port and delivery date and port. In our work, demands represent forecasts (done by customers, vendor, or both) rather than orders. The vendor is responsible for meeting these demand forecasts by regular replenishments of customer sites. The VMI context increases the difficulty of the problem as explained in Section 1.6.

Time constraints

For all articles that deal with time constraints, the delivery/pickup has a time windows (demand has to be satisfied within a certain time interval) regardless of this time window being hard (Fisher and Rosenwein, 1989) or being soft (Fagerholt, 2000). In our work, demand is dealt with on a daily/weekly basis with the assumption that the customer uses the material continually. Therefore, our problem is more dynamic compared to other problems. Dynamic demand increases the complexity of the problem significantly.

Solution approaches

Most solution algorithms are heuristic and meta-heuristic algorithms rather than optimisation-based approaches. Available optimisation based approaches are based on the arc-flow or the path flow-models. Arc-flow models find the best feasible path by evaluating the arcs to be included in the schedule. Path-flow models divide the solution into two phases. The first phase is the generation of feasible schedules regardless of any heuristic optimisation to those schedules. The second phase is usually solving a set-portioning problem where every ship is allowed only one schedule regardless of the method or LP-relaxation used. We use a dynamic RTN formulation to solve the VMI problem as a whole (see Chapter 3 for details of the mathematical formulation). By supplying allowed journeys between locations (along with times and costs) as input data, the procedure is more robust to include the return of ships to reload from production plant(s). The RTN formulation of the whole problem can be viewed as a simulation-based optimisation of the VMI system. As a result, every fine detail of the problem (inventory levels and ship locations) can be tracked as time passes. Another advantage of the RTN formulation is the flexibility of adding more VMI aspects to the system. Production planning instead of supply information is easily incorporated with this formulation. In addition, we propose novel optimisation-based approaches such as *time aggregation* and *rolling horizon algorithms*. Those approaches produce near-optimal solutions for large problems in reasonable CPU times.

2.4 Discussions and Motivations

Based on the previous introduction and literature review, we can see how the proposed PhD research project is unique in many aspects. First, the VMI system is considered to be a new emerging strategy and potential benefits of this system have not been totally realised.

Secondly, the context in which we evaluate the VMI system is different than what is available in the literature. We try to explore the total potential benefits of VMI implementation in a more practical, dynamic, and simulation-based methodology. In the literature, usually only partial components of the VMI system are considered in detail. One category of literature work couples production and inventory management while distribution is evaluated only as lead-times (Waller *et al.*, 1999). Another category of literature work couples distribution and inventory management in vehicle routing context (Dror and Ball, 1987) and in ship scheduling context (Miller, 1987). In this category, extensive work is done on the distribution system using heuristic algorithms. In our project, a full optimisation of the VMI system is proposed. All three major components of VMI systems (production planning, distribution, and inventory management) are simultaneously considered in a dynamic model.

Thirdly, the VMI system considered in this project is optimised using ships as the mode of transportation. A formulation similar to the resource-task network, RTN (usually used in the field of process scheduling) is utilised to model the MILP problem of the VMI system. The RTN formulation (Pantelides, 1994) explores the specific dynamic details of the system in terms of products, journeys, the locations of ships, and inventory levels. The added complexity of the RTN formulations in such supply chain scale systems is dealt with using different mathematical algorithms such as time aggregation. The tight aggregate model is designed and used as an upper bound for the detailed problem. Finally, a novel more robust rolling horizon approach is designed and utilised to solve large-scale problems efficiently.

In conclusion, a major contribution is expected from the VMI system in this project. A novel assessment of the VMI system as an emerging supply management strategy is anticipated. Novelty arises in the dynamic consideration of such systems, the overall simulation-based approach proposed to optimise these systems, and in the mathematical algorithms used to apply those approaches. Ultimately, real-life industrial case studies using the proposed VMI system are considered. We expect that the scale of these case studies and the performance of the VMI strategy compared to other strategies can enhance the contribution of this research. The developed system is incorporated into a prototype CDST, which will facilitate the operations and popularise the VMI concept.

Chapter 3

Model Formulation and Direct Solution Approach

3.1 Discrete Time Formulation

To model the VMI system, we follow a similar approach to the RTN formulation proposed by Pantelides (1994). Based on the standard RTN framework, a unified approach is taken to represent all *resources* in the scheduling problem. Consequently, there is no difference between products and equipments since both are treated as resources. On the other hand, a *task* is any operation that uses one or more of the available resources.

Consider an example VMI problem that consists of one plant (**P**), one customer (**C**), one ship (**S**), and one product (**X**). Theoretically speaking, an RTN sequence of tasks might be as follows:

1) A "charge" task to load **X** from **P** to **S**.
2) A "journey" task to transfer **X** from **P** to **C**.
3) A "discharge" task to unload **X** from **S** to **C**.

In spite of the generic resemblance, our VMI system diverges from the standard RTN formulation. Because we actually have a journey task, ship resources clearly follow the standard RTN formulation. On the other hand, product resources do not follow the standard RTN formulation because we do not have an explicit charge/discharge task , but rather we just use continuous variables to denote charging and discharging (see Section 3.2). In other words, resources (ships and products) are not treated uniformly in our VMI model. Following the modified RTN logic, variables in our model are divided into two types; state variables and event variables. Figure 3.1 below shows how these variables are arranged along the time horizon.

State variables

At $t=0$, a fixed value is specified for all state variables. Then, constraints for state variables are generated at for each time period ($t=1,\ldots,H+1$). In our VMI problem, the main state variables are:

1) Inventory in a port
2) Inventory in a ship
3) Existence of a ship in a location

Event variables

An event may take place at each time period ($t = 1,\ldots,H$). The end of the time horizon is dealt with as a state (i.e. no event takes place at $H+1$). In our problem, the main event variables are:

1) Start of a journey
2) Loading from port to ship
3) Unloading from ship to port

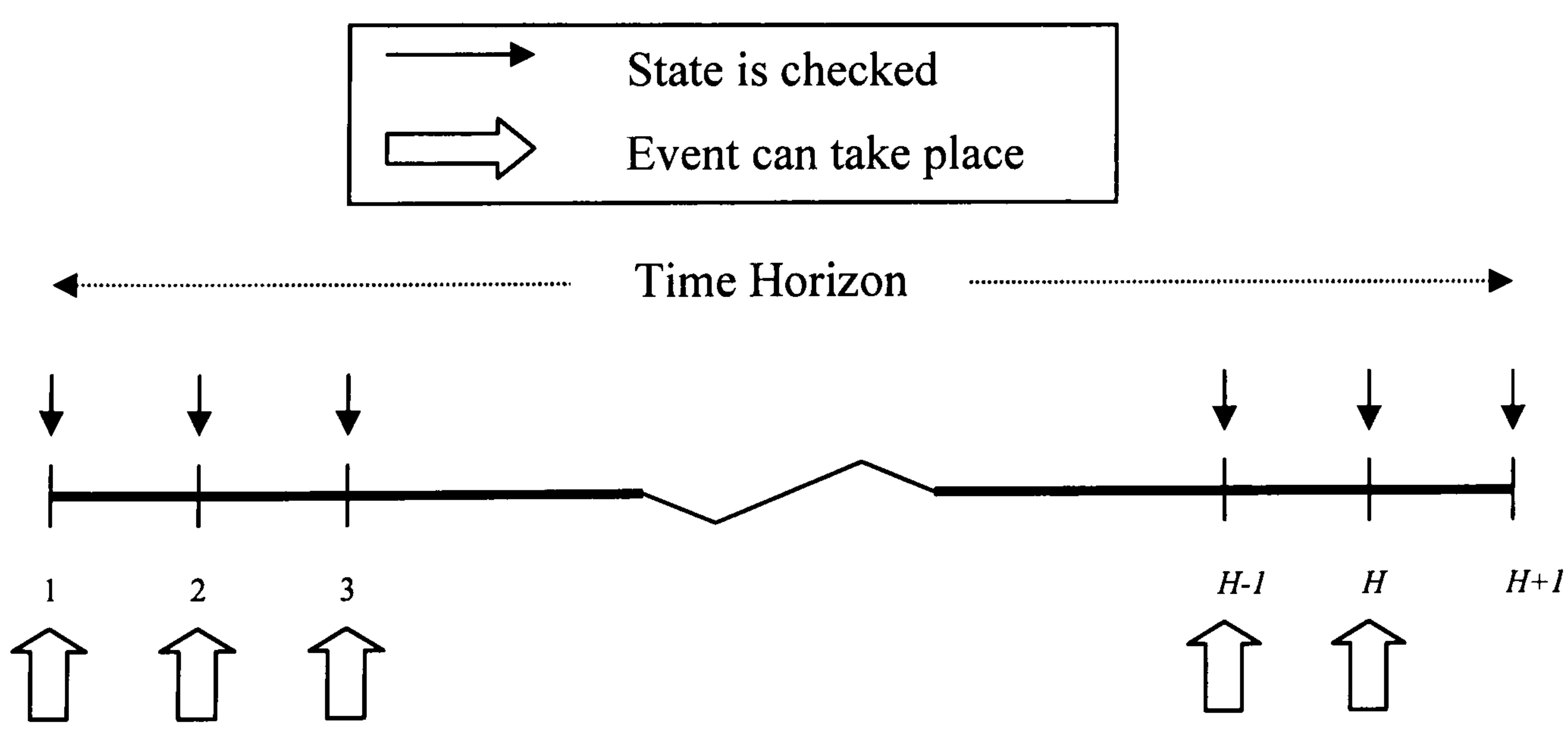

Figure 3.1: Time discretisation diagram

3.2 Detailed Mathematical Formulation

The mathematical formulation of the VMI problem is presented in detail as follows:

Indices

i	Products
l	Ports which include production sites and customer locations
s	Vendor-owned ships available for transport
j	Possible journeys between ports
t	Time periods $(1,2,\ldots,H+1)$ where 0 is the initial state

Sets

p	Production sites only ($p \subseteq l$)

Parameters

Δt	Discretisation interval
H	Length of the time horizon under consideration
FD_{ilt}	Demand forecasts of product i at time t for VMI customer location l
XD_{ipt}	External (non-VMI) customers' demand of product i at time t taken from production site p
α	Penalty cost multiplier in the objective function
C_{sj}	Cost of journey j on ship s
τ_{sj}	Duration of journey j on ship s
T_l	Total quantity that can be loaded/unloaded at port l based on the pumping rate (tonnes per Δt)
SL_j	Starting location for journey j
FL_j	Finishing location for journey j
ψ_{il}	Storage capacity of product i at port l
ψ_s	Total storage capacity of ship s
σ_{ip}	Maximum production rate of product i at production site p
ξ_p^0	Start of shut-down period at production site p
ξ_p^{end}	End of shut-down period at production site p
σ_{ip}^{shut}	Maximum production rate of product i at production site p during shut-down period

μ_{ip} Cost of producing a single unit of product i at production site p ($\mu_{ip} > 0$)

ρ_i^{vmi} Selling price of product i to VMI customers

ρ_i^{ext} Selling price of product i to external (non-VMI) customers

$LI\_t_{il}^0$ Initial inventory of product i at port l

$SI\_t_{is}^0$ Initial inventory of product i at ship s

$LI\_t_{il}^{end}$ Final desired inventory of product i at port l

$SI\_t_{is}^{end}$ Final desired inventory of product i at ship s

β_l Maximum number of ships allowed in port l

$R\_t_{slt}^0$ 1 if the initial port for ship s at t is port l ; 0 otherwise

δ_{sl} 1 if ship s is allowed to visit port l ; 0 otherwise

$\eta_{s,l,t}$ Length of the maintenance period for ship s at port l starting at time t ; 0 otherwise

Continuous variables

Qd_{islt} Amount of product i discharged from ship s to customer location l at time t

Qc_{islt} Amount of product i charged from production site l to ship s at time t

LI_{ilt} Inventory of product i at port l at time t

SI_{ist} Inventory of product i on ship s at time t

D_{ilt} Actual satisfied demand of product i at time t for customer location l

Δ_{ilt} A penalty cost incurred for not meeting demand of product i at time t for customer location l

Binary variables

X_{sjt} 1 if s starts journey j at time t; 0 otherwise

R_{slt} 1 if s is at port l at time t; 0 otherwise

The objective is to maximise the total profit. Profit is defined as the total revenues minus the total costs. Total revenues are represented by total demand satisfied at all customer locations multiplied by the price of each product. Total costs consist of cost of production, transportation, and penalties. Note that a negative sign is added to the production costs term since supply is always negative (i.e. $D_{i,l:l\in p,t} \leq 0$). The value α is based on the problem in hand. As α goes to infinity demand becomes a hard constraint. Equations 3.1-3.23 represent the mathematical formulation of the detailed model.

Maximise

revenues

$$\sum_i \sum_t (\sum_{l:l\notin p} \rho_i^{vmi} \times D_{ilt} + \sum_{l:l\in p} \rho_i^{ext} \times XD_{ilt})$$

production

$$-\sum_i \sum_{l:l\in p} \mu_{il} \sum_t -(D_{ilt} + XD_{ilt})$$

transportation (3.1)

$$-\sum_s \sum_j C_{sj} \sum_t X_{sjt}$$

penalty

$$-\alpha \sum_i \sum_{l:l\notin p} \sum_t \Delta_{ilt}$$

Subject to the following Constraints:

Penalty cost

The penalty cost incurred for not meeting demand requirements is calculated by the taking the difference between forecasted and actual demands at all times. Note that Constraints 3.2 are generated for customer locations only.

$$\Delta_{ilt} \geq FD_{ilt} - D_{ilt} \qquad \forall\ i,\ l=customer\ location,\ t=1,2,\ldots,H \quad (3.2)$$

Initial port constraints

Initially, a ship can either be at a certain port or on the way to that port. If a ship s exists in a port l then $t^* = t^0$ and $R\_t^0_{s,l,t} = 1$ for that ship-port combination. However, if a ship s is on its way to a certain port l, it will be at that port at a time $t^* > t^0$. Where, t^* represents the activation time of every ship s. Therefore, Constraints 3.3 ensure that all ships are not activated before their respective t^*. Constraints 3.4 assign the initial port of each ship at its activation time t^*.

$$R_{s,l,t<t^*} = 0 \qquad \forall\ s,\ l,\ t^* = t : (R\_t^0_{s,l,t} = 1) \quad (3.3)$$

$$R_{s,l,t=t^*} = 1 \qquad \forall\ s,\ l,\ t^* = t : (R\_t^0_{s,l,t} = 1) \quad (3.4)$$

Ship allocation constraints

For each ship-port combination at the beginning of each time, that ship exists in that port only if it was there during the previous time period, or it has just arrived from another port. On the other hand, a particular ship will not exist in a particular port at the beginning of a time period if it was not there from the previous time period, or it has just left the location at the beginning of that time period. Note that R is defined as a binary variable in the formulation. However, during solution, R is actually treated as a continuous variable with lower and upper bounds ($0 \le R \le 1$) to reduce the running time. To avoid infeasibilities, Constraints 3.5 are generated starting with (t^*+1) where t^* is the activation time for every ship.

$$R_{s,l,t} = R_{s,l,t-1} + \sum_{j:l=FL_j} X_{s,j,t'-\tau_{js}} - \sum_{j:l=SL_j} X_{s,j,t} \qquad \forall\ s,\ l,\ t=t^*+1,\ t^*+2, \ldots, H+1 \quad (3.5)$$

where $t' > \tau_{js}$

Loading time constraints

At any time period, the loading/unloading quantity is less than the maximum loading rate per time period. Constraints 3.5-3.7 assure that a ship s remains in port l for a certain loading/unloading time. Loading/unloading time depends on the pumping rate at l. If a ship is just passing through l, the LHS in Constraints 3.6-3.7 will be equal to zero. Hence, the loading/unloading time (RHS) will be equal to zero.

$$\sum_i Qc_{islt} \le T_l \times R_{slt} \qquad \forall\ s,\ l,\ t=1,2,\ldots,H \quad (3.6)$$

$$\sum_i Qd_{islt} \le T_l \times R_{slt} \qquad \forall\ s,\ l,\ t=1,2,\ldots,H \quad (3.7)$$

Port mass balances

The forecasted inventory in each port at the beginning of each time interval is equal to the inventory at the beginning of the previous interval in addition to any material transferred to that port at the beginning of that time interval minus any material transferred from that port at the beginning of that time interval minus the forecasted demand taken from that port at the beginning of that time interval.

$$LI_{i,l,t} = LI_{i,l,t-1} + \sum_{s}(Qd_{islt} - Qc_{islt}) - (D_{ilt} + XD_{ilt}) \qquad \forall\ i,l,\ t=1,2,\dots,H+1 \quad (3.8)$$

Ship mass balances

The inventory of each product in each ship at the beginning of each time interval is equal to the value from the last time period, plus any material charged to that ship at the beginning of the time interval minus any material discharged from that ship at the beginning of that time interval.

$$SI_{i,s,t} = SI_{i,s,t-1} + \sum_{l}(Qc_{islt} - Qd_{islt}) \qquad \forall\ i,l,\ t=1,2,\dots,H+1 \quad (3.9)$$

Initial inventory constraints

Initial actual inventories of each product at every port (Constraints 3.10) and every ship (Constraints 3.11) are specified by the user.

$$LI_{i,l,t=t0} = LI\_t^0_{il} \qquad \forall\ i,l \quad (3.10)$$

$$SI_{i,s,t=t0} = SI\_t^0_{is} \qquad \forall\ i,s \quad (3.11)$$

Final inventory constraints

Final desired inventories of each product at every port (Constraints 3.12) and every ship (Constraints 3.13) are specified by the user.

$$LI_{i,l,t=H+1} = LI\_t^{end}_{il} \qquad \forall\ i,l \quad (3.12)$$

$$SI_{i,s,t=H+1} = SI\_t^{end}_{is} \qquad \forall\ i,s \quad (3.13)$$

Capacity constraints

For any port or ship, inventories of all products at any time should not exceed the storage capacity. In Constraints 3.14, port storage capacities are given per product. In Constraints 3.15, only ship total storage capacities are considered because of the nature of flexible-compartment ships.

$$LI_{ilt} \le \psi_{il} \qquad \forall\ i,\ l,\ t=1,2,\ldots,H+1 \quad (3.14)$$

$$\sum_i SI_{ist} \le \psi_s \qquad \forall\ s,\ t=1,2,\ldots,H+1 \quad (3.15)$$

Demand constraints

The actual demand satisfied should not exceed demand forecasts for customer locations (Constraints 3.16). For production sites, demand is negative to represent supply of material. However, supply should not exceed the maximum production rate of each product (Constraints 3.17).

$$0 \le D_{ilt} \le FD_{ilt} \qquad \forall\ i,\ l=\text{customer location},\ t=1,2,\ldots,H+1 \quad (3.16)$$

$$-\sigma_{il} \le D_{ilt} \le 0 \qquad \forall\ i,\ l=\text{production site},\ t=1,2,\ldots,H+1 \quad (3.17)$$

Shut-down constraints

During a shut-down period, the daily production rate for each production site is reduced to a user-specified quantity $\sigma_{ip}^{shut} < \sigma_{ip}$.

$$-\sigma_{il}^{shut} \le D_{ilt} \le 0 \qquad \forall\ i,\ l=\text{production site},\ t\in[\xi_p^0, \xi_p^{end}] \quad (3.18)$$

Maximum number of ships in a port

The total number of ships existing at a port should not exceed that port's maximum capacity of ships at any time.

$$\sum_s R_{slt} \le \beta_l \qquad \forall\ l,\ t=1,2,\ldots,H+1 \quad (3.19)$$

Port restriction constraints

Some ships are not allowed to visit certain ports because of many reasons such as depth restrictions. A user specified parameter δ_{sl} is equal to zero if a ship s is not allowed to visit port l.

$$\sum_{j:l=FL_j} \sum_{t} X_{s,j,t} = 0 \qquad \forall\ s,l : \delta_{sl} = 0 \quad (3.20)$$

Ship maintenance constraints

If maintenance is required for ship s at port l at any time t, then this ship should remain at that port for the total maintenance period $\eta_{s,l,t}$.(Constraints 3.21) In addition, no loading or unloading is allowed during the maintenance period (Constraints 3.22-3.23).

$$R_{s,l,t} = 1 \qquad \forall\ s,\ l,\ t \in [\hat{t},\ \hat{t} + \eta_{s,l,t}] \quad (3.21)$$

$$Qc_{i,s,l,t} = 0 \qquad \forall\ i,\ s,\ l,\ t \in [\hat{t},\ \hat{t} + \eta_{s,l,t}] \quad (3.22)$$

$$Qc_{i,s,l,t} = 0 \qquad \forall\ i,\ s,\ l,\ t \in [\hat{t},\ \hat{t} + \eta_{s,l,t}] \quad (3.23)$$

where $t^{\wedge}=t : (\eta_{s,l,t} > 0)$

Equations 3.1-3.23 above conclude the detailed model which can be used to dynamically model ship movement, underlying loading, and production in a compact form.

3.3 Direct Solution Approach

The detailed model can be used to solve the entire VMI problem directly. Figure 3.2 shows a flow chart of the direct approach. The input data of the VMI system are inserted into the detailed model. An integer feasible solution is ultimately found for any reasonable problem size.

The solution provided by the direct approach is the global optimal solution of the VMI problem. This solution can be used as a bound to benchmark other solution approaches explained in the following chapters.

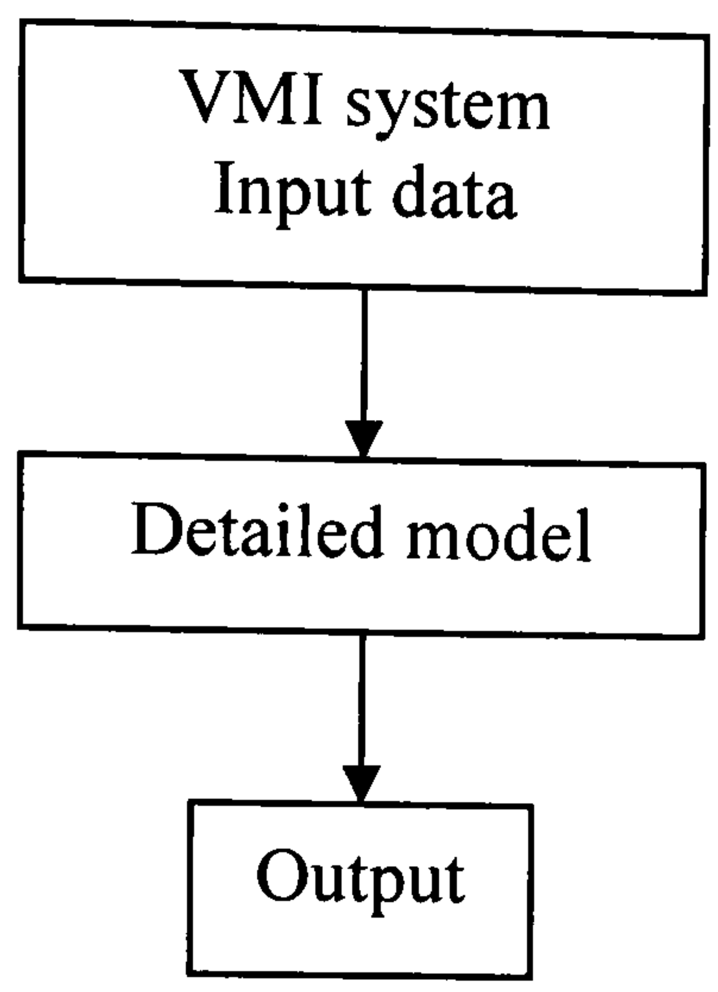

Figure 3.2: Direct approach flow chart

In Chapter 6, an illustrative example is used to evaluate the efficiency of the direct solution approach. Compared to heuristic algorithms, great benefits are attained when applying the direct approach. However, results show that as the complexity of the VMI problem increases, the direct solution consumes substantial CPU times (see Chapter 6). Hence, it is necessary to explore other optimisation-based approaches to solve complex VMI problems. Such approaches can capture the level of details embedded in the RTN formulation while reducing the solution time. Therefore, we introduce an aggregate RTN model of the VMI problem in Chapter 4. The aggregate model is used within different solution approaches to solve the VMI problem.

Chapter 4

Iterative Approach using Aggregate Time Formulation

As described in Section 3.1, the detailed VMI problem is modelled using discrete time representation where the time horizon is divided into equal intervals. The length of each interval (i.e. discretisation period) is equal to the duration of the shortest event. As opposed to continuous time representations, an event can take place only at the start of those intervals.

The routing and scheduling elements involved in VMI systems make these problems quite complex. As the time horizon, number of ships, or number of feasible journeys increases, the VMI problem size increases. As a result, obtaining detailed solutions for large-scale VMI problems is very difficult. Therefore, time aggregation can be used as an alternative or as part of a decomposition procedure to solve these problems. In this chapter, the aggregate model is used as a part of an iterative solution approach to the VMI problem. In chapter 5, the aggregate model is combined with the detailed model to form a novel rolling horizon approach. Near-optimal (sometimes optimal) solutions can be produced while the VMI level of detail is maintained.

In this chapter, we use the same basic time aggregation concept explained by Wilkinson (1996) using only one aggregated time period. In other words, the detailed time horizon is replaced by one aggregated time period of length *H*. Hence, all detailed variables in the VMI model are represented by new aggregate variables. These variables are expressed over the entire horizon *H* instead of time intervals. For example, we are not interested in the quantity delivered per interval (Qd_{islt}). Instead, we are interested in the total quantity delivered over the entire horizon ($\sum_t Qd_{islt}$). The same can be said about all other continuous variables (demand and inventories). Then, the problem reduces to a ship routing problem instead of a ship-scheduling problem because the time element disappears. This routing problem includes extra constraints such as journey times and capacity limitations.

One major difference between our aggregate formulation and process scheduling aggregate formulations is the representation of the binary variables. Recall from Chapter 3 that the main binary variable in the detailed model was the journey decisions (X_{sjt}). Since the time element in the aggregate representation disappears, a new integer (not binary) variable ($X_{sj} = \sum_t X_{sjt}$) is expected to represent journeys in the aggregated time period, where X_{sj} is the number of journeys j undertaken by ship s during the entire horizon H. However, we cannot use these variables in our aggregate model, because of a problem unique to routing; sub-tours.

To introduce sub-tours, we can construct a simple example of one ship (s) and four ports ($p_1 \ldots p_4$) as shown in figure 4.1. The total number of valid journeys between ports is 12 (i.e. 4^2-4). Aggregate constraints (similar to Equations 4.3 and 4.4) ensure that charging/discharging can only take place at visited ports. The set of journeys shown in figure 4.1 can be feasible in terms of charging/discharging since every port is visited once. However, the resulting set of integer journey variables ($X_{s1}= X_{s2}= X_{s3}= X_{s4}=1$) is infeasible as it leads to two sub-tours ($p_1 - p_2 - p_1$ and $p_3 - p_4 - p_3$).

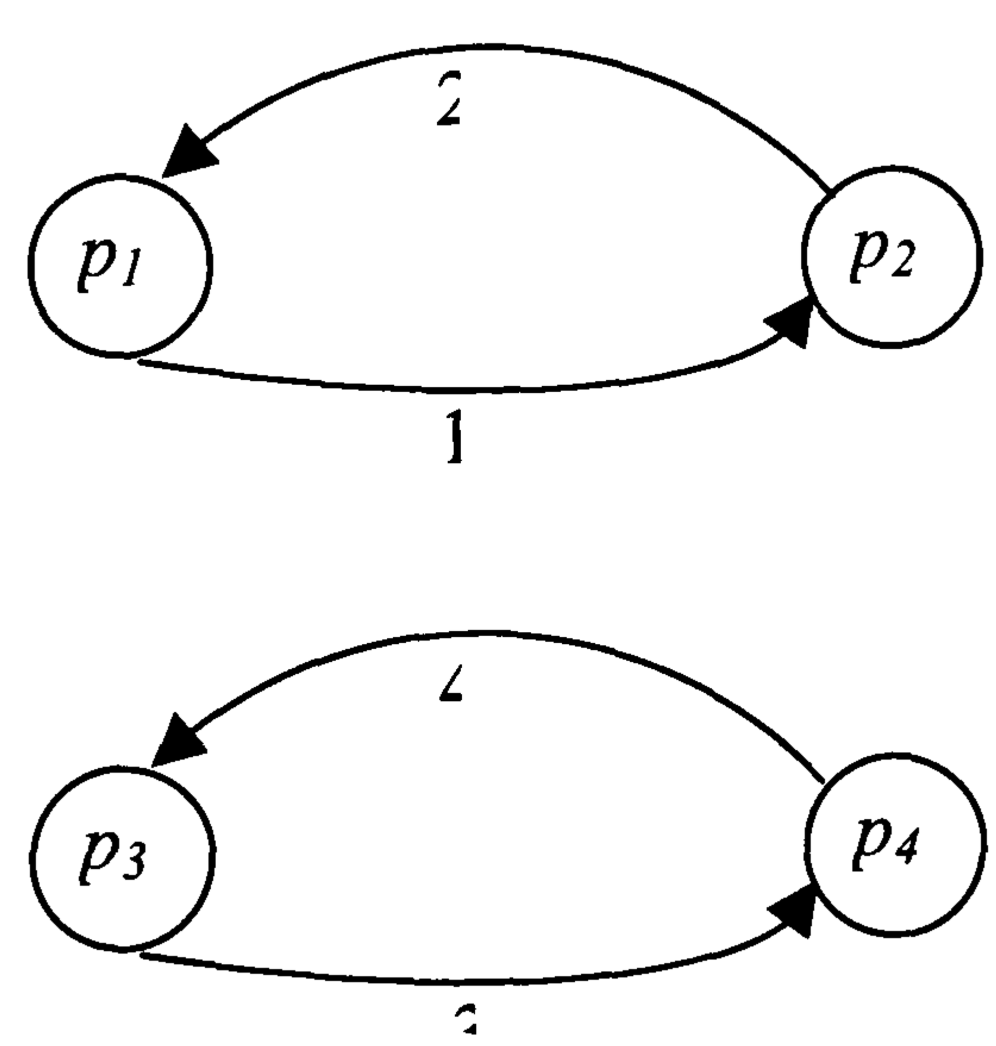

Figure 4.1: Illustrative diagram of sub-tours

Because we need to deal with sub-tours, a new binary variable is introduced in the aggregate model to represent journeys (N_{sjz}).

Where $$N_{sjz} = \begin{cases} 1 & \textit{if } j \textit{ is the } z^{th} \textit{ journey of } s \\ 0 & \textit{otherwise} \end{cases}$$

The added index z represents the order at which journey events occur. This index is essential in the proposed novel sub-tour elimination constraints (Equation 4.9).

As previously mentioned, the aggregate model is only used to generate ship-journey pre-matches for the detailed model to determine the timings of these journeys. However, there is no assurance that the generated pre-matches will produce a feasible detailed solution. Hence, the generation of pre-matches is an iterative procedure (see flow chart in Section 4.2). Note that the set of ports in the aggregate model also includes an extra dummy location. The dummy location acts as a *sink node* in any routing sequence (i.e. every ship has to terminate in the dummy location). Journey costs as well as sailing times to the dummy location are equal to zero.

4.1 Aggregate Mathematical Formulation

Indices

i	Products
l	Ports which include production sites, customer locations, and the dummy location
s	Vendor-owned ships available for transport
j	Possible journeys between ports in addition to journeys to dummy location
z	Journey number $(1,2,...j_{max})$ where j_{max} is estimated (see appendix)
k	Iteration number

Sets

p	Production sites only ($p \subseteq l$)

Parameters

H	Length of the time horizon under consideration
TFD_{il}	Total demand forecasts of product i for VMI customer location l ($TFD_{il} = \sum_t FD_{ilt}$)
TXD_{ip}	Total (non-VMI) demand of product i taken from production site p ($TXD_{il} = \sum_t XD_{ilt}$)

α Penalty cost multiplier in the objective function

C_{sj} Cost of journey j on ship s

τ_{sj} Duration of journey j on ship s

T_l Total quantity that can be loaded/unloaded at port l (tonnes per $\Delta\ t$) based on the pumping rate

SL_j Starting location for journey j

FL_j Finishing location for journey j

ψ_{il} Storage capacity of product i at port l

ψ_s Total storage capacity of ship s

σ_{ip} Maximum production rate of product i at production site p

ξ_p^0 Start of shut-down period at production site p

ξ_p^{end} End of shut-down period at production site p

μ_{ip} Cost of producing a single unit of product i at production site p ($\mu_{ip} > 0$)

ρ_i^{vmi} Selling price of product i to VMI customers

ρ_i^{ext} Selling price of product i to external (non-VMI) customers

$LI\_t_{il}^0$ Initial inventory of product i at port l

$SI\_t_{is}^0$ Initial inventory of product i at ship s

$LI\_t_{il}^{end}$ Final desired inventory of product i at port l

$SI\_t_{is}^{end}$ Final desired inventory of product i at ship s

R_t_{slt} 1 if the initial port for ship s is port l at time t; 0 otherwise

$R\_t_{sl}^0$ 1 if the initial port for ship s is port l; 0 otherwise ($R\_t_{sl}^0 = \sum_t R\_t_{slt}$)

t_s^* Activation time of ship s, $t_s^* = t : \sum_l R\_t_{slt} = 1$

δ_{sl} 1 if ship s is allowed to visit port l ; 0 otherwise

$\eta_{slt^\wedge}$ Length of the maintenance period for ship s at port l starting at time $t^\wedge$; 0 otherwise

Continuous variables

$\hat{Qd}_{isl}$ Total amount of product i discharged from ship s to customer location l over H

$\hat{Qc}_{isl}$ Total amount of product i charged from production site l to ship s over H

TD_{il} Total actual satisfied demand of product i for customer location l ($TD_{il} = \sum_t D_{ilt}$)

$T\Delta_{il}$ Total penalty cost for not meeting demand of product i for location l ($T\Delta_{il} = \sum_t \Delta_{ilt}$)

Binary variables

N_{sjz} 1 if j is the zth journey of ship s; 0 otherwise

The objective in the aggregate model is to maximise the total profit (similar to the detailed model). Note that the transportation term is replaced with aggregate journeys N_{sjz} instead of the detailed journeys X_{sjt}. Equations 4.1-4.15 represent the mathematical formulation of the aggregate model.

Maximise

revenues

$$\sum_i \left(\sum_{l:l \notin p} \rho_i^{vmi} \times TD_{il} + \sum_{l:l \in p} \rho_i^{ext} \times TXD_{il} \right)$$

production

$$-\sum_i \sum_{l:l \in p} \mu_{il} (-TD_{il} - TXD_{il})$$

transportation

$$-\sum_s \sum_j C_{sj} \sum_z N_{sjz}$$

penalty

$$-\alpha \sum_i \sum_{l:l \notin p} T\Delta_{il} \qquad (4.1)$$

Subject to the following Constraints:

<u>Penalty cost</u>

The penalty cost incurred for not meeting demand requirements is calculated by the taking the difference between forecasted and actual fulfilled demands over the entire time horizon for customer locations only.

$$T\Delta_{il} \geq TFD_{il} - TD_{il} \qquad \forall\ i,\ l = \textit{customer location} \qquad (4.2)$$

Loading constraints

The total amounts of material transferred to/from any port should not exceed the capacity of the ship. Note that an extra term is added to the RHS of Constraints 4.3. This term accounts for any material initially existing on the ship.

$$\sum_i \hat{Q}d_{isl} \leq \sum_{j:l=FL_j} \psi_s \sum_z N_{sjz} + (\sum_i SI\_t^0_{is} : [R\_t^0_{sl} = 1]) \qquad \forall\, s, l \quad (4.3)$$

$$\sum_i \hat{Q}c_{isl} \leq \sum_{j:l=SL_j} \psi_s \sum_z N_{sjz} \qquad \forall\, s, l \quad (4.4)$$

Port mass balances

The inventory of each product in each location at the end of the time horizon is equal to any material present initially there in addition to the total material transferred to that location minus any material transferred from that location minus the total demand at that location.

$$LI\_t^{end}_{il} = LI\_t^0_{il} + \sum_s (\hat{Q}d_{isl} - \hat{Q}c_{isl}) - (TD_{il} + TXD_{il}) \qquad \forall\, i, l \quad (4.5)$$

Ship mass balances

The inventory of each product in each ship at the end of the time horizon is equal to any material present initially there in addition to the total material charged to that ship minus any material discharged from that ship.

$$SI\_t^{end}_{is} = SI\_t^0_{is} + \sum_l (Q^{\wedge}c_{isl} - Q^{\wedge}d_{isl}) \qquad \forall\, i, s \quad (4.6)$$

Time constraints

The loading time of first journey only, journey duration times, and unloading times of all journeys should not exceed the total active time horizon ($H+1-t_s^*$). Note that maintenance time $\eta_{s,l,t^{\wedge}}$ is subtracted from the RHS in Constraints 4.7. Loading and unloading time in the aggregate model is estimated by dividing the ship capacity (ψ_s) over the maximum pumping rate per interval (T_l).

$$\sum_{j}\sum_{l=SL_j}\frac{\psi_s}{T_l}\times N_{s,j,z=1}+\sum_{j}(\tau_{sj}+\sum_{l=FL_j}\frac{\psi_s}{T_l})\sum_{z}N_{sjz}\leq(H+1-t_s^{*})-\sum_{l}\sum_{t^{\wedge}}\eta_{slt^{\wedge}} \qquad \forall s \quad (4.7)$$

Journey constraints

The z^{th} journey of each ship must be constrained to a maximum value of one. If no journey is needed, the LHS in Constraints 4.8 will be equal to zero.

$$\sum_{j}N_{sjz}\leq 1 \qquad \forall s,\ z \quad (4.8)$$

Sub-tour elimination constraints

Sub-tours are eliminated using connectivity constraints. Note that each side of the inequality in Constraints 4.9 is at most equal to one. As a result, this constraint assures that a ship can never start a journey from a location (i.e. LHS = 1) unless its previous journey finished at that location (i.e. RHS = 1).

$$\sum_{j:l=SL_j}N_{s,j,z+1}\leq\sum_{j:l=FL_j}N_{s,j,z} \qquad \forall s,\ l,\ z\neq j_{max} \quad (4.9)$$

Logical constraints:

In any tour, every ship has to terminate at the dummy port. Therefore, for every ship, the number of journeys departing towards the dummy port is equal to one (Constraints 4.11). For all other ports (production sites and demand locations), only one of three situations is possible. First, a port is used as the origin of a tour (i.e. a ship is initially there) resulting in a value of -1 in Constraints 4.10. Second, a port is visited during a tour (and existed to another port) resulting in a value of 0 in Constraints 4.10. Third, a port is not visited in a tour also resulting in a value of 0 in Constraints 4.10.

$$-1\leq(\sum_{j:l=FL_j}\sum_{z}N_{s,j,z}-\sum_{j:l=SL_j}\sum_{z}N_{s,j,z})\leq 0 \qquad \forall s,\ l\neq dummy \quad (4.10)$$

$$(\sum_{j:l=FL_j}\sum_{z}N_{s,j,z}-\sum_{j:l=SL_j}\sum_{z}N_{s,j,z})=1 \qquad \forall s,\ l=dummy \quad (4.11)$$

Initial port constraints

A ship has to leave from where it is initially. Note that the RHS in Constraints 4.12 will take a value of either 0 (i.e. ship is not initially there) or a value of -1 (i.e. ship is initially there). Constraints 4.10-4.12 ensure consistent tours in the aggregate model.

$$(\sum_{j:l=FL_j}\sum_{z} N_{s,j,z} - \sum_{j:l=SL_j}\sum_{z} N_{s,j,z}) = -R_{s,l,t_s^*} \qquad \forall\ s,\ l \neq dummy \quad (4.12)$$

Port restriction constraints

Some ships are not allowed to visit certain ports because of many reasons such as depth restrictions. A user specified parameter δ_{sl} is equal to zero if a ship s is not allowed to visit port l.

$$\sum_{j:l=FL_j}\sum_{z} N_{s,j,z} = 0 \qquad \forall\ s,l : \delta_{sl} = 0 \quad (4.13)$$

Ship maintenance constraints

To account for ship maintenance in the aggregate model, we ensure that a ship s exists initially at port l, or make sure that s visits port l at least once. Note that Constraints 4.7 account for the time spent in maintenance. On the other hand, Constraints 4.14 consider only the port at which maintenance is required.

$$\sum_{j:l=FL_j}\sum_{z} N_{s,j,z} + R_{s,l,t_s^*} \geq 1 \qquad \forall\ s,\ l\ iff \sum_{t} \eta_{slt} > 0 \quad (4.14)$$

Integer cuts

As part of the iterative procedure, aggregate solutions that do not lead to feasible detailed solutions are excluded. For each iteration k (except for the first iteration), the integer cuts are added as extra constraints to the aggregate model. At each iteration k, the number of integer cut constraints is k-1. For each iteration k, U_k is the sets of variables N_{sjz} equal to one, and L_k is the sets of variables N_{sjz} equal to zero (Jorgensen, 2000).

$$\sum_{U_k} N_{s,j,z} - \sum_{L_k} N_{s,j,z} \leq card(U_k) - 1 \qquad \forall\, k \neq 1 \quad (4.15)$$

4.2 Iterative Solution Approach

Figure 4.2 shows a flow chart of the iterative approach. The input data are inserted into the aggregate model. An optimal solution to the aggregated problem is generated in terms of ship-journey matches. These ship-journey matches are then inserted into the detailed model for exact timings to be accounted for. To reduce running times, the detailed model is restricted to use the aggregate pre-matches without any changes.

Note that the iterative approach combines the detailed and aggregate model in a sequential manner. First, the aggregate model only (Equations 4.1-4.15) is solved to optimality leading to certain ship-journey pre-matches for the k^{th} iteration (N^k_{sjz}). Then, the detailed model only is solved while restricting the space of binary decisions only to those pre-matches. In other words, an extra constraint (Equation 4.16) is added to the set of detailed constraints (Equations 3.1-3.23). This procedure is repeated every k^{th} iteration.

$$\sum_{t} X_{sjt} = \sum_{z} N^k_{sjz} \qquad \forall\, k, s, j \quad (4.16)$$

Since we use soft demand in the detailed model, a solution will always be generated no matter what pre-matches are used. So, an optimality condition is used as a criterion for stopping the iterative procedure. If the detailed solution is within a certain specified percentage of the aggregate solution (e.g. 5%), then the iterative process stops. In other words, the aggregate solution is used as an upper bound to the detailed solution. To prohibit the iterative algorithm from running infinitely, a maximum running time or/and a maximum number of iterations is specified for the iterative procedure.

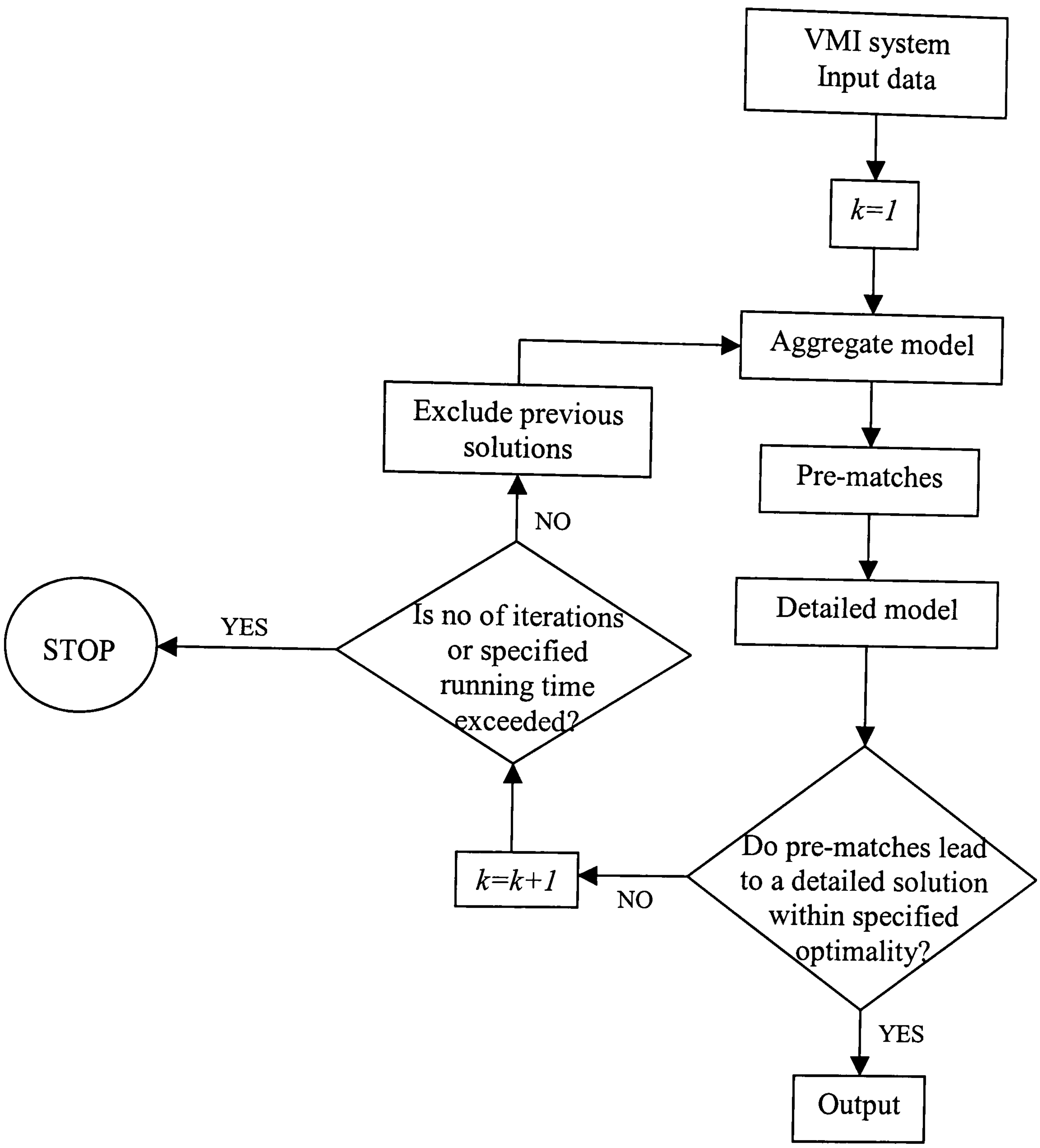

Figure 4.2: Iterative approach general flow chart

It is important to note that deriving the aggregate formulation from the detailed one is not a straightforward process. This is due to the nature of the VMI problem (existence of sub-tours) which necessitates some divergence from the basic aggregate time formulations (i.e. introduction of N_{sjz}). Consequently, proving other aggregate RTN theoretical properties (Wilkinson, 1996) of our formulation is mathematically difficult. Such properties include having an aggregate formulation that is a strict relaxation of the detailed one (i.e. proving that every feasible solution to the detailed problem is also a feasible solution to the aggregate problem). So, it is fair to say that there is no proof that our detailed solution will always be within the aggregate solution feasible region (although this was the case in the illustrative example problem in Chapter 6).

Satisfying these properties is important to validate the iterative approach, where the aggregate solution is used as an upper bound to the detailed one. However, the major contribution of the aggregate model is expected in the rolling horizon algorithm (Chapter 5) where time aggregation is linked with the detailed model. In a rolling horizon framework, the aggregate model is not used as an upper bound to the detailed solution. Instead time aggregation is used only as a tool to predict future decisions and targets. Hence, the impact of having a less tight aggregate formulation is expected to be reduced.

The illustrative example in Chapter 6 shows that optimal solutions are attainable with the use of the iterative approach. A substantial reduction in CPU time is expected with static demand. However, the more dynamic demand is, the greater the gap between the aggregate solution and the detailed solution (see Chapter 6). Hence, we introduce a more robust rolling horizon approach in Chapter 5 to solve the VMI problem.

Chapter 5

Rolling Horizon Approach

5.1 Background and General Definition

Solving planning problems with a rolling horizon has been used in many applications such as process scheduling problems (Dimitriadis, 2000). The idea of the rolling horizon, however, has been available for a long time. The basic concept of the rolling horizon is based on dividing a main problem with a long time horizon into sub-problems with shorter time horizons. As a result, more weight is given to near-future decisions relative to far-future decisions. As the number of variables or the time horizon increases, solving scheduling problems can be very expensive and computing times increase dramatically. Hence, the rolling horizon is also used to reduce the computing times of large problems. The rolling horizon has been used for IRPs by Bard *et al.* (1998) and by Jaillet *et al.* (2002). To our knowledge, no published work on the modelling of ship scheduling exists with a rolling horizon. In our work, we apply the forward rolling horizon (FRH) approach to solve the entire VMI problem. Furthermore, we introduce a novel hybrid forward rolling horizon (HFRH) approach to solve the same VMI problem.

A rolling horizon algorithm is based on dividing the scheduling horizon into a sequence of sub-problems. In each of these sub problems only part of the scheduling problem is solved in detail while the problem is aggregated for the rest of the time horizon. In other words, the time horizon is divided into two time blocks (TBs), one is detailed and the other is aggregate. Figure 5.1 shows a flow chart of the general rolling horizon approach.

The main idea behind the rolling horizon algorithm is that solving the MILP in detail for a small part of the time horizon is relatively simple compared to solving the MILP in detail for the entire time horizon. When the detailed TB covers the entire time horizon, the MILP will still be simpler because some of the variables (only binary variables in our case) will already have been fixed from previous intervals. In addition to producing near-optimal results, this approach can reduce the computation time substantially.

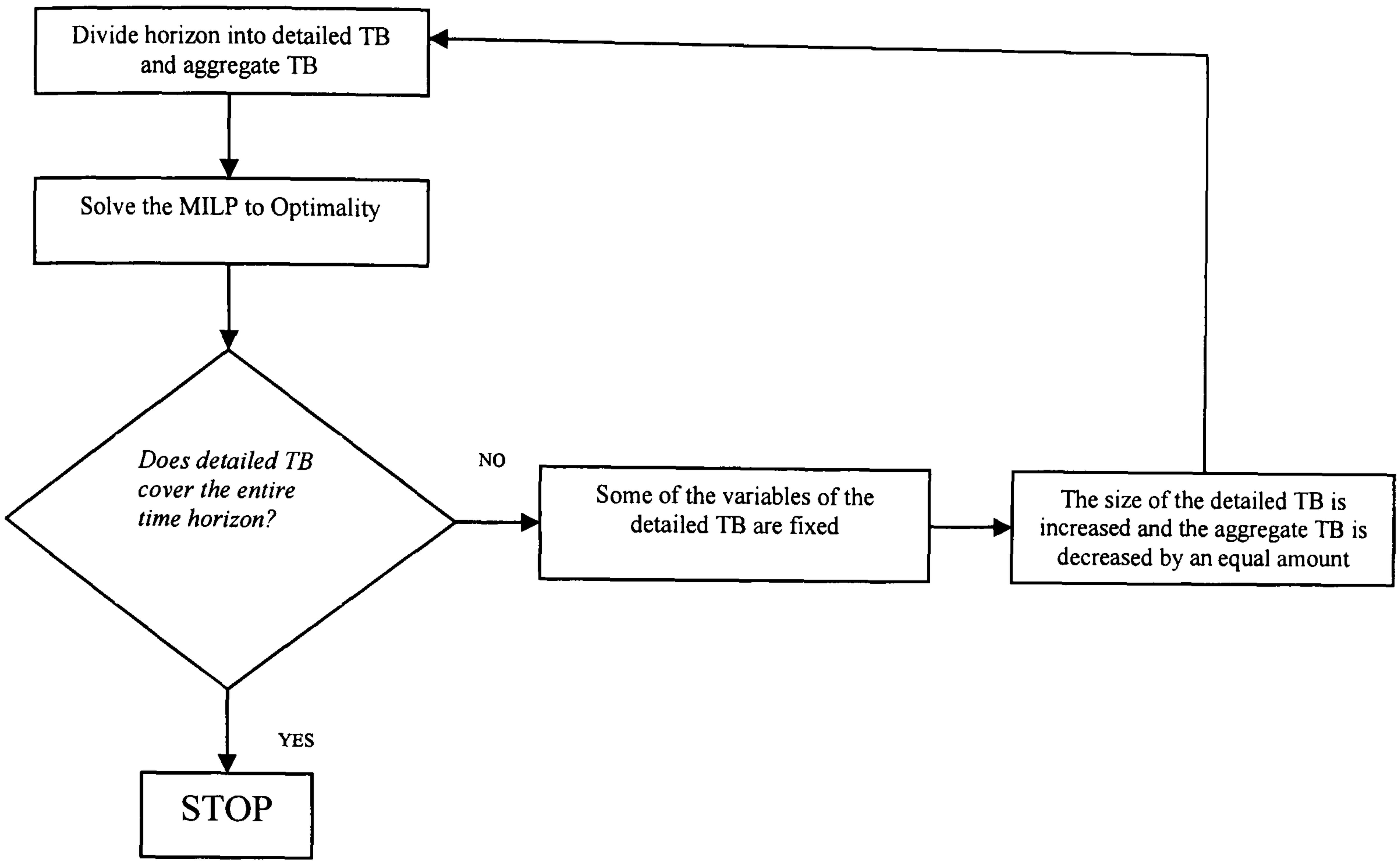

Figure 5.1: General rolling horizon approach flow chart

For a total time horizon of H, the number of iterations (i.e. rolling horizon time intervals) is n. First, H is divided into a detailed TB of one interval and an aggregate TB of $(n-1)$ intervals. In the second iteration, the detailed TB is increased by one interval and the aggregate TB is decreased to $(n-2)$ intervals. This procedure is continued until the detailed TB covers the whole time horizon H. At the beginning of any iteration, some of the variables are fixed as Figure 5.1 shows.

Recall that in Chapter 4, the iterative approach combines the aggregate and the detailed models sequentially to solve the VMI problem. Ship-journey matches provided by the aggregate model were used to produce an optimal solution in the detailed model. However, in the rolling horizon approach, the aggregate and detailed models are combined simultaneously and time aggregation is used for a different reason. We are not interested in any ship-journey matches from the aggregate TB in the rolling horizon approach. However, the two models are linked through boundary conditions to solve the problem (see Section 5.2.1). Time aggregation is used for future journey and inventory considerations. Hence, using time aggregation will force the optimisation process to look ahead and keep enough inventories at locations in order to foresee any future demand needs for those locations. Time aggregation can also be used in a "receding horizon" procedure where only a portion of the solution is implemented and then the scheduling problem is re-solved periodically and updated. This might be very practical in situations where journey times are subject to considerable uncertainty.

Define the two extreme cases of solution algorithms as exact and heuristic. An exact algorithm solves the scheduling problem directly where an optimal solution is guaranteed "ultimately". On the other hand, a heuristic algorithm solves the scheduling problem indirectly by using multi-phase approaches or search algorithms and optimal solutions are not guaranteed. However, the running time reduction in the case of heuristics may be worth the sacrifice of the optimal solution. Since all routing and scheduling problems are NP-hard, popular algorithms to solve these problems are heuristics rather than exact algorithms. Following the same reasoning, solving a problem with a rolling horizon can be considered as a *pseudo-exact* algorithm. The rolling horizon approach is exact in the sense that for the sub-problem in hand, an exact solution is found. A rolling horizon is heuristic approach in the sense that the whole problem is divided into sub-problems.

Theoretically, there is no guarantee that an overall optimal solution will be found using the rolling horizon approach. As Chapter 6 shows, solving VMI problems directly in reasonable times using RTN-type formulations is extremely difficult. Therefore, the rolling horizon approach seems to be a sensible optimisation-based algorithm to solve such problems. Moreover, the rolling horizon algorithm might lead to infeasible solutions due to many reasons. The main reason is the level of accuracy of the aggregate representation. Some variables can be overestimated or underestimated in the aggregate TB, which leads to infeasibilities in the detailed TB of any subsequent iteration. In addition, choosing the boundaries between the aggregate TB and the detailed TB can greatly affect the feasibility of the solution.

However, in our work, infeasibilities that might arise because of the rolling horizon are avoided. First, we only fix binary variables between iterations. Consequently, approximations of continuous variables in the aggregate TB cannot affect the feasibility of the solution. Secondly, we use soft demand constraints instead of hard demand constraints. Consequently, not meeting demand requirements because of the rolling horizon boundaries will not lead to infeasibilities.

5.2 Forward Rolling Horizon Approach

A rolling horizon approach is implemented either in forward or backward fashion. Figure 5.2 shows a diagram of the forward rolling horizon (FRH) approach.

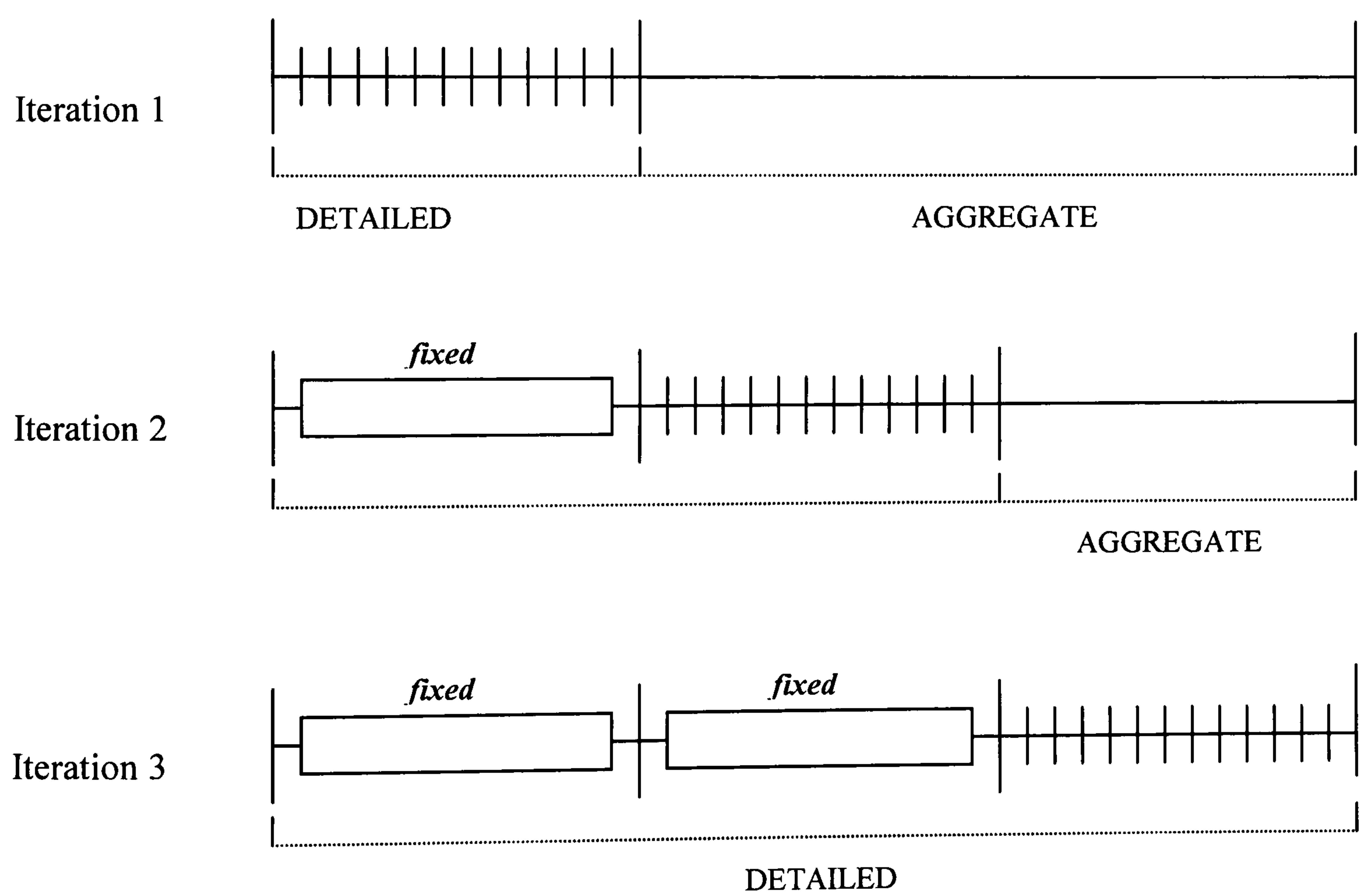

Figure 5.2: Forward rolling horizon diagram for three time intervals

The backward rolling horizon (BRH) approach follows the same procedure except it starts from the end and works backward to the beginning of the time horizon. We only apply FRH because we are dealing with ship routing and scheduling within the VMI system. The final state of the VMI system is not defined as ships can end their journeys in any site/location. However, in FRH, the initial state of the VMI system (initial location of ships and initial inventories) is defined and the optimisation can progress normally. In process scheduling systems, BRH can be applied because the final state of the system is well-defined (e.g. target inventories of materials).

5.2.1 Forward rolling horizon mathematical formulation

The FRH approach explained above uses the detailed model to solve the problem in the detailed TB. The aggregate TB is linked with the detailed TB through two sets of boundary conditions. Those conditions are mass balances and journey continuations. In mass balance constraints, we ensure that the amount of material at the beginning of the aggregate TB is equal to the amount of material at the end of the detailed TB (Equations 5.27 and 5.28). As for journey constraints, we allow ships to start a journey in the detailed TB and end in the aggregate TB (Equations 5.25 and 5.29). If a ship does not start a journey in the detailed TB, then this ship will be at its previous port at the start of the aggregate TB (Equation 5.34). Because the rolling horizon approach combines the detailed and the aggregate models, most of the mathematical information for this approach is shown in Sections 3.4 and 4.1. We only show here any additional indices, parameters, or variables.

Indices

k An index for rolling horizon time intervals (specified by user)

Parameters

w_k^0 Beginning of the k^{th} rolling horizon time interval ($w_1^0 = t^0$)

w_k^{end} End of the k^{th} rolling horizon time interval ($w_{card(k)}^{end} = H+1$)

Continuous variables

$\hat{Q}d_{isl}$ Quantity of product i discharged from ship s to customer location l over aggregate TB

$\hat{Qc}_{isl}$ Quantity of product i charged from production site l to ship s over aggregate TB

Binary variables

N_{sjz} 1 if j is the z^{th} journey of ship s; 0 otherwise

The objective is to maximise the total profit. Profit is defined as the total revenues minus the total costs. The total revenues are represented by total satisfied demand multiplied by the price of each product. The total costs consist of cost of production, transportation, and penalties. Here, we have two types of transportation costs; detailed journey costs and aggregate journey costs. Note that the objective function is optimised k times subject to the same constraints. However, binary variables are fixed while the time domain for each constraint is changing every k^{th} time. Constraints in the detailed TB are generated k times. Constraints in the aggregate TB are generated k-1 times since no aggregate TB exists at the last interval (i.e. the detailed TB covers the remainder of the time horizon). Equations 5.1-5.36 represent the mathematical formulation of the aggregate model. Constraints 5.2-5.15 are general constraints applied over the entire time horizon. Constraints 5.16-5.24 are generated in the detailed TB only. Constraints 5.24-5.36 are generated in the aggregate TB only.

Maximise

revenues

$$\sum_i \sum_t \Big(\sum_{l:l \notin p} \rho_i^{vmi} \times D_{ilt} + \sum_{l:l \in p} \rho_i^{ext} \times XD_{ilt}\Big)$$

production

$$-\sum_i \sum_{l:l \in p} \mu_{il} \sum_t -(D_{ilt} + XD_{ilt}) \tag{5.1}$$

transportation

$$-\sum_s \sum_j C_{sj} \Big(\sum_t X_{sjt} + \sum_z N_{sjz}\Big)$$

penalty

$$-\alpha \sum_i \sum_{l:l \notin p} \sum_t \Delta_{ilt}$$

Subject to the following constraints:

Penalty cost

The penalty cost incurred for not meeting demand requirements is calculated by the taking the difference between forecasted and actual demands at all times. Note that Constraints 5.2 are generated for customer locations only.

$$\Delta_{ilt} \geq FD_{ilt} - D_{ilt} \qquad \forall\ i,\ l=customer\ location,\ t=1,2,\ldots,H \quad (5.2)$$

Initial port constraints

Initially, a ship can either be at a certain port or on the way to that port. If a ship s exists in a port l then $t^* = t^0$ and $R\_t^0_{s,l,t} = 1$ for that ship-port combination. However, if a ship s is on its way to a certain port l, it will be at that port at a time $t^* > t^0$, where, t^* represents the activation time of every ship s. Therefore, Constraints 5.3 ensure that all ships are not activated before their respective t^*. Constraints 5.4 assign the initial port of each ship at its activation time t^*.

$$R_{s,l,t<t^*} = 0 \qquad \forall\ s,\ l,\ t^* = t : (R\_t^0_{s,l,t} = 1) \quad (5.3)$$

$$R_{s,l,t=t^*} = 1 \qquad \forall\ s,\ l,\ t^* = t : (R\_t^0_{s,l,t} = 1) \quad (5.4)$$

Initial inventory constraints

Initial actual inventories of each product at every port (Constraints 5.5) and every ship (Constraints 5.6) are specified by the user.

$$LI_{i,l,t=t0} = LI\_t^0_{il} \qquad \forall\ i,l \quad (5.5)$$

$$SI_{i,s,t=t0} = SI\_t^0_{is} \qquad \forall\ i,s \quad (5.6)$$

Final inventory constraints

Final desired inventories of each product at every port (Constraints 5.7) and every ship (Constraints 5.8) are specified by the user.

$$LI_{i,l,t=H+1} = LI\_t_{il}^{end} \qquad \forall\, i,l \quad (5.7)$$

$$SI_{i,s,t=H+1} = SI\_t_{is}^{end} \qquad \forall\, i,s \quad (5.8)$$

Demand constraints

The actual demand satisfied should not exceed demand forecasts for customer locations (Constraints 5.9). For production sites, demand is negative to represent supply of material. However, supply should not exceed the maximum production rate of each product (Constraints 5.10).

$$0 \le D_{ilt} \le FD_{ilt} \qquad \forall\ i,\ l=customer\ location,\ t=1,2,\dots,H+1 \quad (5.9)$$

$$-\sigma_{il} \le D_{ilt} \le 0 \qquad \forall\ i,\ l=production\ site,\ t=1,2,\dots,H+1 \quad (5.10)$$

Shut-down constraints

During a shut-down period, the daily production rate for each production site is reduced to a user-specified quantity $\sigma_{ip}^{shut} < \sigma_{ip}$.

$$-\sigma_{il}^{shut} \le D_{ilt} \le 0 \qquad \forall\ i,\ l=production\ site,\ t\in[\xi_p^0, \xi_p^{end}] \quad (5.11)$$

Port restriction constraints

Some ships are not allowed to visit certain ports because of many reasons such as depth restrictions. A user specified parameter δ_{sl} is equal to zero if a ship s is not allowed to visit port l.

$$\sum_{j:l=FL_j} \sum_t X_{s,j,t} = 0 \qquad \forall\ s,l : \delta_{sl} = 0 \quad (5.12)$$

Ship maintenance constraints

If maintenance is required for ship s at port l at any time t, then this ship should remain at that port for the total maintenance period $\eta_{s,l,t}$ (Constraints 5.13). In addition, no loading or unloading is allowed during the maintenance period (Constraints 5.14-5.15).

$$R_{s,l,t} = 1 \qquad \forall\ s,\ l,\ t \in [\hat{t},\ \hat{t} + \eta_{s,l,t}] \quad (5.13)$$

$$Qc_{i,s,l,t} = 0 \qquad \forall\ i,\ s,\ l,\ t \in [\hat{t},\ \hat{t} + \eta_{s,l,t}] \quad (5.14)$$

$$Qc_{i,s,l,t} = 0 \qquad \forall\ i,\ s,\ l,\ t \in [\hat{t},\ \hat{t} + \eta_{s,l,t}] \quad (5.15)$$

where $t^\wedge = t : (\eta_{s,l,t} > 0)$

Detailed TB ship allocation constraints

For each ship-port combination at the beginning of each time, that ship exists in that port only if it was there at the previous time period, or it has just arrived from another port. On the other hand, a particular ship does not exist in a particular port at the beginning of a time period if it was not there from the previous time period, or it has just left the location at the beginning of that time period. To avoid infeasibilities, Constraints 5.17 are generated starting with (t^*+1) where t^* is the activation time for every ship. In addition, Constraints 5.17 are generated up to the end of the k^{th} interval.

$$R_{s,l,t} = R_{s,l,t-1} + \sum_{j:l=FL_j} X_{s,j,t'-\tau_{js}} - \sum_{j:l=SL_j} X_{s,j,t} \qquad \forall\ s,\ l,\ t = t^*+1,\ t^*+2, \ldots,\ w_k^{end} \quad (5.17)$$

where $t' > \tau_{js}$

Detailed TB loading time constraints

At any time period, the loading/unloading quantity is less than the maximum loading rate per time period. Constraints 5.17-5.19 assure that a ship s remains in port l for a certain loading/unloading time. Loading/unloading time depends on the pumping rate at l. If a ship is just passing through l, the LHS in Constraints 5.18-5.19 will be equal to zero. Hence, the loading/unloading time (RHS) will be equal to zero. Constraints 5.18-5.19 are generated up to the end of the k^{th} interval.

$$\sum_i Qc_{islt} \le T_l \times R_{slt} \qquad \forall\ s,\ l,\ t = 1, 2, \ldots,\ w_k^{end} \quad (5.18)$$

$$\sum_i Qd_{islt} \le T_l \times R_{slt} \qquad \forall\ s,\ l,\ t = 1, 2, \ldots,\ w_k^{end} \quad (5.19)$$

Detailed TB port mass balances

The forecasted inventory in each port at the beginning of each time interval is equal to the inventory at the beginning of the previous interval in addition to any material transferred to that port at the beginning of that time interval minus any material transferred from that port at the beginning of that time interval minus the forecasted demand taken from that port at the beginning of that time interval. Constraints 5.20 are generated up to the end of the k^{th} interval.

$$LI_{i,l,t} = LI_{i,l,t-1} + \sum_{s}(Qd_{islt} - Qc_{islt}) - (D_{ilt} + XD_{ilt}) \qquad \forall\ i,l,\ t=1,2,\dots,\ w_k^{end} \quad (5.20)$$

Detailed TB ship mass balances

The inventory of each product in each ship at the beginning of each time interval is equal to the value from the last time period, plus any material charged to that ship at the beginning of the time interval minus any material discharged from that ship at the beginning of that time interval. Constraints 5.21 are generated up to the end of the k^{th} interval.

$$SI_{i,s,t} = SI_{i,s,t-1} + \sum_{l}(Qc_{islt} - Qd_{islt}) \qquad \forall\ i,s,\ t=1,2,\dots,\ w_k^{end} \quad (5.21)$$

Detailed TB capacity constraints

For any port or ship, inventories of all products at any time should not exceed the storage capacity. In Constraints 5.22, port storage capacities are given per product. In Constraints 5.23, only ship total storage capacities are considered because of the nature of flexible-compartment ships. Constraints 5.22-5.23 are generated up to the end of the k^{th} interval.

$$LI_{ilt} \le \psi_{il} \qquad \forall\ l,\ t=1,2,\dots,\ w_k^{end} \quad (5.22)$$

$$\sum_{i} SI_{ist} \le \psi_{s} \qquad \forall\ s,\ t=1,2,\dots,\ w_k^{end} \quad (5.23)$$

Detailed TB maximum number of ships in a port

The total number of ships existing at a port should not exceed that port's maximum capacity of ships at any time. Constraints 5.24 are generated up to the end of the k^{th} interval.

$$\sum_{s} R_{slt} \leq \beta_l \qquad \forall\ l,\ t=1,2,\ldots,w_k^{end} \qquad (5.24)$$

Aggregate TB loading constraints

The total amount of material transferred to/from any port should not exceed the capacity of the ship. The last term in the RHS of Constraints 5.25 represents journeys starting in the previous interval and ending in the next k^{th} interval.

$$\sum_{i} Q\hat{\ } d_{isl} \leq \sum_{j:l=FL_j} \psi_s \Big(\sum_{z} N_{sjz} + \sum_{t=w_k^{end}-\tau_{sj}+1}^{t=w_k^{end}-1} X_{sjt}\Big) \qquad \forall\ s,l \qquad (5.25)$$

$$\sum_{i} Q\hat{\ } c_{isl} \leq \sum_{j:l=SL_j} \psi_s \sum_{z} N_{sjz} \qquad \forall\ s,l \qquad (5.26)$$

Aggregate TB port mass balances

The inventory of each product in each location is equal to any material present initially there in addition to the total material transferred to that location minus any material transferred from that location minus the total demand at that location. Constraints 5.27 are generated for the k^{th} aggregate TB only (i.e. $H+1-w_k^{end}$).

$$LI\_t_{il}^{end} = LI_{i,l,t=w_k^{end}} + \sum_{s} (\hat{Q}d_{isl} - \hat{Q}c_{isl}) - \sum_{t} (D_{ilt} + XD_{ilt}) \qquad \forall\ i,l \qquad (5.27)$$

Aggregate TB ship mass balances

The inventory of each product in each ship is equal to any material present initially there in addition to the total material charged to that ship minus any material discharged from that ship. Constraints 5.28 are generated for the k^{th} aggregate TB only (i.e. $H+1-w_k^{end}$).

$$SI\_t_{is}^{end} = SI_{i,l,t=w_k^{end}} + \sum_l (\hat{Q}c_{isl} - \hat{Q}d_{isl}) \qquad \forall\ i,s \quad (5.28)$$

Aggregate TB time constraints

Constraints 5.29 assure that the duration of any sequence of tasks undertaken by any ship will not exceed the aggregate time block. The first term represents the overlapping time of inter-interval ship journey durations in addition to their unloading times. The second term represents the durations of all aggregate journeys in addition to their unloading times. The RHS represents the k^{th} active aggregate time horizon. Note that the maintenance time $\eta_{s,l,t}$ is subtracted from the RHS. Loading and unloading time in the aggregate model is estimated by dividing the ship capacity (ψ_s) by the maximum pumping rate per interval (T_l).

$$\sum_j \sum_{t=w_k^{end}-\tau_{sj}+1}^{t=w_k^{end}-1} (\tau_{sj} - w_k^{end} + t + \sum_{l=FL_j} \frac{\psi_s}{T_l}) \times X_{sjt}$$
$$+ \sum_j (\tau_{sj} + \sum_{l=FL_j} \frac{\psi_s}{T_l}) \sum_z N_{sjz} \leq H + 1 - \max(w_k^0, t_s^*) - \sum_l \sum_{\hat{t}} \eta_{sl\hat{t}} \qquad \forall\ s \quad (5.29)$$

Aggregate TB journey constraints

The z^{th} journey of each ship must be constrained to a maximum value of one. If no journey is needed, the LHS in Constraints 5.30 will be equal to zero.

$$\sum_j N_{sjz} \leq 1 \qquad \forall\ s,\ z \quad (5.30)$$

Sub-tour elimination constraints

Sub-tours are eliminated using connectivity constraints. Note that each side of the inequality below is at most equal to one. As a result, this constraint assures that a ship can never start a journey from a location (i.e. LHS = 1) unless its previous journey finished at that location (i.e. RHS = 1).

$$\sum_{j:l=SL_j} N_{s,j,z+1} \leq \sum_{j:l=FL_j} N_{s,j,z} \qquad \forall\ s,\ l,\ z \neq j_{max} \quad (5.31)$$

Aggregate TB logical constraints:

A ship can start from any port, but it has to end up in the dummy location (similar to Equations 4.10-4.11).

$$-1 \leq \left(\sum_{j:l=FL_j} \sum_{z} N_{s,j,z} - \sum_{j:l=SL_j} \sum_{z} N_{s,j,z} \right) \leq 0 \qquad \forall\ s,\ l \neq dummy \quad (5.32)$$

$$\left(\sum_{j:l=FL_j} \sum_{z} N_{s,j,z} - \sum_{j:l=SL_j} \sum_{z} N_{s,j,z} \right) = 1 \qquad \forall\ s,\ l = dummy \quad (5.33)$$

Initial port constraints

A ship has to leave from where it is initially. Note that the RHS of the Constraints 5.34 will take a value of either 0 or -1. A value of -1 means that a ship was there at the end of the k^{th} interval or it has arrived from the previous interval during an inter-interval journey.

$$\left(\sum_{j:l=FL_j} \sum_{z} N_{s,j,z} - \sum_{j:l=SL_j} \sum_{z} N_{s,j,z} \right) = -\left(\sum_{j=FL_j} \sum_{t=w_k^{end}-\tau_{sj}+1}^{t=w_k^{end}-1} X_{sjt} + R_{s,l,t=w_k^{end}} \right) \qquad \forall\ s,\ l \neq dummy \quad (5.34)$$

Aggregate TB port restriction constraints

Some ships are not allowed to visit certain ports because of many reasons such as depth restrictions. A user specified parameter δ_{sl} is equal to zero if a ship s is not allowed to visit port l.

$$\sum_{j:l=FL_j} \sum_{z} N_{s,j,z} = 0 \qquad \forall\ s,l : \delta_{sl} = 0 \quad (5.35)$$

Aggregate TB ship maintenance constraints

To account for ship maintenance in the aggregate model, we ensure that a ship s exists initially at port l, or it has made an inter-interval journey to l, or makes an aggregate journey to port l at least once.

$$\sum_{j=FL_j} \sum_{t=w_k^{end}-\tau_{sj}+1}^{t=w_k^{end}-1} X_{sjt} + \sum_{j:l=FL_j} \sum_{z} N_{sjz} + R_{s,l,t=w_k^{end}} \geq 1 \qquad \forall\ s,\ l \neq dummy,\ t' \quad (5.36)$$

where $t' = t : (\eta_{s,l,t} > 0$ *and* $t > w_k^{end})$

Equations 5.1-3.36 above conclude the FRH mathematical model which combines the detailed model (Chapter 3) and the aggregate model (Chapter 4). Note that the FRH includes one detailed TB and one aggregate TB. In Section 5.3 we introduce a novel hybrid FRH model that also includes one detailed TB and one aggregate TB. However, another type of aggregation is implemented and included in the overall model in order to further improve the proposed FRH approach.

5.3 Hybrid Forward Rolling Horizon Approach

Figure 5.3 shows a flow chart of the HFRH. In each time interval, the problem is not solved for all available ships in the detailed TB. Instead, we start with one ship and solve for that ship only. Then, another ship is added and we solve for that ship while the schedule of the first ship is fixed. This procedure continues until we cover all available ships. However, while solving in detail for each ship; other ships are represented in an aggregate manner during that detailed TB.

The reason for ship aggregation is to allow every ship to consider the performance of other ships during the optimisation process. However, ship aggregation can not capture the exact details of the performance of all ships. Consequently, optimal solutions can be missed because every ship's schedule is fixed between iterations. Moreover, the order in which ships are passed to the solver can affect the final solution. Therefore, it is fair to say that the HFRH approach does not guarantee optimality.

Nevertheless, ship aggregation was successful to eliminate the effect of the order of ships in case of the illustrative example. As a result, optimal solutions were attained with an arbitrary order of ships (see Chapter 6).

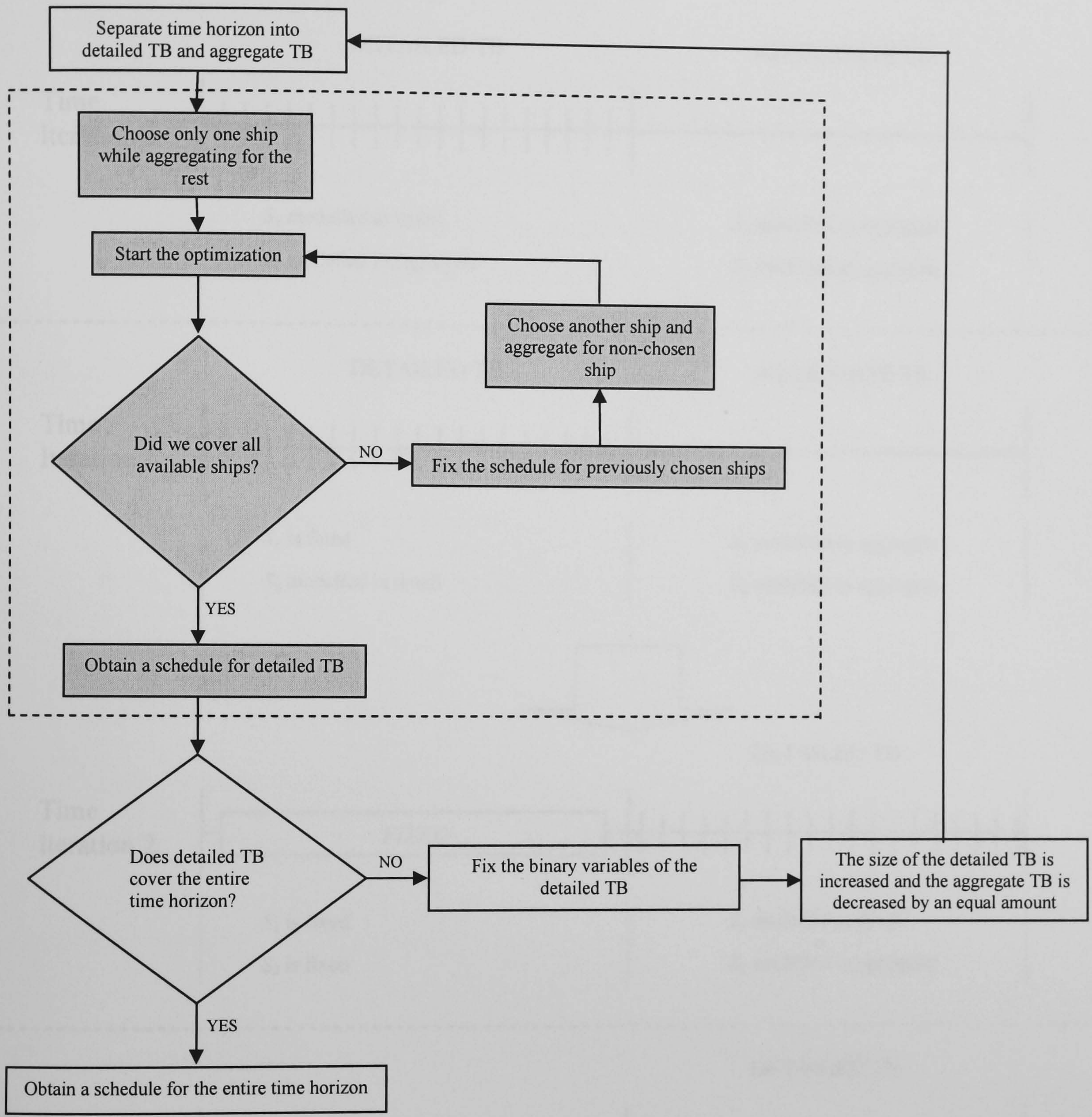

Figure 5.3: Hybrid forward rolling horizon flow chart

The diagram in Figure 5.4 shows how the HFRH works for a case of two ships and two time intervals. Note that we execute the optimisation four times (i.e. $\text{card}(k) \times \text{card}(s)$). In the 'fixed' parts of the problem, the binary variables are fixed, but the continuous ones can be re-optimised.

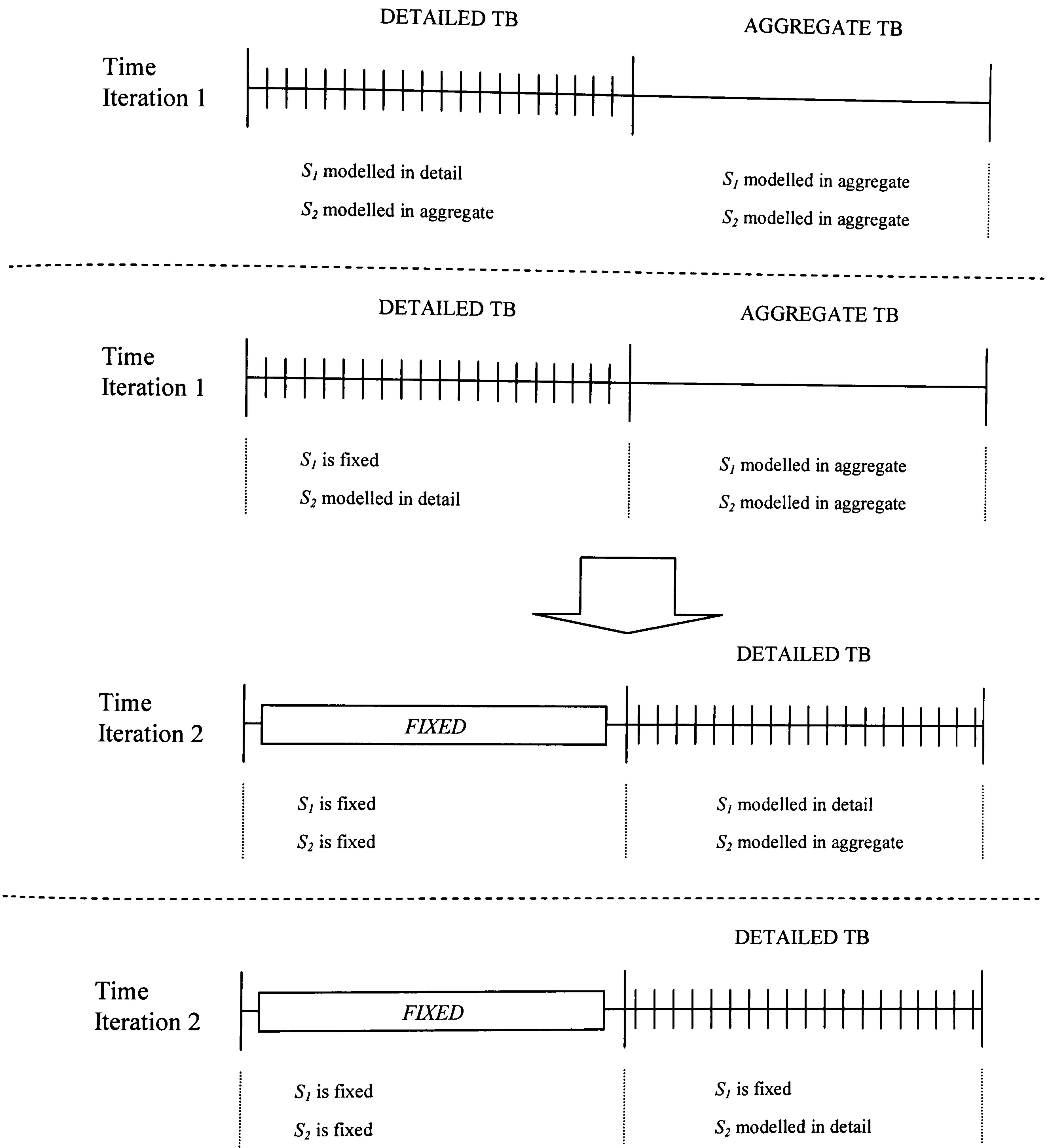

Figure 5.4: Hybrid forward rolling horizon diagram for two time intervals and two ships

5.3.1 Hybrid forward rolling horizon mathematical formulation

The HFRH approach explained above uses the detailed model to solve the problem in the detailed TB for a set of detailed ships s_{det} and set of aggregate ships s_{agg}. The aggregate TB is linked with the detailed TB through two sets of boundary conditions (see Section 5.2.1). Most of the mathematical information for this approach is shown in section 5.2.1. We only show here any additional indices, sets or variables.

Indices

k an index for rolling horizon time intervals (specified by user)

z an index for aggregate journey number $(1,2,\ldots j_k^{\max 2})$ in the aggregate TB

y an index for aggregate journey number $(1,2,\ldots j_k^{\max 1})$ in the detailed TB

Sets

s_{det} ships modelled in detail during detailed TB ($s_{det} \subseteq s$)

s_{agg} ships modelled in aggregate during detailed TB ($s_{agg} \subseteq s$)

Continuous variables

$\overset{1\wedge}{Qd}_{isl}$ Quantity of product i discharged from ship s to customer location l over 1st aggregation

$\overset{1\wedge}{Qc}_{isl\,l}$ Quantity of product i charged from production site l to ship s over 1st aggregation

$\overset{2\wedge}{Qd}_{isl}$ Quantity of product i discharged from ship s to customer location l over 2nd aggregation

$\overset{2\wedge}{Qc}_{isl\,l}$ Quantity of product i charged from production site l to ship s over 2nd aggregation

Binary variables

N_{sjz} 1 if j is the z^{th} journey of ship s; 0 otherwise (2nd aggregation)

M_{sjy} 1 if j is the y^{th} journey of ship s; 0 otherwise (1st aggregation)

The objective is to maximise the total profit. Profit is defined as the total revenues minus the total costs. The total revenues are represented by the total demand satisfied at all customer locations multiplied by the price of each product. The total costs consist of the cost of production, transportation, and penalties. Here, we have three types of transportation costs; detailed journey costs X and aggregate journey costs M and N. Note that the objective function is optimised *(card(k).card(s))* times subject to the same constraints. Every k^{th} interval, we start with detailed ships s_{det} and aggregate ships s_{agg}. During that interval, the problem is solved to optimality, previous s_{det} are fixed, and s_{agg} is reduced. This procedure is continued until we cover all ships in detail during that interval (i.e. $s_{agg} = \emptyset$ and $s_{det} = s$). We move to the next interval using the same strategy until we cover all intervals. Equations 5.37-5.61 represent the mathematical formulation of the HFRH approach.

Maximise

revenues

$$\sum_i \sum_t (\sum_{l:l \notin p} \rho_i^{vmi} \times D_{ilt} + \sum_{l:l \in p} \rho_i^{ext} \times XD_{ilt})$$

production

$$-\sum_i \sum_{l:l \in p} \mu_{il} \sum_t -(D_{ilt} + XD_{ilt})$$

transportation

$$-\sum_s \sum_j C_{sj} (\sum_t X_{sjt} + \sum_y M_{sjy} + \sum_z N_{sjz})$$

penalty

$$-\alpha \sum_i \sum_{l:l \notin p} \sum_t \Delta_{ilt} \tag{5.37}$$

Subject to the following constraints:

All general and detailed constraints are similar to those of the FRH approach described by Equations 5.2-5.24. Here we only show the unique aggregate constraints. As previously mentioned, we have two types of aggregations. Constraints 5.38-5.49 represent the 1st aggregation constraints which deal with aggregate ships in the detailed TB. In the 1st aggregation constraints, journeys are represented using the variable M_{sjy}. Note that the 1st aggregation constrains are generated for the subset s_{agg}

only. On the other hand, Constraints 5.50-5.61 represent the 2nd aggregation constraints which deal with ships in the aggregate TB. In the 2nd aggregation constraints, journeys are represented using the variable N_{sjz}. Note that the 2nd aggregation constraints are generated for the subset s_{det} only.

1st aggregation discharge constraints

The total material discharged into any port should not exceed the capacity of that ship. In addition to aggregate journey capacity constraints, the RHS contains two extra terms. First term is generated during the first time interval only and accounts for any material initially existing on that ship. The second term is generated for all successive intervals ($k>1$) and accounts for material on ships making inter-interval journeys.

$$\begin{aligned}
&\sum_i Q^{1\wedge} d_{isl} \le \\
&(\sum_i SI_t^0_{is} : [R_{s,l,t=t^*} = 1]) : k = 1 \\
&+ (\sum_{j:l=FL_j} \sum_{t=w_k^0-\tau_{sj}+1}^{t=w_k^0-1} X_{sjt}) : k > 1 \\
&\sum_{j:l=FL_j} \psi_s * \sum_y M_{sjy}
\end{aligned} \qquad \forall\, s = s_{agg},\, l \quad (5.38)$$

1st aggregation charge constraints

The total material charged into any ship from any port should not exceed the capacity of the ship.

$$\sum_i Q^{1\wedge} c_{isl} \le \sum_{j:l=SL_j} \psi_s \sum_y M_{sjy} \qquad \forall\, s = s_{agg},\, l \quad (5.39)$$

1st aggregation port mass balances

The Inventory of each product in each location is equal to any material present initially there in addition to the total material transferred to that location minus any material transferred from that location minus the total demand at that location. Note that, Constraints 5.40 are generated only for the k^{th} detailed TB under optimisation (i.e. w_k^{end} -w_k^0).

$$
\begin{aligned}
LI_{i,l,t=w_k^{end}} &= LI_{i,l,t=w_k^0} \\
&+ \sum_{s=s_{agg}} (\overset{1\wedge}{Q}d_{isl} - \overset{1\wedge}{Q}c_{isl}) \\
&+ \sum_{s=s_{\det}} \sum_{t=w_k^0}^{t=w_k^{end}} (Qd_{islt} - Qc_{islt}) \qquad \forall\, i, l \quad (5.40) \\
&- \sum_{t=w_k^0}^{t=w_k^{end}} (D_{ilt} + XD_{ilt})
\end{aligned}
$$

1st aggregation ship mass balances

The inventory of each product on each ship is equal to any material present initially there in addition to the total material charged to that ship minus any material discharged from that ship. Constraints 5.41 are generated for aggregate ships during the k^{th} detailed TB under optimisation (i.e. w_k^{end} - w_k^0).

$$
SI_{i,l,t=w_k^{end}} = SI_{i,l,t=w_k^0} + \sum_{l} (\overset{1\wedge}{Q}c_{isl} - \overset{1\wedge}{Q}d_{isl}) \qquad \forall\, i,\ s=s_{agg} \quad (5.41)
$$

1st aggregation time constraints

Constraints 5.42 assure that the duration of any sequence of tasks undertaken by any ship will not exceed the aggregate time block. The first term represent material initially present at ships. The second term represents the overlapping time of inter-interval ship journey durations in addition to their unloading times. The third and fourth terms represent the times of all aggregate journeys in addition to their unloading times. The RHS represents the k^{th} active aggregate time horizon. Note that maintenance time $\eta_{s,l,t}$ is subtracted from the RHS. Loading and unloading time in the aggregate model is estimated by dividing the ship capacity (ψ_s) by the maximum pumping rate per interval (T_l).

$$(\sum_{j}\sum_{l=SL_j}\frac{\psi_s}{T_l}\times M_{s,j,y=1}):k=1$$

$$(\sum_{j}\sum_{t=w_k^0-\tau_{sj}+1}^{t=w_k^0-1}(\tau_{sj}-w_k^0+t+\sum_{l=FL_j}\frac{\psi_s}{T_l})\times X_{sjt}):k>1$$

$$+\sum_{j}(\tau_{sj}+\sum_{l=FL_j}\frac{\psi_s}{T_l})\sum_{y}M_{sjy} \qquad \forall\ s=s_{agg} \qquad (5.42)$$

$$+\sum_{j}(\tau_{sj}+\sum_{l=FL_j}\frac{\psi_s}{T_l})\sum_{z}N_{sjz}$$

$$\leq H+1-\max(w_k^0,t_s^*)-\eta_{s,l,t^{\wedge}}$$

<u>1st aggregation journey constraints</u>

The y^{th} journey of each ship must be constrained to a maximum value of one. If no journey is needed, the LHS of Constraints 5.43 will be equal to zero.

$$\sum_{j}M_{sjy}\leq 1 \qquad \forall\ s=s_{agg},\ y \qquad (5.43)$$

<u>1st aggregation sub-tour elimination constraints</u>

Sub-tours are eliminated using connectivity constraints. Note that each side of the inequality below is at most equal to one. As a result, Constraints 5.44 assure that a ship can never start a journey from a location (i.e. LHS = 1) unless its previous journey finished at that location (i.e. RHS = 1).

$$\sum_{j:l=SL_j}M_{s,j,y+1}\leq\sum_{j:l=FL_j}M_{s,j,y} \qquad \forall\ s=s_{agg},\ l,\ y\neq j_{\max}^{1} \qquad (5.44)$$

<u>1st aggregation logical constraints:</u>

A ship can start from any port, but it has to end up in the dummy location.

$$-1\leq(\sum_{j:l=FL_j}\sum_{y}M_{s,j,y}-\sum_{j:l=SL_j}\sum_{y}M_{s,j,y})\leq 0 \qquad \forall\ s=s_{agg},\ l\neq dummy \qquad (5.45)$$

$$(\sum_{j:l=FL_j}\sum_{y}M_{s,j,y}-\sum_{j:l=SL_j}\sum_{y}M_{s,j,y})=1 \qquad \forall\ s=s_{agg},\ l=dummy \qquad (5.46)$$

1st aggregation initial port constraints

A ship has to leave from where it is initially. Note that the RHS of constraints 5.47 will take a value of either 0 or -1. A value of -1 means that a ship was there at the end of the k^{th} interval or it has arrived from the previous interval during an inter-interval journey.

$$\left(\sum_{j:l=FL_j}\sum_{y} M_{s,j,y} - \sum_{j:l=SL_j}\sum_{y} M_{s,j,y}\right) = -\left(\sum_{j=FL_j}\sum_{t=w_k^0-\tau_{sj}+1}^{t=w_k^0-1} X_{sjt} + R_{s,l,t=w_k^0}\right) \qquad \forall\ s=s_{agg},\ l\neq dummy \quad (5.47)$$

1st aggregation port restriction constraints

Some ships are not allowed to visit certain ports because of many reasons such as depth restrictions. A user specified parameter δ_{sl} is equal to zero if a ship s is not allowed to visit port l.

$$\sum_{j:l=FL_j}\sum_{y} M_{s,j,y} = 0 \qquad \forall\ s=s_{agg}, l : \delta_{sl}=0 \quad (5.48)$$

1st aggregation ship maintenance constraints

To account for ship maintenance in the detailed TB, we ensure that a ship s exists initially at port l, or it has made an inter-interval journey to l, or makes an aggregate journey to port l at least once.

$$\sum_{j=FL_j}\sum_{t=w_k^0-\tau_{sj}+1}^{t=w_k^0-1} X_{sjt} + \sum_{j:l=FL_j}\sum_{y} M_{sjy} + R_{s,l,t=w_k^0} \geq 1 \qquad \forall\ s=s_{agg},\ l\neq dummy,\ t' \quad (5.49)$$

where $t'=t : (\eta_{s,l,t} > 0$ *and* $w_k^0 < t < w_k^{end})$

2nd aggregation loading Constraints

The total material transferred to/from any port should not exceed the capacity of the ship. The last term in the RHS of Constraints 5.50 represents journeys starting in the previous interval and ending in the next k^{th} interval.

$$\sum_i Q^{2\wedge}d_{isl} \leq \sum_{j:l=FL_j} \psi_s(\sum_z N_{sjz} + \sum_{t=w_k^{end}-\tau_{sj}+1}^{t=w_k^{end}-1} X_{sjt}) \qquad \forall\ s=s_{det},\ l \quad (5.50)$$

$$\sum_i Q^{2\wedge}c_{isl} \leq \sum_{j:l=SL_j} \psi_s \sum_z N_{sjz} \qquad \forall\ s=s_{det},\ l \quad (5.51)$$

<u>2nd aggregation port mass balances</u>

The inventory of each product in each location is equal to any material present initially there in addition to the total material transferred to that location minus any material transferred from that location minus the total demand at that location. Constraints 5.52 are generated for the k^{th} aggregate TB only (i.e. $H+1-w_k^{end}$).

$$LI\_t_{il}^{end} = LI_{i,l,t=w_k^{end}} + \sum_s (\overset{2\wedge}{Q}d_{isl} - \overset{2\wedge}{Q}c_{isl}) - \sum_t (D_{ilt} + XD_{ilt}) \qquad \forall\ i,\ l \quad (5.52)$$

<u>2nd aggregation ship mass balances</u>

The inventory of each product on each ship is equal to any material present initially there in addition to the total material charged to that ship minus any material discharged from that ship. Constraints 5.53 are generated for the k^{th} aggregate TB only (i.e. $H+1-w_k^{end}$).

$$SI\_t_{is}^{end} = SI_{i,l,t=w_k^{end}} + \sum_l (\overset{2\wedge}{Q}c_{isl} - \overset{2\wedge}{Q}d_{isl}) \qquad \forall\ i,\ s \quad (5.53)$$

<u>2nd aggregation time Constraints</u>

Constraints 5.54 assure that the duration of any sequence of tasks undertaken by any ship will not exceed the aggregate time block. The first term represents the overlapping time of inter-interval ship journey durations in addition to their unloading times. The second term represents the times of all aggregate journeys in addition to their unloading times. The RHS represents the k^{th} active aggregate time horizon. Note that maintenance time $\eta_{s,l,t}$ is subtracted from the RHS. Loading and unloading time in the aggregate model is estimated by dividing the ship capacity (ψ_s) by the maximum pumping rate per interval (T_l).

$$\sum_{j}\sum_{t=w_k^{end}-\tau_{sj}+1}^{t=w_k^{end}-1}(\tau_{sj}-w_k^{end}+t+\sum_{l=FL_j}\frac{\psi_s}{T_l})\times X_{sjt}$$

$$+\sum_{j}(\tau_{sj}+\sum_{l=FL_j}\frac{\psi_s}{T_l})\sum_{z}N_{sjz} \qquad \forall\ s=s_{det} \quad (5.54)$$

$$\leq H+1-\max(w_k^0,t_s^*)-\eta_{s,l,t^{\wedge}}$$

<u>2nd aggregation journey constraints</u>

The z^{th} journey of each ship must be constrained to a maximum value of one. If no journey is needed, the LHS of constraints 5.55 will be equal to zero.

$$\sum_{j}N_{sjz}\leq 1 \qquad \forall\ s,\ z \quad (5.55)$$

<u>2nd aggregation sub-tour elimination constraints</u>

Sub-tours are eliminated using connectivity constraints. Note that each side of the inequality below is at most equal to one. As a result, Constraints 5.56 assures that a ship can never start a journey from a location (i.e. LHS = 1) unless its previous journey finished at that location (i.e. RHS = 1).

$$\sum_{j:l=SL_j}N_{s,j,z+1}\leq\sum_{j:l=FL_j}N_{s,j,z} \qquad \forall\ s,\ l,\ z\neq j^2_{\max} \quad (5.56)$$

<u>2nd aggregation logical constraints:</u>

A ship can start from any port, but it has to end up in the dummy location.

$$-1\leq(\sum_{j:l=FL_j}\sum_{z}N_{s,j,z}-\sum_{j:l=SL_j}\sum_{z}N_{s,j,z})\leq 0 \qquad \forall\ s,\ l\neq dummy \quad (5.57)$$

$$(\sum_{j:l=FL_j}\sum_{z}N_{s,j,z}-\sum_{j:l=SL_j}\sum_{z}N_{s,j,z})=1 \qquad \forall\ s,\ l=dummy \quad (5.58)$$

<u>2nd aggregation initial port constraints</u>

A ship has to leave from where it is initially. Note that the RHS of constraints 5.59 will take a value of either 0 or -1. For detailed ships, a value of -1 means that a ship was there at the end of the k^{th} interval or it has arrived from the previous interval during an inter-interval journey. For aggregate ships, the terminal port is the port ship has just left on its way to the dummy location.

$$
\begin{aligned}
&(\sum_{j:l=FL_j}\sum_{z} N_{s,j,z} - \sum_{j:l=SL_j}\sum_{z} N_{s,j,z}) = \\
&-(\sum_{j=FL_j}\sum_{t=w_k^{end}-\tau_{sj}+1}^{t=w_k^{end}-1} X_{sjt} + R_{s,l,t=w_k^{end}}) : s = s_{\det} \\
&-(\sum_{j:FL_j=dummy}\sum_{y} M_{s,j,y}) : s = s_{agg}
\end{aligned}
\qquad \forall\ s,\ l = dummy \quad (5.59)
$$

<u>2nd aggregation port restriction constraints</u>

Some ships are not allowed to visit certain ports because of many reasons such as depth restrictions. A user specified parameter δ_{sl} is equal to zero if a ship s is not allowed to visit port l.

$$
\sum_{j:l=FL_j}\sum_{z} N_{s,j,z} = 0 \qquad \forall\ s,l : \delta_{sl} = 0 \quad (5.60)
$$

<u>2nd aggregation ship maintenance constraints</u>

To account for ship maintenance in the aggregate model, we ensure that a ship s exists initially at port l, or it has made an inter-interval journey to l, or makes an aggregate journey to port l at least once.

$$
\sum_{j=FL_j}\sum_{t=w_k^{end}-\tau_{sj}+1}^{t=w_k^{end}-1} X_{sjt} + \sum_{j:l=FL_j}\sum_{z} N_{sjz} + R_{s,l,t=w_k^{end}} \geq 1 \qquad \forall\ s,\ l \neq dummy,\ t' \quad (5.61)
$$

$$
where\ t' = t : (\eta_{s,l,t} > 0\ and\ t > w_k^{end})
$$

The FRH and HFRH approaches involve many decisions that are problem specific. Such decisions deal with the number of time intervals used and the position of time boundaries between the detailed and aggregate time blocks. Time intervals/boundaries can affect the optimisation process in terms of the optimal solutions and running times. Those intervals/boundaries should be selected based on the length of the time horizon and the durations of tasks involved. Therefore, a rolling horizon time interval must not be too long or it will result in large CPU times. In addition, a rolling horizon interval must not be too short to exclude some tasks. In our problem, the effect of time intervals is minimised due to the fact that we allow a journey to start in one interval and terminate in the next interval. For example, assume that a ship performs certain tasks in the detailed TB with only two remaining days in that TB. This ship can start another inter-interval journey (J) of five days starting in the detailed TB and ending in the aggregate TB. As a result, three days of that ship's total time will be subtracted from the aggregate TB (see Figure 5.5). We also assure that the next journey in the aggregate TB starts from where the previous journey ended. In addition, we account for any material carried over from the detailed TB to the aggregate TB though the proper mass balance equations (see Section 5.2.1).

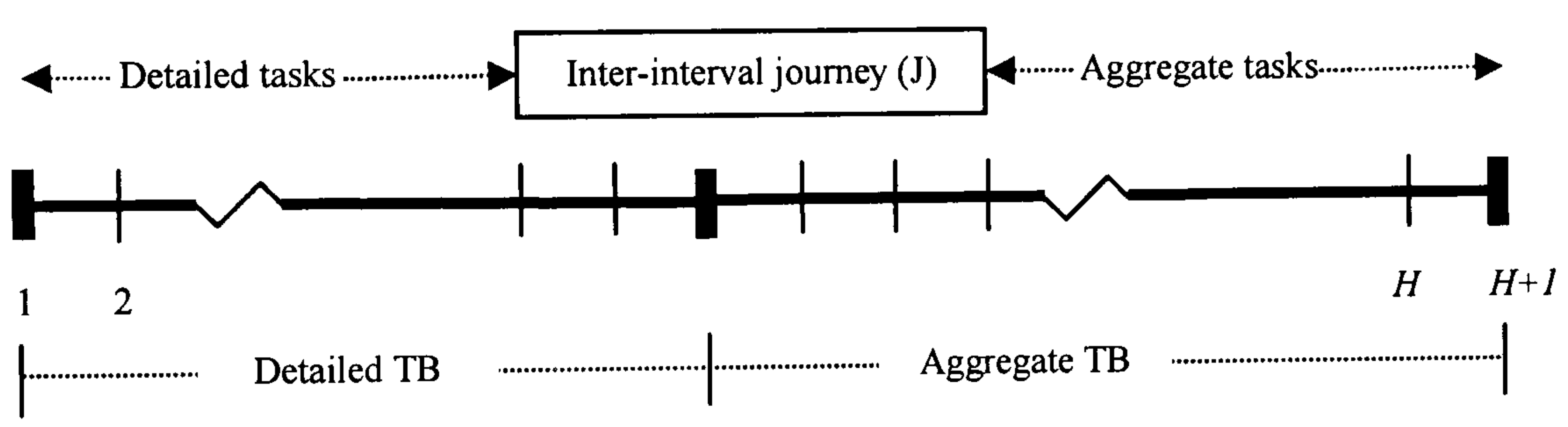

Figure 5.5: Illustrative diagram for inter-intervals journeys

In the next chapter all the mathematical approaches described in Chapters 3, 4 and 5 are tested using an illustrative example. Results show that the FRH and the HFRH are the best solution approaches compared to the direct and iterative approaches.

Chapter 6

Results and Discussion

The GAMS software (Brooke *et al.*, 1992) was used to model the VMI system. All GAMS runs were performed on the Linux machines using the CPLEX package v9.5 as the MILP solver.

6.1 Illustrative Example Specifications

To evaluate the solution approaches described earlier, an illustrative example is solved in this chapter using different demand data. Two sets of demand data are compared. The first set uses static demand while the second set uses dynamic demand. The illustrative example specifications are explained below.

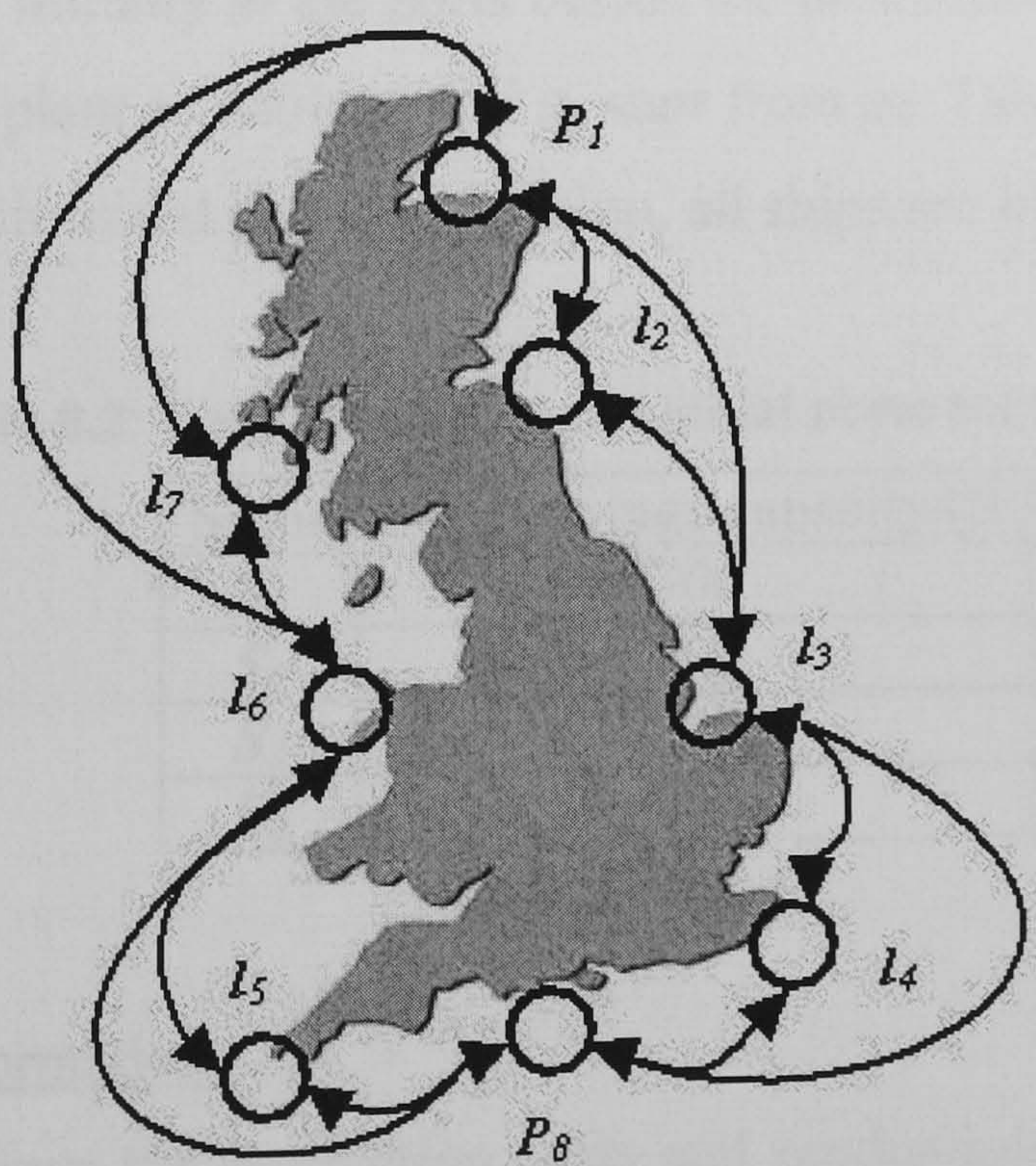

Figure 6.1: Illustrative example port layout on the UK map

Available ports:

As the map in Figure 6.1 shows, ports include two production plants and six customer locations. The capacity of each port is given in metric tonnes (t) as shown in Table 6.1.

Table 6.1: Port capacities for the illustrative example

Production plants	Total storage capacity (t)
p_1	1000
p_8	1000
Customers	
l_2	500
l_3	500
l_4	500
l_9	500
l_9	500
l_9	500

Available ships:

All ships are initially at the ports beside the production plants. We assume that s_1 and s_3 start from plant p_1 while s_2 and s_4 start from p_8. Table 6.2 shows the ship capacities as well as their initial ports. In addition, all ships are initially empty.

Table 6.2: Ship capacities and initial ports for the illustrative example

Ships	Total storage capacity (t)	Initial port
S_1	300	at p_1
S_2	300	at p_8
S_3	300	at p_1
S_4	300	at p_8

Products information:

Table 6.3 shows the production costs and production rates of the single product i at every production plant.

Table 6.3: Production costs and rates for the illustrative example

Product name	Price ($/t)	Production cost ($/t) at		Production rate (t/day) at A	
		p_1	p_8	p_1	p_8
i	2000	1000	1000	200	200

Journey information:

Note that journeys on different ships can have different durations and different costs. However, inbound and outbound journey durations and costs are the same for each ship. In this example, we assume that the journey durations are fixed for all ships while the journey costs are different. Table 6.4 shows the journey costs and durations on every ship in the fleet.

Table 6.4: Journey costs and durations for the illustrative example

Journey no.	From	To	S_1		S_2		S_3		S_4	
			$\$\times10^3$	days	$\$\times10^3$	days	$\$\times10^3$	days	$\$\times10^3$	days
1	p_1	l_2	51	1	62	1	107	1	83	1
2	p_1	l_3	75	2	85	2	128	2	110	2
3	p_1	l_9	46	1	61	1	100	1	74	1
4	p_1	l_9	99	2	106	2	143	2	137	2
5	l_2	p_1	51	1	62	1	107	1	83	1
6	l_3	p_1	75	2	85	2	128	2	110	2
7	l_9	p_1	46	1	61	1	100	1	74	1
8	l_9	p_1	99	2	106	2	143	2	137	2
9	l_2	l_3	79	1	83	1	64	1	83	1
10	l_3	l_2	79	1	83	1	64	1	83	1
11	l_4	l_3	73	1	80	1	66	1	87	1
12	l_3	l_4	73	1	80	1	66	1	87	1
13	l_9	l_9	57	1	73	1	67	1	74	1
14	l_9	l_9	57	1	73	1	67	1	74	1
15	l_9	l_9	63	1	77	1	88	1	57	1
16	l_9	l_9	63	1	77	1	88	1	57	1
17	p_8	l_4	57	1	73	1	67	1	74	1
18	p_8	l_3	79	2	83	2	130	2	115	2
19	p_8	l_9	52	1	66	1	104	1	88	1
20	p_8	l_9	89	2	84	2	124	2	113	2
21	l_4	p_8	57	1	73	1	67	1	74	1
22	l_3	p_8	79	2	83	2	130	2	115	2
23	l_9	p_8	52	1	66	1	104	1	88	1
24	l_9	p_8	89	2	84	2	124	2	113	2

Loading times:

A ship requires different loading/unloading times depending on the pumping rate in the port. In this example, we assume that all loading and unloading times are fixed at one day.

Time horizon:

The scheduling horizon under consideration for this example is 10 days with a discretisation interval of one day. Based on trial and error, a time horizon of 10 days is short enough to yield a detailed solution using the direct approach. In addition, 10 days is long enough to be divided into two reasonable rolling horizon intervals of 5 days each.

Initial and final inventories in ports:

Initial inventories of product i at each port refer to amounts of products existing before the optimisation starts. Final inventories refer to the desired quantities of product i to be present at each port at the end of the last day. In our model, initial and final inventories are assumed to be equal to the safety stock.

Table 6.5: Initial and desired final inventories for the illustrative example

Port	Initial inventory of i (t)	Final inventory of i (t)
p_1	400	400
p_8	400	400
l_2	100	100
l_3	100	100
l_4	100	100
l_9	100	100
l_9	100	100
l_9	100	100

Regardless of the demand data, the illustrative problem mathematical model information is the same. Table 6.6 shows the number of equations, number of binary variables, and the number of continuous variables for the illustrative example. The mathematical data for the illustrative example are generated with the detailed model using the constraints in Section 3.2. The fully relaxed solution is obtained by treating all binary variables in the detailed model as continuous variables.

Table 6.6: Illustrative example mathematical model information

Mathematical model information	
Number of equations	1500
Number of binary variables	960
Number of continuous variables	2480
Fully relaxed solution	1,400,000

6.2 General Heuristic Algorithms

To show the benefits of optimisation, two general heuristics (based on industry rules of thumb) are tailored to solve the illustrative example. Optimisation-based approaches are then compared to heuristic algorithms in terms of objective values.

6.2.1 Nearest port heuristic (*H1*)

H1 is based on starting to replenish demand locations that are closer to production sites. This heuristic is more applicable to problems with static demand. The general procedure for this heuristic is described as follows:

1. Start with the production site with the most available ships.
2. Send a full load of the largest available ship to the nearest demand location. If a full shipload more than covers the entire demand of that location, go to the nearest demand location and so on until the product on that ship is used. Otherwise, go to step 3.
3. Send the ship to nearest production site to reload. Go to step 2.
4. Continue until the demand at all locations is satisfied. At anytime, if the duration of sending the same ship to another location exceeds the time horizon, then send another available ship from the nearest available production site.

This procedure is described further with numeric data in the illustrative example's static demand case in Section 6.3.

6.2.2 Priority heuristic (*H2*)

H2 is based on committing to demand locations with higher priority. Priority can be evaluated in terms of demand time and customer importance. This heuristic is more applicable to problems with dynamic demand. The general procedure for this heuristic is described as follows:

1. Start with the production site with the most available ships.
2. Send a full load of the largest available ship to location with the earliest demand requirement. If a full shipload more than covers the entire demand of

that location, go to the nearest the location with next earliest demand and so on until the product on that ship is used. Otherwise, go to step 3.

3. Send the ship to nearest production site to reload. Go to step 2.
4. Continue until the demand at all locations is satisfied. At anytime, if more than one location has similar demand priorities, then send the available ship to the nearest one.

This procedure is described further with numeric data in the illustrative example's dynamic demand case in Section 6.4.

6.3 Static Demand Data

The example explained above is first solved using a set of static demand. In other words, we assume that demand is needed only at the final day (day 10). For simplicity, we assure that the total demand (in tonnes) for every customer does not exceed a full shipload. Therefore, demand data are generated using a uniform distribution with bounds [200,300]. Table 6.7 below shows the demand requirement for each customer in the problem.

Table 6.7: Static demand data with a uniform distribution [200,300]

	l_2	l_3	l_4	l_5	l_6	l_7
$t_1 \rightarrow t_9$	0	0	0	0	0	0
t_{10}	220	280	260	230	230	220

To apply ***H1***, we need to know which production site has the most available ships. Since both production points have two ships each of the same size, we arbitrarily start with production point p_1. We send S_1 from p_1 with 300 units towards l_2. The product on S_1 covers the entire demand of l_2, so we send S_1 from l_2 to l_3 with 80 units. Because the duration of sending S_1 back to p_1 then to l_3 again will exceed day 10, we send S_3 from p_1 to l_3 with a full load. S_3 discharges 200 units in l_3 and then goes to l_4 carrying the remaining 100 units. Because no journeys are allowed from p_1 to l_4 directly, S_2 is sent from p_8 to l_4. However, S_2 carries only 200 units to l_4 because no allowed route is available from l_4 to another unsatisfied demand location. Then, S_2 returns to p_8 to reload. S_2 is sent with a full load to l_9. 230 units are discharged at l_9 and the rest is carried to l_9. Because the duration of sending S_2 back to p_8 then to l_9 again will exceed day 10, we send S_4 with a full load from p_8 to l_9. S_4 discharges only 160 units at l_9 and the rest is carried to l_9. Since l_9 is the only remaining unsatisfied location, we look for a feasible ship to cover its demand. The best way is to send a ship from p_1 to l_9 with the reaming 80 units. The only ship that can make it within the time horizon is S_1. Therefore, we send S_1 from l_3 (its final location) to p_1 to reload. Then we send S_1 to l_9 carrying the remaining 60 units. Figure 6.2 shows a Gantt chart of the schedule obtained for the static demand case using heuristic ***H1***.

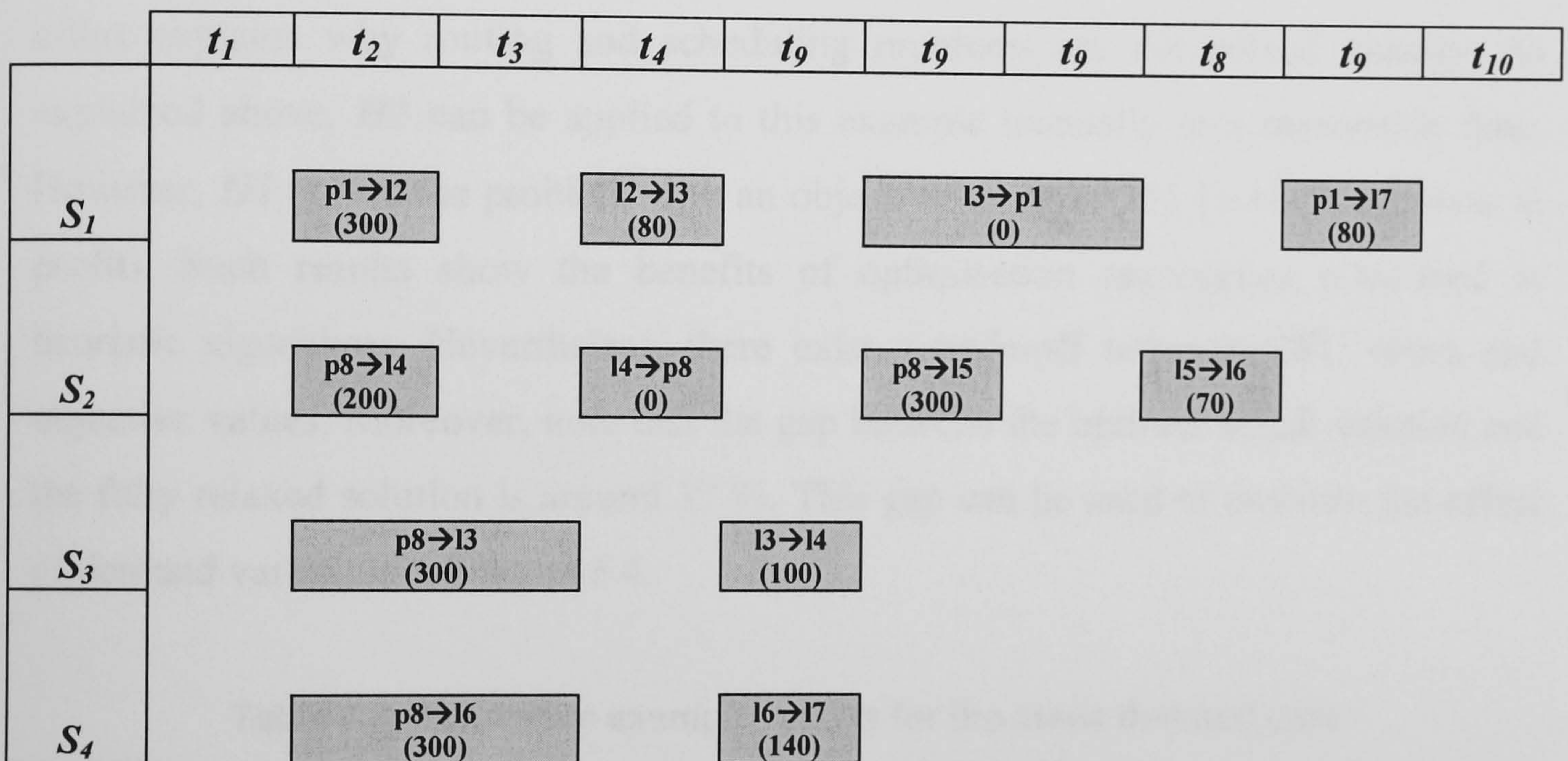

Figure 6.2: Gantt chart of the illustrative example solution with static demand using heuristic *H1*

Table 6.8 below shows the ship-journey matches, the best obtained solutions, and the CPU times used to reach those solutions for the case of static demand. Note that CPU time calculations are different depending on the approach used. In case of the direct approach, the CPU time is the resource usage produced by GAMS. As for the iterative approach, the CPU time is the summation of all resource usages of aggregate and detailed solutions for all iterations used. For the rolling horizon approaches, the CPU time is the summation of all resource usages of all solutions over all time intervals. Note that the number of iterations field is applicable only in case of the iterative approach while the number of intervals is applicable only in case of the rolling horizon approaches. On the other hand, optimality criteria are also different depending on the used approaches. In case of the direct approach, a 10% relative optimality (gap between integer solution and best bound) is specified. As for the iterative approach, a 5% optimality gap between the aggregate and detailed solutions is specified. Finally, a 0% relative optimality is specified for the FRH and HFRH approaches.

All optimisation-based approaches solve the problem to optimality with 100% demand satisfaction for all customer locations. Although the optimal solution is exactly the same based on the ship-journey matches, CPU times differ greatly. Even for a short time horizon of ten days, the direct approach takes approximately 11 hours, which is totally unreasonable for the size of this problem. This high computational effort explains why routing and scheduling problems are not solved exactly. As explained above, ***H1*** can be applied to this example manually in a reasonable time. However, ***H1*** solves the problem with an objective value of 736 ($\approx$16 % decrease in profit). Such results show the benefits of optimisation approaches compared to heuristic algorithms. Nevertheless, there exist a trade-off between CPU times and objective values. Moreover, note that the gap between the optimal MILP solution and the fully relaxed solution is around 37 %. This gap can be used to evaluate the effect of demand variation in Section 6.4.

Table 6.8: Illustrative example results for the static demand case (Fully relaxed solution =1,400,000)

Approach used	*CPU time*	*Best solution*	*Ship-journey matches*
Nearest port heuristic (***H1***)	-	736,000	S_1(j1, j9,j6,j3) S_2(j17,j21,j19,j16) S_3(j2,j12) S_4(j20,j14)
Direct approach	10.5 hours	879,000	S_1(j3, j7,j1) S_2(j19,j23,j20) S_3(j2) S_4(j17)
Iterative approach *(1 iteration used)*	27 seconds	879,000	S_1(j3, j7,j1) S_2(j19,j23,j20) S_3(j2) S_4(j17)
FRH approach $t_1 \rightarrow t_5$ *(interval 1)* $t_6 \rightarrow t_{10}$ *(interval 2)*	133 seconds	879,000	S_1(j3, j7,j1) S_2(j19,j23,j20) S_3(j2) S_4(j17)
HFRH approach $t_1 \rightarrow t_5$ *(interval 1)* $t_6 \rightarrow t_{10}$ *(interval 2)*	20 seconds	879,000	S_1(j3, j7,j1) S_2(j19,j23,j20) S_3(j2) S_4(j17)

On the other hand, the iterative approach reaches the same optimal solution in 27 seconds using only one iteration. Hence, for such problems where demand is not dynamic, the use of aggregation can be of great benefit. Moreover, the FRH and the HFRH both reach the same optimal solution in much shorter times compared to the direct approach. Note that the simplicity of the static demand justifies the use of time aggregation. Problems with more dynamic demand are almost impossible to solve with time aggregation, as will be illustrated later.

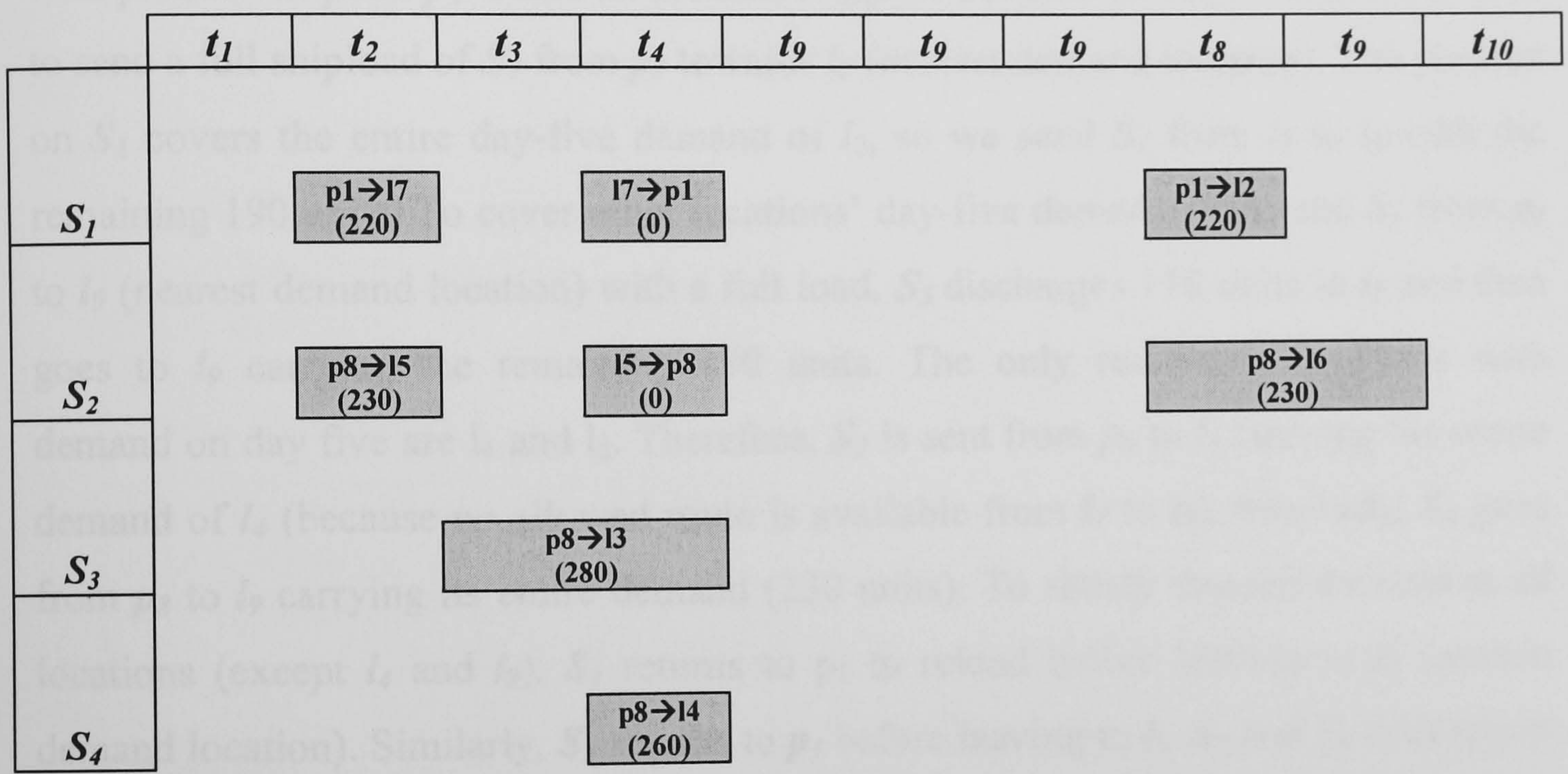

Figure 6.3: Gantt chart of the illustrative example optimal solution with static demand

A Gantt chart of the optimal solution (using optimisation-based approaches) is shown in Figure 6.3. Because the static demand quantity does not exceed a full shipload, every location is visited only once in the optimal schedule. A more realistic demand dataset is solved in the next section.

6.4 Dynamic Demand Data

To study the effect of demand variation with time, we use the same total demand required by each customer. However, we split this demand over the entire time horizon. Therefore, half of the demand is required on the fifth day and half is required on the tenth day. Table 6.9 shows the demand of each customer in the problem.

Table 6.9: dynamic demand data with a uniform distribution [200,300]

	l_2	l_3	l_4	l_5	l_6	l_7
$t_1 \rightarrow t_4$	0	0	0	0	0	0
t_5	110	140	130	115	115	110
$t_6 \rightarrow t_9$	0	0	0	0	0	0
t_{10}	110	140	130	115	115	110

To apply ***H2***, we need to know which production site has the most available ships. Since both production points have two ships each of the same size, we arbitrarily start with production point p_1. Since all locations require demand on the 5th day, we choose to send a full shipload of S_1 from p_1 towards l_2 (nearest demand location). The product on S_1 covers the entire day-five demand of l_2, so we send S_1 from l_2 to l_3 with the remaining 190 units. To cover other locations' day-five demands, we send S_3 from p_1 to l_9 (nearest demand location) with a full load. S_3 discharges 110 units in l_9 and then goes to l_9 carrying the remaining 190 units. The only remaining locations with demand on day five are l_4 and l_5. Therefore, S_2 is sent from p_8 to l_4 carrying the entire demand of l_4 (because no allowed route is available from l_4 to l_9). Similarly, S_4 goes from p_8 to l_9 carrying its entire demand (230 units). To satisfy day-ten demand at all locations (except l_4 and l_9), S_1 returns to p_1 to reload before leaving to l_2 (nearest demand location). Similarly, S_3 returns to p_1 before leaving to l_9. S_2 and S_4 both return to p_8 to replenish locations l_3 and l_9 respectively (see Figure 6.4).

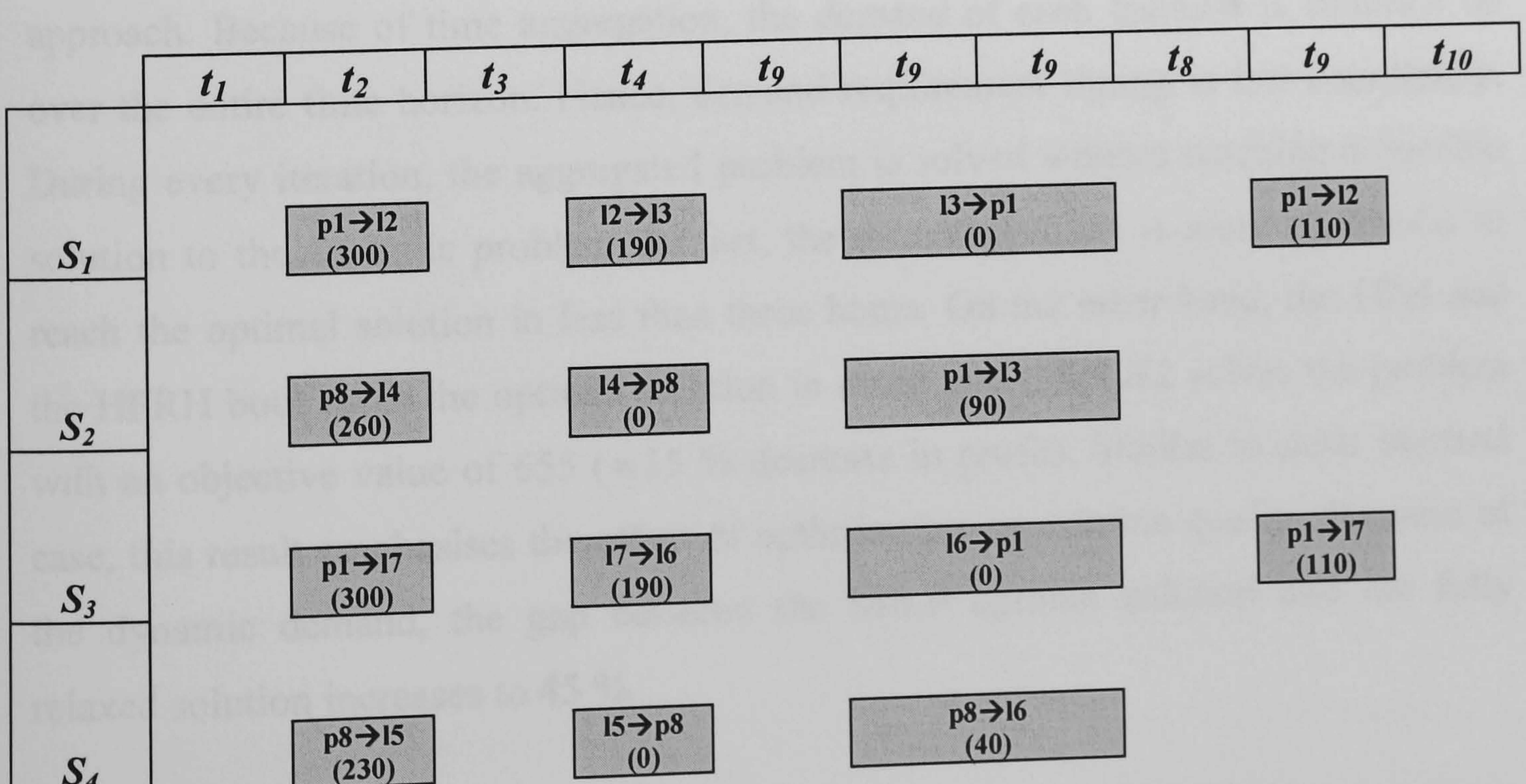

Figure 6.4: Gantt chart of the illustrative example solution with dynamic demand using heuristic *H2*

Table 6.10 shows the ship-journey matches, the best obtained solutions, and the CPU times used to reach those solutions for the dynamic demand case.

Table 6.10: Illustrative example results for the dynamic demand case (Fully relaxed solution =1,400,000)

Approach used	*CPU time*	*Best solution*	*Ship-journey matches*
Priority heuristic (***H2***)	-	655,000	S_1(j1, j9,j6,j1) S_2(j17,j21,j18) S_3(j3,j13,j8,j3) S_4(j19,j23,j20)
Direct approach	2.5 hours	769,000	S_1(j3, j7,j1) S_2(j17,j21,j19) S_3(j1, j9) S_4(j19, j16)
Iterative approach *(1000 iterations used)*	3 days	None	None
FRH approach $t_1 \rightarrow t_5$ *(interval 1)* $t_6 \rightarrow t_{10}$ *(interval 2)*	15 minutes	769,000	S_1(j3, j7,j1) S_2(j17,j21,j19) S_3(j1, j9) S_4(j19, j16)
HFRH approach $t_1 \rightarrow t_5$ *(interval 1)* $t_6 \rightarrow t_{10}$ *(interval 2)*	93 seconds	769,000	S_1(j3, j7,j1) S_2(j17,j21,j19) S_3(j1, j9) S_4(j19, j16)

Note here that the iterative approach runs for over three days without reaching the optimal solution. This is due to the type of time aggregation involved in the iterative approach. Because of time aggregation, the demand of each location is summed up over the entire time horizon. Hence, demand requirement timing is lost completely. During every iteration, the aggregated problem is solved without reaching a feasible solution to the dynamic problem. In fact, the direct approach is more successful to reach the optimal solution in less than three hours. On the other hand, the FRH and the HFRH both reach the optimal solution in much less time. ***H2*** solves the problem with an objective value of 655 ($\approx$15 % decrease in profit). Similar to static demand case, this result emphasises the effect of optimisation on solution quality. Because of the dynamic demand, the gap between the MILP optimal solution and the fully relaxed solution increases to 45 %.

As we mentioned in Section 1.6, VMI problems include production, routing and scheduling, and inventory management. Therefore, VMI problems are very difficult to solve exactly even for a simple problem like our example. In addition, the use of heuristic algorithms can result in great deviations from the optimal solutions. Hence, the use of other optimisation-based approaches is necessary to solve such problems in reasonable CPU times.

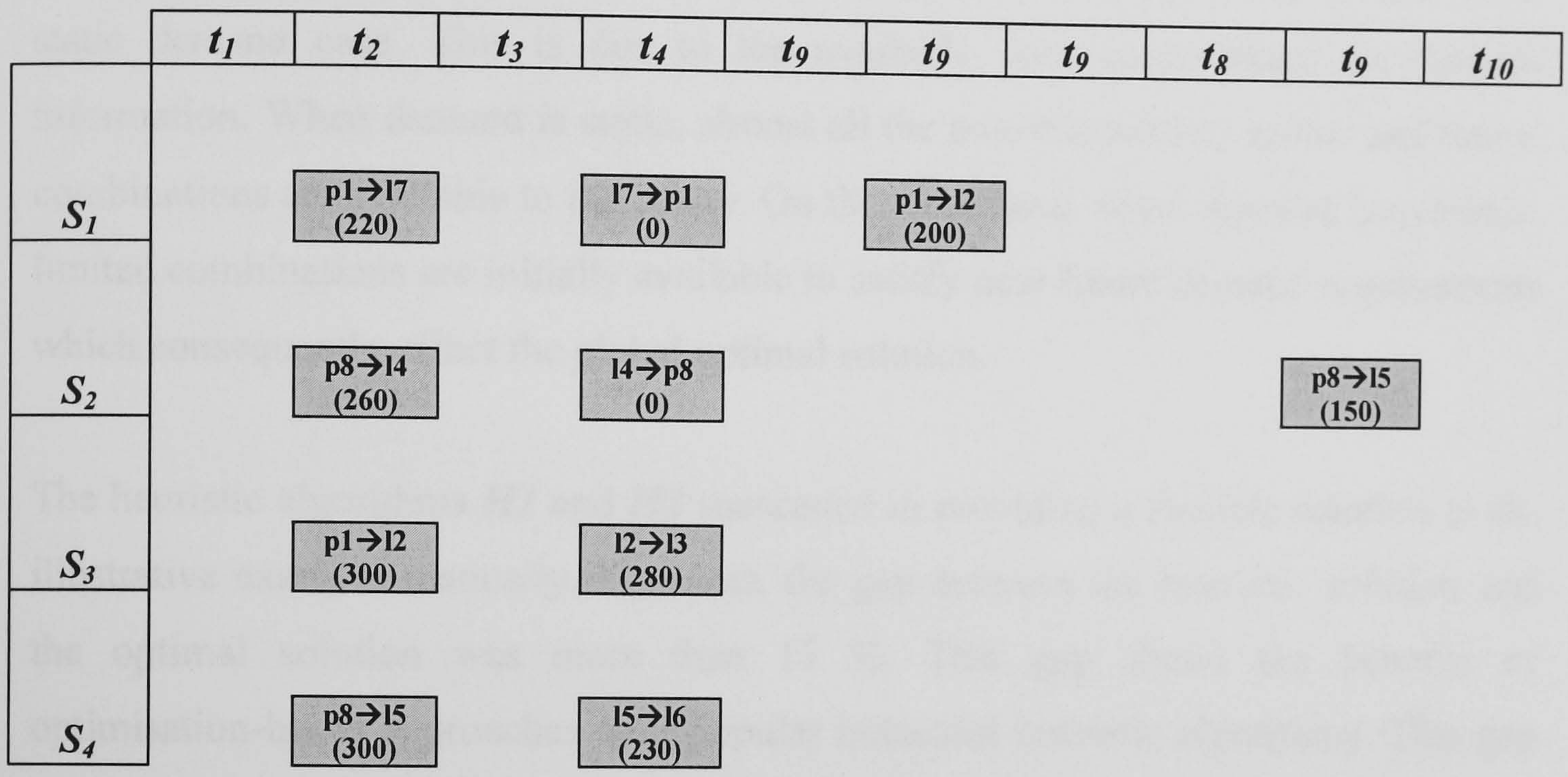

Figure 6.5: Gantt chart of the illustrative example optimal solution with dynamic demand

A Gantt chart of the optimal solution is shown in Figure 6.5. For some locations (l_2 and l_9), initial inventories do not cover the demand required at t_9. Such locations are visited more than once to ensure product availability. As a result, the final optimal solution value in this case is less than that of the static demand case where every location is visited only once.

The illustrative example is only used to compare all the solution approaches in terms of solution quality and CPU time. Because of the nature of the example problem and the short time horizon, no inventory levels are shown for the illustrative example.

6.5 Conclusions

The four proposed optimisation-based approaches are tested using an illustrative example of reasonable size with two case scenarios. First, the example is solved with a set of static (non-dynamic) uniformly distributed demand data. Second, the same example is solved with a set of relatively dynamic uniformly distributed demand data. The direct solution approach takes relatively longer CPU times to solve this simple example. For larger size examples, the direct approach can run indefinitely without reaching a feasible solution. Note that the CPU time for the direct approach in the case of dynamic demand was much shorter that the CPU time consumed in case of the static demand case. This is due to the available state space based on demand information. When demand is static, almost all the possible journey routes and times' combinations are available to the solver. On the other hand, when demand is dynamic, limited combinations are initially available to satisfy near future demand requirements which consequently affect the global optimal solution.

The heuristic algorithms ***H1*** and ***H2*** succeeded in providing a feasible solution to the illustrative example manually. However, the gap between the heuristic solution and the optimal solution was more than 15 %. This gap shows the benefits of optimisation-based approaches over popular industrial heuristic algorithms. This gap can increase substantially based on the problem size.

For the static demand case, the iterative approach led to the optimal solution in a very short time. However, dynamic demand increases the gap between the aggregate and the detailed solutions to make it difficult for the iterative approach to solve the problem. In other words, as the demand becomes more dynamic, it becomes harder for the detailed solution to feasibly fit the aggregately-generated ship-journey pre-matches. Hence, the iterative approach is recommended to solve problems with static demand only.

Both the FRH and the HFRH approaches solve the example with static and dynamic demand data in reasonable CPU times. Both approaches proved to be robust enough regardless of the type of demand data. In addition, the HFRH approach showed a reasonable edge over the FRH approach. While the solution quality is the same, the HFRH approach consumed relatively shorter CPU times than the FRH approach. Considering the type of mathematical modelling used, the HFRH approach is the most robust solution approach for our VMI system.

Chapter 7

Industrial Case Studies

In this chapter the VMI system is tested using real industrial case studies from different worldwide companies. Due to confidentiality, the following case studies are described on a no-name basis. Hence, company names as well as any other sensitive information are not disclosed. Only symbols are used to represent various details of these case studies. However, the context of the VMI problem is not distorted in any way during these tests.

7.1 Industrial Case 1: LNG shipping

Liquefied Natural Gas (LNG) is an important source of energy. Mainly, LNG is used as a raw material to produce electricity. A pioneer LNG producer company in the Arabian Gulf region (***A***) has been associated with a giant LNG consumer company in the Far East (***B***). A 20-year agreement between the two companies was signed in the late nineties. Based on this agreement, ***A*** will supply ***B*** with four million tonnes of LNG annually. This quantity will be delivered using the supplier-owned ships. Although ***B*** is ***A***'s main customer, ***B*** is not the only customer at the moment. Other European customers occasionally buy LNG from ***A*** using their own ships. Hence, ***A*** has only one VMI customer (***B***) and many external non-VMI customers.

Currently, the company is conducting a feasibility study to investigate the possibility of adding another VMI customer in the Far East (***C***). They would like to know if the company's existing plant capacity and shipping capabilities are sufficient. In addition, they want to assess the economic benefits of such a move. Because of the current scheduling technique and the lack of coordination between shipping and production, ***A*** is barely able to fulfil its commitments to its main customer. So, the possibility of adding another VMI customer seemed unrealistic at the beginning. However, the company's management is undertaking major changes to its current scheduling techniques in order to increase their potential profit. We were privileged to take part in this study as it represents a great opportunity to test the proposed CDST.

7.1.1 Problem specifications

The first step is to solve the original VMI problem with only one main customer and occasional non-VMI customers. This step can be used to benchmark the new move. In other words, by calculating the actual profit at the moment, we can assess the benefits of adding another VMI customer. The second step is to solve the problem after adding *C* as another VMI customer. The original problem specifications are given below:

<u>Available ports:</u>

As mentioned above, there is only one production plant and one customer. Table 7.1 shows the storage capacities and number of ships for both ports. Note that the storage capacity of ***B*** is much greater compared to that of *A*. For this case, the capacity of ***B*** is not an effective factor in the optimisation.

Table 7.1: Port information for industrial case 1

Ports	Total storage capacity (t)	Maximum allowed number of ships
A	100,000	5
B	5,000,000	4

<u>Available ships:</u>

To fully optimise the VMI problem, the correct specification of the initial locations of all ships is essential. Therefore, some ships might be on their way to some port as opposed to actually being there (see Table 7.2).

Table 7.2: Ship information for industrial case 1

Ships	Total storage capacity (t)	Initial port	Time needed to be there (days)
*S*1	63,000	at ***B***	0
*S*2	63,000	on its way to ***B***	4
*S*3	63,000	on its way to ***B***	8
*S*4	63,000	on its way to ***B***	12
*S*5	63,000	at ***A***	0
*S*6	63,000	on its way to ***A***	4
*S*7	63,000	on its way to ***A***	8
*S*8	63,000	on its way to ***A***	12

Products information:

Table 7.3 shows the production rate, cost, and price of LNG. Note that the selling price shown in the table is particular to ***B*** only.

Table 7.3: Product information for industrial case 1

Product name	Price ($/t)	Production cost ($/t) at *A*	Production rate (t/day) at *A*
LNG	255	107	17,000

Journey information:

All costs and durations of journeys are the same for all available ships in the fleet. A round trip journey from *A* to ***B*** costs $3.4 million. A one way (inbound or outbound) journey between *A* and ***B*** takes 16 days on all ships.

Loading times:

A ship takes one day to be charged at *A* and one day to be discharged at ***B***.

Time horizon:

The scheduling period under consideration is one year (366 days) with a discretisation interval of one day.

Demand information:

A is committed to supplying ***B*** with four million tonnes of LNG during the year of interest. To fully represent the annual contract, we mathematically assume that all this demand is only needed at the end of the year (day 366). In reality, the annual demand is scattered over the entire year. However, no particular pattern can be suggested since many factors can affect the optimal schedule. For example, the total quantity delivered in a given month can be much more/less than the previous month due to plant shut-downs or ship maintenance. Moreover, this assumption does not affect the optimisation process because satisfying the total demand requires that the entire quantity is actually scattered over the entire year.

Initial inventories in ports:

The initial inventory at *A* is 70,000 tonnes. We assume that ***B*** is initially empty.

Initial inventories on ships:

As Table 7.4 shows, ships S_1-S_4 are on their way to or at B initially with full loads onboard. Ships S_5-S_8 are on their way to or at A with no material onboard.

Table 7.4: Initial ship inventories for industrial case 1

Ship	Initial Inventory
S_1	63,000
S_2	63,000
S_3	63,000
S_4	63,000
S_5	0
S_6	0
S_7	0
S_8	0

Plant shut-downs:

Plant A contains three production trains with total production capacity of 17,000 t/day. Train 1 has a production capacity of 9,000 t/day, while train 2 and train 3 together have a production capacity of 8,000 t/day. Each production train undergoes planned maintenance shut-downs every other year for exactly 35 days. In a particular year, train 2 and train 3 are shut-down. The next year, only train1 is shut down. Therefore, during shut down periods the total production rate decreases to 9,000 or 8,000 t/day depending on the year. For the one-year horizon in this problem we assume that train 1 is shut down and the total production rate reduces to 8,000 t/day for the 35-day shut-down period (see Table 7.5).

Table 7.5: Shut-down information for industrial case 1

Plant	Shut-down starting time	Shut-down ending time	Production rate during shut-down (t/day)
A	15/3	20/4	8,000

Ship maintenance:

Every year two ships undergo regular maintenance. Usually this maintenance period overlaps with the shut down period. To assume the worst case scenario, we will force maintenance to coincide with the shut-down period for S_1 and S_2. Table 7.6 shows the starting and ending time of maintenance for ships S_1 and S_2.

Table 7.6: Ship maintenance information for industrial case 1

Ships	Port at which maintenance is required	Starting time	Ending time
S_1	*A*	15/3	20/4
S_2	*A*	15/3	20/4

External customers demand:

Occasionally some European customers are sold LNG. Those customers use their own ships to collect it. External customer demand (in terms of quantity and shipping dates) is usually known in advance. For the sake of optimising the VMI system to the full extent, external customers are given priority for any supply. Therefore, all the required data from external customers will be the shipment date, the quantity of the shipment (*Q*), and the number of ships used (*NS*). Because these customers ship their products themselves, the selling price is slightly reduced. Table 7.7 shows the external demand information for this case study.

Table 7.7: External demand information for industrial case 1

Date→		23/1		2/4		7/7		8/9		11/10	
Product	Price	*NS*	*Q*	*NS*	*Q*	*NS*	*Q*	*NS*	*Q*	*NS*	*Q*
LNG	$235	1	57,000	1	57,000	1	50,000	1	50,000	1	57,000

7.1.2 Problem solution

As shown in Chapter 6, the HFRH proved to be the most robust solution approach. Other approaches failed to produce results for this case study. In order to compare the optimal solution with the heuristic one, we solve the same problem using heuristic ***H2*** (see Section 6.4).

For this case study, we divide the time horizon into five intervals. We arbitrarily choose a length of 75 days for the first four intervals, the remaining 66 days is the length of the fifth and final interval. The mathematical information for the original problem of this case study is shown in table 7.8.

Table 7.8: Original problem mathematical information for industrial case 1

Number of equations	35,371
Number of binary variables	5,760
Number of continuous variables	31,975
Rolling horizon time intervals	Five intervals (t1→t75→t150→t225→t300→t366)

The results of solving the original problem using ***H2*** and the HFRH approach are shown in Table 7.9. Because ***H2*** is applied manually, no CPU time is shown for this algorithm. The relative optimality (gap between integer solution and best node in GAMS) is specified at 1% for the HFRH approach.

Table 7.9: Original problem results for industrial case 1
(Fully relaxed solution = $ 694 million)

	Priority heuristic (***H2***)	HFRH approach
Profit	$ 513 million	$ 513 million
Total material delivered to ***B***	4,032,000 t (64 full shiploads)	4,032,000 t (64 full shiploads)
Average daily production rate at ***A***	10,920 t/day	10,920 t/day
Ship utilisation percentage	73%	73%
Solution CPU time	-	30 minutes
Relative optimality	-	1%

Because we only have one customer, the HFRH and ***H2*** reach the same objective value. However, applying ***H2*** manually requires a lot of effort because of the need of production plans co-ordination. In addition, ship maintenance, external demand and plant shut downs for this case study can disturb any systematic method to solve the problem manually. In general, the same number of full shiploads is required which leads to the same final solution. Because having only one customer restricts the space of journey decisions, the gap between the best solution and the fully relaxed one is 25%.

Note that the original problem is optimised with an average daily production of 10,920 tonnes including shut-downs. The ship utilisation (busy days/idle days excluding maintenance) is 73%. Certainly, there is enough room to push production and shipping to their limits and study the possibility of adding another customer.

7.1.3 Potential modifications

In addition to the problem details explained in Section 7.1.1, another customer ***C*** is added to the problem. As a result, a new journey route is added between *A* and *C*. According to the company, the round trip journey between *A* and ***C*** costs $ 2.8 million. All ships take 14 days for a one way journey between *A* and *C*. The unloading time at ***C*** is one day. To represent the commitment to the contract, we decided to treat ***B***'s demand as hard constraints. However, we do not introduce any demand for ***C*** as the optimisation should find the best possible and profitable quantity to be delivered to ***C***. The mathematical information of the modified problem is shown in table 7.10.

Table 7.10: Modified problem mathematical information for industrial case 1

Number of equations	45,573
Number of binary variables	11,520
Number of continuous variables	41,974
Rolling horizon time intervals	Five intervals (t1→t75→t150→t225→t300→t366)

The results of solving the modified problem using ***H2*** and the HFRH are shown in Table 7.11. Because it ***H2*** is applied manually, no CPU time is shown for this algorithm. Due to the increased complexity of the modified problem, the relative optimality specification is increased to 5% for the HFRH approach.

Table 7.11: Modified problem results for industrial case 1
(Fully relaxed solution =$ 769 million)

	Priority heuristic (***H2***)	HFRH approach
Profit	$ 570 million	$ 597 million
Total material delivered to ***B***	4,032,000 t (64 full shiploads)	4,032,000 t (64 full shiploads)
Total material delivered to ***C***	567,000 t (9 full shiploads)	819,000 t (13 full shiploads)
Average daily production rate at *A*	12,460 t/day	13,150 t/day
Ship utilisation percentage	84%	88%
Solution CPU time	-	100 minutes
Relative optimality	-	5%

Because we have two customers in the modified problem, the HFRH and ***H2*** final solutions differ slightly. Consequently, the optimal solution is better than the heuristic one. Recall from Chapter 6 that ***H2*** is a priority heuristic. Since ***B*** is the main customer with a VMI contract, ***B*** is given priority over ***C***. Hence, using ***H2***, we try to satisfy the demand of ***B*** entirely before we deliver any material to ***C***. As a result, we can build on the original problem's schedule for this heuristic.

Due to adding ***C***, the number of equations and continuous variables increased considerably. Since adding another customer increases the space of available options, the number of binary variables has doubled. However, the optimisation process pushed production and shipping to their limits. The average daily production increased by 12% while ship utilisation increased to 88%. As a result, the potential profit increased to \$597 million ($\approx$16% increases in profit). Due to the extra number of binary and continuous variables, the CPU time was tripled. Taking into account the increased problem size, the CPU time of 100 minutes is still considered reasonable. It is worth mentioning that the gap between the best solution and the fully relaxed one remains at 25% in spite of the added complexity. This shows that the complexity (although increased in terms of binary variables) is still ineffective due to the exclusion of journeys between customers ***B*** and ***C***.

Figures 7.1 and 7.2 are Gantt charts of the modified problem solutions using ***H2*** and HFRH, respectively. As shown in Figure 7.1, all ships are initially serving only customer ***B***. After the entire demand of ***B*** is satisfied, attention is focused entirely to deliver material to ***C***. Such results emphasise the limitations of heuristic solutions since delivering product to customer ***C*** only at the end of the year is not realistic. On the other hand, figure 7.2 shows that material is delivered to both customers with no specified order. In other words, deliveries to both customers are relatively scattered over the entire year. This shows the versatility of optimisation-based approaches not only in the objective value, but also in the physical quality of the solution.

7.1.4 Recommendations

Based on the results of the previous section, adding another customer in the Far East is feasible even with existing production and shipping capabilities. A new contract to supply customer ***C*** with 819,000 tonnes of LNG annually can be very profitable to company ***A***. A potential 16% increase in profit can be attained annually while no new investment is needed. The existing production capacity will be pushed to nearly its upper bound while available ships will be busier during the year. These results were presented to ***A***'s management to support the move of adding another long-term VMI customer in the Far East. We still await the company's response as to how beneficial this study was in aiding the decision making policy.

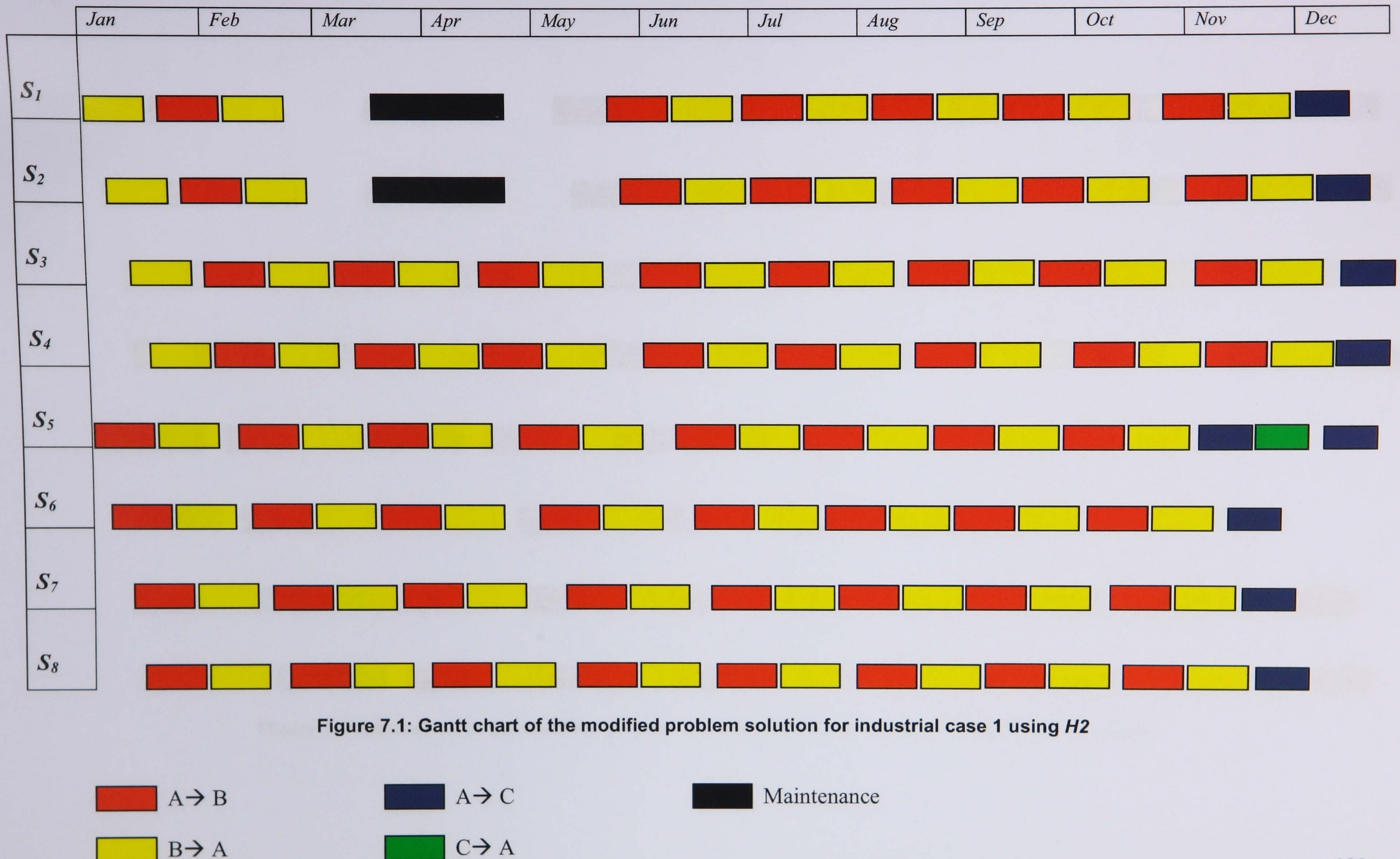

Figure 7.1: Gantt chart of the modified problem solution for industrial case 1 using *H2*

Figure 7.2: Gantt chart of the modified problem solution for industrial case 1 using HFRH approach

7.2 Industrial Case 2: Oil products shipping

An international oil company deals with a problem of optimising a VMI system between a refinery (***P***) and multiple terminals. Six oil products must be shipped to satisfy the demand of the geographically-dispersed terminals. Only three flexible-compartment ships are used to deliver the products to the terminals. However, the vendor's current production and shipping capabilities cannot satisfy the total demand of all these terminals. As a result, the vendor occasionally requests those products to be produced and delivered by third parties at different prices. This does not violate the VMI contract as the vendor is still responsible for delivering the products to those customers. Nevertheless, it is more costly to the vendor to go to third parties.

Recently, the company have introduced new ships/terminals into this system and the only available solution is based on heuristic algorithms. In addition, due to confidentiality reasons, no production costs and product prices are available to us. Hence, the objective becomes minimising the total transportation cost instead of maximising the total profit. Transportation costs include third party shipping as well as vendor-owned shipping. The problem for us was to find the optimal ship schedule that minimises the total quantity ordered from third parties. In addition, a log of all quantities of all products to be ordered from third parties is produced with the optimal solution. The VMI problem specifications are given in the next subsection.

7.2.1 Problem specifications

A layout of the refinery, the terminals, and available journey routes is shown in the Figure 7.3. Two sets of terminals are available; primary terminals (***A***, ***B***, & ***C***) and secondary terminals (***D***, ***E***, ***F***, ***G***, & ***H***). Double arrows on the layout represent the possibility of inbound and outbound journeys. Because of profitability issues and depth restrictions, the demand of the secondary terminals must be satisfied entirely by ***P***. On the other hand, extra material can be ordered from third parties to satisfy the demand of the primary terminals only.

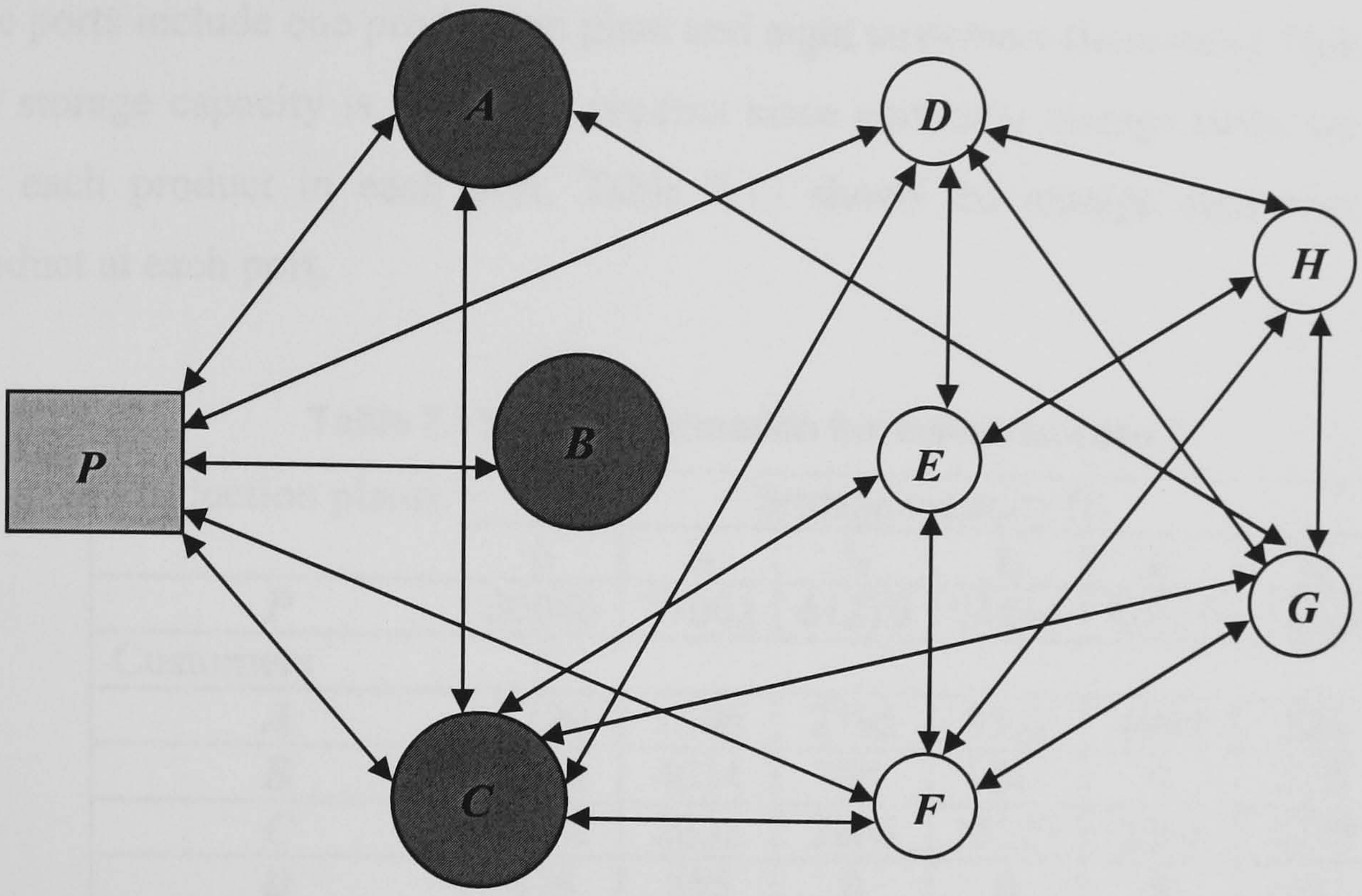

Figure 7.3: Port layout for industrial case 2

Time horizon:

The scheduling period under consideration is one month (30 days). Because the shortest event (journey or loading) takes place in a half day, the discretisation interval is chosen to be a half day (12 hours). Therefore, the total number of discrete time periods is equal to sixty.

Products information:

Six oil products are available. As Table 7.12 shows, no production costs or product prices are given. Only the production rates at ***P*** of all products are provided.

Table 7.12: Product information for industrial case 2

Product name	Production rate (t/day) at ***P***
i_1	5232
i_2	10529
i_3	2957
i_4	2460
i_9	2906
i_9	140

Available ports:

The ports include one production plant and eight customers (terminals). Note here that the storage capacity is given per product since particular storage tanks are specified for each product in each port. Table 7.13 shows the storage capacities for each product at each port.

Table 7.13: Port information for industrial case 2

Production plants	Storage capacity (t)					
	i_1	i_2	i_3	i_4	i_9	i_9
P	36046	77002	61270	26844	42019	12629
Customers						
A	2220	2286	2785	5532	4942	224
B	4358	4014	995	1241	0	510
C	4198	2635	2466	15327	2504	1379
D	476	455	0	0	0	0
E	219	258	532	0	0	0
F	279	339	320	0	0	0
G	1025	865	1220	0	0	0
H	0	292	560	0	0	0

Available ships:

Three flexible compartment ships are available for transport. Due to the number of compartments and mixing specifications, Ships S_1 and S_2 can carry products i_1, i_2, and i_3 only. Ship S_3 can carry products i_4, i_9, and i_9 only. Hence, any ship can take any combination of the three specified products as long as the total storage capacity is not exceeded. All ships originate from the production plant ***P***. S_1 has the smallest storage capacity while Ships S_2 and S_3 have the same total storage capacity (see Table 7.14).

Table 7.14: Ship information for industrial case 2

Ships	Total storage capacity (t)	Initial port
S_1	2400	at ***P***
S_2	4250	at ***P***
S_3	4250	at ***P***

Journey information:

As the layout in Figure 7.3 shows, the journey network is complex. This network was constructed by the vendor based on the geographical position of each port. As Table

7.15 shows, journey costs and durations are equal for ships S_2 and S_3 while S_1 differs because of its smaller size.

Table 7.15: Journey information for industrial case 2

Journey no.	From	To	S_1		S_2		S_3	
			$	½ day	$	½ day	$	½ day
1	**P**	**A**	16429	2	20047	2	20047	2
2	**P**	**B**	15586	1	20192	1	20192	1
3	**P**	**C**	11999	1	15330	1	15330	1
4	**P**	**D**	20919	3	25100	3	25100	3
5	**P**	**F**	24567	4	29267	4	29267	4
6	**A**	**P**	16429	2	20047	2	20047	2
7	**B**	**P**	15586	1	20192	1	20192	1
8	**C**	**P**	11999	1	15330	1	15330	1
9	**D**	**P**	20919	3	25100	3	25100	3
10	**F**	**P**	24567	4	29267	4	29267	4
11	**A**	**C**	6796	1	8288	1	8288	1
12	**A**	**G**	11353	2	13513	2	13513	2
13	**C**	**A**	6796	1	8288	1	8288	1
14	**C**	**D**	11285	2	13342	2	13342	2
15	**C**	**E**	9115	2	10450	2	10450	2
16	**C**	**F**	19490	4	22733	4	22733	4
17	**C**	**G**	10998	2	12768	2	12768	2
18	**D**	**C**	11285	2	13342	2	13342	2
19	**D**	**E**	4558	1	5225	1	5225	1
20	**D**	**G**	10998	2	12768	2	12768	2
21	**D**	**H**	6321	1	7733	1	7733	1
22	**E**	**C**	9115	2	10450	2	10450	2
23	**E**	**D**	4558	1	5225	1	5225	1
24	**E**	**F**	10375	2	12283	2	12283	2
25	**E**	**H**	10878	2	12958	2	12958	2
26	**F**	**C**	19490	4	22733	4	22733	4
27	**F**	**E**	10375	2	12283	2	12283	2
28	**F**	**G**	14933	3	17508	3	17508	3
29	**F**	**H**	14933	3	17508	3	17508	3
30	**G**	**A**	11353	2	13513	2	13513	2
31	**G**	**C**	10998	2	12768	2	12768	2
32	**G**	**D**	10998	2	12768	2	12768	2
33	**G**	**F**	14933	3	17508	3	17508	3
34	**G**	**H**	6321	1	7733	1	7733	1
35	**H**	**D**	6321	1	7733	1	7733	1
36	**H**	**E**	10878	2	12958	2	12958	2
37	**H**	**F**	14933	3	17508	3	17508	3
38	**H**	**G**	6321	1	7733	1	7733	1

Loading times:

Based on the pumping rate for each port, we can calculate the maximum charge/discharge quantity per half day. Table 7.16 shows the maximum loading/unloading rate per half day for every port.

Table 7.16: Loading information for industrial case 2

Plant	Maximum loading quantity (tonnes per half day)
P	4800
Customers	Maximum unloading quantity (tonnes per half day)
A	2400
B	2640
C	3240
D	1200
E	1680
F	600
G	2400
H	2400

Initial inventories in ports:

The refinery ***P*** and the primary terminals have initial inventories of products while secondary terminals do not have any initial inventories.

Table 7.17: Initial port inventories for industrial case 2

Production plants	Initial inventories (t)					
	i_1	i_2	i_3	i_4	i_9	i_9
P	21200	16300	27600	13700	26300	7000
Customers						
A	1730	2190	2310	4040	2370	110
B	1560	2870	0	410	0	310
C	2130	1410	1140	760	1180	560
D	0	0	0	0	0	0
E	0	0	0	0	0	0
F	0	0	0	0	0	0
G	0	0	0	0	0	0
H	0	0	0	0	0	0

Demand information:

The demand over the time horizon of one month is given per product for every terminal. The demand is daily for the primary terminals and monthly for the secondary terminals. Since the discretisation interval is half a day, the even time periods represent full days. For readability, we present the demand data in three separate tables. Table 7.18 shows the daily demand for terminals' ***A***, ***B*** and ***C*** of products i_1, i_2, and i_3 only.

Table 7.18: Primary terminals demand of products (i_1, i_2, & i_3) for industrial case 2

Time	*A*			*B*			*C*		
	i_1	i_2	i_3	i_1	i_2	i_3	i_1	i_2	i_3
t_2	180	180	140	320	310	0	210	210	90
t_4	180	180	140	320	310	0	210	210	90
t_9	180	180	140	320	310	0	210	210	90
t_8	180	180	140	320	310	0	210	210	90
t_{10}	180	180	140	320	310	0	210	210	90
t_{12}	120	130	65	180	110	0	120	150	0
t_{14}	120	130	65	180	110	0	120	150	0
t_{19}	180	180	140	320	310	0	210	210	90
t_{18}	180	180	140	320	310	0	210	210	90
t_{20}	180	180	140	320	310	0	210	210	90
t_{22}	180	180	140	320	310	0	210	210	90
t_{24}	180	180	140	320	310	0	210	210	90
t_{29}	120	130	65	180	110	0	120	150	0
t_{28}	120	130	65	180	110	0	120	150	0
t_{30}	180	180	140	320	310	0	210	210	90
t_{32}	180	180	140	320	310	0	210	210	90
t_{34}	180	180	140	320	310	0	210	210	90
t_{39}	180	180	140	320	310	0	210	210	90
t_{38}	180	180	140	320	310	0	210	210	90
t_{40}	130	130	65	180	110	0	120	150	0
t_{42}	130	130	65	180	110	0	120	150	0
t_{44}	180	180	140	320	310	0	210	210	90
t_{49}	180	180	140	320	310	0	210	210	90
t_{48}	180	180	140	320	310	0	210	210	90
t_{90}	180	180	140	320	310	0	210	210	90
t_{92}	180	180	140	320	310	0	210	210	90
t_{94}	130	130	65	200	110	0	140	150	0
t_{99}	130	130	65	200	110	0	140	150	0
t_{98}	180	180	140	320	310	0	210	210	90
t_{90}	180	180	140	320	310	0	210	210	90

Table 7.19 shows the demand for terminals ***A***, ***B*** and ***C*** of products i_4, i_9, and i_9 only. Note that the primary terminals' demand for those products is given every five days.

Table 7.19: Primary terminals demand of products (i_4, i_5, & i_6) for industrial case 2

Time	*A*			*B*			*C*		
	i_4	i_9	i_9	i_4	i_9	i_9	i_4	i_9	i_9
t_{10}	865	545	0	0	0	0	1080	155	0
t_{20}	865	545	0	0	0	140	1080	155	0
t_{30}	865	545	0	0	0	0	1080	155	100
t_{40}	865	545	0	0	0	140	1080	155	0
t_{90}	865	545	60	0	0	0	1080	155	0
t_{90}	865	545	60	0	0	140	1080	155	100

Table 7.20 shows the monthly demand for the secondary terminals' of products i_1, i_2, and i_3 only. Note that the secondary terminals' demand of products i_4, i_9, and i_9 is equal to zero.

Table 7.20: Secondary terminals demand of products (i_1, i_2, & i_3) for industrial case 2

Time	*D*			*E*			*F*			*G*			*H*		
	i_1	i_2	i_3	i_1	i_2	i_3	i_1	i_2	i_3	i_1	i_2	i_3	i_1	i_2	i_3
t_{90}	390	360	0	110	150	220	60	70	160	380	580	60	0	220	60

External demand:

Beside the customers shown above, the vendor supplies other VMI customers using other means than shipping. Such means include trains, pipelines, and trucks. We include the total material taken from the same refinery (*P*) using other means of transportations. Table 7.21 shows the total external quantities of each product needed every five days.

Table 7.21: External demand information for industrial case 2

Time / Product	t_{10}	t_{20}	t_{30}	t_{40}	t_{90}	t_{90}
i_1	20,000	20,000	30,000	20,000	20,000	15,000
i_2	30,000	50,000	50,000	50,000	40,000	30,000
i_3	10,000	10,000	15,000	15,000	15,000	10,000
i_4	9,000	9,000	9,000	9,000	9,000	9,000
i_9	10,000	15,000	15,000	15,000	10,000	10,000
i_9	600	600	500	500	500	500

Port restrictions:

Because of depth restrictions, the larger ships (S_2 and S_3) cannot enter the secondary terminals' ports. As Table 7.22 shows, a value of 1 means that a ship can enter a port while a value of 0 means a ship cannot enter a port.

Table 7.22: Port restrictions for industrial case 2

Ship	Port								
	P	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*
S_1	1	1	1	1	1	1	1	1	1
S_2	1	1	1	1	0	0	0	0	0
S_3	1	1	1	1	0	0	0	0	0

7.2.2 Problem solution

As shown in Chapter 6, the HFRH proved to be the most robust solution approach. Other approaches failed to produce results for this case study. In order to compare the optimal solution with the heuristic one, we solve the same problem using heuristic ***H2*** (see Section 6.4).

For this case study, we divide the time horizon into six intervals of ten periods (five days) each. The mathematical information for the original problem of this case study is shown in table 7.23.

Table 7.23: Original problem mathematical information for industrial case 2

Number of equations	17,392
Number of binary variables	4,080
Number of continuous variables	41,719
Rolling horizon time intervals	Six intervals (t1→t10→t20→t30→t40→t50→t60)

Given the complex network of journeys above, the number of binary variables is around 4000 due to the port restrictions. The results of solving the original problem using ***H2*** and the HFRH approach are shown in Table 7.24. Because ***H2*** is applied manually, no CPU time is shown for this algorithm. The relative optimality (gap between integer solution and best node in GAMS) is specified at 5% for the HFRH approach.

Table 7.24: Original problem results for industrial case 2 (Fully relaxed solution = 6,300)

	Priority heuristic (***H2***)	HFRH approach
Transportation cost	\$ 760,000	\$ 721,000
Ship utilisation percentage	80%	79%
Solution CPU time	-	22 minutes
Relative optimality	-	5%

The manual scheduling system which the company currently uses is based on heuristics and decomposition. Since only S_1 is allowed into secondary terminals, no other ship is assigned any journeys to secondary terminals. Therefore, ***H2*** is applied for S_1 with the secondary terminals only. In addition, ***H2*** is applied for S_2 and S_3 with the primary terminals only. In other words, S_1 does not deliver any material to the primary terminals. Figure 7.4 shows a Gantt chart of the original problem solution using ***H2***. The heuristic solution is within 5% of the optimal solution. This result explains the company's reliance on heuristic algorithms for this complex problem.

On the other hand, the HFRH approach obtains a solution that is better in quality compared to the heuristic one. This improvement is due to letting S_1 visit primary terminals as well as secondary terminals. Figure 7.5 shows a Gantt chart of the original problem solution using HFRH. The main difference between figures 7.4 and 7.5 is that ***S1*** is allowed to visit primary terminals in the second chart. An interesting phenomenon is the multiple visits of ships to adjacent ports in a short time periods. This is due to demand variability and capacity constraints in case of the primary terminals. In other words, a ship visits a port, moves to the next port, then comes back to visit the same previous port without loading in between.

Both the HFRH ship utilisation percentage and total CPU time seem reasonable for this problem. The transportation costs include third party shipping which is equal to the vendor's shipping costs multiplied by two. Doubling the third party shipping costs is a reasonable assumption since the vendor is trying to minimise orders from third parties. Note that there is a substantial gap between the optimal HFRH solution and the fully relaxed one due to the huge binary decisions involved.

Figure 7.4: Gantt chart of the original problem solution for industrial case 2 using *H2*

Journey by S_1 Journey by S_2 Journey by S_3

Figure 7.5: Gantt chart of the original problem solution for industrial case 2 using HFRH

Journey by S_1 Journey by S_2 Journey by S_3

Taking into account material ordered from third parties, the demand of all terminals was fully satisfied. Table 7.25 shows a log of quantities of each product ordered from third parties for primary terminals A and B. No material is needed for primary terminal C. Note that the minimum order quantity from third parties is equal to the capacity of the smallest ship (2400 tonnes).

Table 7.25: Original problem optimal third party quantities for industrial case 2

Day required	Quantities (t) ordered from third parties to be delivered to ***A***					
	i_1	i_2	i_3	i_4	i_9	i_9
10	2170	490	0	0	0	0
Day required	Quantities (t) ordered from third parties to be delivered to ***B***					
	i_1	i_2	i_3	i_4	i_9	i_9
28	522	1776	0	0	0	420

Third party quantities are generated by the model per discrete period (for every half day). However, we forced the model to generate those quantities at even time periods which represent full days in reality. In the case of the original problem, two full shiploads are needed from third parties. One shipload of products i_1 and i_2 is required for terminal A. Another shipload of products i_1, i_2, and i_9 is required for terminal B.

As previously mentioned, the existing production capacity of refinery P cannot satisfy the demand of all terminals for some products. As a result, the production rates for such products are operating around their upper bounds to minimise the total quantity ordered from third parties (see Table 7.26).

Table 7.26: Original problem optimal production rates for industrial case 2

Product name	Average production rate (t/day) at ***P***	Maximum production rate (t/day) at ***P***	Throughput
i_1	4748	5232	91 %
i_2	8921	10529	85 %
i_3	2703	2957	91 %
i_4	2186	2460	89 %
i_9	2640	2906	91 %
i_9	117	140	84 %

Figure 7.6 shows the inventory levels of each product at the primary terminals only. No secondary terminal inventories are shown since the demand is static for those terminals. While the initial inventory of each product is specified by the user, we force the model to reserve an equal amount of inventory at the end of the time horizon. As seen in Figure 7.6, the inventory levels are roughly the same for most products during the entire month. This shows the efficiency of the VMI system that keeps adequate levels of products during the entire planning horizon.

7.2.3 Potential modifications

Utilising the current production and shipping capabilities is a first priority to the vendor. The objective minimises the cost of transportation between ports. Due to the higher production and transportation costs, the optimisation process reduces the total quantities ordered from third parties. However, we can further explore potential investments that might lead to better solutions. These modifications are evaluated in terms of operational (transportation) costs only. No investment costs are considered in these evaluations. We solve the problem after each modification to evaluate the potential profit. We only use the HFRH approach to solve the modified problems and compare the results to the original problem's solution. Due to the increased complexity of the modified problems, the relative optimality specification is increased to 10% for the HFRH approach.

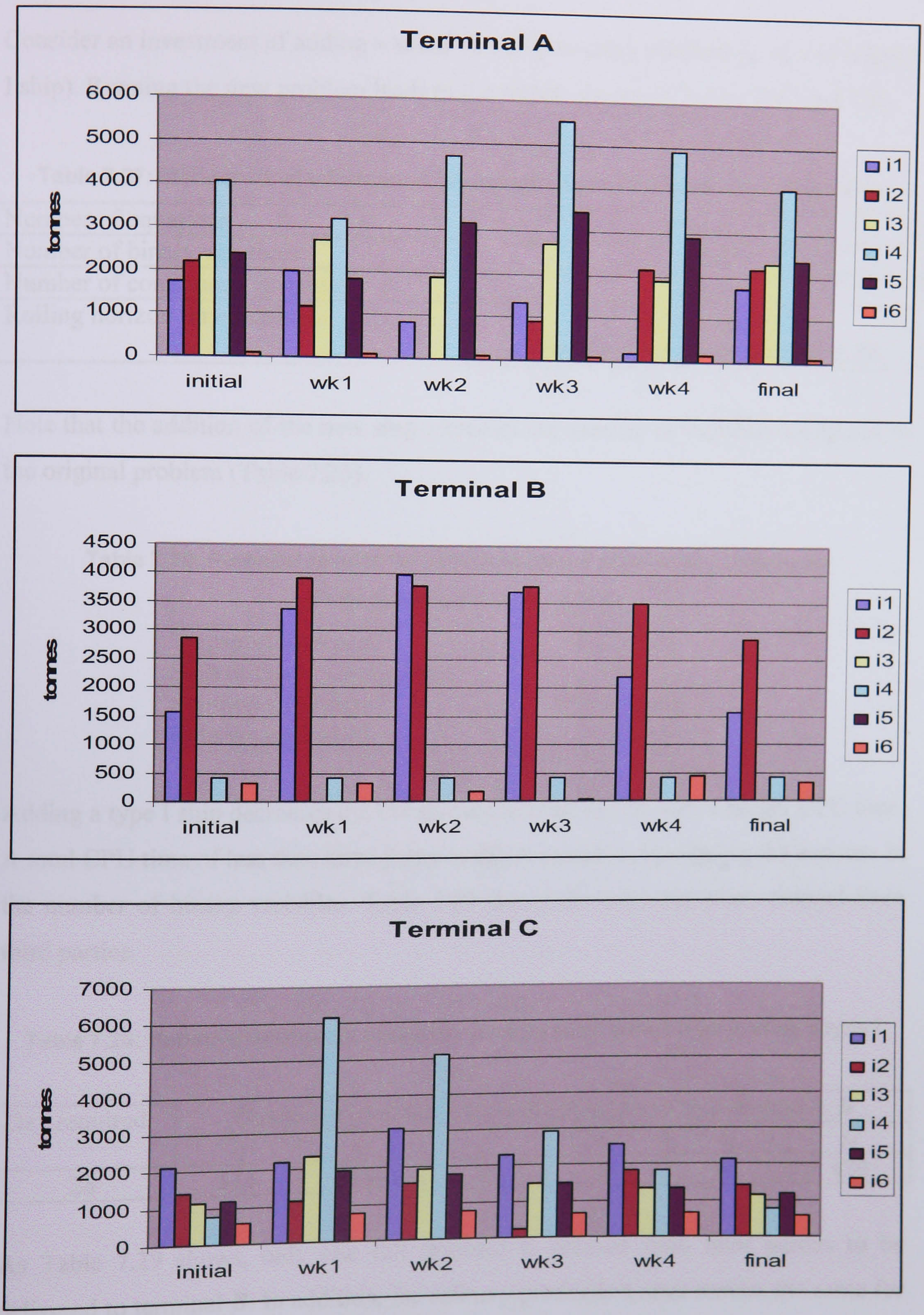

Figure 7.6: Original problem inventory levels for industrial case 2

9.2.3.1 Adding a new type I ship

Consider an investment of adding a small size ship to carry products i_1, i_2, and i_3 (type I ship). Running the new problem leads to the results shown in Tables 7.27 and 7.28.

Table 7.27: Mathematical information for industrial case 2 after adding type I ship

Number of equations	19,613
Number of binary variables	6,360
Number of continuous variables	52,213
Rolling horizon time intervals	Six intervals (t1→t10→t20→t30→t40→t50→t60)

Note that the addition of the new ship increases the number of variables compared to the original problem (Table 7.23).

Table 7.28: Potential results for industrial case 2 after adding type I ship (Fully relaxed solution = 5,000)

Transportation cost	$ 674,000
Ship utilisation percentage	80%
Solution CPU time	145 minutes
Relative optimality	10%

Adding a type I ship decreases the transportation costs by 7% with a longer CPU time. A total CPU time of less than three hours is still reasonable considering the increase in the number of binary variables. Table 7.29 shows the total quantities ordered from third parties.

Table 7.29: Potential third party quantities for industrial case 2 after adding a type I ship

Day required	Quantities (t) ordered from third parties to be delivered to ***B***					
	i_1	i_2	i_3	i_4	i_9	i_9
28	514	1466	0	0	0	420

As Table 7.29 shows, only one full shipload is ordered from third parties to be delivered to terminal ***B***. In addition, the refinery production rates remain the same for all products. Table 7.30 shows that only the production rate of i_2 increased by 1%.

Table 7.30: Potential production rates for industrial case 2 after adding a type I ship

Product name	Average production rate (t/day) at ***P***	Maximum production rate (t/day) at ***P***	Throughput	Increase in throughput
i_1	4820	5232	92 %	0 %
i_2	8947	10529	85 %	1 %
i_3	2702	2957	91 %	0 %
i_4	2186	2460	89 %	0 %
i_9	2640	2906	91 %	0 %
i_9	117	140	84 %	0 %

9.2.3.2 Adding a new type II ship

We now consider an investment of adding a small size ship to carry products i_4, i_9, and i_9 (type II ship). Running the new problem leads to the results shown in Table 7.31.

Table 7.31: Potential results for industrial case 2 after adding type II ship (Fully relaxed solution = 5,000)

Transportation cost	$ 718,000
Ship utilisation percentage	80%
Solution CPU time	100 minutes
Relative optimality	10%

Note that the addition of the new ship increases the number of variables compared to the original problem (similar to adding a type I ship). Therefore, the same mathematical information is obtained by adding a type II ship (see Table 7.27).

Adding a type II ship leads to a negligible improvement compared to the original transportation cost with a longer CPU time. A total CPU time of less than two hours is still reasonable considering the increase in the number of binary variables. Nevertheless, the main advantage of adding at type II ships is the elimination of third party orders. No material is ordered from third parties for any of the primary terminals. Consequently, as Table 7.32 shows, the production rate of i_9 increased by 10% to cover for third party quantities ordered for terminal ***B*** in the original problem.

Table 7.32: Potential production rates for industrial case 2 after adding a type II ship

Product name	Average production rate (t/day) at ***P***	Maximum production rate (t/day) at ***P***	Throughput	Increase in throughput
i_1	4837	5232	92 %	0 %
i_2	8996	10529	85 %	1 %
i_3	2702	2957	91 %	0 %
i_4	2186	2460	89 %	0 %
i_9	2640	2906	91 %	0 %
i_9	131	140	94 %	10 %

9.2.3.3 Adding new pumps to terminals A and B

Looking at the optimal solution of the original problem, we see that only terminals *A* and ***B*** required delivery from third parties. So, as a potential investment, we consider buying new pumps from these terminals. We assume that the unloading rate from at these two terminals will be doubled as a result. Running the new problem leads to the results shown in Table 7.33. Note that doubling the unloading rate does not change the number of variables compared to the original problem (see Table 7.23).

Table 7.33: Potential results for industrial case 2 after adding type II ship

(Fully relaxed solution = 6,300)

Transportation cost	$ 638,000
Ship utilisation percentage	79%
Solution CPU time	80 minutes
Relative optimality	10%

Adding new pumps at *A* and ***B*** increases the total ship-discharge quantities at those terminals, which reduces the total number of ship loads required to satisfy those terminals' demand. Consequently, the total transportation costs decreases by 12% after adding new pumps at *A* and ***B***. Table 7.34 shows the total quantities ordered from third parties.

Table 7.34: Potential third party quantities for industrial case 2 after adding new pumps

Day required	Quantities (t) ordered from third parties to be delivered to ***B***					
	i_1	i_2	i_3	i_4	i_9	i_9
28	1272	708	0	0	0	420

As Table 7.34 shows, only one full shipload is ordered from third parties to be delivered to terminal ***B***. In addition, refinery production rates remain the same for all products. Table 7.35 shows that only the production rate of i_2 increased by 1% compared to the original problem.

Table 7.35: Potential production rates for industrial case 2 after adding new pumps

Product Name	Average production rate (t/day) at ***P***	Maximum production rate (t/day) at ***P***	Throughput	Increase in throughput
i_1	4794	5232	92 %	0 %
i_2	8973	10529	85 %	1 %
i_3	2702	2957	91 %	0 %
i_4	2186	2460	89 %	0 %
i_9	2640	2906	91 %	0 %
i_9	117	140	84 %	0 %

7.2.4 Recommendations

A complex VMI problem to deliver six oil products from a refinery to geographically dispersed terminals is described. The vendor is committed to meet the demand of each product for all terminals. Besides using its three flexible-compartment ships for delivery, the vendor is responsible for ordering any necessary extra products from third parties. Terminals are classified into two categories; primary and secondary terminals. Primary terminals (***A***, ***B***, and ***C***) have higher demands of all six products and relatively more accessible ports. Secondary terminals (***D***, ***E***, ***F***, ***G***, and ***H***) have lower demands of only three products while having less accessible ports. As a result, the demand of the secondary terminals must be fully satisfied using the vendors' ships. On the other hand, the demand of the primary terminals can be satisfied using the vendors' ships as well as third-party ships. The original problem is solved to lead to a near-optimal production and shipping schedule. In addition, a log of all required third-party quantities by the primary terminals is provided to the vendor. The log shows the required quantity, the product type, and the day at which this quantity is required.

Different investment scenarios are tested each with a possible improvement in the optimal solution. Adding a type I ship results in a 7% reduction in the transportation cost. Doubling the pumping rate at terminals ***A*** and ***B*** results in a 12% reduction in the transportation cost. In both scenarios, only one shipload of third party quantities is needed. On the other hand, adding a type II ship results in a negligible change in the total transportation cost. However, no third party quantities are needed in this scenario. The decision is left to the vendor to evaluate the trade-off between total costs and third party orders. Obviously, avoiding third party orders might have an extra monetary value to the vendor. Hence, adding a type II ship could be the best option to the vendor provided that the investment cost is reasonable.

Chapter 8
Conclusions and Future Directions

8.1 Conclusions

Outsourcing non-core business processes is a growing objective in most industries. VMI plays an important role in achieving that objective. By letting the vendor take over material replenishment in the manufacturer's site, the manufacturer will concentrate on other core processes. In return, end customers will receive better service quality while costs are minimised for both manufacturers and vendors. In industrial VMI systems, a single or multiple products are shipped from production plant(s) to satisfy demand forecasts in other customer locations using minimum transportation cost. Achieving the optimisation goal includes satisfying other constraints such as storage capacity, production capabilities, and time requirements. Hence, a VMI system integrates the three major components of production planning, distribution, and inventory management. Our work efficiently modelled a shipping-based VMI system in order to assess its benefits. A CDST was designed to aid in the process of optimising the performance of industrial VMI systems.

The VMI system has been modelled as an MILP using the RTN formulation. Two types of mathematical models were designed to represent the VMI system; detailed and aggregate models. The detailed model represents the VMI system dynamically while the aggregate model focuses on steady-state overall decisions. These two models are combined carefully to construct different solution approaches. To evaluate these solution approaches, the VMI problem was solved directly using the detailed model only. An illustrative example VMI problem was used to test all solution approaches including the direct approach. The optimisation-based solutions were compared to solutions provided by industrial heuristic algorithms. To evaluate the robustness of all solution approaches, the example was solved with two different sets of demand data. The first set of data was static while the other set was relatively more dynamic. Since all routing and scheduling problems are NP-hard, the direct approach consumed unreasonably long CPU times to reach the optimal solution. This justifies the use of heuristic algorithms for such problems even on the expense of the objective value. The industrial heuristics ***H1*** and ***H2*** proved to be practical tools in producing a feasible solution for complex problems. However, our work focuses on the design of optimisation-based approaches to solve complex VMI problems.

The iterative approach combines the aggregate and the detailed models sequentially. In other words, ship-journey combinations generated by the aggregate model are used as pre-matches to the detailed problem. In case of static demand data, the iterative approach solves the illustrative example in much less time compared to the direct solution approach. However, the iterative approach fails to solve the problem with dynamic demand data. In fact, the direct solution approach was more successful than the iterative approach in reaching an optimal solution for the dynamic demand dataset. Time aggregation proved to be effective only for the static demand dataset. Static demand greatly reduces the gap between the aggregate and detailed solution. On the other hand, the more dynamic the demand is, the greater the gap between the aggregate and detailed results. Therefore, time aggregation can only be applied to cases with static demand where promising results are expected.

The direct solution approach requires unrealistically long CPU times to reach optimal solutions while the iterative approach fails to solve dynamic VMI problems. The rolling horizon approach seems to be a good compromise to balance the trade-off between detailed modelling and long CPU times. The forward rolling horizon (FRH) approach combines the aggregate and detailed models simultaneously using tight boundary conditions. When used to solve the illustrative example problem, FRH produced optimal solutions (similar to the ones produced by the direct approach). Moreover, FRH was robust to solve the problem with both static and dynamic datasets. A novel HFRH approach was proposed to solve the VMI problem. In the HFRH approach, ships are passed one by one to the optimisation process while other ships are aggregated during that time block. The HFRH approach proved to be more robust even when compared to the plain FRH approach. The same optimal solutions were reached in relatively shorter CPU times. Because all other approaches failed to produce optimal solutions for the industrial case studies, the HFRH approach was the only successful approach to solve those case studies in Chapter 7.

Recall from Section 5.3 that we do not pass the ships to the HFRH algorithm in any specific order. Instead, ships are passed in an arbitrary order. As explained in Section 5.3, the order in which ships are solved can affect the global optimal solution. This was not the case for illustrative example problem in Chapter 6 where solutions provided by the HFRH approach matched the optimal solutions (obtained by the direct approach). However, we cannot guarantee that this also applies for the industrial case studies in Chapter 7. Although solutions provided by the HFRH were more successful than those provided by heuristics, further sensitivity analysis (to the order of ships) can be performed to the industrial case studies.

8.2 Future Directions

In this section some potential work areas are suggested for any future research. These areas were not included in our work because of time limitations. However, for some areas, more reasons are mentioned as to why they were not included in our work.

- <u>Optimisation of time intervals:</u>

At the moment, rolling horizon time intervals are chosen manually and equally. To speed up the solution time, those intervals can be optimised using a small MILP pre-model. If we choose the number of binaries as a metric, then the time horizon should be divided such that the number of binary variables in each interval is roughly the same. A minimum interval length is also provided to avoid very short intervals. This pre-model can result in uneven time intervals with approximately equal numbers of binary variables.

- <u>Ship queuing:</u>

Currently, the VMI model restricts the total number of ships existing in any port to a maximum user-specified capacity. Consequently, a ship will remain in any previous port in order not to violate this constraint. A more realistic method of modelling port capacity is to construct a ship queuing system in ports. If the total number of ships is exceeded, any incoming ships will queue for a vacant berth. Queuing can be presented in terms of extra waiting time at the port in addition to any loading/unloading times.

Mathematically, the queuing system can be achieved by introducing a shadow port for every available port. If the actual port is full, the ship will go to the shadow port and wait for a vacancy in the actual port. An imaginary journey of zero duration between the actual and shadow ports is added to the model.

- Graphical user interface:

In order to complete the CDST, a user friendly-interface should be designed for data input and schedule output. The interface can greatly facilitate the testing of future case studies. MS Access coupled with Visual Basic can be used to design the interface.

- Incorporation of fixed-compartment ships:

Ships used in the VMI system are considered to be flexible-compartment ships. These ships are provided with special movable bulkheads. As a result, if more than one product is being shipped, any different quantities of these products can be shipped together provided that the total quantity does not exceed the ship capacity. On the other hand, fixed-compartment ships are provided with separate storage facilities onboard. Therefore, shipping more than one product requires a special kind of optimisation in order to efficiently utilise the total storage capacity. Incorporation of fixed-compartment ships requires an MILP model and linking this model with the entire VMI system will greatly affect the total solution speed. Fixed-compartment ships are usually old and most newly-manufactured ships are flexible-compartment ships. However, considering these ships in VMI systems poses an interesting modelling challenge.

- Demand forecasting:

In industrial VMI systems, a contract between the vendor and the customer guarantees that material will always be available to cover the forecasted demands. Demand forecasting in such systems is done by the customer, and the vendor agrees only to meet these forecasts for a long period of time (usually a whole year). A penalty cost for not meeting demand forecasts is negotiated between the two parties. Nevertheless, to further improve the CDST, demand forecasting can be included as pre-optimisation stage.

- Consideration of uncertainty:

Deterministic VMI problems are NP-hard by nature. Adding uncertainty will make the problem more difficult to solve even with the proposed solution approaches. Moreover, uncertainty is present in every decision in the VMI system such as journey durations, demand fluctuations, unplanned shut-downs, ship breakdown, etc. Including all these uncertainties in the model is impossible. The current CDST is designed to provide an optimal schedule for a specific time horizon. To account for any unforeseen disturbances, the CDST can be used to resolve the problem more frequently while extending the time horizon with every resolution. A future objective is to try to derive "robust" schedules that will perform well under different scenarios.

- Production planning extension

VMI systems integrate production planning, distribution, and inventory management. In addition to the optimal ship schedule, we optimise the system from the vendor's perspective by providing the optimal daily production plans. Those plans are optimal because they meet the shipping schedule and the vendor's storage capacity. A further optimisation of production planning can be considered to explore the raw material availabilities. In other words, the supply chain can be extended to cover the raw material supplies to the vendor by other members. Because of the production schedules' dependencies on materials, the use of Material Requirements Planning (MRP) can be of great benefit (Martin, 1995).

References

Achabal, D. D., S. H. Mcintyre, S. A. Smith and K. Kalayanam (2000). "A Decision Support System for Vendor Managed Inventory." Journal of Retailing 76(4): 430-454.

Assad, A. A. (1988). Modeling and Implementation Issues in Vehicle Routing. Vehicle Routing: Methods and Studies. Amsterdam, Elsevier Science Publishers B. V.: 7-45.

Axsater, S. (2001). "A Note on Stock Replenishment and Shipment Scheduling for Vendor-Managed Inventory Systems." Management Science 47(9): 1306-1310.

Bard, J. F., L. Huang, P. Jaillet and M. Dror (1998). "A Decomposition Approach to the Inventory Routing Problem with Satellite Facilities." Transportation Research 32(2): 189-203.

Beasley, J. E. (2003). Or-Notes, Imperial College London. http://mscmga.ms.ic.ac.uk/jeb/or/contents.html

Bramel, J. and D. Simchi-Levi (1995). "A Location Based Heuristic for General Routing Problems." Operations Research 43(4): 649-660.

Brooke, A., D. Kendrick, A. Meeraus and R. Raman (1992). Gams: A Users Guide: Release 2.25, the Scientific Press Series. Danvers, MA, Boyd and Fraser Publishing Co.

Burkard, R. E., V. G. Deineko, R. V. Dal, J. A. A. V. D. Veen and G. J. Woeginer (1998). "Well-Solvable Special Cases of the Traveling Salesman Problem: A Survey." Society for Industrial and Applied Mathematics 40(3): 496-546.

Campbell, A., L. Clarke, A. Kleywegt and M. Savelsbergh (1997). The Inventory Routing Problem. The Logistics Institute, School of Industrial and Systems Engineering, Georgia Institute of Technology

Cetinkaya, S. and C.-Y. Lee (2000). "Stock Replenishment and Shipment Scheduling for Vendor-Managed Inventory Systems." Management Science 46(2): 217-232.

Cheng, L. and M. A. Duran (2004). "Logistics for World-Wide Crude Oil Transportation Using Discrete Event Simulation and Optimal Control." Computers and Chemical Engineering **28**: 897-911.

Christiansen, M. and K. Fagerholt (2000). "A Combined Ship Scheduling and Inventory Allocation Problem." Journal of the Operational Research Society 51: 834-842.

Christiansen, M. and K. Fagerholt (2001). "A Travelling Salesman Problem with Allocation, Time Window and Precedence Constraints - an Application to Ship Scheduling." International Transactions in Operational Research 7: 231-244.

Christiansen, M. and K. Fagerholt (2002). "Robust Ship Scheduling with Multiple Time Windows." Naval Research Logistics 49: 611-625.

Christiansen, M., K. Fagerholt and D. Ronen (2004). "Ship Routing and Scheduling: Status and Perspectives." Transportation Science 38(1): 1-18.

Christopher, M. (1998). Logistics and Supply Chain Management. London, Financial Times Prentice Hall.

Cohen, M. A. and H. L. Lee (1988). "Strategic Analysis of Integrated Production Distribution Systems: Models and Methods." Operations Research 36(2): 216-228.

Dantzig, G. B., D. R. Fulkerson and S. M. Johnson (1954). "Solution of a Large-Scale Travelling-Salesman Problem." Operations Research 2: 393-410.

Desrochers, M., J. K. Lenstra and M. W. P. Savelsbergh (1990). "A Classification Scheme for Vehicle Routing and Scheduling Problems." European Journal of Operational Research 46: 322-332.

Dimitriadis, A. D. (2000). Algorithms for the Solution of Large-Scale Scheduling Problems. Centre for Process Systems Engineering. PhD thesis, Imperial College London

Disney, S. M., A. T. Potter and B. M. Gardner (2003). "The Impact of Vendor Managed Inventory on Transport Operations." Transportation Research Part E(39): 363-380.

Disney, S. M. and D. R. Towill (2002). "A Discrete Transfer Function Model to Determine the Dynamic Stability of a Vendor Managed Inventory Supply Chain." International Journal of Production Research 40(1): 179-204.

Disney, S. M. and D. R. Towill (2002). "A Procedure for the Optimization of the Dynamic Response of a Vendor Managed Inventory System." Computers & Industrial Engineering 43: 27-58.

Dong, Y. and K. Xu (2002). "A Supply Chain Model of Vendor Managed Inventory." Transportation Research, Part E 38: 75-95.

Dror, M. and Michael Ball (1987). "Inventory/Routing: Reduction from Annual to an Short-Period Problem." Naval Research Logistics 34: 891-905.

Fagerholt, K. (2000). "Evaluating the Trade-Off between the Level of Customer Service and Transportation Costs in a Ship Scheduling Problem." Maritime Policy & Management 27(2): 145-153.

Fagerholt, K. (2001). "Theory and Methodology Ship Scheduling with Soft Time Windows: An Optimization Based Approach." European Journal of Operational Research 131: 559-571.

Fisher, M. L. and M. B. Rosenwein (1989). "An Interactive Optimization System for Bulk-Cargo Ship Scheduling." Naval Research Logistics 36: 27-42.

Gallego, G. and D. Simchi-Levi (1990). "On the Effectiveness of Direct Shipping Strategy for the One Warehouse Multi-Retailer R-Systems." Management Science 336: 240-243.

Garey, M. R. and D. S. Johnson (1979). Computers and Intractability: A Guide to the Theory of Np-Completeness. New York, Freeman.

Jaillet, P., J. F. Bard, L. Huang and M. Dror (2002). "Delivery Cost Approximations for Inventory Routing Problems in a Rolling Horizon Framework." Transportation Research 36(3): 292-300.

Jalisi, Q. W. Z. (2000). Logistics and Supply Chain Optimal Planning and Scheduling. Department of Electrical and Electronic Engineering. PhD thesis, Imperial College London

Jetlund, A. S. and I. A. Karimi (2004). "Improving the Logistics of Multi-Compartment Chemical Tankers." Computers and Chemical Engineering 28: 1267-1283.

Jorgensen, J. B. (2000). "Towards Automatic Decentralized Control Structure Selection ".Computers and Chemical Engineering.846-841 :**24**

Kaipia, R., J. Holmstrom and K. Tnskanen (2002). "Vmi: What Are You Losing If You Let Your Customer Pace Orders?" Production Planning & Control 13(1): 17-25.

Kao, C. and H. T. Lee (1996). "Discrete Time Parallel-Machine Scheduling; a Case of Ship Scheduling." Engineering Optimisation 26: 287-294.

Karimi, I. A., M. Sharafali and H. Mahalingam (2005). "Scheduling Tank Containers Movements for Chemical Logistics." American Institute of Chemical Engineers 51(1): 178-197.

Kim, S.-H. and K.-K. Lee (1997). "An Optimization-Based Decision Support System for Ship Scheduling." Computers & Industrial Engineering 33: 689-692.

Kuk, G. (2004). "Effectiveness of Vendor-Managed Inventory in the Electronics Industry: Determinants and Outcomes." Information & Management **41**: 645-654.

Laporte, G. (1992). "The Travelling Salesman Problem: An Overview of Exact and Approximate Algorithms." European Journal of Operational Research 59: 231-247.

Laporte, G. and I. H. Osman (1995). "Routing Problems: A Bibliography." Annals of Operations Research 61: 227-262.

Lieberman, G. J. and F. S. Hillier (2001). Introduction to Operations Research. NY, McGraw-Hill.

Martin, A. J. (1995). Distribution Resource Planning. New York, John Wiley & Sons Inc.

Miller, D. M. (1987). "An Interactive, Computer-Aided Ship Scheduling System." European Journal of Operational Research 32: 363-379.

Nygreen, B. and M. Christiansen (1999). "Modelling Path Flows for a Combined Ship Routing and Inventory Management Problem." Annals of Operations Research 82: 391-412.

Pantelides, C. C. (1994). "Unified Frameworks for Optimal Process Planning and Scheduling." Proceedings on the Second Conference on Foundations of Computer-Aided Operations" (D. W. T. Rippin and J. Hale ,eds.), pp. 253-274.

Ronen, D. (1983). "Cargo Ships Routing and Scheduling: Survey of Models and Problems." European Journal of Operational Research 12: 119-126.

Ronen, D. (1986). "Short-Term Scheduling of Vessels for Shipping Bulk or Semi-Bulk Commodities Originating in a Single Area." Operations Research 34(1): 164-173.

Ronen, D., G. G. Brown and G. W. Graves (1987). "Scheduling Ocean Transportation of Crude Oil." Management Science 33(3): 335-346.

Rusdiansyah, A. and D. Tsao (2005). "An Integrated Model of the Periodic Delivery Problems for Vending-Machine Supply Chains." Journal of Food Engineering **70**: 421-434.

Taras, J. (2001). Vendor Managed Inventory.Com. 2001.www.vendormanagedinventory.com

Toni, A. F. D. and E. Zamolo (2005). "From a Traditional Replenishment System to Vendor-Managed Inventory: A Case Study from the Household Electrical Appliances Sector." International Journal of Production Economics **96**: 63-79.

Ukovich, F. B. W., R. Pesenti and D. Favaretto (1998). "Dynamic Routing-and-Inventory Problems: A Review." Transportation Research, Part A 32(8): 585-598.

Viswanathan, S. and K. Mathur (1997). "Integrating Routing and Inventory Decisions in One-Warehouse Multiretailer Multiproduct Distribution Systems." Management Science 43(3): 294-312.

Waller, M., M. E. Johnson and T. Davis (1999). "Vendor-Managed Inventory in the Retail Supply Chain." Journal of Business Logistics 20(1): 183-203.

Wilkinson, S. J. (1996). Aggregate Formulations for Large-Scale Process Scheduling Problems. Department of Chemical Engineering and Chemical Technology. PhD thesis, Imperial College London

Williams, H. P. (1999). Model Building in Mathematical Programming. Chichester, John Wiley & Sons Ltd.

Appendix

Calculation of j_{max}

$j_{\max}$ is the maximum number of aggregate journeys (j_{max} = card(z)). Because we do not know the total number of journeys a ship can take, we estimate the maximum number of journeys it can take during the entire time horizon. j_{max} is calculated by dividing the total time horizon (H) by the shortest possible sequence of events. The following equation shows the mathematical derivation. Note that the denominator contains the minimum journey and unloading durations (excluding the journey to the dummy location). Then, an extra journey is added to cover the trip to the dummy location. The division values might lead to non-integer value. Therefore, it is rounded up to the next integer.

$$j_{\max} = [\frac{H}{\min(\tau_{sj}) + \min(\psi_s / T_l)}] + 1$$