

A MULTI-STAGE DECISION-SUPPORT METHODOLOGY FOR THE
OPTIMIZATION-BASED LINER-NETWORK DESIGN

by

PANAGIOTIS S. TSILINGIRIS

Submitted in Partial Fulfillment of the Requirements of the Degree of Diploma in Naval
Architecture and Marine Engineering at the School of Naval Architecture and Marine
Engineering of the National Technical University of Athens

June, 2005

Thesis Supervisor: Professor Dr. Harilaos N. Psaraftis

Signature of Author
June 11, 2005

A MULTI-STAGE DECISION-SUPPORT METHODOLOGY FOR THE
OPTIMIZATION-BASED LINER-NETWORK DESIGN

by
PANAGIOTIS S. TSILINGIRIS

Submitted in Partial Fulfillment of the Requirements of the Degree of Diploma in Naval
Architecture and Marine Engineering at the School of Naval Architecture and Marine
Engineering of the National Technical University of Athens

KEYWORDS: (holistic) liner-network design, ship routing and scheduling, fleet
deployment, transshipment

ABSTRACT

The central contribution of this thesis is to provide a decision-support optimization-based methodology for a broad class of problems in liner shipping. The present thesis aims at the optimization of liner networks by transforming the total network design into a sequential multi-stage optimization process in terms of *Ship Routing & Scheduling*, *Fleet Deployment*, and *Transshipment*. However, the stages of the methodology proposed hereinafter are not completely autonomous as in the sense of conventional multi-stage optimization, but they interact with each other as well as with the decision-maker. By fixing the various sources of non-linearity and by broking down the total network design into the sequential solution of the aforesaid set of subproblems we have managed to accomplish our goals via the use of Linear Programming, Dynamic Programming and Integer Programming.

Firstly, the model determines the *sequence of ports* in each route. Secondly, the *frequency of service* is decided. Thirdly, we resolve *re-routing/minor routing* through visualizing network efficiency by certain graphs and/or by the application of certain quantitative criteria. Fourthly, we present and apply an Integer Programming Model for the *allocation of ships to routes*. Finally, once the abovementioned stages/procedures have been successfully applied and we have an efficient network from the routing and scheduling point of view as well as an optimal one from the fleet deployment point of view, we can optionally apply the *transshipment* module for the optimization of regional, elementary or not, sub-networks.

An extensive eclectic literature review has been attempted including mathematical programming techniques, algorithms and complexity, the Traveling Salesman Problem, the Vehicle Routing Problem and the implementation of OR models in liner shipping not only for the better understanding of the optimization methodology but also to make the present thesis a reference for an undergraduate/graduate student who wants to research in the field.

Thesis Supervisor: Professor Dr. Harilaos N. Psaraftis

In memory of my grandfather Lucas...

*“Keep Ithaca always in mind.
Arriving there is what you are destined for.”*

Excerpt from Ithaca
Written in Greek by C. P. Cavafy
Translated by Edmund Keeley and Philip Sherrard

Acknowledgements

I gratefully acknowledge Professor Harilaos N. Psaraftis for supervising the present thesis. He has patiently read and commented on my thesis drafts and has challenged me to focus on the practical aspect of my research. Moreover, I would like to thank him for introducing me to the world of quantitative decision-making. I am deeply indebted to Professor Psaraftis for his support throughout my studies at the National Technical University of Athens as well as for the continuation of my studies at the MIT/Zaragoza Logistics Center.

I deeply thank Assistant Professor Dimitrios V. Lyridis for his friendship and his assistance on various academic and personal matters. I also thank him for serving on my thesis committee.

I am grateful to my parents, Stasinos and Eleftheria Tsilingiris, for nurturing me in a love for learning.

I also thank my friends during my studies at the NTUA. I would like to mention the name of Vassilis Tourloupis for his assistance during the last period of my studies.

I would like to specially thank Nicholas Apostolou for his invaluable support on various academic matters.

Last but not least, I would like to thank Demi Rigou for her assistance in the typing of the present thesis as well as for the encouragement.

May God bless you all.

TABLE OF CONTENTS^j

Page	
Abstract.....	2
Dedication.....	3
Acknowledgements.....	5
Table of Contents.....	6
List of Figures.....	7
List of Tables	9
Abbreviations.....	10
 1. INTRODUCTION	
1.1. Scope and organization of the Thesis.....	12
 5. A MULTI-STAGE DECISION-SUPPORT METHODOLOGY FOR THE OPTIMIZATION-BASED LINER-NETWORK DESIGN	
5.1. Introduction and outline.....	15
5.2. Methodology philosophy and assumptions.....	16
5.3. Fleet Routing.....	21
5.4. Deciding frequency of service and re-routing.....	26
5.5. Allocation of ships to routes.....	34
5.6. Transshipment.....	44
5.7. Travel costs and travel times estimation subroutine.....	48
5.8. The revenue calculation.....	56
5.9. Data and their sequential application to the methodology.....	59
5.10. Conclusions	67
5.11. Recommendations for future work.....	68
 6. RECAPITULATION.....	71
 7. REFERENCES	73
 APPENTICES	
Appendix A: Elements of Graph Theory.....	77
Appendix B: Additional information about LINGO.....	81
Appendix C: Full LINGO output of the fleet deployment example.....	92

^j In the present Web version of the Thesis included are parts from chapter five, “thenceforth”.

LIST OF FIGURES

Figure 1.1: Characteristics of maritime transportation demand and supply

Figure 2.1 Types of model representation.

Figure 2.2 Multistage decision process.

Figure 2.3 Disjoint subtours.

Figure 2.4 Sorting a hand of cards using insertion sort

Figure 2.5 The operation of INSERTION-SORT

Figure 3.1 Liner pricing case 1: marginal cost pricing

Figure 3.2 Liner pricing case 2: fixed pricing

Figure 3.3 Container movements (lifts) 1973-94

Figure 3.4 Liner fleet by ship type

Figure 3.5 Container ship fleet 1980-96

Figure 3.6 Container ship fleets by capacity 1996

Figure 3.7 Container ship

Figure 3.8 Multi-purpose vessels

Figure 4.1 Minimum-cost flow problem.

Figure 4.2 Transportation network.

Figure 4.3 Maximal Flow Problem

Figure 4.4 Special cases and generalizations of the TSP.

Figure 4.5: Liner network design for traditional liner operations including some but not all routes

Figure 4.6: Liner network design for a hub and spoke system. Example of three overlapping routes

Figure 4.7 (a): Fleet deployment with non-overlapping routes and a common hub

Figure 4.7 (b): Fleet deployment with non-overlapping routes and several hubs

Figure 5.1. Flow Diagram of the Multi-Stage Optimization Concept

Figure 5.2. Ship Capacity vs. Frequency of Service

Figure 5.3. Ship cargo levels

Figure 5.4: Regional hub-and-spoke networks

Figure A.1. (a) graph; (b) multigraph; (c) pseudograph; (d) digraph; (e) network

Figure A.2. Parts of the Graph in Fig. 5.1(a): (a) Subgraph; (b) Partial Graph; (c) Partial Subgraph.

LIST OF TABLES

Table 1.1: Development of international seaborne trade (millions of tons)
Table 1.2: World fleet by vessel type (million dwt)
Table 1.3: Productivity of the World Fleet
Table 1.4: Comparison of Operational Characteristics of Freight Transportation Modes
Table 2.1 Classification of Analytical and Simulation Models
Table 3.1 World container fleet 1960-95
Table 3.2 World container stock by principal type
Table 3.3 The six building blocks of liner costs
Table 3.4 The liner service cash flow model
Table 3.5 The container and break bulk trades 1980-1994
Table 3.6 Major liner routes and trade imbalances, 1996
Table 3.7 World container port traffic (1994)
Table 3.8 Principal dimensions of flat roof steel containers
Table 3.9 The container ship fleet, by size and hull characteristics
Table 4.1 Examples of Network Flow Problems ³
Table 4.2 Tableau for Minimum-Cost Flow Problem
Table 4.3 Unit Production and Shipping Costs
Table 4.4 Summary of literature review on strategic ship routing and scheduling
Table 4.5: Summary of commercial cargo routing and scheduling in industrial shipping
Table 4.6: Summary of literature review on inventory ship routing and scheduling
Table 4.7: Summary of literature review on routing and scheduling in tramp shipping
Table 5.1 Calculation of cost and times coefficients
Table 5.2 Initial Ship allocation
Table 5.3 Resultant Ship allocation
Table 5.4 Comparison of Frequencies

Abbreviations

ABS	American Bureau of Shipping
ADTSPTW	A priori Dynamic Traveling Salesman Problem with Time Windows
COA	Contract of Affreightment
DARP	Dial-A-Ride Problem
DARPTW	Dial-A-Ride Problem with Time Windows
DTSPTW	Dynamic Traveling Salesman Problem with Time Windows
FAK	Freight All Kinds
FD	Fleet Deployment
GATT	General Agreement
IACS	International Association of Ports and Harbours
ILO	International Labour Organization
ILP	Integer Linear Programming
IMO	International Maritime Organization
IP	Integer Programming
ISO	International Standards Organization
LP	Linear Programming
MDVRP	Multi-Depot Vehicle Routing Problem
m-PDPTW	multi-vehicle Pickup and Delivery Problem with Time Windows
OPEC	Organization of Economic Co-operation and Development
OR	Operations Research
PILP	Pure Integer Linear Programming
PTSP	Periodic Traveling Salesman Problem
PVRP	Periodic Vehicle Routing Problem
TSP	Traveling Salesman Problem
TSPDC	Traveling Salesman Problem with mixed Deliveries and Collections
TSPTW	Traveling Salesman Problem with Time Windows

UN	United Nations
UNCTAD	United Nations Conference on Trade and Development
VRP	Vehicle Routing Problem
VRPSTW	Vehicle Routing Problem with Soft Time Windows
VRPTW	Vehicle Routing Problem with Time Windows

1. INTRODUCTION

1.1. Scope and organization of the Thesis

The central contribution of this thesis is to provide a decision-support optimization-based methodology for a broad class of problems in liner shipping. The present thesis aims at the optimization of liner networks by transforming the total network design into a sequential multi-stage optimization process in terms of *Ship Routing & Scheduling*, *Fleet Deployment*, and *Transshipment*. However, the stages of the methodology proposed hereinafter are not completely autonomous as in the sense of conventional multi-stage optimization, but they interact with each other as well as with the decision-maker. By fixing the various sources of non-linearity and by broking down the total network design into the sequential solution of the aforesaid set of subproblems we have managed to accomplish our goals via the use of Linear Programming, Dynamic Programming and Integer Programming.

Subchapters 1.2 and 1.3 familiarize the reader with maritime transport, its planning problems and the relevant OR-applications. In chapter 2, a brief introduction to the mathematical techniques used in the present methodology is accomplished. Chapter 3 describes as best as possible the main characteristics of the liner shipping market. In chapter 4 an extensive literature review has been accomplished

The fifth chapter, which is relatively self-contained, is organized as follows: In Section 5.2, the philosophy of the methodology is presented along with the principal assumptions of the model. Section 5.3, describes the optimization methodology regarding fleet routing & scheduling. Section 5.4, illustrates the re-routing methodology as well as some criteria for the network efficiency evaluation. Section 5.5 is dedicated to the analysis of the optimization model that allocates ships to routes. Section 5.6 briefly speaks of the optional and final stage of the optimization procedure, that is, transshipment. In section 5.7, a mathematical model is fully described in order to calculate the cost and time

components in liner services. In section 5.8, elements of revenue calculation are presented. In section 5.9, the data required and their sequential application to our methodology is summarized. In Section 5.10, conclusion and summary of results is presented. In Section 5.11, recommendations for future work are presented.

Finally, in chapter 6, a brief recapitulation has been accomplished.¹

I note that the present thesis contains significant parts of the following sources (in parentheses are the certain parts of the thesis that these sources have been used):

- PSARAFTIS, H. N. (1995) Dynamic vehicle routing: status and prospects, *Annals of Operations Research*, 61, 143-164. (§4.4)
- POWELL, B. J. and PERAKIS, A. N. (1997) Fleet deployment optimization for liner shipping: an integer programming model. *Maritime Policy and Management* 24, 183-192. (§5.5, 5.7)
- MARIELLE CHRISTIANSEN & KJETIL FAGERHOLT, Ship routing and scheduling - Status and trends, Transportation Science, *Transportation Science*, vol. 38. no1, Feb. 2004. (§1.2, 1.3, 4.5)
- JOHNSON, D. S. AND PAPADIMITRIOU, C. H. (1985) Computational Complexity, eds. LAWLER, E. L., LENSTRA, J. K., RINNOOY KAN, A. H. G. and SHMOYS, D. *The traveling salesman problem*, Wiley Interscience (§4.2)
- STOPFORD, M. (1997) Maritime Economics, *Routhledge*, 2nd edition. (§3)
- PSARAFTIS, H. N. (1984) On the practical importance of asymptotic optimality in certain heuristic algorithms, *Networks*, Vol. 14, pp.587-596. (§5.11)
- BRADLEY, HAX, and MAGNANTI (1977), Applied Mathematical Programming. *Addison-Wesley*. (§2.1-2.4)
- CORMEN H., T et al. (2001) Introduction to Algorithms, MIT Press. (§2.7)

All the aforesaid papers are excellent ones; hence, I have used original parts of them without modifying them significantly so as not to “adulterate” them. I state self-consciously that I have absolutely no intention to “usurp” the work of prominent researchers whose work I greatly respect.

I included in the present thesis reviews of mathematical modeling techniques, algorithms, etcetera, because the present thesis is based on academic subjects that do not belong to

¹ In the present Web version of the Thesis included are parts from chapter five, “thenceforth”.

the curriculum of the NAME department at the NTUA. Additionally, my wish is that the present thesis will substitute a reference for an undergraduate/postgraduate student who wants to familiarize himself with the field.

5. A MULTI-STAGE DECISION-SUPPORT METHODOLOGY FOR THE *OPTIMIZATION-BASED LINER-NETWORK DESIGN*

5.1. Introduction & Outline

Liner shipping is the type of maritime transport that has received the least attention of researchers, at least in the quantitative aspects of it. This is due to the nature of some dominant variables and factors that affect the operation of this type of market, such as minimum required service frequencies, government regulations, etc., which discourage a systematic approach to the transportation system analysis and optimization. To my best knowledge, no model exists that can tackle both the vessel routing & scheduling problem and the fleet deployment problem in a unified way. However, some companies may have solved some simplified subproblems in-house but their methodologies and results have been kept confidential for obvious reasons.

The present chapter attempts to describe as laconically as possible a methodology aiming at the optimization of liner networks. We believe that this methodology is generic enough and it can possibly be successfully applied to a plethora of liner shipping companies (including SCS). The procedure for the multi-criteria optimization of a transportation network is convoluted enough and cannot be presented as a single-stage process. This thesis is organized as follows: In Section 5.2, the philosophy of the methodology is presented along with the principal assumptions of the model. Section 5.3, describes the optimization methodology regarding fleet routing & scheduling. Section 5.4, illustrates the re-routing methodology as well as some criteria for the network efficiency evaluation. Section 5.5 is dedicated to the analysis of the optimization model that allocates ships to routes. Section 5.6 briefly speaks of the optional and final stage of the optimization procedure, that is, transshipment. In section 5.7, a mathematical model is fully described in order to calculate the cost and time components in liner services. In section 5.8, elements of revenue calculation are presented. In section 5.9, the data required and their sequential application to our methodology is summarized. In Section 5.10, conclusion and summary of results is presented. Finally, in Section 5.11, recommendations for future work are presented.

5.2. Methodology Philosophy & Assumptions

5.2.1. *Methodology Philosophy*

The present methodology attempts to tackle the *Ship Routing and Scheduling Problem* as well as the *Fleet Deployment* problem. *Routing* is the assignment of a sequence of ports to a vessel, *scheduling* is assigning times (or time windows) to the various events on a ship's route. *Deployment* refers to the assignment of the vessels in the fleet to trade routes. *Deployment* is usually used when vessels are designated to perform multiple consecutive trips on the same route, and therefore is associated with liners and a longer planning horizon. Liners follow a published sailing schedule and face more stable demand.

The goal of the optimization methodology is of a strategic nature, rather than oriented to the day-to-day decision process, which involves decisions about adjustments of schedules, slight routing modifications, etc., for which human intervention is essential. As a result, the initial condition of the ships will not be included in the problem.

5.2.2. **Assumptions**

Due to the inherent complexity that characterizes liner networks, it seems absolutely intractable to manage to optimize a network holistically without making significant assumptions.

5.2.2.1. *Speed of ships*

It is essential that all ships assigned to the same route sail at a similar speed in order to keep *frequency of service* constant (the term frequency of service refers to the time between arrivals/sailings at/from a port). However, since the speed of each ship is determined independently, the speeds of the various ships assigned to a route at the previous step will probably be different from one ship to another. Times in port per voyage on a given route are the same for all the ships. Delays caused by waiting and

sailing in canals and restricted waters in general are also the same for all ship types operating in the same route; consequently total voyage times are equal for all the ships on a given route. In this way, the intervals between arrivals of consecutive ships at port are constant. In our model, it is assumed that there is exactly one call per voyage to each port, therefore the number of calls per year at a port is also the number of voyages per year in that route. Although an assumption is thought in most cases to work to our disadvantage, this one implies a remarkable uniformity of schedules, something really beneficial to every company in liner shipping.

Naturally, in the actual operations of a liner shipping company, there is a plethora of factors which cause delays/advances in the schedules and are dealt with the everyday decision process by the operations department, e.g. bad weather (at sea and at port), cargo demand variations, etc. The small sailing time differences can be dealt similarly due to small speed differences among ships in the same route. These decisions are associated with when exactly a ship must sail (should it stop loading now and leave the rest of the cargo for the next ship?), how many stevedore gangs should work on a ship on a particular day (this will determine the length of the port stay), etc. For example, slower ships could be assigned slightly less cargo in ports or more stevedore gangs in order to reduce their port stay. It is possible though that the output allocates “speed incompatible” ships in the same route. In this case, we can through sensitivity analysis re-allocate each particular ship to the best alternative route. The inclusion of additional incompatibility constraints may be tried in order to avoid the assignment of incompatible ships to the same route.

The optimal speed determination problem is decoupled from the present methodology. *The most profitable speed for each ship should be found or a priori decided*, and in this way the operating costs of the ships at sea for a given voyage will be fixed. In this way, we avoid non-linear programming.

5.2.2.2. Resistance of Ships

The relationship versus resistance of the ships is basically the same for the typical different loading conditions that occur in liner operations. This implies that there are no long sailing legs where the ships are in a pure ballast condition; that is a rather realistic assumption for most cases. From a marine engineer's point of view, it is worth noting that when speaking of medium or small ships (like most/all ships in the present analysis) the (deadweight)/(Total Displacement) ratio is not as high as in larger ships (such as tankers or bulkers) and therefore the change in hydrodynamic conditions when the ship passes from loaded to ballast condition is not as dramatic. In addition, when the ship is in off-design conditions (as in the ballast case), its performance may not improve substantially, in spite of the significant reduction in displacement.

5.2.2.3. Cargo Movements & Demand Forecasting

The model focuses on vessels that transport one type of cargo, such as containers (our case). In line with this, a ship's cargo capacity along with the amount of cargo to be carried between pairs of ports can be given in terms of twenty-foot equivalent units (TEUs).

We make the assumption that the total amount of cargo offered per year between pairs of ports is independent of the service frequency. This assumption implies that the shipping company will neither lose nor gain cargo if frequencies of service are modified; that is a realistic assumption for the case of moderate variations. Regarding cargo demand forecasting, it seems absolutely intractable for a reliable estimation to occur inasmuch as we have TEU movements from only 4 years. Moreover, these data refer to non-homogeneous itineraries that have changed from year to year and even during the same year. Needless to say, the concept of generated demand further complicates the cargo forecasting. In order to enable demand forecasting we should have cargo data collected from many years referring to the same routes and port sequences and to stable frequencies of service. For these reasons, we think that we should ask from the decision-maker to disclose her demand estimates for us. Taking to account that the character of the hereinafter-proposed methodology is of a strategic nature, I suggest that the decision maker should inform us of their cargo movement goals and the last should be used as input to our model.

5.2.2.4. *Cost Minimization \Rightarrow Profit Maximization*

The objective function should be the annual operating costs of a fleet of liners and no optimization procedure is needed based on the revenue function. The reason is that the minimization of the annual operating costs is equivalent to maximizing profits per unit time insofar as the cargo movement requirements are met, the freight rates are assumed fixed, and the revenue is, therefore, constant.

5.2.2.5. *Miscellaneous Remarks*

The present methodology presents great adaptability to the variability of:

- Fuel oil prices as well as generator fuel prices;
- change in the number and types of the owned ships and of the ships available to charter-in; and
- changes in time and cost components

Moreover, it has components that can among others calculate:

- the cargo levels on board at each leg ij of a voyage;
- the least-distance port sequence;
- cost and time components in an exact way; and
- multiple quantitative criteria which evaluate the efficiency of a network.

Please see in advance the flow scheme of the multi-stage optimization concept.

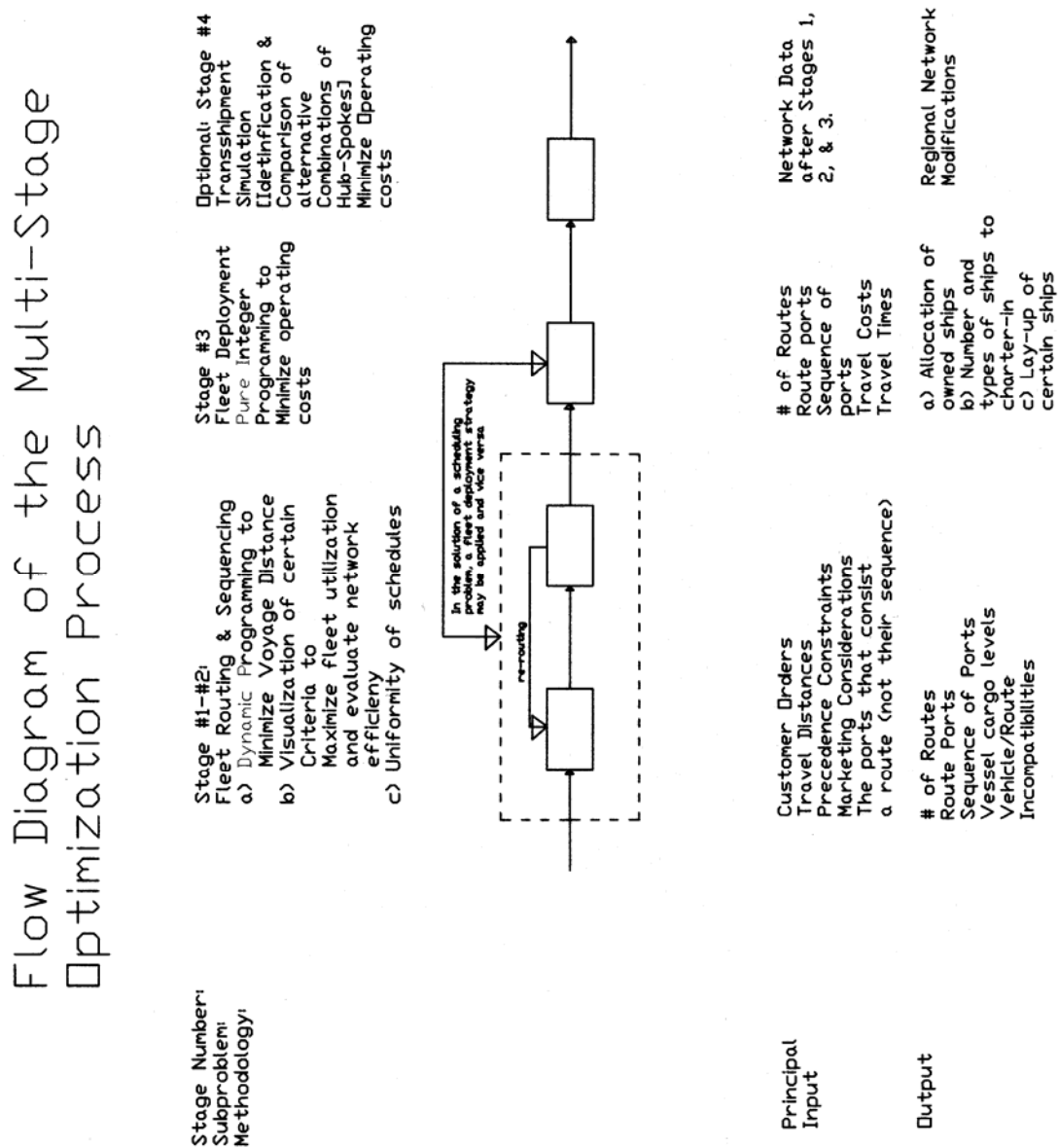


Figure 5.1. Flow Diagram of the Multi-Stage Optimization Concept

5.3. Fleet Routing

For the purpose of routing determination, traveling salesman problem models can be extremely useful to our analysis. However, it is often obvious which the best sequence of ports is in a given trade, because of geographical considerations and precedence constraints. Sometimes, because of geographical considerations alone, the set of routes may also be obvious (not our case).

Our methodology starts from an initial given and fixed set of routes. That means that the number of routes is predetermined as well as the ports that each route is comprised of. For example, we can have a route called Spain-East Med, which is comprised of the ports Valencia, Alexandria, Limassol, Beirut, Lattakia, Mersin. For the purpose of port sequencing determination, the thereunder computer program called TSPdyn can be applied.

TSPdyn is a computer program that can solve effectively every case of the Asymmetric Traveling Salesman Problem. It is written in C++ Programming Language. The compiler used is the “Borland C++ Compiler 5.5”. The only restriction is that it can solve problems that have up to approximately 15 nodes. TSPdyn is going to be applied to each route; as a result, the 15 nodes limit will not be a problem. I deem that there is not going to be a route of more than 7-8 ports. However, it can be extended to bigger sizes provided that the hardware is capable of solving it. We have assumed that a path exists between each pair of nodes. It is a rather realistic assumption as the program aims at ocean transportation. The set S is represented by a bitset.

The most important advantage of TSPdyn is that it is generic enough so that the user need not make complicated changes to adapt to different sizes of the network or to unexpected changes of the costs/distances. Specifically, the user gives the number of nodes while the data are represented in a different file. As a result, it can be used also for other applications.

TSPdyn is based upon the following formulation:

Given a set $S \subseteq \{2,3,\dots,k\}$ and $k \in S$, we let $C(S, k)$ be the optimal cost* of starting from city 1, visiting all the cities in S , and ending at city k . We begin by finding $C(S, k)$ for $|S|=1$, which is simply

$$C(\{k\}, k) = d_{1k} \quad \text{all } k=2,\dots,n$$

To calculate $C(S, k)$ for $|S| > 1$, we argue that the best way to accomplish our journey from 1 to all of S , ending at k , is to consider visiting m immediately before k , for all m , and looking up $C(S - \{k\}, m)$ in our preceding table. Thus:

$$C(S, k) = \min_{m \in S - \{k\}} [C(S - \{k\}, m) + d_{mk}]$$

This must be calculated for all sets S of a given size and for each possible city m in S . (We also must save the city m for which a minimum is achieved, so that we can reconstruct the optimal tour by backtracking.) If we count each value of $C(S, k)$ as one storage location, we need space equal to (Memory Required)

$$\sum_{k=1}^{n-1} k(COMBIN(n-1, k)) = (n-1)2^{n-2} = O(n2^n)$$

locations and number of additions and comparisons equal to (CPU Time)

$$\sum_{k=2}^{n-1} k(k-1)COMBIN(n-1, k) + (n-1) = (n-1)(n-2)2^{n-2} + (n-1) = O(n^2 2^n)$$

These are exponential functions of the problem size n , and may seem prohibitively large. But when we consider the fact that there are $(n-1)!$ distinct tours in a naive enumeration, we see that in fact this approach results in enormous savings. Since there is no algorithm known for the TSP that is better than exponential, the dynamic programming approach cannot be dismissed out of hand, although branch-and-bound algorithms have proven more effective in this application. Practically, the algorithm is implemented easily for up to 15 nodes.

I do not present the TSPdyn in C++ Programming Language in the web version of my thesis.

The output of TSPdyn when implemented to a pragmatic application

* TSPdyn assumes that the cost is linearly proportional to the distance sailed.

Suppose we have the following matrix of distances (or costs):

Pod/Pol	1	2	3	4	5	6	7	8	9	10
1	0	8	5	9	12	14	12	16	17	22
2	8	0	9	15	17	8	11	18	14	22
3	5	9	0	7	9	11	7	12	12	17
4	9	15	7	0	3	17	10	7	15	18
5	12	17	9	3	0	8	10	6	15	15
6	14	8	11	17	8	0	9	14	8	16
7	12	11	7	10	10	9	0	8	6	11
8	16	18	12	7	6	14	8	0	11	11
9	17	14	12	15	15	8	6	11	0	10
10	22	22	17	18	15	16	11	11	10	0

These data are contained in a different folder and are read from there.

The output of TSPdyn is (exactly as it comes out on MS-DOS screen):

Number of nodes : 10
Costs file : d2.dat

Costs table :

```
node  1  2  3  4  5  6  7  8  9 10
-----
1 |  0  8  5  9 12 14 12 16 17 22
2 |  8  0  9 15 17  8 11 18 14 22
3 |  5  9  0  7  9 11  7 12 12 17
4 |  9 15  7  0  3 17 10  7 15 18
5 | 12 17  9  3  0  8 10  6 15 15
6 | 14  8 11 17  8  0  9 14  8 16
7 | 12 11  7 10 10  9  0  8  6 11
```

8	16	18	12	7	6	14	8	0	11	11
9	17	14	12	15	15	8	6	11	0	10
10	22	22	17	18	15	16	11	11	10	0

Path: 2 6 5 4 8 10 9 7 3 1

The above path sequence gives optimum cost: 73

A Comparison with the results from other optimization techniques

The same application has been solved via three heuristic techniques: The Nearest-Neighbor, The Closest Insertion Heuristic and The Geometric Heuristic. They were solved by hand.

Their results are presented below:

The Nearest-Neighbor Heuristic	95
The Closest Insertion Heuristic	79
The Geometric Heuristic	73

TSPdyn via Dynamic Programming 73

In terms of results we realize the efficacy of this algorithm. A comparison of computing time would be also interesting but it has not taken place

To continue, the output of TSPdyn will give us the optimal port sequencing in each route from a distance, that reflects also cost, minimization point of view. However, this does not mean that we should espouse its results insofar as we have to take into account the

decision-maker's marketing considerations as well as precedence constraints associated with various port pairs. It is highly possible that practically the precedence constraints along with the marketing considerations will be the driving forces that will determine the sequence of ports in each route. TSPdyn results do not necessarily lead to the minimization of transit time per box, neither to the minimization to the intermediate stops for each box.

We can compare the routes that are finally determined, if different from the output of TSPdyn, by means of the following formulas:

$$ERD_r = \frac{d_r - d_{TSP_r}}{d_{TSP_r}} \times 100\%, \text{ for each } r = 1, \dots, R$$

where:

ERD_r is a numerical factor showing us how “profligate” in terms of sailing distance is the route r . ($ERD_r \geq 0$, $ERD_r = 0$ in the case that the final route is the output of TSPdyn)

d_r is the total sailing distance on (actual) route r

d_{TSP_r} is the total sailing distance of the TSP output route

If we want to acquire a quantitative sense of the distance “profligacy” in all R routes together, this can simply be the average of ERD_r :

$$\overline{ERD} = \frac{1}{R} \times \sum_{r=1}^R \frac{d_r - d_{TSP_r}}{d_{TSP_r}} \times 100\%$$

Optionally, the calculation of the average intermediate port stops for each box and of the average transit time per box would provide us (in quantitative terms) with some supplementary information of the network's efficiency.

Once the set of routes and the sequence of ports in each route are determined, an analysis will follow to determine the minimum required size (capacity) of the ships that can be allocated to that trade.

5.4. Deciding Frequency of service and re- Routing

We make the assumption that the cargo is generated evenly throughout the year.

5.4.1. Amount of cargo moved

We make the additional assumption that the cargo is generated evenly throughout the year. From a (given) three-dimensional matrix Q (TEUs) representing the amounts of cargo (TEUs) to be moved per year from port i to port j on route r , the amounts of cargo to be loaded or unloaded in every port are:

$$Q_{ir} = \sum_{j=1}^{I_r} [Q_{ijr} + Q_{jir}] \quad (1)$$

where:

Q_{ir} is the amount of cargo to be moved (loaded or unloaded) per annum, by all ships at port i on route r

Q_{ijr} is the amount of cargo to be carried per annum from port i to port j on route r

Q_{jir} is the amount of cargo to be carried per annum from port j to port i on route r

I_r is the number of ports on route r

The targeted number of voyages per year defines the amount of cargo that has to be loaded and unloaded per call (i.e. per voyage) at each port, as follows:

$$q_{ir} = Q_{ir} [F_r / 365] \quad (2)$$

where:

q_{ir} is the amount of cargo to be unloaded and loaded at the i -th port of route r .

F_r is the frequency of service, as defined before; the term $[F_r / 365]$ is the inverse of

the number of voyages per year in route r .

5.4.2. *Vessel cargo levels*

The present model assumes that cargo may be carried between any pair of ports on a given route. Of course, once a given port sequence in a route is established, the possible origin-destination couples are also defined; (on the other hand, that port sequence is established taking into consideration the normal flow of cargo); for instance, if ports 1 and 2 are located in Spain (Valencia, Barcelona), ports 3 and 4 in Eastern Mediterranean (Lattakia, Damietta), and the port sequence is 1-2-3-4, the ships in this route should not pick cargo at port 2 for port 1 (assuming that the frequency may not be changed), because it will imply the carriage of that cargo to Eastern Mediterranean and then back to Spain, causing delays to the customers (excessive transit time), loss of cargo space and additional costs to the ship operator.

One important component of the present model is the calculation of the “cargo level” of our ship for each one of the ij legs in a route, i.e. the amount of cargo remaining on board in those legs. These cargos will suggest an optimal frequency of service, from the capacity utilization point of view only. This analysis, together with marketing considerations will be the basic information that the shipping company will use for the establishment of the service frequencies.

First, we assume that there is only one round voyage per year in each route and we find the highest loading level in each one of them. With that information, we can find the optimal ship capacity for a given number of voyages per year or the optimal number of voyages per year for a given ship capacity. The number of voyages per year is just 365 divided by the value of the frequency of service.

Therefore:

$$L_{ijr} = \sum_{f=1}^i \sum_{g=j}^f Q_{fgr} \quad (\text{for } i = I_r)$$

$$L_{ijr} = \sum_{f=j}^{I_r} \sum_{g=j}^f Q_{fgr} + \sum_{f=1}^i \sum_{g=1}^f Q_{fgr} + \sum_{f=1}^i \sum_{g=j}^{I_r} Q_{fgr} \quad (\text{for } i \neq I_r) \quad (3)$$

$$L_r = \max L_{ijr}$$

όπου:

L_{ijr} is the amount of cargo on board for a ship sailing from port i to port j on route r,
for the case of one voyage per year (a port is serviced every 365 days)

L_r is the amount of cargo in the most heavily loaded leg

Q_{fgr} is the amount of cargo to be carried per year from port f to port g on route r

The minimum required capacity of ships that are to operate on route r is:

$$RC_r = L_r / (365 / F_r) \quad (4)$$

where F_r is the established frequency of service.

On the other hand, if ships of type k with given capacity V_k are assigned to route r, then the minimum required number of voyages per year in that route is:

$$RV_r = L_r / V_k \quad (5)$$

and the corresponding value of frequency of service is:

$$F_r = 365 / RV_r \quad (6)$$

Clev is a computer program that calculate straightforwardly all the above formulations regarding cargo levels, frequencies of service and minimum required capacities. It is written in C++ Programming Language. The compiler used is the “Borland C++ Compiler 5.5”.

I do not present the C++ program in this version of my thesis.

An application of clev:

We have two routes which consist of 6 ports each and with the input and output as follows.

Here the calculations have been made in MS Excel

6							
1	2	3	4	5	6		
0	5	6	6	7	8	L12	89
34	0	6	6	7	8	L23	125
34	5	0	6	7	8	L34	155
34	5	6	0	7	8	L45	185
34	5	6	6	0	8	L56	209
34	5	6	6	7	0	L61	227
						L1	227
Fr	7.78					RC1	4.838521
Vk	6					RV1	37.83333
						F1	9.647577
6							
1	2	3	4	5	6		
0	37	98	6	42	8	L12	740
3	0	32	123	43	18	L23	815
45	29	0	21	150	8	L34	718
12	19	18	0	123	50	L45	580
34	54	91	111	0	21	L56	521
123	5	111	99	12	0	L61	766
						L2	815
Fr	5.111					RC2	11.41223
Vk	10					RV2	81.5
						F2	4.478528

While here is the output of cleav exactly as it comes out on MS-DOS screen. (the reader can check the correspondence with the previous MS-Excel output)

Number of routes : 2
Cargo loads file : cargo4.dat
Frequency of service : 1
Ship Capacity : 1

Cargo requirements :

route 1

0	5	6	6	7	8
34	0	6	6	7	8
34	5	0	6	7	8
34	5	6	0	7	8
34	5	6	6	0	8
34	5	6	6	7	0

route 2

0	37	98	6	42	8
3	0	32	123	43	18
45	29	0	21	150	8
12	19	18	0	123	50
34	54	91	111	0	21
123	5	111	99	12	0

Cargo levels:

route 1

0	89	0	0	0	0
0	0	125	0	0	0
0	0	0	155	0	0
0	0	0	0	185	0
0	0	0	0	0	209
227	0	0	0	0	0

route 2

0	740	0	0	0	0
0	0	815	0	0	0
0	0	0	718	0	0
0	0	0	0	580	0
0	0	0	0	0	521
766	0	0	0	0	0

Maximum cargo level per route:

1 227

2 815

Minimum required capacity per route:

1 7.78 4.83852

2 5.11 11.41

Corresponding Frequency of service per route:

1 6 9.64758

2 10 4.47853

5.4.3. Tools for fleet re-routing and network efficiency evaluation

Graphs of (4) showing the required ship capacity, RC_r versus frequency of service, F_r , are very useful for visualizing the frequency-capacity tradeoff in the different routes. (See figure 5.2)

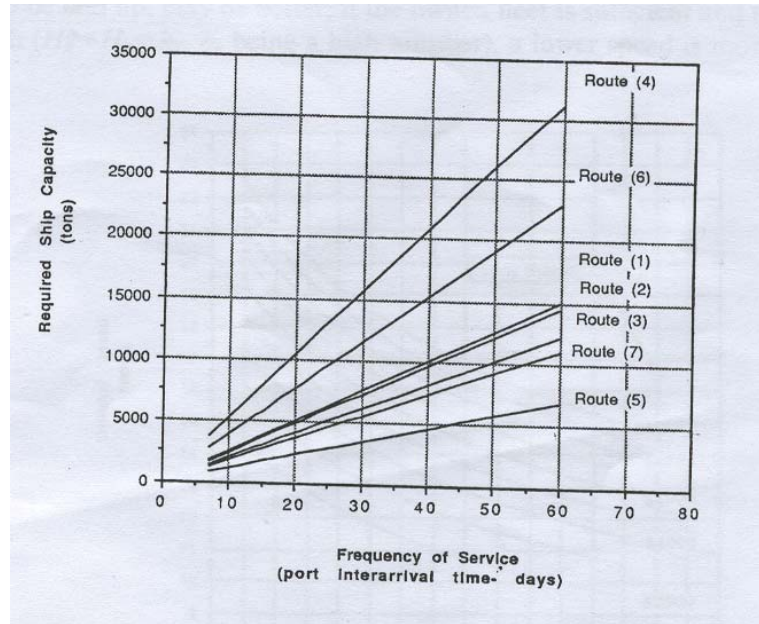


Figure 5.2. Ship Capacity vs. Frequency of Service

Graphs of (3) showing the loading condition of the ships in the various legs of a specific route (load levels vs. cumulative distance) provide an insight to the utilization of the ships and provide hints for minor routing or frequency of service modifications. (See figure 5.3) At this point the fixing of the frequency of service is required in order to avoid non-linear programming afterwards.

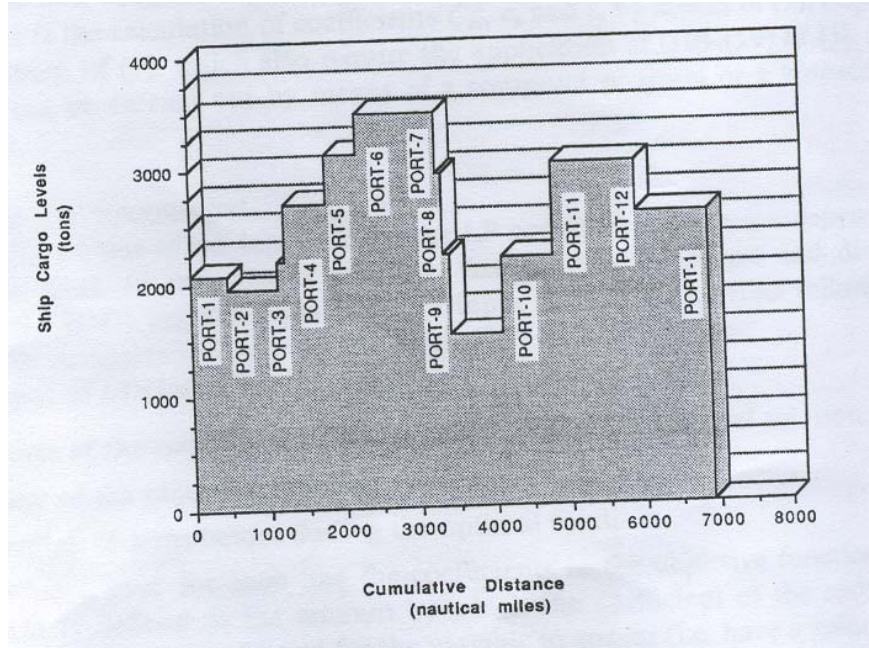


Figure 5.3. Ship cargo levels

Suppose we have pre-allocated certain ships to certain routes. The initial utilization of each ship operating on a certain route can be further examined via the two following formulas: (we will re-calculate the following formulas after the allocation of ships to routes at section 5 of this report.)

$$\overline{ASU}_{(k,r)} = \frac{\sum_{i=1}^{I_r} L_{i(i+1)r}}{I_r \times CAP_k} \quad (\text{note: for } i=I_r, i+1 \text{ returns } 1)$$

where \overline{ASU} is simplified Average Ship Utilization, and

$$\overline{ASU}'_{(k,r)} = \frac{\sum_{i=1}^{I_r} L_{i(i+1)r} * d_{i(i+1)r}}{\sum_{i=1}^{I_r} d_{i(i+1)r} \times CAP_k} \quad (\text{note: for } i=I_r, i+1 \text{ returns } 1)$$

where \overline{ASU}' is leveled Average Ship Utilization, and

$$\sum_{i=1}^{I_r} d_{i(i+1)r} \quad \text{is the total sailing distance of route } r;$$

CAP_k is the capacity (in TEUs) of ship k , which operates in route r .

The values of \overline{ASU} and \overline{ASU}' belong to the closed space $[0,1]$.

0 reflects a totally empty ship throughout the voyage and 1 a totally full-loaded ship during the voyage.

Now, the utilization of the fleet can be examined via the following formulas:

$$\overline{AFU} = 1/K * \sum_{k=1}^K \overline{ASU}_k$$

$$\overline{AFU}' = 1/K * \sum_{k=1}^K \overline{ASU}_k'$$

$$\overline{\overline{AFU}} = \sum_{k=1}^K (\overline{ASU}_k' \times CAP_k) \times \frac{1}{\sum_{k=1}^K CAP_k}$$

where:

\overline{AFU} is the simplified Average Fleet Utilization

\overline{AFU}' is the single-stage leveled Average Fleet Utilization

$\overline{\overline{AFU}}$ is the two-stage leveled Average Fleet Utilization

K is the number of ships (attention, here subscript k denotes number of ships and not number of ship types)

The values of \overline{AFU} , \overline{AFU}' and $\overline{\overline{AFU}}$ belong to the closed space $[0,1]$.

0 reflects a totally empty fleet in all routes and 1 a totally full-loaded fleet throughout all the voyages in all routes.

My understanding is that from a vehicle routing perspective it would be an omission not to take into account the effects of specific route characteristics on a certain ship's cost-size evaluation. The following formula reflects the rather convoluted relationship among ship size, operating route and cost.

$$ES_{kr} = \frac{C_{kr}'}{V_k}$$

where:

ES_{kr} is the economies of scale factor for a k type ship operating on route r (\$/TEU)

C'_{kr} are the (total) annual operating costs of a type k ship in route r (for its exact calculation please refer to the appendix)

V_k is the cargo capacity of ship k (in TEUs)

At this point the decision-maker should reevaluate the routes and the sequence of ports in each one. The routing & scheduling of the fleet must be finalized before embarking on the next optimization stage. We deem that the aforescribed tools could assist the decision-maker in reaching effectively this decision.

5.5. Allocation of Ships to Routes

Our model starts from a given and fixed set of routes with a defined sequence of ports and other fixed characteristics.

The output of the optimization model should include the following:

- (a) allocation of the owned ships to the routes;
- (b) number and type of ships to charter-in and their allocation to certain routes (if any);
- (c) whether to lay-up ships of which type and for how long (at this point the possibility of chartering out or scrapping should be considered by the shipping company);
- (d) optimal value of objective function, i.e. total annual operating costs (see below).

The objective function is the total annual operating costs but we also remind that our network already has a really remarkable advantage: uniformity of schedules, that is, all ships in the same routes have similar schedules.

The following Pure Integer Programming model is proposed. (Powell and Perakis (1997))

5.5.1. Decision variables

The decision variables in the model are the following:

N_{kr} the number of type k ships operating on route r

Y_k the number lay-up days per year of a type k ship

for $k=1, \dots, K$ and $r=1, \dots, R$ K is the number of ships or ship types και R is the number of routes.

5.5.2. Objective function

The objective function in the model minimizes the sum of the operating costs and the lay-up costs. The objective function in terms of the decision variables is:

$$\sum_{k=1}^K \sum_{r=1}^R C'_{kr} N_{kr} + \sum_{k=1}^K Y_k e_k \quad (2)'$$

where:

C'_{kr} operating costs of a type k ship on route r

e_k daily lay-up cost for a type k ship

5.5.3. Constraints

5.5.3.1. Ships Availability

The maximum number of type k ships operating cannot be greater than the maximum number of ships of type k available. Hence:

$$\sum_{r=1}^R N_{kr} \leq N_k^{\max} \text{ for each type } k \text{ ship} \quad (3)'$$

where: N_k^{\max} maximum number of type k ships available

5.5.3.2. Service Frequency

Service frequency is the driving force in liner shipping. With all rates being set by conferences, the main product differentiation is on service. To ensure that minimum service frequencies are met, the following constraint is included:

$$\sum_{k=1}^K t'_{kr} N_{kr} \geq M_r \quad \text{for all } r \quad (4)'$$

where $t'_{kr} = T_k / t_{kr}$

(5)'

t_{kr} voyage time of type k ship on route r

t'_{kr} yearly voyages of a type k ship on route r

T_k shipping season for a type k ship

M_r number of voyages required per year on route r

By finding the highest load level for any given leg of route r and comparing this with given ship capacity, we find minimum voyages per year for a specific route.

5.5.3.3. Route/Ship incompatibility

Some ships may be unable to operate on a given route due to cargo constraints, government regulations, and/or environmental constraints. It is necessary to eliminate these ships from the model. Hence:

$$N_{kr} = 0 \text{ for given } (k,r) \text{ pairs} \quad (6)'$$

5.5.3.4. Lay-up time

The lay-up time is equal to the time a ship is not operating during the year. This includes dry docking and repair time..

$$Y_k = 365 N_k^{\max} - T_k \sum_{r=1}^R N_{kr} \quad (7)'$$

5.5.3.5. Non-negativity

Obviously, the decision variables must be non-negative.

$$N_{kr} \geq 0. \quad (8)'$$

Based on the aforesaid formulation of Powell and Perakis we have written in LINGO programming language the following formulation:

I do not present it in the present version of my thesis.

Fleet Deployment: A pragmatic application

The following example regards the fleet deployment for Flota Mercante Grancolombiana (FMG) liner shipping company. Since we have not collected the *all* the data required from a shipping company, we take advantage of the data from the paper of Jaramillo and Perakis. The fleet consists of six types of owned ships and five types of chartered ships (one long-term charter and four short-term charters). The per ship cost coefficients C'_{kr} and the per ship time coefficients t'_{kr} are contained in the following table.

Calculation of C'kr and t'kr:			$C'kr = Ckr \cdot Tk / tkr$ $t'kr = Tk / tkr$ Ckr in \$, C'kr in \$K													
			Route													
		r	1		2		3		5		6		7		8	
		k	Ckr	tkr	Ckr	tkr	Ckr	tkr	Ckr	tkr	Ckr	tkr	Ckr	tkr	Ckr	tkr
Ship Type																
Owned																
Type Alpad	1	591830.13	40.65	462725.36	31.49	536638.18	37.92	1103799.10	76.14	767132.67	52.91	1117575.43	77.39	484063.62	33.16	
Type Ciarm	2	591314.80	38.90	462404.17	30.04	526655.10	35.71	1096674.87	72.71	762837.77	50.01	1104410.37	72.63	480381.09	31.82	
Type Cisan	3	589413.52	38.90	460502.89	30.04	526655.10	35.71	1094438.07	72.71	760936.49	50.01	1102509.09	72.63	480381.09	31.82	
Type Ciman	4	484151.35	39.45	374588.87	30.49	451804.68	36.40	915945.73	73.78	632262.36	50.91	926810.42	74.12	411981.91	32.24	
Type Rimug	5	468319.94	39.45	361409.48	30.49	439932.32	36.40	891221.16	73.78	610345.49	50.91	896180.81	74.12	401500.49	32.24	
Type Cibun	6	553962.39	38.90	431317.22	30.04	500614.50	35.71	1040099.50	72.71	713145.00	50.01	1035988.37	72.63	456899.55	31.82	
Chartered																
Type Golfo Ch.	7	590998.99	38.90	462404.17	30.04	526655.10	35.71	1096674.87	72.71	762837.77	50.01	1104410.37	72.63	480381.09	31.82	
Type Meghan A.	8	473054.37	38.90	368127.80	30.04	429189.14	35.71	891023.66	72.71	610025.25	50.01	887621.46	72.63	393252.98	31.82	
Type Monsun	9	574604.16	38.90	448615.38	30.04	513312.25	35.71	1074983.62	72.71	736313.85	50.01	1066482.99	72.63	468509.34	31.82	
Type Metesif	10	350056.72	42.02	268725.28	32.62	340700.08	39.65	685807.39	78.84	455894.78	55.19	678133.95	81.13	308139.21	34.22	
Type Angeliki	11	453408.52	42.02	357858.41	32.62	403795.00	39.65	944481.11	78.84	588499.92	55.19	855803.23	81.13	362963.94	34.22	
Shipping Season (Tk) 345																
			Route													
		r	1		2		3		5		6		7		8	
		k	Ckr	tkr	Ckr	tkr	Ckr	tkr	Ckr	tkr	Ckr	tkr	Ckr	tkr	Ckr	tkr
Ship Type																
Owned																
Type Alpad	1	5023.42	8.49	5070.06	10.96	4882.75	9.10	5001.43	4.53	5001.99	6.52	4982.07	4.46	5035.72	10.40	
Type Ciarm	2	5244.32	8.87	5310.45	11.48	5087.72	9.66	5203.62	4.74	5262.97	6.90	5246.11	4.75	5207.87	10.84	
Type Cisan	3	5227.46	8.87	5288.62	11.48	5087.72	9.66	5193.00	4.74	5249.85	6.90	5237.08	4.75	5207.87	10.84	
Type Ciman	4	4234.50	8.75	4238.18	11.31	4282.03	9.48	4282.92	4.68	4284.32	6.78	4314.11	4.65	4408.33	10.70	
Type Rimug	5	4096.04	8.75	4089.07	11.31	4169.50	9.48	4167.31	4.68	4135.81	6.78	4171.54	4.65	4296.18	10.70	
Type Cibun	6	4913.04	8.87	4953.44	11.48	4836.15	9.66	4935.17	4.74	4920.13	6.90	4921.09	4.75	4953.30	10.84	
Chartered																
Type Golfo Ch.	7	5241.52	8.87	5310.45	11.48	5087.72	9.66	5203.62	4.74	5262.97	6.90	5246.11	4.75	5207.87	10.84	
Type Meghan A.	8	4195.48	8.87	4227.74	11.48	4146.15	9.66	4227.82	4.74	4208.69	6.90	4216.33	4.75	4263.30	10.84	
Type Monsun	9	5096.11	8.87	5152.10	11.48	4958.82	9.66	5100.69	4.74	5079.98	6.90	5065.95	4.75	5079.16	10.84	
Type Metesif	10	2874.25	8.21	2841.85	10.58	2964.52	8.70	3001.21	4.38	2849.66	6.25	2883.69	4.25	3106.93	10.08	
Type Angeliki	11	3722.86	8.21	3784.46	10.58	3513.53	8.70	3695.59	4.38	3678.53	6.25	3639.21	4.25	3659.72	10.08	

Table 5.1 Calculation of cost and times coefficients

In Lingo programming language we have the following model.

I do not present it in the present version of my thesis.

Output:

An excerpt from the output of lingo is the following:

Global optimal solution found.

Objective value: 91831.00

Extended solver steps: 1046

Total solver iterations: 3793

N_LAYUP_DAYS(1) 120.0000 0.000000

N_LAYUP_DAYS(2) 40.00000 0.000000

N_LAYUP_DAYS(3)	60.00000	0.000000
N_LAYUP_DAYS(4)	20.00000	0.000000
N_LAYUP_DAYS(5)	20.00000	0.000000
N_LAYUP_DAYS(6)	20.00000	0.000000
N_LAYUP_DAYS(7)	40.00000	0.000000
N_LAYUP_DAYS(8)	730.0000	0.000000
N_LAYUP_DAYS(9)	730.0000	0.000000
N_LAYUP_DAYS(10)	40.00000	0.000000
N_LAYUP_DAYS(11)	385.0000	0.000000
N_SHIPS_ON_ROUTE(1, 2)	0.000000	1930.500
N_SHIPS_ON_ROUTE(1, 3)	0.000000	1742.500
N_SHIPS_ON_ROUTE(1, 4)	3.000000	1861.500
N_SHIPS_ON_ROUTE(1, 5)	0.000000	1861.500
N_SHIPS_ON_ROUTE(1, 6)	0.000000	1842.500
N_SHIPS_ON_ROUTE(1, 7)	0.000000	1895.500
N_SHIPS_ON_ROUTE(2, 1)	0.000000	2139.000
N_SHIPS_ON_ROUTE(2, 2)	0.000000	2205.000
N_SHIPS_ON_ROUTE(2, 3)	0.000000	1982.000
N_SHIPS_ON_ROUTE(2, 4)	0.000000	2098.000
N_SHIPS_ON_ROUTE(2, 5)	0.000000	2157.000
N_SHIPS_ON_ROUTE(2, 6)	1.000000	2141.000
N_SHIPS_ON_ROUTE(2, 7)	1.000000	2102.000
N_SHIPS_ON_ROUTE(3, 1)	0.000000	2122.000
N_SHIPS_ON_ROUTE(3, 2)	0.000000	2183.000
N_SHIPS_ON_ROUTE(3, 3)	0.000000	1982.000
N_SHIPS_ON_ROUTE(3, 4)	1.000000	2088.000
N_SHIPS_ON_ROUTE(3, 5)	0.000000	2144.000
N_SHIPS_ON_ROUTE(3, 6)	2.000000	2132.000
N_SHIPS_ON_ROUTE(3, 7)	0.000000	2102.000
N_SHIPS_ON_ROUTE(4, 1)	0.000000	0.000000
N_SHIPS_ON_ROUTE(4, 2)	1.000000	1685.000

N_SHIPS_ON_ROUTE(4, 3)	0.000000	0.000000
N_SHIPS_ON_ROUTE(4, 4)	0.000000	0.000000
N_SHIPS_ON_ROUTE(4, 5)	0.000000	1731.000
N_SHIPS_ON_ROUTE(4, 6)	0.000000	0.000000
N_SHIPS_ON_ROUTE(4, 7)	0.000000	1855.000
N_SHIPS_ON_ROUTE(5, 1)	0.000000	0.000000
N_SHIPS_ON_ROUTE(5, 2)	1.000000	1570.500
N_SHIPS_ON_ROUTE(5, 3)	0.000000	0.000000
N_SHIPS_ON_ROUTE(5, 4)	0.000000	0.000000
N_SHIPS_ON_ROUTE(5, 5)	0.000000	1616.500
N_SHIPS_ON_ROUTE(5, 6)	0.000000	0.000000
N_SHIPS_ON_ROUTE(5, 7)	0.000000	1777.500
N_SHIPS_ON_ROUTE(6, 1)	0.000000	1842.500
N_SHIPS_ON_ROUTE(6, 2)	0.000000	1882.500
N_SHIPS_ON_ROUTE(6, 3)	0.000000	1765.500
N_SHIPS_ON_ROUTE(6, 4)	0.000000	1864.500
N_SHIPS_ON_ROUTE(6, 5)	0.000000	1849.500
N_SHIPS_ON_ROUTE(6, 6)	1.000000	1850.500
N_SHIPS_ON_ROUTE(6, 7)	0.000000	1882.500
N_SHIPS_ON_ROUTE(7, 1)	0.000000	2136.000
N_SHIPS_ON_ROUTE(7, 2)	0.000000	2205.000
N_SHIPS_ON_ROUTE(7, 3)	2.000000	1982.000
N_SHIPS_ON_ROUTE(7, 4)	0.000000	2098.000
N_SHIPS_ON_ROUTE(7, 5)	0.000000	2157.000
N_SHIPS_ON_ROUTE(7, 6)	0.000000	2141.000
N_SHIPS_ON_ROUTE(7, 7)	0.000000	0.000000
N_SHIPS_ON_ROUTE(8, 1)	0.000000	4195.000
N_SHIPS_ON_ROUTE(8, 2)	0.000000	4227.000
N_SHIPS_ON_ROUTE(8, 3)	0.000000	4146.000
N_SHIPS_ON_ROUTE(8, 4)	0.000000	4227.000
N_SHIPS_ON_ROUTE(8, 5)	0.000000	4208.000

N_SHIPS_ON_ROUTE(8, 6)	0.000000	4216.000
N_SHIPS_ON_ROUTE(8, 7)	0.000000	0.000000
N_SHIPS_ON_ROUTE(9, 1)	0.000000	5096.000
N_SHIPS_ON_ROUTE(9, 2)	0.000000	5152.000
N_SHIPS_ON_ROUTE(9, 3)	0.000000	4958.000
N_SHIPS_ON_ROUTE(9, 4)	0.000000	5100.000
N_SHIPS_ON_ROUTE(9, 5)	0.000000	5079.000
N_SHIPS_ON_ROUTE(9, 6)	0.000000	5065.000
N_SHIPS_ON_ROUTE(9, 7)	0.000000	0.000000
N_SHIPS_ON_ROUTE(10, 1)	0.000000	0.000000
N_SHIPS_ON_ROUTE(10, 2)	0.000000	0.000000
N_SHIPS_ON_ROUTE(10, 3)	0.000000	0.000000
N_SHIPS_ON_ROUTE(10, 4)	0.000000	0.000000
N_SHIPS_ON_ROUTE(10, 5)	2.000000	2849.000
N_SHIPS_ON_ROUTE(10, 6)	0.000000	0.000000
N_SHIPS_ON_ROUTE(10, 7)	0.000000	0.000000
N_SHIPS_ON_ROUTE(11, 1)	0.000000	3722.000
N_SHIPS_ON_ROUTE(11, 2)	0.000000	2784.000
N_SHIPS_ON_ROUTE(11, 3)	0.000000	3513.000
N_SHIPS_ON_ROUTE(11, 4)	1.000000	3695.000
N_SHIPS_ON_ROUTE(11, 5)	0.000000	3678.000
N_SHIPS_ON_ROUTE(11, 6)	0.000000	3639.000
N_SHIPS_ON_ROUTE(11, 7)	0.000000	0.000000

For a complete list of LINGO output please refer to the attachment.

As it can be seen from the LINGO solver status the solver that is used is the Brach-and-Bound. We had 59 integer variables and the elapsed run time was 1 sec in a Pentium M Processor 1.8 GHz.

LINGO Solver Status [FDP]

Solver Status Model Class: ILP State: Global Opt Objective: 91831 Infeasibility: 0 Iterations: 3793		Variables Total: 224 Nonlinear: 0 Integers: 59	
Extended Solver Status Solver Type: B-and-B Best Obj: 91831 Obj Bound: 91831 Steps: 1046 Active: 0		Constraints Total: 118 Nonlinear: 0	
		Nonzeros Total: 324 Nonlinear: 0	
		Generator Memory Used (K) 65	
		Elapsed Runtime (hh:mm:ss) 00:00:01	

Update Interval: 2

The initial allocation of the ships is described in the table hereinbelow.

		Route							Total
Ship type		1	2	3	4	5	6	7	
Owned	1	3			3				6
	2						2		2
	3				2		1		3
	4							1	1
	5		1						1
	6			1					1
	7								
Chartered	8		1	1			1		3
	9								0
	10								0
	11					2			2
	Total	3	2	2	5	2	4	1	19

Table 5.1 Initial Ship allocation

The IP optimal allocation is given in the table hereinafter. The minimum objective function yields a total operating cost of \$91 831 000. This is compared with \$ 93 148 000 for the current allocation. This corresponds to a reduction in total operational costs of 1.4% (a saving of \$1 317 000 per year).

Analyzing the resulting allocation shows that all owned ships (k=1 to 6) and the long-term charter k=7 are in use for the entire shipping season. This is due to the high lay-up costs associated with these ship type. None of type ship 9 is allocated. This ship has the highest operating cost of any of the short-term charters.

	Ship type	Route							Total
		1	2	3	4	5	6	7	
Owned	1	3			3				6
	2			1				1	2
	3				1		2		3
	4		1						1
	5		1						1
	6						1		1
Chartered	7			1			1		2
	8								0
	9								0
	10					2			2
	11				1				1
Total		3	2	2	5	2	4	1	19

Table 5.2 Resultant Ship allocation

Since service is a priority in liner shipping it is necessary to meet the target frequencies. The IP optimization model ensures that all target frequencies are met. Routes 1,3,5 exactly meet the target frequencies while on routes 2,4,6, and 7 the frequency is improved. The improvement ranges from 1.3 days to 3.3 days. (Please see below.)

	Route						
	1	2	3	4	5	6	7
Target frequency	14	14	21	15	30	23	35
IP model difference	14	10.7	21	12.9	29.2	20.4	33.7
	0.0	-3.3	0.0	-2.1	0.0	-2.6	-1.3

Table 5.3 Comparison of Frequencies

5.6. Transshipment

Once the above stages have been successfully applied we will have an efficient network from the routing and scheduling point of view as well as an optimal one from the fleet deployment point of view. At this point, we will have to discuss if a transshipment evaluation is necessitated. The transshipment option will deal with regional elementary networks. Its ambition will be to serve special needs or regional network inefficiencies. The planning problem consists of choosing which of a possible set of predefined routes to use. My suggestion is that it must be solved through the identification and comparison of any interesting possible alternatives in the network as it appears after the application of the Routing, Scheduling and Deployment methodology.

I think that a simulation model work very well for this purpose insofar as simulation models are similar to gaming models except that all human decision-makers are removed from the modeling process. The model provides the means to evaluate the performance of a number of alternatives, supplied externally to the model by the decision-maker, without allowing for human interactions at intermediate stages of the model computation.

However, if someone wants adopt an optimization-based approach the following formulation is suggested (Christiansen et al. (2004)):

Containers are usually both faster and cheaper to load and unload than the general cargo that is stuffed in them. This means that containers are often loaded and unloaded several times between their origin and their final destination. One type of maritime transportation systems for containers is a so-called *hub and spoke network* or a *trunk line*

and feeder system. In such systems we have a trunk line operating between the major ports (hubs) and a system of feeder ships working in the geographical region around each hub port visited by the trunk line. The ports feeding containers to a hub are the spokes. Thus a container is typically loaded and unloaded three times. First a feeder ship transports the container from its initial loading port to a trunk line hub port. Then a trunk line ship transports the container to another trunk line hub port, and finally another feeder ship takes the container to its final unloading port.

Here we study a short-sea application of a feeder system around one trunk line hub port with a homogenous fleet of feeder ships. We model the transportation of containers between one hub port and a set of feeder ports (spokes) in one geographical region. Each container is either loaded or unloaded in the hub.

The demands both to and from a spoke port are assumed to increase with the number of visits in the port during the planning horizon. These demands are upper bounds on the number of containers available for transportation, but the shipping company is not obliged to satisfy the total demand.

The planning problem consists of choosing which of a possible huge set of predefined routes to use and how many voyages to sail along the chosen routes, while maximizing the net revenue. Figure 5.4 illustrates the problem with one hub and several spokes. The designed routes might be overlapping.

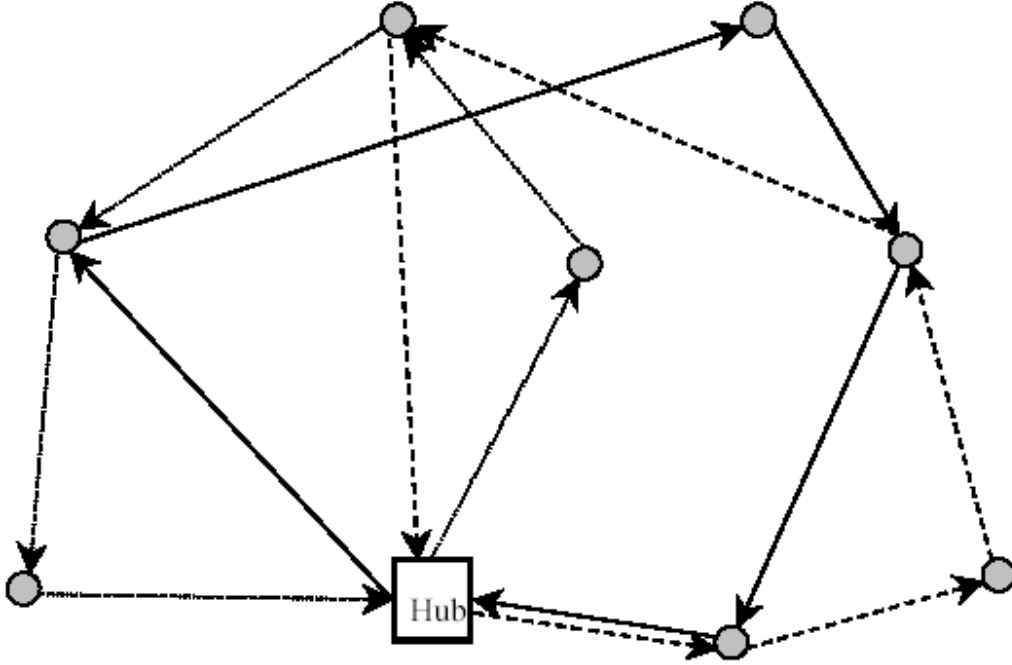


Figure 5.4: Regional hub-and-spoke networks

In the mathematical description of the problem, let R be the set of predefined routes indexed by r and N be the set of ports apart from the hub indexed by i . Further, let N_r be the set of ports apart from the hub visited on route r . The routes that visit port i are given by the set R_i . The ports called after port i on route r belong to the set F_{ir} and the ports called before and including port i on route r belong to the set G_{ir} . Let M be the set of possible calls at the same port during the planning horizon indexed by m .

We assume that there are fixed revenues, R_{Ui} and R_{Li} , for unloading and loading a container in port i . The cost consists of three parts. We call the fixed cost of operating a ship during the planning horizon C_F . The cost of sailing one voyage along route r is C_{Vr} and the cost of unloading (loading) one container in port i on route r is C_{Uir} (C_{Lir}). Since the fleet is homogenous and the unit costs are specified before we know the loading pattern along the routes, we will normally have C_{Uir} and C_{Lir} independent of r . The time each ship is available during the planning horizon is called the shipping season S . The sailing time for one voyage along route r is T_{Vr} and the capacity measured in number of

containers of a ship is Q . The demand is specified in the following way: D_{Uim} (D_{Lim}) is the incremental demand for unloading (loading) in port i when the number of visits at that port increases from $m-1$ to m .

In the mathematical formulation, we use the following types of variables. The integer variable s represents the number of ships in operation and $u_r, \forall r \in R$, represents the number of voyages along route r during the planning horizon. The number of containers unloaded and loaded in port i on route r during the planning horizon is given by p_{ir} and $q_{ir}, \forall r \in R, i \in N_r$, respectively. The integer number of calls at port i is $h_i, \forall i \in N$, and finally, the binary variable $g_{im}, \forall i \in N, m \in M$, is equal to 1 if port i is called at least m times during the planning period.

A liner network design model for a network with one hub and several spokes is as follows:

$$\max \left[\left(\sum_{r \in R} \sum_{i \in N_r} (R_{Ui} - C_{Uir}) p_{ir} \right) + \left(\sum_{r \in R} \sum_{i \in N_r} (R_{Li} - C_{Lir}) q_{ir} \right) - C_{Fs} - \sum_{r \in R} C_{Vr} U_r \right], \quad (5.1)$$

Subject to:

$$\sum_{r \in R} T_{Vr} u_r - S_s \leq 0, \quad (5.2)$$

$$\sum_{i \in N_r} p_{ir} - Q_{u_r} \leq 0, \quad \forall r \in R \quad (5.3)$$

$$\sum_{j \in G_{ir}} q_{jr} + \sum_{j \in F_{ir}} p_{jr} - Q_{u_r} \leq 0, \quad \forall r \in R, i \in N_r \quad (5.4)$$

$$\sum_{r \in R_i} u_r - h_i = 0, \quad \forall i \in N, \quad (5.5)$$

$$\sum_{m \in M} g_{im} - h_i = 0, \quad \forall i \in N, \quad (5.6)$$

$$g_{i(m-1)} - g_{im} \geq 0, \quad \forall i \in N, m \in M \quad (5.7)$$

$$\sum_{r \in R_i} p_{ir} - \sum_{m \in M} D_{Uim} g_{im} \leq 0, \quad \forall i \in N \quad (5.8)$$

$$\sum_{r \in R_i} q_{ir} - \sum_{m \in M} D_{Lim} g_{im} \leq 0, \forall i \in N \quad (5.9)$$

$$p_{ir}, q_{ir} \geq 0, \forall r \in R, i \in N_r \quad (5.10)$$

$$h_i, s, u_r \geq 0 \text{ _and_ integer}, \forall r \in R, i \in N \quad (5.11)$$

$$g_{im} \in \{0, 1\}, \forall i \in N, m \in M \quad (5.12)$$

The objective function (5.1) maximizes the net revenue over the planning horizon. We calculate the number of needed ships in (5.2) in a way that might be too simple. The constraints ensure that the total available sailing time for all the ships together is larger than the sum of the voyages' times. We have not verified that the available time of the ships can be split in such a manner that each ship can perform an integer number of voyages during the planning horizon. Constraints (5.3) - (5.4) take care of the capacity when the ships leave the hub and the spokes on the route. Constraints (5.5) - (5.6) use the number of voyages along the routes to calculate the number of calls at each port. The precedence constraints (5.7) for the g_{im} variables are not needed if the incremental increase in the demand diminishes with increasing number of calls. The numbers of containers unloaded and loaded in the ports are bounded by the demand constraints (5.8) - (5.9). Finally, the formulation involves binary, integer and non-negativity requirements on the variables in (5.10)-(5.12).

5.7 Travel Costs and Travel Times estimation subroutine

(Borrowed from Perakis and Jaramillo (1991))

The costs referred to in this work as operating costs include the following:

- (a) fuel costs (propulsion fuel, generator fuel);
- (b) daily running costs (they will be defined in a following section);
- (c) port charges;
 - (i) charges per call which do not depend on the time of stay (docking/undocking,

- pilotage, tugboat charges, port maintenance fees, etc.);
- (ii) charged per unit of time of stay at port (wharfage, anchorage, etc.);
- (d) canal fees.

Conversely, several important costs are excluded from the model as they are effectively fixed, as per our assumptions. Those costs are:

- (e) stevedoring loading/unloading costs;
- (f) agency fees;
- (g) commissions to cargo brokers;
- (h) communications billed by agents, most of them cargo related;
- (i) container rental and maintenance.

The above-listed costs are assumed given and constant in the present work because they depend on the types, amounts, origins and destinations of the cargoes carried. The overhead costs are also excluded.

Ship daily running costs

For the owned ships, this cost has typically the following components:

- (a) daily capital cost;
- (b) salaries and benefits of the crew;
- (b) maintenance and repair (labor and parts);
- (d) insurance of hull and machinery;
- (e) lubricants;
- (f) supplies and miscellaneous.

The daily capital cost (DCC) can be computed from the cash flows related to the purchase of the ship and the salvage value. The net present value (NPV) of those cash flows should be computed. If favorable financing is obtained for ship's acquisition, an adjusted (APV) should be computed. This APV is the result of subtracting the present value of the subsidized borrowing from the above NPV.

The annualized capital cost

$$ACC=APV/AF(y,t) \quad (7)$$

$$\text{where: } AF(y,t)=y^{-1}(1+y)^{-t}[(1+y)^t-1] \quad (8)$$

is a factor converting to present value an annuity of t years at an annual interest rate of y . The daily capital cost (DCC) is $ACC/365$. The DCC is added to the other costs per day in order to obtain the daily running costs H_k for ship k . For chartered vessels, H_k is simply the hire rate (for long term charter).

When ship k is laid-up for medium periods of time, some of the cost components of H_k may decline substantially (e.g. maintenance and lubricants); other costs may also decline, depending on the particular case; the type of labor contract with the crew will define how much crew reduction the company can make. This is usually the most important input in deciding to lay-up a ship. If crew can be reduced, food and other provisions can also be reduced. Insurance costs could be reduced, depending on the agreement between the shipping and the insurance companies; frequently the insurance policies require payments in advance covering long periods. In such cases, no cost reduction is obtained for laying up the ship, at least during the first year or semester after the lay-up takes place. The daily cost for the ship k while laid-up will be denoted in the present thesis by h_k .

Voyage costs

$$C'_{kr} = C_{kr} * T_k / t_{kr}$$

C'_{kr} operating costs of a type k ship on route r

C_{kr} is the operating costs per voyage for ship k on route r

T_k duration of shipping season for a type k ship

t_{kr} voyage time of a type k ship on route r

The total costs will be divided into costs at sea and costs in port. A “voyage” is defined as one round trip in one of the established routes.

$$C_{kr} = C_{kr}^p + C_{kr}^s \quad (9)$$

where:

C_{kr}^p is the operating costs at port per voyage for ship k on route r

C_{kr}^s is the operating costs at sea per voyage for ship k on route r

C_{kr} is the operating costs per voyage for ship k on route r

Costs at sea

$$C_{kr}^s = t_{kr}^s A_{kr}^s + m_{kr} + t_r^m H_k \quad (10)$$

t_{kr}^s is the sailing time of ship k, per voyage on route r (days)

A_{kr}^s is the operating costs per unit time at sea for ship k on route r
(\$/days)

m_{kr} is the canal fees per voyage for ship k on route r (\$/voyage)

t_r^m is the delay due to sailing in restricted waters, including waiting times (days)

The sailing time for ship k on route r is:

$$t_{kr}^s = d_r / (24 S_k) \quad (11)$$

d_r is the total sailing distance in route r (nautical miles)

S_k is the speed of vessel k (knots)

The costs at sea per day for vessel k on route r are:

$$A_{kr}^s = f_k p_r^f + g_k^s p_r^g + H_k \quad (12)$$

f_k is the consumption per unit time of propulsion fuel of ship k (tons/day)

p_r^f is the price of propulsion fuel on route r (\$/ton)

g_k^s is the consumption per unit time of fuel for electricity generation at sea of ship k

p_r^g is the price of fuel for electricity generation on route r (\$/ton).

Canal fees are usually established per gross register ton (GRT). The costs due to canal fees are therefore calculated as follows:

$$m_{kr} = cf_r RT_{kr} cn_r \quad (13)$$

where:

cf_r is the canal fee in route r (\$/register ton)

RT_{kr} gross register tonnage of ship k , for the canal in route r

cn_r is the number of canal crossings/voyage in route r

The delay caused by sailing in restricted waters (including canals and entries/departures to/from ports) and by waiting in queues before passing canals is

$$t_r^m = (d_r^m / 24) [(1/S_m) - (1/S_k)] + t_r^w \quad (14)$$

όπου:

d_r^m is the distance to be sailed in restricted waters in route r (nautical miles)

S_m is the average speed in restricted waters for all ships (knots)

t_r^w is the waiting time (at anchorage) due to canal queues per voyage in route r ,
for any ship (days)

$$t_r^w = cn_r cw_r \quad (15)$$

where:

cw_r is the waiting time per canal crossing on route r (days)

$$d_r^m = \sum_{i=1}^{I_r} d_{ir}^m + cn_r cd_r \quad (16)$$

µε:

d_{ir}^m is the distance of restricted waters sailing associated with port ir

cd_r is the canal sailing distance on route r

Costs at ports

We must consider both fixed (per call basis) and variable (per day or port stay) costs::

$$C_{kr}^p = \sum_{i=1}^{I_r} [t_{ir}^p A_{irk}^p + u_{irk}] \quad (17)$$

where:

C_{kr}^p are the operating costs at ports per voyage for the ship k on route r

t_{ir}^p is the time per call at port i of route r, can also be called time per voyage at that port
as we are denoting multiple calls per voyage to the same port as different ports

A_{irk}^p are the operating costs per unit time for vessel k at port I of route r (\$/day)

u_{irk} are the fixed costs per call at port I of route r for ship k.

Fixed costs per call u_{irk} include:

1. docking/undocking;
2. pilotage;
3. tugboat fees;
4. navigation aids maintenance fees (if any).

Depending on the port authority, there may be other costs to be included in this category. The rates are usually based on the ship's length, draught, deadweight tonnage, or GRT.

The time a ship spends in a port is proportional to the amount of cargo loaded and unloaded. We assume that there is only one type of cargo (for example, containers), therefore a single loading/unloading rate (here called "productivity") can be applied in order to find the time needed to move the cargo requirements. In addition, in order to be realistic, our model includes an allowance for inactive time in port; this is a time interval during which the ship is not performing its normal operations in port. Inactivities can be classified depending on their cause, as follows:

1. caused by the port authority
2. caused by the agent
3. caused by the shipper (as when waiting for a given cargo to arrive to the loading port)
4. caused by the shipping company.
5. other, e.g. bad weather.

The time at port is:

$$t_{ir}^p = q_{ir} / n_{ir} + w_{ir} \quad (18)$$

where:

q_{ir} amount of cargo to be unloaded and loaded per voyage at the i-th port of route r

n_{ir} is the productivity, or the rate of unloading or loading cargo at port I of route r (tons or containers per day)

w_{ir} is the allowance per call for inactive time at port i of route r (days), as defined above

For the cost per unit time of the ship while at port, the model includes the fuel cost, the ship's daily running cost and the variable port fees:

$$A_{irk}^p = g_k^p p_r^g + H_k + v_{irk} \quad (19)$$

g_k^p is the average fuel consumption at ports, mainly for electricity generation
(tons/day)

p_r^g is the price for electricity generation in route r (\$/ton)

v_{irk} is the variable port fees (per unit time) for ship k at port i of route r.

The variable port fees depend on the specific port; the most usual fees in this category are anchorage and wharfage charges. As with fixed or per call charges, the level of variable fees is established depending in the ship's length, draught, DWT, or GRT.

Daily *Lay-up costs*

The lay-up costs are:

$$e_k = h_k + g_{kl} p_t + D_k \quad (20)$$

where:

e_k is the total lay-up costs per day for ship k on route r (\$/day)

h_k daily running costs of the ship k while laid-up

g_{kl} is the generator fuel consumption in lay-up condition (ton/day)

p_t is the price of generator fuel at the lay-up location (\$/ton)

D_k is the additional daily cost of ship k while laid-up, including anchorage charges, transportation for the crew to and from the ship, etc.

Total voyage time

The total voyage time is the sum of the times at sea and at port plus any delay due to restricted operation:

$$t_{kr} = t_{kr}^s + t_r^p + t_r^m \quad (21)$$

where:

$$t_r^p = \sum_{i=1}^{I_r} t_{ir}^p$$

is the total time in port on route r (days), and the other terms have already been defined.

5.8. The revenue calculation

As it is already stated, the objective function should be the annual operating costs of a fleet of liners and no optimization procedure is needed based on the revenue function. The reason is that the minimization of the annual operating costs is equivalent to maximizing profits per unit time insofar as the cargo movement requirements are met, the freight rates are assumed fixed, and the revenue is, therefore, constant.

The classification of revenue

Before discussing revenue, something must be said about the way revenue is received. There are several different ways a shipowner can earn revenue, each of which brings a different distribution of risk between the shipowner and the charterer and a different apportionment of costs. The risks are shipping market risk, which concerns availability of cargo and the freight rate paid and operational risk arising from the ability of the ship to perform the transport. The costs are those discussed in the previous section. Each of the revenue arrangements deals with these items differently:

- *Voyage charter.* This system is used in the voyage charter market, the specialist bulk market and in a rather different way in the liner trades. The freight rate is paid per unit of cargo transported, for example \$20 per ton. Under this

arrangement, the shipowner generally pays all the costs, except possibly cargo handling, and is responsible both for managing the running of the ship and for the planning and execution of the voyage. He takes both the operational and the shipping market risk. If no cargo is available; if the ship breaks down; or if it has to wait for cargo he loses out.

- *Time charter.* The charter hire is specified as a fixed daily or monthly payment for the hire of the vessel, for example \$5,000 per day. Under this arrangement, the owner still takes the operational risk, since if the ship breaks down he does not get paid. The charterer pays fuel, port charges, stevedoring and other cargo-related costs. He takes the market risk, paying the agreed daily hire regardless of market conditions (unless the charter rate is linked to the market in some way).
- *Bareboat charter.* This is essentially a financial arrangement in which the charter hire only covers the financing cost of the ship. The owner finances the vessel and receives a charter payment to cover expenses. All operating costs, voyage costs and cargo-related costs are covered by the charterer, who takes both the operational and shipping market risk.

For simplicity the discussion in this chapter assumes that revenue is earned as a unit freight rate per ton mile of cargo carried.

Freight revenue and ship productivity

However, the basic revenue calculation involves two steps: first, determining how much cargo the vessel can carry in the financial period, measured in whatever units are appropriate (tons miles, cubic metres, etc.), and, second, establishing what price or freight rate the owner will receive per unit transported. In more technical terms, the revenue per deadweight of shipping capacity can be viewed as the product of the ship's productivity, measured in ton miles of cargo transported per annum, and the freight rate per ton mile, divided by the ship's deadweight, thus:

$$R_{tm} = \frac{P_{tm} \times FR_{tm}}{DWT_{tm}}$$

where: R = revenue per dwt per annum

P = productivity in ton miles of cargo per annum

FR = freight rate per ton mile of cargo transported

t = time period

m = ship type

The concept of a ship's 'productivity' is useful because it measures overall cargo-carrying performance, encompassing both operating performance in terms of speed and cargo deadweight and flexibility in terms of obtaining backhaul cargoes. For example, a combined carrier potentially has a much higher productivity than a tanker because it can carry a backhaul of dry cargo if one is available. The analysis of productivity can be carried further by subdividing into its component parts as follows:

$$P_{tm} = 24 \cdot S_{tm} \cdot LD_{tm} \cdot DWU_{tm}$$

where: S = average operating speed per hour

LD = loaded days at sea per annum

DWU = deadweight utilization

This definition states that ship productivity, measured in terms of ton miles of cargo transported in year t , is determined by the distance the vessel actually travels in 24 hours, the number of days it spends loaded at sea in a year, and the extent to which it travels with a full deadweight of cargo. By further examination of each of these components a precise definition of productivity can be obtained.

Maximizing loaded days at sea

A ship's time is divided between 'productive' loaded days at sea and unproductive loaded days spent in ballast, in port, or off hire. A change in any of these variables will affect the loaded days at sea, as follows;

$$LD_{tm} = 365 - OH_{tm} - DP_{tm} - BAL_{tm}$$

Where: LD = loaded days at sea

OH = days off hire per annum

DP = days in port per annum

BAL = days in ballast per annum

Days off hire reflect time spent for repairs, breakdowns, holidays, etc.

Port days depend upon the type of ship, the loading facilities available and the cargo being loaded. The more time the ship spends in port the less it spends carrying cargo.

Days spent in ballast is the third and most important determinant of loaded days at sea.

5.9 Data and their sequential application to the methodology

As described in the previous subchapter, the nature of the decision-support system is of a dynamic nature. As a result, there must be a close interaction between the decision-maker and the optimization process (or, better, the person responsible for the application of this procedure) for the successful implementation of the methodology.

The subchapter neither includes mathematical formulas, nor describes certain parts of the methodology. In order to be laconic, this report contains only the data needed in each stage of the optimization process. Due to the multi-stage nature of the process, the plain

citing of the data needed would not suffice. Conversely, it is my understanding that the decision-maker should know which data is needed in every single part of the optimization “chain”. For an analytical description of the methodology (including methodology philosophy, assumptions, and mathematical formulas among others) please refer to previous chapters.

Please note that the sequential application of real numbers is in full accordance with the sequence defined in the previous report titled “Description of the multi-stage liner-network optimization methodology”.

Everything that is between the $\{\}$ symbols refers to the same autonomous optimization stage. Everything that is between \langle, \rangle symbols refers to decisions required promptly to be taken by the decision-maker (yes/no decisions or more complicated ones).

5.9.3. Fleet Routing *(the numbering here follows the respective in the relevant subchapter)*

{

Input:

Route data.

- a) Names of the routes and their corresponding numbers (r). The correspondence to numbers is not required.
- b) I_r number of ports on route r (for each route).
- c) Which ports comprise each route? Their sequence is not needed.
- d) C_{ij} distance between all port pairs in each route. The distances between port pairs that belong to different routes are not required.
- e) [Optional, but highly recommended] The port that will be the initial one (and naturally the last) in each route.

Output:*Route data.*

- a) The routing sequence of ports in each route; a port is assigned its number (i) according to its position in the routing sequence.
- b) d_{TSP_r} the total sailing distance of the TSP output route (for each route r)

}

<

The decision maker should now either (i) accept the proposed port sequence or (ii) propose his own one taking into consideration the output of TSPdyn in each route.

>

If (i) we skip the following step

ELSE (if (ii))

{

Input:*Route data.*

- a) The routing sequence of ports in each route **according to the decision-maker**; a port is assigned its number (i) according to its position in the routing sequence.

Ouput:

- a) ERD_r is a numerical factor showing us how “profligate” in terms of sailing distance is the route r. ($ERD_r \geq 0$, $ERD_r = 0$ in the case that the final route is the output of TSPdyn).
- b) \overline{ERD} is a numerical factor showing us of the distance “profligacy” in all R routes together, or simply, the average of ERD_r .
- c) [Optional] The average intermediate port stops for each box.
- d) [Optional] The average transit time per box.

<

The decision maker should now confirm the routing sequence of ports in each route after the calculation of the additional above-described formulas.

>

5.9.4. Deciding Frequency of service and re- Routing

{

Input:

- a) Q_{ijr} is the amount of cargo to be carried per annum from port i to port j on route r (for each route)

Output:

- a) L_{ijr} is the amount of cargo on board for a ship sailing from port i to port j on route r, for the case of one voyage per year (a port is serviced every 365 days)
- b) L_r is the amount of cargo in the most heavily loaded leg

If, F_r , the established frequency of service for each route is known then the model calculates:

RC_r the minimum required capacity of ships that are to operate on route r

or/and:

If ships of type k with given capacity V_k are assigned to route r then the model calculates:

RV_r the minimum required number of voyages per year in that route

F_r the corresponding value of frequency of service

}

Additional Output:

The following graph is very useful for visualizing the frequency-capacity tradeoff in the different routes

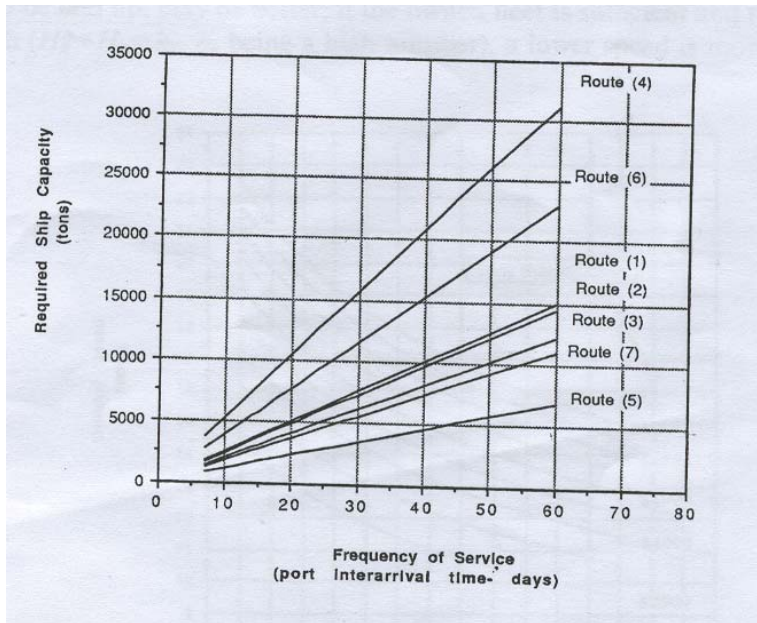


Figure 5.2. Ship Capacity vs. Frequency of Service

Graphs showing the loading condition of the ships in the various legs of a specific route (load levels vs. cumulative distance) provide an insight to the utilization of the ships and provide hints for minor routing or frequency of service modifications.

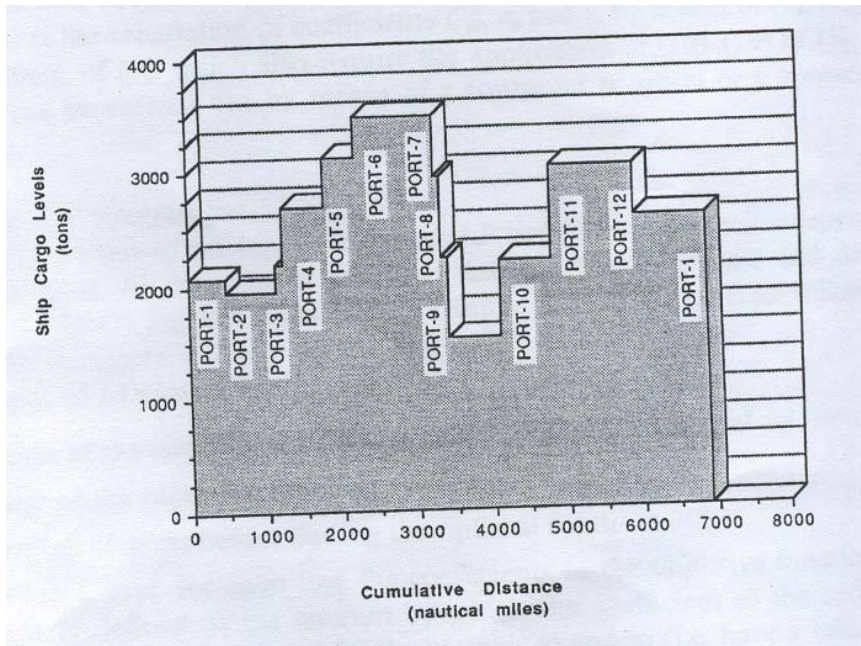


Figure 5.3. Ship cargo levels

<

At this point the fixing of the frequency of service is required in order to avoid non-linear programming afterwards. The decision maker could take advantage of the outputs so far, the two last graphs, as well as take into deliberation marketing considerations.

Consequently, F_r , the frequency of service as defined in the previous report, is known.

>

{

If we have pre-allocated certain ships to certain routes, ELSE we skip the following step.

Input:

- a) Which ship operates in each route?

Output:

- a) \overline{ASU} is the simplified Average Ship Utilization

- b) \overline{ASU} ' is leveled Average Ship Utilization
- c) \overline{AFU} is the simplified Average Fleet Utilization
- d) \overline{AFU} ' is the single-stage leveled Average Fleet Utilization

Additional Input:

- b) C_{kr}' are the (total) annual operating costs of a type k ship in route r

Additional Output:

- e) ES_{kr} is the economies of scale factor for a k type ship operating on route r (\$/TEU)

}

5.9.5. Allocation of Ships to Routes

Input:

- a) $C_{kr}'^*$ operating costs of a type k ship on route r
 $[C_{kr}' = C_{kr} * T_k / t_{kr}]$, where
 C_{kr} is the operating costs per voyage for ship k on route r
 T_k duration of shipping season for a type k ship
 t_{kr} voyage time of a type k ship on route r
- b) e_k daily lay-up cost for a type k ship
- c) N_k^{\max} maximum number of type k ships available
- d) t_{kr}' yearly voyages of a type k ship on route r . That can be calculated via
 $[t_{kr}' = T_k / t_{kr}]$, where

* For a mathematical description of the calculation of C_{kr} , t_{kr} coefficients please refer to the appendix of my previous report.

T_k shipping season for a type k ship

t_{kr} voyage time of type k ship on route r]

e) T_k shipping season for a type k ship

f) M_r number of voyages required per year on route r

g) $N_{kr}=0$, that is, specific ship-route incompatibilities for given (k,r) pairs
due to special cargo characteristics/requirements

Output:

a) N_{kr} the number of type k ships operating on route r

b) Y_k the number lay-up days per year of a type k ship

5.9.6. Transshipment

Transshipment is an optional module; its additional input and output will not be developed herein.

5.10. Conclusions

We have developed a general model for the optimization-based design of a liner network. The problem is originally of a non-linear structure; however, by fixing the various sources of non-linearity and by broking down the total network design into the sequential solution of a set of subproblems we have managed to accomplish our goals through they use of Linear Programming, Dynamic Programming and Integer Programming.

As it has been already described the methodology provides the OR practitioner and/or decision-maker with the following benefits:

1. Uniformity of schedules
2. Adaptability to the:
 - variability of fuel oil prices as well as generator fuel prices;
 - change in the number and types of the owned ships and of the ships available to charter-in; and
 - changes in time and cost components
3. substantial savings in operating costs (1.4% or a saving of \$1 317 000 per year regarding a pragmatic example and regarding only the fleet deployment component)
4. all target frequencies are met and some are improved.

Moreover, the methodology has components that can among others calculate:

- the cargo levels on board at each leg ij of a voyage;
- the least-distance port sequence;
- cost and time components in an exact way; and
- multiple quantitative criteria which evaluate the efficiency of a network.

5.11. Recommendations for future research

Certain parts do not appear in the present version

Supply chain Focus

The aforementioned focus on supply chain performance in the planning of fleet schedules is logical as we already see trends pointing in the direction of a competition between supply chains even more than between shipping companies. Shipping companies must consider themselves as a total logistics provider (or at least as a part of a total logistics provider) instead of only a provider of sea transport services. In this way we can investigate how ocean shipping can be integrated into a multimodal door-to-door supply chain.

Additional Recent Trends in Shipping

The OR practitioner who aims at waterborne transport should take into account the additional following trends:

- Mergers and pooling collaboration resulting in larger operational fleets.
- New generation of planners with more computer experience.
- Developments in software and hardware that facilitates rich models and intuitive GUIs.
- Shift from industrial to tramp shipping, resulting in more market interaction and new opportunities and challenges for optimization-based decision support tools.
- More focus on strategic planning issues such as fleet sizing.

Robustness

What inhibits these other firms from using the algorithms that have been developed?

The answer seems to be a lack of robustness in currently implemented algorithms that makes them hard to transfer from one company to another. According to Psaraftis (1984) “from a practitioner's viewpoint, an algorithm's *robustness* is likely to be much more important than that algorithm's ability to produce arbitrarily small errors for arbitrarily

large problem sizes.” This view seems to be to the point although expressed 21 years ago. In talking with researchers involved in implementing vehicle routing algorithms, one hears a common story. A heuristic algorithm is selected which appears reasonable for a particular situation. Under testing, the algorithm may work well in most cases, but occasionally produces obviously unreasonable results. The heuristic is 'then 'patched up' to fix the troublesome cases, leading to an algorithm with growing complexity and computational requirements. After considerable effort, a procedure is created that works well for the situation at hand, but one is left with the disquieting feeling that what has been produced is extremely sensitive to nuances in the data and will not perform well when transferred to other environments. It's not uncommon that a heuristic developed for a particular geographic region of a company's operation will perform poorly in another region served by the same company. What's needed are more robust tools for vehicle routing.

As the need for robustness has become more clear during the last decade, fortunately the resources for achieving robustness have also grown. Rapidly decreasing computation costs are pushing the tradeoff between computation time and solution quality in the direction of higher quality solutions. The accuracy of data on the cost of travel between customers has been greatly improved by the creation of road network databases. Finally, the base of fundamental research on which to draw has greatly expanded, including optimization research on related models like the traveling salesman problem and new approaches to heuristic problem solving growing out of the artificial intelligence community.

One could imagine various approaches to achieving robustness. The simplest might be to provide an interactive interface for vehicle routing algorithms based on a graphic display that would allow a human dispatcher to manually correct difficulties with solutions. A more complex approach could draw on past research in artificial intelligence. For example, we could use an expert system to capture the expertise used by an operations research analyst in developing and tuning a vehicle routing algorithm for a particular application. The expert system could have available to it all existing vehicle routing algorithms so as to be able to select an appropriate algorithm for a particular application.

Test data could be supplied to the expert system with which to evaluate the performance of candidate algorithms and tune algorithm parameters as appropriate.

My own prediction is that optimization algorithms offer the best promise for achieving robustness. Although optimization has not been considered a practical approach for real problems in the past, rapidly decreasing computation costs and promising new research are causing a reevaluation of this assumption. In a practical application, an optimization algorithm need not be run to full optimality but can be stopped as soon as an acceptable solution has been obtained. As such, these algorithms offer all the computational tractability advantages of heuristics, but they also allow a user to control the tradeoff between solution quality and computational cost.

Finally, significant research has been conducted on the closely related traveling salesman problem. This provides a base of theoretical research on which to draw for vehicle routing optimization. A number of successful vehicle routing optimization algorithms are adaptations of traveling salesman algorithms. The dramatic increase in the size of traveling salesman problems solvable to optimality also suggests what could be accomplished for vehicle routing with a concerted research effort.

6. RECAPITULATION

Initially we have performed an extensive review of:

- Mathematical modeling techniques;
- Algorithms and complexity;
- Network models;
- The traveling salesman problem;
- The vehicle routing problem;
- The implementation of OR models in liner shipping

Most important of all, we have developed a general model for the optimization-based design of a liner network. The problem is originally of a non-linear structure; however, by fixing the various sources of non-linearity and by broking down the total network design into the sequential solution of a set of subproblems we have managed to accomplish our goals through they use of Linear Programming, Dynamic Programming and Integer Programming.

As it has been already described the methodology provides the OR practitioner and/or decision-maker with the following benefits:

1. Uniformity of schedules
2. Adaptability to the:
 - Variability of fuel oil prices as well as generator fuel prices;
 - Change in the number and types of the owned ships and of the ships available to charter-in; and
 - Changes in time and cost components
3. Substantial savings in operating costs (1.4% or a saving of \$1 317 000 per year regarding a pragmatic example and regarding only the fleet deployment component)
4. All target frequencies are met and some are improved.

Moreover, the methodology has components that can among others calculate:

- the cargo levels on board at each leg ij of a voyage;
- the least-distance port sequence;
- cost and time components in an exact way; and
- multiple quantitative criteria which evaluate the efficiency of a network.

Last but not least, we recommend certain methodologies which advance the state-of-the-art in contemporary ship routing and scheduling research.

References^{*}

1. ACHUTHAN, CACCETTA, AND HILL (2003), An Improved Branch-and-Cut Algorithm for the Capacitated Vehicle Routing Problem, *Transportation Science* Vol. 37, No. 2, May 2003, pp. 153–169.
2. BALDACCI R., HADJICONSTANTINO E., MINGOZZI A. (2003) An Exact Algorithm for the Traveling Salesman Problem with Deliveries and Collections, *NETWORKS*, Vol. 42(1), 26 –41.
3. BRADLEY, HAX, and MAGNANTI (1977), Applied Mathematical Programming. *Addison-Wesley*.
4. CHRISTOFIDES, N., Vehicle Routing, eds. LAWLER, E. L., LENSTRA, J. K., RINNOOY KAN, A. H. G. AND SHMOYS, D. B. *The traveling salesman problem*, Wiley Interscience.
5. CORDEAU, J-F., GENDREAU M., LAPORTE G. (1997) A Tabu Search Heuristic for Periodic and Multi-Depot Vehicle Routing Problems, *Networks*, Vol. 30, pp. 115-119.
6. CORMEN H., THOMAS et al. (2001) Introduction to Algorithms, MIT Press.
7. DESROSIERS, J. et al. Time constrained routing and scheduling, Ball, M. O. et al., eds. *Handbooks in OR & MS, Vol. 8, Network Routing*, Elsevier Science B. V.
8. FAGERHOLT, K. (1999). Optimal fleet design in a ship routing problem. *International Transactions in Operational Research* 6(5), 453-464.
9. FISHER, M. (1995) Vehicle routing, Ball, M. O. et al., eds. *Handbooks in OR & MS, Vol. 8, Network Routing*, Elsevier Science B. V.
10. FOULDS, L. R., (1984) Combinatorial Optimization for Undergraduates, *Springer-Verlag*.
11. JANSSEN, J. O. and D. SHNEERSON (1982). The optimal ship size. *Journal of Transport Economics and Policy* 16(3), 217-238.
12. JARAMILLO, D. I. and PERAKIS, A. N. (1991), Fleet deployment optimization for liner shipping. Part 2: Implementation and results. *Maritime Policy and*

^{*} Not an exhaustive list

- Management* 18, 235-262.
13. JAW, J. J., ODoni A. R., PSARAFTIS, H. N. and WILSON N. (1986) A heuristic algorithm for the multi vehicle advance request dial-a-ride problem with time windows, *Transportation Research*, Vol. 20 (3), pp. 243-257.
 14. JOHNSON, D. S. AND PAPADIMITRIOU, C. H. (1985) Computational Complexity, eds. LAWLER, E. L., LENSTRA, J. K., RINNOOY KAN, A. H. G. AND SHMOYS, D. B. *The traveling salesman problem*, Wiley Interscience.
 15. JOHNSON, D. S. AND PAPADIMITRIOU, C. H. (1985) Performance guarantees for heuristics, eds. LAWLER, E. L., LENSTRA, J. K., RINNOOY KAN, A. H. G. AND SHMOYS, D. B. *The traveling salesman problem*, Wiley Interscience.
 16. LAPORTE, G. and DEJAX, P. (1989) Dynamic location routing problems, *Journal of the Operations Research Society* 40, pp. 471-482.
 17. LAPORTE, G., NOBERT, Y., DESROCHERS, M. (1985) Optimal routing under capacity and distance restrictions, *Operations Research*, 33(5).
 18. LAWLER, E. L., LENSTRA, J. K., RINNOOY KAN, A. H. G. AND SHMOYS, D. B. (1985) The traveling salesman problem, *Wiley Interscience*.
 19. LINGO User's Guide (2004), *Lindo Systems Inc*.
 20. LYRIDIS, D. V., PSARAFTIS, H. N., VENTIKOS, N., ZACHARIOUDAKIS, P., DILZAS, K. (2005) Cost-benefit analysis for ship automation retrofit: the case of icebreaker Frej, *Marine Technology*, Vol. 42 (2).
 21. MALANDRAKI, C. and DASKIN, M. (1992) Time dependent vehicle routing problems: Formulations, properties, and heuristic algorithms, *Transportation Science* 26, pp. 185-200.
 22. MARIELLE CHRISTIANSEN & KJETIL FAGERHOLT, Ship routing and scheduling - Status and trends, *Transportation Science*, *Transportation Science*, vol. 38. no1, Feb. 2004.
 23. MOSKOWITZ, H., PRECKEL, P. AND YANG, A. (1993) Decision Analysis with incomplete utility and probability information, *Operations Research*, Vol. 41, No. 5.
 24. PERAKIS, A. N. and JARAMILLO, D. I. (1991), Fleet deployment optimization

- for liner shipping. Part 1: Background, problem formulation and solution approaches. *Maritime Policy and Management* 18, 183-200.
25. POWELL, B. J. and PERAKIS, A. N. (1997) Fleet deployment optimization for liner shipping: an integer programming model. *Maritime Policy and Management* 24, 183-192.
 26. Profit Optimization for Container Carriers (White Paper) presented by the ROI Container Cargo Alliance.
 27. PSARAFTIS, H. N. (1983) Analysis of an $O(N^2)$ heuristic for the single vehicle many-to-many Euclidean dial-a-ride problem, *Transportation Research*, Vol. 17 (2), pp. 133-145.
 28. PSARAFTIS, H. N. (1983) k-interchange procedures for local search in a precedence constrained routing problem, *European Journal of Operational Research* 13, 391-402.
 29. PSARAFTIS, H. N. (1984) On the practical importance of asymptotic optimality in certain heuristic algorithms, *Networks*, Vol. 14, pp.587-596.
 30. PSARAFTIS, H. N. (1988) Dynamic vehicle routing problems, GOLDEN B. L. AND ASSAD, A. A., eds. *Vehicle Routing: Methods and Studies*, North Holland, Amsterdam, The Netherlands, 223-248.
 31. PSARAFTIS, H. N. (1995) Dynamic vehicle routing: status and prospects, *Annals of Operations Research*, 61, 143-164.
 32. PSARAFTIS, H. N. (1999) Foreword to the focused issue on maritime transportation, *Transportation Science*, 33(1), 1-2.
 33. PSARAFTIS, H. N., SOLOMON, M. M., MAGNANTI, T. L. and KIM, T. (1990), Routing and scheduling on a shoreline with release times, *Management Science* 36, 212-223.
 34. ROBERTO WOLFLER CALVO (2000) A New Heuristic for the Traveling Salesman Problem with Time Windows, *Transportation Science*, Vol. 34, No. 1, pp. 113-124.
 35. Sarlis Container Services Official Website (<http://www.sarlis.gr>).
 36. SCHRAGE, L. (2003) Optimization Modeling with LINGO, *Lindo Systems Inc.*.
 37. STOPFORD, M. (1997) Maritime Economics, 2nd edition, Routhledge.

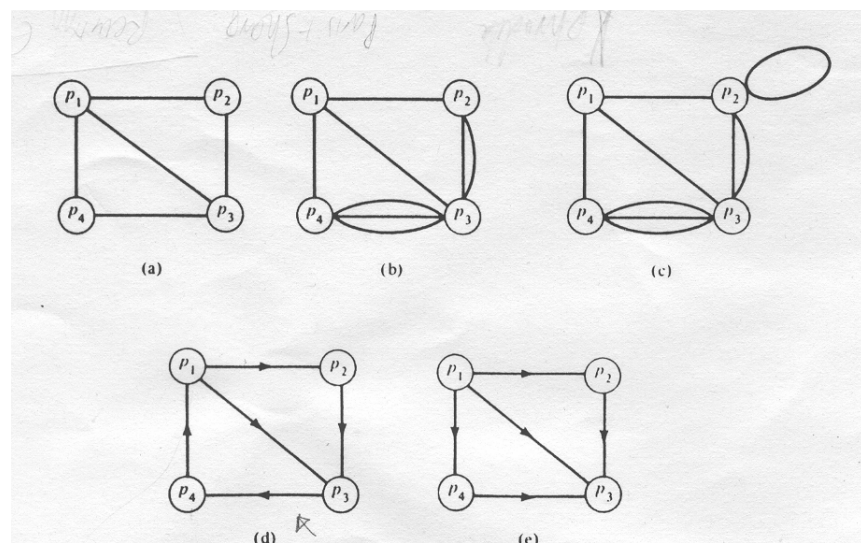
38. THOMPSON, M. (1988) Local Search Algorithms for Vehicle Routing and other Combinatorial Problems, Ph.D. Dissertation, *MIT*.
39. THOMPSON, P. M. AND PSARAFTIS, H. N. (1993) Cyclic Transfer Algorithms for Multivehicle Routing and Scheduling Problems, *Operations Research*, Vol. 41, No. 5.

Appendix A. Graph Theory

Over the last 20 years graph theory (GT) has developed into a systematic tool for analysing practical problems from a wide variety of disciplines. Since its beginning, 250 years ago, remarkably few GT works have used a common terminology. This is not necessarily inefficient as it allows each author to introduce a language tailor-made for his or her purposes. Herein a modern terminology is chosen. This section should be used like a dictionary-to be referred to when necessary. It contains a few ideas not referred to earlier. These were included when they arise naturally in the development and may prove useful to the reader if he or she delves more deeply into GT and its applications.

A *graph* G consists of a finite, nonempty set V of *vertices* together with a given set E of unordered pairs of distinct vertices of V . Each element $(p, q) \in E$ is called an *edge* and is said to *join* vertices p and q . Synonyms for "vertex" are *point*, *node*, and *junction*. Synonyms for "edge" are *line*, *arc*, *branch*, and *link*. If edge $e = \{p, q\} \in E$, p and q are both said to be *incident* with e and *adjacent* to each other. An edge $\{p, q\}$ is denoted by pq when ambiguity does not arise.

One of the features that makes GT more fun is our ability to represent any graph by a picture. This is always theoretically possible as V is assumed finite. Each vertex in V can be represented by a geometric point in the plane and each edge in E by a geometric line joining the geometric points with which it is incident. As an example, the graph $G = (V, E)$, where $V = \{p_1, p_2, p_3, p_4\}$ and $E = \{\{p_1, p_2\}, \{p_2, p_3\}, \{p_3, p_4\}, \{p_4, p_1\}, \{p_1, p_3\}\}$ is



depicted in Fig. A.1(a):

Figure A.1. (a) graph; (b) multigraph; (c) pseudograph; (d) digraph; (e) network

There are a number of useful structures that bear a close resemblance to the structure of a graph. Recall that no more than one edge can join any two vertices of a graph. Structures in which this constraint is relaxed are called *multigraphs*. Thus a *multigraph* is an ordered pair of sets (V, E) , where V is finite and nonempty and E is a class of unordered pairs of distinct vertices of V where repetitions in the class are allowed. An example of a multigraph is given in Fig. A.1(b). Recall that each edge in a graph and a multigraph joins two *distinct* vertices. Structures in which this constraint is relaxed and multiple edges are allowed are called *pseudographs*. Thus a *pseudograph* is an ordered pair of sets (V, E) , where V is finite and nonempty and E is a class of unordered pairs of vertices of V where repetitions in the class are allowed. An example of a pseudograph is given in Fig. A.1(c). Recall that each edge in a graph is represented by an *unordered* pair of vertices of V . Sometimes it is useful to give the edges an orientation or direction. When this is carried out, the graph is termed *directed* and is called a *digraph*. Thus a *digraph* is an ordered pair of sets (V, A) , where V is finite and nonempty and A is a set of ordered pairs of distinct vertices of V . An example of a digraph is given in Fig. A.1(d). Much graph theoretic terminology can and will be used analogously for digraphs in this chapter. There is a special class of digraphs called *networks* which are of extreme importance. They can be introduced as follows. If $a = (p_i, p_j)$ is an arc in a digraph $D = (V, A)$, then a is said to be *directed away* from p_i and *directed toward* p_j . The vertices of a network are usually called nodes. Any node which has at least one arc directed away from it is said to be a *source*. Any node which has at least one arc directed toward it is said to be a *sink*. Of course nodes can have arcs directed both toward and away from them making them both sources and sinks. A source (sink) which is not a sink (source) is termed a *proper source (sink)*. A digraph through the arcs of which a commodity is assumed to flow and which usually has at least one proper source and at least a proper sink is termed a *network*. An example of a network is given in Fig. A.1(e).

Returning to graphs, it is sometimes appropriate to examine just a part of a graph. This can be done in a number of different ways. Given a graph $G = (V, E)$ we could consider:

(1) just some of its vertices and all the existing edges in E joining them; (2) all of its vertices and just some of the existing edges joining them; or (3) some of its vertices and some of the edges joining them provided we include all vertices incident with any edges considered. As an example:

(1) Let U be a nonempty subset of V . The graph whose vertex set is U and whose edge set comprises exactly the edges of E which join vertices in U , is termed a *subgraph* of G .

(2) Let F be a subset of E . (V, F) , the graph with the same vertex set as G , but only the edges in F , is termed a *partial* or *spanning graph* of G .

(3) A graph formed by the combination of (1) and (2) where care is taken to include the end points (incident vertices) of each edge selected is termed a *partial subgraph* of G .

Figures A.2(a)-A.2(c) provide examples of a subgraph, partial graph, and partial subgraph, respectively, of the graph in Fig. A.2(a).

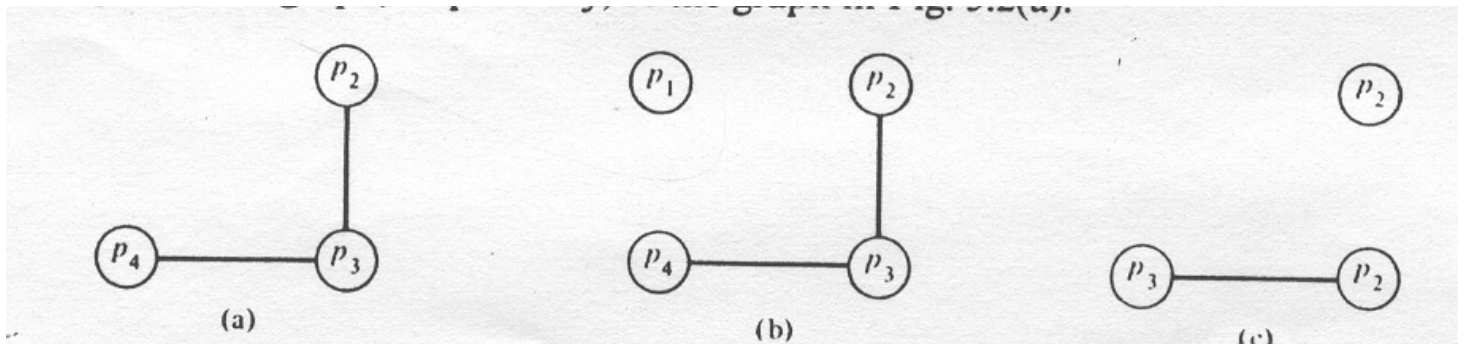


Figure A.2. Parts of the Graph in Fig. 5.1(a): (a) Subgraph; (b) Partial Graph; (c) Partial Subgraph.

One of the most important graph theoretic concepts is that of connectivity. A *walk* in a graph G is a sequence of vertices and edges of G of the form:

$$\langle p_1, \{p_1, p_2\}, p_2, \{p_2, p_3\}, p_3, \dots, p_{n-1}, \{p_{n-1}, p_n\}, p_n \rangle$$

Note that the sequence begins and ends with a vertex and each edge is incident with the vertices immediately preceding and succeeding it. It is said to *join* p_1 and p_n . When there

is no ambiguity, it is denoted by $\langle p_1, p_2, p_3, \dots, p_{n-1}, p_n \rangle$. A walk is termed *closed* if $p_1 = p_n$ and *open* if $p_1 \neq p_n$. A walk is termed a *trail* if all of its edges are distinct and a *path* if all its vertices (and necessarily its edges) are distinct. A closed walk with at least three vertices and all of its vertices distinct is called a *cycle*. A graph G is *connected* if every pair of vertices are joined by at least one path and *k connected* if at least k vertices must be removed to make G no longer connected. In Fig. A.1(a), $\langle p_1, p_2, p_3, p_1, p_2 \rangle$ is an open walk; $\langle p_1, p_2, p_3, p_4, p_1, p_3 \rangle$ is a trail; $\langle p_1, p_2, p_3 \rangle$ is a path; and $\langle p_1, p_2, p_3, p_4, p_1 \rangle$ is a cycle. A connected partial subgraph of a graph is termed a *spanning subgraph*.

A digraph is termed *connected* if the underlying graph derived from it by ignoring the orientation of its arcs and replacing multiple edges by a single edge is connected. A digraph is termed *strongly connected* if there is a directed sequence of arcs from each of its vertices to each other vertex. A strongly connected subdigraph d of a digraph D is said to be minimal if d has no proper subdigraph of two or more vertices which is strongly connected.

A graph without cycles is termed *acyclic*. This leads us to one of the most important classes of graphs. A connected acyclic graph G is termed a *tree*. There are many equivalent definitions of a tree including:

- (1) Every two vertices of G are joined by a unique path.
- (2) G is connected and its number of vertices exceeds its number of edges by one.

Appendix B

FILE MENU

<i>Command</i>	<i>Description</i>
NEW	Open a new Model window.
OPEN...	Open a file.
SAVE	Save a file.
SAVE AS...	Save an existing file under a new name.
CLOSE	Close the active window.
PRINT	Print the active window.
PRINTER SETUP...	Select a printer and print options.
LOG OUTPUT...	Write all Reports Window activity to a file.
TAKE COMMANDS...	Open a batch file with LINGO commands.
IMPORT LINDO FILE...	Translate a LINDO model to its LINGO equivalent.
EXIT	Quit LINGO.

EDIT MENU

<i>Command</i>	<i>Description</i>
UNDO	Undo the last action.
CUT	Clear selected text and place on clipboard.
COPY	Place selected text on clipboard.
PASTE	Insert clipboard text.
FIND/REPLACE...	Find specified text/Replace with specified text.
GO TO LINE...	Go to specified line.

MATCH PARENTHESIS	Find close parenthesis corresponding to selected open parenthesis.
PASTE FUNCTION	Paste a LINGO function at insertion point.
SELECT ALL	Select the entire window.
CHOOSE NEW FONT...	Choose a new font for the active window.

LINGO MENU

<i>Command</i>	<i>Description</i>
SOLVE	Solve the model currently in memory.
SOLUTION...	View the solution currently in memory.
RANGE	Create a range report on the model currently in memory.
LOOK...	Look at specified lines of the current model.
GENERATE...	Generate an algebraic, LINDO, or MPS version of the model currently in memory.
EXPORT TO SPREADSHEET...	Export solution values to a spreadsheet file.
OPTIONS...	Set various LINGO parameters
WORKSPACE LIMIT...	Set LINGO's memory allocation

WINDOWS MENU

Command	Description
OPEN COMMAND	

WINDOW	Opens a window for command line operation of LINGO.
OPEN STATUS WINDOW	Opens the LINGO Status Window with results of the last solver run.
SEND TO BACK	Sends the frontmost window to the back.
CLOSE ALL	Closes all windows..
CASCADE	Arranges windows in cascade from upper left to lower right.
TILE	Arranges windows in equal size to fill screen.
ARRANGE ICONS	Moves icons representing minimized documents to within the LINGO window.
LIST OF WINDOWS	A list of open windows appears at the bottom of the Windows menu, from which you can choose one to bring to the front. The currently active window is checked.

HELP MENU

Command	Description
HELP	Open LINGO help.

FUNCTIONS, OPERATORS AND RELATIONS

1. Arithmetic Operators

Arithmetic operators work with numeric operands. LINGO has five binary (two-operand) arithmetic operators, shown here.

^	Power.
*	Multiplication.
/	Division.
+	Addition.

- Subtraction.

2. Relational and Logical Operators

RELATIONAL OPERATORS

#EQ#	Returns TRUE if the operands are equal, FALSE if not.
#NE#	Returns TRUE if the operands are not equal, FALSE if the operands are equal.
#GT#	Returns TRUE if the left operand is strictly greater than the right operand, FALSE if not.
#GE#	Returns TRUE if the left operand is greater than or equal to the right operand, FALSE otherwise.
#LT#	Returns TRUE if the left operand is strictly less than the right operand, FALSE if not.
#LE#	Returns TRUE if the left operand is less than or equal to the right operand, FALSE otherwise.

LOGICAL OPERATORS

#NOT#	Negates the logical value of its operand. #NOT# is a unary operator, with its sole argument to the right.
#AND#	Returns the logical AND of its two operands. #AND# returns TRUE only if both its arguments are TRUE, otherwise it returns FALSE.

#OR# Returns the logical OR of its two operands. #OR# returns FALSE only if both its arguments are FALSE, otherwise it returns TRUE.

3. Equality and Inequality Relations

= The expression to the left must equal the one on the right.
<= The expression to the left must be less than or equal to the expression on the right.
>= The expression to the left must be greater than or equal to the expression on the right.

For the sake of convenience, LINGO will also interpret the strict inequality relations "<" and ">" as their loose inequality counterparts.

Financial Functions

@FPA(I,N) Returns the present value of an annuity, specifically, a stream of \$1 payments per period at interest rate of I per period, for N periods, starting one period from now. I is not a percentage but rather a non-negative number representing the interest rate. To get the present value of an annuity stream of \$X payments, multiply the result by X.

@FPL(I,N) Returns the present value of a lump sum of \$1, N periods from now if the interest rate is I per period. I is not a percentage but rather a nonnegative number representing the interest rate.

4. General Mathematical Functions

- @ABS(X)** Returns the absolute value of X.
- @COS(X)** Returns the cosine of X, where X is the angle in radians.
- @EXP(X)** Returns the constant e (2.718281...) to the power X.
- @LGM(X)** Returns the natural (base e) logarithm of the gamma function of X. For integral values of X the following is true:
$$@LGM(X) = @LOG(X-1).$$
- @LOG(X)** Returns the natural logarithm of X.
- @SIGN(X)** Returns -1 if $X < 0$, returns +1 if $X \Rightarrow 0$.
- @SIN(X)** Returns the sine of X, where X is the angle in radians.
- @SMAX(list)** Returns largest value in list of scalars.
- @SMIN(list)** Returns smallest value in list of scalars.
- @TAN(X)** Returns the tangent of X, where X is the angle in radians.

5. Set-Looping Functions

The | condition part in the specification is optional. The available functions are listed below.

@FOR (set_name : constraint_expressions)

Generates constraints independently for each element of the set named in set_name.

@MAX (set_name : expression)

Returns the maximum value taken by expression over the set.

@MIN (set_name : expression)

Returns the minimum value expression takes on over the set.

@SUM (set_name : expression)

Returns the sum of expression over set_name.

6. Variable Domain Functions

@BND(L, X, U) Limits the variable or attribute X to greater or equal to L and less than or equal to U. For operations across sets, you may use the @FOR operator to apply @BND to every element. For example, to set lower and upper bounds of 1 and 100 on the X attribute of all elements of the set

MYSET, enter: @FOR(MYSET(I): @BND(1, X, 100))

@BIN(X) Limits the variable or attribute X to a binary integer value (0 or 1.) For operations across sets, you may use the @FOR operator to apply @BIN to the desired elements. For example, to make the X attribute of all elements of the set MYSET binary integer, enter:

@FOR(MYSET(I): @BIN(X))

@FREE(X) Removes the default lower bound of zero on a variable or attribute, allowing it to take any positive or negative value. For operations across sets, you may use the **@FOR** operator to apply **@FREE** to the desired elements. For example, to make the X attribute of all elements of the set MYSET unconstrained in sign, enter **@FOR(MYSET(I): @FREE(X))**

@GIN(X) Limits the variable or attribute X to only integer values. For operations across sets, you may use the **@FOR** operator to apply **@GIN** to the desired elements. For example, to make the X attribute of all elements of the set MYSET integer, enter:
@FOR(MYSET(I): @GIN(X))

7. Probability Functions

@PSN(X) Cumulative standard normal probability distribution. A standard normal random variable has mean 0.0 and standard deviation 1.0 (the bell curve, centered on the origin). The value returned by **@PSN** is the area under the curve to the left of the point on the ordinate indicated by X.

@PSL(X) Unit normal linear loss function. **@PSL** returns the expected value of $\text{MAX}(0, Z-X)$, where Z is a standard normal random variable. In inventory modeling, **@PSL(X)** is the expected amount by which demand exceeds a level X, if the demand has a standard normal distribution.

@PPS(A,X) Cumulative Poisson probability distribution. **@PPS** returns the probability that a Poisson random variable, with mean value A, is less than or equal to X. If X is noninteger, **@PPS** uses linear interpolation to compute the probability.

@PPL(A,X) Linear loss function for the Poisson distribution. **@PPL** returns the expected value of $\text{MAX}(0, Z-X)$, where Z is a Poisson random variable with mean value A .

@PBN(P,N,X) Cumulative binomial probability. **@PBN** returns the probability that a sample of N items, from a universe with a fraction of P of those items defective, has X or less defective items. It is extended to noninteger values of X and N by linear interpolation.

@PHG(POP,G,N,X) Cumulative hypergeometric probability. Given a population of size POP , of which number G are good, and a sample without replacement of N items from the population, **@PHG** returns the probability that X or fewer items in the sample are good. It is extended to noninteger values of POP , G , N , and X by linear interpolation.

@PEL(A,X) Erlang's loss probability for a service system with X servers and an arriving load of A , no queue allowed. The result of **@PEL** can be interpreted as either the fraction of time all servers are busy or the fraction of customers lost due to all servers being busy when they arrive. It is extended to noninteger values of X by linear interpolation. The arriving load, A , is the expected number of customers arriving per unit time multiplied by the expected time to process one customer.

@PEB(A,X) Erlang's busy probability for a service system with X servers and an arriving load of A , with infinite queue allowed. The result of **@PEB** can be interpreted as either the fraction of time all servers are busy or the fraction of customers that must wait in the queue. It is extended to noninteger values of X by linear interpolation.

The arriving load, A , is the expected number of customers arriving per unit time multiplied by the expected time to process one customer.

@PFS(A, X, C) Expected number of customers waiting for or under repair in a finite source Poisson service system with X servers in parallel, C customers, and a limiting load A . It is extended to noninteger values of X and C by linear interpolation. A is the number of customers multiplied by the mean service time divided by the mean repair time.

@PFD(N, D, X) Cumulative distribution function for the F distribution with N degrees of freedom in the numerator and D degrees of freedom in the denominator.

@PCX(N, X) Cumulative distribution function for the Chi-squared distribution with N degrees of freedom.

@PTD(N, X) Cumulative distribution function for the t distribution with N degrees of freedom.

@RAND(X) Returns a pseudo-random number between 0 and 1, depending deterministically on X . Typical use is $U(I) = \text{@RAND}(U(I-1) + 1)$.

8. Other Functions

@IN (set_name , set_element) Returns 1 if set_element is present in set_name; returns 0 if not present.

@SIZE (set_name) Returns the number of elements in the set.

`@WARN('text', condition)` If the condition is true, the message text' is displayed when the model is solved. This is useful for warning the user of any abnormal data values.

`@WRAP(I,N)` Returns I if I is in the interval [1, N]. Otherwise, @WRAP subtracts N from I until I is in the interval [1, N], then returns I. (Formally, returns J such that $J = I - K * N$, where K is a positive or negative integer such that J is in the interval [1, N]). @WRAP is undefined if $N < 1$. The @WRAP function is useful, for instance, in staffing models when you want to assign a seven day period that wraps around from Thursday to Tuesday.

Appendix C. Full LINGO Output of Fleet Deployment Example

Model Title: Deployment

Global optimal solution found.

Objective value: 91831.00

Extended solver steps: 1046

Total solver iterations: 3793

Variable	Value	Reduced Cost
SHIPPING_SEASON(1)	345.0000	0.000000
SHIPPING_SEASON(2)	345.0000	0.000000
SHIPPING_SEASON(3)	345.0000	0.000000
SHIPPING_SEASON(4)	345.0000	0.000000
SHIPPING_SEASON(5)	345.0000	0.000000
SHIPPING_SEASON(6)	345.0000	0.000000
SHIPPING_SEASON(7)	345.0000	0.000000
SHIPPING_SEASON(8)	345.0000	0.000000
SHIPPING_SEASON(9)	345.0000	0.000000
SHIPPING_SEASON(10)	345.0000	0.000000
SHIPPING_SEASON(11)	345.0000	0.000000
N_LAYUP_DAYS(1)	120.0000	0.000000
N_LAYUP_DAYS(2)	40.00000	0.000000
N_LAYUP_DAYS(3)	60.00000	0.000000
N_LAYUP_DAYS(4)	20.00000	0.000000
N_LAYUP_DAYS(5)	20.00000	0.000000
N_LAYUP_DAYS(6)	20.00000	0.000000
N_LAYUP_DAYS(7)	40.00000	0.000000
N_LAYUP_DAYS(8)	730.0000	0.000000
N_LAYUP_DAYS(9)	730.0000	0.000000
N_LAYUP_DAYS(10)	40.00000	0.000000

N_LAYUP_DAYS(11)	385.0000	0.000000
LAYUP_COST(1)	9.100000	0.000000
LAYUP_COST(2)	9.000000	0.000000
LAYUP_COST(3)	9.000000	0.000000
LAYUP_COST(4)	7.400000	0.000000
LAYUP_COST(5)	7.300000	0.000000
LAYUP_COST(6)	8.900000	0.000000
LAYUP_COST(7)	9.000000	0.000000
LAYUP_COST(8)	0.000000	0.000000
LAYUP_COST(9)	0.000000	0.000000
LAYUP_COST(10)	0.000000	0.000000
LAYUP_COST(11)	0.000000	0.000000
N_SHIPS_MAX(1)	6.000000	0.000000
N_SHIPS_MAX(2)	2.000000	0.000000
N_SHIPS_MAX(3)	3.000000	0.000000
N_SHIPS_MAX(4)	1.000000	0.000000
N_SHIPS_MAX(5)	1.000000	0.000000
N_SHIPS_MAX(6)	1.000000	0.000000
N_SHIPS_MAX(7)	2.000000	0.000000
N_SHIPS_MAX(8)	2.000000	0.000000
N_SHIPS_MAX(9)	2.000000	0.000000
N_SHIPS_MAX(10)	2.000000	0.000000
N_SHIPS_MAX(11)	2.000000	0.000000
N_VOYAGES_REQUIRED(1)	25.00000	0.000000
N_VOYAGES_REQUIRED(2)	22.00000	0.000000
N_VOYAGES_REQUIRED(3)	19.00000	0.000000
N_VOYAGES_REQUIRED(4)	22.70000	0.000000
N_VOYAGES_REQUIRED(5)	12.00000	0.000000
N_VOYAGES_REQUIRED(6)	19.00000	0.000000
N_VOYAGES_REQUIRED(7)	10.40000	0.000000
N_SHIPS_ON_ROUTE(1, 1)	3.000000	1883.500

N_SHIPS_ON_ROUTE(1, 2)	0.000000	1930.500
N_SHIPS_ON_ROUTE(1, 3)	0.000000	1742.500
N_SHIPS_ON_ROUTE(1, 4)	3.000000	1861.500
N_SHIPS_ON_ROUTE(1, 5)	0.000000	1861.500
N_SHIPS_ON_ROUTE(1, 6)	0.000000	1842.500
N_SHIPS_ON_ROUTE(1, 7)	0.000000	1895.500
N_SHIPS_ON_ROUTE(2, 1)	0.000000	2139.000
N_SHIPS_ON_ROUTE(2, 2)	0.000000	2205.000
N_SHIPS_ON_ROUTE(2, 3)	0.000000	1982.000
N_SHIPS_ON_ROUTE(2, 4)	0.000000	2098.000
N_SHIPS_ON_ROUTE(2, 5)	0.000000	2157.000
N_SHIPS_ON_ROUTE(2, 6)	1.000000	2141.000
N_SHIPS_ON_ROUTE(2, 7)	1.000000	2102.000
N_SHIPS_ON_ROUTE(3, 1)	0.000000	2122.000
N_SHIPS_ON_ROUTE(3, 2)	0.000000	2183.000
N_SHIPS_ON_ROUTE(3, 3)	0.000000	1982.000
N_SHIPS_ON_ROUTE(3, 4)	1.000000	2088.000
N_SHIPS_ON_ROUTE(3, 5)	0.000000	2144.000
N_SHIPS_ON_ROUTE(3, 6)	2.000000	2132.000
N_SHIPS_ON_ROUTE(3, 7)	0.000000	2102.000
N_SHIPS_ON_ROUTE(4, 1)	0.000000	0.000000
N_SHIPS_ON_ROUTE(4, 2)	1.000000	1685.000
N_SHIPS_ON_ROUTE(4, 3)	0.000000	0.000000
N_SHIPS_ON_ROUTE(4, 4)	0.000000	0.000000
N_SHIPS_ON_ROUTE(4, 5)	0.000000	1731.000
N_SHIPS_ON_ROUTE(4, 6)	0.000000	0.000000
N_SHIPS_ON_ROUTE(4, 7)	0.000000	1855.000
N_SHIPS_ON_ROUTE(5, 1)	0.000000	0.000000
N_SHIPS_ON_ROUTE(5, 2)	1.000000	1570.500
N_SHIPS_ON_ROUTE(5, 3)	0.000000	0.000000
N_SHIPS_ON_ROUTE(5, 4)	0.000000	0.000000

N_SHIPS_ON_ROUTE(5, 5)	0.000000	1616.500
N_SHIPS_ON_ROUTE(5, 6)	0.000000	0.000000
N_SHIPS_ON_ROUTE(5, 7)	0.000000	1777.500
N_SHIPS_ON_ROUTE(6, 1)	0.000000	1842.500
N_SHIPS_ON_ROUTE(6, 2)	0.000000	1882.500
N_SHIPS_ON_ROUTE(6, 3)	0.000000	1765.500
N_SHIPS_ON_ROUTE(6, 4)	0.000000	1864.500
N_SHIPS_ON_ROUTE(6, 5)	0.000000	1849.500
N_SHIPS_ON_ROUTE(6, 6)	1.000000	1850.500
N_SHIPS_ON_ROUTE(6, 7)	0.000000	1882.500
N_SHIPS_ON_ROUTE(7, 1)	0.000000	2136.000
N_SHIPS_ON_ROUTE(7, 2)	0.000000	2205.000
N_SHIPS_ON_ROUTE(7, 3)	2.000000	1982.000
N_SHIPS_ON_ROUTE(7, 4)	0.000000	2098.000
N_SHIPS_ON_ROUTE(7, 5)	0.000000	2157.000
N_SHIPS_ON_ROUTE(7, 6)	0.000000	2141.000
N_SHIPS_ON_ROUTE(7, 7)	0.000000	0.000000
N_SHIPS_ON_ROUTE(8, 1)	0.000000	4195.000
N_SHIPS_ON_ROUTE(8, 2)	0.000000	4227.000
N_SHIPS_ON_ROUTE(8, 3)	0.000000	4146.000
N_SHIPS_ON_ROUTE(8, 4)	0.000000	4227.000
N_SHIPS_ON_ROUTE(8, 5)	0.000000	4208.000
N_SHIPS_ON_ROUTE(8, 6)	0.000000	4216.000
N_SHIPS_ON_ROUTE(8, 7)	0.000000	0.000000
N_SHIPS_ON_ROUTE(9, 1)	0.000000	5096.000
N_SHIPS_ON_ROUTE(9, 2)	0.000000	5152.000
N_SHIPS_ON_ROUTE(9, 3)	0.000000	4958.000
N_SHIPS_ON_ROUTE(9, 4)	0.000000	5100.000
N_SHIPS_ON_ROUTE(9, 5)	0.000000	5079.000
N_SHIPS_ON_ROUTE(9, 6)	0.000000	5065.000
N_SHIPS_ON_ROUTE(9, 7)	0.000000	0.000000

N_SHIPS_ON_ROUTE(10, 1)	0.000000	0.000000
N_SHIPS_ON_ROUTE(10, 2)	0.000000	0.000000
N_SHIPS_ON_ROUTE(10, 3)	0.000000	0.000000
N_SHIPS_ON_ROUTE(10, 4)	0.000000	0.000000
N_SHIPS_ON_ROUTE(10, 5)	2.000000	2849.000
N_SHIPS_ON_ROUTE(10, 6)	0.000000	0.000000
N_SHIPS_ON_ROUTE(10, 7)	0.000000	0.000000
N_SHIPS_ON_ROUTE(11, 1)	0.000000	3722.000
N_SHIPS_ON_ROUTE(11, 2)	0.000000	2784.000
N_SHIPS_ON_ROUTE(11, 3)	0.000000	3513.000
N_SHIPS_ON_ROUTE(11, 4)	1.000000	3695.000
N_SHIPS_ON_ROUTE(11, 5)	0.000000	3678.000
N_SHIPS_ON_ROUTE(11, 6)	0.000000	3639.000
N_SHIPS_ON_ROUTE(11, 7)	0.000000	0.000000
OPERATING_COSTS(1, 1)	5023.000	0.000000
OPERATING_COSTS(1, 2)	5070.000	0.000000
OPERATING_COSTS(1, 3)	4882.000	0.000000
OPERATING_COSTS(1, 4)	5001.000	0.000000
OPERATING_COSTS(1, 5)	5001.000	0.000000
OPERATING_COSTS(1, 6)	4982.000	0.000000
OPERATING_COSTS(1, 7)	5035.000	0.000000
OPERATING_COSTS(2, 1)	5244.000	0.000000
OPERATING_COSTS(2, 2)	5310.000	0.000000
OPERATING_COSTS(2, 3)	5087.000	0.000000
OPERATING_COSTS(2, 4)	5203.000	0.000000
OPERATING_COSTS(2, 5)	5262.000	0.000000
OPERATING_COSTS(2, 6)	5246.000	0.000000
OPERATING_COSTS(2, 7)	5207.000	0.000000
OPERATING_COSTS(3, 1)	5227.000	0.000000
OPERATING_COSTS(3, 2)	5288.000	0.000000
OPERATING_COSTS(3, 3)	5087.000	0.000000

OPERATING_COSTS(3, 4)	5193.000	0.000000
OPERATING_COSTS(3, 5)	5249.000	0.000000
OPERATING_COSTS(3, 6)	5237.000	0.000000
OPERATING_COSTS(3, 7)	5207.000	0.000000
OPERATING_COSTS(4, 1)	4234.000	0.000000
OPERATING_COSTS(4, 2)	4238.000	0.000000
OPERATING_COSTS(4, 3)	4282.000	0.000000
OPERATING_COSTS(4, 4)	4282.000	0.000000
OPERATING_COSTS(4, 5)	4284.000	0.000000
OPERATING_COSTS(4, 6)	4314.000	0.000000
OPERATING_COSTS(4, 7)	4408.000	0.000000
OPERATING_COSTS(5, 1)	4096.000	0.000000
OPERATING_COSTS(5, 2)	4089.000	0.000000
OPERATING_COSTS(5, 3)	4169.000	0.000000
OPERATING_COSTS(5, 4)	4167.000	0.000000
OPERATING_COSTS(5, 5)	4135.000	0.000000
OPERATING_COSTS(5, 6)	4171.000	0.000000
OPERATING_COSTS(5, 7)	4296.000	0.000000
OPERATING_COSTS(6, 1)	4913.000	0.000000
OPERATING_COSTS(6, 2)	4953.000	0.000000
OPERATING_COSTS(6, 3)	4836.000	0.000000
OPERATING_COSTS(6, 4)	4935.000	0.000000
OPERATING_COSTS(6, 5)	4920.000	0.000000
OPERATING_COSTS(6, 6)	4921.000	0.000000
OPERATING_COSTS(6, 7)	4953.000	0.000000
OPERATING_COSTS(7, 1)	5241.000	0.000000
OPERATING_COSTS(7, 2)	5310.000	0.000000
OPERATING_COSTS(7, 3)	5087.000	0.000000
OPERATING_COSTS(7, 4)	5203.000	0.000000
OPERATING_COSTS(7, 5)	5262.000	0.000000
OPERATING_COSTS(7, 6)	5246.000	0.000000

OPERATING_COSTS(7, 7)	5207.000	0.000000
OPERATING_COSTS(8, 1)	4195.000	0.000000
OPERATING_COSTS(8, 2)	4227.000	0.000000
OPERATING_COSTS(8, 3)	4146.000	0.000000
OPERATING_COSTS(8, 4)	4227.000	0.000000
OPERATING_COSTS(8, 5)	4208.000	0.000000
OPERATING_COSTS(8, 6)	4216.000	0.000000
OPERATING_COSTS(8, 7)	4263.000	0.000000
OPERATING_COSTS(9, 1)	5096.000	0.000000
OPERATING_COSTS(9, 2)	5152.000	0.000000
OPERATING_COSTS(9, 3)	4958.000	0.000000
OPERATING_COSTS(9, 4)	5100.000	0.000000
OPERATING_COSTS(9, 5)	5079.000	0.000000
OPERATING_COSTS(9, 6)	5065.000	0.000000
OPERATING_COSTS(9, 7)	5079.000	0.000000
OPERATING_COSTS(10, 1)	2874.000	0.000000
OPERATING_COSTS(10, 2)	2841.000	0.000000
OPERATING_COSTS(10, 3)	2964.000	0.000000
OPERATING_COSTS(10, 4)	3001.000	0.000000
OPERATING_COSTS(10, 5)	2849.000	0.000000
OPERATING_COSTS(10, 6)	2883.000	0.000000
OPERATING_COSTS(10, 7)	3106.000	0.000000
OPERATING_COSTS(11, 1)	3722.000	0.000000
OPERATING_COSTS(11, 2)	2784.000	0.000000
OPERATING_COSTS(11, 3)	3513.000	0.000000
OPERATING_COSTS(11, 4)	3695.000	0.000000
OPERATING_COSTS(11, 5)	3678.000	0.000000
OPERATING_COSTS(11, 6)	3639.000	0.000000
OPERATING_COSTS(11, 7)	3659.000	0.000000
N_VOYAGES_PER_YEAR(1, 1)	8.490000	0.000000
N_VOYAGES_PER_YEAR(1, 2)	10.96000	0.000000

N_VOYAGES_PER_YEAR(1, 3)	9.100000	0.000000
N_VOYAGES_PER_YEAR(1, 4)	4.530000	0.000000
N_VOYAGES_PER_YEAR(1, 5)	6.520000	0.000000
N_VOYAGES_PER_YEAR(1, 6)	4.460000	0.000000
N_VOYAGES_PER_YEAR(1, 7)	10.40000	0.000000
N_VOYAGES_PER_YEAR(2, 1)	8.870000	0.000000
N_VOYAGES_PER_YEAR(2, 2)	11.48000	0.000000
N_VOYAGES_PER_YEAR(2, 3)	9.660000	0.000000
N_VOYAGES_PER_YEAR(2, 4)	4.740000	0.000000
N_VOYAGES_PER_YEAR(2, 5)	6.900000	0.000000
N_VOYAGES_PER_YEAR(2, 6)	4.750000	0.000000
N_VOYAGES_PER_YEAR(2, 7)	10.84000	0.000000
N_VOYAGES_PER_YEAR(3, 1)	8.870000	0.000000
N_VOYAGES_PER_YEAR(3, 2)	11.48000	0.000000
N_VOYAGES_PER_YEAR(3, 3)	9.660000	0.000000
N_VOYAGES_PER_YEAR(3, 4)	4.740000	0.000000
N_VOYAGES_PER_YEAR(3, 5)	6.900000	0.000000
N_VOYAGES_PER_YEAR(3, 6)	4.750000	0.000000
N_VOYAGES_PER_YEAR(3, 7)	10.84000	0.000000
N_VOYAGES_PER_YEAR(4, 1)	8.750000	0.000000
N_VOYAGES_PER_YEAR(4, 2)	11.31000	0.000000
N_VOYAGES_PER_YEAR(4, 3)	9.480000	0.000000
N_VOYAGES_PER_YEAR(4, 4)	4.680000	0.000000
N_VOYAGES_PER_YEAR(4, 5)	6.780000	0.000000
N_VOYAGES_PER_YEAR(4, 6)	4.650000	0.000000
N_VOYAGES_PER_YEAR(4, 7)	10.70000	0.000000
N_VOYAGES_PER_YEAR(5, 1)	8.750000	0.000000
N_VOYAGES_PER_YEAR(5, 2)	11.31000	0.000000
N_VOYAGES_PER_YEAR(5, 3)	9.480000	0.000000
N_VOYAGES_PER_YEAR(5, 4)	4.680000	0.000000
N_VOYAGES_PER_YEAR(5, 5)	6.780000	0.000000

N_VOYAGES_PER_YEAR(5, 6)	4.650000	0.000000
N_VOYAGES_PER_YEAR(5, 7)	10.70000	0.000000
N_VOYAGES_PER_YEAR(6, 1)	8.870000	0.000000
N_VOYAGES_PER_YEAR(6, 2)	11.48000	0.000000
N_VOYAGES_PER_YEAR(6, 3)	9.660000	0.000000
N_VOYAGES_PER_YEAR(6, 4)	4.740000	0.000000
N_VOYAGES_PER_YEAR(6, 5)	6.900000	0.000000
N_VOYAGES_PER_YEAR(6, 6)	4.750000	0.000000
N_VOYAGES_PER_YEAR(6, 7)	10.84000	0.000000
N_VOYAGES_PER_YEAR(7, 1)	8.870000	0.000000
N_VOYAGES_PER_YEAR(7, 2)	11.48000	0.000000
N_VOYAGES_PER_YEAR(7, 3)	9.660000	0.000000
N_VOYAGES_PER_YEAR(7, 4)	4.740000	0.000000
N_VOYAGES_PER_YEAR(7, 5)	6.900000	0.000000
N_VOYAGES_PER_YEAR(7, 6)	4.750000	0.000000
N_VOYAGES_PER_YEAR(7, 7)	10.84000	0.000000
N_VOYAGES_PER_YEAR(8, 1)	8.870000	0.000000
N_VOYAGES_PER_YEAR(8, 2)	11.48000	0.000000
N_VOYAGES_PER_YEAR(8, 3)	9.660000	0.000000
N_VOYAGES_PER_YEAR(8, 4)	4.740000	0.000000
N_VOYAGES_PER_YEAR(8, 5)	6.900000	0.000000
N_VOYAGES_PER_YEAR(8, 6)	4.750000	0.000000
N_VOYAGES_PER_YEAR(8, 7)	10.84000	0.000000
N_VOYAGES_PER_YEAR(9, 1)	8.870000	0.000000
N_VOYAGES_PER_YEAR(9, 2)	11.48000	0.000000
N_VOYAGES_PER_YEAR(9, 3)	9.660000	0.000000
N_VOYAGES_PER_YEAR(9, 4)	4.740000	0.000000
N_VOYAGES_PER_YEAR(9, 5)	6.900000	0.000000
N_VOYAGES_PER_YEAR(9, 6)	4.750000	0.000000
N_VOYAGES_PER_YEAR(9, 7)	10.84000	0.000000
N_VOYAGES_PER_YEAR(10, 1)	8.210000	0.000000

N_VOYAGES_PER_YEAR(10, 2)	10.58000	0.000000
N_VOYAGES_PER_YEAR(10, 3)	8.700000	0.000000
N_VOYAGES_PER_YEAR(10, 4)	4.380000	0.000000
N_VOYAGES_PER_YEAR(10, 5)	6.250000	0.000000
N_VOYAGES_PER_YEAR(10, 6)	4.250000	0.000000
N_VOYAGES_PER_YEAR(10, 7)	10.08000	0.000000
N_VOYAGES_PER_YEAR(11, 1)	8.210000	0.000000
N_VOYAGES_PER_YEAR(11, 2)	10.58000	0.000000
N_VOYAGES_PER_YEAR(11, 3)	8.700000	0.000000
N_VOYAGES_PER_YEAR(11, 4)	4.380000	0.000000
N_VOYAGES_PER_YEAR(11, 5)	6.250000	0.000000
N_VOYAGES_PER_YEAR(11, 6)	4.250000	0.000000
N_VOYAGES_PER_YEAR(11, 7)	10.08000	0.000000
VOYAGE_TIME(1, 1)	0.000000	0.000000
VOYAGE_TIME(1, 2)	0.000000	0.000000
VOYAGE_TIME(1, 3)	0.000000	0.000000
VOYAGE_TIME(1, 4)	0.000000	0.000000
VOYAGE_TIME(1, 5)	0.000000	0.000000
VOYAGE_TIME(1, 6)	0.000000	0.000000
VOYAGE_TIME(1, 7)	0.000000	0.000000
VOYAGE_TIME(2, 1)	0.000000	0.000000
VOYAGE_TIME(2, 2)	0.000000	0.000000
VOYAGE_TIME(2, 3)	0.000000	0.000000
VOYAGE_TIME(2, 4)	0.000000	0.000000
VOYAGE_TIME(2, 5)	0.000000	0.000000
VOYAGE_TIME(2, 6)	0.000000	0.000000
VOYAGE_TIME(2, 7)	0.000000	0.000000
VOYAGE_TIME(3, 1)	0.000000	0.000000
VOYAGE_TIME(3, 2)	0.000000	0.000000
VOYAGE_TIME(3, 3)	0.000000	0.000000
VOYAGE_TIME(3, 4)	0.000000	0.000000

VOYAGE_TIME(3, 5)	0.000000	0.000000
VOYAGE_TIME(3, 6)	0.000000	0.000000
VOYAGE_TIME(3, 7)	0.000000	0.000000
VOYAGE_TIME(4, 1)	0.000000	0.000000
VOYAGE_TIME(4, 2)	0.000000	0.000000
VOYAGE_TIME(4, 3)	0.000000	0.000000
VOYAGE_TIME(4, 4)	0.000000	0.000000
VOYAGE_TIME(4, 5)	0.000000	0.000000
VOYAGE_TIME(4, 6)	0.000000	0.000000
VOYAGE_TIME(4, 7)	0.000000	0.000000
VOYAGE_TIME(5, 1)	0.000000	0.000000
VOYAGE_TIME(5, 2)	0.000000	0.000000
VOYAGE_TIME(5, 3)	0.000000	0.000000
VOYAGE_TIME(5, 4)	0.000000	0.000000
VOYAGE_TIME(5, 5)	0.000000	0.000000
VOYAGE_TIME(5, 6)	0.000000	0.000000
VOYAGE_TIME(5, 7)	0.000000	0.000000
VOYAGE_TIME(6, 1)	0.000000	0.000000
VOYAGE_TIME(6, 2)	0.000000	0.000000
VOYAGE_TIME(6, 3)	0.000000	0.000000
VOYAGE_TIME(6, 4)	0.000000	0.000000
VOYAGE_TIME(6, 5)	0.000000	0.000000
VOYAGE_TIME(6, 6)	0.000000	0.000000
VOYAGE_TIME(6, 7)	0.000000	0.000000
VOYAGE_TIME(7, 1)	0.000000	0.000000
VOYAGE_TIME(7, 2)	0.000000	0.000000
VOYAGE_TIME(7, 3)	0.000000	0.000000
VOYAGE_TIME(7, 4)	0.000000	0.000000
VOYAGE_TIME(7, 5)	0.000000	0.000000
VOYAGE_TIME(7, 6)	0.000000	0.000000
VOYAGE_TIME(7, 7)	0.000000	0.000000

VOYAGE_TIME(8, 1)	0.000000	0.000000
VOYAGE_TIME(8, 2)	0.000000	0.000000
VOYAGE_TIME(8, 3)	0.000000	0.000000
VOYAGE_TIME(8, 4)	0.000000	0.000000
VOYAGE_TIME(8, 5)	0.000000	0.000000
VOYAGE_TIME(8, 6)	0.000000	0.000000
VOYAGE_TIME(8, 7)	0.000000	0.000000
VOYAGE_TIME(9, 1)	0.000000	0.000000
VOYAGE_TIME(9, 2)	0.000000	0.000000
VOYAGE_TIME(9, 3)	0.000000	0.000000
VOYAGE_TIME(9, 4)	0.000000	0.000000
VOYAGE_TIME(9, 5)	0.000000	0.000000
VOYAGE_TIME(9, 6)	0.000000	0.000000
VOYAGE_TIME(9, 7)	0.000000	0.000000
VOYAGE_TIME(10, 1)	0.000000	0.000000
VOYAGE_TIME(10, 2)	0.000000	0.000000
VOYAGE_TIME(10, 3)	0.000000	0.000000
VOYAGE_TIME(10, 4)	0.000000	0.000000
VOYAGE_TIME(10, 5)	0.000000	0.000000
VOYAGE_TIME(10, 6)	0.000000	0.000000
VOYAGE_TIME(10, 7)	0.000000	0.000000
VOYAGE_TIME(11, 1)	0.000000	0.000000
VOYAGE_TIME(11, 2)	0.000000	0.000000
VOYAGE_TIME(11, 3)	0.000000	0.000000
VOYAGE_TIME(11, 4)	0.000000	0.000000
VOYAGE_TIME(11, 5)	0.000000	0.000000
VOYAGE_TIME(11, 6)	0.000000	0.000000
VOYAGE_TIME(11, 7)	0.000000	0.000000
VOYAGES_PER_YEAR(1, 1)	0.000000	0.000000
VOYAGES_PER_YEAR(1, 2)	0.000000	0.000000
VOYAGES_PER_YEAR(1, 3)	0.000000	0.000000

VOYAGES_PER_YEAR(1, 4)	0.000000	0.000000
VOYAGES_PER_YEAR(1, 5)	0.000000	0.000000
VOYAGES_PER_YEAR(1, 6)	0.000000	0.000000
VOYAGES_PER_YEAR(1, 7)	0.000000	0.000000
VOYAGES_PER_YEAR(2, 1)	0.000000	0.000000
VOYAGES_PER_YEAR(2, 2)	0.000000	0.000000
VOYAGES_PER_YEAR(2, 3)	0.000000	0.000000
VOYAGES_PER_YEAR(2, 4)	0.000000	0.000000
VOYAGES_PER_YEAR(2, 5)	0.000000	0.000000
VOYAGES_PER_YEAR(2, 6)	0.000000	0.000000
VOYAGES_PER_YEAR(2, 7)	0.000000	0.000000
VOYAGES_PER_YEAR(3, 1)	0.000000	0.000000
VOYAGES_PER_YEAR(3, 2)	0.000000	0.000000
VOYAGES_PER_YEAR(3, 3)	0.000000	0.000000
VOYAGES_PER_YEAR(3, 4)	0.000000	0.000000
VOYAGES_PER_YEAR(3, 5)	0.000000	0.000000
VOYAGES_PER_YEAR(3, 6)	0.000000	0.000000
VOYAGES_PER_YEAR(3, 7)	0.000000	0.000000
VOYAGES_PER_YEAR(4, 1)	0.000000	0.000000
VOYAGES_PER_YEAR(4, 2)	0.000000	0.000000
VOYAGES_PER_YEAR(4, 3)	0.000000	0.000000
VOYAGES_PER_YEAR(4, 4)	0.000000	0.000000
VOYAGES_PER_YEAR(4, 5)	0.000000	0.000000
VOYAGES_PER_YEAR(4, 6)	0.000000	0.000000
VOYAGES_PER_YEAR(4, 7)	0.000000	0.000000
VOYAGES_PER_YEAR(5, 1)	0.000000	0.000000
VOYAGES_PER_YEAR(5, 2)	0.000000	0.000000
VOYAGES_PER_YEAR(5, 3)	0.000000	0.000000
VOYAGES_PER_YEAR(5, 4)	0.000000	0.000000
VOYAGES_PER_YEAR(5, 5)	0.000000	0.000000
VOYAGES_PER_YEAR(5, 6)	0.000000	0.000000

VOYAGES_PER_YEAR(5, 7)	0.000000	0.000000
VOYAGES_PER_YEAR(6, 1)	0.000000	0.000000
VOYAGES_PER_YEAR(6, 2)	0.000000	0.000000
VOYAGES_PER_YEAR(6, 3)	0.000000	0.000000
VOYAGES_PER_YEAR(6, 4)	0.000000	0.000000
VOYAGES_PER_YEAR(6, 5)	0.000000	0.000000
VOYAGES_PER_YEAR(6, 6)	0.000000	0.000000
VOYAGES_PER_YEAR(6, 7)	0.000000	0.000000
VOYAGES_PER_YEAR(7, 1)	0.000000	0.000000
VOYAGES_PER_YEAR(7, 2)	0.000000	0.000000
VOYAGES_PER_YEAR(7, 3)	0.000000	0.000000
VOYAGES_PER_YEAR(7, 4)	0.000000	0.000000
VOYAGES_PER_YEAR(7, 5)	0.000000	0.000000
VOYAGES_PER_YEAR(7, 6)	0.000000	0.000000
VOYAGES_PER_YEAR(7, 7)	0.000000	0.000000
VOYAGES_PER_YEAR(8, 1)	0.000000	0.000000
VOYAGES_PER_YEAR(8, 2)	0.000000	0.000000
VOYAGES_PER_YEAR(8, 3)	0.000000	0.000000
VOYAGES_PER_YEAR(8, 4)	0.000000	0.000000
VOYAGES_PER_YEAR(8, 5)	0.000000	0.000000
VOYAGES_PER_YEAR(8, 6)	0.000000	0.000000
VOYAGES_PER_YEAR(8, 7)	0.000000	0.000000
VOYAGES_PER_YEAR(9, 1)	0.000000	0.000000
VOYAGES_PER_YEAR(9, 2)	0.000000	0.000000
VOYAGES_PER_YEAR(9, 3)	0.000000	0.000000
VOYAGES_PER_YEAR(9, 4)	0.000000	0.000000
VOYAGES_PER_YEAR(9, 5)	0.000000	0.000000
VOYAGES_PER_YEAR(9, 6)	0.000000	0.000000
VOYAGES_PER_YEAR(9, 7)	0.000000	0.000000
VOYAGES_PER_YEAR(10, 1)	0.000000	0.000000
VOYAGES_PER_YEAR(10, 2)	0.000000	0.000000

VOYAGES_PER_YEAR(10, 3)	0.000000	0.000000
VOYAGES_PER_YEAR(10, 4)	0.000000	0.000000
VOYAGES_PER_YEAR(10, 5)	0.000000	0.000000
VOYAGES_PER_YEAR(10, 6)	0.000000	0.000000
VOYAGES_PER_YEAR(10, 7)	0.000000	0.000000
VOYAGES_PER_YEAR(11, 1)	0.000000	0.000000
VOYAGES_PER_YEAR(11, 2)	0.000000	0.000000
VOYAGES_PER_YEAR(11, 3)	0.000000	0.000000
VOYAGES_PER_YEAR(11, 4)	0.000000	0.000000
VOYAGES_PER_YEAR(11, 5)	0.000000	0.000000
VOYAGES_PER_YEAR(11, 6)	0.000000	0.000000
VOYAGES_PER_YEAR(11, 7)	0.000000	0.000000

Row	Slack or Surplus	Dual Price
1	91831.00	-1.000000
2	0.000000	0.000000
3	0.000000	0.000000
4	0.000000	0.000000
5	0.000000	0.000000
6	0.000000	0.000000
7	0.000000	0.000000
8	0.000000	0.000000
9	2.000000	0.000000
10	2.000000	0.000000
11	0.000000	0.000000
12	1.000000	0.000000
13	0.4700000	0.000000
14	0.6200000	0.000000
15	0.3200000	0.000000
16	0.1000000E-01	0.000000
17	0.5000000	0.000000

18	0.000000	0.000000
19	0.4400000	0.000000
20	0.000000	-1681.000
21	0.000000	-1729.000
22	0.000000	-1729.000
23	0.000000	-1761.000
24	0.000000	-1577.500
25	0.000000	-1650.500
26	0.000000	-1648.500
27	0.000000	-1652.500
28	0.000000	-2102.000
29	0.000000	-4263.000
30	0.000000	-5079.000
31	0.000000	-2874.000
32	0.000000	-2841.000
33	0.000000	-2964.000
34	0.000000	-3001.000
35	0.000000	-2883.000
36	0.000000	-3106.000
37	0.000000	-3659.000
38	0.000000	-9.100000
39	0.000000	-9.000000
40	0.000000	-9.000000
41	0.000000	-7.400000
42	0.000000	-7.300000
43	0.000000	-8.900000
44	0.000000	-9.000000
45	0.000000	0.000000
46	0.000000	0.000000
47	0.000000	0.000000
48	0.000000	0.000000

49	3.000000	0.000000
50	0.000000	0.000000
51	0.000000	0.000000
52	3.000000	0.000000
53	0.000000	0.000000
54	0.000000	0.000000
55	0.000000	0.000000
56	0.000000	0.000000
57	0.000000	0.000000
58	0.000000	0.000000
59	0.000000	0.000000
60	0.000000	0.000000
61	1.000000	0.000000
62	1.000000	0.000000
63	0.000000	0.000000
64	0.000000	0.000000
65	0.000000	0.000000
66	1.000000	0.000000
67	0.000000	0.000000
68	2.000000	0.000000
69	0.000000	0.000000
70	0.000000	0.000000
71	1.000000	0.000000
72	0.000000	0.000000
73	0.000000	0.000000
74	0.000000	0.000000
75	0.000000	0.000000
76	0.000000	0.000000
77	0.000000	0.000000
78	1.000000	0.000000
79	0.000000	0.000000

80	0.000000	0.000000
81	0.000000	0.000000
82	0.000000	0.000000
83	0.000000	0.000000
84	0.000000	0.000000
85	0.000000	0.000000
86	0.000000	0.000000
87	0.000000	0.000000
88	0.000000	0.000000
89	1.000000	0.000000
90	0.000000	0.000000
91	0.000000	0.000000
92	0.000000	0.000000
93	2.000000	0.000000
94	0.000000	0.000000
95	0.000000	0.000000
96	0.000000	0.000000
97	0.000000	0.000000
98	0.000000	0.000000
99	0.000000	0.000000
100	0.000000	0.000000
101	0.000000	0.000000
102	0.000000	0.000000
103	0.000000	0.000000
104	0.000000	0.000000
105	0.000000	0.000000
106	0.000000	0.000000
107	0.000000	0.000000
108	0.000000	0.000000
109	0.000000	0.000000
110	0.000000	0.000000

111	0.000000	0.000000
112	0.000000	0.000000
113	0.000000	0.000000
114	0.000000	0.000000
115	0.000000	0.000000
116	2.000000	0.000000
117	0.000000	0.000000
118	0.000000	0.000000
119	0.000000	0.000000
120	0.000000	0.000000
121	0.000000	0.000000
122	1.000000	0.000000
123	0.000000	0.000000
124	0.000000	0.000000
125	0.000000	0.000000
126	120.0000	0.000000
127	40.00000	0.000000
128	60.00000	0.000000
129	20.00000	0.000000
130	20.00000	0.000000
131	20.00000	0.000000
132	40.00000	0.000000
133	730.0000	0.000000
134	730.0000	0.000000
135	40.00000	0.000000
136	385.0000	0.000000

It may be useful if we clarify the examples below.

Reduced Cost:

a) you may interpret a variable's reduced cost as the amount that the objective function of the variable would have to improve before it would become profitable to give the variable

in question a positive value in the optimal solution. For example, if a variable had a reduced cost of 10, the objective coefficient of that variable would have to increase by 10 units and/or decrease by 10 units in a minimization problem for the variable to become an attractive alternative to enter into the solution.

b) it may be interpreted as the amount of penalty you would have to pay to introduce one unit of that variable into the solution. Again, if you have a variable with a reduced cost of 10, you would have to pay a penalty of 10 units to introduce the variable into the solution. In other words, the objective value would fall by 10 units in a maximization model or increase by 10 units in a minimization model.

Slack or Surplus:

The Slack or Surplus tells you how close you are to satisfying a constraint as an equality. This quantity, on (\leq) constraints, is generally referred to as slack. On (\geq), this quantity is called a surplus.

If a constraint is exactly satisfied as an equality, the slack or surplus value will be zero. If a constraint is violated, as in an infeasible solution, the slack or surplus value will be negative.

Dual Price (Shadow Price)

LINGO gives a dual price for each constraint. You can interpret the dual price as the amount that the objective would improve as the right-hand side, or constant term, of the constraint is increased by one unit.

In a maximization problem, improve means the objective value would increase. However, in a minimization problem, the objective value would decrease if you were to increase the right-hand side of a constraint with a positive dual price.

Dual prices are sometimes called shadow prices, because they tell you how much you should be willing to pay for additional units of a resource.