A Two-Phase Algorithm for Tramp Ship Routing Problems by a Column Generation Approach

Kazuhiro Kobayashi[1], Toshiyuki Kano[1], and Mikio Kubo[2]

[1] Center for Logistics Research,
National Maritime Research Institute, Tokyo, Japan
kobayashi@nmri.go.jp
[2] Department of Logistics and Information Engineering,
Tokyo University of Marine Science and Technology, Tokyo, Japan

Abstract. A column generation approach for tramper ship routing problems is studied. We show that an efficient algorithm can be obtained by decomposing the problem into two simple problems. The master problem of the column generation is a set partitioning problem and the subproblem is the time-dependent shortest path problems with time windows. Numerical results of problems using general-purpose integer-programming solver and the dynamic programming algorithm are included to demonstrate the performance of the proposed method.

Key words: ship routing problem, column generation, shortest path problem

1 Introduction

Shipping companies have many complex, extensive planning problems at three different levels. The strategic planning problems include the determination of the size and the number of company-owned ships compared with the cargo requirements. The tactical planning problems include the design of service network by creating the ship routes, and the assignment of ships to these routes. The operational planning problems include the determination of the plan to utilize the fleet. This paper investigates the tramp ship routing problem for oil-product transportation in the operational level and shows a solution approach for this problem.

A shipping company makes contracts with cargo owners to carry a set of cargoes in the planning horizon. The company tries to minimize the operating cost for carrying these cargoes because (a) improving fleet utilization can lead to significant improvements in financial results, and (b) increasing fleet utilization can reduce damage to the environment because of reductions in transport operations. The objective of the ship routing problem is to minimize the sum of the costs for all ships in the fleet while ensuring that all cargoes are carried

* This study was partially supported by the New Energy and Industrial Technology Development Organization (NEDO) of Japan.

from their loading port to their discharging port. In addition to determining the sequence of ports to be visited by each ship, it is also necessary to determine the way to allocate cargoes into the compartments on the ship.

In this paper, we propose a decomposition of the problem into two relatively simple problems, which parallels the steps typically taken by planners in the shipping company. This enables us to derive an efficient algorithm. We also propose the column generation process to solve the problem in a short amount of time.

This paper is organized as follows. In Sect. 2, we describe the background of this research. In Sect. 3, we describe the planning problem that we consider. The two important constraints in this problem is the time windows constraint and the allocation constraints. We explain these two constraints. In Sect. 4, we propose a decomposition of the planning problem into the two relatively simple problems; the pairing problem and the routing problem. In Sect. 5, we describe the column generation process to solve the routing problem. The subproblem is the time-dependent shortest path problem with the time windows constraint and the allocation constraint, which is $\mathcal{NP}$-hard.We propose a reformulation of the subproblem, which can easily be solved by a dynamic programming approach. In Sect. 6, we show the numerical results. In Sect. 7, we give the concluding remarks.

2 Background

A ship involves a major capital investment and its daily operating costs can be in tens of thousands of dollars. Hence, it is important to develop optimization-based decision support systems for efficiently managing the fleet operations. See the surveys by Ronen [1, 2] and Christiansen et al. [3]. The surveys cover a large variety of planning problems at different planning levels.

A lot of underlying mathematical structures are common to the problems in road transportation, marine transportation and air transportation. However there are some differences of operating environments between these transportation modes. For example, the ship routing involves a much larger variety in problem structures and operating environments than standard vehicle routing in road transportation. Therefore, it is necessary to develop a tailor-made system for each company. Another example is that there is a relatively high degree of uncertainty in the operation of ships, as they often experience delays due to the changes of weather conditions. Due to the high uncertainty and long voyages, it is difficult to fix the schedules more than a few voyages ahead in time. Therefore, the number of feasible solutions in ship routing problems is often smaller than that in the vehicle routing problems in road transportation. Moreover, ship routing problems are also often relatively well constrained, because of incompatibilities between ships, ports and cargo requirements. This reduces the solution space. Because of the relatively small and well constrained problem, ship routing problems are often approached by first generating feasible schedules for the ships in the fleet, and then solving the problem as a set partitioning problem, where the columns represent the schedules. Brown et al.[4] considered a crude oil scheduling problem. Their approach generates all feasible solutions and then solves a set partitioning problem. The model determines optimal speeds for the ships and the best routing of empty legs, as well as which cargoes should be carried by spot charters. In their problem, a cargo is a full shipload.

The ship routing problem that we consider becomes a multi-vehicle pickup and delivery problem with time windows [5] with the allocation constraint [6]. Fagerholt and Christiansen considered a problem that is quite similar to the one we consider, except that the each ship in the fleet is equipped with a flexible compartments that can be partitioned into several smaller parts in a given number of ways [7]. The problem we consider may include a huge number of feasible routes. Thus we use the implicit enumeration by a column generation approach. In the early paper, Appelgren [8, 9] proposed column generation approaches to solve the ship routing problem. Recently, two chapters in [10] are devoted to column generation approaches for some ship routing problems.

3 Problem Description

The shipping company receives cargo transportation requirements in the form by quantity, product name, loading port, discharging port, loading day and discharging day (Table 1). To satisfy the transportation demands, a fleet of ships with different capacities and speed is available. Due to seasonal variations in demand, the fleet is not designed to handle all cargoes throughout the year. Thus, some of the cargoes may be serviced by spot charters with known costs. The decision of whether to use a spot charter or to use a ship in the controlled fleet depends on the cost of using spot charters. The planner in the shipping company tries to find a feasible route for each ship in the fleet, and to determine which cargoes are to be carried by spot charters, such that the total costs are minimized.

Table 1. cargo information

loading port	discharging port	loading day	discharging day	product	quantity
Tokyo	Osaka	10	12	A	600
Osaka	Nagoya	9	11	B	300

For the ships in the fleet, only the variable costs are considered, as the fixed costs have no influence on the planning of optimal schedules in the planning horizon. Here we assume that the cost of a plan is defined as the sum of the operating costs of all ships and the costs of using spot charters. Here, the operating cost of a ship is defined as a linear function of its traveling distance.

3.1 Time Windows Constraint

Time windows constraint should be satisfied for each cargo, both at the loading port and the discharging port. The time windows constraint ensures that the loading operation of a cargo starts by the loading day and the discharging operation starts by the discharging day. Generally, the ports are closed for service at night. More specifically, it is open for loading and discharging between the specified opening time and the closing time. Let us consider the loading port which opens at 8:00 am and closes at 3:00 pm. In case that the ship arrives at the port between 8:00 am and 3:00 pm, the loading operation starts as soon as the ship arrives at the port. When a ship arrives after the 3:00 pm, the loading operation starts at 8:00 am in the next day. Therefore, the total time in port will depend on the ship's arrival time. Note that the loading and discharging times for each cargo are given.

3.2 Allocation Constraint

Several cargoes can be carried by a single ship as long as there are separated. It is not allowed to mix different cargoes on board the ship. In order to separate several cargoes each other, each ship is equipped with five or six fixed compartments. This enables a ship to carry several cargoes simultaneously as long as different cargoes are loaded into different compartments. Note that each cargo should be carried by a single ship. In other words, it is not allowed to divide a cargo into multiple parts and deliver them by multiple different ships. The allocation constraint ensures that the allocation of cargoes to the compartments are feasible throughout the visiting sequence. Once a cargo is loaded in some compartments, it will not be relocated in another set of compartments before discharging. In contrast to full shipload cases, the ship often discharge only some of cargoes on board before it again starts loading. Figure 1 shows an example of an allocation of two cargoes to five compartments.

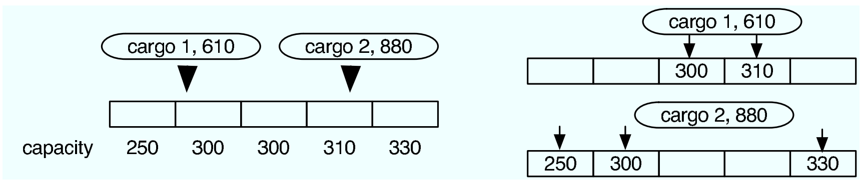

Fig. 1. allocation of two cargoes to five compartments

4 Decomposition of the Problem into Two Problems

We propose a method that decomposes the planning problem into two relatively simple problems. The first problem is the pairing problem, and the second problem is the routing problem. The solution of the pairing problem is used as an input to the routing problem. As a solution of the routing problem, we obtain a schedule for each ship and a plan to use spot charters.

4.1 Pairing Problem

In order to carry multiple cargoes by a single ship, there should exist a visiting sequence of ports for which the time windows constraint and the allocation constraint are satisfied. If the time windows constraint and the allocation constraint are satisfied for a given visiting sequence, we call it a feasible sequence. Due the differences of the characteristics between the ships, the feasibility of a visiting sequence depends on the ship to be used.

Let us consider a set of two cargoes in Table 2 which are carried by a ship with the five compartments as in Fig. 1. For these cargoes, a feasible visiting sequence of ports is shown in Fig. 2. The column 'load. pt.' indicates the loading port, 'disc. pt.' the discharging port, 'load. day' the loading day, 'disc. day' the discharging day, 'prod.' the product, and 'quant.' the quantity. Now we define a pairing as a set of cargoes for which there exists a visiting sequence of ports that is feasible by at least one ship in the fleet. The objective of the pairing problem is to obtain a set of efficient pairings so that efficient schedules can be generated in the second phase routing problem.

Table 2. pairing consisting of two cargoes

	load. pt.	disc. pt.	load. day	disc. day	prod.	quant.
cargo 1	Fukuoka	Tokyo	4	6	A	610
cargo 2	Hiroshima	Tokyo	5	6	B	880

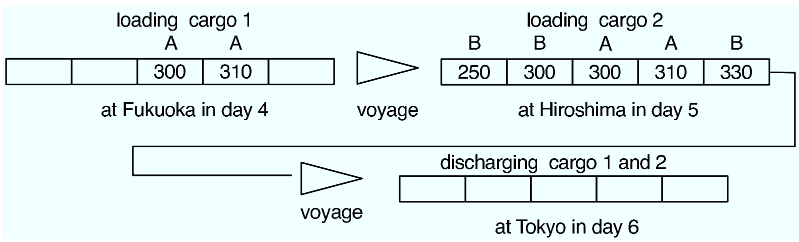

Fig. 2. feasible visiting sequence

There may exist several feasible visiting sequences for a given set of cargoes. In [7], Fagerholt and Christiansen investigate the problem to find an optimal visiting sequence for a ship with flexible compartments. They call the problem a travelling salesman problem with allocation, time window and precedence constraints and gave a dynamic programming algorithm. Instead of finding an optimal sequence, we use a simple heuristic to obtain a feasible sequence. For a given set of cargoes, the heuristic arranges the ports in ascending order of the loading/discharging day. It checks if the time windows constraint is satisfied at each port, and checks if there exists a feasible allocation of cargoes to the compartments. The feasibility of the allocation is checked by using small mixed-integer programs. If the sequence satisfies the time windows constraint and the allocation constraint, we obtain a feasible visiting sequence.

For each pairing, we calculate its latest arrival time at the first loading port. We can calculate it from the latest arrival time at each port in the pairing. We call it the latest arrival time of the pairing. If the ship arrives at the first loading port later than this time, the time windows constraint will not be satisfied at some ports in the pairing.

Let V denote the set of ships in the fleet indexed by v, K the set of cargoes indexed by k, P the set of pairings indexed by p. Moreover, let C_p denote a cost of using pairing p, F_k the cost for cargo k to be carried without any other cargoes. Let x_p be the binary variable that is equal to one if pairing p is used and zero otherwise, y_k a binary variable that is equal to one if cargo k is carried without any other cargo and zero otherwise. Let a_{kp} be a constant that is equal to one if pairing p includes cargo k, and zero otherwise. The pairing problem can be formulated as the following set partitioning problem:

$$\begin{aligned} \text{minimize} \ & \sum_{p \in P} C_p x_p + \sum_{k \in K} F_k y_k \\ \text{subject to} \ & \sum_{p \in P} a_{kp} x_p + y_k = 1 \quad \forall k \in K \\ & x_p \in \{0,1\} \qquad \forall p \in P \\ & y_k \in \{0,1\} \qquad \forall k \in K \ . \end{aligned}$$

The first constraint ensures that each cargo is included in a pairing or carried without any other cargoes.

As a solution of the pairing problem, we obtain a set of efficient pairings. Let P' be the set of pairings $p \in P$ with $x_p = 1$, and U be the set of cargoes $t \in T$ with $y_t = 1$. Let T be defined as $P' \cup U$. Let us call the set T a task set. And let us call a element of T a task. A task t can be carried by a ship, whereas it cannot be carried by another ship due to the differences of ship characteristics.

4.2 Routing Problem

As a solution of the pairing problem, we obtain a task set T. A sequence of tasks represents a schedule of a ship. Let us define a route as a sequence of tasks which can be carried by a ship satisfying the time windows constraint and the allocation constraint. The objective of the routing problem is to find a feasible route for each ship in the fleet, and to determine which cargoes are to be carried by spot charters, such that the total costs are minimized.

Let R_v be the set of feasible routes indexed by r, C_{rv} be the cost to use route r by vessel v, F_t the cost to carry task t by a spot charter, a_{tr} be the constant that is equal to one if task t is included in route r and zero otherwise. Let x_{rv} be binary variable that is equal to one if route r is assigned to ship v and zero otherwise. and y_t be a binary variable that is equal to one if task t is carried by a spot charter. The routing problem can be formulated as the following set partitioning problem:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{v \in V} \sum_{r \in R_v} C_{rv} x_{rv} + \sum_{t \in T} F_t y_t && (1) \\
\text{subject to} \quad & \sum_{v \in V} \sum_{r \in R_v} a_{tr} x_{rv} + y_t = 1 \quad \forall t \in T && (2) \\
& \sum_{r \in R_v} x_{rv} \leqq 1 \qquad \forall v \in V && (3) \\
& x_{rv} \in \{0,1\} \qquad \forall r \in R_v, \forall v \in V && (4) \\
& y_t \in \{0,1\} \qquad \forall t \in T \ . && (5)
\end{aligned}
$$

The first constraint ensures that each task is carried by a ship in the fleet or by a spot charter. The second constraint ensures that at most one route is assigned to each ship. The objective function minimizes the sum of the cost of operating all ships in the fleet and the costs of using spot charters.

In this formulation, an element of R_v $(v \in V)$ corresponds to a feasible route for ship v with respect to the time windows constraint and the allocation constraint, and it also corresponds to a column in the problem. We assume that each set R_v $(v \in V)$ is known in advance. However, the number of elements R_v may be too large to allow an explicit enumeration. Thus we use the column generation approach to implicitly enumerate feasible routes.

5 Column Generation for the Routing Problem

In this section, we introduce a column generation approach to solve the routing problem. In the routing problem (1)-(5), only the assignment constraints (2) are coupling the ships while the remaining constraints are dealing with each ship separately. This suggests the use of Danzig-Wolfe decomposition to break up the overall problem into a subproblem for each ship and a master problem [10].

In the routing problem, the problem is decomposed into a master problem with a promising collection of ship routes and a subproblem for each ship. The master problem is a set partitioning problem and the subproblem is the shortest path problem with time windows constraint and the allocation constraint. The restricted master problem is the LP relaxation problem of the following set partitioning problem:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{v \in V} \sum_{r \in \tilde{R}_v} C_{rv} x_{rv} + \sum_{t \in T} F_t y_t && (6) \\
\text{subject to} \quad & \sum_{v \in V} \sum_{r \in R_v} a_{tr} x_{rv} + y_t = 1 \quad \forall t \in T && (7)
\end{aligned}
$$

$$\sum_{r \in R_v} x_{rv} \leqq 1 \qquad \forall v \in V \tag{8}$$

$$x_{rv} \in \{0,1\} \qquad \forall r \in \tilde{R}_v, v \in V \tag{9}$$

$$y_p \in \{0,1\} \qquad \forall p \in P \ . \tag{10}$$

where $\tilde{R}_v$ is a subset of R_v. Each variable x_{rv} counts the number of times that route r by ship v is used. This is not necessarily integer because the integrality condition of x_{rv} is relaxed. The solution of this problem x_{rv} might be integer but this is not guaranteed. If it is integer, a feasible solution of the routing problem has been found, which is not necessarily optimal.

5.1 The Subproblem

The subproblem for each ship in the column generation process for the routing problem becomes the time-dependent shortest path problem with time windows constraint. In the mathematical description of the subproblem, there is an underlying network where the nodes correspond to the tasks and an arc represents the voyage between two tasks. A source s and a sink t are also defined on this network. The source s represents the initial state of the ship at the beginning of the planning horizon. The sink t represents the final state of the ship in the planning horizon. Let V denote the set of these nodes. If the ship can service the task j directly after servicing the task i, an arc (i, j) is defined from i to j. An arc is defined from s to each node at which the ship can arrive by its latest arrival time. Moreover, an arc is defined from each node in the network (except t) to t. Let E denote the set of these arcs. A travelling cost c_{ij} and a travelling time t_{ij} are associated with the arc (i, j). The cost c_{ij} is defined as a linear function of the travelling distance from the last discharging port of the task i and the first loading port of the task j. The time t_{ij} from i to j is defined as a sum of the travelling time from i to j and the process time of the task j. Here the process time of j is defined as a difference between the arrival time at the first loading port of j and the departure time from its last discharging port (Fig. 3). Then, a route of the ship is represented by an $s-t$ path in the network $G = (V, E)$. Moreover, the subproblem in the column generation process is to find a sequence from s through some of the nodes to t containing arrival times at the nodes.

In the formulation, the time t_{ij} depends on the departure time from i. The time that the ship stays idle in a port depends on its arrival time at the port. When a ship arrives at the port at 3:00 am, it stays idle at the port until 8:00 am in the next day. When a ship arrives at the port at 5:00 am, it stays idle until 8:00 am in the next day. In the former case, the waiting time is 5 hours, whereas it is 3 hours in the latter case. Therefore, the time t_{ij} is time-dependent.

The time-dependent shortest path problem with time windows can be solved by a label-correcting algorithm described in [11]. However, it is the pseudo-polynomial time algorithm and the number of labels can be exponential in the number of nodes. In order to solve the problem in a shorter amount of time, we

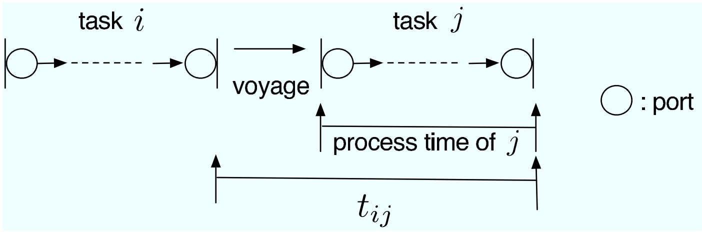

Fig. 3. definition of t_{ij} from i to j

reformulate the problem as the time-dependent shortest path problem (without the constraints) by discretizing the planning horizon.

Reformulation of the subproblem To solve the time-dependent shortest path problem with the time windows constraint, each node is duplicated into all possible times for leaving from that node. By discretizing the planning horizon and duplicating the nodes representing each task, sink and source, we get a time-space network. Let $\mathcal{T}$ be defined as

$$\mathcal{T} = \{\tau \in \mathbb{Z} \mid \tau_{\min} \leqq \tau \leqq \tau_{\max}\} \quad , \tag{11}$$

where $\tau_{\min}$ is the beginning of the planning horizon and $\tau_{\max}$ is the end of the planning horizon. Each of the new duplicated nodes is denoted by i_τ for task i and the discretized time τ. It represents the state that the ship leaves from the last discharging port of task i at τ. The set of nodes is defined by

$$\mathcal{V} = \{i_\tau \mid i \in V, \tau \in \mathcal{T}\} \quad . \tag{12}$$

Let c_{ij}^τ be the travelling cost from i to j when leaving from i at time τ. Let d_{ij}^τ denote the corresponding travelling time, where we assume that d_{ij}^τ is a positive integer and $\tau + d_{ij}^\tau \in \mathcal{T}$. When the ship leaves from i at time τ, it arrives at task j at time $\tau' = \tau + d_{ij}^\tau$. This voyage is represented by an arc $(i_\tau, j_{\tau'})$. That is, the set of arcs is defined by

$$\mathcal{E} = \left\{(i_\tau, j_{\tau'})\,|(i,j) \in E, \tau' \in \tau + d_{ij}^\tau, \tau, \tau' \in \mathcal{T}\right\} \quad . \tag{13}$$

The time-space network is defined by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. This time-space network for the subproblem is a nice structured acyclic network. The optimization problem can now be easily solved as a shortest path problem by a dynamic programming algorithm on this network.

We show an example of the time-dependent shortest path problem with time windows in Fig. 4. Two values by a node is the earliest arrival time and the latest arrival time. For simplicity, we assume that only the time from s to 1 is time-dependent, and the cost is 1 for all arcs. The time for an arc is shown by the arc. Let us assume that the period is discretized into $\{1, 2, 3, 4, 5, 6\}$. The time from s to 1 is $(t_{s1}^1, t_{s1}^2, t_{s1}^3, t_{s1}^4, t_{s1}^5) = (4, 4, 1, 1, 1)$. The shortest path in this network is $s \to 1 \to t$, in which the leaving time from s is 3, the leaving time from 1 is 4, and the arriving time at t is 5. The time-space network for this problem is shown in Fig. 5. The shortest path is shown in the heavy line.

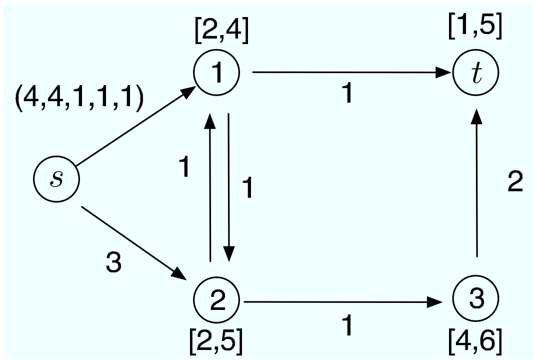

Fig. 4. time-dependent shortest path problem with time windows in $G = (V, E)$

5.2 Column Generation Process

We start the column generation process by solving the restricted master problem (6)-(10) in which $\tilde{R}_v$ is empty for all $v \in V$. Solving the restricted master problem yields a dual variable $\boldsymbol{\lambda}$ for the first constraint (7). Let λ_i denote the dual variable associated with the constraint for $i \in T$. Now the value of $\lambda_i (i \in T)$ are transferred to the subproblem. More specifically, the cost c_{ij}^{τ} in the network for the subproblem for each ship is substituted by the reduced cost $\tilde{c}_{ij}^{\tau}$ defined as

$$\tilde{c}_{ij}^{\tau} = c_{ij}^{\tau} - \lambda_i \qquad (i, j \in T \setminus \{s, t\}, \tau \in \mathcal{T}) \ . \tag{14}$$

Then we find the path with the minimal reduced cost by solving the subproblem for each ship. Note that solving the subproblem is an implicit enumeration of all feasible routes. This column generation process terminates when the optimal value of the subproblem for each ship is nonnegative.

The underlying network in the subproblem is acyclic and the cost of an arc can be negative. Thus, we use the Bellman-Ford algorithm to solve the subproblems efficiently [12].

When the column generation process terminates, we obtain the optimal value of the restricted master problem. The integrality condition of variable x_{rv} is relaxed in the restricted master problem so that the optimal solution is fractional in general. Therefore, it is necessary to obtain the integer solution by a branch-and-bound framework or some heuristics. Here, we use a heuristic, which imposes the integrality condition of x_{rv} and solve it with the routes $\tilde{R}_v$ generated so far.

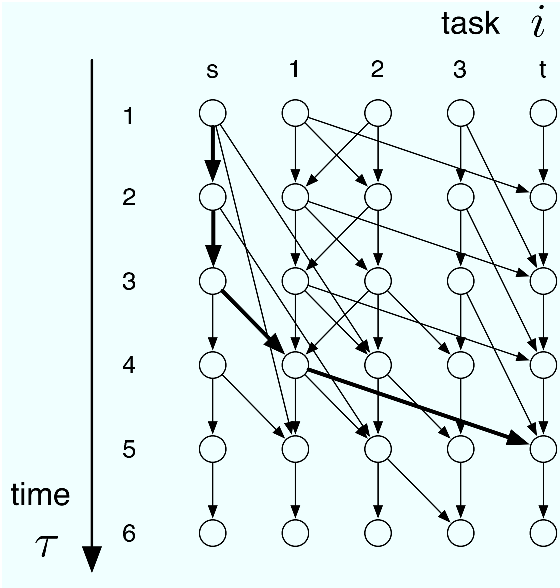

Fig. 5. time-space network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

6 Numerical Results

The proposed two-phase algorithm for the problem is tested on data from real problems considering sea transportation of oil-products in Japan. The numerical experiment was done on Intel Core 2 Duo 2.67GHz with 2GB memory. We used Xpress Optimizer Version 17.10.04 for solving the integer and linear programs [13].

The problem that we solved in case 1-3 consists of 128 cargoes, distributed on a five week planning period. It is the real planning problem that a Japanese shipping company faced. The fleet consists of 7 ships. The computational results are shown in Table 3. A figure in '#spot' column denotes the number of cargoes that are carried by spot charters. A figure in 'dist.' column denotes the total traveling distance by all ships. A figure in 'cpu' column denotes the total computational time (second) of the proposed algorithm. A figure in 'decr.' column denotes the decrease of the total traveling distance. The time unit for discretizing the planning horizon is one hour.

Case 1 The objective is to decrease the number of cargoes which are carried by spot charters. In the actual plan, which is generated by planners in the shipping company, the number of cargoes carried by spot charters is 10. The proposed algorithm found a plan in which the number of cargoes carried by spot charters is 9. The total travelling distance in the plan is nearly the same with the actual plan. The computational time in this case is 249 seconds, which is admissible in the practical planning environment.

Case 2 Here the objective is to decrease the total traveling distance under the condition that the number of cargoes carried by the spot charters is equal to or smaller than 10. The total travelling distance of the plan obtained by the proposed algorithm is smaller than that of the actual plan by 7.9 percent.

Case 3 Here the objective is to decrease the total travelling distance under the condition that the number of cargoes carried by the spot charter is equal to or smaller than 11. In this case, we evaluate how much the traveling distance decreases by increasing the number of spot charters. The total travelling distance of the plan obtained by the proposed algorithm is smaller than that of the actual plan by 8.0 percent. From this result, we found that increasing only one spot charter is not enough to decrease the travelling distance.

Table 3. numerical experiments for a fleet of 7 ships

case	# spot	dist.	cpu	decr.
actual plan	10	38973	–	–
case 1	9	38708	249	0.7%
case 2	10	36105	250	7.9%
case 3	11	35846	245	8.0%

The problems that we solved in case 4 and 5 are faced by another shipping company. The fleet consists of 14 ships. In the planning problem in this company, the cargo owners specifies the pairings. Thus, we only solve the routing problem. Moreover, the fleet can handle all cargoes in the planning period. The computational results are shown in Table 4.

Case 4 The problem consists of 63 cargoes, distributed on a two week planning period. Here the objective is to decrease the total travelling distance. The total traveling distance of the plan obtained by the proposed algorithm is smaller than that of the actual plan by 5.4 percent. The computational time in this case is 73 seconds.

Case 5 The problem consists of 91 cargoes, distributed on a three week planning period. Here the objective is to decrease the total travelling distance. The total traveling distance of the plan obtained by the proposed algorithm is smaller than that of the actual plan by 6.0 percent. The computational time is 275 seconds.

Table 4. numerical experiments for a fleet of 14 ships

case	dist.	cpu	decr.
actual plan for case 4	39945	–	–
case 4	37788	73	5.4 %
actual plan for case 5	36826	–	–
case 5	34422	275	6.0 %

7 Concluding Remarks

We have shown that the decomposition of the ship routing problem into two problems enables us to develop an algorithm for finding the solution efficiently. A column generation approach is presented for the routing problem, in which the subproblem is reformulated by discretizing the planning horizon. The obtained time-space network is a nice-structured acyclic network and the solution can easily be obtained. Numerical results shown in Sect. 6 demonstrate that the problem can be solved efficiently by the proposed algorithm.

References

1. Ronen, D.: Cargo ships routing and scheduling: Survey of models and problems. European Journal of Operational Research **12** (1983) 119–126
2. Ronen, D.: Ship scheduling: The last decade. European Journal of Operational Research **71**(3) (1993) 325–333
3. Christiansen, M., Fagerholt, K., Ronen, D.: Ship routing and scheduling : Status and perspectives. Transportation Science **38**(1) (2004) 1–18
4. Brown, G.G., Graves, W., Ronen, D.: Scheduling ocean transportation of crude oil. Management Science **33** (1987) 335–346
5. Desrosiers, J., Dumas, Y., Solomon, M.M., Soumis, F.: Time constrained routing and scheduling. In Ball, M.O., Magnanti, T.L., Monma, C.L., Nemhauser, G.L., eds.: Network Routing. Volume 8 of Handbooks in Operations Research and Management Science. Elsevier Science Publishers B.V., Amsterdam, The Netherlands (1995) 35–139
6. Fagerholt, K., Christiansen, M.: A combined ship scheduling and allocation problem. Journal of the Operations Research Society **51** (2000) 834–842
7. Fagerholt, K., Christiansen, M.: A travelling salesman problem with allocation, time window and precedence constraints – an application to ship scheduling. International Transaction in Operational Research **7** (2000) 231–244
8. Appelgren, L.H.: A column generation algorithm for a ship scheduling problem. Transportation Science **3** (1969) 53–68
9. Appelgren, L.H.: Integer programming methods for a vessel scheduling problem. Transportation Science **5** (1971) 64–78
10. Desauliniers, G., Desrosiers, J., Solomon, M.M., eds.: Column Generation. GERAD 25th Anniversary Series. Springer-Verlag (2005)
11. Desrochers, M., Soumis, F.: A generalized permanent labelling algorithm for the shortest path problem with time windows. IFOR **26** (1988) 191–212
12. Corman, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. MIT Press (2001)

13. Dash Optimization: The Xpress-Optimizer. http://www.dashoptimization.com/.