Analysis of an Exact Algorithm for the Vessel Speed Optimization Problem

Lars Magnus Hvattum
Department of Industrial Economics and Technology Management, Norwegian University of Science and Technology, Trondheim, Norway

Inge Norstad
Department of Industrial Economics and Technology Management, Norwegian University of Science and Technology, Trondheim, Norway
Norwegian Marine Technology Research Institute (MARINTEK), Trondheim, Norway

Kjetil Fagerholt
Department of Industrial Economics and Technology Management, Norwegian University of Science and Technology, Trondheim, Norway

Gilbert Laporte
Canada Research Chair in Distribution Management, HEC Montréal, Montréal, Canada

Increased fuel costs together with environmental concerns have led shipping companies to consider the optimization of vessel speeds. Given a fixed sequence of port calls, each with a time window, and fuel cost as a convex function of vessel speed, we show that optimal speeds can be found in quadratic time. NETWORKS, Vol. 62(2), 132–135 2013

Keywords: maritime transportation; fuel emission; scheduling

1. INTRODUCTION

Fuel consumption in international shipping has increased by ~50% over the last 10 years. The International Maritime Organization has estimated that in 2007 the total fuel consumption in international shipping was 277 million tonnes [4]. Given a fuel oil price of 450 USD/tonne, a 1% worldwide reduction in the fuel consumption would yield cost reductions of more than 1.2 billion USD and a reduction in CO_2 emissions of 10.5 million tonnes. In [2], computational studies on real world ship routing and scheduling problems show that including speed as a decision variable in route optimization reduces fuel consumption by 14% on average, compared with using fixed speeds. The effect of speed optimization on routing costs and environmental emissions is, therefore, considerable.

In the speed optimization problem (SOP) [1], we are given a sequence $i = 0, \ldots, n$ of nodes corresponding to port calls made by a ship. A time window $[\underline{t}_i, \bar{t}_i]$ is associated with each node i. Variable t_i indicates the arrival time at node $i = 1, \ldots, n-1$. The arrival time at node 0 is $t_0 = \underline{t}_0 = \bar{t}_0$ and the arrival time at node n is $t_n = \underline{t}_n = \bar{t}_n$. Let $d_i > 0$ be the distance from node i to node $i+1$, and let $v_i = d_i/(t_{i+1} - t_i)$ be the travel speed from i to $i+1$. The ship has a minimum and maximum cruising speed, denoted by $\underline{v}$ and $\bar{v}$, and the fuel cost per distance unit is described by a function $c(v_i)$. Without loss of generality, service times at the ports may be ignored. The SOP can be described by the following nonlinear model:

$$\text{minimize } z = \sum_{i=0}^{n-1} d_i c(v_i) \tag{1}$$

Received September 2010; accepted October 2012
Following the initial online publication of this article, several errors were subsequently identified. This notice is included in the online and print versions to indicate that both have been corrected 24 June 2013.
Correspondence to: L. M. Hvattum; e-mail: lars.m.hvattum@iot.ntnu.no
Contract grant sponsor: Research Council of Norway (DESIMAL Project)
Contract grant sponsor: Canadian Natural Sciences and Engineering Research Council; Contract grant number: 39682-10
DOI 10.1002/net.21503
Published online 4 March 2013 in Wiley Online Library (wileyonlinelibrary.com).

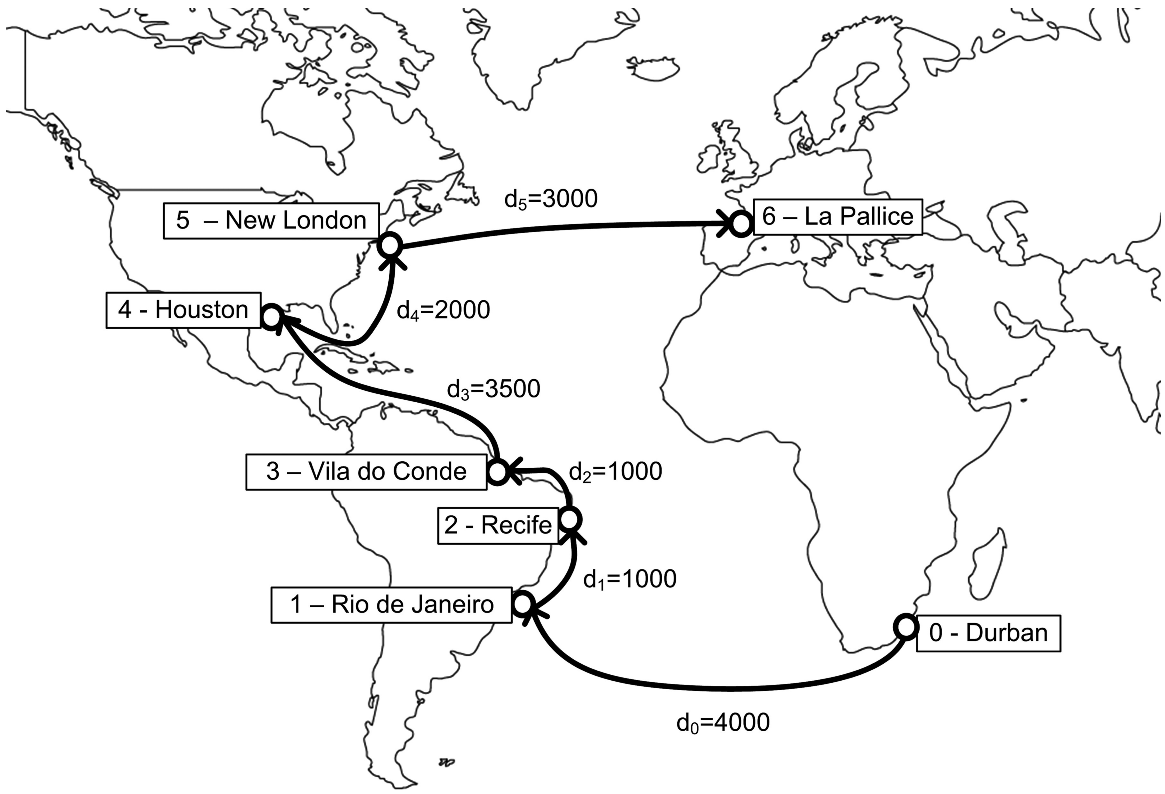

FIG. 1. Example of a shipping route with seven port calls.

subject to

$$t_{i+1} - t_i - d_i/v_i = 0 \qquad i = 0, \ldots, n-1 \qquad (2)$$

$$\underline{t}_i \leq t_i \leq \bar{t}_i \qquad i = 0, \ldots, n \qquad (3)$$

$$0 < v_i \leq \bar{v} \qquad i = 0, \ldots, n-1. \qquad (4)$$

According to [1,3], as for most cargo ships fuel consumption per distance unit is approximately quadratic in speed over the domain $[\underline{v}, \bar{v}]$, the cost $c(v_i)$ per distance unit will be a continuous and convex function that is constant from 0 to $\hat{v} = \arg\min_{\underline{v} \leq v_i \leq \bar{v}} f(v_i)$, where $f(v_i)$ is the cost per distance unit as a function of actual cruising speed. In [2], this formed the basis for the following recursive algorithm for the SOP. Consider the subsequence $s, \ldots, e$, initially with $s = 0$ and $e = n$, where t_s and t_e are fixed. Calculate a constant speed v^{R} sufficient to reach node e at time t_e. If this speed does not violate any time window, it is optimal. Otherwise, consider any node p with the largest time window violation, that is $b_i \leq b_p$ for all $i = s+1, \ldots, e-1$, where $b_i = \max\{0, t_i - \bar{t}_i, \underline{t}_i - t_i\}$. Fix the time of arrival at p, so that it just falls within its time window, and repeat the algorithm recursively for $s, \ldots, p$ and for $p, \ldots, e$. This process is detailed in Algorithm 1.

In [2], this algorithm was shown to produce solutions with zero optimality gaps, but no proof of the correctness of the algorithm was provided. We give an example of the execution of the algorithm in Section 2 and prove that the algorithm is indeed exact when $c(v_i)$ is a convex and non-decreasing function on $[0, \bar{v}]$ in Section 3.

Algorithm 1 *OptimizeSpeed*(s, e)

$\beta \leftarrow 0$
$p \leftarrow 0$
$v^{\mathrm{R}} \leftarrow \sum_{i=s}^{e-1} d_i/(t_e - t_s)$
for $i \leftarrow s+1$ to e **do**
    $v_i \leftarrow v^{\mathrm{R}}$
    $t_i \leftarrow t_{i-1} + d_i/v_i$
    $b_i \leftarrow \max\{0, t_i - \bar{t}_i, \underline{t}_i - t_i\}$
    if $b_i > \beta$ **then**
        $\beta \leftarrow b_i$
        $p \leftarrow i$
    end if
end for
if $\beta > 0$ and $t_p > \bar{t}_p$ **then**
    $t_p \leftarrow \bar{t}_p$
    OptimizeSpeed(s, p)
    OptimizeSpeed(p, e)
end if
if $\beta > 0$ and $t_p < \underline{t}_p$ **then**
    $t_p \leftarrow \underline{t}_p$
    OptimizeSpeed(s, p)
    OptimizeSpeed(p, e)
end if

2. EXAMPLE

To illustrate the algorithm, we consider a shipping route with seven port calls. Figure 1 depicts the route of a cargo

TABLE 1. Numerical example.

				Iteration 1			Iteration 2			Iteration 3		
i	d_i	$\underline{t}_i$	$\bar{t}_i$	v_i	t_i	b_i	v_i	t_i	b_i	v_i	t_i	b_i
0	4,000	0	0	15.10	0.0		15.83	0.0		14.88	0.0	
1	1,000	10	12	15.10	11.0		15.83	10.5		14.88	11.0	
2	1,000	14	16	15.10	13.8	0.2	15.83	13.2	0.8*	17.05	14.0	
3	3,500	16	18	15.10	16.6		15.83	15.8	0.2	17.05	16.6	
4	2,000	22	25	15.10	26.2	1.2*	13.89	25.0		13.89	25.0	
5	3,000	30	32	15.10	31.7		13.89	31.0		13.89	31.0	
6		40	40		40.0			40.0			40.0	

ship starting in Durban, South Africa ($i = 0$) and ending in La Pallice, France ($i = 6$). The first columns of Table 1 show the nodes i, the distance d_i between node i and $i + 1$, and the earliest ($\underline{t}_i$) and the latest ($\bar{t}_i$) times at which the ship can start service at node i.

In the first iteration, the start node $s = 0$ and the end node $e = 6$ have respective arrival times $t_s = 0$ and $t_e = 40$. The ship sails 24 h/day and travels 14,500 miles over a 40-day period, yielding an average speed of 15.10 knots. The algorithm performs three iterations to reach the optimal solution. Table 1 shows for each iteration, the speeds, v_i, the arrival times, t_i, and the time window violations, b_i. The largest time window violation is identified by an asterisk.

After the first iteration, two of the nodes have infeasible arrival times. At node $i = 2$, the ship will arrive 0.2 days early, and at node $i = 4$ it will arrive 1.2 days late. The arrival at node $i = 4$, which has the largest time window violation, is therefore fixed to its nearest feasible value, $t_4 = 25$. The sequence of nodes is now split at $i = 4$. For the subsequence $0, \dots, 4$, the new optimal speed is 15.83, whereas for the subsequence $4, \dots, 6$ the optimal speed is 13.89.

After iteration 2, we see that the ship arrives too early at both nodes $i = 2$ and $i = 3$. As $b_2 \geq b_3$, we fix $t_2 = 14$ and split the sequence $i = 0, \dots, 4$ at $i = 2$. At iteration 3, the new optimal speed for the subsequence $0, \dots, 2$ is 14.88 knots, and for $2, \dots, 4$ it is 17.05. As every node now has a feasible arrival time, the algorithm terminates.

3. ANALYSIS OF ALGORITHM 1

In the following, denote the cumulative distance from node 0 to node i by $D_i = \sum_{j=0}^{i-1} d_j$.

Proposition 1. *Consider problem (1)–(4) in which constraints (3) are relaxed for $i = 1, \dots, n-1$. An optimal solution to the relaxed problem is obtained by taking $v_i = v^{\mathrm{R}}$ for $i = 0, \dots, n-1$, where $v^{\mathrm{R}} = D_n/(t_n - t_0)$.*

Proof. Fuel cost is a convex, continuous and non-decreasing function c of speed v per distance unit. Moreover, v is a step function taking the value v_0 over $[0, D_1]$ and the value v_i over each interval $(D_i, D_{i+1}]$, $i = 1, \dots, n-1$.

If v is not constant over $[0, D_n]$, then there must exist two indices $j, k \in \{0, \dots, n-1\}$ for which $v_k > v_j$. Because c is convex, its rate of growth is non-decreasing and therefore, for any δ satisfying $0 < \delta < v_k - v_j$, we have

$$\frac{c(v_j + \delta) - c(v_j)}{\delta} \leq \frac{c(v_k) - c(v_k - \delta)}{\delta}.$$

In other words, $c(v_j + \delta) + c(v_k - \delta) \leq c(v_j) + c(v_k)$. The inequality is strict when c is strictly convex on $[v_j, v_k]$.

Now consider a subinterval D_j^S of $(D_j, D_{j+1}]$ with length $d = \min\{d_j, d_k\}$, and a subinterval D_k^S of $(D_k, D_{k+1}]$ with the same length d. The time used to traverse these subintervals using the current speed is $T = \frac{d}{v_k} + \frac{d}{v_j} = \frac{d(v_k+v_j)}{v_k v_j}$. If we increase the speed on D_j^S by δ and decrease the speed on D_k^S by δ, the total cost does not increase, but the time used decreases to $T_\delta = \frac{d}{v_k-\delta} + \frac{d}{v_j+\delta} = \frac{d(v_k+v_j)}{v_k v_j + \delta(v_k - v_j) - \delta^2}$, where $T_\delta < T$ since $0 < \delta < v_k - v_j$. Since c is non-decreasing, this means that there exists an $0 < \epsilon < \delta$ such that we can keep the time taken to traverse the subintervals equal to T without increasing the total cost by using the speed $v_k - \delta - \epsilon$ on D_k^S and the speed $v_j + \delta - \epsilon$ on D_j^S.

That is, $c(v_j) + c(v_k) \geq c(v_j + \delta) + c(v_k - \delta) \geq c(v_j + \delta - \epsilon) + c(v_k - \delta - \epsilon)$ when c is convex and non-decreasing. Furthermore, at least one of the inequalities holds strictly if c is strictly increasing on $[v_j, v_k]$. This proves that there cannot exist any solution with a better cost than using a constant speed v on $[0, D_n]$. ■

Proposition 2. *Consider the sequence $0, \dots, n$ of nodes and an optimal solution to the relaxation defined in Proposition 1, with speeds $v_i^{\mathrm{R}} = D_n/(t_n - t_0)$ and arrival times t_i^{R}. Let $b_i^{\mathrm{R}} = \max\{0, t_i^{\mathrm{R}} - \bar{t}_i, \underline{t}_i - t_i^{\mathrm{R}}\}$, and consider p such that $b_i^{\mathrm{R}} \leq b_p^{\mathrm{R}}$ for $i = 1, \dots, n-1$. If $b_p^{\mathrm{R}} > 0$ and $t_p^{\mathrm{R}} > \bar{t}_p$, then there exists an optimal solution with $t_p = \bar{t}_p$.*

Proof. Assume that there is an optimal solution of (1)–(4) such that no arrival times can be moved to a later point without incurring a higher cost, and with $t_p = t_p^R - \Delta_p < \bar{t}_p$, implying that $\Delta_p > b_p^R$. Define the speed from node 0 to node i as $V_i = D_i/t_i$. Since $t_p < t_p^R$ we know that $V_p > v^R$. The proof considers separately the cases $v_{p-1} < v^R$ and $v_{p-1} \geq v^R$.

For $v_{p-1} < v^R$, consider $q < p$ such that $v_{q-1} > v^R$ and $v_i \leq v^R$ for $i = q+1, \ldots, p$. Such a q must exist, since $V_p > v^R$. Now, since $v_{q-1} > v_q$, by using the same argument as in the proof of Proposition 1, we can increase the speed on some subinterval of $(D_q, D_{q+1}]$ and decrease the speed on some subinterval of $(D_{q-1}, D_q]$ to obtain a solution of same duration and smaller cost. This is possible as long as $t_q < \bar{t}_q$. Hence, q is either not as defined, or we must have an optimal solution with $t_q = \bar{t}_q$. Then $t_q = t_q^R - \Delta_q = t_q^R - b_q^R$. However, since the speed from q to p is always less than v^R, it follows that $\Delta_q > \Delta_p$, which leads to the contradiction $b_q^R = \Delta_q > \Delta_p > b_p^R$.

For $v_{p-1} \geq v^R$, consider $q > p$ such that $v_q < v^R$ and $v_{i-1} \geq v^R$ for $i = p, \ldots, q$. Since $V_p > v^R$ and $V_n = v^R$, such a q must exist. Now, by using again the same argument as in the proof of Proposition 1, we must have $t_q = \bar{t}_q$ (and $p \neq q$) since $v_{q-1} > v_q$. Then $t_q = t_q^R - \Delta_q = t_q^R - b_q^R$. However, since the speed from p to q is never less than v^R, it follows that $\Delta_q \geq \Delta_p$, which leads to the contradiction $b_q^R = \Delta_q \geq \Delta_p > b_p^R$. ∎

Proposition 3. *Consider the sequence $0, \ldots, n$ of nodes and an optimal solution to the relaxation of Proposition 1, with speeds $v_i^{\mathrm{R}} = D_n/(t_n - t_0)$ and arrival times t_i^{R}. Take $b_i^{\mathrm{R}} = \max\{0, t_i^{\mathrm{R}} - \bar{t}_i, \underline{t}_i - t_i^{\mathrm{R}}\}$, and consider p such that $b_i^{\mathrm{R}} \leq b_p^{\mathrm{R}}$ for $i = 1, \ldots, n-1$. If $b_p^{\mathrm{R}} > 0$ and $t_p^{\mathrm{R}} < \underline{t}_p$ then there exists an optimal solution with $t_p = \underline{t}_p$.*

Proof. The proof is analogous to that of Proposition 2 and is omitted. ∎

Proposition 4. *The algorithm described in Section 1 for the SOP is exact.*

Proof. The result follows from Propositions 1–3. Indeed, the speed calculated in line 3 of Algorithm 1 corresponds to the optimal solution of the relaxed problem in Proposition 1, and this is optimal for the overall problem when there are no time window violations. Otherwise, the arrival time of node p is fixed to a value that is optimal according to either Proposition 2 or Proposition 3. ∎

Proposition 5. *The algorithm described in Section 1 for the SOP has a worst case running time of $O(n^2)$.*

Proof. The algorithm can be seen as a tree search algorithm. At the root node, the algorithm takes $O(n)$ operations to find the optimal relaxed speed and to find the largest time window violation. If a violation is found, the number of free speed variables is decreased by one, and the problem is solved by combining two subproblems that are solved on the next level of the tree. Each level of the tree requires $O(n)$ operations, and the number of levels is $O(n)$. ∎

Acknowledgments

An error in a previous version of Proposition 2 was discovered after input from Wang Xiaoyu, Teo Chee Chong, and Zhang Zhibin. For this we are most obliged. The authors also thank two anonymous referees for their comments regarding the initial manuscript. The research was carried out with financial support from the DESIMAL project, partly funded by the Research Council of Norway, and from the Canadian Natural Sciences and Engineering Research Council under grant 39682-10. This support is gratefully acknowledged.

REFERENCES

[1] K. Fagerholt, G. Laporte, and I. Norstad, Reducing fuel emissions by optimizing speed on shipping routes, J Oper Res Soc 61 (2010), 523–529.

[2] I. Norstad, K. Fagerholt, and G. Laporte, Tramp ship routing and scheduling with speed optimization, Trans Res C 19 (2011), 853–865.

[3] D. Ronen, The effect of oil prices on the optimal speed of ships, J Oper Res Soc 33 (1982), 1035–1040.

[4] Second IMO GHG Study 2009, Technical Report MEPC 59/10, International Maritime Organization, London, 2009.