Computers and Chemical Engineering 72 (2015) 350–362

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

A MILP-based column generation strategy for managing large-scale maritime distribution problems

Mariana E. Cóccola, Rodolfo Dondo*, Carlos A. Méndez

INTEC (UNL-CONICET), Güemes 3450, 3000 Santa Fe, Argentina

ARTICLE INFO

Article history:
Received 20 January 2014
Received in revised form 10 April 2014
Accepted 18 April 2014
Available online 28 April 2014

Keywords:
Tramp shipping
Ship routing and scheduling
Column generation
MILP-slave-formulation

ABSTRACT

This paper presents a novel column generation algorithm for managing the logistics activities performed by a fleet of multi-parcel chemical tankers. In our procedure, for providing elementary routes, the conventional dynamic programming routes-generator is replaced by an efficient continuous-time MILP-slave problem. The performance of the decomposition method is evaluated by solving several examples dealing with the operations of a shipping company operating in the Asia Pacific Region. Computational results show that the proposed approach outperforms a pure exact optimization model and an alternative heuristic solution method reported in the literature.

1. Introduction

Nowadays, supply chain management and optimization is a critical aspect of modern enterprises and a very active research area (Papageorgiou, 2009). In the current context of globalized economic activities, the maritime transport constitutes an increasingly important supply chain link. The UNCTAD's Review of Maritime Transport (2012) details that around 80 per cent of global trade by volume and over 70 per cent by value is carried by sea and handled by ports worldwide. The current trends in the shipping industry are moving toward a cost-efficient management of the fleet capacities. Main activities of the ships fleet involve the transportation of cargos between several ports, which are geographically distributed in many different cities, countries and/or continents. The cost-effective routing and scheduling of the fleet represents a central decision making process in both the chemical and the shipping industry. According to Christiansen et al. (2007), the daily operating cost of a ship may easily amount to thousands of dollars. Consequently, a proper planning of routes and schedules can significantly improve the economic performance of shipping companies by reducing shipping costs.

From the operational perspective, ship routing and scheduling problems are closely related to pickup and delivery tasks performed by trucks in the classical pickup and delivery (PDP) problem. Surveys on the pickup and delivery problem can be found in Savelsbergh and Sol (1995) and Desaulniers et al. (2002). However, there are some important differences between both transportation modes that must be considered. One of them is that no central port from where ships start and end voyages is defined.

During last years, extensive research has been done on ship scheduling. A comprehensive survey on maritime transportation problems and their proposed solution strategies can be found in Christiansen et al. (2013). The paper remarks that a solution technique that has been successfully applied in tramp shipping problems is the column generation approach (CG). The CG paradigm gained popularity as an efficient technique to solve large optimization problems in transportation and logistics. This technique was first introduced by Dantzig and Wolfe (1960) to solve linear problems with decomposable structures. During the 1990s, it has been successfully extended to a wide range of integer and integer-linear problems and become the leading optimization technique for solving many routing problems. Very large and complex problems were successfully solved at present days with this technique. See e.g. Ropke and Cordeau (2009) and Bettinelli et al. (2011).

The CG approach decomposes the original problem into two subproblems: a restricted master problem (RMP), which selects the best route for each ship and a slave route generator problem. A priori generation of all feasible columns has been the most common method used to solve small instances of routing problems. However, generating all columns for large cases may be intractable, and consequently, the use of dynamic programming algorithms has been usually considered. Brønmo et al. (2007) used a set partitioning problem, where a priori generation of all feasible columns is applied to solve a ship scheduling problem with flexible sizes.

* Corresponding author. Tel.: +54 342 4559175.
E-mail address: rdondo@santafe-conicet.gov.ar (R. Dondo).

http://dx.doi.org/10.1016/j.compchemeng.2014.04.008
0098-1354/

Nomenclature

Subscripts
i, i' cargos
r columns
p, p' ports
s ships

Sets
I cargos
DP_p cargos to be discharged at port p
IP_s first port to visit by ship s
LP_p cargos to be loaded at port p
OB_s subset of cargos that are on-board ship s at beginning of planning horizon
R_s subset of feasible columns for ship s
P ports
S ships

Parameters
$dist_{pp'}$ nautical miles between ports p and p'
dr_i discharge rate for cargo i (tonnes/day)
ept_i earliest pickup service time for cargo i
fc_s cost of fuel per unit distance for ship s (in USD/nm)
K_s maximum number of ports that ship $s$can visit during the planning horizon
lpt_i latest pickup service time for cargo i
lr_i loading rate for cargo i (tonnes/day)
M_c maximum ship capacity
M_d maximum ship traveling distance
M_t maximum ship traveling time
pc_{ps} ship-dependent port cost (in USD)
p_r^s maximum profit obtained by s by transporting all cargos in r
sr_i shipping rate or revenue for cargo i (in USD)
tad inspection time
tcc_s time-charter cost per unit time for ship s (in USD/day)
ti_s arrival time of ship s to the first port visited
v_s average speed (knots) of ship s
$vmax_s$ ship capacity (in tonnes)
$volume_i$ size of cargo i (in tonnes)
a_{ir}^s binary parameter which values 1 if cargo i is included in column $r \in R_s$

Binary variables
$PR_{pp's}$ denoting that port p is visited before ($PR_{p,p's} = 1$) or after ($PR_{p,p's} = 0$) port p' whenever both nodes are serviced by the same ship s
X_{ps} denoting the assignment of port p to ship s
X_r^s determining that cargo combination r is served by s
Y_{is} denoting the assignment of cargo i to ship s

Continuous variables
$LOAD_{ps}$ total cargo loaded on ship s after completing the service at port p
TD_{ps} accumulated ship travel distance to reach port p
TV_{ps} accumulated ship travel time to reach port p
TTD_s total travel distance for ship s
TTV_s total travel time for ship s
$UNLOAD_{ps}$ total cargo unloaded from ship s after completing the service at port p

A similar problem was studied by Brønmo et al. (2010) but, in this case, a dynamic CG approach was used for solving more complex instances. Nishi and Izuno (2014) proposed some CG heuristics to solve a ship routing and scheduling problem for crude oil transportation with split deliveries.

This paper introduces a novel CG approach by replacing the dynamic programming generator by an efficient continuous-time MILP-slave problem. Moreover, the branch-and-price technique is also tested. The branch-and-price is a generalization of the classical branch-and-bound scheme in which upper bounds (for maximization problems) are computed by column generation (Barnhart et al., 2000). Our works aims to find the optimal routes and schedules for a heterogeneous fleet of multi-parcel chemical tankers in order to carry multiple cargos at maximum profit while satisfying additional constraints as pickup time windows and the ship carrying capacity constraint. The best cargo combination for each ship is chosen by the master problem. The slave problem generates multiple columns by solving a continuous time precedence-based MILP formulation that takes into consideration most of problem constraints. In order to generate the maximum number of feasible cargos set per iteration, the solver-options of the branch-and-cut package used for solving the slave problem are properly tuned.

The performance of our proposed approach in terms of solution quality and CPU time has been tested on five cases based on real data from a chemical shipping company that operates a fleet of heterogeneous ships in the Asia Pacific Region. A real-world example, initially introduced by Jetlund and Karimi (2004), involves 10 tankers, 79 cargos, and 36 ports, was first solved. Four larger instances were also successfully solved. Computational results show that the CG strategy outperforms the original MILP mathematical model, which is here adapted as slave problem of the decomposition algorithm. Moreover, the solution quality rises with respect to solutions already reported in the literature.

2. Tramp shipping characteristics

Logistics activities commonly arising in tramp shipping usually involve the movement of multiple cargos between ports. The aim is to maximize the net profit through the cost-efficient routing and scheduling of ship operations. This problem can be taken as a generalization of the road based distribution problems presented by Dondo et al. (2008). The main problem features are:

- The company has a heterogeneous fleet of multi-parcel chemical tankers. The tankers are used to carry liquid bulk commodities such as oil products, chemicals, and other liquids.
- A tramp ship company usually aims to maximize its profit. The total profit is determined by the revenues earned for transporting cargos minus operation costs. The costs for operating a ship are dependent upon its size.
- Tramp ships do not have fixed routes and sailing times as in liner shipping. The ships visit seaports depending up on the cargo availability.
- Each cargo is transported from its pickup location to its delivery location by a single ship. Partial shipments are not allowed. Moreover, not all cargos may be transported. A cargo that is not moved can be outsourced to a third party carrier in the spot market (Jetlund and Karimi, 2004).
- Pickup time windows are usually defined by the cargo owner.
- Every ship can carry multiple cargos at a time. However, the ship carrying capacity stands for a finite load capacity in tonnes that cannot be exceeded.
- No central port, from where voyages start and end, is defined. The ships have usually different start locations which can be either a port or a point at sea. The starting time for each ship schedule

can be different as well. Ship schedules end at the port where the ship delivers the last cargo.
- Generally, the scheduler determines a limited number of ports to visit during the planning horizon. Each port can be visited by a ship just once.
- A port visit time-length has two components: a fixed inspection time and a variable time that is proportional to the amount of tonnes to be picked-up and/or discharged.

3. Major problem variables

Shipping is performed on a known harbors-network infrastructure. The maritime routes are determined by a set of minimum-distance arcs which interconnects ports $p \in P$ in the network. Let the set I be the potential cargos to transport during the planning horizon and let the set S the heterogeneous fleet of multi-parcel chemical tankers. The subsets LP_p and DP_p include all cargos i having their pickup and discharge location at port p, respectively.

In the mathematical representation, three different 0-1 decision variables are used: (i) the assignment variable Y_{is} computing if cargo i is selected to be carried by ships, (ii) the assignment variable X_{ps} taking 1 as value if the port p is visited by s during the planning horizon, and (iii) the general sequencing variable $PR_{pp's}$ determining that port p is visited before ($PR_{pp's} = 1$) o after ($PR_{pp's} = 0$) port p' whenever both are located on the voyage of ship s (i.e. $X_{ps} + X_{p's} = 2$). The variable $PR_{pp's}$ is just defined for $p < p'$, i.e. the relative position of port p is less than that one of port p' in the set P.

In order to determine a detailed schedule for each ship, a set of continuous positive variables has been also defined. They are: (a) the arrival time at port p by ship s (TV_{ps}); (b) the accumulated sailing distance up to port p (TD_{ps}); (c) the overall schedule duration (TTV_s); (d) the overall sailing distance along the voyage of ship s (TTD_s). In addition, the non-negative continuous variables $LOAD_{ps}$ and $UNLOAD_{ps}$ indicate the accumulated amount of tonnes loaded and discharged by ship s, including initial cargos, after visit port p.

Cargos properties, ship capacities, port locations, and fixed-operational costs are all known with certainty and remain invariant over time. The major problem data are:

- The distance between ports ($dist_{pp'}$), which is always expressed in nautical miles.
- The tonnes of cargo i ($volume_i$) and its pickup and discharge rate (lr_i and dr_i).
- The time window within which the pickup of cargo i must start [ept_i, lpt_i].
- The fixed inspection time of each ship at every port (tad).
- The total carrying capacity of ship s ($vmax_s$).
- The average ship speed expressing in knots (v_s).
- The revenues in USD obtained by serving cargo i (sr_i).
- The operation costs depending upon the size of ship s: fuel cost (fc_s), time-charter cost (tc_s), and port charges (pc_{ps}).

4. The MILP-based column generation algorithm

In this section, the CG method is adapted to the tramp ship routing and scheduling problem. With this technique, the routing problem is decomposed into a master problem, which aims to select a set of cargos or column for each ship, and a sub-problem where promising ship schedules for each ship are found.

4.1. The master problem

Let the heterogeneous fleet of multi-parcel chemical tankers be $S = \{1, 2, \ldots, s\}$ and let $R_s = \{1, 2, \ldots, r\}$ be the set of all feasible cargo sets for ship s. This means that s is capable of generating a feasible schedule, with regards to the time windows, the starting point and the ship carrying capacity, in which all cargos in r can be transported. In this paper the terms cargo combination, cargo set and column will be used interchangeably. Let a^s_{ir} be a binary parameter which values 1 if cargo i is included in column $r \in R_s$ and 0 otherwise. We can then formulate the ship routing and scheduling problem as a set partitioning problem (SPP or master problem) as follows:

$$\max \sum_{s \in S} \sum_{r \in R_s} p^s_r x^s_r \tag{1}$$

$$s.t. \quad \sum_{s \in S} \sum_{r \in R_s} \alpha^s_{ir} x^s_r \le 1 \quad \forall i \in I \tag{2}$$

$$\sum_{r \in R_s} x^s_r = 1 \quad \forall s \in S \tag{3}$$

$$x^s_r \in \{0, 1\} \quad \forall s \in S, r \in R_s \tag{4}$$

According to Eq. (1), the objective of the SPP is to select the best cargo set for each ship so that the profit is maximized. The binary variable x^s_r takes 1 as value if the cargo combination $r \in R_s$ is served by s whereas the maximum profit obtained by transporting those cargos is defined as p^s_r. Eq. (2) states that each cargo can, at most, be served by a single ship, while Eq. (3) determines that just one column/cargo set is selected for each ship s. The formulation represents the set of all feasible routes for each ship and its objective is to select the maximum-profit subset of routes.

At the beginning of planning horizon every ship s has on-board a set of contracted cargos $i \in OB_s$ to deliver. Thus, a feasible cargo combination, i.e. a column, is generated a priori for each ship. The formulation (1)–(4) is called the reduced master problem (RMP) because it contains a subset of the set of all feasible columns. When, the linear relaxation of the master problem (LP-master or relaxed RMP) is solved considering just this partial set of columns, the dual variables of Eqs. (2) and (3) are introduced into the objective function of the sub-problem in order to collect new columns with positive reduced costs that can increase the value of the objective function in the master problem. This cyclic procedure is solved until no a column with positive reduced costs exists. When this happens, the current LP-optimal solution cannot be improved further. Here, if the optimal solution of the LP-master is integer, it is also the optimal solution to the RMP and to the ship routing and scheduling problem. Otherwise, a branch-and-bound method is used to close the gap between the upper bound (the result of the relaxation) and the integer solution. This method, in which the column generation technique is applied to determine the upper bound in each node of the branching tree, is denoted as branch-and-price. All steps of the procedure are sketched in Fig. 1.

4.2. The MILP-based subproblem

The sub-problem can be described as the problem of finding the optimal schedule for a given ship under the given restricted master dual variable values (Brønmo et al., 2010). Let π_i the dual variables of constraint (2) and let σ_s be the dual variable values for constraint (3). The objective function, which is defined for each ship, can be formulated then as follows:

$$Max \left[\sum_{i \in I} (sr_i - \pi_i) Y_{is} - tcc_s TTV_s - fc_s TTD_s - \sum_{p \in P} pc_{ps} X_{ps} - \sigma_s \right] \quad \forall s \in S \tag{5}$$

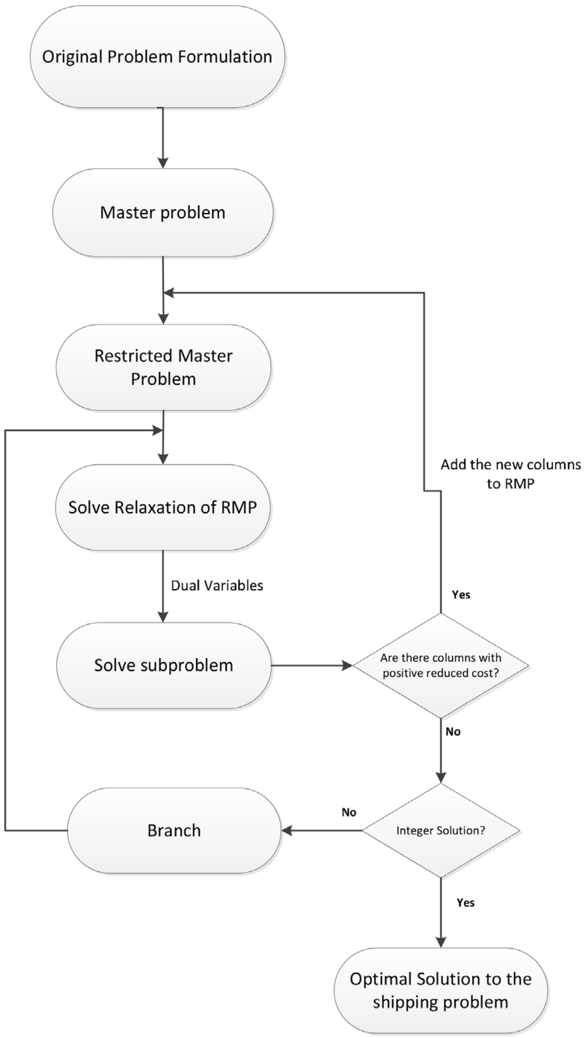

Fig. 1. Branch-and-price procedure for the tank shipping problem.

For each ship s, the goal is to find the feasible schedule with maximum reduced cost respect to the current dual values. Parameter sr_i stands for the shipping rate for cargo i while the costs for operating a ship s include: (i) tcc_s referring to the time-charter cost, (ii) fc_s denoting the fuel cost per unit distance, and (iii) pc_{ps} determining the fixed cost that ship s pays when it visits port p.

In order to incorporate feasible cargo combinations to the pool of columns available for a ship s, a continuous time precedence-based MILP representation is reformulated as the slave problem. In every master-slave iteration, the MILP model is executed $|S|$ times, once per each ship in the fleet. The set of constraints considered in the sub-problem are detailed below.

Eq. (6) enforces the condition that a cargo i can be serviced by ship s ($Y_{is} = 1$) only if the ports, where the cargo must be loaded and discharged, are both visited by s. Binary variable X_{ps} values 1 if there is a visiting of ship s to port p. For the cargos on-board at beginning of planning horizon (OB_s), just its discharge port must be assigned to s.

$$Y_{is} \le X_{ps} \quad \forall i \in I, p \in P : ((i \notin OB_s) \cap (i \in LP_p)) \cup (i \in DP_p) \tag{6}$$

The traveling-distance constraints are stated by Eqs. (7)–(10). Constraint (7) computes the minimum distance to reach the first port p visited by s ($p \in IP_s$). Parameter ti_s defines the estimated arrival time of ship s to port p while the average speed in knots is given by parameter v_s. To calculate the accumulated traveling distance from time zero up to every ship stop, sequencing variables $PR_{pp's}$ are defined to determine the order in which such a pair of ports is visited. Parameter $dist_{pp'}$ sets the distance in nautical miles between two ports (p, p'). M_d is a large positive number. Finally, the overall distance sailed by ship s during the planning horizon is computed by Eq. (10).

$$TD_{ps} = ti_s 24 v_s \quad \forall p \in P : p \in IP_s \tag{7}$$

$$TD_{p's} \ge TD_{ps} + dist_{pp'} - M_d(1 - PR_{pp's}) - M_d(2 - X_{ps} - X_{p's}) \quad \forall (p, p') \in P : p < p' \tag{8}$$

$$TD_{ps} \ge TD_{p's} + dist_{p'p} - M_d PR_{pp's} - M_d(2 - X_{ps} - X_{p's}) \quad \forall (p, p') \in P : p < p' \tag{9}$$

$$TTD_s \ge TD_{ps} \quad \forall p \in P \tag{10}$$

Similarly to above traveling distance constraints, the timing ones are stated by Eqs. (11)–(14). The length of a ship stop in each port comprises a fixed inspection time (tad) plus a variable time proportional to the amount of tonnes to pick-up or deliver by the ship. For each cargo i, the time for loading or discharging is determined by: (i) the volume in tonnes ($volume_i$), (ii) the loading rate (lr_i), and (iii) the discharge rate (dr_i). M_t is an upper bound for the corresponding time variable.

$$TV_{ps} = ti_s \quad \forall p \in P : p \in IP_s \tag{11}$$

$$TV_{p's} \ge TV_{ps} + \sum_{i \in I : (i \notin OB_s) \cap (i \in LP_p)} \frac{Y_{is} volume_i}{lr_i} + \sum_{i \in I : i \in DP_p} \frac{Y_{is} volume_i}{dr_i} + tad + \frac{dist_{pp'}}{24 v_s} - M_t(1 - PR_{pp's}) - M_t(2 - X_{ps} - X_{p's}) \quad \forall (p, p') \in P : p < p' \tag{12}$$

$$TV_{ps} \ge TV_{p's} + \sum_{i \in I : (i \notin OB_s) \cap (i \in LP_{p'})} \frac{Y_{is} volume_i}{lr_i} + \sum_{i \in I : i \in DP_{p'}} \frac{Y_{is} volume_i}{dr_i} + tad + \frac{dist_{p'p}}{24 v_s} - M_t PR_{pp's} - M_t(2 - X_{ps} - X_{p's}) \quad \forall (p, p') \in P : p < p' \tag{13}$$

$$TTV_s \ge TV_{ps} + \sum_{i \in I : i \in DP_p} \frac{Y_{is} volume_i}{dr_i} + tad \quad \forall p \in P, s \in S \tag{14}$$

In case of port $p \in IP_s$, the pair of Eqs. (12) and (13) can be replaced by a single one by considering that the visiting to any other port $p'(X_{p's} = 1)$ occurs always after the visiting to p.

$$TV_{p's} \ge TV_{ps} + \sum_{i \in I : (i \notin OB_s) \cap (i \in LP_p)} \frac{Y_{is} volume_i}{lr_i} + \sum_{i \in I : i \in DP_p} \frac{Y_{is} volume_i}{dr_i} + tad + \frac{dist_{pp'}}{24 v_s} - M_t(1 - X_{p's}) \quad \forall (p, p') \in P : (p \ne p') \cap (p \in IP_s) \tag{15}$$

Eq. (16) enforces the condition that the stop at port p' $(i \in DP_{p'})$ will occur after the stop at port $p(i \in LP_p)$ if the ship s serves a cargo i.

$$\begin{aligned} TV_{p's} \geq TV_{ps} &+ \sum_{i' \in I:(i' \notin OB_s)\cap(i' \in LP_p)\cap(i' \neq i)} \frac{Y_{i's} volume_{i'}}{lr_{i'}} \\ &+ \sum_{i' \in I:(i' \in DP_p)\cap(i' \neq i)} \frac{Y_{i's} volume_{i'}}{dr'_i} + tad + \frac{dist_{pp'}}{24 v_s} - M_t(1 - Y_{is}) \\ &\forall i \in I, (p, p') \in P : (i \notin OB_s) \cap (i \in LP_p) \cap (i \in DP_{p'}) \end{aligned} \tag{16}$$

The pickup of a cargo i should start within the specific time windows $[ept_i, lpt_i]$. This condition is enforced by Eqs. (17)–(19).

$$TV_{ps} \leq lpt_i - 0.5 tad + M_t(1 - Y_{is}) \quad \forall i \in I, p \in P : (i \notin OB_s) \cap (i \in LP_p) \tag{17}$$

$$\begin{aligned} TV_{p's} \geq ept_i &+ 0.5 tad + \frac{dist_{pp'}}{24 v_s} + \frac{Y_{is} volume_i}{lr_i} - M_t(1 - PR_{pp's}) \\ &- M_t(2 - X_{ps} - X_{p's}) - M_t(1 - Y_{is}) \\ &\forall (p, p') \in P, i \in I : (p < p') \cap (i \notin OB_s) \cap (i \in LP_p) \end{aligned} \tag{18}$$

$$\begin{aligned} TV_{ps} \geq ept_i &+ 0.5 tad + \frac{dist_{p'p}}{24 v_s} + \frac{Y_{is} volume_i}{lr_i} - M_t PR_{pp's} \\ &- M_t(2 - X_{ps} - X_{p's}) - M_t(1 - Y_{is}) \\ &\forall (p, p') \in P, i \in I : (p < p') \cap (i \notin OB_s) \cap (i \in LP_{p'}) \end{aligned} \tag{19}$$

Eq. (20) determines that the number of visited ports by a ship must never exceed a maximum amount K defined for the scheduler planner.

$$\sum_{p \in P} X_{ps} \leq k_s \quad \forall s \in S \tag{20}$$

The accumulated amount of tonnes loaded by a ship s from the beginning of planning horizon up to leave port p, including cargos on-board at time zero, is computed by Eqs. (21)–(23).

$$\begin{aligned} LOAD_{ps} \geq \sum_{i \in I: i \in OB_s} Y_{is} volume_i + \sum_{i \in I:(i \notin OB_s)\cap(i \in LP_p)} Y_{is} volume_i \\ \forall p \in P : p \in IP_s \end{aligned} \tag{21}$$

$$\begin{aligned} LOAD_{p's} \geq LOAD_{ps} &+ \sum_{i \in I:(i \notin OB_s)\cap(i \in LP_{p'})} (Y_{is} volume_i) \\ &- M_c(1 - PR_{pp's}) - M_c(2 - X_{ps} - X_{p's}) \quad \forall (p, p') \in P : p < p' \end{aligned} \tag{22}$$

$$\begin{aligned} LOAD_{ps} \geq LOAD_{p's} &+ \sum_{i \in I:(i \notin OB_s)\cap(i \in LP_p)} (Y_{is} volume_i) \\ &- M_c PR_{p'ps} - M_c(2 - X_{ps} - X_{p's}) \quad \forall (p, p') \in P : p < p' \end{aligned} \tag{23}$$

Similarly, Eqs. (24)–(26) determined the tonnes discharged from ship s after visiting port p.

$$UNLOAD_{ps} \geq \sum_{i \in DP_p} Y_{is} volume_i \quad \forall p \in P : p \in IP_s \tag{24}$$

$$\begin{aligned} UNLOAD_{p's} \geq UNLOAD_{ps} &+ \sum_{i \in DP_{p'}} (Y_{is} volume_i) \\ &- M_c(1 - PR_{pp's}) - M_c(2 - X_{ps} - X_{p's}) \quad \forall (p, p') \in P : p < p' \end{aligned} \tag{25}$$

$$\begin{aligned} UNLOAD_{ps} \geq UNLOAD_{p's} &+ \sum_{i \in DP_p} (Y_{is} volume_i) \\ &- M_c PR_{pp's} - M_c(2 - X_{ps} - X_{p's}) \quad \forall (p, p') \in P : p < p' \end{aligned} \tag{26}$$

The ship carrying capacity stands for a finite load capacity in tonnes $(vmax_s)$ that cannot be exceeded. As the difference $(UNLOAD_{ps} - LOAD_{ps})$ computes the current load transported by s after stop at p, Eqs. (27) and (28) forbid ship overcapacity or negative tonnes, respectively.

$$LOAD_{ps} - UNLOAD_{ps} \leq vmax_s X_{ps} \quad \forall p \in P_s \tag{27}$$

$$LOAD_{ps} - UNLOAD_{ps} \geq 0 \quad \forall p \in P_s \tag{28}$$

Eq. (29) enforces the condition that the total cargo loaded on ship s after visiting a port p can never be larger than the total cargos collected by ship s along the voyage plus the initial cargos. Similarly, Eq. (30) states that the total cargo unloaded from s after visiting port p can never be larger than the total amount discharged along the whole voyage. Both constraints compute upper bounds on the values of the variables $LOAD_{ps}$ and $UNLOAD_{ps}$, respectively.

$$LOAD_{ps} - \sum_{i \in I} Y_{is} volume_i \leq vmax_s(1 - X_{ps}) \quad \forall p \in P \tag{29}$$

$$UNLOAD_{ps} - \sum_{i \in I} Y_{is} volume_i \leq vmax_s(1 - X_{ps}) \quad \forall p \in P \tag{30}$$

4.3. The branching strategy

Whenever the column generation procedure ends, we must check out the integrality of the solution of the LP-master. If this solution is integer, it is also the optimal solution to the shipping problem; otherwise, the solution is fractional. In that case a branch-and-price method, in which a column generation procedure is applied to each node of a branch-and-bound tree, is used to close the gap between the upper bound (the result of the relaxation) and the current integer solution (the result of the restricted master problem). The scanning of the tree aims at determining if exists a column that would price out favorably, but is not present in the master problem.

In a branch-and-price algorithm, for maximization problems, the column generation technique is applied to determine the local upper bound (LUB) in each node of the branching tree. If that is integer, the LUB must be compared with the current global lower bound (GLB), which represents the best integer solution found so far. If the LUB is not integer, and its value is lower than GLB then the node is fathomed. Otherwise, the current node is divided into two new child nodes by a branching rule. The best-first strategy is used to chosen the next subspace to be explored, which it will be the one that search for the highest upper bound. Many branching schemes for column generation approaches have been proposed in the literature However, which rule to use is a decision related to the problem we are trying to solve. In our algorithm, we have proposed a branching strategy that focuses on assignments decisions Y_{is}. According to Savelsbergh and Sol (1998), assignment decisions are high-level decisions that have a big impact on the structure of the solution. Suppose that the LP-master returns a fractional solution, this means that some variable x_r^s in the master problem takes a fractional value $(0 < x_r^s < 1)$ and that such column has at least one

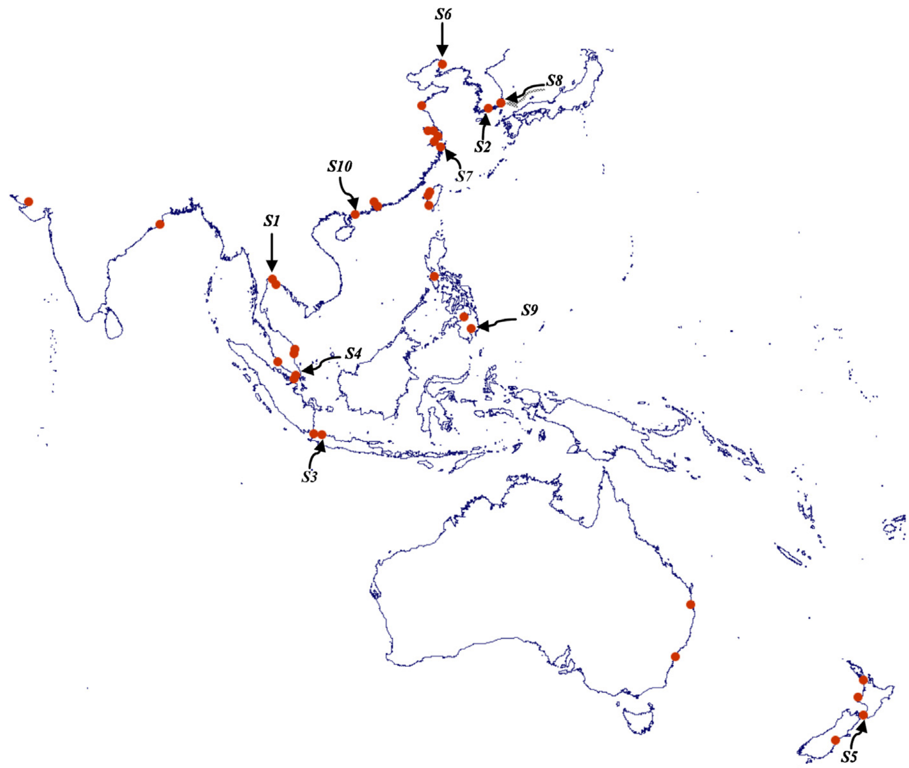

Fig. 2. Ports located in the Asia Pacific Region.

a_{ir}^{s} equal to one. If this happens, we select the column and create two nodes: the left node, in which the variable Y_{is} is set to zero, and the right node, in which Y_{is} is fixed to one. This branching rule can be viewed as a generalization of the one proposed by Ryan and Foster (1981) for set partitioning problems. Those authors proposed to select two customers $i1$ and $i2$ and generate two branch-and-bound nodes: one in which $i1$ and $i2$ are serviced by the same vehicle and one where they are serviced by different vehicles. The pseudo-code of our branch-and-price strategy, which was codified in GAMS, is described in Section 4.4. Note that at the first step, one column, which just includes the cargos on-board at beginning of planning horizon, is created for each heterogeneous ship in the fleet to ensure the feasibility of the first master problem execution. It is worth to mentioning that none of those columns should be selected to branching because the condition $Y_{is} = 0$ will lead to an infeasibility. The subset *exception*(r,s) is used to eliminate infeasible columns from the master problem after executing the branching rule while the subsets *fix_to_zero*(*node, cargo, ship*), *fix_to_one*(*node, cargo, ship*), and *fix*(*node, cargo*) are defined for tracking the value of the variables that have been fixed along the branching tree. To generate alternatives schedules with positive reduced cost per iteration, the option of the CPLEX MILP solver, *solnpool*, is used for storing multiple solutions to the MILP problem in the solution pool.

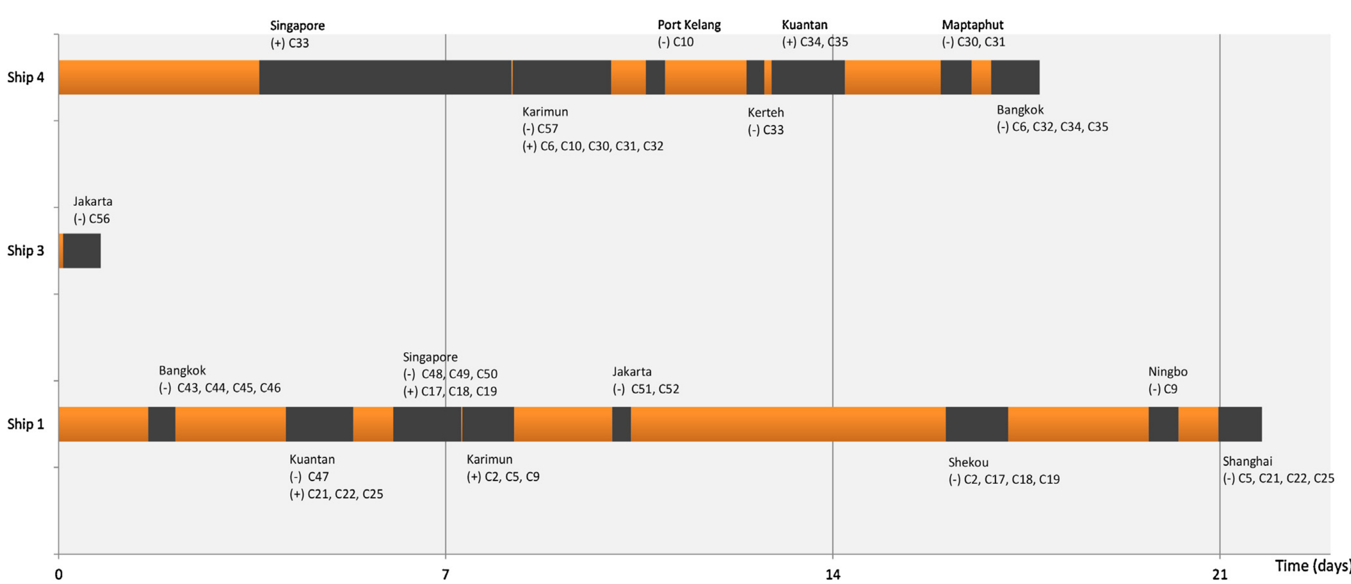

Fig. 3. Optimal schedules for Example 1A.

4.4. Pseudo-code of the branch-and-price algorithm

```
SOLVEthe LP-master problem with the initial pool of columns
WHILEthere are new columns with positive reduced costDO
    LOOP s
        SOLVEsubproblem (with option solnpool);
        Add the columns with positive reduced cost to the Master Problem
    END LOOP
END WHILE
GUB:= solution of the LP-master;GLB:= solution of the IP-master;
IF (GUB>GLB)THEN
    Store root node n0 in the waiting node list
    LUB(n0):=GUB;
    set fix_to_zero(n0, i, s):=no for all (i, s)
    set fix_to_one(n0, i, s):=no  for all (i, s)
    set fix (n0, i):=no for all i
    WHILE waiting node list is not emptyDO
        Choose a node sn from the waiting node list and remove it from the list
        Create left children lnandright children rn and store themin waiting node list
        LOOP children (ln or rn)
            Set fix_to_zero(ln or rn, i, s):= fix_to_zero(sn, i, s) for all (i, s)
            Set fix_to_one(ln or rn, i, s):= fix_to_one(sn, i, s) for all (i, s)
            Set fix(ln or rn, i):= fix(sn, i) for all i
        END LOOP
        Find a column (r,s) with fractional value in $x_r^s$ ; Select a cargo i', where fix (sn, i')=noand$\alpha_{ir}^s$ =1
        Set fix (ln,i'):= yes; Set fix_to_zero(ln, i, s):=yes;
        Set fix (rn,i'):= yes; Set fix_to_one(rn, i, s):=yes;
        LOOP children (ln or rn)
            Setexception(r,s)=no
            LOOP (i,r,s)
                IF fix_to_zero(ln or rn, i, s) and $\alpha_{ir}^s$ =1 THEN
                    exception(r,s)=yes;
                ELSE IFfix_to_one(ln or rn, i, s) and $\alpha_{ir}^s$ =0 THEN
                    exception(r,s)=yes;
                END IF
            END LOOP
            SOLVE LP-master problem
            IF infeasible THEN
                waiting(ln or rn)=no  // Node is fathomed
            ELSE
                LOOP (i,s)
                    Release variables Y(i,s)
                    IF fix_to_zero(ln or rn, i, s) THEN
                        FixY(i,s)=0
                    ELSE IF fix_to_one(ln or rn, i, s) THEN
                        FixY(i,s)=1
                    END IF;
                END LOOP;
                WHILEthere are new columns with positive reduced costDO
                    LOOP s
                        SOLVE subproblem (with option solnpool);
                        Add the columns with positive reduced cost to the
            Master Problem
                    END LOOP
                END WHILE
                LUB (ln or rn)=solution of the LP-master Problem;
                IF LUB(ln or rn) is integer THEN
                    IF LUB(ln or rn)>GLB THEN
                        GLB:=LUB(ln or rn)
                        Remove from the waiting list all nodes with LUB<GLB
                ELSE
                        waiting(ln or rn)=no  // Node is fathomed
                    END IF
                ELSE
                    IF LUB(ln or rn) < GLB THEN
                        waiting(ln or rn)=no  // Node is fathomed;
                END IF
                END IF
            END IF
        END LOOP
    END WHILE
END IF
```

Table 1
Setting options of the branch-and-price algorithm.

MILP solver	CPLEX 12.2
Branching rule	On assignment variables Y_{is}
Nodes selection strategy	Best first search
Maximum CPU time per slave execution	No limit for Example 1A, 1B, and 2 10 s for Example 3 20 s for Example 4
Multiple columns generated per iteration	Yes (option solnpool of CPLEX)
Maximum number of master-slave iterations per branch and price node	20 (root)/5 (no-root)

5. Computational results and discussion

The performance of the proposed strategy has been tested by solving five complex examples all dealing with the distribution activities of a multi-national shipping company operating a fleet of multi-parcel chemical tankers in the Asia Pacific Region. Such examples are modified instances of a real-world case study previously tackled by Jetlund and Karimi (2004). The case study involve 36 ports, 79 cargos, from which 37 cargos are on board at time zero and 10 ships with different properties (sailing speed, total carrying capacity, time charter cost, and port costs). As shown Fig. 2, the company operates in Asia, but the fleet also serves Australia, India, and the Middle East. At beginning of planning horizon, each ships is located at a point of the sea and known its next port to visit and the estimated arrival time to that location. All problem data are similar to those specified by Jetlund and Karimi (2004). The planning horizon is usually 3–4 weeks long. The column generation algorithm and the mathematical models were implemented in GAMS using CPLEX 12.2 as the MILP solver and run on a DELL PRECISION T5500 Workstation with six-core Intel Xeon Processor (2.67 GHz). In addition, the CPLEX solution pool option and its tools associated were used for storing multiple columns with positive reduced cost in each execution of the slave problem. A time limit of 1 CPU hour has been imposed on the solution of every problem instance. All configuration options used in our algorithm are summarized in Table 1.

5.1. Example 1

Two reduced instances of the original case study, called Example 1A and 1B, were initially solved. They take into account just a subset of ships of the fleet. Analyzing the geographical distribution of the ports and knowing beforehand the first location visited by each ship (see Fig. 2), in case A, just the schedules of ships 1, 3, and 4 are simultaneously optimized, while in case B, ships 2, 6, 7, and 8 are considered for scheduling. Since no cargo can be served by more than one ship, the solutions found by this strategy may not be feasible for the company. Despite this, our goal is to compare the performance of the MILP mathematical model, with the one of the decomposition algorithm.

The optimal ship schedules for Examples 1A and 1B are shown in Figs. 3 and 4, respectively. For that schedules, loading/unloading activities are represented with black rectangles while sailing operations are showed in orange color. At each port, the discharged cargo are denoted with symbol (−) while picked-up cargo are symbolized with (+). Table 2 summarizes the computational results of the original MILP mathematical model and the best solution found by the CG strategy, for all problem instances, within the predefined time limit of up to 3 CPU hours. The best profit solutions and the integrality gaps found after 1 h and 2 h of CPU times are also reported. For the Example 1A, the CG method converged after 8 iterations to the optimal solution of USD 350894.68 in just 155 s while the MILP mathematical model was unable to close the gap in 3 h. For case 1B, the original MILP model found the optimal solution of USD 344156.56 after 646 CPU seconds time while the decomposition approach converged to the optimal solution and proved its optimality on the root node in just 41 s and after 6 iterations. The MILP formulation used 9091 s to prove the optimality of the solution by closing the integrality gap. It is important to remark that it was unnecessary to develop the search tree because the integer and optimal solutions for both problems were reported by the CG strategy in the root node of the branch-and-price tree (see Fig. 5).

5.2. Example 2

In Example 2, all ships in the fleet are simultaneously scheduled. Consequently, the amount of linear constraints and the number of

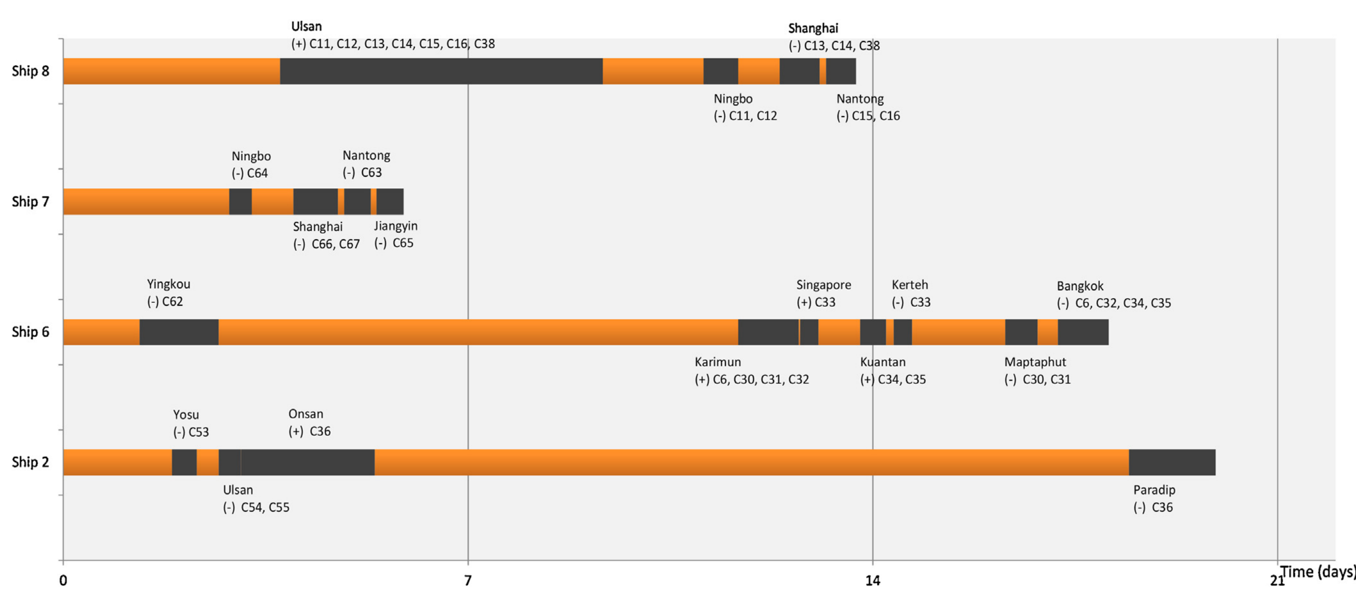

Fig. 4. Optimal schedules for Example 1B.

Table 2
Computational results for the original MILP model and the proposed column generation approach.

Problem instance	Exact optimization approach						Column generation	
	Linear constraints	Binary variables	Continuous variables	Optimality gap (%)	Best solution	CPU seconds	Best solution	CPU seconds
				36.36	339932.53	3600		
1A	24,041	2124	550	29.54	339932.53	7200	350894.68	155[a,b]
				22.97	350894.68	10,800		
				7.83	344152.56	3600		
1B	32,040	2832	733	2.62	344152.56	7200	344152.56	41[a,b]
				–	344152.56	9091		
				47.65	1056320.34	3600		
2	80,033	7080	1831	45.34	1056320.34	7200	1099997.33	440[a]
				40.98	1081279.97	10,800		851[b]
				78.00	1250208.10	3600		
3	99,722	7280	1831	77.00	1250208.10	7200	1460080.88	2042[a]
				60.33	1373396.99	10,800		2767[b]
				66.00	1346578.69	3600		
4	99,722	7280	1831	63.41	1346578.69	7200	1498547.73	2620
				54.65	1411889.42	10,800		

[a] To found the optimal solution.
[b] To prove optimality.

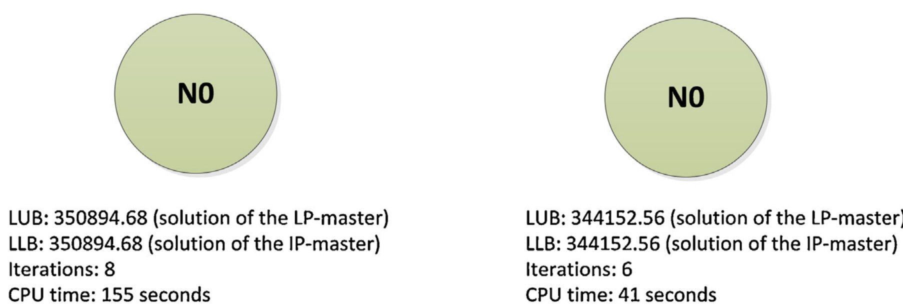

Fig. 5. Upper and lower bounds found in the root nodes for Examples 1A and 1B, respectively.

binary and continuous variables are significantly increased with regards to Examples 1A and 1B (see Table 2). The branch-and-price tree for Example 2 is depicted in Fig. 6. The promising nodes with non-integer solutions and *LUB* > *GLB* is represented in blue color, while those ones with *LUB* < *GLB* is depicted in orange color. Otherwise, when the optimal solution of the LP-master is integer, the node is showed in green color. From Fig. 6, it follows that the gap between the relaxation and the integer solution in the root node is just 0.13%. Consequently, a very good solution of USD 1,099,997 was found by the procedure in just 440 s. For closing the gap, the root node was divided into two new child nodes. After that, the left child node was discarded because no feasible solution contained in the set represented by the subspace can be better than

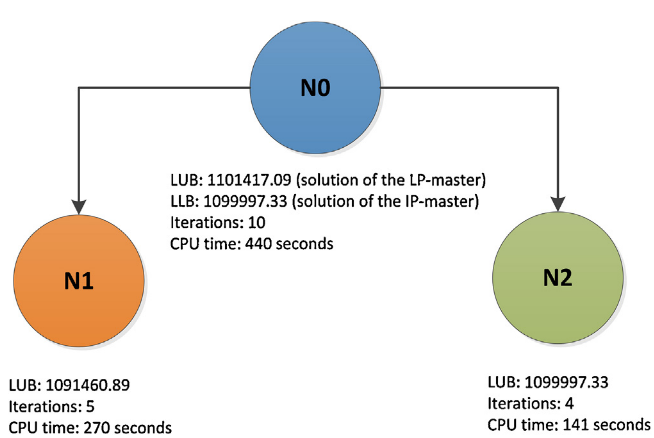

Fig. 6. Tree explored for solving the Example 2.

the existing global lower bound. Finally, the right child node converged to the integer solution found in the root node. The algorithm employed 851 s for proving the optimality of the solution. The optimal schedules for Example 2 are illustrated in Fig. 7. From this picture, it follows that there are several cargos which are not profitable for the company and, consequently, no ship serves those cargos. From the total profit, USD 2416833.70 are revenues from all transported cargos and USD 1316836.37 are operation expenses (USD 168051.33 from fuel cost, USD 206082.00from port charges, and USD 942703.04 from time-charter cost). The expected profit for the company's actual ship-routing and cargo assignment plan is USD 794,634 (Jetlund and Karimi, 2004). Consequently, the new schedules improve profits by approximately 40% with regards to the current schedule used by the company. Moreover, Table 3 shows that the solution reported by the CG procedure is 17.69% better than that one achieved by the heuristic presented by Jetlund and Karimi (2004). It is worth mentioning that the MILP optimization approach provided a solution with a net profit of USD 1056320.34 and an optimality gap of 47.65% after 1 CPU hr, the same net profit with a 45.34% gap after 2 CPU hours, and 40.98% with a net profit of USD 1081279.97 after 3 CPU hours. The best solution found by the MILP optimization approach after 3 h was 1.7% worse than that achieved by our CG strategy in just 440 s.

5.3. Example 3

Twenty additional cargos are considered in Example 3. Probability distributions, determined by analyzing the available original problem data, have been used to generate the information of new cargos. Table 4 details the loading/discharging ports, volumes, service revenues and pickup time windows for the new cargos.

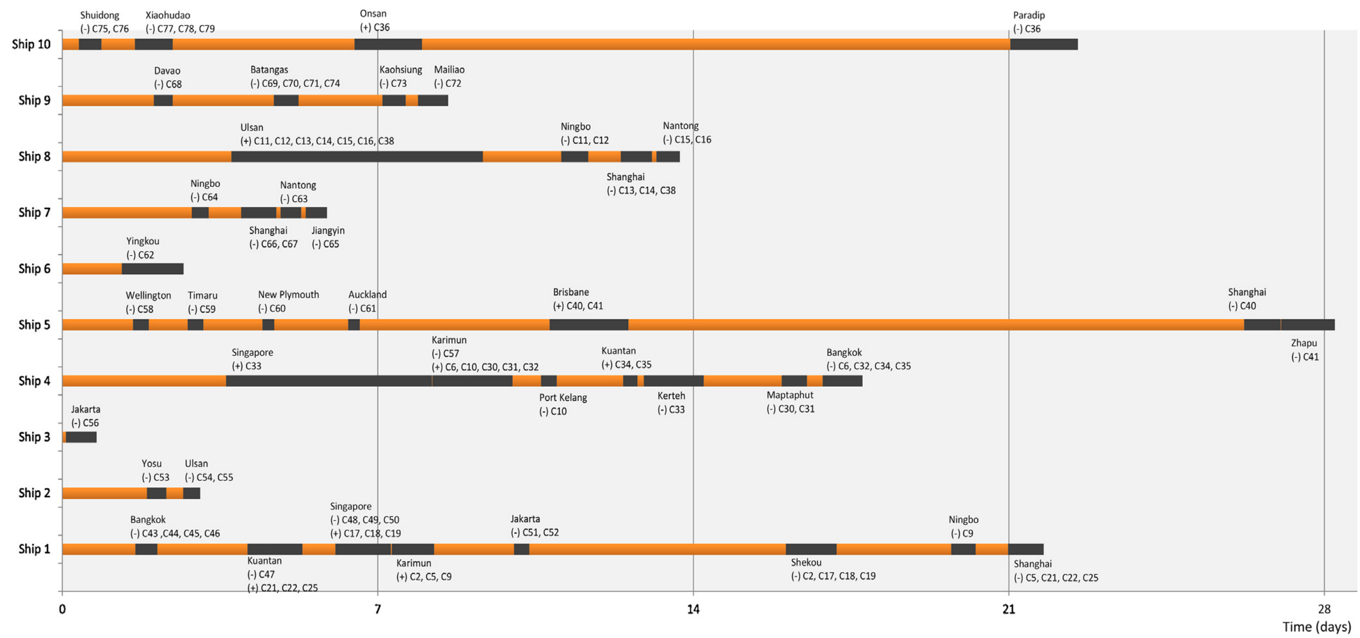

Fig. 7. Optimal schedules for the ship fleet in Example 2.

Table 3
Comparison of solution reported for Example 2.

	Jetlund and Karimi (2004)		CG procedure	
	Cargos served	Profit (USD)	Cargos served	Profit (USD)
S1	1, 5, 8–9, 18–19, 21, 25, 43–52	99,696	2, 5, 9, 17–19, 21–22, 25, 43–52	156,237
S2	11, 13, 36, 53–55	113,177	53–55	15,181
S3	6, 30–32, 56	63,218	56	62,722
S4	2–4, 7, 10, 17, 22–24, 57	116,228	6, 10, 30–35, 57	131,935
S5	40–41, 58–61	171,609	40–41, 58–61	171,609
S6	12, 14–16, 38	73,991	62	70,772
S7	63–67	103,312	63–67	103,312
S8	–	−40,887	11–16, 38	46,841
S9	68–74	125,291	68–74	125,291
S10	75–79	108,997	36, 75–79	216,097
	Total profit (USD)	934,632		1,099,997

Table 4
Information related to the new cargos.

	Origin	Destination	Pickup time windows	Volume (tonnes)	Shipping rate (USD)
C80	Ulsan	Nantong	21–25 April	876	25,000
C81	Kuantan	Shanghai	25–29 April	1344	40,000
C82	Karimun	Bangkok	21–25 April	84	26,000
C83	Brisbane	Bangkok	23–27 April	630	68,570
C84	Ulsan	Kandla	25–29 April	77	42,500
C85	Singapore	Wellington	26–29 April	1515	40,000
C86	Karimun	Zhapu	25–29 April	1120	48,300
C87	Ulsan	Jiangyin	25–29 April	2543	40,000
C88	Singapore	Davao	25–29 April	1812	60,000
C89	Kuantan	Jakarta	21–25 April	2668	42,500
C90	Singapore	Ningbo	25–29 April	1224	26,250
C91	Karimun	New Plymouth	25–29 April	459	34,000
C91	Karimun	New Plymouth	25–29 April	459	34,000
C92	Ulsan	Shanghai	26–29 April	1539	28,000
C93	Mailiao	Kuantan	25–29 April	1369	50,000
C94	Brisbane	Batangas	21–25 April	797	36,000
C95	Karimun	Shuidong	22–27 April	37	32,000
C96	Ulsan	Shanghai	24–28 April	127	71,460
C97	Karimun	Bangkok	25–29 April	45	30,000
C98	Karimun	Ulsan	25–29 April	89	52,370
C99	Ulsan	Shekou	25–29 April	271	26,000

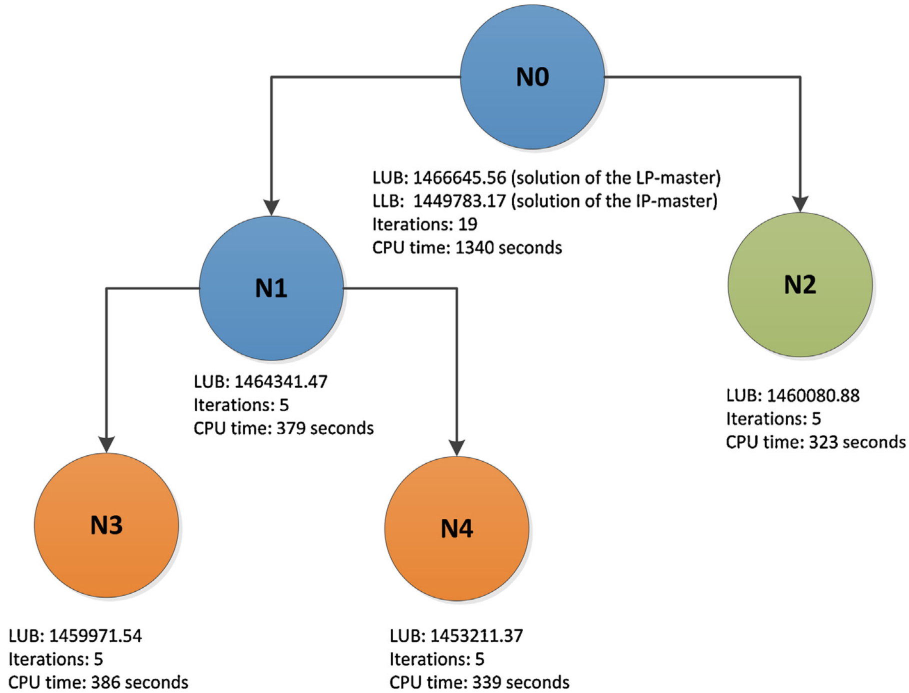

Fig. 8. Tree explored for solving the Example 3.

Remaining problem data remains without changes with respect to Example 2. The presence of additional cargos increases considerably the number of variables and constraints in the MILP sub-problem. In particular, the number of binary variables rises from 7080 to 7280. A time limit of 10 s has been imposed on the solution of every sub-problem execution.

A good and feasible solution of USD 1449783.17 was found in the root node of the branch-and-price tree just after 1340 s through the proposed algorithm while the MILP mathematical model reported a solution of USD 1373396.99 and an optimality gap of 60.33% after 3 CPU hours. The solution found by the original MILP model is 6.3% worse than that one achieved by the decomposition strategy. By analyzing the performance of the decomposition strategy (see Fig. 8), it follows that the gap between the relaxation and the integer solution was just 1.1%. After exploring the branching tree, the integrality gap was closed to 0 and the current best integer solution was enhanced from USD 1449783.17 to USD 1460080.88. This solution was found after 2042 s and exploring all nodes in the tree took 2767 s. The best schedules for Example 3 are shown in Fig. 9. It is observed that, with the same fleet capacity, the number of cargos served rises from 64 to 78 with regards to Example 2. This improves the company's profit by 32.73%. The revenues were

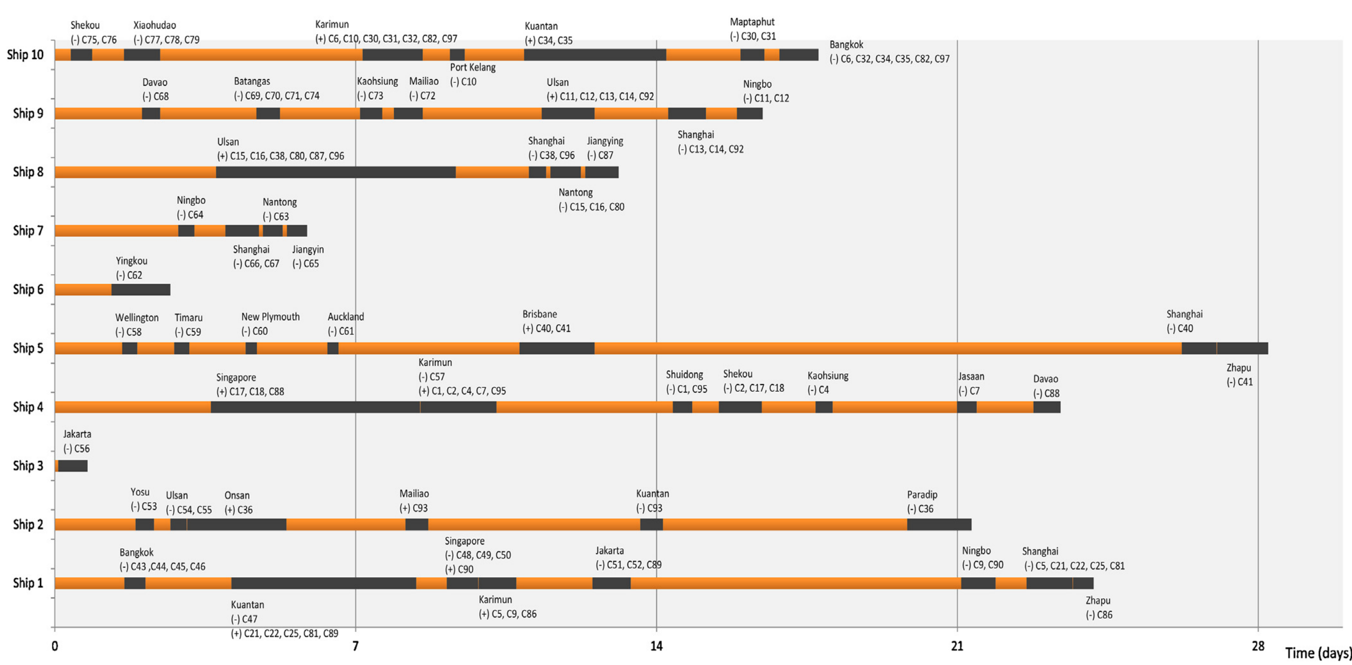

Fig. 9. Best schedules for the ship fleet in Example 3.

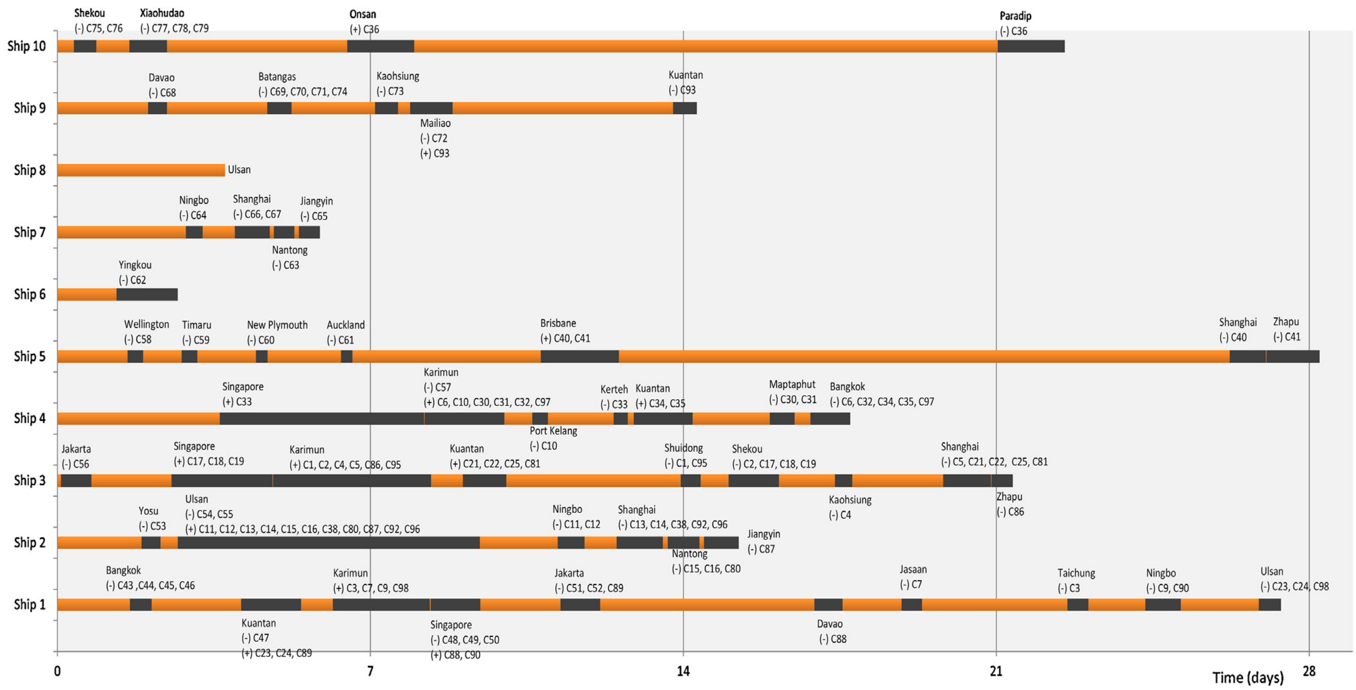

Fig. 10. Best schedules for the ship fleet in Example 4.

improved 28.55% from USD 2416833.70 to USD 3107010.62, while the total operation costs grew 25.07%. That's because new ports can be visited by the ships.

5.4. Example 4

The previous scenario is revisited in Example 4 but now the maximum number of ports that each ship can visit during the planning horizon (parameter K_s) is incremented from 8 to 10 for ships 1, 2, 5, and 6, and from 7 to 9 for ships 3, 4, 7, 8, 9, and 10. Although the number of variables and constraints remain unchanged with regards to Example 3, the mathematical model is hardened by the change of the route-length constraint. Consequently, a time limit of 20 s has been imposed on the solution of every sub-problem. It can be observed in Table 2 that the exact optimization approach reported a solution of USD 1411889.42 with an optimality gap of 54.65% after 3 CPU hours. In contrast, the decomposition algorithm found an integer solution of USD 1498547.73 in 2620 s. The net profit is 7.4% higher than the best value reported by the original mathematical model. The best schedules are depicted in Fig. 10. By analyzing this picture, it follows that the end of planning horizon is maintained with respect to Example 3. However, the amount of cargos served rises from 78 to 84, although that ship 8 is idle.

The CG strategy, also called price-and-branch strategy, has been used to solve the Example 4 because the gap between the relaxation and the integer solution was just 0.35%. According to Desrosiers and Lübbecke (2010), in practical applications it may not be necessary to close the last percents of the optimality gap. As in our case, many practitioners used price-and-branch, i.e., the column generation is used just in the root node to find a very good solution to a real-world problem.

6. Conclusions

The cost-effective routing and scheduling of a ships fleet represents an important decision making problem for the chemical and shipping industry. Therefore, this paper focused on the development of a novel MILP-based column generation algorithm for effectively managing the logistics activities performed by tramp shipping companies operating with a heterogeneous fleet of multi-parcel chemical tankers.

Since the finding of an optimal cargo assignment and scheduling for a ship fleet lead to an NP-hard problem, the computational efficiency of monolithic MILP approaches rapidly deteriorates. Thus, for complex instances, the reported solutions usually feature large integrality gaps. However, if the mathematical model is used jointly with a decomposition method, the resulting algorithm is able of reaching good and frequently optimal solutions for large-scale problems with a relatively low computational cost.

The performance of our proposed approach in terms of solution quality and CPU time has been evaluated by solving five complex examples all dealing with the distribution activities of a multi-national shipping company operating a fleet of multi-parcel chemical tankers in the Asia Pacific Region. Computational results show that the MILP-based CG strategy can efficiently find the optimal solution for relatively small or tightly constrained instances in a very short time. Moreover, the algorithm is able to generate good feasible solutions for more complex instances, involving up to 99 cargos, 36 ports, and 10 vehicles, in a reasonable CPU time. Such computational performance outperforms the exact optimization model and some heuristic decomposition methods presented in the literature to solve the same real-world example.

Acknowledgements

Financial support received from AECID under Grant PCI-A1/044876/11, from UNL under Grant PI-81-481, from ANPCyT under grant PICT-2010-1073, and from SECTEI Santa Fe under grant 2010-161-12 are acknowledged.

References

Barnhart C, Johnson E, Nemhauser G, Savelsbergh M, Vance P. Branch and Price. Column generation for solving huge integer programs. Oper Res 2000;48:316–29.
Bettinelli A, Ceselli A, Righini G. A branch-and-cut-and-price algorithm for the multi-depot heterogeneous vehicle routing problem with time windows. Transport Res C: Emerg Technol 2011;19:723–40.

Brønmo G, Christiansen M, Nygreen B. Ship routing and scheduling with flexible cargo sizes. J Oper Res Soc 2007;58:1167–77.
Brønmo G, Nygreen B, Lysgaard J. Column generation approaches to ship scheduling with flexible cargo sizes. Eur J Oper Res 2010;200:139–50.
Christiansen M, Fagerholt K, Nygreen B, Ronen D. Maritime transportation. In: Barnhart C, Laporte G, editors. Handbook in Operations Research and Management Science: Transportation, vol. 14. Amsterdam: Elsevier, North-Holland; 2007. p. 189–284.
Christiansen M, Fagerholt K, Nygreen B, Ronen D. Ship routing and scheduling in the new millennium. Eur J Oper Res 2013;228:467–83.
Dantzig G, Wolfe P. Decomposition principles for linear programs. Oper Res 1960;8:101–11.
Desaulniers G, Desrosiers A, Erdmann A, Solomon M, Soumis F. VRP with pickup and delivery. In: Toth P, Vigo D, editors. The vehicle routing problem, SIAM monographs on discrete mathematics and applications, vol. 9. Philadelphia: SIAM; 2002. p. 225–42.
Desrosiers J, Lübbecke ME. Branch-price-and-cut algorithms. In: Cochran JJ, Cox LA, Keskinocak P, Kharoufeh JP, Smith JC, editors. Wiley encyclopedia of operations research and management science, vol. 8. New York, NY: Wiley; 2010.
Dondo R, Méndez CA, Cerdá J. Optimal management of logistic activities in multi-site environments. Comput Chem Eng 2008;32:2547–69.
Jetlund AS, Karimi IA. Improving the logistics of multi-compartment chemical tankers. Comput Chem Eng 2004;28:1267–83.
Nishi T, Izuno T. Column generation heuristics for ship routing and scheduling problems in crude oil transportation with split deliveries. Comput Chem Eng 2014;60:329–38.
Papageorgiou LG. Supply chain optimisation for the process industries: advances and opportunities. Comput Chem Eng 2009;33:1931–8.
Ropke S, Cordeau S. Branch and cut and price for the pickup and delivery problem with time windows. Transport Sci 2009;43:267–86.
Ryan D, Foster B. An integer programming approach to scheduling. In: WREN A, editor. Computer scheduling of public transport, urban passenger vehicle and crew scheduling,. Amsterdam: North-Holland; 1981. p. 269–80.
Savelsbergh M, Sol M. The general pick-up and delivery problem. Transport Sci 1995;29:17–29.
Savelsbergh M, Sol M. DRIVE: Dynamic Routing of Independent Vehicles. Oper Res 1998;46:474–90.
UNCTAD. Review of Maritime Transport; 2012, <www.unctad.org/en/PublicationsLibrary/rmt2012_en.pdf>.