

Research Article

Optimizing Ship Speed to Minimize Total Fuel Consumption with Multiple Time Windows

Jae-Gon Kim,¹ Hwa-Joong Kim,² Hong Bae Jun,³ and Chong-Man Kim⁴

¹Department of Industrial and Management Engineering, Incheon National University, Incheon, Republic of Korea

²Graduate School of Logistics, Inha University, Incheon, Republic of Korea

³Department of Industrial Engineering, Hongik University, Seoul, Republic of Korea

⁴Department of Industrial and Management Engineering, Myongji University, Yongin, Republic of Korea

Correspondence should be addressed to Chong-Man Kim; chongman@mju.ac.kr

Received 8 April 2016; Accepted 22 September 2016

Academic Editor: Miguel A. Salido

Copyright © 2016 Jae-Gon Kim et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We study the ship speed optimization problem with the objective of minimizing the total fuel consumption. We consider multiple time windows for each port call as constraints and formulate the problem as a nonlinear mixed integer program. We derive intrinsic properties of the problem and develop an exact algorithm based on the properties. Computational experiments show that the suggested algorithm is very efficient in finding an optimal solution.

1. Introduction

According to the report of World Shipping Council in 2008, fuel cost represents as much as 50–60% of total ship operating cost. Since fuel consumption is known to be the third power function of ship speed [1], many global shipping companies are trying to reduce fuel consumption by slowing down ship speed (called slow steaming). In this study, we consider the ship speed optimization problem with the objective of minimizing total fuel consumption of a (tramp or liner) ship operated on a given route. For a tramp ship, the ship speed optimization problem is a tactical problem which should be solved for every sailing, while it is a strategic problem to be solved just one time when designing a shipping route for a liner ship [2]. We determine ship speed on each leg under time window restrictions related with port calls. Time window sizes are narrow for congested ports, while they are wide for noncongested ones.

There are two kinds of time windows: hard time window and soft one [3, 4]. A hard time window should be kept at all costs while the soft one can be violated with appropriate penalties. There usually exist multiple time windows for each port call depending on the available service time of the port.

Most ports have restricted operating hours since they are closed for service at night and during weekends. In this case, the wide time windows can be regarded as multiple time windows [5]. Also, ports have restrictions on the draft of ships that may safely enter [6, 7]. Many ports have time-dependent draft restrictions due to the tide that leads to multiple time windows at each port. Therefore, our aim in this study is to develop a mathematical model and an exact algorithm for the ship speed optimization problem with multiple hard time windows.

There are some previous studies on the ship speed optimization problem which are related to our problem although they consider different objective functions, decision variables, and constraints. Ronen [1] performs pioneering research on determining optimal ship speed by considering the tradeoff between fuel savings by slow steaming and the loss of revenues due to the resulting voyage extension on the other hand. Ting and Tzeng [8] propose a dynamic programming model for the ship scheduling problem with both soft and hard time window constraints with the objective of meeting the time window constraints as closely as possible. Brown et al. [9] suggest a linear programming model for optimizing operation modes of a naval ship to minimize

the fuel consumption when the ship transit time is given. Corbett et al. [10] evaluate the effect of the speed reduction on mitigation of CO₂ emissions quantitatively. Ronen [11] determines the optimal speed and fleet size (the number of deployed ships) for containerships on a single route with a weekly service cycle. Wang and Meng [12] propose a model and an algorithm to determine an optimal sailing speed of container ships on each leg of each ship route in a liner shipping network while considering transshipment and container routing. Previous literature on the speed optimization problem can also be found in the airline industry such as Lovegren and Hansman [13], Jensen et al. [14], and Aktürk et al. [15].

Our study stems from the ship speed optimization problem to minimize total fuel consumption with hard time windows studied by Fagerholt et al. [16], Norstad et al. [17], Hvattum et al. [18], Kim et al. [2], and Zhang et al. [19]. Fagerholt et al. [16] formulate the problem as a nonlinear programming model and propose a heuristic algorithm by discretizing the time window and reformulating the problem as the shortest path model. Norstad et al. [17] develop a recursive smoothing algorithm (RSA) for speed optimization and present a multistart local search heuristic for the tramp ship routing and scheduling problem with speed optimization. Hvattum et al. [18] prove that the RSA of Norstad et al. [17] guarantees optimality. Kim et al. [2] also propose an exact algorithm for the ship speed optimization problem. Zhang et al. [19] establish the optimality properties for the problem. Although the suggested algorithms and properties are very efficient for the problem with a single time window constraint, they cannot handle multiple time windows which are common in maritime transportation.

In this study, we extend the studies of Hvattum et al. [18] and Kim et al. [2] to solve a more general ship speed optimization problem with multiple time window constraints. The remainder of this paper is organized as follows: the next section presents a mathematical formulation of the problem with a small example problem. In Section 3, we derive optimality properties of the problem. We present an exact algorithm based on the optimality properties in Section 4. Section 5 reports the computational test results on randomly generated test instances followed by concluding remarks in the last section.

2. Problem Description

The notations used throughout the paper are shown in the Notations.

The daily fuel consumption function $f(v)$ can be approximated by a well-known cubic function of speed as $f(v) = c \cdot v^3$, where c is a factor of converting speed to the fuel consumption [1]. Then, the total amount of fuel consumption of a vessel sailing from port i to port $i + 1$ can be expressed as

$$f(v_{i,i+1}) \cdot \frac{d_{i,i+1}}{v_{i,i+1}} = c \cdot d_{i,i+1} \cdot v_{i,i+1}^2. \quad (1)$$

Using (1), the considered problem can be formulated as the following nonlinear mixed integer programming model. We set $t_1 = 0$ since port 1 is the start port.

$$[P] \quad \text{minimize} \quad Z = c \cdot \sum_{i=1}^{n-1} d_{i,i+1} \cdot v_{i,i+1}^2 \quad (2)$$

$$\text{subject to} \quad t_i + p_i + \frac{d_{i,i+1}}{v_{i,i+1}} \leq t_{i+1} \quad (3)$$

for $i = 1, 2, \dots, n-1$

$$\sum_{j=1}^{m_i} z_{ij} = 1 \quad \text{for } i = 2, \dots, n \quad (4)$$

$$\sum_{j=1}^{m_i} z_{ij} \cdot s_i^j \leq t_i \leq \sum_{j=1}^{m_i} z_{ij} \cdot e_i^j \quad (5)$$

for $i = 2, \dots, n$

$$v^{\min} \leq v_{i,i+1} \leq v^{\max} \quad (6)$$

for $i = 1, 2, \dots, n-1$

$$t_i \geq 0 \quad \text{for } i = 1, \dots, n \quad (7)$$

$$z_{ij} \in \{0, 1\} \quad (8)$$

for $i = 2, \dots, n, j = 1, \dots, m_i$.

Objective function (2) expresses the total amount of fuel consumption of the ship. Constraint (3) ensures that cargo service at each port starts after the ship arrives at the port. Constraints (4) and (5) ensure that the start time of cargo services lies in exactly one of the time windows at each port. Constraint (6) represents that the ship speed on a leg is bounded by its allowable minimum and maximum speeds. Constraints (7) and (8) specify feasible ranges of decision variables.

A small example with four ports is used throughout the paper to help readers understand the problem and suggested properties and algorithm. In the example, $c = 0.01$, $v^{\min} = 10$, $v^{\max} = 20$, and cargo service time is assumed to be zero at all ports. Figure 1 depicts time windows at ports, distances between ports, and a feasible solution of the example. In the feasible solution, $v_{1,2} = 13.33$, $v_{2,3} = 20.00$, and $v_{3,4} = 13.33$.

3. Optimality Properties

We define an *outer time window* at a port as a single time window that covers all time windows at the port. Let $[s_i, e_i]$ be the outer time window at port i . Then, we have $[s_i, e_i] = [s_i^1, e_i^{m_i}]$. Using the time window reduction method of Kim et al. [2], we can reduce the size of the outer time window at each port as much as possible without affecting the feasible region of the problem. In the method, the size of the outer time window at a port is reduced by increasing the start time of the outer time window to the earliest possible

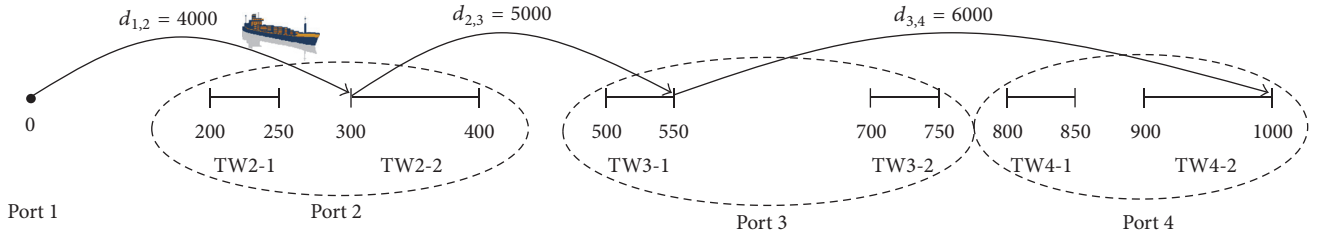


FIGURE 1: Multiple time windows, distance between ports, and feasible arrival time of the ship at each port for the four-port example.

time at which the ship can arrive at the port, while decreasing the end time of the outer time window to the latest possible time at which the ship can arrive. In the remainder of this paper, we assume that $[s_i, e_i]$ was already reduced using the time window reduction method of Kim et al. [2]. Then, the j th time window at port i , $[s_i^j, e_i^j]$, can be updated as follows for $i = 2, \dots, n$ and $j = 1, \dots, m_i$.

- (i) If $s_i > e_i^j$ or $e_i < s_i^j$, the j th time window of port i is no longer valid; thus it is eliminated.
- (ii) If $s_i^j < s_i \leq e_i^j$, let $s_i^j \leftarrow s_i$.
- (iii) If $s_i^j < e_i \leq e_i^j$, let $e_i^j \leftarrow e_i$.

In the remainder of this paper, we assume that $[s_i^j, e_i^j]$ and m_i were already updated. In the four-port example, time windows at each port cannot be reduced any longer.

Proposition 1. An optimal value of $t_n, t_n^* = e_n$.

Proof. We refer readers to Proposition 4 of Kim et al. [2]. \square

Proposition 2. If the time window constraints and the ship speed limits are relaxed, the optimal ship speed, v^R , is a constant along the entire route and obtained as $v^R = \sum_{i=1}^{n-1} d_{i,i+1} / (t_n^* - \sum_{i=1}^{n-1} p_i)$.

Proof. We refer readers to Proposition 1 of Hvattum et al. [18] and Proposition 5 of Kim et al. [2]. \square

Let t_i^R be the time at which service starts at port i and Ψ be a set of ports at which start time of service is not within any of the ports' time windows, when the ship sails at a constant speed v^R . If $i \in \Psi$, then $t_i^R < s_i^1$ or $e_i^j < t_i^R < s_i^{j+1}$ for $1 \leq j < m_i$ or $t_i^R > e_i^{m_i}$. For $i \in \Psi$, we define left and right time gaps of infeasibility at port i , denoted by Δ_i^L and Δ_i^R , respectively, as follows.

- If $t_i^R < s_i^1$, $\Delta_i^R = s_i^1 - t_i^R$ and Δ_i^L does not exist.
- If $e_i^j < t_i^R < s_i^{j+1}$ for $1 \leq j < m_i$, $\Delta_i^L = t_i^R - e_i^j$ and $\Delta_i^R = s_i^{j+1} - t_i^R$.
- If $t_i^R > e_i^{m_i}$, $\Delta_i^L = t_i^R - e_i^{m_i}$ and Δ_i^R does not exist.

Proposition 3. If $\Psi = \emptyset$, v^R is an optimal ship speed throughout the entire route.

Proof. According to Proposition 2, it is obvious. \square

Proposition 4. If $\Psi \neq \emptyset$, there exists at least one port (say port k) in Ψ for which $\max(\Delta_k^L, \Delta_k^R) \geq \min(\Delta_i^L, \Delta_i^R)$ for all $i \in \Psi$ and an optimal value of t_k , denoted by t_k^* , is obtained as follows:

$$t_k^* = \begin{cases} s_k^1, & \text{if } t_k^R < s_k^1, \\ e_k^j \text{ or } s_k^{j+1}, & \text{if } e_k^j < t_k^R < s_k^{j+1}, \text{ where } 1 \leq j < m_k, \\ e_k^{m_k}, & \text{if } t_k^R > e_k^{m_k}. \end{cases} \quad (9)$$

Proof. If $t_i^R < s_i^1$ or $t_i^R > e_i^{m_i}$ (i.e., the outer time window is violated at port i), it is clear that the time window violation should be adjusted by moving t_i to the right or left, respectively. On the other hand, if $e_i^j < t_i^R < s_i^{j+1}$ (i.e., the inner time window is violated at port i), there exist two possible directions of violation adjustment, that is, left or right. If $e_i^j < t_i^R < s_i^{j+1}$, we assume that $t_i^* \leq e_i^j$ or $t_i^* \geq s_i^{j+1}$ arbitrarily. Let Y be the total number of ports at which inner time windows are violated. Then, there exist total 2^Y possible cases to consider. For each case, we define a time gap of infeasibility at port i , denoted by Δ_i , for all $i \in \Psi$ as follows.

$$\text{If } t_i^R < s_i^1, \Delta_i = \Delta_i^R = s_i^1 - t_i^R.$$

$$\text{If } e_i^j < t_i^R < s_i^{j+1} \text{ and } t_i^* \leq e_i^j, \Delta_i = \Delta_i^L = t_i^R - e_i^j, \text{ where } 1 \leq j < m_i.$$

$$\text{If } e_i^j < t_i^R < s_i^{j+1} \text{ and } t_i^* \geq s_i^{j+1}, \Delta_i = \Delta_i^R = s_i^{j+1} - t_i^R, \text{ where } 1 \leq j < m_i.$$

$$\text{If } t_i^R > e_i^{m_i}, \Delta_i = \Delta_i^L = t_i^R - e_i^{m_i}.$$

If $t_i^* \leq e_i^j$ and $e_i^j < t_i^R < s_i^{j+1}$, it means that the violation should be adjusted to the left, while the violation should be adjusted to the right if $t_i^* \geq s_i^{j+1}$. Since it is known whether $t_i^* \leq e_i^j$ or $t_i^* \geq s_i^{j+1}$ if $e_i^j < t_i^R < s_i^{j+1}$ in each of the 2^Y cases, violation adjustment direction is also determined for all violations in each case. This means that the inner time window violation can be treated as the outer time window violation in each of the 2^Y cases. According to Propositions 2 and 3 of Hvattum et al. [18], the (outer) time window constraint with the largest violation should be binding. Let $k = \arg \max_{i \in \Psi} (\Delta_i)$. Then,

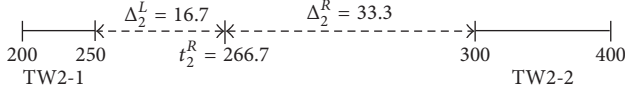


FIGURE 2: Occurrence of left and right time gaps of infeasibility at port 2 resulting from the constant ship speed v^R in the four-port example.

$\Delta_k \geq \min(\Delta_i^L, \Delta_i^R)$ for all $i \in \Psi$. By applying Propositions 2 and 3 of Hvattum et al. [18] to each of the 2^Y cases, we have

$$t_k^* = \begin{cases} s_k^1, & \text{if } t_k^R < s_k^1, \\ e_k^j, & \text{if } e_k^j < t_k^R < s_k^{j+1}, t_k^* \leq e_k^j, \text{ where } 1 \leq j < m_k, \\ s_k^{j+1}, & \text{if } e_k^j < t_k^R < s_k^{j+1}, t_k^* \geq s_k^{j+1}, \text{ where } 1 \leq j < m_k, \\ e_k^{m_k}, & \text{if } t_k^R > e_k^{m_k}. \end{cases} \quad (10)$$

An optimal solution belongs to one of the 2^Y cases and $\max(\Delta_k^L, \Delta_k^R) \geq \Delta_k$ in any case. Therefore, there exists at least one port k for which $\max(\Delta_k^L, \Delta_k^R) \geq \min(\Delta_i^L, \Delta_i^R)$ for all $i \in \Psi$ and

$$t_k^* = \begin{cases} s_k^1, & \text{if } t_k^R < s_k^1, \\ e_k^j \text{ or } s_k^{j+1}, & \text{if } e_k^j < t_k^R < s_k^{j+1}, \text{ where } 1 \leq j < m_k, \\ e_k^{m_k}, & \text{if } t_k^R > e_k^{m_k}. \end{cases} \quad (11)$$

□

In the four-port example, $t_4^* = 1000$ and $v^R = 15$. Since $t_2^R = 266.7$, $t_3^R = 600$, and $t_4^R = 1000$, we have $\Psi = \{2, 3\}$ and $\Delta_2^L = 16.7$, $\Delta_2^R = 33.3$ (as shown in Figure 2), $\Delta_3^L = 50$, and $\Delta_3^R = 100$. According to Proposition 4, $t_3^* = 550$ or 700 . Note that $t_2^* = 250$ or 300 is not ensured because $\max(\Delta_2^L, \Delta_2^R) < \min(\Delta_3^L, \Delta_3^R)$.

4. An Exact Algorithm

Based on the proposed propositions, we develop a recursive partitioning algorithm (RPA) to find an optimal solution for [P]. The RPA is an extension of the RSA of Norstad et al. [17] and Hvattum et al. [18]. The basic idea of the algorithm is to maintain a constant ship speed throughout the entire route as best as possible and divide the ship route into two subroutes recursively if start time of service is not within any of the ports' time windows by fixing start time of service at one of its optimal candidates obtained by Proposition 4. In the algorithm, all possible optimal partitions are evaluated and the best one is selected. Pseudocode of the RPA is given as shown in Algorithm 1.

The RPA can be seen as a tree search algorithm as the RSA of Norstad et al. [17] in that it recursively partitions the problem into two subproblems at each node. In the RSA, the way of optimally partitioning the problem is unique in each

node; that is, the problem is divided into two subproblems by binding the largest violation, while possible candidates of optimal partitioning are up to $2^{|\Psi|}$ for the RPA in each node. Since $|\Psi| \leq n - 2 - i$ in level i , where $i = 0, 1, \dots, n - 2$, the total number of possible ways of partitioning is up to $2^{n-2}(n-2)!$. According to Hvattum et al. [18], the RSA, a one-way partitioning algorithm, has the worst-case running time of $O(n^2)$. The worst-case running time of the RPA is $2^{n-2}(n-2)!$ times that of the RSA. Therefore, the RPA has the worst-case running time of $O(n!)$. Although the RPA has a very bad worst-case performance, it is expected to perform well in practice because $|\Psi|$ is usually much less than n in each level (especially in high levels) and furthermore we only need to consider $k \in \Psi$, where $\max(\Delta_k^L, \Delta_k^R) \geq \min(\Delta_i^L, \Delta_i^R)$ according to Proposition 4, which is validated through computational experiments in the next section.

For the four-port example, RPA(1, 4, 0, 1000) is executed at first to obtain $v^R = 15$ in line (1), $t_2 = 266.7$, $t_3 = 600$, and $t_4 = 1000$ in lines (4)~(5). Since $e_2^1 (=250) < t_2 < s_2^2 (=300)$ and $e_3^1 (=550) < t_3 < s_3^2 (=700)$, we obtain $\Psi = \{2, 3\}$ in lines (7)~(9), and $\Delta_2^L = 16.7$, $\Delta_2^R = 33.3$, $\Delta_3^L = 50$, and $\Delta_3^R = 100$ in lines (17)~(23). Since $\max(\Delta_2^L, \Delta_2^R) < \min(\Delta_3^L, \Delta_3^R)$, Partition(\cdot) is not executed for $k = 2$ in line (25). For $k = 3$, Partition(1, 3, 4, 550) is executed in line (33) and RPA(1, 3, 0, 550) and RPA(3, 4, 550, 1000) are recursively called in line (41). In RPA(1, 3, 0, 550), $v^R = v_{1,2} = v_{2,3} = 16.36$, $t_2 = 266.7$, $t_3 = 550$, and $\Psi = \emptyset$. RPA(1, 3, 0, 550) returns 24,099.2 in line (14). In RPA(3, 4, 550, 1000), $v^R = v_{3,4} = 13.33$, $t_3 = 550$, $t_4 = 1000$, and $\Psi = \emptyset$. RPA(3, 4, 550, 1000) returns 10,666.6. Then, Partition(1, 3, 4, 550) has Z of 34,765.8 in line (41) and the variables \min and $v_{i,i+1}^{\min}$ s are updated in lines (42)~(45). On the other hand, Partition(1, 3, 4, 700) is executed in line (35) and has Z of 38,877.6 after executing RPA(1, 3, 0, 700) and RPA(3, 4, 700, 1000). In Partition(1, 3, 4, 700), \min and $v_{i,i+1}^{\min}$ are not updated since $Z (=38,877.6)$ is greater than $\min (=34,765.8)$. Lastly, RPA(1, 4, 0, 1000) obtains an optimal solution in which $v_{1,2} = 16.36$, $v_{2,3} = 16.36$, $v_{3,4} = 13.33$, $t_2 = 244.50$, $t_3 = 550$, and $t_4 = 1000$ in lines (36)~(39) and returns the objective function value of the optimal solution ($=34,765.8$) in line (40).

5. Computational Experiments

To test performance of the RPA, 25 test instances were randomly generated. We modified the problem generation method of Kim et al. [2] to generate the test instances. In our method, parameters of the test instances were set as follows:

- (1) The number of ports in the route, n , is set to five levels: 10, 20, 30, 40, and 50. (Very large-sized problem instances are included in the test set to gain deeper insights into the performance of the suggested algorithm although the number of port calls is usually less than 20 in the maritime industry.)
- (2) The maximum number of time windows (MNTW) at each port is set to five levels: 2, 4, 6, 8, and 10. (At each

```

RPA( $s, e, s_s, e_e$ )
(1)  $v^R \leftarrow \sum_{i=s}^{e-1} d_{i,i+1} / (e_e - s_s - \sum_{i=1}^{e-1} p_i)$ 
(2) for  $i \leftarrow s$  to  $e - 1$ 
(3)    $v_{i,i+1} \leftarrow v^R$ 
(4) for  $i \leftarrow s + 1$  to  $e$ 
(5)    $t_i \leftarrow t_{i-1} + p_{i-1} + d_{i-1,i} / v_{i-1,i}$ 
(6)  $\Psi \leftarrow \emptyset$ 
(7) for  $i \leftarrow s$  to  $e$ 
(8)   if there does not exist any  $j$  such that  $s_i^j \leq t_i \leq e_i^j$  for all  $j$ 
(9)      $\Psi \leftarrow \Psi \cup \{i\}$ 
(10) if  $\Psi = \emptyset$ 
(11)   total fuel consumption  $\leftarrow 0$ 
(12)   for  $i \leftarrow s$  to  $e - 1$ 
(13)      $Z \leftarrow Z + c \cdot d_{i,i+1} \cdot v_{i,i+1}^2$ 
(14)   return( $Z$ )
(15) if  $\Psi \neq \emptyset$ 
(16)   min  $\leftarrow \infty$ 
(17)   for  $i \in \Psi$ 
(18)     if  $t_i < s_i^1$ 
(19)        $\Delta_i^L \leftarrow 0$  and  $\Delta_i^R \leftarrow s_i^1 - t_i$ 
(20)     else if  $t_i > e_i^{m_i}$ 
(21)        $\Delta_i^L \leftarrow t_i - e_i^{m_i}$  and  $\Delta_i^R \leftarrow 0$ 
(22)     else
(23)        $\Delta_i^L \leftarrow t_i - e_i^j$  and  $\Delta_i^R \leftarrow s_i^{j+1} - t_i$ 
(24)   for  $k \in \Psi$ 
(25)     if  $\max(\Delta_k^L, \Delta_k^R) \geq \min(\Delta_i^L, \Delta_i^R)$  for all  $i \in \Psi$ 
(26)       if  $t_k < s_k^1$ 
(27)          $t_k \leftarrow s_k^1$  and Partition( $s, k, e, t_k$ )
(28)       else if  $t_k > e_k^{m_k}$ 
(29)          $t_k \leftarrow e_k^{m_k}$  and Partition( $s, k, e, t_k$ )
(30)       else
(31)         for  $h \leftarrow 1$  to  $2$ 
(32)           if  $h = 1$ 
(33)              $t_k \leftarrow e_k^j$  and Partition( $s, k, e, t_k$ )
(34)           else
(35)              $t_k \leftarrow s_k^{j+1}$  and Partition( $s, k, e, t_k$ )
(36)   for  $i \leftarrow s$  to  $e - 1$ 
(37)      $v_{i,i+1} \leftarrow v_{i,i+1}^{\min}$ 
(38)   for  $i \leftarrow s + 1$  to  $e$ 
(39)      $t_i \leftarrow t_{i-1} + p_{i-1} + d_{i-1,i} / v_{i-1,i}$ 
(40)   return(min)

Partition( $s, k, e, t_k$ )
(41)  $Z \leftarrow \text{RPA}(s, k, s_s, t_k) + \text{RPA}(k, e, t_k, e_e)$ 
(42) if ( $Z < \min$ )
(43)   min  $\leftarrow Z$ 
(44)   for  $i \leftarrow s$  to  $e - 1$ 
(45)      $v_{i,i+1}^{\min} = v_{i,i+1}$ 

```

ALGORITHM 1

- port, the number of time windows is between 1 and MNTW.)
- (3) v^{\min} and v^{\max} are set to 12 knots and 25 knots, respectively.
 - (4) $d_{i,i+1}$ is randomly generated between 500 and 2,000 nautical miles for $1 \leq i \leq n - 1$, corresponding to about 1~11 sailing days when sailing at 18.5 knots.
 - (5) The factor of converting speed to the fuel consumption, c , is set to 0.02.
 - (6) The outer time window size of port i , $e_i^{m_i} - s_i^1$, is set to 120 hours for all i .
 - (7) $p_i \sim U(24, 48)$, where $U(a, b)$ is a randomly generated real value using a uniform distribution from a to b .

TABLE 1: CPU time and the total number of recursive calls required for the RPA to find optimal solutions.

MNTW	$n = 10$	$n = 20$	$n = 30$	$n = 40$	$n = 50$
2	0.000 [†] (5) [‡]	0.000 (155)	0.015 (3861)	0.047 (24374)	0.670 (410026)
4	0.000 (86)	0.000 (575)	0.016 (9728)	0.063 (51727)	3.994 (2584703)
6	0.000 (37)	0.000 (186)	0.000 (2464)	0.063 (49175)	0.156 (101315)
8	0.000 (13)	0.000 (456)	0.000 (707)	0.047 (20931)	0.047 (21799)
10	0.000 (15)	0.000 (302)	0.015 (9255)	0.016 (7344)	0.031 (15801)

†: the CPU time (seconds).

‡: the total number of recursive calls.

$$(8) s_i = s_{i-1} + p_{i-1} + d_{i-1,i}/v^{\max} \text{ and } e_i = s_i + 120 \text{ for all } i \geq 2, \text{ where } s_1 = e_1 = 1.$$

$$(9) s_i^1 = s_i \text{ and } s_i^j = e_i^{j-1} + a_i/(m_i - 1) \text{ for all } i \text{ and } j \geq 2, \text{ where } a_i = 120/(m_i + 1).$$

$$(10) \text{ If } m_i = 1, e_i^1 = e_i. \text{ Otherwise, } e_i^j = s_i^j + a_i \text{ for all } i \text{ and } j.$$

The RPA was run on a PC with an Intel i5 CPU at 3.3 GHz. Table 1 shows CPU time required by the algorithm and the total number of recursive calls in the algorithm to find an optimal solution for each instance. The RPA was very fast; it required only 0.98 seconds on the average even for the largest-sized instances. The CPU time was affected by the number of ports but not by the MNTW. This is because the CPU time is directly related to the total number of recursive calls of RPA(\cdot) which is again related to $|\Psi|$, the number of ports at which start time of service is not within any of the ports' time windows when the ship sails at a constant speed v^R .

6. Concluding Remarks

In this study, we suggest a nonlinear mixed integer programming model for the ship speed optimization problem with multiple time windows. We extend existing literature to derive optimality properties and develop an exact algorithm for the problem. Computational experiments show that the suggested algorithm is very fast in finding optimal solutions even for very large-sized problem instances. The suggested algorithm itself can be used to determine the economic ship speed and as a subroutine in the ship routing and scheduling algorithms in the shipping industry.

Notations

Parameters

- n : Number of ports on the route
- $d_{i,i+1}$: Nautical distance between port i and port $i + 1$ [nautical miles]
- $f(v)$: Daily fuel consumption of a ship navigating in speed v [tons/day]
- m_i : Number of time windows at port i

- s_i^j : Start time of the j th time window at port i
- e_i^j : End time of the j th time window at port i
- p_i : Cargo service time at port i
- v^{\min} : Minimum allowable speed of the ship [knots]
- v^{\max} : Maximum allowable speed of the ship [knots].

Decision Variables

- $v_{i,i+1}$: Speed of the ship sailing from port i to port $i + 1$ [knots]
- t_i : Time at which service starts at port i
- $z_{i,j}$: Equalling 1 if service starts at port i within the j th time window and 0 otherwise.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by 2012 Assistant Professor Support Program of Engineering College, Myongji University.

References

- [1] D. Ronen, "Effect of oil price on the optimal speed of ships," *Journal of the Operational Research Society*, vol. 33, no. 11, pp. 1035–1040, 1982.
- [2] J.-G. Kim, H.-J. Kim, and P. T.-W. Lee, "Optimizing ship speed to minimize fuel consumption," *Transportation Letters*, vol. 6, no. 3, pp. 109–117, 2014.
- [3] K. Fagerholt, "Evaluating the trade-off between the level of customer service and transportation costs in a ship scheduling problem," *Maritime Policy and Management*, vol. 27, no. 2, pp. 145–153, 2000.
- [4] K. Fagerholt, "Ship scheduling with soft time windows: an optimisation based approach," *European Journal of Operational Research*, vol. 131, no. 3, pp. 559–571, 2001.
- [5] M. Christiansen and K. Fagerholt, "Robust ship scheduling with multiple time windows," *Naval Research Logistics*, vol. 49, no. 6, pp. 611–625, 2002.

- [6] J. G. Rakke, M. Christiansen, K. Fagerholt, and G. Laporte, "The traveling salesman problem with draft limits," *Computers & Operations Research*, vol. 39, no. 9, pp. 2161–2167, 2012.
- [7] J.-H. Song and K. C. Furman, "A maritime inventory routing problem: practical approach," *Computers & Operations Research*, vol. 40, no. 3, pp. 657–665, 2013.
- [8] S. C. Ting and G. H. Tzeng, "Ship scheduling and service network integration for liner shipping companies and strategic alliances," *Journal of the Eastern Asia Society for Transportation Studies*, vol. 5, pp. 765–777, 2003.
- [9] G. G. Brown, J. E. Kline, R. E. Rosenthal, and A. R. Washburn, "Steaming on convex hulls," *Interfaces*, vol. 37, no. 4, pp. 342–352, 2007.
- [10] J. J. Corbett, H. Wang, and J. J. Winebrake, "The effectiveness and costs of speed reductions on emissions from international shipping," *Transportation Research Part D: Transport and Environment*, vol. 14, no. 8, pp. 593–598, 2009.
- [11] D. Ronen, "The effect of oil price on containership speed and fleet size," *Journal of the Operational Research Society*, vol. 62, no. 1, pp. 211–216, 2011.
- [12] S. Wang and Q. Meng, "Sailing speed optimization for container ships in a liner shipping network," *Transportation Research Part E: Logistics and Transportation Review*, vol. 48, no. 3, pp. 701–714, 2012.
- [13] J. A. Lovegren and R. J. Hansman, "Estimation of potential aircraft fuel burn reduction in cruise via speed and altitude optimization strategies," Tech. Rep. ICAT-2011-03, MIT International Center for Air Transportation, 2011.
- [14] L. Jensen, R. J. Hansman, J. Venuti, and T. G. Reynolds, "Commercial airline altitude optimization strategies for reduced cruise fuel consumption," in *Proceedings of the 14th AIAA Aviation Technology, Integration, and Operations Conference (ATIO '14)*, AIAA 2014-3006, Atlanta, Ga, USA, June 2014.
- [15] M. S. Aktürk, A. Atamtürk, and S. Gürel, "Aircraft rescheduling with cruise speed control," *Operations Research*, vol. 62, no. 4, pp. 829–845, 2014.
- [16] K. Fagerholt, G. Laporte, and I. Norstad, "Reducing fuel emissions by optimizing speed on shipping routes," *Journal of the Operational Research Society*, vol. 61, no. 3, pp. 523–529, 2010.
- [17] I. Norstad, K. Fagerholt, and G. Laporte, "Tramp ship routing and scheduling with speed optimization," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 5, pp. 853–865, 2011.
- [18] L. M. Hvattum, I. Norstad, K. Fagerholt, and G. Laporte, "Analysis of an exact algorithm for the vessel speed optimization problem," *Networks*, vol. 62, no. 2, pp. 132–135, 2013.
- [19] Z. Zhang, C.-C. Teo, and X. Wang, "Optimality properties of speed optimization for a vessel operating with time window constraint," *Journal of the Operational Research Society*, vol. 66, no. 4, pp. 637–646, 2015.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

