A Column Generation Algorithm for a Ship Scheduling Problem

LEIF H. APPELGREN

*The Royal Institute of Technology, Stockholm, Sweden**

This paper describes an algorithm for a ship scheduling problem, obtained from a Swedish shipowning company. The algorithm uses the Dantzig-Wolfe decomposition method for linear programming. The subprograms are simple network flow problems that are solved by dynamic programming. The master program in the decomposition algorithm is an LP problem with only zero-one elements in the matrix and the right-hand side. Integer solutions are not guaranteed, but generation and solution of a large number of problems indicates that the frequency of fractional solutions is as small as 1–2 per cent. Problems with about 40 ships and 50 cargoes are solved in about 2.5 minutes on an IBM 7090. In order to resolve the fractional cases, some integer programming experiments have been made. The results will be reported in a forthcoming paper.

A shipowning company is engaged in world-wide operations of a large number of ships. A set of cargoes is given for the planning period, which is 2–4 months. All the cargoes must have loading dates within the planning period, but they may be discharged after the period. Each cargo is characterized by

size
type
earliest loading date
latest loading date
loading port
discharging port
time for loading and discharging

* Now at Salen Shipping Companies, Stockholm, Sweden.

voyage costs
revenue for optional cargoes

The size of the cargoes is of the same magnitude as the size of the ships. Even if two small cargoes could be carried simultaneously on one large ship, this possibility is not considered in the model.

Loading and discharging of one cargo is sometimes performed in several ports, in which case the distance as well as the loading and discharging times must be modified accordingly. The ports where loading is started and where discharging is completed are then used as loading and discharging ports in the cargo data.

The voyage costs consist mainly of port and canal charges and loading and discharging costs. If these costs vary significantly between different ships, they should be individually specified for the ships that can carry a certain cargo.

Most of the cargoes are contracted and must be shipped by the company. From time to time new cargoes become available on the market, cargoes which can be contracted by the company if it is considered profitable. The revenue of such optional cargoes must be specified, whereas the revenue of contracted cargoes is irrelevant.

Each ship is characterized by

size
permitted types of cargo
initial open position
initial open date
speed
fuel consumption
'time value'
position costs

The meaning of 'time value' and position costs will be explained below. The daily running costs, including cost of capital, personnel, insurance etc., are not taken into account in the model since they are fixed during the planning period.

Beside the ship and cargo data, a distance matrix is given.

The scheduling problem is to assign a sequence of cargoes to each ship in the 'best possible' way. The objective is to maximize the revenue of optional cargoes minus the voyage and fuel costs, under the constraint that all contracted cargoes must be shipped.

The above formulation of an objective assumes a completely deterministic problem, whereas the real problem contains several stochastic elements. Delays in loading and discharging, bad weather, breakdowns and changes in customer plans will often make rescheduling necessary. In fact, one main reason for computerized scheduling is that new schedules have to be produced often and at very short notice. These stochastic elements can hardly be taken into account in the scheduling problem, except by adding some margin where delays seem probable. Another most important stochastic element is that new cargo opportunities arise after the completion of the schedule. In order to prepare for these opportunities, the scheduled cargoes should be concentrated to small ships with restricted cargo capabilities, leaving the most capable ships open. Even if all ships are equal, it is preferable to concentrate the cargoes to some ships, because it gives

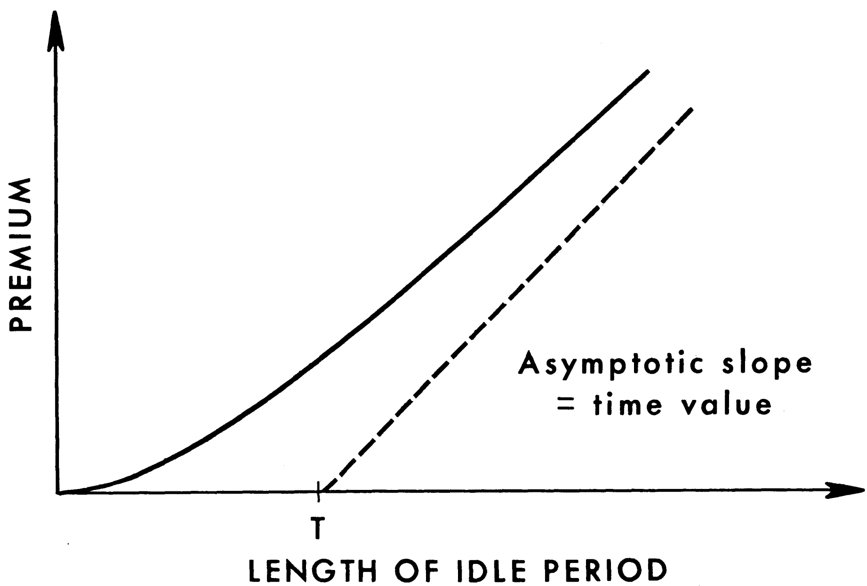

Fig. 1. Premium for idle time within the cargo sequence.

greater flexibility to have one ship entirely free during a time interval, compared to having one ship free during the first half of the interval and another one during the second.

The above reasoning led to the concept of 'time value', i.e., a daily premium for idle time after completion of the last cargo in the scheduled sequence for a ship. This premium makes it preferable to concentrate the idle time to some ships. The time value should vary between ships with size and cargo capability and reflect the expected daily revenue from unknown cargo opportunities. The time value should also vary with time, since the probability of new cargoes generally increases with time.

There is no reason to put such a premium on short periods of idle time within the cargo sequence, but it still seems preferable to concentrate such idle time to long periods for few ships. This can be accomplished by a premium that is a strictly convex function of the length of the idle period (Fig. 1). Then the premium will, for instance, be larger for one 15-day and one 5-day period than for two 10-days periods. The asymptotic slope of the function is equal to the time value for each ship. The breakpoint T can be varied in the algorithm.

Some of the cargoes will be discharged after the end of the planning period. In such cases the schedule uses resources from the next period, which are paid for at a price per day equal to the time value for each ship.

The revenue in the next period depends on the initial position, i.e., the final position of the ships in the current period. For this reason a 'position cost,' depending on ship and position, can be included in the objective function. This cost is set to the product of the expected daily revenue for the ship and the expected time for the first ballast trip in the next period. The position cost is difficult to estimate and has not been used in the problems that have been run.

The objective function thus consists of up to six elements:

1. Revenue from optional cargoes.
2. Premium for idle time at the end of the period.
3. Premium for idle time within the cargo sequence.
4. Deduction for use of ship capacity in the next period.
5. Position cost.
6. Voyage and fuel costs.

THREE LINEAR PROGRAMMING MODELS

THE PROBLEM is almost an ordinary assignment problem, the main difference being that one ship can carry more than one cargo during the period. This makes it impossible to use algorithms for transportation problems. The problem can be formulated as a linear programming problem in several ways, but these will generally not have integer solutions.

Model 1

This is a multicommodity flow model where the nodes have pairs of indices (k, n). k is the cargo index and n indicates one of the alternative loading dates. The flow variables are denoted $x(i, j, m, k, n)$. If $x(i, j, m, k, n) = 1$, ship i carried on the previous voyage cargo j on its mth loading date alternative and carries on the present voyage cargo k on its nth loading date alternative. If $j = 0$, cargo k is first in the cargo sequence for ship i, and if $k = 0$, cargo j is the last one. If both j and k are zero, ship i is idle throughout the period.

The flow variables are defined only for feasible combinations of i, j, m, k, and n, i.e., where the ship is large enough and permitted to carry cargoes j and k and where the time between the loading dates is large enough for loading, discharging, cargo trip, and ballast trip.

The coefficients in the objective function are $v(i, j, m, k, n)$. The revenue consists of three parts:

1. Revenue from cargo k if this cargo is optional.
2. Premium for idle time between cargoes j and k. The idle time is computed as the difference between the loading dates for node (k, n) and node (j, m) minus the time for loading, transportation and discharging of cargo j, and the time for the ballast trip.
3. Voyage and fuel costs. The voyage costs are collected only for the cargo trip (cargo k), while the fuel cost is proportional to the sum of the times for the ballast trip and the cargo trip.

If $k = 0$ the revenue consists of two other parts:

1. Premium for idle time after cargo j or deduction for use of time in the next period. The date when the ship becomes available after cargo j is computed as the loading date for node (j, m) plus the time for loading, transportation, and discharging of cargo j.
2. Position cost. This cost element is taken directly from the input data and is a function of the ship index i and the discharging port for cargo j.

Thus the mathematical formulation of the problem is

$$\text{Maximize } \sum_i \sum_j \sum_m \sum_k \sum_n v(i, j, m, k, n)\, x(i, j, m, k, n),$$

subject to the following constraints:

Flow constraints

$$\sum_k \sum_n x(i, j, m, k, n) = \sum_k \sum_n x(i, k, n, j, m) \text{ for all } i, j, m.$$

Ship constraints

$$\sum_k \sum_n x(i, 0, 0, k, n) = 1 \quad \text{for all } i.$$

Cargo constraints

$$\sum_i \sum_j \sum_m \sum_n x(i, j, m, k, n) \begin{cases} \leq 1 \\ = 1 \end{cases} \quad \text{for all } k.$$

(The equality sign holds for contracted cargoes.)

In addition to these constraints, all variables must be nonnegative. The variables should also be integer, which means that this is an integer linear programming model. If the problem is solved by ordinary LP methods, integer solutions cannot be guaranteed.

In a feasible solution with integer-valued variables, the above constraints have the following interpretation. The flow constraints imply that there must be one variable equal to one for ship i with (j, m) as the previous cargo if there is one variable with (j, m) as the present cargo, i.e., if ship i

carries cargo j on the mth loading date alternative. The ship constraints imply that each ship either is idle throughout the period $[x(i, 0, 0, 0, 0) = 1]$ or starts its cargo sequence with one of the cargoes on one of the loading date alternatives. The cargo constraints guarantee that exactly one ship will be assigned to each contracted cargo and one ship or no ship to each optional cargo.

A realistic problem, with 40 ships, 50 cargoes, and a 60-day period would have about 4000 variables and 1000 constraints. If the period is extended such that the transit times become negligible compared to the period length, the number of variables will increase tenfold and the number of equations will be doubled. The large increase in the number of variables is due to the fact that most combinations (j, k) are prohibited if the period is short, because the separation in time is too small.

Model 2

The large number of flow constraints can be eliminated by listing for each ship all the extremal points of the sets defined by the flow, ship, and non-negativity constraints. These extremal points are equivalent to the feasible cargo sequences for each ship. Hence, in this model we enumerate all feasible sequences and compute their value. Then we can find the best convex combination of sequences for each ship, satisfying the cargo constraints, by means of a linear program.

Assume that there are $N(i)$ feasible sequences for ship i. Put $a(i, j, k) = 1$ if cargo k is in the jth sequence for ship i, otherwise $a(i, j, k) = 0$. The value of this sequence is $v(i, j)$. We obtain the following linear program:

$$\text{maximize } \textstyle\sum_i \sum_{j=1}^{N(i)} v(i, j)\, x(i, j) \text{ subject to}$$

Convexity constraints

$$\textstyle\sum_{j=1}^{j=N(i)} x(i, j) = 1 \quad \text{for all } i.$$

Cargo constraints

$$\textstyle\sum_i \sum_{j=1}^{j=N(i)} x(i, j) a(i, j, k) \begin{cases} \leq 1 \\ = 1 \end{cases} \text{for all } k.$$

Even in this model the variables must be nonnegative.

For the 60-day example mentioned above, the number of variables would be approximately the same in model 1 and model 2 because sequences with more than two cargoes occur very rarely in such a short period and all sequences with up to two cargoes correspond to variables in model 1. If the period is extended, however, the number of sequences will increase up to the order of 10^{13}, which obviously limits the use of this model. The number of constraints is only $40 + 50$, one for each ship and one for each cargo.

Dantzig-Wolfe Decomposition

Below is a short description of Dantzig-Wolfe decomposition. For more information and proofs, *see* DANTZIG.[1]

Given the following LP problem with x_i as vector variables.

$$\text{Maximize } \textstyle\sum_{i=1}^{i=n} v_i x_i,$$

$$\text{subject to } \textstyle\sum_{i=1}^{i=n} A_i x_i = b, \qquad \text{(Common constraints)}$$

$$\text{and } x_i \in S_i \quad \text{for} \quad 1 \leqq i \leqq n, \qquad \text{(Individual constraints)}$$

where the sets S_i are convex.

The algorithm alternates between generation of extreme points (or extreme rays) of the sets S_i and determination of the best convex combination of the generated extreme points. The latter task is accomplished by the *restricted master program*, which is a linear program in the variables λ_i^k, which are the weights for the previously generated extreme points x_i^k.

$$\text{Maximize } \textstyle\sum_{i=1}^{i=n} \sum_{k=1}^{k=K_i} (v_i x_i^k)\lambda_i^k,$$

$$\text{subject to } \textstyle\sum_{i=1}^{i=n} \sum_{k=1}^{k=K_i} (A_i x_i^k)\lambda_i^k = b,$$

$$\text{and } \textstyle\sum_{i=1}^{i=K_i} \lambda_i^k = 1 \qquad \text{for} \quad 1 \leqq i \leqq n \qquad \text{(convexity constraints)}$$

$$\text{and } \lambda_i^k \geqq 0.$$

The vector of optimal dual variables for the common constraints is denoted p and the dual variables for the convexity constraints are denoted $q(i)$. If all the extremal points of the sets S_i were included in the master program, the optimal solution to the master program would be optimal even in the original problem and the optimality criterion $v_i x_i^k - p^T A_i x_i^k - q(i)\, I \leqq 0$ would be satisfied for all i and k. As this is not the normal case, this criterion will not necessarily be satisfied for the extreme points outside the master program. In order to check this $(v_i - p^T A_i)x_i$ is maximized over the set S_i for all i. If any of these maxima V_i exceeds $q(i)$, the corresponding extreme point is introduced into the master program, after which the procedure is repeated. Computation time might be saved if the solution of a subprogram is terminated when the value V_i exceeds $q(i)$, even if this solution is not optimal.

There is some freedom in the Dantzig-Wolfe method concerning the generation and deletion of extreme points, i.e., columns in the master program. The minimum number of columns that must be saved from the previous solution is the basis matrix while on the other extreme all previously generated columns can be saved. Probably the most common method is to assign space for a certain number of columns and, after this space has been filled, to delete one nonbasic column each time a new column is entered.

For generation of new columns, one of the following strategies can be used:

1. Solve one subprogram at a time until a solution with $V(0) > q(i)$ is obtained. This solution is entered into the master program.
2. Solve all the subprograms and enter the solution for which $V(0) - q(i)$ is maximum.
3. Enter the solutions from all the subprograms where $V(0) > q(i)$.

Model 3

We can decompose model 1 by the Dantzig-Wolfe method into a master program with the cargo constraints and a subprogram for each ship with the flow, ship, and nonnegativity constraints.

The objective function for ship i becomes:

$$\sum_j \sum_m \sum_k \sum_n [v(i, j, m, k, n) - p(k)]x(i, j, m, k, n) - q(i),$$

where $p(k)$ and $q(i)$ are the multipliers for the kth cargo constraint and the ith convexity constraint in the master program, respectively. The subprograms are ordinary shortest route problems in a noncyclic network formed by the nodes marked (k, n) and one initial and one terminal node. This problem is easily solved by the following dynamic programming algorithm. The nodes are ordered according to loading date and numbered with an index $s = 1, \cdots, S - 1$. Two tables, $k(s)$ and $n(s)$, give the correspondence between the old and new indices. The initial and terminal nodes are given the indices $s = 0$ and $s = S$, respectively. $V(S)$, $k(0)$, and $k(S)$ are set to zero. For each s starting with the maximum index, we compute

$$V(s) = \max_{s'>s} \{V(s') + v[i, k(s), n(s), k(s'), n(s')] - p[k(s')]\}.$$

Finally we obtain $V(0)$ from which we subtract $q(i)$ to obtain the value of the subprogram. The optimal choices of s', which we denote $s^*(s)$, are stored during the computations. If $V(0) - q(i) > 0$, the solution should be entered into the master program. In such cases, the optimal cargo sequence is obtained from the $s^*(s)$ table. The value of the solution is obtained by subtracting from $V(0)$ the multipliers $p(k)$ for the cargoes in the sequence.

The master program is exactly the same problem as in model 2, although the matrix contains only a small fraction of the extremal points of the subprograms.

THE ALGORITHM

An algorithm has been developed that uses the third LP model above, with Dantzig-Wolfe decomposition and dynamic programming solution of the subprograms. Space is provided in the master program for as many columns besides the basis as the number of ships, such that solutions from all subprograms with $V(0) > q(i)$ can be entered. The voyage and fuel costs are not taken into account in the present algorithm in order to simplify experimentation. The program is written in FORTRAN IV, using the LP subroutine LSUB (SHARE SDA 3384) for the master program. The program was originally run on an IBM 7090 at the Research Institute of Swedish National Defence in Stockholm, having 32 k core storage. As the LSUB subroutine uses the revised simplex method storing the problem matrix and the inverse explicitly, the 40-ship, 50-cargo problem requires $90\ (90 + 40 + 90) = 19800$ words for these matrices only. Apparently this is about the largest problem that can be solved in core on a 32 k computer using the present program. Solution time for this problem size averages about 2.5 minutes.

As a comparison, an experiment was carried out with the same example using model 2. The number of variables was reduced to about 1000 by imposing the arbitrary restriction that the idle time within a sequence was not allowed to exceed seven days. This problem was solved with the LP code M3 on the same computer in about 20 minutes, indicating considerable savings in computer time with the decomposition model.

TABLE I

Objective Row	40	15	0	0	25	15	0	RHS
Ship 1	1	1	1	1	0	0	0	1
Ship 2	0	0	0	0	1	1	1	1
Cargo 1	0	1	0	1	0	1	0	1
Cargo 2	0	0	1	1	0	0	1	1

The requirement for the shipowning company that sponsored this work is an algorithm capable of handling about 60 ships and 150 cargoes during a four-month period. No attempts have yet been made to solve such large problems, because some other difficulties should be resolved first, one which is the occurrence of fractional solutions.

FRACTIONAL SOLUTIONS

THE MASTER program in model 3 and the LP problem in model 2 are essentially matrices from an assignment problem, although more than one cargo may be assigned to one ship. This might lead to fractional solutions, which is illustrated by the following example:

The example consists of two ships, where Ship 1 is faster than Ship 2, and two contracted cargoes, Cargo 1 and Cargo 2. These cargoes are separated in time such that Ship 1 can carry both, while Ship 2 can only carry one of them. The possible columns in the LP problem are given in Table I.

This problem has three feasible integer solutions:

1. Ship 1 carries Cargo 1, Ship 2 carries Cargo 2. Value = 15.
2. Ship 1 carries Cargo 2, Ship 1 carries Cargo 1. Value = 15.
3. Ship 1 carries both cargoes. Value = 25.

The optimal solution of this LP problem is fractional, however. The value is 27.5 and the variable values are 0.5 in the first, fourth, sixth, and seventh column.

In Appendix I, a theorem is given stating that the LP model always has integer solutions under a specific set of assumptions. As these assumptions are quite restrictive, the theorem seems to be of little practical value.

In the runs with the original 40-ship, 50-cargo example, which was run 7 times with slightly different data, no fractional solutions occurred. In order to estimate the frequency of fractional solutions, random problems have been generated and solved by the algorithm. The results from these runs are given in Table II below. The input constants and the probability distributions for the randomly generated data are given in Appendix II.

TABLE II

Run	No of Ships	No. of Cargoes	Speed Range Knots	No. of Problems	No. of Fractional Solutions	Average Solution Time, sec
1	10	15	15–20	300	5	3.7
2	10	15	15–20	300	5	5.4
3	10	15	18	300	4	3.5
4	15	25	15–20	100	2	14.0

The realistic speed range is 15–20 knots. In run No. 3 all ships had the same speed in order to test whether different speeds has any influence on the frequency of fractional solutions. The results show no significant influence, however.

The difference between run No. 1 and run No. 2 is that the cargo revenues in run No. 2 were increased five times relative to the time values in order to resemble a situation with contracted cargoes. This means that more cargoes are carried on the average (an increase from 10.2 to 11.4 carried cargoes was observed), which might increase the frequency of fractional solutions. No such dependence was observed, however, but the average solution time increased by 46 per cent which indicates that problems with contracted cargoes are more complicated.

CONCLUSIONS

THE RESULTS show that the algorithm is very fast and that the frequency of fractional solutions is of the order of 1–2 per cent for problems with up to 15 ships and 25 cargoes. As considerably larger problems are of interest, the algorithm is being transferred to an IBM 360/75 where the experimentation will continue.

In order to obtain optimal integer solutions in the fractional cases, experiments have been made with two integer programming methods. The results are most promising and will be reported in a forthcoming paper.

The type of algorithm used here might be useful also for other scheduling, sequencing, or assignment problems. If the tasks in the problems (i.e., the cargoes in this specific problem) are free to move in time, the algorithm will give fractional solutions much more often because one task may occur several times in one sequence. One area where this is not the case, and where this algorithm is potentially useful, is the assignment of aircraft and airline crews to a flight schedule.

APPENDIX I

A Theorem on Integer Solutions

THEOREM. *If all ships have the same speed, time value, size and cargo capabilities, i.e., if the ships differ only with regard to open time and initial position, and if all cargoes have a fixed loading date, the linear programming algorithm gives an optimal integer solution.*

Proof. The problem can be regarded as a network flow problem for each ship as in the dynamic programming model of the subprograms above. Under the assumptions of the theorem, there is only one node for each cargo. The networks for the individual ships are identical with the exception of the arcs from the initial node to the cargo nodes. The existence of these arcs depend on the open time and initial position for each ship. These networks can be combined into one network with one source for each ship and one common sink. The cargo constraints are taken care of by giving the cargo nodes a maximum capacity of unity. If a cargo is contracted, this is also the minimum flow through that node. The objective function consists of income on all arcs (=idle time income) and on the optional cargo nodes. Apparently this is an ordinary capacitated minimum cost network problem that is known to possess an optimal integer solution.[2]

It is easily seen how this proof breaks down when the assumptions are weakened. If for instance the speeds are different or the loading dates flexible, some arcs might be prohibited for some ships. If the sizes or cargo capabilities are different, some nodes might be prohibited. If finally the time values are different, the income from the flow in one arc will depend on the origin of the flow, thus creating a multicommodity flow problem, which in general has fractional solutions.

APPENDIX II

Input Data for the Generation of Random Problems

THESE DATA were chosen in order to resemble the realistic 40-ship 50-cargo example as much as possible. The ports and the distance matrix were taken directly from that example, with 16 discharging ports and 13 loading ports.

Speed

In the example, the ships were classified into one of six speed classes between 15 and 20 knots. Only three ships belonged to the two slowest classes, and the majority belonged to the fourth and the fifth class. In the simulated problems, the speed class was selected randomly for each ship with equal probability. In some runs the speed range was reduced to zero (standard speed = 18 knots) in order to illustrate the effect of speed differences on the frequency of fractional solutions.

Ship Size

In the realistic example, the ship sizes varied between 170,000 and 411,000 cbft. The sizes were clustered in two groups, one between 170,000 and 280,000 cbft with about 75 per cent of the ships, and the other group between 350,000 and 411,000 cbft.

In the random problems the sizes were chosen according to a uniform distribution between 170,000 and 411,000 cbft.

Cargo Size

Same as above except that the maximum cargo size was 400,000 cbft.

Cargo Type

Ten cargo types with different ships requirements existed in the realistic example. One third of the cargoes were of one type, and the rest were quite uniformly distributed among the other types. In the random problems, equal probabilities were used for ten cargo types.

Ship Capability

In the realistic problem, some ships could carry only two types of cargo, while some ships could carry nine of the ten types. There was a certain correlation between the capabilities for different types, if for instance a ship could not carry type 7, it could not carry any of the types 1, 4, or 10.

In the random problems, the capability of a ship to carry a certain type of cargo was selected randomly with the probability 0.8, which was slightly larger than the average capability in the realistic problem.

Loading and Discharging Ports

The ports for each trip were selected randomly with equal probabilities among the set of loading and discharging ports, respectively. This is a considerable deviation from reality, since there is a strong correlation between loading port, discharging port, and cargo type. One reason for this simplification is to make the simulated problems less tied to the specific example.

Loading Date

In the realistic example, the loading dates were uniformly spread over the period, and therefore the loading dates in the simulated problems were selected according to a uniform distribution between 1 and 120.

TABLE III

Ship Data

Ship	Size 1000 cbft	Time Value	Open Date	Initial Position	Capability for Cargo Types 1	2	3	4	5	6	7	8	9	10
1	277	443	0	1	1	0	0	1	1	0	1	1	1	1
2	410	656	3	15	1	1	1	1	1	0	1	1	0	1
3	384	614	6	11	0	1	1	0	1	1	1	1	1	1
4	333	532	7	1	0	1	1	0	1	1	1	1	0	1
5	372	595	18	14	1	0	1	0	1	1	1	0	1	1
6	371	593	0	11	1	1	1	1	1	1	1	1	1	1
7	212	339	30	6	0	0	1	1	1	1	1	1	1	1
8	290	464	−5	13	1	1	0	1	1	1	1	1	1	1
9	314	502	2	11	1	1	1	1	1	1	1	1	1	1
10	294	470	23	2	1	1	0	1	1	1	1	1	1	1
11	314	502	10	7	1	0	1	1	1	1	1	1	1	1
12	301	481	32	9	1	1	1	0	1	0	1	1	0	1
13	288	460	−3	2	1	1	1	1	1	1	1	0	1	1
14	182	291	32	12	1	1	1	0	1	1	1	0	1	1
15	268	428	17	2	1	1	1	1	0	1	1	1	1	1

TABLE IV

Transit Times (Days) to Loading Port

		1	2	3	4	5	6	7	8	9	10	11	12	13
From Discharging Port	1	10	8	20	24	18	17	24	17	17	8	—	—	—
	2	13	13	1	16	17	6	11	9	29	15	29	28	27
	3	14	14	2	17	18	7	12	10	30	16	31	30	29
	4	14	14	1	17	17	7	12	9	30	16	30	29	28
	5	14	13	1	17	17	6	11	9	30	15	30	29	28
	6	12	12	1	15	16	5	10	8	28	14	28	28	27
	7	12	12	2	15	15	5	10	8	28	14	28	27	26
	8	13	13	6	14	15	4	5	7	23	15	23	23	22
	9	17	16	10	17	19	8	5	11	23	19	23	23	22
	10	16	16	9	17	18	7	5	10	23	18	23	22	21
	11	15	15	9	14	17	7	2	10	20	18	20	19	18
	12	21	20	32	20	26	26	18	28	12	19	3	2	2
	13	16	15	9	17	18	7	4	10	22	18	22	22	21
	14	5	6	10	16	14	7	14	9	24	8	—	—	—
	15	7	8	8	17	15	8	14	10	26	10	29	29	29
	16	24	23	27	17	26	23	16	25	13	23	4	3	2

Discharging Date

The present algorithm assumes that the time between loading and discharging is fixed and independent of the speed of the ships. This is obviously not true and can be changed easily. In the simulated problems, the discharging date was determined

TABLE V

Cargo Data

Cargo	Size 1000 cbft	Revenue	Load Date	Load Date Interv.	Load Port	Disch. Date	Disch. Port	Cargo Type
1	246	119602	4	1	3	32	9	3
2	224	121258	5	1	3	26	8	10
3	195	93605	6	2	7	29	8	6
4	323	239835	17	1	4	50	9	8
5	324	163372	24	5	11	45	12	9
6	312	119591	25	1	3	40	15	10
7	334	156881	44	2	2	74	2	2
8	181	112028	46	1	5	75	12	7
9	372	197541	47	1	1	71	15	7
10	359	149756	49	1	13	67	12	8
11	380	241198	50	1	12	81	11	6
12	252	87572	54	7	8	76	10	2
13	368	162809	55	1	13	86	8	1
14	256	229612	57	2	1	93	13	9
15	228	139652	69	7	7	97	12	7
16	292	254836	73	3	10	109	9	9
17	242	121184	78	3	1	112	10	1
18	201	140840	90	2	11	130	5	4
19	195	166104	95	1	8	142	12	4
20	360	207015	97	6	6	125	4	6
21	348	323952	97	2	12	135	8	9
22	230	133906	114	6	9	146	16	4
23	322	204072	116	1	5	149	16	5
24	356	313813	118	2	13	159	15	7
25	299	297937	119	3	9	172	4	5

as the loading date plus the transit time for the fastest ship type plus time for loading and discharging, which was chosen according to a uniform distribution between 1 and 20 days.

Loading Time Interval

The number of alternative loading dates was chosen according to a skewed distribution between 1 and 8 days, with an average of 3.2 days, in order to resemble the realistic example.

Initial Position and Open Date

The initial position for each ship was selected randomly with equal probabilities for the 16 discharging ports. The open dates were selected uniformly between time -11 and $+38$, because these were the extreme open dates in the realistic example.

TABLE VI

Optimal Solution

Ship	Income	Ballast Time	Idle Time	Cargo	Loading Date	Discharge Date
1	150989	10	68	17	78	112
2	203670	8	33	7	44	74
3	640537	20	0	5	26	47
		2	0	10	49	67
		2	28	21	97	135
4	232262	18	91	23	116	149
5	418546	5	24	9	47	71
		8	18	20	97	125
6	682806	14	3	4	17	50
		5	14	15	69	97
		2	19	24	118	159
7	251696	16	0	8	46	75
		3	12	18	90	130
8	373346	9	1	2	5	26
		7	21	12	54	76
		10	9	19	95	142
9	139454	2	2	3	6	29
10	45590	—	—	—	—	—
11	391684	2	13	6	25	40
		10	23	16	73	109
12	299549	23	64	25	119	172
13	161117	1	6	1	4	32
14	25608	—	—	—	—	—
15	360394	13	27	14	57	93
		22	0	22	115	147

Cargoes 11 and 13 are not shipped in the optimal solution.

Time Value

The time value was set proportional to the size of the ship both in the realistic example and in the simulated problems.

Cargo Revenue

In the realistic example, all cargoes were contracted. As the capacity of the fleet only slightly exceeds the demand in the simulated problems, most problems would be infeasible if all cargoes were contracted. Therefore only optional cargoes were used. The revenue of a cargo was set equal to the product of cargo size, time between loading and discharging and a random factor. This factor was selected uniformly between a constant C and $2C$. This constant should obviously be related to the time value for the ships. In most runs, C was chosen 1.5 times the time value per cubic foot. As the average ballast trip takes 18 days and the average cargo trip 28 days, the revenue from a trip will exceed the loss of idle-time income in most cases provided that the ship is not considerably larger than the cargo size or the required ballast trip is not too long compared with the cargo trip.

In order to resemble a situation with contracted cargoes, the constant C was in some runs increased fivefold, which increases the number of cargoes in the optimal schedule.

APPENDIX III

A Simulated Problem and Its Solution

This example is a 15-ship, 25-cargo problem with equal ship speeds. This was chosen in order to simplify the presentation of transit times (*see* Table IV on p. 65). All the data was determined according to Appendix II, except that the cargo revenues were chosen five times larger than the 'normal' values in order to simulate a situation with contracted cargoes. Idle time premium is used with breakpoint $T = 10$ days. The ship data is given in Table III, the transit times in Table IV, and the cargo data in Table V.

The optimal solution is presented in Table VI: 23 cargoes are shipped, and two ships are idle throughout the period. These ships are too small to carry these non-shipped cargoes, however. The option of delayed loading is used in two cases, cargo 5 is delayed two days on ship 3 and cargo 22 is delayed one day on ship 15.

REFERENCES

1. G. B. Dantzig, *Linear Programming and Extensions*, Princeton University Press, Princeton, N. J., 1963.
2. L. R. Ford and D. R. Fulkerson, *Flows in Networks*, Princeton University Press, Princeton, N. J., 1962.

(Received September, 1968).