Proceedings of the 2011 International Conference on Industrial Engineering and Operations Management
Kuala Lumpur, Malaysia, January 22 – 24, 2011

Constraint Programming Based Column Generation Heuristics for a Ship Routing and Berthing Time Assignment Problem

King-Wah Pang, Chung-Lun Li
Department of Logistics and Maritime Studies
The Hong Kong Polytechnic University, Hung Hom, Hong Kong

Abstract

We develop a constraint programming based heuristic algorithm using column generation technique to solve a ship routing problem which the loading and unloading times of cargoes at pickup and drop-off locations are significant. In order to prevent congestions at the locations, we need to assign berthing time slots to each vessel to perform the loading and unloading tasks at different locations. This problem is motivated by the operations of a feeder vessel service company with company-owned cargo terminals, which the shipping company wishes to coordinate the routing and the berthing time of the vessels.

Keywords
Ship routing; berthing time assignment; constraint programming; column generation.

1. Introduction
We consider a ship routing problem that loading and unloading times of cargoes at pickup and drop-off locations are significant. At each pickup or drop-off location, we need to assign berthing time slots to each vessel to perform the loading and unloading tasks in order to prevent congestions at the locations. This problem is motivated by an application in ship routing and scheduling, where a shipping company runs a fleet of feeder vessels shuttling among various terminals in Hong Kong and Pearl River Delta. The company also owns several small terminals to load/unload the cargoes. Loading and unloading cargo containers at the terminals are time consuming, and without a proper planning of the ship arrival schedule, congestions are common that some vessels have to wait at sea for some time before they can get berthed. The company transports cargo containers from their origins to their destinations according to the customers' orders via these vessels. Besides of loading and unloading containers at those small company-owned cargo terminals, some containers have to be loaded from or unloaded to the public container terminals, where advanced booking of the berthing time slot for the vessels is required. This advanced booking requirement complicates the ship routing and scheduling plan of the vessels, so that it is important for the company to take into consideration the assignment of berthing time slots at those loading/unloading areas in order to avoid congestion and waiting.

In the past several decades, many research studies on ship routing and scheduling problems were presented in the literature. Almost in every decade since 1983, comprehensive surveys on the topic can be found in [1-4]. They have presented the ship scheduling problems and showed the differences between ship scheduling problems and other vehicle routing problems as follows: Each cargo ship has different capacity while truck capacities are generally homogenous; cargo ships operate around the clock, but trucks do not operate at night in most situations. Since cargo ships operate 24/7, they do not need to have a depot to return after serving all customers, but land-based vehicles do. These differences make ship scheduling problems more difficult to solve. Recently, Chen *et al.* [5] studied container-vessel scheduling with bi-directional flows of containers. Agarwal and Ergun [6] considered a integrated ship-scheduling and cargo-routing problem for liner shipping services. Boros *et al.* [7] studied the coordination of ship schedule and container yard operations for the determination of the optimal cycle time. Hwang *et al.* [8] presented a set-packing model to solve a ship scheduling problem with constraints on the profit variance. Pang *et al.* [9] considered a ship routing problem with time clash avoidance constraints at the pickup and drop-off points. Brønmo *et al.* [10] studied a ship scheduling problem with flexible cargo sizes. Chuang *et al.* [11] proposed a fuzzy genetic algorithm for liner shipping planning. To the best of our knowledge, with the exception of [9], no study has been conducted on ship scheduling with berthing time assignment considerations. In [9], they presented a ship routing and berthing time assignment problem as a set partitioning problem, and used column generation approach

with dynamic programming algorithm to solve the subproblems as constrained shortest path problems. In fact, their proposed heuristic has limitation on the number of container batches that can be handled by each vessel when solving the constrained shortest path subproblems. In this paper, we follow the same set partitioning formulation, but we propose using constraint programming algorithm to solve the subproblem that does not have such limitation. That means a vessel can take as many container batches as possible, provided that it is economical and within the capacity limit of the vessel. This newly proposed approach fits the practical application in feeder vessel operations, where companies often prefer the vessels to take as many cargoes as possible in order to reduce the total operating cost.

To tackle our problem, we first describe the model as proposed in [9]. Due to the model complexity, we propose a decomposition method to solve the problem heuristically, applying column generation techniques and constraint programming approaches. We will test the effectiveness of our heuristic via extensive computational experiments using randomly generated data. The rest of the paper is organized as follows: In Section 2, we describe the characteristics of the problem we study. We present our solution method in Section 3. Design of the computational experiments is presented in Section 4, followed by some concluding remarks and future work in Section 5.

2. Problem description

The problem we study can be formulated as a mixed integer linear programming (MILP) model as proposed in [9], which is similar to a multiple vehicles pickup and delivery problem that allows more than one vehicle visit a location so that the coordination of the arrival time of the vehicles at the locations is critical to avoid congestion. Details of the model formulation can be found in [9]. In brief, we have a set of vessels, and each with a capacity. Each vessel has a designated origin and destination. Apart from that, we have a set of containers waiting to be transported in batches. Each container batch has a given origin and a given destination. Each vessel is only allowed to perform a pickup or drop-off operation at a location within a specific time window. In addition, each pickup and drop-off location (i.e., a berth at the cargo terminal) can only serve one vessel at a time. A cargo terminal can be a company-owned cargo loading/unloading area or a public container terminal. For a berth of a company-owned terminal, the time window should be vessel-independent which represents the time period that the berth is in operation. However, for the berths of a public container terminal, it normally requires advanced booking of berthing time windows for a specific vessel, thus, the time window is vessel-dependent. In our model, we assume the vessels are allowed to wait at sea at no cost. We need to arrange the vessels to pick up the containers from their origins and transport them to their destinations with the planning horizon. Referring to the MILP formulation in [9], the problem is a generalization of the classical traveling salesman problem. Thus, the problem is NP-hard in the strong sense. In the next section, we present a decomposition method to solve the problem heuristically.

3. Solution method

In this paper, we follow the set partitioning problem formulation proposed in [9] to reformulate the problem of our study as $\boldsymbol{P_1'}$ and solve it by column generation approach with constrained shortest path problem as the subproblem $\boldsymbol{P_2'}$. We adopt the constraint programming approach to solve the subproblems $\boldsymbol{P_2'}$ to generate new columns that enter to the set partitioning problem $\boldsymbol{P_1'}$. We propose a different solution approach to solve the decomposed subproblems, which resolves the limitation in [9] on the number of container batches that can be handled by each vessel. The set partitioning formulation of our study is an extension of the model in [12] which is formulated as a multiple vehicles pickup and delivery problem with time window and capacity constraints. We extended the formulation with consideration of the berthing time clash avoidance constraint at the terminals. By iteratively solving problem $\boldsymbol{P_1'}$ as a set partitioning problem and solving problem $\boldsymbol{P_2'}$ to generate new columns for $\boldsymbol{P_1'}$, we can obtain an optimal solution to the original problem. Given a feasible solution to the revised problem $\boldsymbol{P_1'}$, a constrained shortest path subproblem is constructed to seek for new columns that enter to the problem $\boldsymbol{P_1'}$. For each vessel v, a route R_v is defined as a directed path which starts from the vessel origin r_v and ends at the vessel destination s_v, and it takes care of a subset of the container batches by picking up the container batches at their origins and delivering them to their destinations. Throughout the route, the load must not exceed the vessel capacity K_v. In addition, the vessel must perform its pickup and drop-off tasks within the given time windows. The detailed formulations of $\boldsymbol{P_1'}$ and $\boldsymbol{P_2'}$, as well as the description of the iterative method, are presented in the following subsections.

3.1. The Set Partitioning Formulation

We follow the same set partitioning formulation proposed in [9] to represent our ship routing and berthing time assignment problem with the following definitions:

Ω_v = the set of all feasible pickup and delivery routes for vessel v;

$\delta_{bR}^v = \begin{cases} 1, & \text{if } b \in B \text{ is served on route } R \in \Omega_v \\ 0, & \text{otherwise} \end{cases}$

$\sigma_{\rho tR}^v = \begin{cases} 1, & \text{if vessel } v \text{ is at berth } \rho \text{ at time } t \text{ on route } R \in \Omega_v \\ 0, & \text{otherwise} \end{cases}$

c_R^v = the cost of route $R \in \Omega_v$.

We define a binary decision variable X_R^v for each vessel $v \in V$ and each route $R \in \Omega_v$. Variable X_R^v equals to 1 if route R is taken by vessel v, and 0 otherwise. The ship routing and berthing time assignment problem can then be reformulated as follows:

$\boldsymbol{P_1'}$:	minimize	$\sum_{v \epsilon V} \sum_{R \epsilon \Omega_v} c_R^v X_R^v$		(1)
	subject to	$\sum_{v \epsilon V} \sum_{R \epsilon \Omega_v} \delta_{bR}^v X_R^v = 1$	(for all $b \in B$)	(2)
		$\sum_{R \epsilon \Omega_v} X_R^v = 1$	(for all $v \in V$)	(3)
		$\sum_{v \epsilon V} \sum_{R \epsilon \Omega_v} \sigma_{\rho tR}^v X_R^v \leq 1$	(for all ρ and t)	(4)
		$X_R^v = 0$ or 1	(for all $v \in V$ and all $R \in \Omega_v$)	(5)

Objective function (1) is to minimize the total travelling cost of the vessels. Constraints (2) ensure each container batch is taken care of by exactly once. Constraints (3) indicate that each vessel can travel only on one route. Constraints (4) ensure that each berth location ρ is berthed with at most one vessel at any time t. In order to simplify the constraints (4) with time t as one dimension of the variable $\sigma_{\rho tR}^v$, the planning horizon defined as T is discretized into short period intervals. For instance, if we discretize the planning horizon T of one day into 5-minute intervals, a planning horizon interval of 1440 minutes equals to 1440/5 = 288 intervals. By discretizing the planning horizon into 5-minute intervals, it reduces the complexity of the set partitioning problem, and also reduces the space requirement for the memory to store the berthing time information in constraints (4) comparing to the case assuming the variable t is continuous. For each route in the solution, we check if each 5-minute time interval at berth location ρ is occupied by a vessel route in the current solution. Constraints (5) define the binary decision variables.

The size of this set partitioning model is very large in general because of the huge number of possible vessel routes in the solution space, and it is impractical to generate all feasible routes so as to solve the problem optimally. Hence, we adopt a similar approach in [13] to iteratively generate new columns representing new admissible routes of a vessel that enter to the problem. We first relax the integer constraints (5) in order to obtain the values of the dual variables π_b ($b \in B$), π_v'($v \in V$) and $\pi_{\rho t}''$(for all ρ and all $t \in T$) associated with constraints (2) – (4) respectively. Let $\mathbf{P_{1,LP}'}$ denote this LP-relaxation problem. Given a feasible solution of $\boldsymbol{P_{1,LP}'}$, together with the corresponding value of the dual variables π_b , π_v' and $\pi_{\rho t}''$, a vehicle is selected, (i.e. v) and a new constrained shortest path subproblem with time-window and capacity constraints is constructed, and defined as $\boldsymbol{P_2'}$. By solving the subproblem $\boldsymbol{P_2'}$ for a vehicle v, a new column is generated to enter the master problem $\boldsymbol{P_1'}$ if a solution to the subproblem $\boldsymbol{P_2'}$ with a negative solution value is detected; otherwise, we select another vessel to construct a new constrained shortest path subproblem. More new columns are generated in this manner by iterating between the problems $\boldsymbol{P_{1,LP}'}$ and $\boldsymbol{P_2'}$ until an optimal solution is detected or until the stopping criteria are met.

3.2. The Constrained Shortest Path Subproblem

According to linear programming duality theory, problem $\boldsymbol{P_{1,LP}'}$ is optimal if and only if the reduced cost $\bar{c}_R^v$ is nonnegative for all $v \in V$ and all $R \in \Omega_v$, where

$$\bar{c}_R^v = c_R^v - \sum_{b \in B} \delta_{bR}^v \pi_b - \pi_v' + \sum_{\rho} \sum_{t \in T} \sigma_{\rho tR}^v \pi_{\rho t}'' \tag{6}$$

Hence, generating a new column for the master problem $\boldsymbol{P_{1,LP}'}$ is equivalent to determining that the following value is negative:

$$min\{\bar{c}_R^v \mid v \in V, R \in \Omega_v\}$$

This is equivalent to determining that the following value is negative:

$$min_{v \in V}\{min\{\bar{c}_R^v \mid R \in \Omega_v\}\} \tag{7}$$

Obtaining the value of $\min\{\bar{c}_R^v \mid R \in \Omega_v\}$ in (7) is the same as constructing a feasible route for vessel v that minimizes the reduced cost $\bar{c}_R^v$ which can be defined as a constrained shortest path problem.

We propose to adopt the constraint programming (CP) approach to solve the constrained shortest path subproblem $\boldsymbol{P_2'}$ because CP formulation offers higher flexibility to include logic constraints that reflects the practical constraints, i.e. the constraints to ensure the vessel visits the drop off location if it has visited the pickup location of the container batch. With this constraint, the solution space is dynamically reduced during the search process by constraint propagation routine. Besides, we can easily introduce extra valid constraints in order to improve the performance by pruning and filtering, comparing with the fixed structure of the MILP formulation of the subproblem. Several research studies have been presented to show the suitability of using constraint programming approach to solve the constrained shortest path problems in vehicle routing related problems. Interested readers can refer to the references [14-16].

We adopt a similar constraint programming formulation for the variable definition as proposed by [17] to model the constrained shortest path subproblem $\boldsymbol{P_2'}$ of our study. We include additional constraints to penalize the vessels for visiting the same berth location at the same time, and add constraints to ensure the vessel visits the drop off location of a container batch if it has visited the batch's pickup location. For a selected vessel v, we solve this constrained shortest path subproblem, and a new column for the master problem $\boldsymbol{P_{1,LP}'}$ is generated once a feasible solution to the subproblem $\boldsymbol{P_2'}$ with a negative objective value is detected.

To formulate the constrained shortest path subproblem $\boldsymbol{P_2'}$, let V denote the set of all vessels. The vessels have different capacities, denoted as K_v, that travel at the same speed. Let B denote the set of all container batches. For simplicity, denote $B = \{1, 2, \dots, |B|\}$. We represent the pickup and drop-off locations of batch b by b and $|B| + b$, respectively. We define w_b as the quantity of containers to be transported for each batch b. We denote the origin of the vessel as location 0, and the destination of the vessel as location $2|B| + 1$. We let H be the location set $H = \{0, 1, 2, \dots, 2|B| + 1\}$, G^+ be the set of all pickup locations $\{1, 2, \dots, |B|\}$, G^- be the set of all drop-off locations $\{|B| + 1, |B| + 2, \dots, 2|B|\}$, and $G = G^+ \cup G^-$. We denote c_{ij} as the cost for the vessel to travel from location i to location j (note: $c_{ij} = 0$ if i and j belong to the same berth location; for example, if $|B| = 10$ and the pickup location of batch 4 is the same as the drop-off location of batch 3, then $c_{4,13} = 0$). Next, we define the reduced cost for the vehicle to go from location i to location j as $\bar{c}_{ij} = c_{ij} - \mu_i$, where μ_i equals to π_b obtained from the dual variable of constraints (2) in problem $\boldsymbol{P_{1,LP}'}$ if $i \in G^+$, μ_i equals 0 if $i \in G^-$, μ_0 equals to π_v' obtained from the dual variable of constraints (3) and $\mu_{2|B|+1}$ equals 0. We further define $\sum_{t=t_1}^{t_2} \pi_{\rho_i t}''$ as the penalty cost for a vessel to occupy the berth ρ_i from time t_1 to time t_2 while the vessel is performing the loading or unloading operation of a container batch at the berth ρ_i. The value of $\pi_{\rho_i t}''$ can be obtained from the dual variable of constraints (4) in problem $\boldsymbol{P_{1,LP}'}$.

To formulate the constrained shortest path problem $\boldsymbol{P_2'}$ for a vessel, we define the following parameters:

$\tilde{\tau}_i$ = service time (i.e., either loading or unloading) of the vessel at location i (note: $\tilde{\tau}_0 = \tilde{\tau}_{2|B|+1} = 0$);
$\tilde{t}_{ij}$ = travel time of the vessel from location i to location j;
$\tilde{e}_i$ = earliest time that the vessel is allowed to occupy location i;
$\tilde{f}_i$ = latest time that the vessel is allowed to occupy location i;
$\tilde{k}$ = K_v = vessel capacity;
$\tilde{d}_i$ = "demand" at location $i = \begin{cases} w_b, & \text{if } i \text{ is the pickup location of batch } b; \\ -w_b, & \text{if } i \text{ is the dropoff location of batch } b; \\ 0, & \text{if } i = 0 \text{ or } 2|B| + 1 \end{cases}$

Define the following decision variables:

$\delta_i \in H \backslash \{0\}$ for all $i \in H \backslash \{2|B| + 1\}$	next visit location after the vessel visits location i;
$A_i \in [\tilde{e}_i, \tilde{f}_i]$ for all $i \in H$	arrival time of the vessel at location i;
$D_i \in [\tilde{e}_i, \tilde{f}_i]$ for all $i \in H$	departure time of the vessel from location i;
$l_i \in [0, \tilde{k}]$ for all $i \in H$	cumulative load of the vessel when it arrives at location i.

The constrained shortest path problem can be formulated as the following constraint programme:

$$\boldsymbol{P}_2'\text{:} \quad \text{minimize} \quad \sum_{i\in H\setminus\{2|B|+1\}} \bar{c}_{i\delta_i} + \sum_{i\in H:\delta_i\neq i} \sum_{t=A_i}^{D_i} \pi''_{\rho_i t} \tag{8}$$

subject to

$$\text{AllDifferent}(\delta) \tag{9}$$

$$D_i = A_i + \tilde{\tau}_i \qquad \text{(for all } i \in H) \tag{10}$$

$$A_{\delta_i} \geq D_i + \tilde{t}_{i\delta_i} \qquad \text{(for all } i \in H\setminus\{2|B|+1\}) \tag{11}$$

$$A_i < A_{|B|+i} \qquad \text{(for all i} \in \text{G}^+) \tag{12}$$

$$l_i + \tilde{d}_i = l_{\delta_i} \qquad \text{(for all } i \in H\setminus\{2|B|+1\}) \tag{13}$$

$$(\delta_i \neq i) \Leftrightarrow \delta_{|B|+i} \neq |B| + i \qquad \text{(for all i} \in \text{G}^+) \tag{14}$$

In this formulation, objective function (8) minimizes the total reduced cost of the vessel route. Constraint (9) ensures conservation of flow of every batch b at all pickup and drop-off locations. The $\text{AllDifferent}(\delta)$ in constraint (9) ensures no two locations have the same immediate successor location as this constraint confines that there is only one outgoing arc from each location. By ensuring the values of the variable δ_i of all locations are different, we can also restrict only one incoming arc connects to each location i. For those locations not belong to the shortest path of the vessel, the value of the variable δ_i will take the value of i automatically. Constraints (10) define the departure time of the vessel leaving from location i, which is bounded by the time window in the variable definition. This constraint also forces the vessel to leave the current location i as soon as it finishes all the loading and unloading operations in order to minimize the penalty cost of occupying the berth. Constraints (11) ensure the vessel arrives at next location δ_i no earlier than the departure time of the vessel at location i plus the travel time from location i to δ_i. This inequality constraint allows the vessel waits at the sea at no cost after it departs from the location i if the location δ_i is currently not available to serve the vessel. This constraint can also be used to ensure no sub-tour in any solution to the constrained shortest path problem. Constraints (12) ensure the vessels visit the container batch's pickup location before visiting the batch's drop-off location. Constraints (13) indicate the cumulative load when the vessel arrives at location δ_i equals to the cumulative load when the vessel arrives at location i plus the demand request at location i. Constraints (14) specify that if the vessel visits a pickup location of a container batch, then it must also visit that batch's drop-off location, and vice versa.

The constrained shortest path problem can be solved by using ILOG Constraint Programming (CP) Optimizer, which is specifically designed to solve this kind of scheduling problem effectively. The CP Optimizer is incorporated with constraint propagation algorithm and neighborhood search heuristic. We do not use the CP Optimizer to solve the constrained shortest path subproblem optimally to generate new columns. It is because we need to solve many of these subproblems for the vessels during the solution search process. It is time consuming to solve the subproblem optimally for any reasonable size application. Also, we do not expect to solve the set partitioning problem $\boldsymbol{P}_{1,LP}'$ optimally, which is costly both in terms of computing time and memory space requirement. Another reason is even we obtain the optimal solution to the revised set partitioning problem $\boldsymbol{P}_{1,LP}'$, we cannot guarantee the obtained solution is the optimal solution to the original problem $\boldsymbol{P}_1'$ that takes only integer values on the decision variables. To balance the tradeoff between the solution quality and computing time requirement, we adopt the CP Optimizer to quickly generate new columns for the set partitioning problem by setting the computing time limit in each iteration.

The iterative approach continues until the stopping criteria are met. In order to determine the stopping criteria, we will test different approaches and assess their suitability through extensive computational experiments. The proposed stopping criteria include the determination of the solution to the LP relaxed MILP formulation of the original problem as the theoretical lower bound, and compare with the solution obtained from our iterative approach. If the solution obtained is within certain percentage difference from the lower bound, the iteration will be stopped. Another stopping criterion is based on the total computing time for the iterative approach. The program will be stopped when the computing time reaches a pre-determined threshold. This total computing time is determined by the time available for the planning process in real application. Furthermore, the third criterion to stop the iteration is when there is no new column could be found when solving the constrained shortest path subproblem for all vessels within a computing time limit. We will also consider combining these stopping criteria together.

4. Computational experiment design

To conduct the experiment for the performance evaluation of the proposed algorithm, the major parameter settings follow that of real-life operations of the feeder vessel service company operates in Hong Kong. The required parameters include the planning horizon, loading and unloading time per container, travel speed of the vessels, location of the terminals and berths that determines the distance between the locations, and the capacity of the vessels. Other parameters that simulate different scale of company operations include the number of berths, number of vessels and number of container batches to be transported by the vessels. For these three parameters, three set of test cases will be randomly generated to represent a small, medium and large scale operations. For each type of the test cases, some random sample problems will be generated and the average solution values will be identified for performance comparison. Besides, the number of containers in each container batch will be randomly generated. The result obtained from the proposed iterative algorithm will be compared with other simple heuristics in order to justify the applicability and suitability of using the proposed approach to solve the real life ship routing and berthing time assignment problems.

5. Conclusions

This paper presents a constraint programming based column generation heuristic for an integrated model for ship routing and berth assignment problem, which is particularly useful for the decision making of the shipping companies that operate feeder vessels and company-owned terminals. An iterative approach is applied to solve the ship routing and berth assignment integrated problem heuristically. In future, extensive computational experiments will be conducted to show the performance of the proposed algorithm compare with some theoretical lower bounds. Based on the computational results, we will fine-tune the heuristic so as to improve its performance. Also, the proposed algorithm can be used as a backbone for the solution approach to solve the extended ship routing and berth assignment problem. It considers the option of transshipment that a container batch can first be loaded to a vessel, then transported and unloaded to a company-owned terminal, and later being picked up by another vessel again to further transport to its destination. This extension helps the feeder vessel service company to further improve their operation efficiency, better utilize the company-owned facilities and coordinate with the public terminal operators.

Acknowledgements

This research work was supported by a research grant of The Hong Kong Polytechnic University under grant number J-BB7G.

References

1. Ronen, D., 1983. Cargo ships routing and scheduling: Survey of models and problems. *European Journal of Operational Research* 12(2), 119-126.
2. Ronen, D., 1993. Ship scheduling: The last decade. *European Journal of Operational Research* 71(3), 325-333.
3. Christiansen, M., Fagerholt, K., Ronen, D., 2004. Ship routing and scheduling: status and perspectives. *Transportation Science* 38(1), 1-18.
4. Christiansen, M., Fagerholt, K., Nygreen, B., Ronen, D., 2007. Maritime transportation. In *Handbooks in Operations Research and Management Science, Volume 14: Transportation*, Barnhart, C. and Laporte, G. (eds.), North-Holland, Amsterdam, 189-284.
5. Chen, Z.-L., Lei, L., Zhong, H., 2007. Container vessel scheduling with bi-directional flows. *Operations Research Letters* 35(2), 186-194.
6. Agarwal, R., Ergun, O., 2008. Ship scheduling and network design for cargo routing in liner shipping. *Transportation Science* 42(2), 175-196.
7. Boros, E., Lei, L., Zhao, Y., Zhong, H., 2008. Scheduling vessels and container-yard operations with conflicting objectives. *Annals of Operations Research* 161, 149-170.
8. Hwang, H.-S., Visoldilokpun, S., Rosenberger, J.M., 2008. A branch-and-price-and-cut method for ship scheduling with limited risk. *Transportation Science* 42(3), 336-351.
9. Pang, K.-W., Zhou, X., Li, C.-L., 2010. Ship routing problem with berthing time clash avoidance constraints. *Working Paper.*
10. Brønmo, G., Nygreen, B., Lysgaard, J., 2010. Column generation approaches to ship scheduling with flexible cargo sizes. *European Journal of Operational Research* 200(1), 139-150.
11. Chuang, T.-N., Lin, C.-T., Kung J.-Y. and Lin, M.-D., 2010. Planning the route of container ships: A fuzzy genetic approach. *Expert Systems with Applications* 37(2010), 2948-2956.

12. Savelsbergh, M., Sol, M., 1998. DRIVE: Dynamic routing of independent vehicles. *Operations Research* 46(4), 474-490.
13. Dumas, Y., Desrosiers, J., Soumis, F., 1991. The pickup and delivery problem with time windows. *European Journal of Operational Research* 54(1), 7-22.
14. Pesant, G., Gendreau, M., Potvin, J.V., Rousseau, J.M., 1998. An exact constraint logic programming algorithm for the traveling salesman problem with time windows. *Transportation Science* 32(1), 12-29.
15. Junker, U., Karisch, S.E., Kohl, N., Vaaben, B., Fahle, T., Sellmann, M., 1999. A framework for constraint programming based column generation. *Principles and Practice of Constraint Programming, Lecture Notes in Computer Science*, Vol. 1713, 261-274.
16. Irnich, S., Desaulniers G., 2005. Shortest path problems with resource constraints. *Column Generation*, Ch. 2, 33-65. Springer, New York.
17. Rousseau, L.M., Gendreau M., Pesant G., 2004. Solving VRPTWs with constraint programming based column generation. *Annals of Operations Research* 130, 199-216.