OPTIMAL CONTAINER LOADING

by

ANASTASIOS HARALAMPOS ASLIDIS

Diploma, National Technical University of Athens, GREECE
(1983)

Submitted to the Department of
Ocean Engineering
in Partial Fulfillment of the
Requirements of the Degree of

MASTER OF SCIENCE IN OCEAN SYSTEMS MANAGEMENT

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1984

Signature of Author Anastasios Aslidis
Department of Ocean Engineering
May 21, 1984

Certified by H. Psaraftis
Thesis Supervisor

Accepted by A Douglas Carmichael
Chairman, Ocean Engineering Departmental Committee

ABSTRACT

The problem of minimization of container handling cost is considered. This cost comes from both container overstowage and horizontal crane movements. This thesis mainly deals with the overstowage cost, which is more significant. The cost of horizontal crane movements is considered indirectly by "blocking" containers from the same origin or to the same destination.

An algorithm is suggested to allocate containers on board a ship through the satisfaction of the trim and metacentric height (GM) requirements. No other hydrostatic or placement restrictions are taken into account; however, their consideration is not expected to change the basic idea of the algorithm.

First, containers are assigned to available positions via the satisfaction of the trim requirement, so that the incurred overstowage cost is zero. Then, the GM of the vessel is calculated and compared with the minimum required. In case the requirement is not satisfied, transverse container interchanges take place moving the heavier containers to lower positions. The use of water ballast is kept as minimum as possible.

The main advantage of the suggested algorithm seems to be that it is faster than other existing allocation procedures. This happens because the algorithm does not use dynamic or integer programming methods.

A computer program has been developed to carry out the necessary operations. The required CPU time is of the order of 15 seconds for 700 containers in a VAX 11/782 computer depending almost linearly on the number of containers to be loaded.

The method is tested with satisfactory results in the case of a vessel visiting a series of ports. However, many opportunities for further research are open.

Supervisor: Prof. H. N. Psaraftis

Thesis title: Optimal container loading

AKNOWLEDGEMENTS

Special thanks are extended to Professor H.N. Psaraftis for supervising this thesis and to Prof. C. Chrissostomidis for providing the design of the containership used for the test of the algorithm.

Also, I would like to thank Ms. Georgia Melenikiotou for her help in editing and typing this thesis.

May 1984

Anastasios Haralampos Aslidis

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Optimal container loading: motivation

The introduction of containerization in the shipping industry took place about 25 years ago, and since then it has continuously evolved. The size of containerships increased from 350 TEU fully cellular containerships to vessels with capacity of the order of 4000 TEU's. At the same time the size of the container itself increased from 10 and 20 foot containers to 35 and 40 foot ones.

Moreover, evolution has occurred not only in the size of vessels and containers, but also in the transportation system as a whole. The concept of intermodal transportation has proved to be economically and practically very efficient. Under this system the transportation network includes both ocean and land routes. Usually the container is loaded and sealed by the shipper and driven by a truck or a train to the port, where a ship takes over moving the container to its ultimate shoreside destination. From there the container is carried by truck or rail for final delivery.

Both competition and increasing shipping costs require every stage of the whole procedure to be carried out efficiently. With vessels carrying a large number of containers today, the time and, consequently, the cost, of loading and unloading them contributes a continuously

---

TEU: Twenty-foot equvalent unit (container)

increasing share of the overall cost.

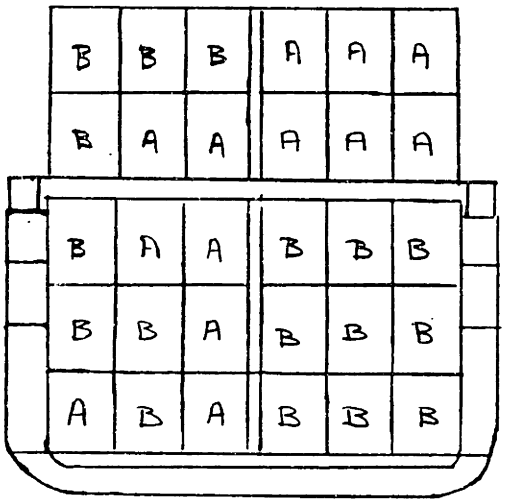
For any given port facilities, the time required for loading and unloading is a function of the arrangement of the cargo on board the vessel (vessel stowage). Both ship operators and port managers are interested in determining the optimal vessel stowage, that is, the one which minimizes port time.

The main reason for time delays is container overstowage; this happens when containers shipped to a port (say, A) are placed below containers shipped to a subsequent port (say, B - see figure 1.1). This may be necessary to satisfy the minimum metacentric height (GM) requirement in the case the containers shipped to port A are heavier than the ones shipped to port B. The result is that some of the containers going to port B must be unloaded and loaded again in order to be able to unload the containers shipped to port A. That is what is known as "overstowage cost".

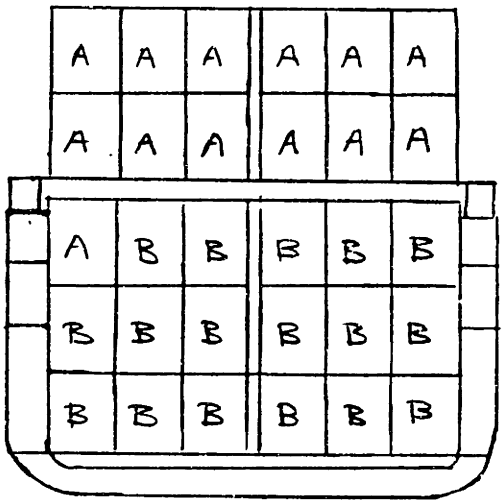
Another factor which affects the (un)loading time is the distribution of the containers along the vessel. If containers with the same destination are spread along the vessel, additional crane movements are necessary, resulting in longer port time. Such delays can be avoided if containers with common destination are "blocked" together.

In the following we are concerned with the problem of minimizing the container handling costs.

1.2 Literature survey



(a) Container allocation with overstowage.



(b) Container allocation with no overstowage.

FIGURE 1.1 Container overstowage

Since containerization is a fairly new phenomenon, few attempts have been made to solving the problem of optimal vessel stowage. Consequently, the related literature is limited. In addition, even though some attempts have been made by some companies, these have not been published in the open literature because they are considered as proprietary information.

The first attempts to solving the problem can be traced back to 1969, when Van Dyke and Webster tried to introduce the use of computers in cargo handling. Although the Maritime Administration supported their research, nothing commercially exploitable surfaced, because, as Scott and Chen claimed in a later paper (1978), the heuristic method which was developed neglected the constraints concerning the decking, racking and lashing strength and the regulations on hazardous cargo transportation.

In fact, the simultaneous consideration of all possible constraints makes the problem extremely difficult. Scott and Chen (1978) adopted three heuristic rules which tried to satisfy implicitly these constraints, as follows:

Containers were aggregated into homogeneous groups based on some container characteristics (such as type, length, height, weight, racking strength and destination) and also on placement restrictions. Ten classes of containers were created. The first nine classes included containers with specific requirements while the tenth class contained containers suitable to be placed anywhere on the vessel. The

containers of the tenth class were stratified into several weight brackets. Dynamic programming methods were used to determine the ranges of the brackets.

The allocation procedure had four stages:

stage 1: The containers of the first nine classes were assigned to individual positions by following three heuristic rules.

stage 2: The containers of the tenth class were distributed to stations by using an integer programming model. The objective in this stage was to maximize the number of containers to be loaded.

stage 3: The allocation of individual containers within each station was determined. Integer programming was used to minimize the transverse moment.

stage 4: The trim, transverse moment and GM were checked. If the corresponding constraints were violated,container interchanges took place until all the constraints are satisfied. If this was not possible, the number of containers on board was reduced by one and the process was repeated.

Among the advantages of the above heuristic was the consideration of as many placement restrictions as possible. But it did not deal directly with overstowage, bending moment or decking or lashing strength. Actually, except for those checked in stage 4, none of the other constraints were satisfied directly. In addition, the method used dynamic and integer programming models, which undoubtedly resulted in

relatively long computer times. A typical integer problem size, as it is referred in the paper, had 29 constraints and 84 integer 0-1 variables. Although the required computer time was not discussed, it is expected to be relatively long. This seems to be the most pronounced disadvantage of the method.

A later work, Shields (1983), sponsored by American President Lines, was presented at a SNAME section meeting in California. The basic idea in this approach was the random generation and evaluation of many different possible loadings. In order to avoid a large number of loadings many of which are presumably not optimal, the loading generation process was biased in a manner to produce good results.

The criterion through which a specific loading was generated was selected randomly among a set of criteria. More specifically, each criterion was assigned a weight and a ranom number generator selected the hierarchy under which the criterion would be applied for the loading of the next group of containers.

The evaluation of the loading was done by imposing penalties each time an increase in the container handling cost occurred or each time a constraint was violated. In the latter case the related penalty was very large, to preclude the selection of that loading as optimal.

Finally the three top (less costly) solutions were approved. The algorithm commenced the loading of containers of the next port using the three selected loadings as starting points. Again the three best loadings were chosen. The

procedure was repeated at each subsequent port. At the final port the less costly solution as well as the intermidiate loadings which result were found and adopted. A diagramatic representation of the algorithm is shown in figure 1.2. The shaded ships represent the selected loadings. The doted line indicates the final loading chosen.

This algorithm took into account many parameters and restrictions and satisfied them fairly well. But it did not guarantee optimality even in a relative sense, since there was nothing to secure that the best combinations after, say, port #1 would result in an overall optimum at the final port. Moreover, the three selected solutions at each port were not necessarily the best. All the above are true in a probabilistic sense. Of course if the number of combinations checked as well as the number of loadings approved to continue to the next ports are increased, the probability to find a better solution increases. But this also results in longer computer times.

1.3 Outline of the thesis

In this thesis we present a simpler approach to the container overstowage problem. The algorithm proposed, although in its first stages of development, seems to be very promising, at least with respect to computer time and memory required.

Although a simplified version of the problem is solved

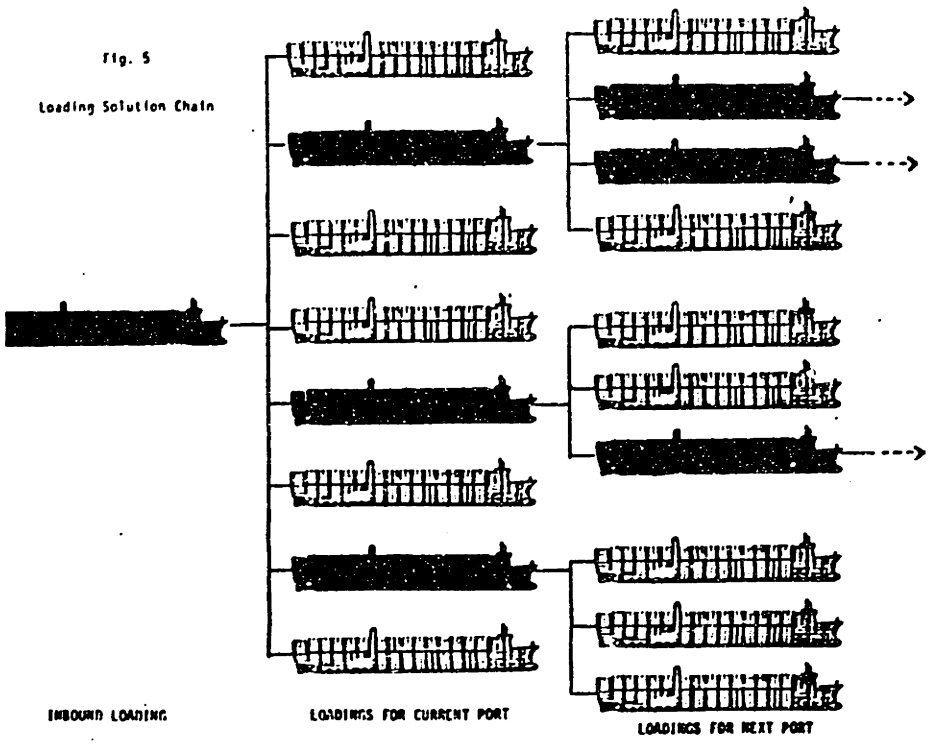


FIGURE I.2 Algorithm proposed by J.Shields: Selection of the three top loadings to continue the rest of the route.

here, the introduction of the complete set of constraints is not expected to change the nature of the algorithm, or significantly complicate the computer program.

The main advantage of the proposed algorithm is that it does not use integer or dynamic programming methods, which by their nature are time consuming, but follows a simple heuristic allocation method.

After some notation is adopted, the complete problem is described in chapter 2. In the same chapter we discuss some simplified assumptions and we conclude with the version of the problem we shall solve.

The next chapter is devoted to the analytical description of the proposed algorithm. Chapter 4 discusses the computer program developed. A discussion of the results along with the presentation of sample outputs of the program are made in chapter 5. Suggested fields of further research is the subject of chapter 6.

Finally, in appendix A1 a user's manual for the program is given. The listing of the computer program can be found in appendix A2.

CHAPTER 2: DEFINITION OF THE PROBLEM

2.1 Notation

Before we describe of the different ways to formulate and solve the problem, it is useful to establish the specific notation to be followed in the rest of this thesis.

First we define the coordinate system in which the vessel is described. This system has its origin at the center plane at the point where the aft perpendicular intersects the base plane (see figure 2.1). The x-coordinate increases forward and the z-coordinate upward.

The following terms are defined with respect to the above system:

1) A cell or position is an individual container location in the vessel, defined by its (x,y,z) coordinates (figure 2.2).

2) A row is the set of positions (cells) which are on the same horizontal plane, defined by its z-coordinate (figure 2.3).

3) A station is the set of container positions on the same transverse plane, defined by its x-coordinate (figure 2.4).

4) A column or stack is the set of container positions on the same logitudinal and transverse planes, defined by its x and y coordinates (figure 2.5).

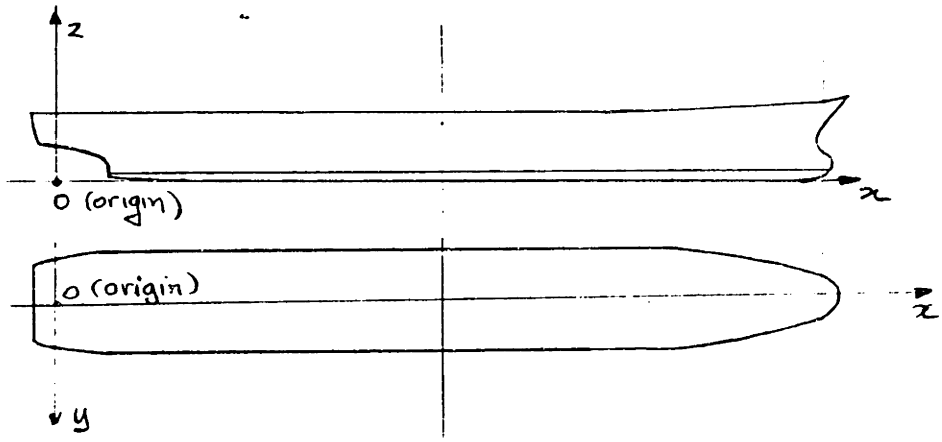


FIGURE 2.1. System of coordinates

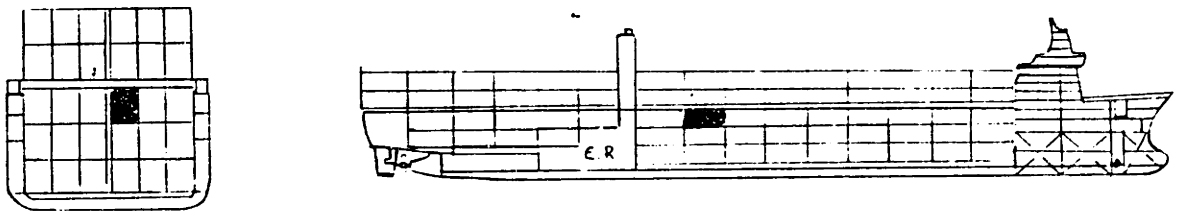


FIGURE 2.2 Cell or container position

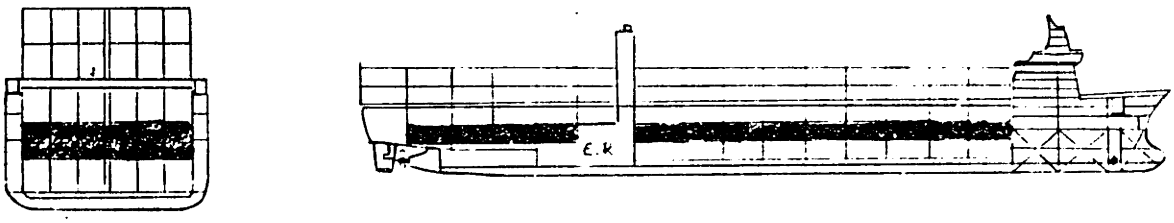


FIGURE 2.3 Row of containers

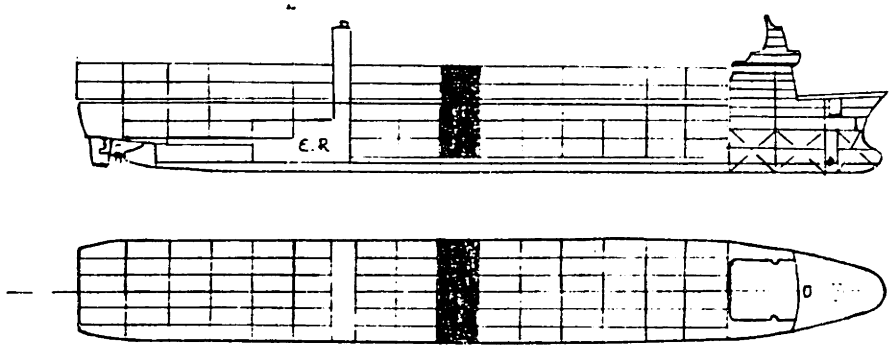


FIGURE 2.4 Station: containers with the same x-coordinate

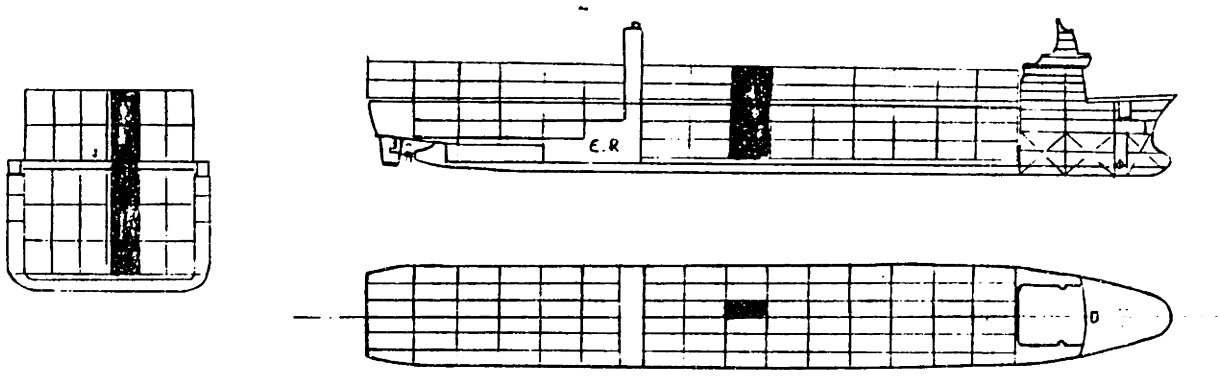


FIGURE 2.5 Column or stack of containers

2.2 The general problem

The optimal container loading problem in the general case can be stated as follows:

Suppose that a ship is scheduled to visit a number of ports, say N. Consider, now, a port -j- of the route. At that port a number Uj of containers are unloaded, while Lj new containers are loaded. In addition there may be a number Cj of containers which, although they have subsequent destinations, must be temporarily unloaded in order to discharge the Uj containers which have to be delivered at the current port.

The question posed is what is the loading arrangement of containers, so as to minimize the cost of handling them in the next ports. In other words it is desired that the total number of container rearrangements be as small as possible.

In general, the (un)loading procedure will be less costly if the crane horizontal movements are kept to a minimum. However, the main contribution to the container handling cost comes from container rearrangements due to overstowage.

In parallel with the cost minimization effort, a number of constraints must be satisfied. Generally, these constraints stem from both hydrostatic requirements of the vessel and placement restrictions with respect to the size, kind, content and strength of the containers. Some of these constraints are summarized below:

1) Longitudinal moment (trim)

The trim of the vessel must be between some prespecified

limits determined by requirements of good performance and safety. Generally, it is desired that the trim should be as close to zero as possible. In case trim is unavoidable, stern trim is prefered to bow trim.

In addition, the rules of classification societies relative to the minimum required bow and stern drafts must be satisfied.

2) Transverse moment

The transverse moment with respect to the center line must be zero or, at least, between very narrow limits around zero. The use of water ballast could be very helpful in satisfying this requirement.

3) Available metacentric height

The metacentric height (GM) of the vessel must be greater than the minimum required (GMmin). Satisfying this constraint implies placing the heavier containers at the lower rows.

This may conflict the objective of minimizing the container overstowage cost, if the heaviest containers are those which go to the nearest destinations.

The minimization of overstowage cost requires the containers which will be unloaded first to be placed at the upper rows. In case these containers happen to be the heaviest, the satisfaction of the GM constraint may require placing them at lower rows, increasing the container handling cost.

4) Bending moment (structural stresses)

The structural stresses allowed on the keel and on the deck are constrained to be no more than the maximum ones approved for the ship by the classification society. The allocation of the containers along the vessel affects the value of the structural stresses to the extent it influences the weight distribution along the ship.

Since the buoyancy distribution has its peak around amidships, a rule of thumb to reduce the value of the bending moment is to fill first the positions around the midsection.

5) Deck strength

It is possible that a stack of containers on deck weighs more than the deck structure can bear. In this case the weight (number) of containers per stack should be restricted.

6) Racking strength

Since above the deck there are no cell guides, the containers of the lower rows hold the containers stowed above them. If the weight of the stack exceeds the strength of the lower container, this may bend diagonally. This limit should be considered when containers are placed on deck.

7) Lashing strength

To avoid container movement, containers on deck are lashed to it. This implies that the stresses developed by containers of a stack should not exceed the stresses allowed by the lashing. This limits the number (weight) of containers per stack.

8) Refrigerated containers (reefers)

There are some categories of containers which have additional requirements, such as refrigerated containers or containers with cargo requiring ventilation etc. These containers can be placed only in positions having the related facilities (for example: reefers are usually placed at the lowest rows on deck (first or second) where power outlets are available).

9) Container support

Since a container is supported on its four lower corners, it must be placed in positions where either the ship or another container provides suitable support. This requirement does not allow containers with different lengths to be stowed in the same stack.

10) Hazardous cargo regulations

Containers which contain cargo considered hazardous (by the U.S. Coast Guard) must be seperated by a minimum distance from other containers also containing hazardous cargo (Code of Federal Regulations, Title 49, Transportation, Parts 100-199).

The above constraints are not of the same importance. Some of them are redundant or overlapping, as it happens with constraints 5, 6 and 7. Some others may be conflicting (1, 2 or 3 versus 8, 9, 10) and not all of them are present in all cases.

Moreover, there may be constraints stemming from draft

restrictions or different port facilities which can restrict the possibility to use the full capacity of the vessel.

Generally, the most important constraints are those related to the trim, stability and strength of the vessel (1,3 and 4), which are involved and must be satisfied in every ship departure.

It is clear that it is very difficult to solve the complete problem at once. This is the reason we chose to start from a relatively simplified version of it, including initially only the most important constraints. This will help us acquire a deeper understanding of the specific problems involved and also test our method. After this is done it would be easier to introduce the rest of the constraints.

The version of the problem which will be solved is described in the following section; for convenience we will call this version of the problem "Problem P".

2.2 The problem to be solved in this thesis

"Problem P"

It is necessary to simplify the general problem, as defined previously, at least in the first stages of the developement of the algorithm. Later, once we are convinced that we are in the correct direction we will return to the complete real case.

The simplifications made are presented and discussed in the following:

1) The first simplification we will make refers to the operations of the vessel at each port. We assume that along a route the vessel is only being loaded (Uj=0) or only being unloaded (Lj=0).

The former case means that the vessel starts empty at port (S) and picks up containers at subsequent ports. All those containers are unloaded at one common final destination (D). The latter case means exactly the reverse. The vessel departs loaded from a common origin (D) and delivers containers at all subsequent ports, without taking new cargo.

The two cases are equivalent because we can always treat the unloading case as the loading one and vice-versa. This can be seen more clearly in figure 2.6. We assume that a vessel visits N ports in a round trip. During its first visit it is being loaded with containers. In the return trip the vessel follows exactly the reverse route. This time, we assume that each container is unloaded at the port from which it had previously been loaded. The equivalence of the two cases is obvious.

2) The next simplification is related to the assumed objectives. Initially, we will optimize the loading with respect to overstowage cost. The cost of horizontal crane movements is relatively smaller than the overstowage cost and can be ignored. However, the algorithm offers the option to choose between "blocking" or "nonblocking" the containers from the same origin.

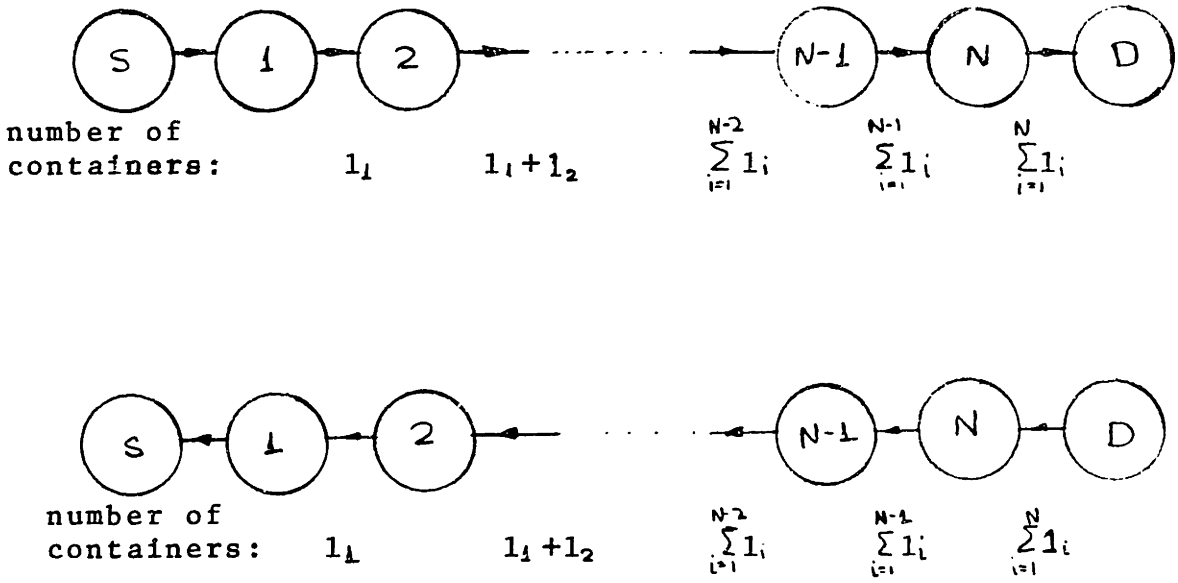


FIGURE 2.6 Equivalence of the loading and unloading case.

3) As far as the containers are concerned, we assume that:

i) they are of the same size (although not necessarily of the same weight),

ii) they do not contain hazardous or refrigerated cargo,

iii) their center of gravity coincides with the center of volume,

iv) the (un)loading schedule is known before the beginning of the trip.

4) The constraints related to decking, racking and lashing strength are ignored for the moment. We suppose that the number of rows on deck is such that these constraints are satisfied.

5) Moreover, the loading is considered symmetric with respect to the center line. This assumption enables us to disregard the constraint related to the transverse moment.

6) The calculation of the structural stresses on the deck and on the keel is not very simple. In order to calculate the bending moment we need not only the weight distribution along the vessel, but also the buyoancy distribution for two wave patterns (trough or peak at midsection with wave length equal to ship length) and several different drafts. The time constraint under which this thesis was carried out does not permit such a detailed analysis.

The algorithm we propose allocates the heaviest containers around amidships in an effort to offset the

buoyancy distribution which has greater values in that area.

Finally, we do not have to worry about the maximization of the number of containers on board, because under the assumptions 3,4 and 6 made above the only factor which can restrict them is their own weight. If the containers are relatively heavier than those the vessel is designed for, the loading procedure will end up with some empty cells. Thus, in case the containers waiting to be loaded violate the draft (displacement) constraint, a method must be developed to decide which containers will be shipped. In the following we will skip this problem, by supposing that the draft constraint is satisfied or that the loading procedure terminates the first time the above constraint is violated.

Concluding this chapter, we restate the version of the problem we plan to solve.

3.2.1 "Problem P"

Suppose that a vessel is to visit N ports. The vessel starts empty at port S and is loaded with Lj containers at port j (j=1,...,N). We assume that the loading schedule is known from the begining. All these containers will be unloaded at a common destination.

We need to find the allocation of the containers at each port so as to minimize the container overstowage cost and satisfy the following constraints:

1) The longitudinal moment must be between specified

limits determined by requirements of good performance and safety,

2) The available metacentric height must be greater than the minimum required.

CHAPTER 3. DESCRIPTION OF THE PROPOSED ALGORITHM

3.1 General description

As it has been discussed in the previous chapter, the final version of the problem requires the direct satisfaction of two constraints - trim and metacentric height (GM) - and the indirect satisfaction of another one - structural stresses (bending moment).

Our objective is to minimize the container handling cost. This cost may have several components, the most important being: a) the cost incurred by overstowage, and b) the cost incurred by crane movements along the length of the vessel. The overstowage cost is more significant and, thus, the proposed algorithm gives more emphasis to it.

We can mathematically formulate the problem if we define variables Xij as:

$$Xij = \begin{cases} 1 & \text{if container } j \text{ is assigned to cell } i \\ 0 & \text{otherwise} \end{cases}$$

The main difficulty of any mathematical formulation using the above variables is the large number of variables used; if MC is the number of available positions on board and NC is the number of containers we want to ship, MC*NC variables should be defined. It is clear that with vessels carrying more than 1000 containers, the number of the required variables is

greater than 1,000,000. This makes almost impossible the use of any formulation based on the above variables. Fewer variables are required in order to produce a solvable model.

The algorithm proposed by this thesis uses very few variables. These variables represent the available cells (positions) on the vessel. If such a cell is empty, the corresponding variable is equal to zero; if the cell is occupied the variable takes on an integer value corresponding to the origin or destination of the container.

Because the loading and unloading cases are equivalent (figure 2.6), we develop the algorithm only for the loading case.

At each port the containers are assigned to cells one by one. The whole procedure is carried in three stages:

a) First, the satisfaction of the trim constraint determines the longitudinal position of each container. Preferably, water ballast is used the least possible. Because the new containers are placed above the already loaded ones, overstowage is not recorded in this stage.

b) After all the containers to be loaded at a specific port are placed aboard, the algorithm calculates the metacentric height (GM) and compares it with the minimum required. If the metacentric height is below the minimum, container interchanges take place in the columns where heavier containers have been placed above lighter ones. Containers are replaced in decreasing order of weight. However, these

rearrangements result in overstowage.

If after all possible interchanges have been made and the GM-constraint is still unsatisfied, the algorithm examines the possibility of performing transverse interchanges among containers of each station. If the GM-constraint is still unsatisfied, the algorithm cannot determine a feasible solution.

c) Finally, the method applies transverse interchanges in order to maximize GM without increasing the container handling cost and/or to minimize the above cost without decreasing GM.

The logical diagram of the algorithm is presented in figure 3.1. Because logitudinal container interchanges are not considered, the final solution is only relatively optimal and in a local sense. This latter feature as well as the blocking of containers with the same origin or destination may be included in a further developed version of this method. In the following three sections we discuss the above stages.

3.2 STAGE ONE: Assignment of containers via the satisfaction of the trim constraint.

3.2.1 Initialization

We suppose that the vessel is fully equipped (fuel, provisions,complement etc) and ready to depart as soon as the containers have been loaded and the required water ballast has been placed.

PROPOSED ALGORITHM

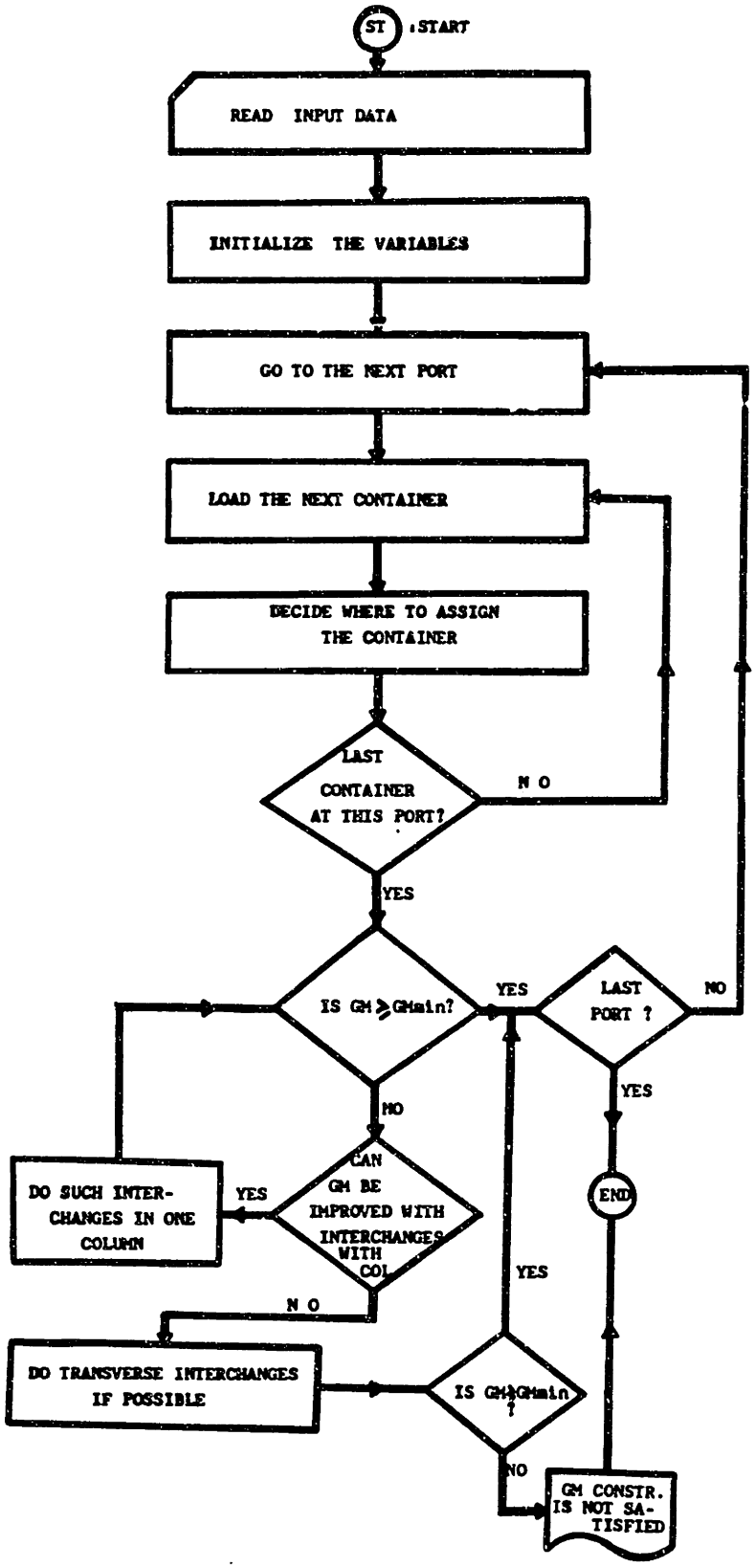


FIGURE 3.1 Flow chart of the proposed algorithm

Furthermore, we assume that the loading schedule as well as the characteristics of the containers (weight) in every port are known from the beginning of the trip.

By knowing which containers are to be loaded at the next port, we can approximately calculate the LCB and displacement. During the loading procedure, whenever water ballast is added or subtracted, the values of the above variables are revised. We call the values of LCB and displacement after the departure from port j final LCB (LCBj) and diplacement (Dj) at port j.

We can use either the trim or the longitudinal moment to express the trim constraint. In the following we find more convenient to use the longitudinal moment as a reference variable. If the moment is known the calculation of the trim does not present any difficulty. We define the moment which results in stern trim as negative; consequently, a moment resulting in bow trim is positive.

In the following we call front tanks the tanks lying between the LCB and bow; similarly, the tanks behind LCB are called rear tanks. We follow the same notation with regard to container positions (cells) as well.

In the begining of the method, the coordinates of the available container positions and some of the hydrostatic curves of the vessel (KM, LCB, MTC1) should be provided. The container loading schedule should also be provided.

Before the loading procedure commences we must decide which containers will be placed on deck. We will solve the

problems of assigning containers on and below deck seperately.

3.2.2 The loading procedure

The loading procedure starts by calculating the longitudinal moment with respect to the final LCB of the next port. Then containers are assigned to cells in front of or behind LCB, if, respectively, the longitudinal moment is negative or positive. Water ballast is used to bring the logitudinal moment within the permissible range, after all the containers of the next port are placed.

In case where the moment is between the permissible bounds and there is a tank containing water ballast, the container is placed at that part of the vessel where the tank is. By reducing the water ballast we keep the moment to its previous level.

Because we intend to minimize water ballast, we increase its amount only if we cannot balance the longitudinal moment by reducing it. By doing so, we have water ballast either in the front or rear tanks.

Figure 3.2 summarizes the criteria under which a container is assigned in front of or behind the LCB.

3.2.3 Some extreme cases

Although this is expected to happen rarely, it is possible to have the following allocation:

-all the front tanks are full

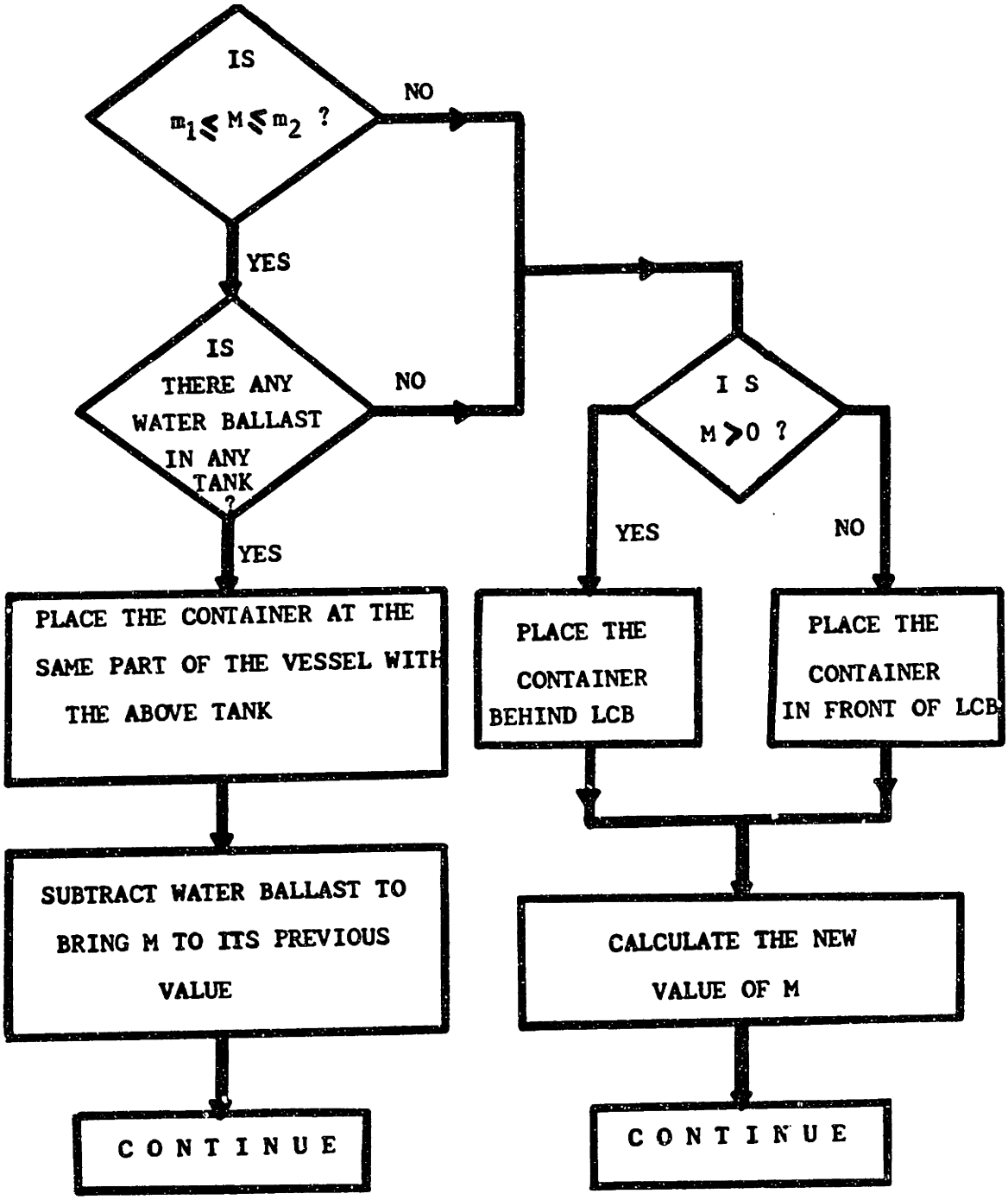


FIGURE3.2Assignment of containers to the front or rear part of the vessel.

-all the front positions are filled with relatively light containers

-there are some empty cells behind the LCB

-the value of the moment is less than the lower bound (stern trim)

In this case we cannot continue the loading procedure. In order to circumvent this difficulty, we can proceed as follows:

1) Interchange columns, by moving the heaviest columns towards the bow, or,

2) Attempt proportional completion of both the front and rear parts of the vessel.

The drawback of the first alternative is that it significantly increases the container handling cost. This results from the need to move already loaded container columns towards the bow. In addition the algorithm becomes too complicated and a feasible solution may not always be reachable.

The second alternative generally causes the use of more water ballast. However, the additional container handling cost is very small. This cost results from the additional crane movements caused by the container allocation in more stations than before. Nevertheless, such container allocation lowers the vertical center of weight of the cargo, and thus, results in greater GM. Moreover, this method results in an increase of the maximum possible GM, because since now more columns are

used, the number of equal-weight containers per column declines.

A mathematical expression of this alternative can be as follows:

Suppose that,

N1 is the number of the available positions behind of LCB

N2 is the number of the available positions ir front of LCB

X1 is the number of the already loaded containers behind LCB

X2 is the number of the already loaded containers in front of LCB

then,

X1/N1 is the percentage of the rear positions filled,

X2/N2 is the percentage of the front positions filled.

We can express the requirement that these percentages must differ at most by, say K, as:

$$X1/N1 - X2/N2 \leqslant K , \qquad 0 \leqslant K \leqslant 1$$

Obviously, if K=1, the new constraint is redundant.

If we adopt the second alternative we can use the difference in the percentages (K), as a parameter, making this criterion effective only when necessary. In the current version of the algorithm the second alternative is adopted. The parameter K is given exogenously.

3.2.4 Cell assignment procedure

The assignment to cells starts from stations near LCB and continues toward the front or rear ones. Each time, the heaviest of the remaining containers of port j are assigned to the closest to LCBj free cell. We choose where to place the container (front or rear part of the vessel) by following the process described in section 4.2.2.

There are, however, two exceptions to the above rule. In case when the percentage rule is in effect, we attempt to reduce the additional water ballast as much as possible; thus, we want to assign a light container the closest possible to LCBj. By contrast, when the moment is out of the permissible range and the next container is to be assigned to the relatively more full part of the vessel, we attempt to change the moment as much as possible. This can be done by assigning a heavy container to the most remote available position from LCB.

The following figure presents the different strategies we can follow. In strategies 1 and 4 we choose the heaviest of the remaining containers to be loaded, while in strategies 2 and 3 we choose the lightest container.

Concerning the vertical position of the container, we can choose between two allocation disciplines, as described below:

1) The vesel is filled station by station. The procedure is carried out independently in the front and rear part of the vessel.

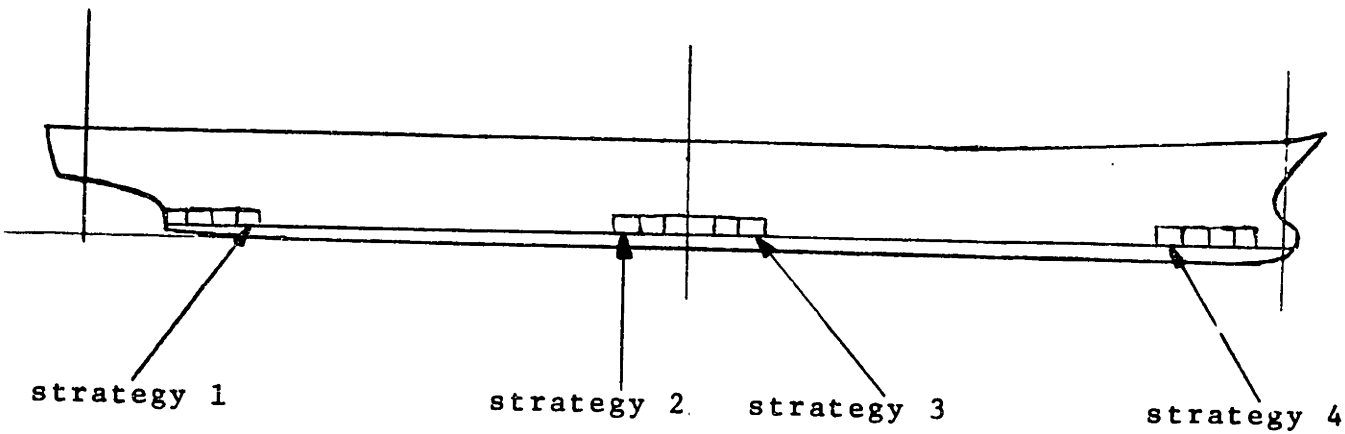


FIGURE 3.3 Cell assignment procedure: strategies
In strategies 1 & 4 the heaviest of the remaining containers is selected, while in strategies 2 & 3 the lightest one is selected.

2) The front and rear parts of the vessel are filled row by row. A container is assigned to a higher row only if all the lower rows of the part it is assigned are filled.

Although the first discipline spreads containers of the same origin along the length of the vessel, it generally results in a higher GM than the second one. This happens because containers are allocated in more columns resulting in fewer containers per column; this latter feature means that the vertical coordinate of the containers' center of weight (KG) is lower in the first than in the second allocation procedure, and, consequenetly, the (GM) is greater.

3.3 STAGE TWO: GM correction

3.3.1 General strategy

It is obvious that when heavy containers are placed above light ones, the GM is not as high as it could be if the reverse could happen. If the minimum GM requirement is not satisfied, we can attempt to make the container allocation feasible by moving containers from the upper rows to the lower ones.

This is done in two steps. First, container interchanges in each column are performed. More specifically, the containers of some column are rearranged in decreasing order of weight. The number of columns where these interchanges take place depends on how much the actual GM is lower than the

minimum required.

If the containers have been placed in decreasing order of weight in all columns and the GM-constraint is still unsatisfied, the algorithm proceeds to the second step. This step involves interchanges among containers of a station.

If all possible interchanges as described above are made and the loading remains infeasible then the algorithm terminates and gives the relevant message. Still, a solution might be found if longitudinal interchanges were also examined. However, the current version of the algorithm does not consider such interchanges.

As we have already mentioned, longitudinal interchanges increase significantly the complexity of the algorithm, because they affect all the previous loadings. Moreover, if we follow the first allocation discipline described in section 3.2.2, the GM improvement after the longitudinal interchanges is expected to be smaller than the one resulting if the second discipline is followed. In the latter case, longitudinal interchanges should be necessary because GM will be generally smaller. This happens because the containers are distributed in fewer columns resulting in greater KG.

3.3.2 Step 1: Container interchanges within columns

When container interchanges are unavoidable, we wish to minimize the incurred container overstowage cost. The problem then, is to select in which columns interchanges should be

made. We can formulate this as a 0-1 integer programming "knapsack" problem, as follows:

Minimize Z=C1*X1 + C2*X2 +.....+ CN*XN

Subject to : B1*X1 + B2*X2 +...+ BN*XN > DGM

where:

Z the additional cost incurred from overstowage,

N the number of columns,

Ci the incurred overstowage cost for column i,

Bi the increase in GM if the containers of column i are placed by decreasing order of weight,

DGM the necessary improvement in GM,

Xi decision variable which is

1 if column i is selected for interchange

and 0 if not.

The formulation considers only columns with positive Bi. Both Bi and Ci are updated every time a new container is placed in the column. In the case that the sum of Bi's is less than DGM, interchanges take place in all columns and the algorithm proceeds to step 2.

A simple heuristic to solving the above problem is to rank-order the columns with respect to the ratio Ri=Bi/Ci and to select those columns with the highest ratios until the constraint is satisfied.

Theoretically, there may be pathological cases in which this "greedy" heuristic gives poor results. However, it is

expected that this will not happen very often in our case, because all the coefficients are of the same order of magnitude. Moreover, the simplicity of the method is a considerable factor for it to be ignored.

In order to implement the above heuristic we have to calculate the Bi and Ci coefficients for each column.

Calculation of Bi.

Bi is defined as the increase in GM when the containers of column i are placed in decreasing order of weight. That is,

$$Bi = DVMOMENT / D$$

where,

DVMOMENT is the decrease of the vertical moment with respect to the keel, and,

D is the displacement of the vessel.

Since the rearrangement of the containers can always be done by consecutively interchanging adjacent containers, the coefficient Bi can be expressed as

$$Bi = \left(\sum_{j=1}^{K} DWj \right) * h / D$$

where, h is the height of an individual container, supposed to be the same for all, and,

DWj is the difference of weights of the two containers involved in the jth adjacent interchange,

K is the total number of adjacent container interchanges in column i to rearrange the containers

by decreasing order of weight.

Because every container will be interchanged with all the containers that are lighter than it, and which are also initially placed in lower rows, the calculation of the DWj is rather simple. The following example illustrates more the situation showing step by step the interchanges that are made.

Initial allocation		Adjacent-container interchanges							Final allocation	
orig.	weight								weight	orig.
5	6.0	6.0	6.0	6.0	3.0	3.0	3.0	3.0	3.0	1
4	7.0	7.0	7.0	3.0	6.0	6.0	4.0	4.0	4.0	3
3	4.0	4.0	3.0	7.0	7.0	4.0	6.0	6.0	5.0	2
2	5.0	3.0	4.0	4.0	4.0	7.0	7.0	5.0	6.0	5
1	3.0	5.0	5.0	5.0	5.0	5.0	5.0	7.0	7.0	4

Bi is calculated as follows:

$$S\ DWj = (5-3) + (4-3) + (7-3) + (6-3) + (7-4) + (6-4) + (7-5) + (6-5) = 19$$

$$Bi = 19 * h / D$$

Calculation of Ci.

If we define as the unit cost of overstowage the cost of discharging and one container and placing that container back on the vessel, then the total cost of rearrangement is equal to the number of the containers which have to be unloaded and loaded again, each time interchanges in a column must be performed.

This cost can be easily calculated if we find which is the lower-placed container that has to be moved; then the number of containers interchanged is equal to the number of

containers above it plus one, minus the number of the containers loaded at the last (current) port.

Considering the previous example, we have:

a) lowest row from which a container should be moved: 1

b) number of containers above it plus one: 5

c) number of containers loaded at the last port (port 5): 1

---

d) total overstowage cost if interchanges take place: 4

Calculation of the ratio Ri

Finally, we calculate the ratio :

$$Ri = Bi/Ci = 19 * h / D / 4 = 4.75*h/D$$

Notice that h and D are the same for all columns i, hence they can omitted from the ratio calcilations.

This ratio is calculated for all columns i which have Bi positive. in the case that both Bi and Ci are equal to zero we set the ratio Ri also equal to zero.

4.3.4 Step 2: Transvrese interchanges.

The algorithm proceeds to Step Two if the GM-constraint is still not satisfied, after Step One.In this step the algorithm examines if transverse container interchanges within a station can increase GM. It must be noted that the containers in the columns of all stations have been already placed by decreasing order of weight, because Step One has been already executed.

The weight of each container in the station (in the

following called 'basic' container) is compared with the weight of the container in the next lower row of all other columns.

If the weight of the 'basic' container is less than or equal to the weight of the container in comparison, no interchange takes place and we move to the next column. Otherwise, the two containers are interchanged. Besides, they interchange their roles: the lighter container becomes now the 'basic' one. Moreover, the algorithm examines if the 'ex-basic' container can be placed even lower in its new column. After completing this latter inter-loop interchanges the algorithm continues to the next column as before.

This procedure is applied to all containers of the station starting from the ones of the upper rows and proceeding to the lower rows. The value of the Bi-coefficient remains zero for all columns, since the containers are placed by decreasing order of weight. The overstowage cost may increase or decrease.

Step Two is executed for all stations of the vessel.

4.4 STAGE THREE: Final improvement

The last stage of the algorithm resembles Step Two of Stage Two. It is executed only if a feasible solution is found in the previous stages and only if interchanges have occurred, because otherwise the allocation within the station is already locally optimal.

First, the algorithm tries to increase GM without

increasing the handling cost. This is realized by placing the heaviest containers among those from the same origin (or to the same destination) as low as possible.

In addition, the algorithm examines if the handling cost can be reduced without decreasing GM. This involves interchanges of equally heavy containers, provided that the final cost is reduced. However, this procedure is very time-consuming and it is questionable if it is worth pursuing.

CHAPTER 4: THE COMPUTER PROGRAM

4.1 Introduction

A FORTRAN computer program has been developed according to the algorithm presented in the previous chapter.

Inputs to the program are the vessel characteristics and the loading schedule. Output is either the allocation of the containers on board, after all the constraints are satisfied, or a message that a feasible solution cannot be found.

The program is interactive. The user is asked to give the value of some parameters or to make some choices during the run of the program. It is comprised, except from its main part, of eight subroutines and three external functions.

Figure 4.1 shows the way that the subroutines are linked to each other. The symbol A <-- B means that subroutine B is called by subroutine A. In most cases the necessary data are transferred among subroutines through "COMMON" blocks.

This chapter briefly describes the operations each part of the program performs. A user's manual as well as sample input files are given in appendix A1. The program listing is presented in appendix A2.

4.2 Description of the program

4.2.1 The MAIN program

The MAIN program performs the following operations:

a) Reads inputs

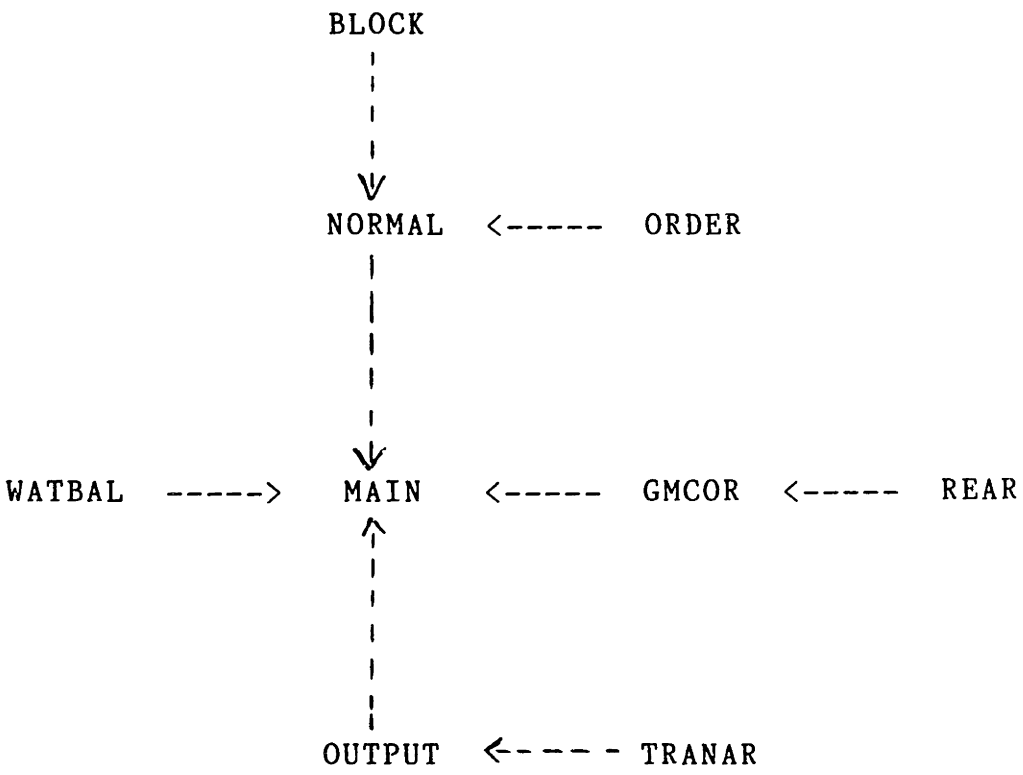


FIGURE 4.1 Program structure

There are three sorts of inputs:

1) Vessel characteristics.

The necessary vessel parameters are the weight of light ship, the initial KG (vertical center of weight), the initial value of the longitudinal moment, the position and capacity of tanks and the available container positions on board.

2) Container loading schedule.

We provide the number as well as the weight of containers to be loaded at each port. Containers of the same origin are given by decreasing order of weight.

3) Interactive input.

Except for the two above sets of inputs which are read from external files, the program asks the operator to enter the values of some parameters during the run. In this way we enter the values of the minimum required GM, the maximum allowed difference in the percentages the front and rear part of the vessel are filled and the permissible range for the longitudinal moment.

b) After these inputs are read, the program performs the following tasks:

1) Decides which containers (if any) will be placed on deck and informs at which port this happens. This decision is simply taken by filling all the below-deck positions first.

2) Gives initial values to variables, where necessary.

c) When all the above are completed the loading procedure commences. The following operations are performed:

1) The program asks the operator to select one out of the two possible allocation disciplines (see section 3.2.4)

2) The program decides in which part of the vessel each container will be placed. Then it calls subroutine NORMAL to assign the container to a specific cell.

3) After placing all containers on board at each port, the program checks the value of the longitudinal moment and corrects it if necessary by using water ballast. However, the use of the water ballast is kept as small as possible. Additionally, the program calculates the metacentric height (GM). If the GM-constraint is not satisfied, subroutine GMCOR is called.

4) At the last port the process terminates. The program asks if the user wants to save the results in an output file.

4.2.2 Subroutine NORMAL

This subroutine assigns the individual container to a specific cell. No container is assigned on deck unless all the cells below it have been filled.

Moreover the subroutine calculates the amount by which GM can be increased if the containers of the column in which the container is placed are rearranged in decreasing order of weight. It also calculates the incurred increase in the handling cost. Then, the increase per unit cost (benefit/cost ratio) is calculated and the columns are ranked by decreasing benefit/cost ratio. This procedure is carried out independently for the containers below and on the deck.

4.2.3 Subroutine BLOCK

Subroutine BLOCK is called by NORMAL in the case that the first allocation procedure is selected. According to this procedure containers from the same origin are blocked.

4.2.4 Subroutine ORDER

This subroutine ranks the columns with respect to their benefit/cost ratio. In the beginning zero,a benefit/cost ratio is given to all columns. Subroutine ORDER is called by NORMAL.

4.2.5 Subroutine WATBAL

When the use of water ballast is necessary, the program calls subroutine WATBAL. This subroutine attempts to bring the longitudinal moment within the permissible range. The subroutine tries to do this first, by reducing the water ballast; if the correction of the moment cannot be achieved this way, water ballast is added.

The tanks around amidships are filled first, in an effort to reduce bending moment effects.

4.2.6 Subroutine GMCOR

This subroutine is called by the MAIN program when the GM-constraint is not satisfied. GMCOR makes container interchanges in the columns where it is possible. The subroutine moves the heavier containers to lower rows, starting by the columns with higher benefit/cost ratios. Interchanges terminate when the GM-constraint is satisfied.

If the GM-constraint is still violated after all possible interchanges are made, subroutine TRANAR is called.

4.2.7 Subroutine TRANAR

TRANAR makes transverse container interchanges among the containers of a station. Containers below and on the deck are treated independently.

4.2.8 Subroutine REAR

While subroutine GMCOR decides in which columns interchanges will take place, subroutine REAR has the task to implement these interchanges by placing the heaviest containers in lower rows.

4.2.9 Subroutine OUTPUT

This subroutine prepares the output file. It calculates the number of containers from the same origin at each station, the overall overstowage cost and the final values of GM, longitudinal moment, trim, draft and displacement.

4.2.10 External functions

External functions are used to describe some of the hydrostatic curves of the vessel as functions of displacement (D). The necessary hydrostatic curves are:

-Center of buoyancy (XLCB(D))

-Distance of metacenter from the keel (XKM(D))

-Moment to change trim one inch (XMTC1(D))

5. RESULTS AND DISCUSSION

We have applied the previously presented algorithm and computer program in the case of a vessel visiting a number of ports. The vessel starts empty at port 0 and is being loaded at aubsequent ports 1,2,3,4,5 and 6 during the trip.

The vessel we use is a typical containership, which can carry 1540 10-ton or 1350 11.8-ton 20-foot containers. Some other general characteristics of the vessel are shown in table 5.1. The hydrostatic curves and available container positions are presented in appendix A1.

The required CPU time is about 15 seconds if 700 containers are loaded and about 27 seconds if 1300 containers are placed.

In general, the required CPU time is proportional to the number of the containers charged. In first approximation the relationship between them is linear.

Two loading schedules have been examined. The first one, shown in table 5.2, partially fills the vessel, while the second one fills almost all the available cells. Generally, the weight of the containers loaded at the same port can vary. In fact, the second loading schedule considers different weights among the containers of a port. Table 5.3 shows the number of containers to be loaded at each port under the second loading schedule.

The available cells on the vessel under consideration are 1540. We assume that, at least for the container weight the

TABLE OF CHARACTERISTICS	
SHIP TYPE: CONTAINER SHIP	
PRINCIPAL CHARACTERISTICS	
Length overall, ft. - ins.	725-0
Length on designed water line, ft. - ins.	695-0
Length between perpendiculors, ft. - ins.	685-0
Beam, maximum molded, ft. - ins.	103-0
Depth to strength deck (upper deck) at side, minimum, molded, ft. - ins.	60-0
Design draft, molded, ft. - ins.	29-6
Scantling draft, molded, ft. - ins.	34-0
Displacement at design draft, molded, tons.	32,200
Displacement at scantling draft, molded, tons.	38,500
Type of main propulsion machinery.	Steam Turbine
Maximum SHP (ABS power rating), main propulsion machinery.	35,000
Estimated fuel consumption at sea, ABS power, bbls of 42 gallons/day.	1210
Estimated fuel consumption in port, bbls of 42 gallons/day.	114
Sea speed, sustained, full load (design draft) at 80 percent of maximum SHP on trial in good weather conditions, knots.	23.3
Sea speed, sustained, ballast condition (22'-0" mean draft) at 80 percent of maximum SHP on trial in good weather conditions, knots.	24.4
Endurance in nautical miles at "sustained sea speed" using 3,300 tons of fuel oil.	10,000
Gross Tonnage, U.S.	25,900
Net Tonnage, U.S.	17,500

TABLE 5.1 Principal characteristics of the vessel used to test the algorithm

vessel is designed for, the ship can be fully loaded in a feasible way.

TABLE 5.2
First loading schedule examined

Port	number of containers to be loaded	Individual container weight
1	100	10 tons
2	100	11 tons
3	100	8 tons
4	200	12 tons
5	200	10 tons

TABLE 5.3
Second loading schedule examined

Port	number of containers to be loaded	Individual container weight
1	100	8 - 20 tons
2	100	8 - 20 tons
3	100	8 - 20 tons
4	200	8 - 20 tons
5	200	8 - 20 tons
6	600	8 - 20 tons

In order to force the algorithm to proceed to container interchanges we have assumed that the minimum required GM is artificially high.

Outputs of the computer program are presented in the next pages. The aggregated allocation of the containers to stations after the last port visited is shown in tables 5.4 and 5.5 for the data of table 5.2. In addition, the GM, the water ballast in tanks as well as the trim of the vessel after the departure from every port are presented. The results of table 5.4 have been obtained by using the first allocation discipline

Number of ports visited until now : 5

Blocking container discipline has been selected
Vessel is filled station by station

PREVIOUS PORTS

Port	cnts loaded	conts' weight	Displacement	Trim	GM
1	50	1000.0 t	17680.6 t	-0.24 f	18.02 f
2	50	1100.0 t	18608.6 t	-0.07 f	15.41 f
3	50	800.0 t	19191.7 t	-0.04 f	13.89 f
4	100	2400.0 t	20272.2 t	-0.01 f	9.99 f

LAST PORT

Displacement = 22372.46 tons
GM = 10.58 feet
KG = 34.42 feet
Minimum required GM = 50.00 feet
TRIM = -0.084 feet
ANGLE = 0.000 rads

No of tank	Water contained	Capacity
1	0.00	493.000
2	0.00	1544.000
3	0.00	298.000
4	0.00	3080.000
5	2080.00	2080.000
6	111.44	835.000
7	0.00	951.000

TOTAL OVERSTOWAGE COST = 58.0 units

STATION:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
port 1:	0	0	0	0	0	0	0	0	0	0	70	30	0	0	0	0	0	0	0	0	0	0
port 2:	0	0	0	0	0	0	0	0	0	0	0	40	60	0	0	0	0	0	0	0	0	0
port 3:	0	0	0	0	0	0	0	0	0	0	0	0	6	66	28	0	0	0	0	0	0	0
port 4:	0	0	0	0	0	0	0	0	0	2	0	0	0	0	26	54	34	34	14	12	20	4
port 5:	0	0	0	0	0	0	0	50	70	68	0	0	0	0	0	0	0	0	0	0	0	12
ov-cost:	0	0	0	0	0	0	0	0	0	0	0	30	0	0	28	0	0	0	0	0	0	0

TABLE 5.4 Allocation of containers and overstowage cost after port5.
K=1.0 , GMmin=50.0 ft, "blocking" selected

Number of ports visited until now : 5

Vessel is filled row by row

PREVIOUS PORTS

Port	cnts loaded	conts' weight	Displacement	Trim	GM
1	50	1000.0 t	17510.5 t	-0.22 f	19.65 f
2	50	1100.0 t	18391.5 t	-0.06 f	17.59 f
3	50	800.0 t	18930.3 t	-0.04 f	15.76 f
4	100	2400.0 t	20306.8 t	-0.14 f	10.40 f

LAST PORT

Displacement = 22635.04 tons
GM = 12.10 feet
KG = 32.90 feet
Minimum required GM = 50.00 feet
TRIM = -0.293 feet
ANGLE = 0.000 rdas

No of tank	Water contained	Capacity
1	0.00	493.000
2	0.00	1544.000
3	0.00	298.000
4	0.00	3080.000
5	2080.00	2080.000
6	374.07	835.000
7	0.00	951.000

TOTAL OVERSTOWAGE COST = 408.0 units

STATION:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
port 1:	0	0	0	0	0	0	0	0	0	0	16	20	16	16	10	10	6	6	0	0	0	0
port 2:	0	0	0	0	0	0	0	0	0	0	24	20	20	12	8	8	4	4	0	0	0	0
port 3:	0	0	0	0	0	0	0	0	0	0	10	10	10	18	16	16	4	4	2	2	4	4
port 4:	0	0	0	0	0	0	0	0	0	0	20	20	20	20	20	20	20	20	12	10	10	8
port 5:	0	0	0	0	26	26	30	30	38	40	0	0	0	0	0	0	0	0	0	0	6	4
ov-cost:	0	0	0	0	0	0	0	0	0	0	66	70	62	62	44	44	20	20	2	2	8	8

TABLE 5.5 Allocation of containers and overstowage cost after port 5.
K=1.0, GMmin=50.0 ft, "blocking" not selected

(blocking of containers from the same origin). The second allocation discipline (vessel is filled row by row) has been used to allocate the containers as presented in Table 5.5.

The results confirm our expectations. In the case in which containers from the same origin are blocked, the GM is smaller than in the case the vessel is filled row by row. Moreover, the amount of water ballast required to make the trim zero is less in the former case than in the latter, because the same number of containers are assigned closer to LCB.

Also the overstowage cost is smaller when the first allocation discipline is used; this is true because since the heaviest containers at each port are loaded first, and there are more containers from the same origin per station (container blocking is selected), the probability the containers of a column to be in decreasing order of weight is higher and the possible interchanges fewer.

We did not calculate, at least in this version of the algorithm, the change (increase or decrease) in the container handling cost after transverse interchnages are made, because of time constraints. In general, we expect this cost to be small compared to the other overstowage cost, especially in the case we select to "block" containers of the same origin.

Tables 5.6 and 5.7 present in more detail the allocation of containers in the above two cases. The real numbers represent container weights and the integers below them

STATION # 9

0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0
0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0
10.00 5	10.00 5	10.00 5	10.00 5	10.00 5	0.00 -1
10.00 5	10.00 5	10.00 5	10.00 5	10.00 5	0.00 -1
10.00 5	10.00 5	10.00 5	10.00 5	10.00 5	0.00 -1
10.00 5	10.00 5	10.00 5	10.00 5	10.00 5	0.00 -1
10.00 5	10.00 5	10.00 5	10.00 5	10.00 5	0.00 -1
10.00 5	10.00 5	10.00 5	10.00 5	10.00 5	0.00 -1
10.00 5	10.00 5	10.00 5	10.00 5	10.00 5	0.00 -1

STATION # 10

0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0
0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0
10.00 5	10.00 5	10.00 5	10.00 5	10.00 5	0.00 -1
10.00 5	10.00 5	10.00 5	10.00 5	10.00 5	0.00 -1
10.00 5	10.00 5	10.00 5	10.00 5	10.00 5	0.00 -1
10.00 5	10.00 5	10.00 5	10.00 5	10.00 5	0.00 -1
10.00 5	10.00 5	10.00 5	10.00 5	10.00 5	0.00 -1
10.00 5	10.00 5	10.00 5	10.00 5	10.00 5	0.00 -1
12.00 4	10.00 5	10.00 5	10.00 5	10.00 5	0.00 -1

STATION # 11

0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0
0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0
10.00 1	10.00 1	10.00 1	10.00 1	10.00 1	0.00 -1
10.00 1	10.00 1	10.00 1	10.00 1	10.00 1	0.00 -1
10.00 1	10.00 1	10.00 1	10.00 1	10.00 1	0.00 -1
10.00 1	10.00 1	10.00 1	10.00 1	10.00 1	0.00 -1
10.00 1	10.00 1	10.00 1	10.00 1	10.00 1	0.00 -1
10.00 1	10.00 1	10.00 1	10.00 1	10.00 1	0.00 -1
10.00 1	10.00 1	10.00 1	10.00 1	10.00 1	0.00 -1

STATION # 12

0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0
0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0
10.00 1	10.00 1	10.00 1	10.00 1	10.00 1	0.00 -1
10.00 1	10.00 1	10.00 1	10.00 1	10.00 1	0.00 -1
10.00 1	10.00 1	10.00 1	10.00 1	10.00 1	0.00 -1
11.00 2	11.00 2	11.00 2	11.00 2	11.00 2	0.00 -1
11.00 2	11.00 2	11.00 2	11.00 2	11.00 2	0.00 -1
11.00 2	11.00 2	11.00 2	11.00 2	11.00 2	0.00 -1
11.00 2	11.00 2	11.00 2	11.00 2	11.00 2	0.00 -1

STATION # 13

0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0
0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0
11.00 2	11.00 2	8.00 3	8.00 3	8.00 3	0.00 -1
11.00 2	11.00 2	11.00 2	11.00 2	11.00 2	0.00 -1
11.00 2	11.00 2	11.00 2	11.00 2	11.00 2	0.00 -1
11.00 2	11.00 2	11.00 2	11.00 2	11.00 2	0.00 -1
11.00 2	11.00 2	11.00 2	11.00 2	11.00 2	0.00 -1
11.00 2	11.00 2	11.00 2	11.00 2	0.00 -1	0.00 -1
11.00 2	11.00 2	11.00 2	11.00 2	0.00 -1	0.00 -1

STATION # 14

0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0
0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0
8.00 3	8.00 3	8.00 3	8.00 3	8.00 3	0.00 -1
8.00 3	8.00 3	8.00 3	8.00 3	8.00 3	0.00 -1
8.00 3	8.00 3	8.00 3	8.00 3	8.00 3	0.00 -1
8.00 3	8.00 3	8.00 3	8.00 3	8.00 3	0.00 -1
8.00 3	8.00 3	8.00 3	8.00 3	8.00 3	0.00 -1
8.00 3	8.00 3	8.00 3	8.00 3	0.00 -1	0.00 -1
8.00 3	8.00 3	8.00 3	8.00 3	0.00 -1	0.00 -1

TABLE 5.6 Detailed view of container allocation at stations around amidships.
K=1.0, GMmin= 50.0 ft, "blocking" selected

STATION # 9

0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0
0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0
0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 -1
0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 -1
0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 -1
10.00 5	10.00 5	10.00 5	10.00 5	0.00 0	0.00 -1
10.00 5	10.00 5	10.00 5	10.00 5	10.00 5	0.00 -1
10.00 5	10.00 5	10.00 5	10.00 5	10.00 5	0.00 -1
10.00 5	10.00 5	10.00 5	10.00 5	10.00 5	0.00 -1

STATION # 10

0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0
0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0
0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 -1
0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 -1
0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 -1
10.00 5	10.00 5	10.00 5	10.00 5	10.00 5	0.00 -1
10.00 5	10.00 5	10.00 5	10.00 5	10.00 5	0.00 -1
10.00 5	10.00 5	10.00 5	10.00 5	10.00 5	0.00 -1
10.00 5	10.00 5	10.00 5	10.00 5	10.00 5	0.00 -1

STATION # 11

0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0
0.00 0	0.00 0	[illegible]	0.00 0	0.00 0	0.00 0
8.00 3	8.00 3	[illegible]	8.00 3	8.00 3	0.00 -1
10.00 1	10.00 1	10.00 1	10.00 1	10.00 1	0.00 -1
10.00 1	10.00 1	10.00 1	11.00 2	11.00 2	0.00 -1
11.00 2	11.00 2	11.00 2	11.00 2	11.00 2	0.00 -1
11.00 2	11.00 2	11.00 2	11.00 2	11.00 2	0.00 -1
12.00 4	12.00 4	12.00 4	12.00 4	12.00 4	0.00 -1
12.00 4	12.00 4	12.00 4	12.00 4	12.00 4	0.00 -1

STATION # 12

0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0
0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0
8.00 3	8.00 3	8.00 3	8.00 3	8.00 3	0.00 -1
10.00 1	10.00 1	10.00 1	10.00 1	10.00 1	0.00 -1
10.00 1	10.00 1	10.00 1	10.00 1	10.00 1	0.00 -1
11.00 2	11.00 2	11.00 2	11.00 2	11.00 2	0.00 -1
11.00 2	11.00 2	11.00 2	11.00 2	11.00 2	0.00 -1
12.00 4	12.00 4	12.00 4	12.00 4	12.00 4	0.00 -1
12.00 4	12.00 4	12.00 4	12.00 4	12.00 4	0.00 -1

STATION # 13

0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0
0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0
8.00 3	8.00 3	8.00 3	8.00 3	8.00 3	0.00 -1
10.00 1	10.00 1	10.00 1	10.00 1	10.00 1	0.00 -1
11.00 2	10.00 1	10.00 1	10.00 1	11.00 2	0.00 -1
11.00 2	11.00 2	11.00 2	11.00 2	11.00 2	0.00 -1
12.00 4	11.00 2	11.00 2	11.00 2	12.00 4	0.00 -1
12.00 4	12.00 4	12.00 4	12.00 4	0.00 -1	0.00 -1
12.00 4	12.00 4	12.00 4	12.00 4	0.00 -1	0.00 -1

STATION # 14

0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0
0.00 0	0.00 0	0.00 0	0.00 0	0.00 0	0.00 0
8.00 3	8.00 3	8.00 3	8.00 3	8.00 3	0.00 -1
10.00 1	8.00 3	8.00 3	8.00 3	8.00 3	0.00 -1
10.00 1	10.00 1	10.00 1	10.00 1	10.00 1	0.00 -1
11.00 2	11.00 2	10.00 1	10.00 1	11.00 2	0.00 -1
12.00 4	11.00 2	11.00 2	11.00 2	12.00 4	0.00 -1
12.00 4	12.00 4	12.00 4	12.00 4	0.00 -1	0.00 -1
12.00 4	12.00 4	12.00 4	12.00 4	0.00 -1	0.00 -1

TABLE 5.7 Detailed view of container allocation at stations around amidships.
K=1.0, GMmin=50, ft, "blocking" not selected

represent the origin of the container. If the latter number is zero or negative the corresponding position (cell) is either empty or not physically available (respectively).

Tables 5.8 and 5.9 present results for the same loading schedule but this time the minimum required GM is set equal to 9.85 feet. In this case we expect that the final GM will be less than it was previously (Table 5.4 and 5.5) because not all the container interchanges are necessary. Actually, we expect GM to be slightly higher than 9.85 feet. The reason that this does not happen at the end (as Tables 5.8 and 5.9 show) is that:

i) the GM constraint is first violated after the containers of the fourth port are placed. At that point interchanges take place and the GM became slightly higher (less than .01 feet) than 9.85 feet (see table 5.8),

ii) the containers of the fifth port are loaded in the lower rows, reducing the vertical center of weight of the cargo and increasing thus the GM enough above the minimum required.

In the above case we cannot avoid the overstowage cost, because the loading must be feasible at every part of the trip. If the incurred overstowage cost is high, we can set into effect the "percentage rule" (see section 3.2.3). By this rule the front and the rear part of the vessel are filled proportionally, and consequently the vertical coordinate of the center of weight increases when new containers are placed.

Number of ports visited until now : 5

Blocking container discipline has been selected
Vessel is filled station by station

PREVIOUS PORTS

Port	cnts loaded	conts' weight	Displacement	Trim	GM
1	50	1000.0 t	17680.6 t	-0.24 f	18.02 f
2	50	1100.0 t	18742.8 t	0.17 f	14.99 f
3	50	800.0 t	19505.3 t	0.04 f	12.86 f
4	100	2400.0 t	21840.1 t	-0.01 f	11.71 f

LAST PORT

Displacement =	23784.45 tons
GM =	11.84 feet
KG =	33.16 feet
Minimum required GM =	50.00 feet
TRIM =	-0.018 feet
ANGLE =	0.000 rads

No of tank	Water contained	Capacity
1	0.00	493.000
2	0.00	1544.000
3	0.00	298.000
4	0.00	3080.000
5	2080.00	2080.000
6	835.00	835.000
7	688.46	951.000

TOTAL OVERSTOWAGE COST = 68.0 units

STATION:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
port 1:	0	0	0	0	0	0	0	0	0	0	70	30	0	0	0	0	0	0	0	0	0	0
port 2:	0	0	0	0	0	0	0	0	0	54	0	40	6	0	0	0	0	0	0	0	0	0
port 3:	0	0	0	0	0	0	0	0	38	16	0	0	46	0	0	0	0	0	0	0	0	0
port 4:	0	0	0	0	0	0	8	70	32	0	0	0	14	66	10	0	0	0	0	0	0	0
port 5:	0	0	0	0	0	44	62	0	0	0	0	0	0	0	44	50	0	0	0	0	0	0
ov-cost:	0	0	0	0	0	0	0	0	38	0	0	30	0	0	0	0	0	0	0	0	0	0

TABLE 5.8 Allocation of containers and overstowage cost after port 5.

K=0.2, GMmin= 50.0 ft, "blocking" selected.

Number of ports visited until now : 5

Vessel is filled row by row

PREVIOUS PORTS

Port	cnts loaded	conts' weight	Displacement	Trim	GM
1	50	1000.0 t	17510.5 t	-0.22 f	19.65 f
2	50	1100.0 t	18650.1 t	-0.02 f	17.06 f
3	50	800.0 t	19439.9 t	-0.01 f	15.03 f
4	100	2400.0 t	21759.9 t	-0.02 f	13.92 f

LAST PORT

Displacement =	23781.20 tons
GM =	13.31 feet
KG =	31.69 feet
Minimum required GM =	50.00 feet
TRIM =	-0.047 feet
ANGLE =	0.000 rads

No of tank	Water contained	Capacity
1	0.00	493.000
2	0.00	1544.000
3	0.00	298.000
4	0.00	3080.000
5	2080.00	2080.000
6	835.00	835.000
7	685.21	951.000

TOTAL OVERSTOWAGE COST = 426.0 units

STATION:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
port 1:	0	0	0	0	0	0	0	0	0	0	16	20	16	16	10	10	6	6	0	0	0	0
port 2:	0	0	0	0	6	8	10	10	10	10	14	10	10	10	2	0	0	0	0	0	0	0
port 3:	0	0	0	0	6	8	10	10	10	10	10	10	4	0	6	8	4	4	0	0	0	0
port 4:	0	0	0	0	14	16	20	20	20	20	10	10	16	18	8	8	4	4	2	2	4	4
port 5:	4	4	8	8	20	14	10	10	10	10	10	10	10	12	18	18	6	6	2	2	4	4
ov-cost:	0	0	0	0	18	24	30	30	30	30	66	70	4	42	28	26	14	14	0	0	0	0

TABLE 5.9 Allocation of containers and overstiwage cost after port 5.

K=0.2, GMmin= 50.0 ft, "blocking" not selected

Of course, the horizontal movements of the crane are now greater. The solution chosen finally should be the one resulting in less total cost.

Some examples when the "percentage rule" is in effect are shown in tables 5.10 and 5.11. The parameter K (see section 3.2.3) has been set equal to 0.2, that is, the fractions by which the two parts of the vessel are filled differ by at most 0.2. However, as we can see this results in more water ballast and of course in more horizontal crane movements. Furthermore, the GM is now greater, because the lower rows are filled first.

Finally, in Tables 5.12 and 5.13 we present the program output for the second loading schedule. The vessel now visits six ports and is loaded with 1300 containers. Some of the containers loaded at port six have been loaded on deck, because all the cells below the deck have been filled. These containers are referred to as containers going to port seven. In fact ports six and seven coincide. The results generally agree with what we have previously discussed.

Number of ports visited until now : 5

Blocking container discipline has been selected
Vessel is filled station by station

PREVIOUS PORTS

Port	cnts loaded	conts' weight	Displacement	Trim	GM
1	50	1000.0 t	17680.6 t	-0.24 f	18.02 f
2	50	1100.0 t	18608.6 t	-0.07 f	15.37 f
3	50	800.0 t	19191.7 t	-0.04 f	13.85 f
4	100	2400.0 t	20272.2 t	-0.01 f	9.85 f

LAST PORT

Displacement = 22372.46 tons
GM = 10.45 feet
KG = 34.55 feet
Minimum required GM = 9.85 feet
TRIM = -0.084 feet
ANGLE = 0.000 rads

No of tank	Water contained	Capacity
1	0.00	493.000
2	0.00	1544.000
3	0.00	298.000
4	0.00	3080.000
5	2080.00	2080.000
6	111.44	835.000
7	0.00	951.000

TOTAL OVERSTOWAGE COST = 8.0 units

STATION:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
port 1:	0	0	0	0	0	0	0	0	0	0	70	30	0	0	0	0	0	0	0	0	0	0
port 2:	0	0	0	0	0	0	0	0	0	0	0	40	60	0	0	0	0	0	0	0	0	0
port 3:	0	0	0	0	0	0	0	0	0	0	0	0	6	66	28	0	0	0	0	0	0	0
port 4:	0	0	0	0	0	0	0	0	0	2	0	0	0	0	26	54	34	34	14	12	20	4
port 5:	0	0	0	0	0	0	0	50	70	68	0	0	0	0	0	0	0	0	0	0	0	12
ov-cost:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0

TABLE 5.10 Allocation of containers and overstowage cost after port 5.
K=1.0, GMmin=9.85 ft, "blocking" selected.

Number of ports visited until now : 5

Vessel is filled row by row

PREVIOUS PORTS

Port	cnts loaded	conts' weight	Displacement	Trim	GM
1	50	1000.0 t	17510.5 t	-0.22 f	19.65 f
2	50	1100.0 t	18391.5 t	-0.06 f	17.52 f
3	50	800.0 t	18930.3 t	-0.04 f	15.69 f
4	100	2400.0 t	20306.8 t	-0.14 f	9.85 f

LAST PORT

Displacement = 22635.04 tons
GM = 11.60 feet
KG = 33.40 feet
Minimum required GM = 9.85 feet
TRIM = -0.293 feet
ANGLE = 0.000 rads

No of tank	Water contained	Capacity
1	0.00	493.000
2	0.00	1544.000
3	0.00	298.000
4	0.00	3080.000
5	2080.00	2080.000
6	374.07	835.000
7	0.00	951.000

TOTAL OVERSTOWAGE COST = 24.0 units

STATION:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
port 1:	0	0	0	0	0	0	0	0	0	0	16	20	16	16	10	10	6	6	0	0	0	0
port 2:	0	0	0	0	0	0	0	0	0	0	24	20	20	12	8	8	4	4	0	0	0	0
port 3:	0	0	0	0	0	0	0	0	0	0	10	10	10	18	16	16	4	4	2	2	4	4
port 4:	0	0	0	0	0	0	0	0	0	0	20	20	20	20	20	20	20	20	12	10	10	8
port 5:	0	0	0	0	26	26	30	30	38	40	0	0	0	0	0	0	0	0	0	0	6	4
ov-cost:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	6	2	2	4	4

TABLE 5.11 Allocation of containers and overstowage cost after port

K=1.0, GMmin=9.85 ft, "blocking" not selected.

Number of ports visited until now : 7

Blocking container discipline has been selected
Vessel is filled station by station

PREVIOUS PORTS

Port	cnts loaded	conts' weight	Displacement	Trim	GM
1	50	1068.0 t	17745.4 t	-0.24 f	17.90 f
2	50	1208.0 t	18769.4 t	-0.08 f	15.10 f
3	50	800.0 t	19352.6 t	-0.04 f	13.56 f
4	100	2400.0 t	20454.7 t	-0.10 f	10.20 f
5	100	2000.0 t	22536.8 t	-0.12 f	10.72 f
6	300	8087.0 t	32310.0 t	-16.82 f	15.65 f

LAST PORT

Displacement = 29774.50 tons
GM = 7.40 feet
KG = 37.60 feet
Minimum required GM = 50.00 feet
TRIM = -0.014 feet
ANGLE = 0.000 rads

No of tank	Water contained	Capacity
1	0.00	493.000
2	0.00	1544.000
3	0.00	298.000
4	0.00	3080.000
5	2080.00	2080.000
6	835.00	835.000
7	124.48	951.000

TOTAL OVERSTOWAGE COST = 80.0 units

STATION:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
port 1:	0	0	0	0	0	0	0	0	0	0	70	30	0	0	0	0	0	0	0	0	0	0
port 2:	0	0	0	0	0	0	0	0	0	0	0	40	60	0	0	0	0	0	0	0	0	0
port 3:	0	0	0	0	0	0	0	0	0	0	0	0	6	66	28	0	0	0	0	0	0	0
port 4:	0	0	0	0	0	0	0	0	0	4	0	0	0	0	26	54	34	34	14	12	20	2
port 5:	0	0	0	0	0	0	0	50	70	66	0	0	0	0	0	0	0	0	0	0	0	14
port 6:	20	20	28	28	66	66	70	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0
port 7:	0	0	0	0	0	0	0	2	24	24	24	24	24	24	24	24	20	20	16	16	8	8
ov-cost:	0	0	0	0	0	0	0	50	0	0	0	30	0	0	0	0	0	0	0	0	0	0

TABLE 5.12 Allocation of containers and overstowage cost after port 7.

K=1.0, GMmin=50.0 ft, "blocking" selected

Number of ports visited until now : 7

Vessel is filled row by row

PREVIOUS PORTS

Port	cnts loaded	conts' weight		Displacement		Trim		GM	
1	50	1068.0	t	17562.3	t	-0.22	f	19.60	f
2	50	1208.0	t	18541.6	t	-0.07	f	17.32	f
3	50	800.0	t	19080.4	t	-0.04	f	15.45	f
4	100	2400.0	t	20456.8	t	-0.05	f	10.48	f
5	100	2000.0	t	22784.6	t	-0.27	f	12.17	f
6	300	8087.0	t	32310.0	t	-16.05	f	16.0f	f

LAST PORT

Displacement = 29782.73 tons
GM = 7.84 feet
KG = 37.16 feet
Minimum required GM = 50.00 feet
TRIM= -0.013 feet
ANGLE= 0.000 rads

No of tank	Water contained	Capacity
1	0.00	493.000
2	0.00	1544.000
3	0.00	298.000
4	0.00	3080.000
5	2080.00	2080.000
6	835.00	835.000
7	132.76	951.000

TOTAL OVERSTOWAGE COST = 524.0 units

STATION:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
port 1:	0	0	0	0	0	0	0	0	0	0	16	20	16	16	10	10	6	6	0	0	0	0
port 2:	0	0	0	0	0	0	0	0	0	0	24	20	20	12	8	8	4	4	0	0	0	0
port 3:	0	0	0	0	0	0	0	0	0	0	10	10	10	18	16	16	4	4	2	2	4	4
port 4:	0	0	0	0	0	0	0	0	0	0	20	20	20	20	20	20	20	20	12	10	10	8
port 5:	0	0	0	0	26	26	30	30	38	40	0	0	0	0	0	0	0	0	0	0	6	4
port 6:	20	20	28	28	40	40	40	40	32	30	0	0	0	0	0	0	0	0	0	0	0	0
port 7:	0	0	0	0	0	2	12	12	12	12	24	24	24	24	24	24	20	20	16	16	8	8
ov-cost:	0	0	0	0	26	26	30	30	38	40	48	60	54	48	36	36	16	16	2	2	8	8

TABLE 5.13 Allocation of containers and overstowage cost after port 7.
K=1.0, GMmin=50.0 ft, "blocking" not selected.

CHAPTER 6: CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER RESEARCH

6.1 Conclusions

The results we have presented and discussed in the previous chapter indicate that the algorithm works quite well.

Although a number of assumptions have been made, none of them restricts the basic idea of the algorithm, that is:

" Containers are assigned to cells in an optimal way with respect to the overstowage cost, ignoring in the beginning some of the constraints; these constraints are checked later and in the case they are not satisfied the containers are rearranged so as to satisfy them."

We believe that the basic advantage of the suggested algorithm is the small computing time it requires compared with the methods discussed in section 1.2. The only integer programming problem that appears is solved by a very simple heuristic. No other integer or dynamic formulations are used. In contrast, the other existing algorithms use such methods, which by their nature are very time consuming.

As we have already mentioned, the introduction of new constraints will increase the complexity of the computer program, without affecting significantly the nature of the algorithm. This will happen because more checks will be necessary before a container is assigned to a cell. Moreover

the rearrangement procedure will be more complicated and the required CPU time will increase.

6.2 Suggested further research

It is clear that this subject has not been exhausted in this thesis. In fact, only the first step has been carried out. There are many areas in which this work can be continued toward a more realistic and efficient method.

The following list includes directions where future research can be fruitful:

1) Refinement of the computer program

In order to obtain results within the frame of this work, a number of simplified assumptions have been made with respect to the approximate calculation of some vessel parameters. (For example, the calculation of the vertical coordinate of the center of weight of water in the tanks). Such calculations can be carried out more accurately in a refined version of the program.

Moreover, some parameters in the program take on the specific value they have in the specific vessel we consider; these parameters should be specified by a general procedure in order to make easy the application of the algorithm for any other vessel. In the program list given in appendix A2 we note the variables or the parts of the program where such improvements should be made.

2) Development of a method to select in which stations

transverse interchanges should be applied when necessary.

In the current version of the algorithm, whenever a need for transverse interchanges appears, these are performed in all stations. Although not expected to increase very much, the overstowage cost may change significantly. The calculation of the overstowage cost is more complicated in this case then when container interchanges within columns are performed.

Once the overstowage cost is calculated, the stations can be ranked with respect to their "benefit/cost" ratio. The order with which stations are selected to apply transverse interchanges can be similar to the one described in section 3.3.2.

3) Development of the computer program for the third stage of the algorithm.

The third stage of the algorithm, that is, the improvement of GM through transverse interchanges without increasing the handling cost and/or reduction of the handling cost without decreasing the GM, has not been included in the developed computer program because of the difficulties in the calculation of the change in the overstowage cost. An improved version of the computer program should include this stage.

4) Relaxation of the loading transverse-symmetry constraint.

Furthermore, we can relax the transverse-symmetry constraint of the loading. In such a case the transverse position of the container will be determined by the

requirement the transverse moment to be equal to zero. We can use water ballast to satisfy this zero moment requirement. We must notice that the use of water ballast will also affect the longitudinal moment. In addition, transverse container inetrchanges should be now performed by taking into account the incurred change in the transverse moment.

5) Consideration of lashing, racking and deck strength constraints.

These constraints only affect the maximum number of containers per stack (column) on deck. They can be easily taken into consideration by restricting the available positions on deck.

REFERENCES

1. "A loading model for a containership", Davit K. Scott, Watson Navigation Company, and, Der-San Chen, University of Alabama. November 15, 1978. Los Angeles, California.

2. "Container stowage: a computer aided pre-planning system", Jonathan J. Shields, SNAME - Northern California Section, San Francisco, California, 14 April 1983.

3. "Principles of Naval Architecture", SNAME Publication, 1977.

APPENDIX A1

APPENDIX A1: USER'S MANUAL

In this appendix we provide the necessary information to run the program. In addition sample data files as well as a sample session are presented.

A1.1 Introduction

The program is interactive. The operator has to provide to the program the following data:

i) Data for the vessel in a file called SHIP.DAT,

ii) The loading schedule in a file called LOAD.DAT,

iii) The value of some parameters during the run of the program,

iv) External functions for the hydrostatic curves of the ship.

A1.2 Inputs to the program

A1.2.2 File SHIP.DAT

This file contains the values of the initial GM, KG, longitudinal moment, displacement e.t.c., for the empty vessel, that is with everything loaded except from the containers.

Also, file SHIP.DAT contains the three dimensional matrix which describes which cells are physically available. This matrix has dimensions:

(number of stations) x (maximum number of rows) x

x (maximum number of columns)

The matrix is given station by station. An example is given in figure A1.1a. The doted lines show the dimensions of the matrix. In the above station only fourteen cells exist physically. We indicate them by 0 while the others are indicated by -1.

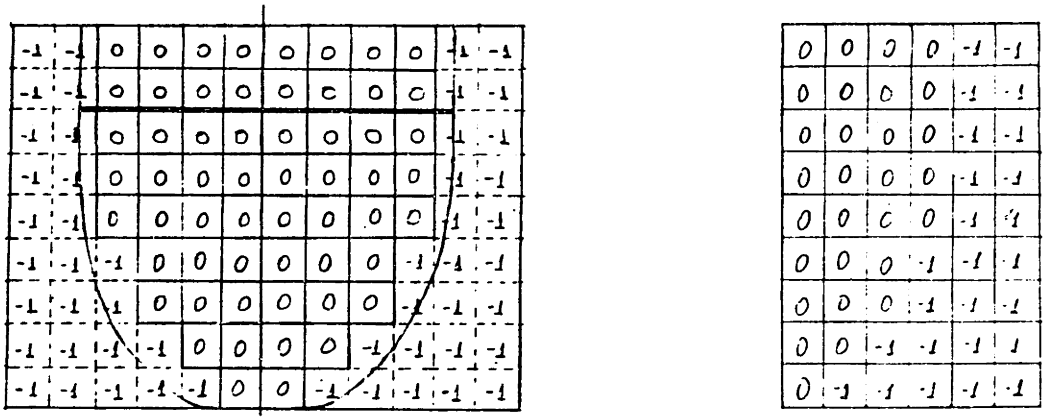
The information given to program is shown in figure A1.1b. Since we have assumed that the loading is symmetric with respect to the center line, only half the station is necessary. No containers are allowed on the center line (figure A1.2). Such stations make uncertain the final number of containers that will be loaded at a port, because containers placed on the center line are single counted while the others are double counted. The program can handle such a case, if this defficiency is ignored. The variable NF specifies if there is(are) station(s) with cells on the center line, as follows:

1 if there is(are) station(s) with cell(s) on center line
NF=

0 otherwise

In case there are both kinds of stations, we set NF=1 and give the first column of the stations with no container on the center line as infeasible. In appendix A2 a sample file SHIP.DAT is displayed.

A1.2.2 File LOAD.DAT



(a) (b)

FIGURE A1.1 (a) Description of the available cells of a station.
(b) Input to the program.

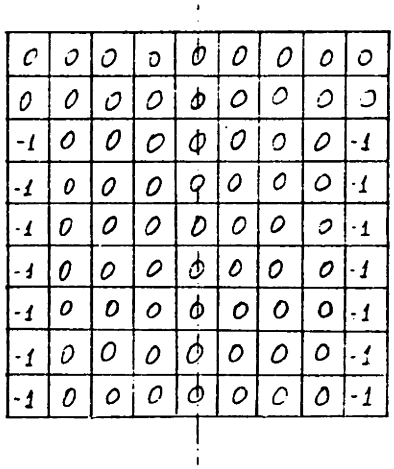


FIGURE A1.2 Station with containers on the center line.

This file contains the loading schedule, that is the:

i) number of ports to be visited,

ii) number of containers per port to be loaded (half of the total number),

iii) weights of containers per port by decreasing order of weight.

A sample file LOAD.DAT is given in appendix A2.

A1.2.3 Interactive inputs

During the run of the program the operator is asked:

i) To enter values for the minimum required GM, parameter K ("percentage rule") and the limits within the longitudinal moment is acceptable,

ii) To choose if (s)he wants to make zero the trim from the beginning (before the loading starts - usually the answer to this question should be zero, because it results in less water ballast),

iii) To select the allocation discipline (to fill the vessel row by row or station by station).

After all the containers of a port are placed the operator has the possibility:

iv) To see the allocation of containers until that time,and,

v) To save the container allocation until that port in two external files (RESULT.DAT and OUTPUT.DAT).

A1.2.4 External Functions

These are some of the hydrostatic curves of the vessel as functions of the displacement. The necessary curves are those giving the LCB, MTC1 (moment to change trim one inch) and KM (distance of the metacenter from the keel).

The above three curves for the vessel we use are shown in figure A1.3; we have made linear approximation of them and constructed the external functions listed at the end of appendix A2.

A1.3 Sample session

In the following we present a sample run of the program. The lines noted by an asterisk (*) are the data and choices the operator has to perform. The computer prompt $ should be displayed on the screen before the run of the program commences:

```
*   $ RUN MAIN
      ENTER GMmin, K, EYILON
*     1.4 1. 10000.
      DO  YOU  WANT TO MAKE THE TRIM ZERO  BEFORE  THE  LOADING
      STARTS?
      IF YES ENTER 1, IF NO ENTER 0
*     0
      SELECT ALLOCATION DISCIPLINE
      ENTER    1  IF YOU WANT TO BLOCK CONTAINERS FROM THE  SAME
                  ORIGIN
               0  IF YOU WANT THE VESSEL TO BE FILLED ROW BY ROW
*     1
      PORT # 1
      CONTAINER # 1
      .
      .
```

```
      .
      CONTAINERS OF PORT # 1 HAVE BEEN PLACED
      DO YOU WANT TO SEE THE ALLOCATION UNTIL NOW?
      ENTER 1 FOR YES, 0 FOR NO.
*     1
      ENTER THE NUMBER (1-22) OF STATION YOU WANT TO SEE
*     15
        0.0    0.0    0.0    0.0    0.0    0.0
          0      0      0      0      0      0
       12.0   12.0   12.0   12.0   12.0   12.0
          7      7      7      7      7      7
        8.0    8.0    8.0    8.0    8.0    0.0
          3      3      3      3      3     -1
       10.0    8.0    8.0    8.0    8.0    0.0
          1      3      3      3      3     -1
       11.0   10.0   10.0   10.0   10.0    0.0
          2      1      1      1      1     -1
       11.0   11.0   11.0   11.0   11.0    0.0
          2      2      2      2      2     -1
       12.0   12.0   12.0   12.0   12.0    0.0
          4      4      4      4      4     -1
       12.0   12.0   12.0   12.0    0.0    0.0
          4      4      4      4     -1     -1
       12.0   12.0   12.0   12.0    0.0    0.0
          1      1      1      4     -1     -1
      DO YOU WANT TO SEE ANOTHER STATION?
      ENTER 1 FOR YES, 0 FOR NO.
*     0
      DO YOU WANT TO SAVE THE RESULT IN AN OUTPUT FILE?
      ENTER 1 FOR YES, 0 FOR NO.
*     1
      PORT # 2
      .
      .
      .
      FORTRAN STOP
      $
```

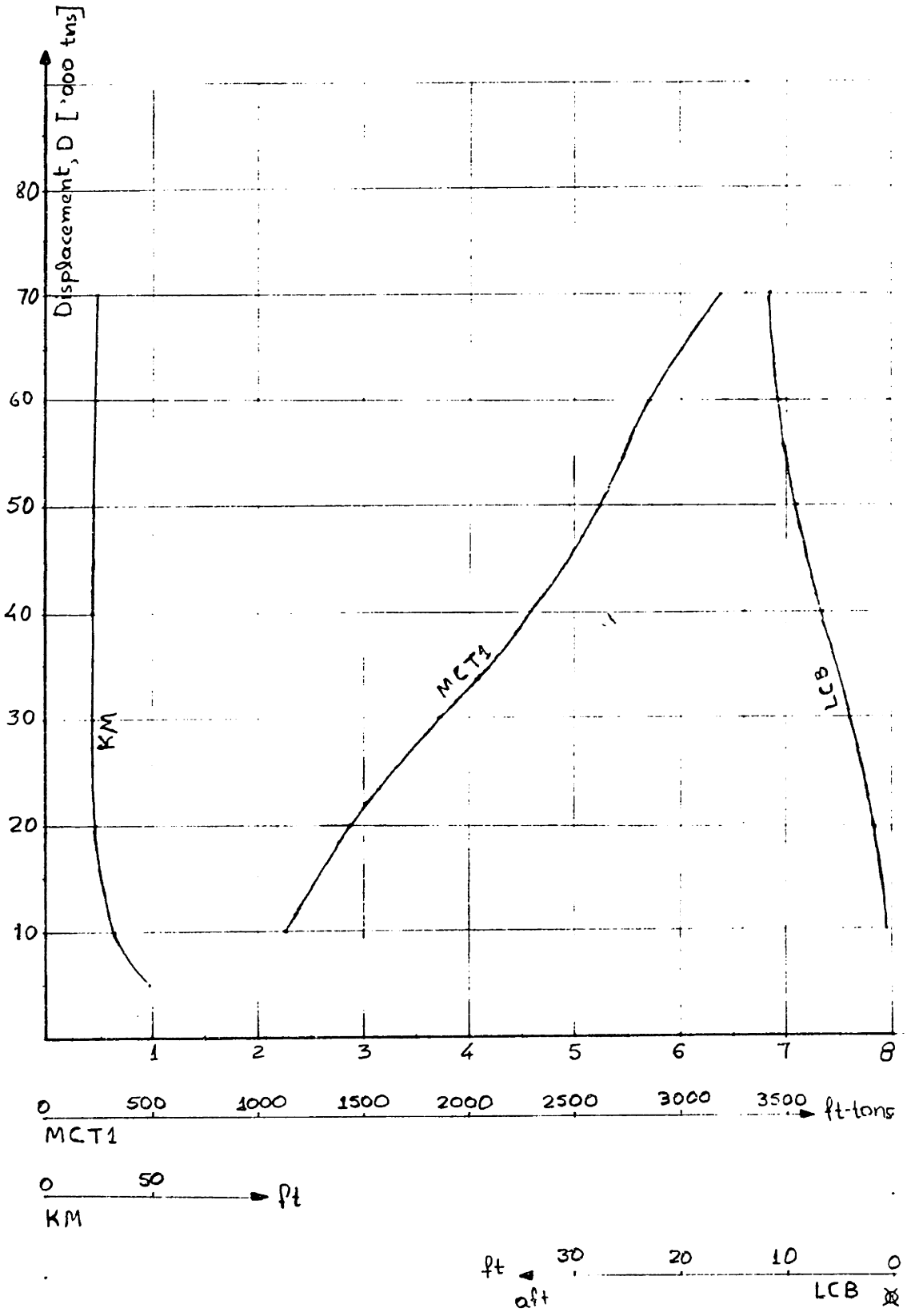


FIGURE A1.3 Some of the hydrostatic curves of the vessel used to test the algorithm.

APPENDIX A2

A2.1 Sample file SHIP.DAT

```
DISPLACEMENT
12881.
INITIAL MOMENT
-508799.5
GM,         KG
20.72       36.78
NUMBER OF TANKS
  7
CAPACITY    X-COORD     Z-COORD
493.        17.4        29.
1544.       93.         21.2
298.        179.5       3.
3080.       259.3       22.1
2080.       346.7       28.
835.        606.6       19.
951.        653.3       39.5
DIMENSIONS NXYZ-ROWS ABOVE OR BELOW DECK
 22  6  2  7  9
X-COORDINATES OF COMTAINER STATIONS
58.71
78.28
102.76
122.33
227.49
247.06
271.55
291.12
315.59
335.16
359.63
379.20
403.66
423.23
447.70
467.27
492.73
511.30
535.77
555.34
579.80
599.37
Z--COORDINATES OF CONTAINER ROWS
9.10
16.31
23.52
30.73
37.94
45.14
52.79
67.35
74.56
NF-1 IF THERE IS CONTAINER ON THE CL
  0
```

HEIGHT OF CONTAINERS
7.21

STATION :
1

0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	-1	-1
0	0	-1	-1	-1	-1
0	0	-1	-1	-1	-1
0	0	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1

STATION :
2

0	0	0	0	0	-1
0	0	0	0	0	-1
				-1	-1
		-1	-1	-1	-1
		-1	-1	-1	-1
		-1	-1	-1	-1
-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1

STATION :
3

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	-1
0	0	0	0	0	-1
		-1	-1	-1	-1
		-1	-1	-1	-1
-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1

STATION :
4

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	-1
0	0	0	0	0	-1
		-1	-1	-1	-1
		-1	-1	-1	-1
-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1

STATION :
5

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	-1	-1
0	0	0	0	-1	-1

STATION :
6

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	-1	-1
0	0	0	0	-1	-1

STATION :
7

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1

STATION :
8

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1

STATION :
9

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1

STATION :
10

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1

STATION :
11

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1

STATION :
12

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1

STATION :
13

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	-1	-1
0	0	0	0	-1	-1

STATION :
14

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	-1	-1
0	0	0	0	-1	-1

STATION :
15

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	-1	-1
0	0	0	0	-1	-1
0	0	0	0	-1	-1
0	0	0	-1	-1	-1
0	0	-1	-1	-1	-1

STATION :
16

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	-1	-1
0	0	0	0	-1	-1
0	0	0	0	-1	-1
0	0	0	-1	-1	-1
0	0	-1	-1	-1	-1

STATION :
17

0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	-1	-1
0	0	0	-1	-1	-1
0	0	0	-1	-1	-1
0	0	-1	-1	-1	-1
0	0	-1	-1	-1	-1
0	0	-1	-1	-1	-1
0	-1	-1	-1	-1	-1

STATION :
18

0	0	0	0	0	-1
0	0	0	0	0	-1
0	0	0	0	-1	-1
0	0	0	-1	-1	-1
0	0	0	-1	-1	-1
0	0	-1	-1	-1	-1
0	0	-1	-1	-1	-1
0	0	-1	-1	-1	-1
0	-1	-1	-1	-1	-1

STATION :
19

0	0	0	0	-1	-1
0	0	0	0	-1	-1
-1	0	0	0	-1	-1
-1	0	0	-1	-1	-1
-1	0	-1	-1	-1	-1
-1	0	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1

STATION :
20

0	0	0	0	-1	-1
0	0	0	0	-1	-1
-1	0	0	0	-1	-1
-1	0	-1	-1	-1	-1
-1	0	-1	-1	-1	-1
-1	0	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1

STATION :
21

-1	0	0	-1	-1	-1
-1	0	0	-1	-1	-1
-1	0	0	0	-1	-1
-1	0	0	0	-1	-1
-1	0	0	-1	-1	-1
-1	0	0	-1	-1	-1
-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1

STATION :
22

-1	0	0	-1	-1	-1
-1	0	0	-1	-1	-1
-1	0	0	-1	-1	-1
-1	0	0	-1	-1	-1
-1	0	0	-1	-1	-1
-1	0	0	-1	-1	-1
-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1

A2.2 Sample file LOAD.DAT

```
 NUMBER OF PORTS
   5
   5   5   5  10  10
 CONTAINERS GOING TO PORT:
      1.
 10.
 10.
 10.
 10.
 10..
 CONTAINERS GOING TO PORT:
      2.
 11.
 11.
 11.
 11.
 11.
 CONTAINERS GOING TO PORT:
      3.
 8.
 8.
 8.
 8.
 8.
 CONTAINERS GOING TO PORT:
      4.
 12.
 12.
 12.
 12.
 12.
 12.
 12.
 12.
 12.
 12.
 CONTAINERS GOING TO PORT:
      5.
 10.
 10.
 10.
 10.
 10.
 10.
 10.
 10.
 10.
 10.
```

A2.3 Program listing

```
C*********************************************************************
C                                                                     C
C     PROGRAM  DIMITRA : OPTIMAL CONTAINER LOADING                    C
C                                                                     C
C*********************************************************************
      PROGRAM AXA
C
C  OPTIMAL CONTAINER LOADING WITH RESPECT TO THE CONTAINER HANDLING
C                            COST
C  The program can accomodate vessels with capacity up to 1500 con-
C  tainers visiting up to 30 ports. Any restriction in the size of
C  the problem can be easily overcome by increasing the size of ar-
C  rays and matrices.
C  Note: Some variables (KK1F,KK1B,KK2F,KK2B e.t.c.) take on the
C  value they have in the specific vessel we have considered; these
C  variables should be corrected if another vessel is used or a ge-
C  neral procedure should be developed for their calculation.

      COMMON/SHIP1/D,XMOMNT,NT,YL(10),CTX(10),CTZ(10),Y(10),NF
      COMMON/SHIP2/IIT,JJT,KKB,KKA,KKT,MC1,MC2
      COMMON/SHIP3/CX(30),CZ(15),NXYZ(30,20,15),WXYZ(30,20,15)
      COMMON/SHIP4/XLB2,XLB3,GM2,GM3,XKG2,XKG3,XKM2,XKM3,GMIN
      COMMON/SHIP5/H
      COMMON/LOAD1/W(1500),NS(20)
      COMMON/LOAD2/NB1,NB2,NA1,NA2,NB,NA
      COMMON/LOAD3/NTB,NTI,DICTIS,EYILON
      COMMON/LOAD4/II1,II2,JJ1B,JJ1F,JJ2B,JJ2F,KK1B,KK1F,
     *KK2B,KK2F,IH,IL,NP
      COMMON/LOAD5/IPOINT,CASE,XX1,XX2
      COMMON/LOAD6/BC1(30,20),CC1(30,20),RC1(30,20),IO1(600,2),
     *BC2(30,20),CC2(30,20),RC2(30,20),IO2(600,2)
      COMMON/NORMA1/N22,N33,N222,N333
      COMMON/BLOCK1/ICHOIC,IBAL
      COMMON/BLOCK2/KBLF,KBLB,JBLF,JBLB,IBLF,IBLB
      COMMON/WATBA1/IT1,IT2
      COMMON/OUTPU1/ISTAT(30)
      COMMON/OUTPU2/PTR(20),PDC(20),PGM(20),DP(20),N(20)

C Initialization
      DO 2 I=1,30
      ISTAT(I)=0
      DO 2 J=1,20
      BC1(I,J)=0
      RC1(I,J)=0
      CC1(I,J)=0
      BC2(I,J)=0
      CC2(I,J)=0
      RC2(I,J)=0
      K=(I-1)*20+J
      IO1(K,1)=0
      IO1(K,2)=0
      IO2(K,1)=0
      IO2(K,2)=0
   2  CONTINUE

C Open input/output units

      OPEN(UNIT=2,NAME='SHIP',TYPE='OLD',FORM='FORMATTED')
```

```
      OPEN(UNIT=3,NAME='LOAD',TYPE='OLD',FORM='FORMATTED')
      OPEN(UNIT=4,NAME='RESULT',TYPE='NEW',FORM='FORMATTED')
      OPEN(UNIT=7,NAME='OUTPUT',TYPE='NEW',FORM='FORMATTED')

C Read vessel characteristics from unit 2

      READ(2,9900)ZZZZ
9900  FORMAT(A40)
      READ(2,9999)D
9999  FORMAT(F12.4)
      XLB2=XLCB(D)
      READ(2,9900)ZZZZ
      READ(2,9999)XMOMNT
      READ(2,9900)ZZZZ
      READ(2,9998)GM2,XKG2
      GM3=GM2
      XKG3=XKG2
      READ(2,9900)ZZZZ
      READ(2,9991)NT
9991  FORMAT(I3)
      READ(2,9900)ZZZZ
      DO 3 I=1,NT
   3  READ(2,9998)YL(I),CTX(I),CTZ(I)
9998  FORMAT(4F10.3)
      READ(2,9900)ZZZZ
      READ(2,9992)IIT,JJT,KKA,KKB,KKT
9992  FORMAT(5I3)
      READ(2,9900)ZZZZ
      READ(2,9999)(CX(I),I=1,IIT)
      READ(2,9900)ZZZZ
      READ(2,9999)(CZ(I),I=1,KKT)
      READ(2,9900)ZZZZ
      READ(2,9991)NF
      READ(2,9900)ZZZZ
      READ(2,9999)H
      DO 1 I=1,IIT
      READ(2,9900)ZZZZ
      READ(2,9991)ISTA
      DO 1 K=1,KKT
      KT=KKT-K+1
      READ(2,9993)(NXYZ(I,J,KT),J=1,JJT)
9993  FORMAT(10I5)
   1  CONTINUE

      write(6,8889)
8889  FORMAT(' ENTER GMIN, K and EYILON; [-EYILON,+EYILON] is the
     * range within the moment is   acceptable.')
      READ(5,*)GMIN,CON,EYILON

C Read data  for the loading schedule from unit 3

      READ(3,9900)ZZZZ

C Read number of ports

      READ(3,9991)M
      write(6,548)M
548   FORMAT('  NUMBER OF PORTS=',I3)
```

```
C Read number of containers to be loaed at each port

      READ(3,9994)(N(I),I=1,M)
9994  FORMAT(15I4)
      DO 10 I=1,M
      IF(I.EQ.1)GO TO 11
      I1=I-1
      NS(I)=NS(I1)+N(I)
      write(6,549)I,NS(I)
      GO TO 10
  11  NS(I)=N(I)
      write(6,549)I,NS(I)
549   FORMAT('  I=',I2,' NS(I)=',I4)
  10  CONTINUE

C Read the weight of the containers
      DO 12 I=1,M
      I1=I-1
      IF(I.EQ.1)MN1=1
      IF(I.GT.1)MN1=NS(I1)+1
      MN2=NS(I)
      READ(3,9900)ZZZZ
      READ(3,9999)BAM
      DO 12 J=MN1,MN2
      READ(3,9999) W(J)
  12  CONTINUE

      XLB2=XLB3
      XLB3=XLCB(D)
      DO 13 I3=2,NT
      I33=I3-1
      IF((CTX(I33).LE.XLB3).AND.(CTX(I3).GE.XLB3)) THEN
      NTB=I33
      GO TO 14
      ENDIF
  13  CONTINUE
  14  CONTINUE

      write(6,*)' DO YOU WANT TO MAKE THE TRIM ZERO BEFORE THE
     * LOADING STARTS?'
      write(6,*)' IF YES ENTER 1, IF NO ENTER 0'
      READ(5,560)IBAL
 560  FORMAT(I3)
      IF(IBAL.EQ.0) GO TO 15
      CALL  WATBAL
  15  write(6,543)(Y(JKL),JKL=1,NT)
543   FORMAT(6F12.3,/,6F12.3)

C  -DECIDE WHICH CONTAINERS WILL BE PLACED BELOW THE DECK
C  -CALCULATE AVAILABLE POSITIONS IN FRONT AND BEHIND OF THE FINAL
C   XLCB(XLB4), BELOW AND ABOVE THE DECK.

      DF=D
      M1=M
      DO 50 I=1,M
      IF(I.EQ.1)GO TO 51
      I1=I-1
      IA=NS(I1)+1
  51  CONTINUE
```

```
      IB=NS(I)
      IF(I.EQ.1)IA=1
      DI=0.0
       DO 60 J=IA,IB
   60  DI=DI+2*W(J)
      DP(I)=DI
   50 DF=DF+DP(I)
      XLB4=XLCB(DF)

      NB1=0
      NB2=0
      DO 70 I=1,IIT
      DO 70 J=1,JJT
      DO 70 K=1,KKB
      IF(NXYZ(I,J,K).LT.0) GO TO 70
      IF(CX(I).LE.XLB4)NB1=NB1+1
      IF(CX(I).GT.XLB4)NB2=NB2+1
   70 CONTINUE
      NA1=0
      NA2=0
      KKB1=KKB+1
      DO 80 I=1,IIT
      DO 80 J=1,JJT
      DO 80 K=KKB1,KKT
      IF(NXYZ(I,J,K).LT.0) GO TO 80
      IF(CX(I).LE.XLB4)NA1=NA1+1
      IF(CX(I).GT.XLB4)NA2=NA2+1
   80 CONTINUE
      NB=NB1+NB2
      NA=NA1+NA2

      DO 90 I=1,M
      IF(NS(I).GE.NB) GO TO 91
   90 CONTINUE
      GO TO 94
   91 M2=I+1
      M1=I
      WRITE(4,8888)M1,M1,M2
      write(6,8888)M1,M1,M2
 8888 FORMAT(' SOME OF THE CONTAINERS TO PORT',I3,'  WILL BE LOADED
     * ABOVE THE DECK --- PORTS ',I3,'  AND',I3,'  COINCIDE')
      M=M+1
      DO 92 I=M2,M
      MM=M-I+M2
      MM1=MM-1
      DP(MM)=DP(MM1)
   92 NS(MM)=NS(MM1)
      M11=M1-1
      NS(M1)=NB
      J=NS(M11)+1
      K=NS(M1)

      DO 93 I=J,K
   93 DP(M1)=DP(M1)+W(I)
      DP(M2)=DP(M2)-DP(M1)
   94 CONTINUE
C
C INITIAL RANKING OF COLUMNS -- ALL COLUMNS ARE GIVEN ARBITRARILY
C  ZERO  BENEFIT-COST RATIO.
```

```
      ID1=0
      ID2=0
      DO 95 I=1,IIT
      DO 95 J=1,JJT
      DO 96 K=1,KKB
      IF(NXYZ(I,J,K).LT.0) GO TO 96
      ID1=ID1+1
      IO1(ID1,1)=I
      IO1(ID1,2)=J
      GO TO 97
   96 CONTINUE
   97 CONTINUE
      KKB1=KKB+1
      DO 98 K=KKB1,KKT
      IF(NXYZ(I,J,K).LT.0) GO TO 98
      ID2=ID2+1
      IO2(ID2,1)=I
      IO2(ID2,2)=J
      GO TO 95
   98 CONTINUE
   95 CONTINUE
      MC1=ID1
      MC2=ID2
C
C  START LOADING
C  A refined version of the program should use a general method to
C  calculate the values of KK1B,KK1F,KK2B,KK2F,KBLB,KBLF. These
C  variables correspond to the aft and front stations in the rear
C  and front part of the vessel (the four first) or to the nearest
C  stations left and right to LCB (KBLB,KBLF)
C
      X1=0
      X2=0
      XA1=0
      XA2=0
      NP=0
      N22=0
      N33=0
      II1=1
      II2=1
      JJ1B=0
      JJ1F=0
      KK1B=1
      KK1F=10
      JJ2B=0
      JJ2F=0
      KK2B=11
      KK2F=22
      KBLF=11
      KBLB=10
      JBLF=0
      JBLB=0
      IBLF=1
      IBLB=1

      ICHOIC=0
      write(6,*) ' SELECT CONTAINER ASSIGNEMENT METHOD'
      write(6,*) '  1 IF YOU WANT TO BLOCK CONTAINERS  FROM THE
     1 SAME ORIGIN'
```

```
      write(6,*) '  0 IF YOU WANT THE VESSEL TO BE FILLED ROW BY ROW'
      READ(5,*)ICHOIC

      N22=0
      N33=0
      N222=0
      N333=0
      IPOINT=1
      NS1=NS(M)

   99 DO 101 I=1,NS1
      write(6,547)I
  547 FORMAT(/,'  CONTAINER=',I3)
      IF(I.EQ.1) GO TO 105
      IF(I.LE.NS(NP)) GO TO 102
  105 NP=NP+1
      IPP=NP-1
      IF(I.EQ.1)IH=1
      IF(I.GT.1)IH=NS(IPP)+1
      IL=0
      DC=D+DP(NP)
      D=DC
      XLB2=XLB3
      XLB3=XLCB(DC)
      XKM3=XKM(DC)
      XKG3=XKG3*(D-DP(NP))/D
      XMOMNT=(XLB3-XLB2)*D+XMOMNT
      IF(IPOINT.GT.2.)IPOINT=2
      DO 103 I3=2,NT
      I33=I3-1
      IF((CTX(I33).LE.XLB3).AND.(CTX(I3).GE.XLB3)) THEN
      NTB=I33
      GO TO 102
      ENDIF
   103 CONTINUE
   102 CONTINUE

      IF(I.GT.NS(M1)) THEN
      XX1=XA1
      XX2=XA2
      NN1=NA1
      NN2=NA2
      ELSE
      XX1=X1
      XX2=X2
      NN1=NB1
      NN2=NB2
      ENDIF

      write(6,*)' MOMENT BEFORE LOADING=',XMOMNT

      IF(IT1.EQ.1.OR.IT2.EQ.1)THEN
      IF(IPOINT.GE.2) GO TO 104
      IF((XX1/NN1).GE.(XX2/NN2+CON).OR.(XX1/NN1).LT.(XX2/NN2-CON))
     1 GO TO 111
      IF(IT1.EQ.1)CASE=0
      IF(IT2.EQ.1)CASE=5
      HMOMNT=XMOMNT
      CALL NORMAL
```

```
      HMOMN1=XMOMNT
      XMOMNT=HMOMN1-HMOMNT
c     write(6,543)(Y(JKL),JKL=1,NT)
      CALL WATBAL
c     write(6,543)(Y(JKL),JKL=1,NT)
      XMOMNT=XMOMNT+HMOMNT
      GO TO 1000
      ENDIF
 111  CONTINUE

      IF(IPOINT.GE.2) GO TO 104
      IF((XMOMNT.LT.0.0).AND.((XX1/NN1).LT.(XX2/NN2)))CASE=4
      IF((XMOMNT.LT.0.0).AND.((XX1/NN1).GE.(XX2/NN2)))CASE=3
      IF((XMOMNT.GE.0.0).AND.((XX1/NN1).LT.(XX2/NN2)))CASE=2
      IF((XMOMNT.GE.0.0).AND.((XX1/NN1).GE.(XX2/NN2)))CASE=1
      IF((XMOMNT.LT.0.0).AND.((XX1/NN1).LT.(XX2/NN2-CON)))IPOINT=2
      IF((XMOMNT.GT.0.0).AND.((XX1/NN1-CON).GT.(XX2/NN2)))IPOINT=2

 104  GO TO (100,200,400,400)IPOINT
 100  CONTINUE
      CALL NORMAL
      IF((N22.EQ.1).OR.(N33.EQ.1))CALL  WATBAL
      GO TO 1000

 200  CONTINUE
      IF(XMOMNT.LT.0.0) THEN
      CASE=2
      CALL  WATBAL
c     write(6,543)(Y(JKL),JKL=1,NT)
      IF(NTI.EQ.1) GO TO 300
      CALL NORMAL
      IF((XX1/NN1).GE.(XX2/NN2-CON/2))IPOINT=1
      ELSEIF(XMOMNT.GT.0.0)THEN
      CASE=3
      CALL WATBAL
c     write(6,543)(Y(JKL),JKL=1,NT)
      IF(NTI.EQ.1) GO TO 300
      CALL NORMAL
      IF((XX2/NN2).GE.(XX1/NN1-CON/2)) IPOINT=1
      ENDIF
      GO TO 1000

 300  CONTINUE
      IF((XX1/NN1).GT.(XX2/NN2)) THEN
      CASE=1
      CALL NORMAL
      IPOINT=3
      ELSE
      CASE=4
      CALL NORMAL
      IPOINT=4
      ENDIF
      SIGN1=SIG(XMOMNT)
      GO TO 1000

 400  CONTINUE
      IF(IPOINT.EQ.3)CASE=3
      IF(IPOINT.EQ.4)CASE=2
      CALL NORMAL
```

```
      SIGN2=SIG(XMOMNT)
      IF(((XX2/NN2-CON/2).LE.(XX1/NN1)).AND.
     *((XX1/NN1).LE.(XX2/NN2-CON/2))) THEN
      IPOINT=1
      ELSE
      IF((SIGN1*SIGN2).LT.0) IPOINT=2
      ENDIF
1000  CONTINUE

      IF(I.GT.NS(M1)) THEN
      XA1=XX1
      XA2=XX2
      ELSE
      X1=XX1
      X2=XX2
      ENDIF
      IF(I.EQ.NS(NP)) GO TO 2000
      GO TO 101

C ******* REDUCTION OF THE WATER BALLAST *******
C  This part ( up to command 2051) is never executed because
C  subroutine WATBAL secures that the vessel has water ballast
C  only in one part (front or rear).
C

2000  BB=0
      BS=0
      DC=D
      XLB2=XLB3
      DO 2010 II=1,NT
      IF(CTX(II).LT.XLB3) THEN
      BB=BB+Y(II)*(XLB3-CTX(II))
      ELSE
      BS=BS+Y(II)*(CTX(II)-XLB3)
      ENDIF
2010  CONTINUE
      IF(BB.LE.BS) THEN
      DO 2020 II=1,NT
      IF(CTX(II).GT.XLB3) GO TO 2021
      XKG2=XKG3
      XKG3=(XKG2*DC-CTZ(II)*Y(II)**2/YL(II))/(DC-Y(II))
      DC=DC-Y(II)
      Y(II)=0
2020  CONTINUE
2021  CONTINUE
      DO 2030 III=1,NT
      II=NT-III+1
      IF(BB.EQ.0) GO TO 2031
      IF(CTX(II).LT.XLB3)GO TO 2031
      DY=BB/ABS(CTX(II)-XLB3)
      IF(DY.LE.Y(II)) THEN
      XKG2=XKG3
      XKG3=(XKG2*DC-CTZ(II)*(Y(II)**2-(Y(II)-DY)**2)/YL(II))/(DC-DY)
      DC=DC-DY
      Y(II)=Y(II)-DY
      BB=0
      ELSE
      XKG2=XKG3
      XKG3=(XKG2*DC-CTZ(II)*Y(II)**2/YL(II))/(DC-Y(II))
```

```
      DC=DC-Y(II)
      BB=BB-Y(II)*ABS(CTX(II)-XLB3)
      Y(II)=0
      ENDIF
2030  CONTINUE
2031  CONTINUE
      ELSE
      DO 2040 III=1,NT
      II=NT-III+1
      IF(CTX(II).LT.XLB3) GO TO 2041
      XKG2=XKG3
      XKG3=(XKG2*DC-CTZ(II)*Y(II)**2/YL(II))/(DC-Y(II))
      DC=DC-Y(II)
      Y(II)=0
2040  CONTINUE
2041  CONTINUE
      DO 2050 II=1,NT
      IF(BS.EQ.0) GO TO 2051
      IF(CTX(II).GT.XLB3) GO TO 2051
      DY=BS/ABS(CTX(II)-XLB3)
      IF(DY.LE.Y(II)) THEN
      XKG2=XKG3
      XKG3=(XKG2*DC-CTZ(II)*(Y(II)**2-(Y(II)-DY)**2)/YL(II))/(DC-DY)
      DC=DC-DY
      Y(II)=Y(II)-DY
      BS=0
      ELSE
      XKG2=XKG3
      XKG3=(XKG2*DC-CTZ(II)*Y(II)**2/YL(II))/(DC-Y(II))
      DC=DC-Y(II)
      BS=BS-Y(II)*ABS(CTX(II)-XLB3)
      Y(II)=0
      ENDIF
2050  CONTINUE
2051  CONTINUE
      ENDIF

C *** Correction of the trim ***

      D=DC
      YTI=0
      DO 2060 II=1,NT
2060  YTI=YTI+Y(II)
      XLB2=XLB3
      XLB3=XLCB(D)
      XMOMNT=XMOMNT+D*(XLB3-XLB2)
c     write(6,543) (Y(JKL),JKL=1,NT)
      IF(ABS(XMOMNT).GT.EYILON)CALL WATBAL
c     write(6,543) (Y(JKL),JKL=1,NT)
      YT2=0
      DO 2070 II=1,NT
2070  YT2=YT2+Y(II)
      IF(ABS(YT2-YTI).GT.(0.1*YTI)) GO TO 2000
      GM3=XKM(D)-XKG3
3000  CONTINUE

C ******* CHECKING AND CORRECTION OF THE GM ********
      write(6,532)GM3,GMIN
532   FORMAT(' GM=',F6.2,' GMmin=',F6.2)
```

```
      IF(GM3.GE.GMIN) GO TO 3001
      IF(I.LE.NS(M1)) THEN
      LP=1
      CALL GMCOR(IO1,BC1,CC1,RC1,LP)
      write(6,*)' KG3=',XKG3
      ELSE
      LP=2
      CALL GMCOR(IO2,BC2,CC2,RC2,LP)
      write(6,*)' KG3=',XKG3
      ENDIF
      write(6,539)GM3
539   FORMAT('  GM=',F6.2)
      IF((DICTIS.EQ.1).AND.(LP.EQ.2)) THEN
      LP=1
      CALL GMCOR(IO1,BC1,CC1,RC1,LP)
      LP=2
      ENDIF
      write(6,*)' KG3=',XKG3
      write(6,539)GM3
      IF(DICTIS.EQ.1)write(6,*)' WARNING: GM CONSTRAINT NOT SATISFIED'

3001  PTR(NP)=XMOMNT
      PDC(NP)=D
      PGM(NP)=GM3

      write(6,8887)
8887  FORMAT('  DO YOU WANT TO SEE THE ALLOCATION UNTIL NOW?',/,
     *' ENTER 1 FOR YES, 0 FOR NO.')
      READ(5,7777)LL
7777  FORMAT(I3)
      IF(LL.EQ.1) THEN
3003  write(6,8886)
8886  FORMAT('  GIVE THE NUMBER (1 - IIT) OF THE STATION YOU
     1 WANT TO SEE')
      READ(5,7777)LI
      DO 3020 LK1=1,KKT
      LK=KKT-LK1+1
      write(6,8885)(WXYZ(LI,LJ,LK),LJ=1,JJT)
      write(6,8884)(NXYZ(LI,LJ,LK),LJ=1,JJT)
8885  FORMAT(10F8.2)
8884  FORMAT(10(3X,I3,2X))
3020  CONTINUE
      write(6,8883)
8883  FORMAT(' DO YOU WANT TO SEE ANOTHER STATION?',/,
     *' ENTER 1 FOR YES, 0 FOR NO.')
      READ(5,7777)LL
      IF(LL.EQ.1) GO TO 3003
      ENDIF
```

```
      write(6,*)' DO YOU WANT TO SAVE THE RESULTS IN AN EXTERNAL FILE?'
      write(6,*)' ENTER 1 FOR YES, 0 FOR NO.'
      READ(5,*)ITYPE
      IF(ITYPE.EQ.0) GO TO 101
      WRITE(4,6667)NP
 6667 FORMAT(///'  PORT #  ',I3)
      DO 4000 LL=1,IIT
      WRITE(4,6666)LL
 6666 FORMAT(/12X,' STATION #',I3)
      DO 4000 LK1=1,KKT
      LK=KKT-LK1+1

      WRITE(4,7654) (WXYZ(LL,LJ,LK),LJ=1,JJT)
 7654 FORMAT(/,10F8.2)
      WRITE(4,8884) (NXYZ(LL,LJ,LK),LJ=1,JJT)
 4000 CONTINUE
      CALL OUTPUT

  101 CONTINUE
      STOP
      END
```

```
C*********************************************************************
      SUBROUTINE NORMAL
C*********************************************************************

C     This subroutine realizes the first allocation discipline.
C     The vessel is filled row by row. If the second allocation
C     discipline has been chosen, it calls SUBROUTINE BLOCK.

C     Note: Look for comments on the way variables KK1B,KK1F,
C     KK2B,KK2F (and KK,KH) take their values in the beginning
C     of the MAIN program.


      COMMON/SHIP1/D,XMOMNT,NT,YL(10),CTX(10),CTZ(10),Y(10),NF
      COMMON/SHIP2/IIT,JJT,KKB,KKA,KKT,MC1,MC2
      COMMON/SHIP3/CX(30),CZ(15),NXYZ(30,20,15),WXYZ(30,20,15)
      COMMON/SHIP4/XLB2,XLB3,GM2,GM3,XKG2,XKG3,XKM2,XKM3,GMIN
      COMMON/LOAD1/W(1500),NS(20)
      COMMON/LOAD4/II1,II2,JJ1B,JJ1F,JJ2B,JJ2F,KK1B,KK1F,
     *KK2B,KK2F,IH,IL,NP
      COMMON/LOAD5/IPOINT,CASE,X1,X2
      COMMON/LOAD6/BC1(30,20),CC1(30,20),RC1(30,20),IO1(600,2)
     *,BC2(30,20),CC2(30,20),RC2(30,20),IO2(600,2)
      COMMON/REAR1/II,KK,J
      COMMON/NORMA1/N22,N33,N222,N333
      COMMON/ORDER1/WC
      COMMON/BLOCK1/ICHOIC
      COMMON/BLOCK2/KBLF,KBLB,JBLF,JBLB,IBLF,IBLB
C
      NCP=NS(NP)
      F=2
      IF(ICHOIC.EQ.1) THEN
      CALL BLOCK
      GO TO 100
      ENDIF
    1 CONTINUE
      IF(CASE.EQ.4) THEN
      II=II2
      JH=JJ2B
      JJ=JJ2F
      KH=KK2B
      KK=KK2F
      N1=1
      WC=W(IH)
      IH=IH+1
      ELSEIF(CASE.EQ.3.OR.CASE.EQ.5) THEN
      II=II2
      JH=JJ2F
      JJ=JJ2B
      KH=KK2F
      KK=KK2B
      N1=-1
      WC=W(IH)
      IH=IH+1
      IF(CASE.EQ.3) THEN
      WC=W(NCP-IL)
      IL=IL+1
      IH=IH-1
      ENDIF
```

```
      ELSEIF(CASE.EQ.2.OR.CASE.EQ.0) THEN
      II=II1
      JH=JJ1B
      JJ=JJ1F
      KH=KK1B
      KK=KK1F
      N1=1
      WC=W(IH)
      IH=IH+1
      IF(CASE.EQ.2) THEN
      WC=W(NCP-IL)
      IL=IL+1
      IH=IH-1
      ENDIF
      ELSEIF(CASE.EQ.1) THEN
      II=II1
      JH=JJ1F
      JJ=JJ1B
      KH=KK1F
      KK=KK1B
      N1=-1
      WC=W(IH)
      IH=IH+1
      ENDIF
      IPOIN1=0
      JJH=JJ+1
      IF(JJH.GT.JJT) GO TO 10
      IF((NXYZ(KK,JJH,II).NE.0).AND.(JJH.EQ.1))JJH=2
      IF(NXYZ(KK,JJH,II).NE.0) GO TO 10
      JJ=JJH
      GO TO 40
10    CONTINUE
      IF((N1*KK).GT.(N1*KH)) GO TO 20
      JJH=JH+1
      IF(JJH.GT.JJT)GO TO 30
      IF((NXYZ(KH,JJH,II).NE.0).AND.(JJH.EQ.1))JJH=2
      IF(NXYZ(KH,JJH,II).NE.0) GO TO 30
      JH=JJH
      IPOIN1=1
      GO TO 40
20    KK=KK-N1
      IF(KK.EQ.KH)GO TO 10
      JJ=1
      IF(NXYZ(KK,JJ,II).NE.0)JJ=2
      IF(NXYZ(KK,JJ,II).NE.0) GO TO 10
      GO TO 40
30    CONTINUE
      IF(II.LT.KKB) THEN
31    NK=0
      NJ=0
      II=II+1
      JH=0
      IF(CASE.EQ.4) THEN
      KH=11
      KK=22
      ELSEIF(CASE.EQ.3.OR.CASE.EQ.5) THEN
      KH=22
      KK=11
```

```
      ELSEIF(CASE.EQ.2.OR.CASE.EQ.0) THEN
      KH=1
      KK=10
      ELSEIF(CASE.EQ.1) THEN
      KH=10
      KK=1
      ENDIF
      IF(NXYZ(KH,1,II).NE.0)JH=1
   32 NJ=0
      JJ=1
      IF(NXYZ(KK,1,II).NE.0)JJ=2
      IF(NXYZ(KK,2,II).NE.0) THEN
      KK=KK-N1
      IF((N1*KK).LT.(N1*KH))NK=1
      NJ=1
      ENDIF
      IF(NK.EQ.1) GO TO 31
      IF(NJ.EQ.1) GO TO 32
      ELSEIF(II.EQ.KKB) THEN
      CASE1=CASE
      IF(CASE.LE.2) THEN
      CASE=5
      N22=1
      IF(N33.EQ.1)GO TO 34
      IH=IH-1
      GO TO 1
      ELSE
      N33=1
      IF(N22.EQ.1)GO TO 34
      CASE=0
      IH=IH-1
      GO TO 1
      ENDIF
      ENDIF
   34 CONTINUE

      IF(N22.EQ.1.AND.N33.EQ.1) THEN
      CASE=CASE1
   33 CONTINUE
      IF(II.GE.KKB.AND.II.LT.KKT)THEN
   41 NK=0
      NJ=0
      II=II+1
      JH=0
      IF(CASE.EQ.4) THEN
      KH=11
      KK=22
      ELSEIF(CASE.EQ.3.OR.CASE.EQ.5) THEN
      KH=22
      KK=11
      ELSEIF(CASE.EQ.2.OR.CASE.EQ.0) THEN
      KH=1
      KK=10
      ELSEIF(CASE.EQ.1) THEN
      KH=10
      KK=1
      ENDIF
      IF(NXYZ(KH,1,II).NE.0)JH1=1
```

```
 42   JJ=1
      NJ=0
      IF(NXYZ(KK,1,II).NE.0)JJ=2
      IF(NXYZ(KK,2,II).NE.0) THEN
      KK=KK-N1
      IF((N1*KK).LT.(N1*KH))NK=1
      NJ=1
      ENDIF
      IF(NK.EQ.1) GO TO 41
      IF(NJ.EQ.1) GO TO 42
      ELSEIF(II.EQ.KKT)THEN
      IF(CASE.LE.2)THEN
      CASE=5
      N222=1
      IF(N333.EQ.1)GO TO 50
      IH=IH-1
      GO TO 1
      ELSE
      CASE=0
      N333=1
      IF(N222.EQ.1)GO TO 50
      IH=IH-1
      GO TO 1
      ENDIF
      ENDIF
      ENDIF

 40   J=JJ
      IF(IPOIN1.EQ.1) J=JH
      IF(IPOIN1.EQ.1)KK=KH
 100  CONTINUE
      IF(J.EQ.1.AND.NF.EQ.1) F=1
      NXYZ(KK,J,II)=NP

      write(6,111)kk,j,ii,nxyz(kk,j,ii)
111   format(' Nxyz(',i2,',',i2,',',i2,')=',i2)
      write(6,*)'  wc=',wc

      XMOMNT=XMOMNT+WC*(CX(KK)-XLB3)*F
      WXYZ(KK,J,II)=WC
      DO 45 KLM=2,II
      KLM1=II-KLM+1
      IF(NXYZ(KK,J,KLM1).NE.NP)GO TO 46
      IF(WXYZ(KK,J,II).GT.WXYZ(KK,J,KLM1)) THEN
      WXYZ(KK,J,II)=WXYZ(KK,J,KLM1)
      WXYZ(KK,J,KLM1)=WC
      ENDIF
 45   CONTINUE
 46   CONTINUE

      XKG3=XKG3+WC*CZ(II)*F/D
      GM3=XKM(D)-XKG3

      IF(ICHOIC.EQ.1) GO TO 200
```

```
      IF(CASE.EQ.1) THEN
      II1=II
      JJ1F=JH
      JJ1B=JJ
      KK1F=KH
      KK1B=KK
      X1=X1+1
      ELSEIF(CASE.EQ.2.OR.CASE.EQ.0) THEN
      II1=II
      JJ1F=JJ
      JJ1B=JH
      KK1F=KK
      KK1B=KH
      X1=X1+1
      ELSEIF(CASE.EQ.3.OR.CASE.EQ.5) THEN
      II2=II
      JJ2B=JJ
      JJ2F=JH
      KK2B=KK
      KK2F=KH
      X2=X2+1
      ELSEIF(CASE.EQ.4) THEN
      II2=II
      JJ2B=JH
      JJ2F=JJ
      KK2B=KH
      KK2F=KK
      X2=X2+1
      ENDIF
200   FACTOR=1.
      CALL ORDER(KK,J,FACTOR)
      RETURN
 50   WRITE(6,*)' SHIP IS FULL , NO MORE CONTAINERS CAN BE LOADED'
      RETURN
      END
```

```
C*********************************************************************
      SUBROUTINE BLOCK
C*********************************************************************

C     This subroutine realizes the second allocation discipline.
C     Containers from the same origin are blocked. The vessel is
C     filled station by station.
C     Note: For comments on the way KBLB,KBLF (and k1) take their
C     values see the note in the MAIN program.

      COMMON/SHIP1/D,XMOMNT,NT,YL(10),CTX(10),CTZ(10),Y(10),NF
      COMMON/SHIP2/IIT,JJT,KKB,KKA,KKT,MC1,MC2
      COMMON/SHIP3/CX(30),CZ(15),NXYZ(30,20,15),WXYZ(30,20,15)
      COMMON/SHIP4/XLB2,XLB3,GM2,GM3,XKG2,XKG3,XKM2,XKM3,GMIN
      COMMON/LOAD1/W(1500),NS(20)
      COMMON/LOAD4/II1,II2,JJ1B,JJ1F,JJ2B,JJ2F,KK1B,KK1F,
     *KK2B,KK2F,IH,IL,NP
      COMMON/LOAD5/IPOINT,CASE,X1,X2
      COMMON/LOAD6/BC1(30,20),CC1(30,20),RC1(30,20),IO1(600,2)
     *,BC2(30,20),CC2(30,20),RC2(30,20),IO2(600,2)
      COMMON/REAR1/II,KK,J
      COMMON/NORMA1/N22,N33,N222,N333
      COMMON/ORDER1/WC
      COMMON/BLOCK1/ICHOIC
      COMMON/BLOCK2/KBLF,KBLB,JBLF,JBLB,IBLF,IBLB

C
 100  CONTINUE
      IF(CASE.GE.3) THEN
      I1=IBLF
      J1=JBLF
      K1=KBLF
      WC=W(IH)
      N1=1
      KCH=22
      IH=IH+1
      ELSE
      I1=IBLB
      J1=JBLB
      K1=KBLB
      WC=W(IH)
      N1=-1
      KCH=1
      IH=IH+1
      ENDIF

      IF(I1.LE.KKB) THEN
      KKC=KKB
      ELSE
      KKC=KKT
      ENDIF

      IF(I1.EQ.(KKB+1).AND.KLBF.EQ.11.AND.KBLB.EQ.10) THEN
      N22=0
      N33=0
      ENDIF

   1  CONTINUE
      JH1=J1+1
```

```
      IF(JH1.GT.JJT) GO TO 10
      IF(NXYZ(K1,JH1,I1).NE.0.AND.JH1.EQ.1) JH1=2
      IF(NXYZ(K1,JH1,I1).NE.0) GO TO 10
      J1=JH1
      GO TO 40

10    CONTINUE
      IF(I1.GE.KKC) GO TO 20
      I1=I1+1
      J1=1
      IF(NXYZ(K1,1,I1).NE.0) J1=2
      IF(NXYZ(K1,J1,I1).NE.0) GO TO 10
      GO TO 40

20    CONTINUE
      IF(N1.EQ.1.AND.K1.GE.KCH) GO TO 30
      IF(N1.EQ.(-1).AND.K1.LE.KCH) GO TO 30
      K1=K1+N1
      I11=I1
      I1=1
      IF(I11.GT.KKB)I1=KKB+1
      J1=1
      IF(NXYZ(K1,1,I1).NE.0)J1=2
      IF(NXYZ(K1,J1,I1).NE.0) GO TO 1
      GO TO 40

30    CONTINUE
      CASE1=CASE
      IF(CASE.LE.2) THEN
      CASE=5
      IF(N22.EQ.1) GO TO 34
      N33=1
      IH=IH-1
      GO TO 100
      ELSE
      CASE=0
      IF(N33.EQ.1) GO TO 34
      N22=1
      IH=IH-1
      GO TO 100
      ENDIF

34    CONTINUE
      IF( N22.EQ.1.AND.N33.EQ.1) THEN
      IF(I1.EQ.KKB) GO TO 31
      IF(I1.EQ.KKT) GO TO 32
      ENDIF

31    CONTINUE
      IBLF=KKB+1
      IBLB=KKB+1
      KBLB=10
      KBLF=11
      JBLF=0
      JBLB=0
      GO TO 100
```

```
32  WRITE(6,*)' VESSEL IS FULL, NO MORE CONTAINERS CAN BE LOADED'
    RETURN
40  CONTINUE
    KK=K1
    J=J1
    II=I1
    IF(CASE.GE.3) THEN
    IBLF=I1
    JBLF=J1
    KBLF=K1
    X2=X2+1
    ELSE
    IBLB=I1
    JBLB=J1
    KBLB=K1
    X1=X1+1
    ENDIF

    RETURN
    END
```

```
C*****************************************************************
      SUBROUTINE ORDER(KK,J,FACTOR)
C*****************************************************************

C     This subroutine calculates the increase in GM and the incurred
C     cost if the containers of the column (KK,J) are placed in
C     decreasing order of weight (the heavier in lower rows). Also
C     the columns are ranked with respect to benefit/cost ratio.

      COMMON/SHIP1/D,XMOMNT,NT,YL(10),CTX(10),CTZ(10),Y(10),NF
      COMMON/SHIP2/IIT,JJT,KKB,KKA,KKT,MC1,MC2
      COMMON/SHIP3/CX(30),CZ(15),NXYZ(30,20,15),WXYZ(30,20,15)
      COMMON/SHIP4/XLB2,XLB3,GM2,GM3,XKG2,XKG3,XKM2,XKM3,GMIN
      COMMON/LOAD1/W(1500),NS(20)
      COMMON/LOAD4/II1,II2,JJ1B,JJ1F,JJ2B,JJ2F,KK1B,KK1F,
     *KK2B,KK2F,IH,IL,NP
      COMMON/LOAD5/IPOINT,CASE,XX1,XX2
      COMMON/LOAD6/BC1(30,20),CC1(30,20),RC1(30,20),IO1(600,2)
     *,BC2(30,20),CC2(30,20),RC2(30,20),IO2(600,2)
      COMMON/REAR1/II,KKLL,JLL
      COMMON/NORMA1/N22,N33,N222,N333
      COMMON/ORDER1/WC
C
      F=2
      IF(J.EQ.1.AND.NF.EQ.1) F=1
      IF(II.GT.KKB)GO TO 1000

C CALCULATION OF BBi & CCi CEFFICIENTS FOR STACKS BELOW THE DECK

C Calculation of BBi coefficient

      DO 100 K=1,II
      IF(NXYZ(KK,J,K).LT.0)GO TO 100
      IF(NXYZ(KK,J,K).EQ.0)WRITE(6,*)' WARNING: SOMETHING IS
     1 WRONG WITH THE ALLOCATION OF TE CONTAINERS'
      IF(WC.LE.WXYZ(KK,J,K))GO TO 100
      BC1(KK,J)=BC1(KK,J)+(WC-WXYZ(KK,J,K))*F*FACTOR
 100  CONTINUE
      IM=1

C Calculation of CCi coefficient

      CF1=0
      IY=0
      DO 200 L=1,II
      IF(NXYZ(KK,J,L).LE.0) GO TO 200
      IF(IY.EQ.1) GO TO 211
      IF(L.EQ.II) GO TO 212

      L1=L+1
      DO 210 LL=L1,II
      IF(WXYZ(KK,J,L).GE.WXYZ(KK,J,LL)) GO TO 210
      IY=1
      CF1=F*(II-L+1)
      GO TO 211
 210  CONTINUE
      GO TO 200
```

```
  211  CONTINUE
       IF(NXYZ(KK,J,L).EQ.NP) CF1=CF1-F
  200  CONTINUE

  212  CC1(KK,J)=CF1

       IF(CC1(KK,J).EQ.0) GO TO 201
       R=BC1(KK,J)/CC1(KK,J)
       GO TO 202
  201  R=0
  202  RC1(KK,J)=R

C       WRITE(6,2222)KK,J,BC1(KK,J),CC1(KK,J),R
C2222   FORMAT(' BC1(',I2,',',I2,')=',F8.2,' CC1=',F8.2,' R=',F8.2)

       NDIS=0
       DO 300 I=1,MC1
       K1=IO1(I,1)
       J1=IO1(I,2)

C       WRITE(6,2221)K1,J1,RC1(K1,J1)
C2221   FORMAT(' RC1(',I2,',',I2,')=',F8.2)

       IF(RC1( K1,J1).EQ.0.0) THEN
       IF(NDIS.EQ.0) GO TO 302
       I111=I-1
       IO1(I111,1)=KK
       IO1(I111,2)=J
       GO TO 311
       ENDIF
       IF(NDIS.EQ.1) THEN
       IF(R.GT.RC1(K1,J1)) THEN
       I111=I-1
       GO TO 312
       ENDIF
       I111=I-1
       IO1(I111,1)=IO1(I,1)
       IO1(I111,2)=IO1(I,2)
       ENDIF
       IF(KK.EQ.K1.AND.J.EQ.J1)THEN
       NDIS=1
       GO TO 301
       ENDIF
       IF(R.GT.RC1(K1,J1)) THEN
       GO TO 302
       ELSEIF(R.EQ.RC1(K1,J1))THEN
       IF(BC1(KK,J).GE.BC1(K1,J1)) GO TO 302
       ENDIF
  301  CONTINUE
  300  CONTINUE
       GO TO 310
  302  NIN=0
       IF(NDIS.EQ.1)GO TO 310
       MA=MC1-I
       IF(MA.EQ.0) GO TO 310
       DO 303 L=1,MA
       M1=MC1+1-L
```

```
      IF((IO1(M1,1).EQ.KK.AND.IO1(M1,2).EQ.J).OR.
     *NIN.EQ.1) THEN
      M11=M1-1
      IO1(M1,1)=IO1(M11,1)
      IO1(M1,2)=IO1(M11,2)
      NIN=1
      ENDIF
  303 CONTINUE
  310 CONTINUE
      IO1(I,1)=KK
      IO1(I,2)=J
      GO TO 311
  312 IO1(I111,1)=KK
      IO1(I111,2)=J
  311 F=2
      RETURN


 1000 CONTINUE

C CALCULATION OF THE BBi & CCi FOR THE CONTAINERS PLACED ON DECK

C Calculation of BBi coefficient

      IN1=KKB+1
      DO 400 K=IN1,II
      IF(WC.LE.WXYZ(KK,J,K))GO TO 400
      BC2(KK,J)=BC2(KK,J)+(WC-WXYZ(KK,J,K))*F*FACTOR
  400 CONTINUE

C Calculation of CCi coefficient

      CF1=0
      IY=0
      IK1=KKB+1
      DO 500 L=IK1,II
      IF(NXYZ(KK,J,L).LE.0) GO TO 500
      IF(IY.EQ.1) GO TO 511
      IF(L.EQ.II) GO TO 512

      L1=L+1
      DO 510 LL=L1,II
      IF(WXYZ(KK,J,L).GE.WXYZ(KK,J,LL)) GO TO 510
      IY=1
      CF1=F*(II-L+1)
      GO TO 511
  510 CONTINUE
      GO TO 500

  511 CONTINUE
      IF(NXYZ(KK,J,L).EQ.NP) CF1=CF1-F

  500 CONTINUE

  512 CC2(KK,J)=CF1
```

```
      IF(CC2(KK,J).EQ.0) GO TO 501
      R=BC2(KK,J)/CC2(KK,J)
      GO TO 502
 501  R=0
 502  RC2(KK,J)=R
      NDIS=0
      DO 600 I=1,MC2
      K1=IO2(I,1)
      J1=IO2(I,2)
      IF(BC2(KK,J).GE.BC2(K1,J1)) GO TO 602
      IF(NDIS.EQ.1) THEN
      IF(R.GT.RC1(K1,J1)) THEN
      I111=I-1
      GO TO 612
      ENDIF
      I111=I-1
      IO2(I111,1)=IO2(I,1)
      IO2(I111,2)=IO2(I,2)
      ENDIF
      IF(KK.EQ.K1.AND.J.EQ.J1)THEN
      NDIS=1
      GO TO 600
      ENDIF
      IF(R.GT.RC2(K1,J1)) THEN
      GO TO 602
      ELSEIF(R.EQ.RC2(K1,J1)) THEN
      GO TO 600
      ENDIF
 600  CONTINUE
      GO TO 610
 602  NIN=0
      IF(NDIS.EQ.0) GO TO 610
      MA=MC2-I
      IF(MA.EQ.0) GO TO 610
      DO 603 L=1,MA
      M1=MC2+1-L
      IF((IO2(I,1).EQ.KK.AND.IO2(I,2).EQ.J).OR.
     *NIN.EQ.1) THEN
      M11=M1-1
      IO2(M1,1)=IO2(M11,1)
      IO2(M1,2)=IO2(M11,2)
      NIN=1
      GO TO 603
      ENDIF
 603  CONTINUE
 610  CONTINUE
      IO2(I,1)=KK
      IO2(I,2)=J
      GO TO 611
 612  IO2(I111,1)=KK
      IO2(I111,2)=J
 611  RC2(KK,J)=R
      F=2
      RETURN
      END
```

```
C*********************************************************************
      SUBROUTINE WATBAL
C*********************************************************************

C     This subroutine makes zero (brings within given limits) the
C     longitudinal moment of the vessel by using water ballast.
C     The use of water ballast is kept to a minimum. Firstly, the
C     subroutine tries to ballance the longitudinal moment by sub-
C     tracting water ballast from the tanks this is possible. If
C     the moment is still unballanced ballast is placed in the
C     proper tanks. The tanks around LCB are filled first.

C     Note: The ertical coordinate of the center of weight of the
C     water in the tanks is calculated by using rough approximations.
C     An improved version of the program should take care of it.


      COMMON/SHIP1/D,XMOMNT,NT,YL(10),CTX(10),CTZ(10),Y(10),NF
      COMMON/SHIP4/XLB2,XLB3,GM2,GM3,XKG2,XKG3,XKM2,XKM3,GMIN
      COMMON/LOAD2/NB1,NB2,NA1,NA2,NB,NA
      COMMON/LOAD3/NTB,NTI,DICTIS,EYILON
      COMMON/WATBA1/IT1,IT2
C
      WRITE(6,534)XMOMNT
534   FORMAT('  MOMENT=',F15.2)
      NTI=0
      DIS=D
 100  CONTINUE
      DIS1=DIS
      IF(XMOMNT.LT.0.0) THEN
      JJ=1
      KK=NTB
      ELSEIF(XMOMNT.GT.0.0) THEN
      JJ=NTB+1
      KK=NT
      ELSE
      GO TO 1002
      ENDIF
  10  CONTINUE
      J=JJ
      K=KK
  11  DO 1 I=J,K
      II=I
      IF(XMOMNT.GT.0.0)II=K-I+J
      IF(Y(II).GT.0.0) GO TO 2
   1  CONTINUE
      GO TO 3
   2  DY=ABS(XMOMNT/(CTX(II)-XLB3))
      IF(DY.LE.Y(II)) GO TO 20
      XMOMNT=XMOMNT-Y(II)*(CTX(II)-XLB3)

      DIS1=DIS
      DIS=DIS-Y(II)
      XKG2=XKG3
      XKG3=(XKG2*DIS1-(Y(II)*Y(II)/YL(II)*CTZ(II)))/DIS
```

```
      Y(II)=0
      IF(XMOMNT.LT.0.0)J=I+1
      IF(XMOMNT.GT.0.0)K=II-1
      GO TO 11
   20 CONTINUE
      XMOMNT=XMOMNT-DY*(CTX(II)-XLB3)
      Y(II)=Y(II)-DY
   21 DIS1=DIS
      DIS=DIS-DY
      XKG2=XKG3
      XKG3=(XKG2*DIS1-((Y(II)+DY)*(Y(II)+DY)-Y(II)*Y(II))*CTZ(II)
     1/YL(II))/DIS

      GO TO 40

    3 CONTINUE
      IF(XMOMNT.LT.0.0) THEN
      JJ=NTB+1
      KK=NT
      ELSEIF(XMOMNT.GT.0.0) THEN
      JJ=1
      KK=NTB
      ELSE
      GO TO 1002
      ENDIF
      J=JJ
      K=KK
   32 DO 5 L=J,K
      LL=L
      IF(XMOMNT.GT.0.0)LL=K-L+J
      IF(Y(LL).LT.YL(LL)) GO TO 31
    5 CONTINUE
      GO TO 1000
   31 CONTINUE
      DY=ABS(XMOMNT/(CTX(LL)-XLB3))
      IF(DY.LE.(YL(LL)-Y(LL)))GO TO 30

      DIS1=DIS
      DIS=DIS+YL(LL)-Y(LL)
      XKG2=XKG3
      XKG3=(XKG2*DIS1+(YL(LL)*CTZ(LL)-Y(LL)*Y(LL)*CTZ(LL)/YL(LL)))/DIS

      XMOMNT=XMOMNT+(YL(LL)-Y(LL))*(CTX(LL)-XLB3)
      Y(LL)=YL(LL)
      IF(XMOMNT.LT.0.0)J=L+1
      IF(XMOMNT.GT.0.0)K=LL-1
      GO TO 32
   30 XMOMNT=XMOMNT+DY*(CTX(LL)-XLB3)

      Y(LL)=DY+Y(LL)
   33 DIS1=DIS
      DIS=DIS+DY
      XKG2=XKG3
      XKG3=(XKG2*DIS1+(Y(LL)*Y(LL)/YL(LL)-(Y(LL)-DY)*(Y(LL)-DY)/YL(LL))*
     1CTZ(LL))/DIS
```

```
   40  XLB2=XLB3
       XLB3=XLCB(DIS)
       XMOMNT=DIS*(XLB3-XLR2)
       IF(ABS(XMOMNT).GT.EYILON) GO TO 100
 1002  D=DIS
       WRITE(6,534)XMOMNT
       IT1=0
       IT2=0

       DO 50 I=1,NTB
       IF(Y(I).GT.0) THEN
       IT1=1
       GO TO 51
       ENDIF
   50  CONTINUE
   51  NTB1=NTB+1
       DO 52 I=NTP1,NT
       IF(Y(I).GT.0) THEN
       IT2=1
       GO TO 53
       ENDIF
   52  CONTINUE

   53  GM3=XKM(D)-XKG3

       RETURN

 1000  WRITE(6,1001)
 1001  FORMAT(' WATER  BALLAST CANNOT BE USED ANY MORE')
       D=DIS
       NTI=1
       WRITE(6,534)XMOMNT

       GM3=XKM(D)-XKG3

       RETURN
       END
```

```
C*********************************************************************
      SUBROUTINE GMCOR(IO,BC,CC,RC,LP)
C*********************************************************************

C     This subroutine applies interchanges among the containers of the
C     necessary number of columns in order to satisfy the GM-constraint.
C     It starts with the column with the highest benefit/cost ratio. It
C     proceeds until the constraint is satisfied; otherwise an infeasi-
C     bility message appears.

      COMMON/SHIP1/D,XMOMNT,NT,YL(10),CTX(10),CTZ(10),Y(10),NF
      COMMON/SHIP2/IIT,JJT,KKB,KKA,KKT,MC1,MC2
      COMMON/SHIP4/XLB2,XLB3,GM2,GM,XKG2,XKG3,XKM2,XKM3,GMIN
      COMMON/SHIP5/H
      COMMON/LOAD3/NTB,NTI,DICTIS,EYILON
      COMMON/LOAD4/II1,II2,JJ1B,JJ1F,JJ2B,JJ2F,KK1B,KK1F,
     *KK2B,KK2F,IH,IL,NP
      COMMON/OUTPU1/ISTAT(30)
      DIMENSION BC(30,20),CC(30,20),RC(30,20),IO(600,2)
C
      WRITE(6,*)' ***** SUBROUTINE GMCOR *****'
C
      IF(GM.GE.GMIN)RETURN
      DGM=GMIN-GM
      DKGD=DGM*D
      XNBC=DKGD/H
  100 K=IO(1,1)
      J=IO(1,2)
      IF(BC(K,J).EQ.0) GO TO 10
      BCK=BC(K,J)
      ccx=cc(k,j)
      ISTAT(K)=ISTAT(K)+NINT(CCX)

      BC(K,J)=0.0
      RC(K,J)=0
      IF(LP.EQ.1)MM=MC1
      IF(LP.EQ.2)MM=MC2
      M1=MM-1
      DO 1 I=1,M1
      II=I+1
      IO(I,1)=IO(II,1)
      IO(I,2)=IO(II,2)
    1 CONTINUE
      IO(MM,1)=K
      IO(MM,2)=J
      KY=0
      IF(XNBC.LT.BCK) KY=1
      DKGD=DKGD-BCK*H
      XNBC=XNBC-BCK
      GM=GM+BCK*H/D
      XKG3=XKM(D)-GM
      CALL REAR(K,J,LP)
      IF(KY.EQ.1)GO TO 20
      GO TO 100
   10 CONTINUE
      IF(LP.EQ.2)GO TO 11
   11 DICTIS=1
```

```
      DO 200 M=1,22
      CALL TRANAR(M)
      IF(GM.GE.GMin) then
      dictis=0
      GO TO 20
      ENDIF
200   CONTINUE

20    CONTINUE


      RETURN
      END
```

```
C*********************************************************************
      SUBROUTINE REAR(K,J,LP)
C*********************************************************************

C     This subroutine interchanges the containers of column (K,J)
C     rearranging them in decreasing order of weight.

      COMMON/SHIP2/IIT,JJT,KKB,KKA,KKT,MC1,MC2
      COMMON/SHIP3/CX(30),CZ(15),NXYZ(30,20,15),WXYZ(30,20,15)
      COMMON/REAR1/II,KKL,JL
C
       write(6,*)' ***** subroutine rear *****'
      IF(LP.EQ.1) THEN
      M1=1
      M2=KKB
      ELSEIF(LP.EQ.2) THEN
      M1=KKB+1
      M2=KKT
      ENDIF
      DO 2 L=M1,M2
      DO 2 M=M1,M2
      IF(NXYZ(K,J,M).LE.0) GO TO 2
      MM=M+1
      IF(WXYZ(K,J,M).LT.WXYZ(K,J,MM)) THEN
      H=WXYZ(K,J,MM)
      WXYZ(K,J,MM)=WXYZ(K,J,M)
      WXYZ(K,J,M)=H
      NHI=NXYZ(K,J,MM)
      NXYZ(K,J,MM)=NXYZ(K,J,M)
      NXYZ(K,J,M)=NHI
      ENDIF
    2 CONTINUE
      RETURN
      END
```

```
c**********************************************************************
      subroutine tranar(k)
c**********************************************************************

c
c     This subroutine performs container interchanges among the
c     containers of station k. It calculates the improvement in
c     GM, but it does not calculate the change in the container
c     handling costs.
c

      COMMON/SHIP1/D,XMOMNT,NT,YL(10),CTX(10),CTZ(10),Y(10),NF
      COMMON/SHIP2/IIT,JJT,KKB,KKA,KKT,MC1,MC2
      COMMON/SHIP3/CX(30),CZ(15),NXYZ(30,20,15),WXYZ(30,20,15)
      COMMON/SHIP4/XLB2,XLB3,GM2,GM,XKG2,XKG3,XKM2,XKM3,GMIN
      COMMON/SHIP5/H
      COMMON/OUTPU1/ISTAT(30)

      WRITE(6,*) ' *** subroutine tranar *** '

      IA=1
      IBB=KKB
      FACTOR=0

      JNF=1
      IF(NF.EQ.1)JNF=2

100   DO 101 IB1=IA,IBB

      IB=IBB-IB1+IA

      DO 11 JK=JNF,JJT
      DO 12 I=IA,IB
      II=IB-I+IA
      IF(NXYZ(K,JK,II).LE.0) GO TO 12
      GO TO 13
 12   CONTINUE
 13   CONTINUE

      DO 14 JL=JNF,JJT
      IF(JL.EQ.JK) GO TO 14
      IN=II
 15   IN=IN-1
      IF(IN.EQ.(IA-1)) GO TO 14
      IF(NXYZ(K,JL,IN).LE.0) GO TO 15
      IF(WXYZ(K,JK,II).GT.WXYZ(K,JL,IN)) THEN
      WC=WXYZ(K,JK,II)
      WXYZ(K,JK,II)=WXYZ(K,JL,IN)
      WXYZ(K,JL,IN)=WC

      GM=GM+2*(WXYZ(K,JL,IN)-WXYZ(K,JK,II))*ABS(IN-II)*H/D
      XKG3=XKM(D)-GM
```

```
      NC=NXYZ(K,JK,II)
      NXYZ(K,JK,II)=NXYZ(K,JL,IN)
      NXYZ(K,JL,IN)=NC
      ENDIF

      DO 16 IN1=IA,IN-1
      IN2=IN-IN1
      IN0=IN2+1
      IF(NXYZ(K,JL,IN2).LE.0) GO TO 16
      IF(WXYZ(K,JL,IN0).GT.WXYZ(K,JL,IN2)) THEN
      WC=WXYZ(K,JL,IN0)
      WXYZ(K,JL,IN0)=WXYZ(K,JL,IN2)
      WXYZ(K,JL,IN2)=WC

      GM=GM+2*(WXYZ(K,JL,IN2)-WXYZ(K,JL,IN0))*ABS(IN0-IN2)*H/D
      XKG3=XKM(D)-GM

      NC=NXYZ(K,JL,IN0)
      NXYZ(K,JL,IN0)=NXYZ(K,JL,IN2)
      NXYZ(K,JL,IN2)=NC

      ISTAT(K)=ISTAT(K)+F

      GO TO 16
      ENDIF
      GO TO 14
   16 CONTINUE
   14 CONTINUE
   11 CONTINUE
  101 CONTINUE

      IF(FACTOR.EQ.0) THEN
      FACTOR=1
      IA=KKB+1
      IBB=KKT
      GO TO 100
      ENDIF

      RETURN
      END
```

```
**********************************************************************
      subroutine OUTPUT
**********************************************************************

C     This subroutine prepares the output of the program.

      COMMON/SHIP1/D,XMOMNT,NT,YL(10),CTX(10),CTZ(10),Y(10),NF
      COMMON/SHIP2/KK,JJT,KKB,KKA,KKT,MC1,MC2
      COMMON/SHIP3/CX(30),CZ(15),NXYZ(30,20,15),WXYZ(30,20,15)
      COMMON/SHIP4/XLB2,XLB3,GM2,GM3,XKG2,XKG3,XKM2,XKM3,GMIN
      COMMON/LOAD4/II1,II2,JJ1B,JJ1F,JJ2B,JJ2F,KK1B,KK1F,
     *KK2B,KK2F,IH,IL,NP
      COMMON/LOAD7/NPOR1(30,20),NPOR2(30,20)
      COMMON/BLOCK1/ICHOIC,IBAL
      COMMON/OUTPU1/ISTAT(30)
      COMMON/OUTPU2/PTR(20),PDC(20),PGM(20),DP(20),NC(20)
      DIMENSION NUM(30,20)

      DO 1 I=1,30
      DO 1 J=1,20
      NUM(I,J)=0
   1  CONTINUE

      WRITE(7,1114)NP
1114  FORMAT(//,'  Number of ports visited until now :',I2)

      IF(ICHOIC.EQ.1) THEN
      WRITE(7,1120)
1120  FORMAT(//,' Blocking container procedure has been selected')
      WRITE(7,*)' Vessel is filled station by station'
      ELSE
      WRITE(7,1121)
1121  FORMAT(/,' Vessel is filled row by row')
      ENDIF

      IF(IBAL.EQ.1)WRITE(7,11
1122  FORMAT(//,'The loading starts with zero initial moment',//)

      WRITE(7,*)
      WRITE(7,*)
      WRITE(7,*)
      WRITE(7,*)' PREVIOUS PORTS'

      NPP=NP-1
      WRITE(7,2223)
      DO 200 MI=1,NPP
      DR=PDC(MI)
      TRIM=PTR(MI)/XMTC1(DR)
      WRITE(7,2222)MI,NC(MI),DP(MI),PDC(MI),TRIM,PGM(MI)
2223  FORMAT(' Port   cnts loaded   conts" weight  Displacement
     *    Trim       GM')
2222  FORMAT(2X,I2,7X,I4,8X,F7.1,7X,F7.1,7X,F6.2,7X,F5.2)
 200  CONTINUE

      WRITE(7,*)
      WRITE(7,*)
      WRITE(7,*)
      WRITE(7,*)'  LAST PORT'
```

```
      WRITE(7,1115)D
1115  FORMAT(/,'  Displacement =',F10.2,' tons')

      WRITE(7,1116)GM3,XKG3,GMIN
1116  FORMAT('                    GM =',F10.2,' feet',/,
     *       '                    KG =',f10.2,' feet',/,
     *       ' Minimum required GM =',F10.2,' feet')

      TRIM=XMOMNT/XMTC1(D)
      XLBP=680.
      ANGLE=ATAN2(TRIM,XLBP)
      WRITE(7,1117)TRIM,ANGLE
1117  FORMAT(' TRIM=',F8.3,' feet    ANGLE=',F8.3,' degr.')

      WRITE(7,*)'                                                    '
      WRITE(7,*)' No of tank     Water contained    Capacity  '
      WRITE(7,*)'    '

      DO 40 JH=1,NT
      WRITE(7,1123)JH,Y(JH),YL(JH)
1123  FORMAT(2X,I5,9X,F10.2,9X,F10.3)
  40  CONTINUE

      COST=0
      DO 30 K=1,KK
      COST=COST+ISTAT(K)
  30  CONTINUE
      WRITE(7,1118)COST
1118  FORMAT(//,'   TOTAL OVERSTOWAGE COST =',F6.1,' units')
      DO 100 K=1,KK

      DO 10 I=1,KKT
      DO 10 J=1,JJT
      IF(J.EQ.1.AND.NF.EQ.1)IW=1
      IF(J.GT.1.OR.NF.NE.1)IW=2
      IF(NXYZ(K,J,I).LE.0) GO TO 10
      N=NXYZ(K,J,I)
      NUM(K,N)=NUM(K,N)+IW
  10  CONTINUE

 100  CONTINUE

      WRITE(7,1111)
1111  FORMAT(//,' STATION:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
     * 16 17 18 19 20 21 22')
      DO 20 L=1,NP
      WRITE(7,1112)L,(NUM(KL,L),KL=1,22)
1112  FORMAT(' port ',I2,':',22I3)
  20  CONTINUE

      WRITE(7,1113)(ISTAT(K),K=1,22)
1113  FORMAT(/,' ov-cost:',22I3)

      RETURN
      END
```

```
C*********************************************************************
      REAL FUNCTION XKM(D)
      IF(D.GE.20000)THEN
      XKM=45
      ELSE
      XKM=45+(20000-D)/15000*47.5
      ENDIF
      RETURN
      END
C*********************************************************************
      REAL FUNCTION XMTC1(D)
      XMTC1=((D-10000)/60000*8.25+4.55)*500
      RETURN
      END
C*********************************************************************
      REAL FUNCTION YLCB(D)
      IF(D.GE.50000)THEN
      XLCB=685/2-18-((D-50000)/20000*5)
      ELSE
      XLCB=685/2-1-(D-10000)/40000*17
      ENDIF
      RETURN
      END
C*********************************************************************
      REAL FUNCTION SIG(X)
      IF(X.LT.0)SIG=-1
      IF(X.GT.0)SIG=1
      IF(X.EQ.0)SIG=0
      RETURN
      END
```