Speed optimization problem in reducing ship fuel consumption

Hanwen Bi
College of Engineering and Computer Science
Australian National University
Canberra, Australia
hanwen.bi@anu.edu.au

Chaoxing Huang
College of Engineering and Computer Science
Australian National University
Canberra, Australia
chaoxing.huang@anu.edu.au

Chensheng Zhang
College of Engineering and Computer Science
Australian National University
Canberra, Australia
chensheng.zhang@anu.edu.au

***Abstract*—We look into the problem of reducing ship fuel consumption by optimizing ship speed at different segments of the voyage. The problem can be solved by constrained quadratic optimisation but it over-kills the reality as well as suffers from computation inefficiency. We redefine the problem in discrete time space as a shortest path problem by taking the stay time of the ship at each port into consideration. A 2-step rough-refinement based method is proposed to further improve the computation efficiency.**

***Index Terms*—optimisation, shortest-path, fuel-consumption, speed,time-constraint, 2-step method**

I. Introduction

Over the past decades, maritime industry has significant impact on international trade, and containerization is a major element in globalization. In 21st century, enhancing the sail efficiency and decreasing fuel cost of commercial ship have become a hot issues in maritime industry. To reduce fuel consumption, multinational maritime companies have taken a set of operational and technical measures such as slow steaming, ship system maintenance, automatic controls and waste heat and Engine Energy Recovery. Slow steaming is considered as one of the most efficient and economic among these measures according to Oceana Organization. Slow steaming is an operational measure that container cargo ship sails significantly less than their maximum speed and reduces the fuel consumption of container ship. A number of research reveals the model between fuel consumption and optimal ship speed. Ronens pioneer models[1] determine the optimal speed of ship in view of fuel saving by slow steaming. Fagerholt et al. [2]proposed an optimal approach considering soft time window apporach to obtain better schedules and reductions in the transportation costs. Fagerholt et al.[3] extended their study to treat the optimizing ship speed on a route as a non-linear optimization problem. The arrival time of ships are discretized and the optimization problem is solved as a shortest path problem on a directed acyclic graph without considering the stay time of the ship in each port. Wang and Meng proposed an optimal operating strategy which takes service frequency into account for a long-haul liner service route [4]. They developed a mixed-integer non-linear programming model and the result shows that the branch-and-bound based e-optimal algorithm is efficient. However, they have not considered the time window constraints of port. In[6], ship fuel reduction was achieved by a schedule planning model that involves the interaction with the ports with dynamic programming. Time window constraint is an essential aspect when considering the reducing ship fuel consumption problem as an optimisation problem. Kim et.al [5]proposed an optimal solution algorithm based on the time window reduction method. Their computational experiment result indicates the developed algorithm has significant impacts on reducing fuel costs. A holistic based method that considered time-window constraint is proposed in [7], which achieves a global optimized solution.

In our study, we tackle the speed optimization problem to reduce the ship fuel consumption in a realistic shipping route, which connects the port of Sydney and Shanghai by passing various of important ports in the Pacific area. We first solve the problem by using non-linear optimization method in continuous space as benchmark. Based on Fagerholt's work in [3], we refine the problem in discrete space by further taking the stay time of the ship at each port into consideration. Finally, to futher improve the computation efficiency of the algorithm, we firstly derive a rough solution by using the discrete time space method. After that, we find the accurate solution by searching the shortest path of the neighbor nodes of the rough solution. Comparison study on these three method is carried out by numerical experiments. We release our code here: https://github.com/huangchaoxing/Speed-optimization-in-reducing-ship-fuel-consumption

The rest of the report will be as follows. We define the optimization problem and the original mathematical model in continuous space as well as its limitation in section II. In

section III, we introduce our methods in discrete space that tackle the limitation in section II and validate our solution through numerical experiment on a realistic shipping route. Section IV is the conclusion.

II. System model and Problem definition

A. Problem definition

We first consider the background scenario of cargo ship voyaging. Firstly, the owner of the ship will report the planned (estimated) arrival time at each port to the corresponding harbour based on the service cruising speed. The service cruising speed of a ship is known if a certain type of ship is given. Every port has a service time window length for the ship to ensure the serving efficiency, which means the ship cannot arrive at the port too early or too late. At the first glance, one can save the fuel consumption by driving the ship at a very low speed, but this may risk of arriving at those port within the service time window. On the other hand, a large speed can possibly ensures a good timing, but the fuel consumption will increase. Thus, the overall problem can then be defined as:
how to find the optimal cruising speeds for different segments in the whole voyage to reduce the fuel consumption as possible as we can while the ship should arrive at each port within the given service time-window, and the speed of the ship should not violate the speed limitation. Intuitively, this is a constrained optimization problem. The fuel consumption is the object and the speeds for different voyage segments is the possible solution. The constraints are the compliance of the arrival time, and the speed limitation. We will introduce the initial mathematical model in the following context.

B. Initial mathematical model in continuous space

1) Parameters and notations:
$d_{i,i+1}$: The nautical distance between port i and port $i+1$ (in nautical mile).
$v_{i,i+1}$: The speed of the ship sailing from port i to port $i+1$ (in knots).
t_i^{min}: The minimum service time of port i.
t_i^{max}: The maximum service time of port i.
t_i: The arriving time of the ship at port i.
m_i: The maintenance duration of the ship at port i.
p_i: The pilotage duration of the ship at port i.
$f_{i,i+1}(v)$: The fuel consumption of the ship between the voyage of two ports(in ton).

2) Optimization problem for non-linear function:
According to [5], if the sail distance between two ports is fixed, the fuel consumption of this segment of voyage is:

$$\frac{d_{i,i+1}v_{i,i+1}^2}{24} \tag{1}$$

Thus if we add up the fuel consumption of all those segment, we come to the objective function which is the total consumption of fuel:

$$\sum_{i=1}^{n-1} \frac{d_{i,i+1}v_{i,i+1}^2}{24} \tag{2}$$

Next, we consider the relationship between the arrival time at port i and port $i+1$. After arriving at port i, the ship will go through the process of pilotage and maintenance at port i and the voyage from port i to port $i+1$ before arriving at port i+1.We then have:

$$t_i + p_i + d_{i,i+1}/v_{i,i+1} + m_i = t_{i+1}, 1 \leq i \leq n-1 \tag{3}$$

It means the arrival time at port i plus the time that the ship spends at port i as well as the time for sailing to port $i+1$ is exactly the arrival time at port $i+1$.

The optimization problem can be written as:

$$\textbf{Minimize Cost}= \sum_{i=1}^{n-1} \frac{d_{i,i+1}v_{i,i+1}^2}{24}$$

Subject to

$$t_i + p_i + d_{i,i+1}/v_{i,i+1} + m_i = t_{i+1}, 1 \leq i \leq n-1 \tag{4}$$

$$v^{min} \leq v_{i,i+1} \leq v^{max}, 1 \leq i \leq n-1 \tag{5}$$

$$t_i^{min} \leq t_i \leq t_i^{max}, 1 \leq i \leq n-1 \tag{6}$$

The objective function is the sum of all the fuel consumption between ports and the speeds for different segment is the decision variable. The first constraint is the time relationship which is mentioned above. The second constraint indicates that the ship speed should not surpass the upper bound speed due to the engine limitation while be no lower than the lowest speed for safety consideration. The third constraint means that the ship should arrive at the port within the pre-planned time-window, neither too late nor too early.
The initial problem is a convex optimisation problem with linear constraint. Ideally, many optimization solver can solve this problem effectively. However, some problems still exist. Firstly, solving an optimisation problem of a non-linear function is computational expensive. Secondly, this method though can precisely compute the optimal speed and the consequent arrival time in continuous space, it somewhat overkills the problems in reality. Since the voyage may affect by natural factors like weather or currents, and the port schedule may slightly change according to reality condition, it is impossible and unnecessary to reserve a pilotage or maintenance service at a very precise time.
We refine the problem to a comparatively sparse discrete space, and the solution is introduced in the next section.

III. Solutions and Numerical Validation

A. Shortest path problem with discrete time nodes

In **[3]**, the ship fuel reduction problem is transformed into a shortest path problem with discrete time nodes without considering the pilotage duration and maintenance duration at every port. Here, we re-frame the shortest path problem by taking the pilotage and maintenance duration into consideration.
Firstly, we perform the time window compression by recompute the actual available largest service time at each port, since the ship cannot arrive at the very last hour because there will

be not time left for pilotage and maintenance. The updated largest service time t_{new}^{max} is computed as:

$$t_{i,new}^{max} = t_i^{max} - m_i - p_i \tag{7}$$

Now, the time window at each port becomes $[t_i^{min}, t_{i,new}^{max}]$.

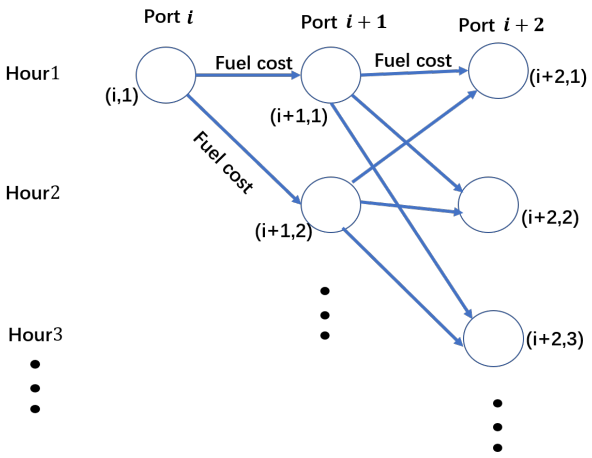

Fig. 1. Discretized time graph illustration

We discrete the time window into different potential arriving hours and set them as nodes as in the graph. For example, if the compressed time window of a port is $[1, 4]$, then we will have 4 nodes and those nodes are $[1, 2, 3, 4]$(hour). We naturally have those arcs between nodes,and those arc only points from the nodes of a time-window to those in the next time-window, which indicates that the ship travels from one port the next port. The graph is an acyclic graph and the weights on every arcs is the potential fuel cost travelling from t_i to t_{i+1},and the decision variable is the choice if we should choose a given path. In other words, the speed during one voyage between two ports affect the potential arrival time within the time window, thus resulting in different path of passing those time nodes. Our goal is to find a path that can minimize the fuel cost. Since the fuel cost is exactly the weight on each arc, the problem is then becoming a shortest path of acyclic graph problem. The graph is shown in Fig.1 . Note that those nodes will not be fully connected since some of the arcs results in violating the speed limitation. For example, some of the arcs pointing upward stand for a large speed and it may surpass the maximum speed of the ship. Those arcs are pruned to form the final graph.

Currently, we have a classic shortest path problem in an acylic grpah, which is expressed by the following equations:

$$\textbf{Minimize Cost}= \sum_{i,k,k'} c_{(k,i)(k',i+1)} x_{(k',i),(k,i+1)}$$

Subject to

$$\sum_k x_{(k,i-1),(k',i)} = \sum_k x_{(k',i),(k,i+1)} \tag{8}$$

$$x_{(k,i),(k',i+1)} \in \{0, 1\} \tag{9}$$

Here, k is the number of hour in a given time window of each port and i,j are the port index. To solve the above shortest path problem, we can use the Dijkstra's algorithm[10]. We refer this method as "1-step",since it directly apply the Dijkstra's algorithm on the constructed graph. We will explore the possibility to further improve the algorithm's efficiency in the next section.

B. Solving shortest path problem by using a local-graph method

Converting the ship fuel optimization problem to a shortest path problem can make it easy to solve, but when a more precise solution is required, the discrete time-slice would be smaller, and the nodes numbers, inverse proportion to the discrete time-slice, will become quite large. This makes the computation complexity increase. To overcome this problem, based on the monotone increasing characteristic of the objective function, we propose a novel solution of this optimization problem which can decrease the computation complexity remarkably.

Firstly, by given final discrete time-slice S we calculate the total nodes number of the network, which is denoted by N. From N, calculate the node number of the rough solution step as N_r,where N_r is the closest integer to the $\sqrt{N}$. From the reduced node number, we can infer the discrete time-slice of the rough solution S_r. S_r is larger than S to ensure that the time-graph is sparser than that in the 1-step algorithm . Once the sparse time-graph is obtained, we apply the shortest path algorithm as being mentioned to get a rough solution. Because the objective function is monotone increasing, the precise solution should be near to the rough solution, which means that when we use S to further obtain a local sub-graph that only considers the neighbourhood of the rough solution. We can discretize across the local time range of $[A_i - S_r, A_i + S_r]$, where A_i is the arriving time of the i^{th} port calculated from the rough solution. We then solve the shortest path problem in the local sub-graph to refine the results from the rough solution. The algorithm is shown in Algorithm 1. This approach significantly reduces the total nodes numbers of two graph, making the computation more efficient.

Algorithm 1 The 2-step algorithm

COMPUTE $N_1 = fix(\frac{T_1}{S}) + 1$
COMPUTE $N_1' = fix(\sqrt{N_1})$, $S_r = fix(\frac{T_1}{N_1'-1})$
COMPUTE $A_r =$ shortest-path-solver (S_r, W, D, V)
LOOP for $i = 1, 2, 3 \ldots n-1$ do
$W'(i,1) = max(W(i,1), A_i^r - S_r)$
$W'(i,2) = min(W(i,2), A_i^r + S_r)$
END LOOP
COMPUTE F,S_p,A =shortest-path-solver (S, W', D, V)
END

N_1 is the node number of the first arriving port, T_1 is the real time-window length of the first arriving port, and S is the time-slice of precise solution. $fix()$ is the MATLAB function. S_r is the time-slice for the rough solution. $A^r \in \mathbb{R}^{(n-1)}$ contains the optimal arriving time at each port, calculated by the rough

TABLE I
ROUTE AND PARAMETERS CONFIGURATION

Route	Distance(Nautical mile)	Route	Time window(hour)	Maintenance duration(hour)
Sydney → Melbourne	512	Sydney	[0,0]	0
Melbourne → Adelaide	470	Melbourne	[26,44]	3
Adelaide → Fremantle	1325	Adeleide	[66,84]	3
Fremantle → Jakarta	1733	Fremantle	[152,170]	3
Jakarta → Singapore	483	Jakarta	[268,286]	4
Singapore → Hong Kong	1415	Singapore	[315,333]	2
Hong Kong → Xiamen	260	Hong Kong	[409,427]	2
Xiamen → Shanghai	486	Xiamen	[447,465]	4
		Shanghai	[488,506]	

solution, $W \in \mathbb{R}^{2\times(n-1)}$, is the time-window of the ports, containing the earliest arriving time and latest departure time at each port. $D \in \mathbb{R}^{(n-1)}$ are the distances of the segments. $V \in \mathbb{R}^2$ is the speed constraint. W' is the new time window for the local graph. F is the optimal fuel consumption and $S_p \in \mathbb{R}^{(n-1)}$ are the optimal speed in those segments. $A \in \mathbb{R}^{(n-1)}$ are the final optimal arriving times at each port.

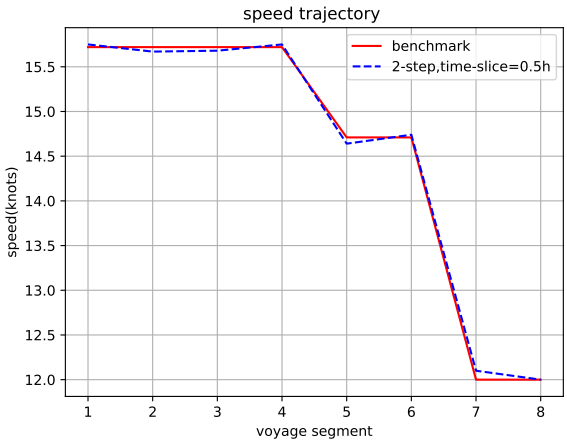

Fig. 2. Speed trajectory comparison with the benchmark

C. Numerical validations

1) Experiment setup: We conducted our experiments by simulating the algorithms on a real world shipping route that the ship sails from Sydney to Shanghai via 7 intermediate ports in the Pacific area. The data is originated from [8]. The time-windows of all the ports are all set as 18 hours and the minimum service time is computed by using the maximum service time of the last port and the service speed of the ship (18.5 knots) [9].To simulate the unexpected time changes in reality, the minimum service time of each port(apart from the very first port) is randomly added an integer ranging from -5 to +5 with uniform probability. The pilotage duration of the ship is 4 hours. The ship route and the parameter setup are shown in Table.I. We use the Bioinformatics and Optimization toolbox in MATLAB2019a[11,12] to solve the problem. The CPU being used in the experiment is 3.1 GHz Intel Core i5.

TABLE II
BENCHMARK SOLUTION FOR THE SPEED OF DIFFERENT SEGMENT

Segment	speed(knots)
Sydney → Melbourne	15.72
Melbourne → Adelaide	15.72
Adelaide → Fremantle	15.72
Fremantle → Jakarta	15.72
Jakarta → Singapore	14.71
Singapore → Hong Kong	14.71
Hong Kong → Xiamen	12
Xiamen → Shanghai	12

TABLE III
COMPARISON OF FUEL CONSUMPTION ON THE CONTINUOUS OPTIMIZED CONDITION AND THE SERVICE SPEED CONDITION

	Optimized	Service speed(18.5 knots)
Fuel consumption(ton)	1491.36	2249.50
Computation time(second)	0.789	–

2) The result on continuous model as benchmark: We firstly use the optimization toolbox to solve the original quadratic problem to verify the feasibility and obtain the benchmark. The comparison of the fuel consumption of using optimized speeds and the service speed is shown in Table.III. We also include the computation time of the benchmark in Table.III. It can be seen that the algorithm can save 758.14 tons of fuels,which is 30% of the original cost through out the whole voyage compared to the un-optimized case. The solution for the speed of every segment is shown in Table.II as benchmark. It can be seen that the speeds are all lower than the service speed while not less than the minimum, which indicates the algorithm manage to find a low speed to save fuel cost while not breaching the speed constraint.

3) The result on shortest path method: After verifying the feasibility of the original optimization model and obtaining the result as benchmark, we further conduct comparison experiments on the proposed shortest path methods, both on the 1-step and 2-step scenarios. The computation time results are shown in Table.IV. We also compared the computation time of 1-step and 2-step algorithms with different time-slices. The result shows that the 2-step algorithm is more computational

TABLE IV
COMPUTATION TIME WITH DIFFERENT DISCRETE TIME-SLICE

time-slice(hour)	2-step	1-step
4	0.007s	0.008s
2	0.009s	0.014s
1	0.011s	0.0188s
0.5	0.0195s	0.036s
0.2	0.027s	0.0687s

TABLE V
FUEL CONSUMPTION WITH DIFFERENT DISCRETE TIME-SLICE

time-slice(hour)	2-step	1-step	error with benchmark
4	1516.78	1516.78	1.704%
2	1503.72	1503.72	0.829%
1	1495.56	1495.56	0.282%
0.5	1491.96	1491.96	0.040%
0.2	1491.93	1491.93	0.038%

efficient, especially when the time-slice is small(denser time-graph). Besides, we also compared the estimated error of the fuel consumption of one-step and two-step algorithms with the given benchmark derived from the continuous time model. In Table. V, we can find that the one-step algorithm has the same result with two-step algorithm with fixed time-slice, and the error increases with the increase of time-slice. When the time-slice is 0.5 hour, the error is only 0.040%, which is acceptable. We use the two-step algorithm with 0.5-hour time-slice to solve this fuel optimization problem to get the optimal speed of each segment and compare the solution with the benchmark speed of each segment. Fig 2 illustrates the result of this comparison, which indicates that the solution is close to the benchmark.

D. Discussion

1) Why the 2-step algorithm is more computational efficient: In the 1-step algorithm, we simply solve a dense graph globally by including all the nodes and arcs into consideration, which includes many redundant path search in the computation. In the 2-step algorithm, we first look into a much more sparse graph to roughly find the range of the precise solution, this avoid the brutal path searching which is mentioned in the 1-step algorithm. Since the refinement stage is on a local sub-graph, the shortest path algorithm in this step will not consume a high number of computation resources. In one word, a sparser graph is the philosophy behind the improvement of the computation efficiency. A strict math proof is worth looking into in the future.

2) The necessity of transforming the problem from continuous space to discrete space: Though the computation time of the discrete method in the experiment is much shorter than that of the continuous benchmark, one might question that a computation time of less than 0.8 second is already efficient enough. However, there can be a situation that the number of ports in the whole voyage is very large. Also, if we take the factors like weather, ocean currents and even the status of other ship around into consideration, the model will be much more complicated while we might want to run the algorithm in real time in this scenario. Thus, it is necessary to improve the efficiency of the algorithm.

IV. CONCLUSION

In conclusion, we improved the Fagerholt's discrete time space algorithm [3], by considering the pilotage time and service time in each port. To further enhance the computation efficiency, we proposed a two-step discrete time space algorithm.

We use a real-world ship route containing 9 ports to test our algorithm and compare the test result with the benchmark derived by continuous time space algorithm. The validation shows that the two-step algorithm is more computational effective, and it can get a more precise solution when time-slice is small.

Future works include mathematical proof on why our algorithm improves the computation efficiency. Besides, it is also worth exploring the scenario that the speed can change within every segment due to the factors of weather and ocean current
.

REFERENCES

[1] D. Ronen, The Effect of Oil Price on the Optimal Speed of Ships, Journal of the Operational Research Society, vol. 33, no. 11, pp. 10351040, 1982.
[2] K. Fagerholt, Ship scheduling with soft time windows: An optimisation based approach, European Journal of Operational Research, vol. 131, no. 3, pp. 559571, 2001.
[3] 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. 523529, 2010.
[4] Q. Meng and S. Wang, Optimal operating strategy for a long-haul liner service route, European Journal of Operational Research, vol. 215, no. 1, pp. 105114, 2011.
[5] 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. 109117, 2014.
[6] S. Wang, A. Alharbi, and P. Davy, "Ship route schedule based interactions between shipping lines and port operators," Ocean Container Transport Logistics Making Global Supply Chain Effective. Elsevier, Amsterdam, 2014.
[7] S. Wang, A. Alharbi, and P. Davy, "Liner ship route schedule design with port time windows," Transportation Research Part C: Emerging Technologies, vol. 41, pp. 1-17, 2014
[8] S. Wang and Q. Meng, "Liner ship fleet deployment with container transshipment operations," Transportation Research Part E: Logistics and Transportation Review, vol. 48, no. 2, pp. 470-484, 2012.
[9] 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.
[10] E. W. Dijkstra, "A note on two problems in connexion with graphs," Numerische mathematik, vol. 1, no. 1, pp. 269-271, 1959.
[11] Matlab 2019a Optimization toolbox. Mathwork.
[12] Matlab 2019a Bioinformatics toolbox. Mathwork.