Scheduling Ocean Transportation of Crude Oil

Gerald G. Brown; Glenn W. Graves; David Ronen

Management Science, Vol. 33, No. 3 (Mar., 1987), 335-346.

Stable URL:
http://links.jstor.org/sici?sici=0025-1909%28198703%2933%3A3%3C335%3ASOTOCO%3E2.0.CO%3B2-F

Management Science is currently published by INFORMS.

MANAGEMENT SCIENCE
Vol. 33, No. 3, March 1987
Printed in U.S.A.

SCHEDULING OCEAN TRANSPORTATION OF CRUDE OIL*

GERALD G. BROWN, GLENN W. GRAVES AND DAVID RONEN
Naval Postgraduate School, Monterey, California 93943-5100
University of California, Los Angeles, California 90024
Naval Postgraduate School, Monterey, California 93943-5100

A crude oil tanker scheduling problem faced by a major oil company is presented and solved using an elastic set partitioning model. The model takes into account all fleet cost components, including the opportunity cost of ship time, port and canal charges, and demurrage and bunker fuel. The model determines optimal speeds for the ships and the best routing of ballast (empty) legs, as well as which cargos to load on controlled ships and which to spot charter. All feasible schedules are generated, the cost of each is accurately determined and the best set of schedules is selected. For the problems encountered, optimal integer solutions to set partitioning problems with thousands of binary variables have been achieved in less than a minute.
(TRANSPORTATION—PLANNING; SET PARTITIONING; ENUMERATIVE METHODS)

1. Introduction

Proper scheduling of ocean transportation presents large economic potential. Vessels cost tens of thousands of dollars daily and consume fuel while under way, at a similar rate. Thus, a system which exploits cost components to provide an optimal schedule for a fleet is valuable.

We present and solve a crude oil tanker routing and scheduling problem faced by a major oil company which controls a fleet of several dozen crude oil tankers of similar sizes (220–270 thousand deadweight tons), and uses them to ship crude oil from the Middle East to Europe and North America. The daily cost of such a ship for the period studied is around $10,000, and the daily cost of bunker fuel ranges from $5,000 to $40,000, depending on the cruising speed.

A voyage usually has a single loading port and a single discharging port and the cargo is a full shipload. Because the crude is shipped on a recurrent basis (at least several loads a month from any port and to any destination), the exact size of the ship which shows up to load a cargo is not critical, and the ships may be treated as having the same size for the purpose of cargo assignment. Some voyages may require loading or discharging the cargo at two adjacent ports.

Ships in ballast (empty) may be routed either through the Suez Canal (paying canal passage tolls) or around the Cape of Good Hope (often a much longer voyage). Loaded ships cannot pass through the Suez Canal and must be routed around the Cape. (See Figure 1.)

Each cargo is specified by its loading port, loading date, discharging port and discharging date (thus defining a cargo "slate"). Most cargos on a slate may be carried either by the fleet under the operator's control or by spot charters; due to operational considerations, certain cargos must not be carried by spot charters. In addition, certain ships may not be allowed to enter certain ports or to pass through the Suez Canal, or may be not available for scheduling during certain parts of the planning horizon.

The planning horizon extends up to three months forward, but changes in the cargo slate, ship availability, and crude requirements of refineries and customers necessitate

* Accepted by Alexander H. G. Rinnooy Kan; received May 1983.

0025-1909/87/3303/0335$01.25

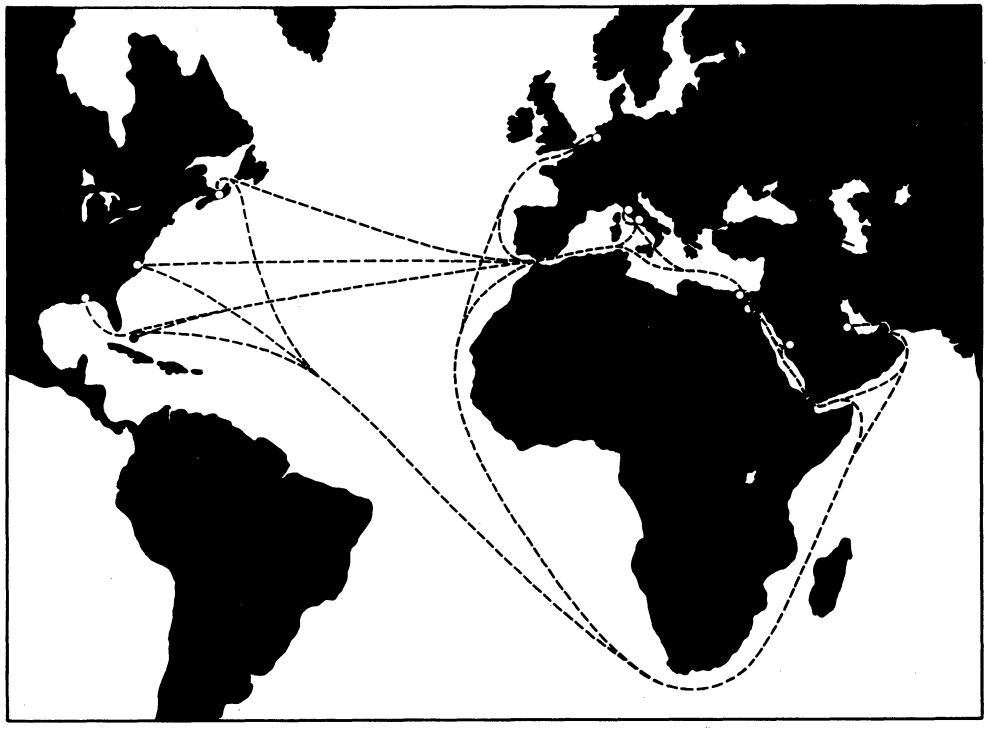

FIGURE 1

frequent updates. Such changes may require cruising speed alterations and diversion of ships underway to a different loading or discharging port. Voyage legs (between two ports) may take from two days up to two months.

The major concern of the fleet operator is how to ship crude oil at minimal cost. Scheduling decisions incur the following types of costs:

1. *Daily cost* represents the opportunity (or possession) cost of ships.
2. *Bunker fuel* consumption is a super-linear function of the cruising speed of a vessel.
3. *Fuel for auxiliary systems* is consumed while not steaming (in steam ships), or daily (in motor ships).
4. *Port and Canal dues* are ship-specific costs incurred when entering a port or passing through the Suez Canal.
5. *Spot charter costs* accrue from hiring ships to carry single cargos.
6. *Cost of an idle ship* is a combination of daily cost, fuel for auxiliary systems and port charges. For chartered vessels this is the amount of money paid to owners for delays, or demurrage cost.

Additional cost components exist, but they are not affected by the fleet scheduling decisions. The problem environment is summarized in Table 1.1.

2. Problem Formulation

Ship scheduling problems have attracted moderate attention in the published literature. A comprehensive review of ship routing and scheduling problems and models was provided by Ronen (1983). Ship scheduling problems similar to the one presented here were treated in a military environment by Dantzig and Fulkerson (1954) who minimized the fleet size, Flood (1954) who minimized the length of the ballast legs, and McKay and Hartley (1974) who tried to minimize fleet operating costs and the cost of

TABLE 1.1
Problem Structure, Assumptions and Constraints

1. *The Operation*
   * Major Oil Company
   * Shipping crude oil from the Middle East to Europe and North America
   * Several loading ports
   * Several discharging ports
   * No return cargos
   * Shipments on recurrent basis
2. *The Ships*
   * Similar sizes (220–270 KDWT)
   * No loaded legs through the Suez Canal
   * Limitations on ports of call and Suez Canal passage
   * A ship may be unavailable during part of the planning horizon
   * Ships may be diverted at sea
3. *The Ports*
   * Known loading/discharging duration for each port
4. *The Cargos*
   * A single loading port and a single discharging port area per cargo
   * A cargo is a full shipload
   * A cargo is specified by: loading port, loading date, discharging port, and discharging date
   * Most cargos may be spot chartered
5. *The Costs*
   * Daily cost of the ships
   * Bunker fuel
   * Fuel for auxiliary systems
   * Port and Canal dues
   * Spot charters
   * Idle ship, or demurrage

buying oil products at the loading ports. McKay and Hartley used binary route selection variables (an approach similar to the one taken here), but employed continuous solutions and an approximate heuristic. Similar, nonmilitary ship scheduling problems were dealt with by Laderman, Gleiberman and Egan (1966) who tried to minimize the number of ships used, Rao and Zionts (1968) and Dantzig, Blattner and Rao (1967) who tried to minimize operating and chartering costs, and Appelgren (1969, 1971) who maximized profit contribution of optional cargos.

Fleet scheduling decisions are discrete: in order to minimize costs the problem invites formulation as an integer program. Moreover, in order to include the nonlinear relationship between cruising speed and bunker fuel costs, and to allow evaluation of alternate routes for ballast legs, a schedule selection model is attractive. Schedule selection models were applied to planning cargo transportation by McKay and Hartley (1974), Foster and Ryan (1976), and Crawford and Sinclair (1977), all of whom considered only subsets of the feasible schedules and used rounding heuristics for noninteger solutions. Appelgren (1971) used Dantzig-Wolfe decomposition combined with a branch-and-bound algorithm to solve a cargo selection problem, and attributed his success to the simple structure of the master LP problem.

We have not found work which considers the relationship between cruising speed and bunker fuel costs (for a further discussion of this aspect see Ronen 1982), alternate routing of a ship leg, nor complete operating expenses of the fleet. In addition, ship operators have recently achieved substantial improvements in data management and computational technology which we seek to exploit.

We approach the problem by using: (1) a column generator which provides a com-

plete set of feasible schedules, (2) an accurate cost calculator which evaluates every schedule, (3) preordering and segmentation of the coefficient matrix, and (4) an efficient solution procedure. We formulate the problem as an *Elastic Set Partitioning Problem* (ESPP), which is a modification of the following *Set Partitioning Problem* (SPP):

Index Use. $i = 1, \ldots, n$ ships; $j \in J(i)$ set of feasible schedules for ship i; $k = 1, \ldots, l$ cargos.

Data

$$v_{kj} = \begin{cases} 1 & \text{if schedule } j \text{ carries cargo } k, \\ 0 & \text{otherwise,} \end{cases}$$

c_j = cost of schedule j (a function of ship and cargos).

Decision Variables

$$y_j = \begin{cases} 1 & \text{if schedule } j \text{ is selected,} \\ 0 & \text{otherwise.} \end{cases}$$

Classical Formulation (SPP)

$$\text{Min} \sum_j c_j y_j \qquad \text{s.t.} \quad \sum_{j \in J(i)} y_j = 1 \qquad \text{for each ship } i, \tag{1}$$

$$\sum_j v_{kj} y_j = 1 \qquad \text{for each cargo } k, \tag{2}$$

$$y_j = \{0, 1\} \qquad \text{for each schedule } j. \tag{3}$$

Constraints of type (1) require each ship to follow exactly one schedule during the planning horizon, and constraints of type (2) stipulate that every cargo will be loaded. Every schedule is a feasible sequence of cargos during the planning horizon with specified routing of ballast legs, and the set of schedules $J(i)$ for ship i includes all the feasible schedules (and only feasible schedules) for that ship during that period. All feasible schedules with different cruising speeds in ballast, as well as schedules with alternate routing of the ballast legs (Suez or Cape) are included in the feasible set for each ship. (Ship cruising speed while loaded is not a decision variable because it is determined by the loading and discharging dates of the cargo.) A schedule may include idle periods as well as ballast legs. One candidate schedule for each ship represents total idleness and one schedule for each cargo that may be spot chartered will represent such a charter. The maximal length of time a ship is permitted to wait for loading a cargo may be specified, thus reducing the number of feasible schedules (this option was not exercised here due to the economies of the problem, where ships may not necessarily cover their variable operating costs).

3. Solution Methods

We accommodate complex realistic detail and exploit economic opportunities in this problem by employing a schedule generator, a cost calculator and an optimization model.

For a given planning horizon the *schedule generator* accepts a complete description of the fleet and cargos, as well as data describing loading ports, discharging ports, canal passages and sea routes. Initial conditions include the current position, status and planned availability of all ships.

The generator provides schedules and speed ranges which satisfy all operating restrictions and reflect intricate rules and relationships specific to individual ships, ports, and

cargos, or interactions of combinations of these. These details contribute crucial realism to the acceptability of the model. Only feasible independent schedules are generated (in the sense that no generated schedule can be composed from other generated schedules).

One-pass schedule generation over time with no back-tracking and complete utilization of computer memory is performed as follows. A cargo is feasible for a ship schedule if the ship can arrive in time to load it. For every ship, each cargo that is feasible as first cargo in a schedule is seeded as first cargo in a successive contiguous column (schedule). Then, each one of these (formerly generated seed) columns is concatenated ahead of each feasible second cargo and inserted as a new column. This process is repeated for additional cargos, until no additional feasible cargo can be added to a column within the planning horizon. The columns are stored in two arrays: one contains the cargo indexes and the other has one entry for each column pointing to the location of the first cargo of that column in the former array. Most of the computation in the schedule generator is composed of index references, not arithmetic.

The *cost calculation model* uses the same input data as the schedule generator to determine the complete cost of any schedule (or set of schedules). It is also a simulation model, and includes imbedded logic to insure optimal choice of speed underway, as well as cost components ranging from bunker fuel consumption to ship-cargo-specific port entry fees.

The cost calculations involve a considerable amount of arithmetic, but the computations are only applied to complete feasible schedules.

From the schedules, and associated costs, the *optimization model* constructs a set partitioning problem representing the actual minimum-cost scheduling process. SPP's are widely believed to be very difficult to solve, especially at large scale.

However, we have found that much of this unruly reputation derives from shortcomings in SPP modeling practice and in the design of classical linear programming (LP)-based optimization systems. In particular, we scrutinize the effects of:

1. Degeneracy,
2. Numerical Instability,
3. Integer Enumeration, and
4. Exploiting Problem Knowledge

on the algorithm for solving set partitioning problems.

The costs represented in the SPP at hand are far too great to permit retreat to the use of an approximate heuristic. Worse yet, the costs are pathological for heuristics we know to be widely used in this industry (as revealed by Ronen 1986).

SPP's are *highly degenerate.* A solution to an SPP using a minimal number of schedules (columns) is maximally primal degenerate. Further, SPP's frequently exhibit massive dual degeneracy as well (arising from problem cost structure). These degeneracy conditions inflict grievous performance penalties on optimization systems lacking constructive degeneracy resolution.

Numerical instability is endemic to SPP's. SPP bases reliably exhibit near-singularity, especially for complex routing and scheduling problems possessing multitudes of large, nearly interchangeable schedule sets. Perversely, stability problems become most acute when the SPP bases achieve maximal dimension approaching an optimal continuous solution, and consequently, throughout integer enumeration.

Frequent basis reinversion cosmetically treats the symptoms of degeneracy and numerical instability; cycling solutions are interrupted (at least probabilistically) by row reordering, and numerical precision is improved. However, computational experience with solving set partitioning problems provides compelling evidence that such frequent interruption of solution progress is prohibitively costly.

Integer enumeration of SPP's can be particularly vexing for classical optimization methods, which focus exclusively on achieving a fixed hierarchy of solution properties:

1. Feasibility, and then
2. Continuous Optimality, and then
3. Integer Optimality.

During classical (fixed order) integer enumeration of an SPP, fixing (or reversing) binary variables in an optimal, feasible restricted solution insures immediate loss of feasibility. This requires iterations to reachieve feasibility, and then optimality (including perhaps several degenerate iterations to verify an optimal solution representation). The number of schedule (column) exchanges required can be immense.

For many optimization systems, dual (feasibility seeking) simplex operations are much more costly than primal (optimality seeking) operations. Coupled with the continuing necessity to further delay progress with reinversions, the situation frequently become hopeless.

Problem knowledge is seldom restricted to the static structure of the SPP at hand. For models of real systems, there is always a wealth of information available, ranging from industry "thumb rules" to actual manual solutions. Some classical optimization systems admit advanced starting solutions. This is seldom of much help for a SPP: a particular (infeasible and/or suboptimal) solution is far less valuable than some expression of the rules and preferences used in its derivation.

Our solution approach for SPP uses the X-System (Brown and Graves 1975, 1986), an optimization system endowed with several useful advanced features. To exploit these, it is convenient to restate SPP.

In actual operations, some constraints may be violated, but at a cost. In particular, total idleness (demurrage) of ships and spot chartering of cargos may be viewed as constraint violation mechanisms embedded in SPP. The following formulation makes this more explicit.

Elastic Formulation (ESPP)

$$\text{Min} \sum_j \tilde{c}_j \tilde{y}_j + \sum_i (\underline{d}_i \underline{\delta}_i + \bar{d}_i \bar{\delta}_i) + \sum_k (\underline{s}_k \underline{\sigma}_k + \bar{s}_k \bar{\sigma}_k)$$

$$\text{s.t.} \quad \sum_{j \in \tilde{J}(i)} \tilde{y}_j + \underline{\delta}_i - \bar{\delta}_i = 1 \qquad \text{for each ship } i, \tag{1e}$$

$$\sum_j \tilde{v}_{kj} \tilde{y}_j + \underline{\sigma}_k - \bar{\sigma}_k = 1 \qquad \text{for each cargo } k, \tag{2e}$$

$$\tilde{y}_j \in \{0, 1\} \qquad \text{for each schedule } j. \tag{3e}$$

Constraints (1e) admit total idleness of ships (at total demurrage costs $\underline{d}$), and preserve model integrity with disruption costs $\bar{d}$. Constraints (2e) permit spot charters (costing $\underline{s}$), with $\bar{s}$ model composition costs analogous to $\bar{d}$. For cargos which should not be spot chartered, $\underline{s}_k$ also becomes a relatively high model cost. The variables $\tilde{y}$ in ESPP do not include schedules for total idleness of ships or spot chartering of cargos; $\tilde{J}$ is the consequent contraction of the index set J in SPP; δ and σ are logical variables with cost coefficients referred to as *elastic penalties*.

The elastic penalties $\bar{d}$ and $\bar{s}$ are not infinite. Rather, they are set to be just large enough to produce the desired model composition. In this work, a constraint class penalty $\bar{D}$ is specified for constraints (1e), and each penalty is determined by: $\bar{d}_i = \bar{D}/(|\tilde{J}(i)| + 1)$. $\bar{s}$ is developed in the same fashion for (2e) from a constraint class penalty, $\bar{S}$. $\bar{D}$ and $\bar{S}$ require some thought by the modeler and benefit from knowledge of the problem. In this context, the elastic penalties may be viewed as bounds on the LP dual variables.

The X-System algorithm is ambivalent between primal and dual operations, and uses an extremely condensed working data structure. Elastic penalties are incorporated *logically* by a generalized simplex algorithm. Complete, constructive degeneracy resolu-

tion (i.e., Graves 1965) is applied when cycling is encountered (usually near the verification of a solution). An extremely fast reinversion is used (only) in the presence of apparent numerical instability; the reinversion exploits problem knowledge in the form of static basis factorization (e.g., generalized upper bounds (1e)), key row and/or column agendum, elasticity, and degeneracy, to produce an equivalent, condensed inverse representation.

Elastic integer enumeration works reliably and exploits problem knowledge, principally expressed by the elastic penalties. The (fixed order) enumeration seeks (elastically feasible) integer solutions by immediate *rounding* at each branch vertex, using *rounding penalties* developed from the elastic (ESPP) objective. Vertices are developed by branching away from the highest rounding penalty encountered. Fathoming is controlled by tolerance for deviations of (elastic objective) restriction values from incumbent value, and by enumeration depth (maximum fixed variables in any restriction).

An additional *block partitioning* refinement has proven to be very effective with large SPP's. A lexicographic in-situ sort is employed to express the SPP in block-staircase form (e.g., Christofides and Korman 1975, Garfinkel and Newhauser 1969, Pierce 1968). Rows may also be reordered by length (e.g., Marsten 1974).

We find the intrinsic ordering by available date (of ships or cargos) to be attractive (expressing problem knowledge). A prototypic BLOCK algorithm is shown in Appendix A.

Next, the (continuous relaxation of the) SPP problem monolith is divided into a number b of *block sets* (say $b = 4$, each with about $1/b$th of the rows from ESPP), forming a suite of distinct subproblems. The subproblems are solved in a *cascade*—a concatenating sequence of block sets: each successive solution is appended to its predecessors until the monolith is solved. In each cascade concatenation, constraints not in the block set are relaxed, and variables not in the block set are fixed at their most recently determined values; a basis crash is used to seed the block set and hasten its solution (and basis save for following crashes). (Acknowledging the time value of features in our ESPP, we use a block set cascade such as: $\{1, 1\}$, $\{2, 2\}$, $\{1, 2\}$, $\{3, 3\}$, $\{1, 3\}$, . . . , $\{b, b\}$, $\{1, b\}$, where $\{1, 1\}$ includes the earliest $1/b$th of the rows from ESPP.)

4. Results

Current operational data have been used to evaluate our approach. A representative test problem is described in Table 4.1.

Table 4.2 shows some model characteristics for various time horizons. The number of candidate ship-schedules is the number of (binary) decision variables in the (condensed) ESPP. Cargos per schedule gives the average number of cargos carried by each ship-schedule; this schedule length is a common measure of difficulty for SPP problems. Schedule generation and cost calculation times are shown (accurate to the precision displayed for IBM 3033 (AP) with VM/CMS using FORTRAN H (Extended) OPTIMIZE (2) in 1-megabyte maximum virtual storage).

TABLE 4.1
Test Conditions

0–80	Day Planning Horizon
50	Cargos (25 may be spot chartered)
24	Controlled Ships
3	Loading Ports
9	Discharging Ports

TABLE 4.2
Model Generation

Time Horizon Days	Candidate Ship-Schedules	Cargos per Schedule	Schedules Generation Seconds	Cost Calculation Seconds
10	24	1.0	0.00	0.01
20	89	1.1	0.03	0.04
30	188	1.4	0.08	0.04
40	367	1.5	0.12	0.09
50	786	1.9	0.15	0.20
60	1,535	2.2	0.21	0.51
70	3,408	2.6	0.38	1.36
80	7,349	3.2	1.00	3.53

Our concern with the (relatively short) schedule generation and cost calculation times stems from a likely requirement that these steps be executed as imbedded transactions in an on-line information management system (IBM/IMS).

Optimal solution times using the X-System are given in Table 4.3. Block identification time and the resulting number of blocks are shown. All problems have been solved to integer optimality. Optimization time and pivots are given with the percentage of time devoted to solving the LP relaxation.

For these runs, all numeric error tolerance levels, enumeration rules and algorithmic control parameters were set to default values, and no problem-specific or solution-trajectory sensitive dynamic tuning was invoked. A 4-block set problem cascade was used.

The X-System expended almost all effort on dual pivot selection. Note the small number of solution pivots in relation to the long solution times. With this cautious strategy, *no* reinversions were demanded during *any* solution.

Although the SPP-derived LP's are notoriously difficult, the associated integer optimization is often relatively easy. We have never had difficulty with any of these integer ESPP models, once the LP is solved. (However, this is not the case for all ESPP models we have encountered.)

To test the efficacy of our blocked problem cascade approach, Table 4.4 shows the equivalent optimal solution times without this feature. We see that the solution times and instability both increase for larger problems.

By forcing primal pivot selection, solution times and instability are much worse. These results are too ghastly to report here.

TABLE 4.3
*Optimization (BLOCK Cascade Sets b = 4)**

Time Horizon Days	BLOCK Sec.	Blocks	Optimization Sec.	Optimization Pivots	% LP Time
10	0.05	8	3.78	25	99.7
20	0.07	16	3.83	72	99.5
30	0.10	20	4.07	96	98.8
40	0.19	26	5.24	260	98.7
50	0.41	31	6.25	173	98.1
60	0.70	37	9.13	202	97.3
70	1.47	45	23.50	376	95.7
80	3.41	50	80.43	798	97.4

* Times shown are achieved from a cold start. A faster hot start procedure is used in practice.

TABLE 4.4
*Optimization (No BLOCK Cascade)**

Time Horizon Days	Optimization		% LP Time	Stability Reinversions
	Sec.	Pivots		
10	3.35	26	99.4	1
20	3.66	60	98.9	0
30	3.82	93	98.7	0
40	4.81	199	97.9	1
50	7.65	276	98.2	0
60	14.17	364	97.8	1
70	45.77	586	98.0	1
80	163.28	857	98.5	2

* This is not the preferred approach.

Performance can be improved by tuning. The 80-day model solves in 7.61 seconds and 603 pivots with the assistance of a trivial starting solution (first-come first-served by ship availability). Manual assistance from a human scheduler can produce even better performance.

Some properties of optimal shipping plans are shown in Table 4.5. Controlled ships are company-owned and time-chartered ships; this controlled fleet increases in size with the time horizon because ships are released from prior commitments (assignments not under our control) as time progresses. The idle ships are those that become available for scheduling too late to load any cargo within the planning horizon, and those that are found uneconomical to operate with the given cargos and fleet. (Most of the idle ships fall in the former category.) Similarly, the number of cargos shipped grows over time, and is composed of controlled ship assignments and spot charters. The maximum number of cargos carried on a single schedule in the optimal plan is shown.

The cost per deadweight-ton-mile (DWTM) on loaded legs is a relative figure-of-merit which is affected by numerous factors. (The actual costs are proprietary information, but the order of magnitude of total costs is one million dollars per day.) Percentages of total costs by category help visualize the optimal fleet mix scheduling decisions.

Commonality of solution elements over an increasing time horizon would lend reassurance of solution stability and robustness. Although it is difficult to meaningfully convey in tabular form the degree of commonality among these solutions over time, Table 4.6 is an attempt to do this. The "cargos changed" column shows the number of cargos assigned to a different ship from one time horizon to the following one (con-

TABLE 4.5
Optimal Shipping Plan

Time Horizon Days	Controlled Ships		Cargos		Max. Cargos	Cost Per DWTM	Cost % Used, Idle; Spot
	Used	Idle	Controlled	Spot			
10	5	8	5	3	1	896	67, 6; 27
20	11	5	12	4	2	907	76, 4; 20
30	13	5	15	5	3	896	78, 4; 18
40	16	2	19	7	3	875	76, 1; 23
50	16	3	21	10	4	896	73, 2; 25
60	17	5	22	15	4	904	65, 2; 33
70	20	2	30	15	4	940	68, 1; 31
80	22	2	34	16	5	935	69, 0; 31

TABLE 4.6
Stability of Optimal Shipping Plans

Time Horizon Days	Cargos		Cargos Changed		Ship Additions
	Controlled	Spot	Controlled	Spot	
10	5	3			
20	12	4	4	0	3
30	15	5	0	0	2
40	19	7	3	1	0
50	21	10	2	1	1
60	22	15	7	0	3
70	30	15	12	3	0
80	34	16	2	0	2

trolled cargos changed among controlled ships or to spot charters and spot cargos changed to a controlled ship). "Ship Additions" is the number of controlled ships that have become available since the preceding time horizon. These ships are usually at discharging ports, and it takes time to get them to a loading port.

Generally, conditions never seem to stabilize. That is, new availability of one additional cargo or ship can be expected to wreak havoc on the entire shipping plan. Even ideally configured ships that naturally gravitate to regularly available compatible cargos will be completely rescheduled upon introduction of a new, superficially unrelated cargo or ship. These results demonstrate a crucial advantage of optimization: lucrative opportunities are revealed which are not intuitively apparent.

Our numerical experience confirms opinions expressed by schedulers in interviews. Ship scheduling is complex. The workload is too high to permit extensive manual analyses of the future effects of frequent changes in availabilities; it is extremely unlikely that such analyses could consider costs in realistic detail. Accordingly, manual ship scheduling is an art.

Considering the huge operating costs, the costs of mistakes (say, a hundred thousand, or even a million dollars for some minor oversight), the complexity of the scheduling task, and the volatility of requirements and availabilities, we wonder why modeling plays virtually no role in the shipping industry, and why optimization in particular is not the premier line of attack.

5. Practical Aspects and Extensions

The schedule of the ships is driven by the crude slate (i.e., the specification of the cargos and their loading and discharging dates and ports). The crude slate is determined by the crude availability at the loading ports and the requirements of the refineries and other customers. Ideally, the scheduling of both the crude and the ships should be done simultaneously, but scheduling the crude is outside the scope of this work. The issue of multiple commodities has not arisen although several grades of crude are involved: the ships have sufficient compartmentation to accommodate the parcel sizes of the various grades.

Part of the crude is shipped through the Suez-Mediterranean (SUMED) pipeline; this necessitates two loadings, one at the port of origin and a second at the Mediterranean SUMED outlet (after being discharged at the Suez inlet). Due to storage capacity limitations, the quantities of noncommon grades of crude shipped through the SUMED pipeline must follow certain parcel sizes. If the model presented here is expanded to schedule the crude, it can easily be modified to determine the best routing of each crude shipment (SUMED vs. around the Cape). Adding crude scheduling to the

model will require consideration of crude availabilities at other sources and may even require consideration of where to buy the crudes.

Another issue which has arisen from this work is the comparison of alternate shipping plans. A current plan is maintained three months in advance, but due to frequent changes in ship availability and crude requirements and availability, the plan is changed every few days. Thus, high uncertainty is involved in scheduling events near the end of the planning horizon, and the cost of these far future events might better be discounted to a present value.

An additional consideration in comparing shipping plans is the inclusion of ballast steaming time near the end of the planning horizon (for ships which unload and do not load another cargo within that horizon, or analogously, loaded voyages that extend beyond the planning horizon). These costs have been included in the present model in order to allow evaluation of out-chartering alternatives.

The operator wants to minimize the cost of the shipping plan and *also* to maximize the quantities loaded at the initial loading ports; thus, he has been provided with the total initial loadings for each plan, as well as a measure of the uncommitted controlled ship capacity during the planning horizon. Multi-objective model enhancements may be adopted to deal directly with several of these issues.

Column generation can be much more selective. Both the schedule generator and the cost calculation model can be used to edit candidate schedules a priori. Mechanized pricing can be used to generate schedules on-the-fly during optimization, but this would require a complete redesign of the schedule generator. Neither restriction has been necessary for the model at hand.

Ships are often used as floating storage with no firm discharge date or port, and are accommodated by the model with a dummy discharge port. (The discharge date in such a case is far beyond the planning horizon.)

The possibility of diverting ships in ballast while at sea to a different loading port is built into the model, but diversion of loaded ships to different discharging ports is not done automatically because it involves changes in crude slates.

The model has been developed for interactive use to allow the human scheduler to manually assign cargos to ships before the model determines the best assignment of the remaining cargos and ships. We find that human intervention is necessary to cope with situations that are not reflected in our models, and to gain better acceptance of the results by the scheduler.[1]

[1] We thank Bruce Bishop, Jim Craig and Chevron Corporation for their support of this research. Insight Corporation provided us the opportunity to work on the problem. This project was completed in 1982, when David Ronen was at the University of Missouri-St. Louis. Rick Rosenthal generously contributed an assiduous final edit of the penultimate, accepted manuscript.

Appendix A

Algorithm BLOCK

Purpose: Label each LP row and column with a block index which can be used to express the problem in block-staircase or other conveniently partitioned form for efficient solution with a problem cascade.

Input: The LP coefficient matrix A.

Output: A set of block indices for rows RB and a set of block indices for columns CB.

Comment: The basic algorithm is a "greedy" addition heuristic. The statement in square brackets can be modified for other selection rules.

Step 0: "Initialization"

(a) $RB = \Phi$, $CB = \Phi$, $NB = 0$.

Comment: NB is a running block index counter.

(b) For each row i, a row count $RC_i = \sum_{a_{ij} \neq 0} 1$.

Step 1. "Block Identification"

[Let i be a row with $RB_i = \Phi$ and the smallest row count.]

If $i \neq \Phi$ then

(a) Start new block with NB = NB + 1.
(b) Label row RB_i = NB.
(c) For each column j such that $a_{ij} \neq 0$ and $CB_j = \Phi$,
  (i) Label column CB_j = NB.
  (ii) For each row i^* such that $a_{i^*j} \neq 0$ and $RB_{i^*} = \Phi$,
    (α) Let $RC_{i^*} = RC_{i^*} - 1$,
    (β) If $RC_{i^*} \leq 0$ then let RB_{i^*} = NB.
(d) Repeat Step 1.

Step 2. "Termination".
End of Algorithm BLOCK.

References

APPELGREN, L. H., "A Column Generation Algorithm for a Ship Scheduling Problem," *Transportation Sci.*, 3 (February 1969), 53–68.

———, "Integer Programming Methods for a Vessel Scheduling Problem," *Transportation Sci.*, 5 (February 1971), 64–78.

BAUSCH, D. O., "Computational Advances in the Solution of Large-Scale Set Covering and Set Partitioning Problems," M.S. Thesis, Naval Postgraduate School, 1982.

BROWN, G. G. AND G. W. GRAVES, *XS Mathematical Programming System*, perpetual working paper, 1986.

——— AND ———, "Elastic Programming: A New Approach to Large-Scale Mixed-Integer Optimization," paper presented at the ORSA/TIMS meeting, Las Vegas, Nevada, 17 November 1975.

CHRISTOFIDES, N. AND S. M. KORMAN, "A Computational Survey of Methods for the Set Covering Problem," *Management Sci.*, 21 (January 1975), 591–599.

CRAWFORD, J. L. AND G. B. SINCLAIR, "Computer Scheduling of Beer Tanker Deliveries," *Internat. J. Physical Distribution*, 7 (1977), 294–304.

CULLEN, F. H., J. J. JARVIS AND H. D. RATLIFF, "Set Partitioning Based Heuristics for Interactive Routing," *Networks*, 11 (1981), 125–143.

DANTZIG, G. B., W. O. BLATTNER AND M. R. RAO, "Finding a Cycle in a Graph With Minimum Cost to Time Ratio With Application to a Ship Routing Problem," in *Theory of Graphs*, P. Rosenstiehl (Ed.), Gordon and Breach, New York, 1967.

——— AND D. R. FULKERSON, "Minimizing the Number of Tankers to Meet a Fixed Schedule," *Naval Res. Logist. Quart.*, 1 (1954), 217–222.

FLOOD, M. F., "Application of Transportation Theory to Scheduling a Military Tanker Fleet," *Oper. Res.*, 1 (February 1954), 150–162.

FOSTER, B. A. AND D. M. RYAN, "An Integer Programming Approach to the Vehicle Scheduling Problem," *Oper. Res. Quart.*, 27 (1976), 367–384.

GARFINKEL, R. S. AND G. L. NEMHAUSER, "The Set Partitioning Problem: Set Covering with Equality Constraints," *Oper. Res.*, 17 (1969), 848–856.

GRAVES, G. W., "A Complete Constructive Algorithm for the General Mixed Linear Programming Problem," *Naval Res. Logist. Quart.*, 12 (March 1965), 1–34.

LADERMAN, J., L. GLEIBERMAN AND J. F. EGAN, "Vessel Allocation by Linear Programming," *Naval Res. Logist. Quart.*, 13 (1966), 315–320.

MARSTEN, R. E., "An Algorithm for Large Set Partitioning Problems," *Management Sci.*, 20 (1974), 774–787.

MCKAY, M. D. AND H. O. HARTLEY, "Computerized Scheduling of Seagoing Tankers," *Naval Res. Logist. Quart.*, 21 (1974), 255–264.

PIERCE, J. F., "Application of Combinatorial Programming to a Class of All-Zero-One Integer Programming Problems," *Management Sci.*, 15 (1968), 191–209.

RAO, M. R. AND S. ZIONTS, "Allocation of Transportation Units to Alternative Trips—A Column Generation Scheme with Out-of-Kilter Subproblems," *Oper. Res.*, 16 (January 1968), 52–63.

RONEN, D., "Cargo Ships Routing and Scheduling: Survey of Models and Problems," *European J. Oper. Res.*, 12 (1983), 119–126.

———, "The Effect of Oil Price on the Optimal Speed of Ships," *J. Oper. Res. Soc.*, 33 (1982), 1035–1040.

———, "Scheduling of Vessels for Shipment of Bulk and Semi-bulk Commodities Originating in a Single Area," *Oper. Res.*, 34 (1986), 164–173.