Scheduling the Service of Cargo Vessels in a Single Port with Spatial and Temporal Constraints

Loukas Chatzivasili[1⋆], Emmanouil S. Rigas[2], and Nick Bassiliades[3]
lchatz01@ucy.ac.cy, erigas@auth.gr, nbassili@csd.auth.gr

[1] Department of Computer Science, University of Cyprus, Nicosia, Cyprus
[2] School of Medicine, Aristotle University of Thessaloniki, Thessaloniki, Greece
[3] School of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece

Abstract. Nowadays, a large number of cargo vessels arrive at ports and need to be serviced on a daily basis. This creates a hard to solve problem related to finding the most efficient order to service these vessels. In this context, we study the problem of scheduling the service of cargo vessels arriving at a single port and considering certain spatial and temporal constraints. We propose an optimal scheduling algorithm and two equivalent greedy ones. The optimal algorithm is based on Integer Linear Programming (ILP). The first greedy approach is a First Come First Served (FCFS) heuristic algorithm, meaning that whichever vessel arrives first at the port will be served first. The second greedy algorithm, which we refer to as "Smart" applies a heuristic mechanism, that considers vessels' latest preferred departure time from the port after being serviced and the maximum unloading/loading speed of each dock. Through a detailed evaluation that uses realistic data, we observe that the optimal algorithm outperformed the two greedy ones in terms of the average waiting time, delay, and unloading times of vessels to be serviced, but with a significantly higher execution time.

Keywords: scheduling · heuristic · search · vessel · port management

1 Introduction

From antiquity, the transport of goods and people from one part of the Earth to another was conducted mainly through sea, with various types of vessels. A *vessel* is a broader notion than a *ship* and refers to anything that can float and be moved or guided, either manually or with the help of another person or object.[4] As a normal result of the huge industrial development of the last years, vessels in general and more specifically merchant ships are classified into various categories and types based on the construction material, the means of propulsion, the

⋆ Corresponding author

[4] https://www.shippingandfreightresource.com/difference-between-a-ship-and-a-vessel/

equipment they have, the type of cargo they carry, etc. The most common merchant ship regarding cargo transportation, not people transportation, is cargo ship. A cargo ship can carry all kinds of cargo such as raw materials, operating and maintenance materials of industrial plants, or other marketable products that may be consumed by humans.[5]

Despite the fact that vessels/ships are the number one means of transportation of materials, in contrast to land vehicle scheduling, relatively little work has been done in vessel routing and scheduling. In general, vessel 'scheduling' is mainly related to time windows attached to the calls of the vessels in the ports [11]. Service scheduling of vessels refers to the proper coordination of vessels to or within a port, or multiple ports, in order to achieve the most efficient service concerning time, space, energy and other affected factors. To satisfy and optimize such scheduling problems, it is extremely important to take into account a number of factors and constraints such as the speed of the vessels, the type and size of the vessels' cargo, the size of the port of reference, docks and quay cranes at the port [6], amongst others. Vessel scheduling is important not only to ensure the safe and efficient handling of traffic on busy waterways [12] but also to minimize fuel consumption and the related emissions [10] and bunker consumption from maritime transport [1] in liner shipping with either certain or uncertain port times.

The existing related literature [5] can be divided into three main categories: 1) General ship routing and scheduling problems, 2) Vessel scheduling problems at a port of reference, and 3) "Green" vessel scheduling problems. In the first category, Gatica and Miranda [7] propose a network-based model that includes discrete time windows for picking and delivering cargo. This allows for a broad variety of features and practical constraints to be implicitly included in the model. Whereas, in [2] a heuristic algorithm is used that is based on a variable neighborhood search, considering a number of neighborhood structures to find a solution to the problem. In [9], assuming fixed ship speeds, the problem of maximising profit is addressed. Then, a variable neighborhood search metaheuristic is proposed. In the second category, Golias et. al [8] describe an approach aiming at: (a) increasing berth productivity by cutting down on the total service time and delayed departures for all ships, and (b) cutting down on the total emissions and fuel used by all ships as they travel to their next port of reference. Another common problem is crane scheduling to service vessels arriving in the port of reference while minimizing their aggregate cost of delay [3]. In the third category, Dulebenets [4] presents a non-linear mathematical model which directly accounts for the carbon dioxide (CO_2) emission costs in the sea and at ports of call. In a similar study [10], an optimal vessel schedule in the liner shipping route is designed to minimize fuel consumption and emissions considering uncertain port times and frequency requirements.

This work aims to present a simplified version of the problem of service scheduling of cargo vessels in a port taking into account the following factors: (a) number of vessels arriving to a port of reference, (b) number of docks of the

[5] https://www.marineinsight.com/types-of-ships/what-are-cargo-ships/

port, (c) attributes of vessels (e.g., average sailing speed, cargo volume etc.) (d) port's elements (e.g. number of docks and average unload speed of their cranes). More specifically, the problem aims to figure out the proper order of service of various cargo vessels in a specific port to which they arrive in order to be serviced as quickly as possible and to reduce possible delays. The problem is solved using an optimal approach and two greedy heuristic algorithms, influenced by the algorithms of the related literature.

2 Problem definition

This paper considers the problem of several cargo vessels arriving at a single port that need to be serviced. The port can serve at the same time as many vessels as the docks it has, assuming that each vessel can be assigned to a maximum of one dock. Also, two or more vessels can arrive at the port at exactly the same time and expect to be serviced as soon as possible. The goal is to service all vessels arriving at a port in a certain period of time as quickly as possible and reduce possible delays observed in relation to the service of ships in a port (i.e.,their waiting time outside the port if there is no dock available in the port to enter it). Towards this, some spatial and temporal constraints (e.g., length, cargo, sailing speed) concerning the vessels and the port are taken into account.

More specifically, we denote the set of cargo vessels $\alpha \in A \subseteq \mathbb{N}$ and the port of reference p, which has a number of docks $d \in D \subseteq \mathbb{N}$ each one equipped with a crane, for unloading cargo from ships/vessels. Moreover, we assume a set of discrete time points $t \in T \subseteq \mathbb{N}$ to exist. Every vessel α has its length $l_\alpha \subseteq \mathbb{R}$, its volume of carried cargo $U_a \subseteq \mathbb{R}$ and two types of velocity. One is the average velocity of the vessel $\alpha, V_a^{aver} \subseteq \mathbb{R}$ and the other is the velocity of the vessel α at a specific time point, $V_{\alpha,t} \subseteq \mathbb{R}$. As already mentioned, the port of reference has a number of docks $d \in D \subseteq \mathbb{N}$ and each one of them has a length $\mu_d \subseteq \mathbb{R}$ and an unloading speed $s_d \subseteq \mathbb{R}$ concerning its crane.

On the same date and time t all vessels start from a different initial position to reach the port of reference. Depending on its velocity (i.e., we assume each ship to travel with a fixed average velocity) and its initial position, a vessel α arrives at the port at a specific time point $t_\alpha^{arr} \subseteq \mathbb{T}$, departs from the port at another specific time point $t_\alpha^{dep} > t_\alpha^{arr} \subseteq \mathbb{T}$ and has a latest departure time $t_\alpha^{latest} \geq t_\alpha^{dep} > t_\alpha^{arr} \subseteq \mathbb{T}$ after its cargo has been unloaded at a dock d_i of the port. Given the vessels' and the port of reference constraints, the port applies a scheduling algorithm to decide the sequence that the vessels should be serviced. In cases where it is not possible for all vessels to be served in time, the algorithm finds a solution that satisfies as many vessels as possible. For better specification of the problem, a set of restrictions has been determined:

1. A vessel α enters a dock d of the the port of reference p only if the dock is *available* and its length l_α is less than or equal to the length of the dock μ_d that is trying to enter ($l_\alpha \leq \mu_d$).

2. In case of a vessel α arrives at a port p and cannot enter one of its docks d, as assumption 1 is not satisfied, then waits outside the port until both of the aforementioned constraints apply.
3. A vessel α enters a port p, is serviced and departs from this port until a deadline called latest departure time t_{α}^{latest}.

3 Proposed Scheduling Algorithms

To solve the proposed problem of scheduling cargo vessels in a single port, an optimal and two greedy heuristic approaches are developed, all of the three being offline algorithms meaning the requests of the vessels are known in advance. All three approaches take into account spatial and temporal constraints and are aiming to serve all vessels arriving at a port of reference while minimizing delays. In the case that a vessel arrives at a port but there is no free dock to enter, it is forced to wait outside the port for some time (wait time). Also, if a vessel's real departure time from the port (after its cargo has been unloaded) exceeds its latest allowed departure time (deadline), a delay occurs.

3.1 Optimal Algorithm

In this section, we present a centralized, static, optimal Mixed Integer Programming (MIP) formulation of the problem, which is used for benchmarking purposes, using CPLEX component libraries [6]. More specifically, in this work, we used the Python programming language and corresponding CPLEX's libraries for Python to implement the optimal algorithm for our problem. The aim of this formulation is to find the optimal order of service for the vessels so as to maximize the number of vessels serviced and minimize delays regarding vessels' departures from the port of reference after being serviced. The formulation contains two binary decision variables: 1) decision variable $\epsilon_{\alpha,d} \in \{0,1\}$ denoting whether a vessel α is serviced at the dock d and 2) $\delta_{\alpha,d,t} \in \{0,1\}$ denoting whether a vessel α is at dock d at time point t. Thus, the following objective function is used under a number of constraints:

Objective Function:

$$\text{Maximize}\left(\sum_{\alpha\in A}\sum_{d\in D}\epsilon_{\alpha,d} - 0.000001\times\sum_{\alpha\in A}\sum_{d\in D}\sum_{t\in T}(\ \delta_{\alpha,d,t}\times t)\ \right) \tag{1}$$

The first part of the function is a double-sum that represents the number of vessels that are serviced at a dock of the port of reference (ideally all vessels arriving to the port). The second part of the function is a triple-sum that stands for the service time of a vessel at the port aiming at minimizing this time. The set of discrete time points T is equal to the difference of the latest t_{latest} of all vessels and the common starting date and time of all vessels' towards the port

[6] https://www.ibm.com/docs/en/icos/20.1.0?topic=mc-what-is-cplex

of reference. Notice that, we multiply the second part of the function by a very small number so that it never becomes greater than the first part and results in a negative number.

Constraints:

$$\forall \alpha \sum_{d \in D} \epsilon_{\alpha,d} \leq 1 \tag{2}$$

$$\forall \alpha \forall d \sum_{t_\alpha^{arr} \leq t_\alpha \leq t_\alpha^{latest}} \delta_{\alpha,d,t} = [\ (\ u_\alpha / s_d)] \times \epsilon_{\alpha,d} \tag{3}$$

$$\forall \alpha \forall d \sum_{t=1}^{T} |\delta_{\alpha,d,t+1} - \delta_{\alpha,d,t}| = 2 \times \epsilon_{\alpha,d} \tag{4}$$

$$\forall \alpha \forall d \epsilon_{\alpha,d} \leq [\ \mu_\delta / l_\alpha] \tag{5}$$

$$\forall d \forall t \sum_{a=1}^{A} \delta_{\alpha,d,t} \leq 1 \tag{6}$$

In more detail, every vessel can be assigned to a maximum of one dock of a port of reference (constraint 2) and stays at this dock as long time as it takes the dock's crane to unload its cargo (constraint 3). Also, a vessel remains at a dock for consecutive discrete time points (constraint 4) and a vessel is able to enter a dock and be serviced only when its length l_α is less than or equal to the length of the dock μ_d that is trying to enter (constraint 5). Lastly, if a dock is already occupied by a vessel, then no more vessels can enter it (constraint 6). In simple words, a dock can only service one vessel at a time. These constraints also hold in the heuristic algorithms that are presented below.

3.2 FCFS Heuristic Algorithm

In this case, vessels are serviced by the order they arrive at the port of reference, which means that whichever vessel arrives first at the port will be also served first (First Come First Served). Initially, the distance between each vessel and the port of reference is calculated according to the Euclidean distance between two points in the Euclidean space. According to the average velocity of each vessel, the travel time for each one to arrive from an initial starting location to the port of reference is estimated. Given that all vessels start their voyage towards the port at a specific time point (date-time) and having calculated the travel time for each vessel, it is now easy to find the arrival times of vessels. Up to this point the same procedure is followed in the "Smart" heuristic algorithm that is described next. Based on the order in which vessels arrive at the port along with random selection of dock, each vessel enters a specific dock where its cargo is unloaded by a dock crane. As soon as this procedure is completed, the vessel departs from the port of reference.

3.3 "Smart" Heuristic Algorithm

To solve the problem more effectively, another heuristic algorithm (see Algorithm 1) was developed to maximize the number of serviced vessels and at the same time minimize possible delays. In relation to the previous case (Section 3.2) the latest departure time and the selection of the best possible dock for the service of each vessel are additionally taken into account. As mentioned before, up to a point the procedure is the same as in the FCFS heuristic with only some minor changes. This time, vessels are firstly sorted by the order they arrive at the port and secondly by their latest departure time. In cases where two or more vessels arrive at the port of destination at exactly the same time, then priority is given to the vessel that has the soonest latest departure time in order to minimize possible delays. Towards this goal, the proposed algorithm selects the quicker dock in terms of its unloading speed in case restrictions 1 and 2 (section 2) apply to more than one dock for a vessel. If there is only one dock that meets restrictions 1 and 2 for a vessel, then it is automatically chosen.

4 Evaluation

In this section, we evaluate all three algorithms on different sets of vessels and a different number of the ports' docks, to determine their ability to maximize the number of serviced vessels and minimize delays. We examine the average waiting time of all vessels outside the port, the average delay time of vessels according to their latest departure time from the port, and the average unload time of vessels' cargo at the assigned dock. Regarding the time window the algorithms are running, this is set in a range starting from the moment the first vessel arrives at the port until the time corresponding to the latest deadline of all vessels. To this end, we use realistic datasets for vessels retrieved from MarineTraffic[7]. Due to the fact that the complexity of the Optimal algorithm grows exponentially, we divide our experiments into two categories. The first two, include all three algorithms and use small vessels' datasets, while the third uses only the heuristic approaches but larger datasets.

4.1 Algorithms' performance evaluation

Experiment 1 (10 Docks, 10 Vessels) The first experiment considers a port having 10 docks and 10 vessels. The fact that the number of port's docks is equal to the number of vessels arriving at the port, means that in the extreme case that all vessels arrive at the port at exactly the same time there will be an available dock for each vessel. The initial date and time of the problem are set to be: date 2022-01-01 and time 00:00:00. The latest departure dates and times (deadlines) of the vessels from the port are set to be until the end of the first ten days of the year, which is: date 2022-01-10 and time 23:59:59. The following figure 1 presents the results obtained using all three examined algorithms to calculate

[7] https://www.marinetraffic.com/

Algorithm 1 "Smart" Heuristic Algorithm

Require: p, A, t_α^{latest}(for each $\alpha \in A$), t
Create Lists: , $delays$, $waitTimes$
Create Dictionaries: $entryDates$, $travelTimes$, $unloadingTimes$, $assignedVessels$
for each $\alpha \in A$ **do**
    Calculate $travel_{time}$ using distance between α, p and V_α^{aver}
    Add $travel_{time}$ and (t_α^{latest}) to $travelTimes$
Sort $travelTimes$ ▷ By: 1) Min($travel_{time}$), 2) Min(t_α^{latest})
for each $\alpha \in A$ **do**
    Calculate t_α^{arr}
while [$size(unloadTimes) = \emptyset$] or [$size(unloadTimes) \leq size(A)$] **do**
    for each $\alpha \in A$ **do**
        if $unloadTimes(\alpha) \neq \emptyset$ **then**
            Calculate t_α^{dep}
            if $t_\alpha^{dep} = t$ **then**
                Find $wait_{time}(\alpha)$ and add to $waitTimes$
                Find $delay(\alpha)$ and add to $delays$
                Set d assigned to α to "$available$"
        if ($t_\alpha^{arr} \leq t$) and (α not in ($assignedVessels$) **then**
            for each $d \in D$ **do**
                if ($\mu_d \geq l_\alpha$) and (d is "$available$") **then**
                    Calculate $unload_{time}$
            Select min_d ▷ Criteria: s_d
            $min_{time} = unload_{time}(min_d)$
            if $min_{time} \neq \emptyset$ **then**
                Set d to "$Not\ available$"
                Add min_{time} to $unloadingTimes$
                Add t to $entryDates$
                Assign α to min_d
                Set min_d to $\emptyset$
                Set min_{time} to $\emptyset$
    Go to the next time point, t_{++}
    if (each $d \in D$ is "$available$") and ($size(assignedVessels) = size(A)$) **then**
        $break$
return $waitTimes, delays, unloadTimes$

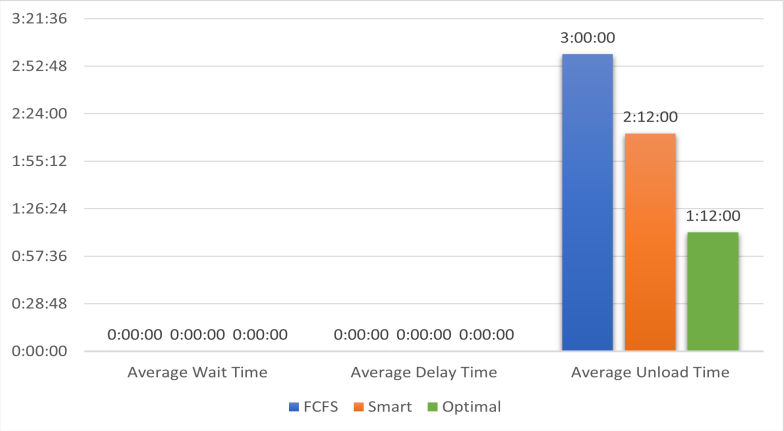

Fig. 1. Average Wait, Delay and Unload Time for Experiment 1

the average wait time, the average delay time and the average unload time of all serviced vessels at the port of reference. The y-axis of the figure is in the form hours:minutes:seconds. According to figure 1, both average wait time and average delay time are minimal for all three methods. Regarding average unload time we observe a reduction of about 50 minutes or 27.78% from FCFS heuristic to "Smart" heuristic and about another 50 minutes reduction or a percentage of 37.78% from "Smart" heuristic to Optimal algorithm.

Experiment 2 (10 Docks, 20 Vessels) The second experiment also considers a port having the same 10 docks as in experiment 1, but this time 20 vessels instead of 10 (experiment 1) arrive at the port. Note that based on the data of the vessels and that of the port, some of the vessels will arrive at the port at exactly the same date-time. Also, it is worth mentioning that the number of docks of the port are fewer than the total number of vessels arriving at the port which means that not all vessels will be able to be served at the port at exactly the same time. The starting date and time of the vessels from the port have not changed compared to the first experiment. Figure 2 presents the results obtained for all three algorithms. In this experiment, we observe that the Optimal algorithm manages to eliminate the average delay time and reduces the average unload time by about 1 and 2 hours compared to the FCFS and the "Smart" heuristics respectively. However, the Optimal's average wait time has increased from zero to almost 10 minutes in relation to the other two algorithms. This occurs because the heuristic algorithms do not examine future arrivals and when a vessel arrives they serve it immediately in most cases. Whereas, the Optimal takes into account all future arrivals, and considers how to minimize all criteria in total, so it may let some vessels wait for some time outside the port in order to maximize the other criteria in the set of requests. Thus we can state that the Optimal algorithm is the best option among the three algorithms.

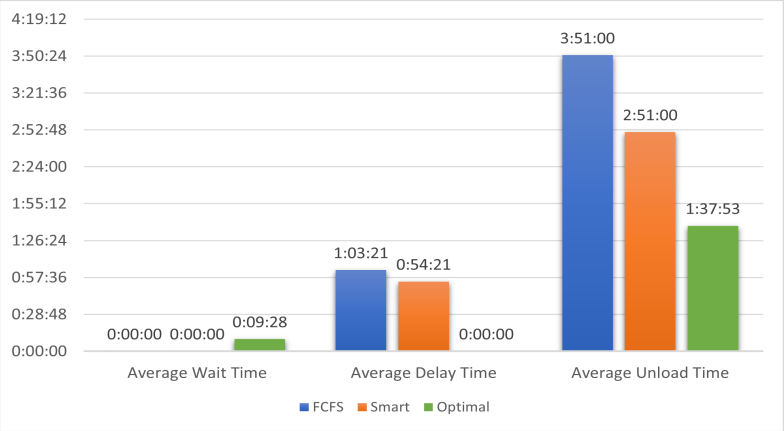

Fig. 2. Average Wait, Delay and Unload Time for Experiment 2

Experiment 3 (20 Docks, 100 Vessels) This final experiment considers 20 docks and 100 vessels and is conducted only for FCFS and "Smart" heuristics. Optimal algorithm is not used because of its high complexity due to the high number of vessels leading to an exponentially increase of the execution time. Similarly to the previous experiments, some of the vessels will arrive at the port at exactly the same date and time (two-two, three-three, etc). The vessels' data used in this experiment are exactly the same as these of experiment 2. The number of docks at the port is smaller than the total number of vessels arriving at the port, at a ratio of 1 dock to 5 vessels. Again that means that not all vessels will be able to be served at the port at the same time, but this time more vessels arrive simultaneously at the port than in the second experiment. The starting date and time of the vessels from the port have also not changed. Average wait

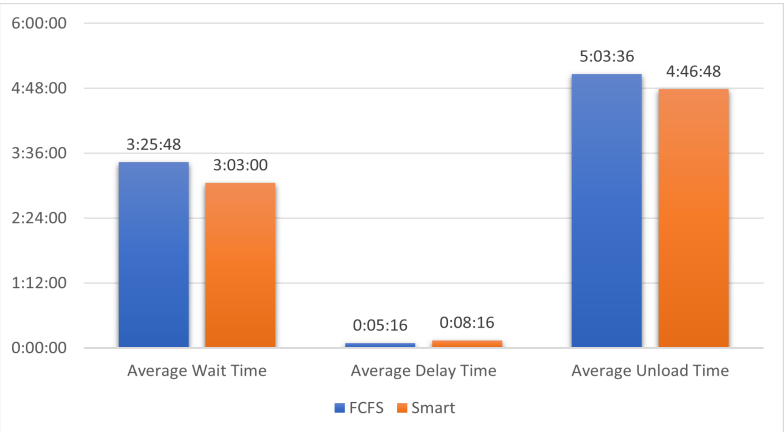

Fig. 3. Average Wait, Delay and Unload Time for Experiment 3

time and average unload time using "Smart" heuristic are lower compared to the FCFS heuristic algoritm (figure 3). In contrast, regarding the average delay time we can observe that FCFS is the one with the lower time result. Such a case may occur because "Smart" heuristic algorithm takes into account two different criteria (soonest deadline and quicker dock's crane in terms of unload speed) in the case where two or more vessels arrive at the port at exactly the same time. However, overall the "Smart" heuristic algorithm reduces average wait time by about 10.73% (i.e., 22 minutes) and average unload time by 5.61% (i.e., 17 minutes), whereas it is only three minutes lower regarding delay time compared to the FCFS heuristic algorithm. Considering all these findings, it is undeniable that this time the "Smart" heuristic is the best possible option.

4.2 Execution Times

Figures 4 and 5 summarize the execution times (E.T), in seconds, of each algorithm for each experiment presented above. Regarding the first experiment, it is

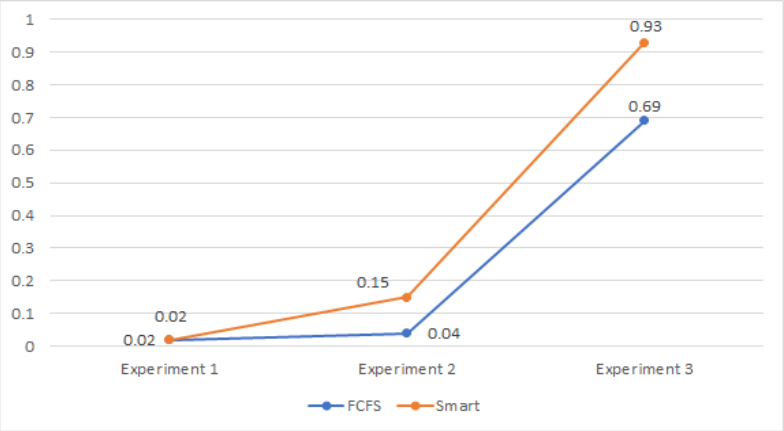

Fig. 4. Execution times of the 2 Heuristic Algorithms in seconds

clear that the two heuristic procedures solve the problem in just a few fractions of a second, whereas the optimal needs 85 seconds or 1.5 minutes to do so. In fact, its execution time is thousand times longer even though it is the most efficient. In the second experiment, again the Optimal algorithm needs a lot more time than the others, especially since the number of vessels has increased and the possible solutions to examine are multiplying. Finally, in the last experiment, the execution time of FCFS heuristic is about 88% smaller than the "Smart" heuristic but still, both execution times are short (under 1 second). Another essential observation is that the execution time of the two heuristic approaches is increasing slightly, while we increase the number of vessels. On the other hand, the execution time of the Optimal algorithm is increasing rapidly, because there

are numerous combinations of vessels and docks for the algorithm to consider (to find the optimal solution). Consequently, when we tried to run this algorithm on the data of the third experiment, it was unable to terminate in a reasonable amount of time. In such cases, the "Smart" heuristic algorithm is a very quick and effective way to solve the proposed scheduling problem.

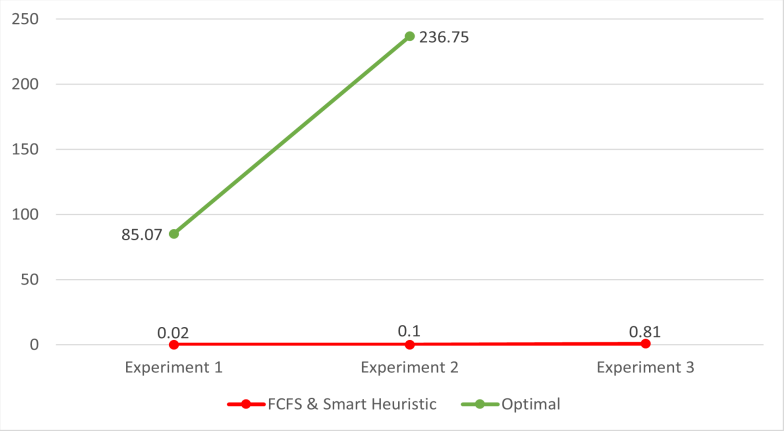

Fig. 5. Execution times of Optimal's and Heuristics' Algorithms in seconds

5 Conclusions and future work

The aim of this work was to study the problem of scheduling the service of cargo vessels in a port taking into account spatial and temporal constraints. Toward this goal, two greedy scheduling algorithms and an optimal algorithm were implemented. The first approach is a First Come First Served (FCFS) heuristic algorithm that takes into account only the order in which vessels arrive at the port of reference and assigns a vessel to a random dock of the port for service. The second approach is a so-called "Smart" heuristic algorithm that takes into account not only the order in which they arrive at the port but also the deadline of departure for each ship and chooses the quicker dock in terms of cargo unloading. Last but not least, the optimal algorithm implemented, solves this scheduling problem as a MIP problem and finds the optimal solution. Through an empirical evaluation, we conclude that even though all three algorithms managed to serve all the vessels arriving at the port of reference, the Optimal algorithm fully dominates the heuristic approaches. However its execution time increases rapidly as the number of vessels and docks examined increases, so on large datasets it is not feasible to execute it in a reasonable amount of time. In such cases, "Smart" heuristic algorithm usually leads to considerably better results than the simple FCFS heuristic. However, the factors and constraints considered by the "Smart" algorithm may sometimes not be enough to get good results for all the variables

under consideration, although it is still possible to serve all vessels. The latter reveals that there is clearly room for improvement in this algorithm.

Various extensions could be considered as future work of this study: 1) The "Smart" heuristic algorithm that was developed could take into account more factors and constraints such as energy constraints related to vessels and their fuels, but also environmental constraints related to the "Green" scheduling problem, most notably carbon dioxide (CO_2) emissions [10, 8]. 2) Another limitation that could be taken into account for vessel scheduling problem is the costs (operating, maintenance, etc. cost) that vessels will have until arriving at the port of reference as well as the cost that the port will shoulder to service all these vessels that arrive at it in a certain period of time. 3) Apply mechanism design techniques to incentivize vessels to report their desired latest departure time.

References

1. Brouer, B.D., Dirksen, J., Pisinger, D., Plum, C.E., Vaaben, B.: The vessel schedule recovery problem (vsrp) – a mip model for handling disruptions in liner shipping. European Journal of Operational Research **224**(2), 362–374 (2013)
2. Castillo-Villar, K.K., González-Ramírez, R.G., Miranda González, P., Smith, N.R.: A heuristic procedure for a ship routing and scheduling problem with variable speed and discretized time windows. Mathematical Problems in Engineering **2014** (2014)
3. Daganzo, C.F.: The crane scheduling problem. Transportation Research Part B: Methodological **23**(3), 159–175 (1989)
4. Dulebenets, M.A.: Green vessel scheduling in liner shipping: Modeling carbon dioxide emission costs in sea and at ports of call. International Journal of Transportation Science and Technology **7**(1), 26–44 (2018)
5. Dulebenets, M.A., Pasha, J., Abioye, O.F., Kavoosi, M.: Vessel scheduling in liner shipping: a critical literature review and future research needs. Flexible Services and Manufacturing Journal **33**(1), 43–106 (2021)
6. Fu, Y.M., Diabat, A., Tsai, I.T.: A multi-vessel quay crane assignment and scheduling problem: Formulation and heuristic solution approach. Expert Systems with Applications **41**(15), 6959–6965 (2014)
7. Gatica, R.A., Miranda, P.A.: Special issue on latin-american research: a time based discretization approach for ship routing and scheduling with variable speed. Networks and Spatial Economics **11**(3), 465–485 (2011)
8. Golias, M., Boile, M., Theofanis, S., Efstathiou, C.: The berth-scheduling problem: Maximizing berth productivity and minimizing fuel consumption and emissions production. Transportation Research Record **2166**(1), 20–27 (2010)
9. Malliappi, F., Bennell, J.A., Potts, C.N.: A variable neighborhood search heuristic for tramp ship scheduling. In: International Conference on Computational Logistics. pp. 273–285. Springer (2011)
10. Qi, X., Song, D.P.: Minimizing fuel emissions by optimizing vessel schedules in liner shipping with uncertain port times. Transportation Research Part E: Logistics and Transportation Review **48**(4), 863–880 (2012)
11. Ronen, D.: Cargo ships routing and scheduling: Survey of models and problems. European Journal of Operational Research **12**(2), 119–126 (1983)
12. Sluiman, F.: Transit vessel scheduling. Naval Research Logistics (NRL) **64**(3), 225–248 (2017)