International Journal of Industrial Engineering Computations 10 (2019) 491–504

Contents lists available at GrowingScience

International Journal of Industrial Engineering Computations

homepage: www.GrowingScience.com/ijiec

A modified sailfish optimizer to solve dynamic berth allocation problem in conventional container terminal

Issam El Hammouti[a*], Azza Lajjam[b], Mohamed El Merouani[a] and Yassine Tabaa[a]

[a]*Faculty of Sciences, Abdelmalek Essaadi University, Tetuan, Morocco*
[b]*National school of applied sciences, Abdelmalek Essaadi University, Tetuan, Morocco*

C H R O N I C L E

Article history:
Received January 16 2019
Received in Revised Format March 27 2019
Accepted April 5 2019
Available online April 5 2019

Keywords:
Modified sailfish optimizer algorithm
Meta-heuristic
Container Terminal
Berth Allocation Problem
Optimization

A B S T R A C T

During the past two decades, there has been an increase on maritime freight traffic particularly in container flow. Thus, the Berth Allocation Problem (BAP) can be considered among the primary optimization problems encountered in port terminals. In this paper, we address the Dynamic Berth Allocation Problem (DBAP) in a conventional layout terminal which differs from the popular discrete layout terminal in that each berth can serve multiple vessels simultaneously if their total length is equal or less than the berth length. Then, a Modified Sailfish Optimizer (MSFO) meta-heuristic based on hunting sailfish behavior is developed as an alternative for solving this problem. Finally, computational experiments and comparisons are presented to show the efficiency of our method against other methods presented in the literature in one hand. We also discuss the productivity of a container terminal based on different scenarios which can happen.

1. Introduction

In recent years through economic liberalization and global trade growth, the demand for maritime transportation services has witnessed an increase in a very important way. In this regard, container terminals, thanks to its role as a transshipment point that connects the maritime and land transport, are considered the principal responsible for the good management of container's mass flow arrived at the port. In dealing with this growth and to satisfy the need of vessel-owner in terms of minimizing port transit costs, port operators must improve the productivity and the performance of container terminals. This productivity deals with the capacity to serve the incoming vessels with a minimum waiting and handling time. It is usually measured by vessel's turnaround time which is the time spent between vessel's arrival and departure. After vessel's arrival, port operators must take many decisions. First, they have to find a quay space for this vessel. Next, containers are transferred from vessels to the quay using quay cranes. Then, containers are handled to the yard for stowage purpose. Finally, these containers leave the storage area to ground or maritime transportation. From these operations, many problems arise to be treated in a container terminal (Meisel, 2009). It includes quay crane scheduling problem covered by

* Corresponding author
E-mail: issam.elhammouti@gmail.com (I. E. Hammouti)

2019 Growing Science Ltd.
doi: 10.5267/j.ijiec.2019.4.002

Lajjam et al. (2014), stowage planning addressed by Avriel et al. (1998) and berth allocation planning which is the focus of our paper. Berth Allocation Problem (BAP) is the first critical decision to take by the operational planning service. It consists of assigning a set of incoming vessels to a set of berths along the quay in order to optimize a performance measure considered as the minimization of vessel's total stay time at a port (Fig. 1). In order to carry out this assignment, vessel-owners calling at a port have to transmit to the terminal planning service a dataset concerning their vessel some days before their arrival at the port. This dataset contains the length and the draft of the vessel, the number of containers to be loaded/unloaded as well as the estimated arrival time.

In addition, it is necessary to take into account the factor of terminal's layouts type to solve the BAP. As reported by Meisel (2009) and Imai et al. (2013), we can resume it in the following four types:

- Discrete layout, the quay is divided into a specified number of berths. See Fig. 2(A)
- Continuous layout where the quay is not divided; therefore, vessels can carry out the berthing conforming their need for space on the quay. See Fig. 2(C)

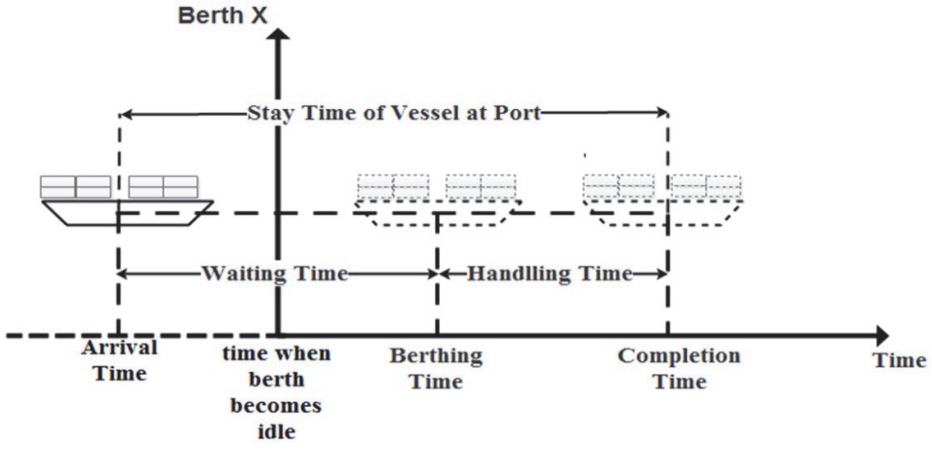

Fig. 1. Stay time of vessel at a port

- Indented layout is characterized by its capability of fast handling from both sides of a vessel as shown in Fig. 2(D) and Imai et al. (2013).
- Conventional layout, the quay is split in a discrete manner, except that small vessels can share a single berth but big vessels can be positioned just into one single berth as shown in Fig.2 (B) and Imai et al. (2013). We should note that this type of layout is known in some papers of the literature as a hybrid layout.

According to Imai et al. (2013), it has been demonstrated that a conventional terminal (see Fig.2 (B)) is more efficient than an indented at the level of stay time. For this reason, we have opted for the conventional terminal in our study. Moreover, in the literature we have found different formulations of the BAP problem because of the diversity of terminal layouts. As a consequence, different methods and meta-heuristic have been proposed to solve all existing varieties. So, we propose a Modified Sailfish Optimizer meta-heuristic approach to solve the DBAP in a conventional container terminal. In fact, the Sailfish Optimizer meta-heuristic has provided effective results for solving combinatorial problems, the fact that encouraged us to use this meta-heuristic.

The next section presents a literature review on the BAP. Then, model formulation is presented in Section 3. In Section 4, a meta-heuristic based on a modified Sailfish optimizer is adapted as a method of resolution. Next, a number of experimental results are presented in Section 5. Finally, Section 6 concludes the paper.

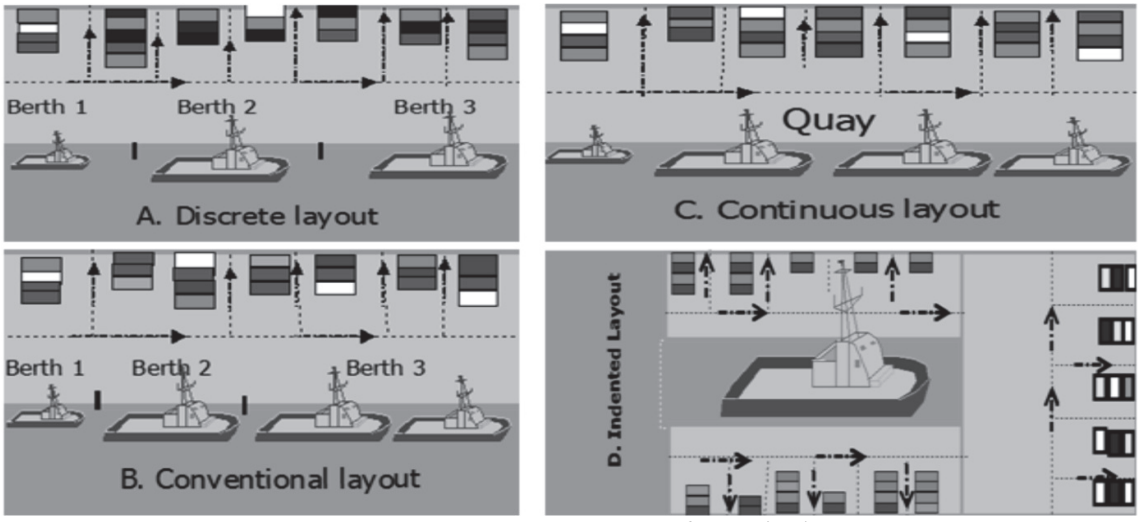

Fig. 2. Type of Terminal Layouts

2. Literature review

In the literature, BAP has been approached by several scientific articles. However, there are some differences between these publications because of terminal's layout type (see the section 1) and ship's arrival type (dynamic or static). In the dynamic arrival, the vessels which have not arrived yet at the port in the beginning of the planning are also taken into account in the planning. Whereas, all the vessels should be already in the port before the planning starts in the static one. Among all of those literature works, dynamic discrete case (DDBAP) and BAP in dynamic continuous case (DCBAP) are the ones which have received the most attention. However, in our paper we have focalized on the DBAP in a conventional layout which represent also realistic constraint found in container terminals. In all cases (DDBAP, DCBAP and DBAP in a conventional layout) mathematical modeling of these problems lead to NP-hard problems that require meta-heuristic methods to be developed for their resolution. In the following, a large list of literature works on all varieties of BAP is presented.

Lai and Shih (1992) developed a heuristic algorithm for solving the BAP based on the policy of First Come First Served (FCFS). The concept of dynamic BAP appeared for the first time in Imai's et al. (2001) paper. For resolution, the authors used a heuristic based on lagrangian relaxation method. Two years later, Imai et al (2003) improved their model considering different service priorities between vessels, and resolved the problem using a genetic algorithm (GA). In the category of works witch adapted the Genetic Algorithm (GA) as a method of resolution, Nishimura et al. (2001) employed GA to solve the DDBAP considering different water depth configuration and servicing multiple vessels in a berth at the same time. Theofanis et al. (2007) studied the DDBAP and proposed a resolution approach based on GA to minimize the total weighted service time of all ships. Imai et al. (2013) proposed a linear mathematical model for the DDBAP which permit the simultaneous berthing of multiple vessels at the indented berth, and solved this model with a GA. For its part, Arango et al. (2011) addressed the BAP using simulation and optimization at the same time. First, they proposed different scenarios of simulation with Arena software in order to minimize the total operating time for each vessel in the port of Seville. Then, a mathematical model for the DDBAP and a heuristic procedure based on GA algorithm are presented to solve the problem.

Furthermore, Cordeau et al. (2005) implemented a heuristic based on tabu search for the DDBAP and DCBAP; they assessed the quality of solutions found with their heuristic by making a comparison with the exact solution found by CPLEX. Buhrkal et al. (2011) presented three different mathematical programming models of the DDBAP and proposed a formulation of the problem as a generalized set partition problem (GSPP). For testing their formulation, they used the instances from Cordeau et al. (2005) and obtained optimal solutions using Cplex. In Oliveira et al. (2012), a meta-heuristic based on a

clustering search with simulated annealing (CS-SA) is presented as an alternative for DDBAP which improve the results presented by Cordeau et al. (2005) and Buhrkal et al. (2011). López-Plata et al. (2015) proposed an improvement of solutions for DDBAP based on Decentralized Cooperative Meta-heuristic (DCM). They compared their method with other methods in the literature for testing their results. After three years, Lin et al. (2018) developed a new version of SA based on different vessel assignment strategies to solve DCBAP. The results presented in this work are considered the best so far. To minimize the total service time for each vessel in a dynamic and discrete container terminal, Nishi et al. (2017) proposed a dynamic programming-based on met heuristic as a resolution method. In this work different comparisons with others methods of the literature have been realized by the authors to show the performance of the proposed met heuristic.

Recently, many studies have addressed the BAP under uncertain factors such as the arrival time or the service time for each vessel. In this context, Budipriyanto et al. (2015) developed a conceptual model of the vessel-berth allocation given the variability of vessel arrival and handling time. The objective of this model is the reduction of the total processing time and the improvement of resource's utility (berth, quay crane and container yard). In Budipriyanto et al. (2017), authors presented a study where they demonstrated that the strategy of collaboration between terminals is the primary factor that minimizes delays caused by uncertainty in vessel's arrival and handling time. In dealing with the factor of uncertainty, Liu et al. (2017) addressed the BAP under a number of discrete scenarios. Each scenario contains different arrival and handling time for each vessel which can happen with a certain probability. The objective of this work is to minimize the total service cost (El hammouti et al., 2018).

3. Model formulation

In this section, we introduce a mathematical formulation for the DBAP in a conventional terminal, inspired from the linear model published by imai et al. (2013). In this model, they calculated the stay time of a ship in the port as the difference between its completion and arrival time to the port (Fig.1). Furthermore, to simplify and clarify our model we considered the following assumptions and notation:

Assumptions:

- Length of each berth is 400m
- The berth allocation ignores the FCFS rule
- The length of each vessel range from 200 to 400 (meters)
- All berths have the same water depths.
- The safety distance between berthed vessels is included in the length of the vessel.
- If a ship is assigned to a berth, it will remain in that position until all cargo-handling operations are completed.
- Multiple ships can be served at the same berth simultaneously if their total length does not exceed the overall berth length.
- The vessel handling time depends on the assigned berth
- The planning horizon is one week.

Sets:

i (= $1 \ldots I$) $\in B$ set of berths
j (=$1, \ldots, T$) $\in V$ set of vessels
k (=$1, \ldots, T$) $\in U$ set of service orders

Parameters:

M = A very large number
A_j = Arrival time of vessel j

BL_i = Length of berth i
L_j = Length of vessel j
S_i = Time when berth i becomes idle for the first time in the planning horizon
C_{ij} = Handling time of vessel j at berth i

Decision variables:

X_{ijk} =1 if vessel j is handled as kth vessel at berth i, and =0 otherwise
$W_{ikk'}$ =1 if both the k th and k' th vessel to be assigned are berthed simultaneously in berth i
b_{ik} = beginning time of handling for k th vessel at berth i
f_{ik} = completion time of handling kth vessel at berth i

Objective function

$$min\ z\ =\ \sum_{i\in B}\sum_{k\in U} f_{ik}\ -\sum_{j\in V} A_j, \tag{1}$$

subject to:

$$\sum_{i\in B}\sum_{k\in U} X_{ijk} = 1, \qquad \forall\, j \in V \tag{2}$$

$$\sum_{j\in V} X_{ijk} \leq 1, \qquad \forall\, i \in B, k \in U \tag{3}$$

$$b_{ik} \geq \sum_{j\in V} (\max(S_i - A_j)\, X_{ijk} \qquad \forall\, i \in B, k \in U \tag{4}$$

$$f_{ik} = b_{ik} + \sum_{j\in V} C_{ij} X_{ijk}, \qquad \forall\, i \in B, k \in U \tag{5}$$

$$b_{ik} \leq b_{ik'}, \qquad \forall\, i \in B, k, k'(>k) \in U \tag{6}$$

$$f_{ik} \leq b_{ik'} + w_{ikk'}M, \qquad \forall\, i \in B, k, k'(>k) \in U \tag{7}$$

$$B_{ik'} \leq f_{ik} + (1 - w_{ikk'})M, \qquad \forall\, i \in B, k, k'(>k) \in U \tag{8}$$

$$\sum_{j\in V} L_j\ X_{ijk} + \sum_{j\in V} L_j\ X_{ijk'} + (w_{ikk'}1)\, M \leq BL_i, \qquad \forall\, i \in B, k, k'(>k) \in U \tag{9}$$

$$\sum_{k'\in U} W_{ikk'} \leq 1, \qquad \forall\, i \in B, k \in U \tag{10}$$

$$X_{ijk} \in \{0,1\}, \qquad \forall\ i \in B\ , j \in V, k \in U \tag{11}$$

$$W_{ikk'} \in \{0,1\}, \qquad \forall\, i \in B, k, k'(>k) \in U \tag{12}$$

$$b_{ik} \geq 0, f_{ik} \geq 0, \qquad \forall\, i \in B, k \in U \tag{13}$$

The objective function (1) minimizes the total stay time of vessels at the port. Constraint (2) ensures that at any berth and any order of service a vessel must be served. Constraint (3) enforces that at the same berth, each vessel must be served at an order of service different, note that although two vessels are simultaneously at a berth, their service orders must be different. Constraint (4) enforces that, the starting time of the handling of a given vessel in a given berth is equal to the maximum between their arrival time and the time when berth becomes idle in the planning horizon. Constraint (5) determines vessel departure time. Constraint (6) determines the service order of vessels assigned to the same berth. Constraints (7) and (8) guarantee that if two vessels are served simultaneously, their services coincide in time. Constraint (9) enables two vessels to be served at the same berth if their total length is equal or less than the berth length. Constraint (10) ensures that the number of vessels served simultaneously in a specific berth does not exceed two. Constraints (11), (12) and (13) define the type of decision variables (X_{ikj}, W_{ikk}, b_{ik}, f_{ik}).

4. The Sailfish Optimizer

Recently, Shadravan et al. (2019) developed a new meta-heuristic called Sailfish Optimizer which combines the behavior of both a group of sailfish as a predator and a group of sardine as the prey. The sailfish is classified as a social predator because it is characterized by working in group to catch and

hunts its prey. In a cooperative hunting, predators use different strategies to kill. For example the group of sailfish is characterized by the alternation of attacks strategies. It consists in that, at a given moment, each member of group attack alone the school of prey (sardine) and injure or hunt some of them, while the other members of group save their power. When a sailfish attacks the prey school he can update his position around them. In addition, the sailfish can also update his position to occupy empty space around the prey school and simulate encircling the prey (Shadravan et al., 2019). On the other hand, in order to escape from the following attacks of the sailfishes, the prey group (sardine) changes the position when a member of their group is injured (Shadravan et al., 2019). In the following subsections we present the general procedure of Sailfish optimizer algorithm.

4.1.Initialization

- A sailfish and sardine population is initialized randomly, to each sailfish and sardine is assigned a randomized position x_i^k and y_j^k consecutively. Where, $i \in$ {sailfishes}, $j \in$ {sardines} and $k \in$ {number of iteration}
- The position of each sailfish (x_i^k) or sardine (y_j^k) represents a feasible solution for the problem at k^{th} iteration

4.2. Evaluation and the elitism procedure

At each new generated population, an evaluation based on the fitness function (F (.)) of the position of each agent search (sailfish or sardine) is carried out. In a minimization problem, the best sailfish which has the smallest fitness from the sailfish population is saved as the elite sailfish (x_{eli}^k, $eli \in$ {set of sailfish}) i.e. $F(x_{eli}^k) <= F(x_i^k), \forall\ k$. Similarly, the best sardine which has the smallest fitness in the sardine population is saved as the injured sardine (y_{inj}^k, $inj \in$ {set of sardine}) i.e. $F(y_{inj}^k) <= F(y_j^k), \forall\ k$. The saving of the elite sailfish and the injured sardine is equivalent to an elitism procedure because allows us not to lose the good solutions when we update the position of the search agents.

4.3. *Position updates by the sailfish*

Over course of iteration, each member of group from sailfish population can update his position around the school of the prey. The change of the position in the sailfish's behavior is realized by the alternation of attacks strategies or by the capacity to occupies empty space around the prey school. Mathematically, the update position of sailfish is based principally on the position of elite sailfish and the injured sardine as its shown in Eq. (14)

$$x_i^{k+1} = x_{eli}^k - \lambda_k * (\beta * ((x_{eli}^k + y_{inj}^k) / 2) - x_i^k) \quad (14)$$

where x_i^{k+1} is the new position of sailfish at (k+1)th iteration, x_i^k is the current sailfish i position, β is a number generated randomly between 0 and 1, x_{eli}^k and y_{inj}^k are consecutively the position of the current elite sailfish and the position of the current injured sardine and λ_k is a coefficient generated at each k^{th} iteration by the following equation:

$$\lambda_k = (2 \times \beta \times PD) - PD \quad (15)$$

where, β is a number generated randomly between 0 and 1 and PD presents the density of the school prey. Due to the alternation of attacks on the prey school some sardines will be injured and be hunted by the sailfishes therefore the number of prey will decrease over time. The PD parameter is determined by the Eq. (16)

$$PD = 1 - (N_{sh} / (N_s + N_{sh})) \quad (16)$$

where N_{sh} and N_s are respectively the number of sailfish and the number of sardines in each iteration. In this regard, the initial school of sardine usually is larger than the group of sailfish. For this reason, we suppose that $N_s = 3 \times N_{sh}$. The value of λ_k is a decisive factor in the algorithm, because by adjusting the

value of λ_k each sailfish can update his position moving toward the prey and at the same time round it in different directions.

4.4. position updates by the sardine

At the beginning of the hunt, both the sailfish power attack and the sardine escape ability are usually very high. Therefore, at the first stage of the hunt, the sailfishes just injure the sardines in school prey without being able to catch them. Over time, the attack power of the sailfishes decreases as well as the ability to escape from the sardines. Indeed, the sailfishes accuse the effort of the alternation of attacks strategies while the sardines accuse the injuries in their body. This leads to that, at the last stage of the hunt preys to lose the ability to escape from the attacks. Hence, the capture success rate of sailfishes becomes high. To take into account the behavior of sardine against the attacks of sailfish. Each sardine in our algorithm can update its position based on the following equation:

$$y_j^{k+1} = r \times (x_{eli}^k - y_j^k + AP), \tag{17}$$

where y_j^{k+1} and y_j^k are the new and the current position of sardine j consecutively, r is a random numbers between 0 and 1, x_{eli}^k is the best position of elite sailfish found so far, and AP shows the amount of sailfish's Attack Power at each iteration that is generated as follows:

$$AP = A \times (1 - (2 \times Itr \times \varepsilon)), \tag{18}$$

where A and ε *are* two factors that decrease the value of power attack (AP) and Itr is the number of current iteration. As mentioned previously, at the first stage of the hunt, the capture success rate is small because most of sardines can change their positions and escape from the sailfish attack. However, at the end of the hunting, the ability of sardine to escape decreases which makes the capture success rate higher. So that, the number of sardines that can update their position decreases with time. In our algorithm the last stage of hunting was considered when AP is less than 0.5. Based on power attack AP, the number of sardine that update their position at last stage of hunt (AP<0.5) is given by Eq. (19):

$$\alpha = N_s \times AP, \tag{19}$$

where AP is less than 0.5 just a selected number α of sardines can update their position. On the contrary, when AP is more than 0.5 we considered that all sardine will be updated.

4.5. Remove and substitution of the hunted sardine

In the proposed algorithm, to model the fact that a sardine j is hunted by a sailfish i, the position of this later is replaced by the position of the sardine j when we have $F(y_j^k) < F(x_i^k)$, as given by the Eq. (20).

$$x_i^k = y_j^k \quad if \quad F(y_j^k) < F(x_i^k), \tag{20}$$

where x_i^k is the position of sailfish i at k^{th} iteration and y_j^k *is* the position of sardine j at k^{th} iteration. After the substitution, the sardine j should be removed from the sardine population. A flowchart of SFO is shown in Fig. 3.

5. Modified Sailfish Optimizer for DBAP at a conventional terminal

In this section we present a Modified Sailfish Optimizer (MSFO) to solve DBAP, the main modification is to add a local search technique at the last stage of the SFO algorithm (AP <0.5). This modification makes the SFO more efficient at level of exploitation and ensures an improvement in solution quality.

5.1. Solution representation by Sailfish and sardine

A primary issue to develop any meta heuristic is the solution representation. In our algorithm both sailfish and sardine have the same structure solution. This solution was represented as a table of dimension [1, n], whose values are real numbers in the interval [1, m+1[. where m represents the number of berths and n represents the number of vessels. The integer part of each real number in the table represents the berth assigned to each vessel. However, the fractional part represents the service order of each vessel in berth

where it is assigned. The first table cell represents the vessel number 1, the second represents the vessel number 2 and so on until the last cell which represents the vessel number *n* (n is the table dimension). A sorting in an ascending order of the table is carried out to separate vessels according to their berth and their service order in this berth. Indeed, the service order of a set of vessels assigned to the same berth can be retrieved from the increasing order of fractional parts, i.e a vessel with a higher number will be served after vessels with lower numbers.

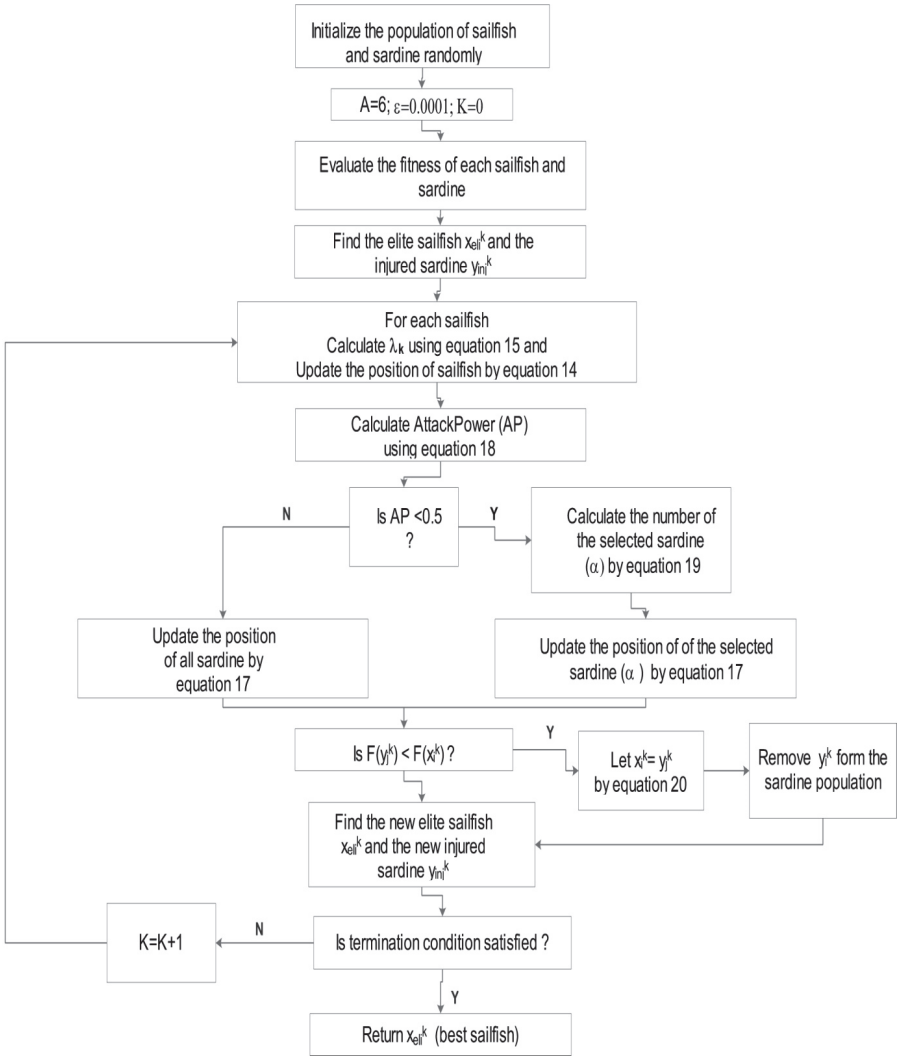

Fig. 3. Flowchart for SFO

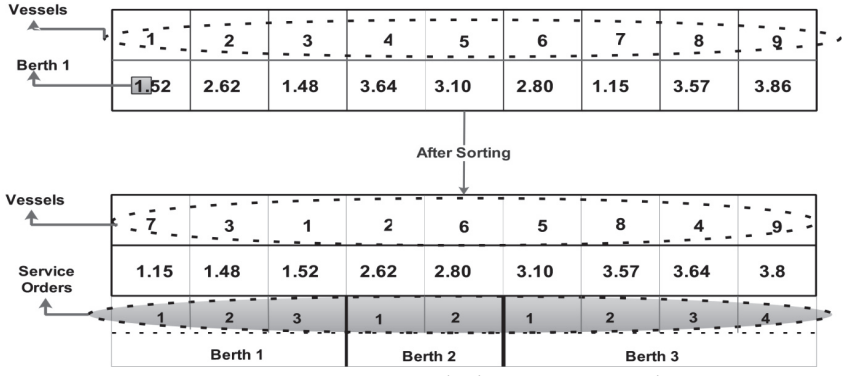

Fig. 4. Solution representation

Fig. 4 shows the representation of a feasible solution found by a Sailfish for a DBAP in a conventional layout with nine vessels and three berths. For each vessel corresponds a real number contained in the interval [1,4[whose integer part determines the berth of vessel and the fractional part determines the service order of each vessel with respect to vessels assigned to the same berth. For example, the vessels {1,3,7} are berthing in berth 1 and according to the sorting in an increasing order of the fractional parts the vessel 7 will be the first to be served followed by the vessel 3 and 1, the same for others berths. To limit the search only in the feasible solution space, we will proceed as follows. For example, for solution presented in Fig. 4 if the value in a cell is less than 1, we will randomly generate a value in [1, 2 [for the cell. If the value is lower than 4 (number of berths plus one), we will randomly generate a value in [1, 2 [and subtract this from 4 (number of berths plus one).

5.2. Generation of the local solution for the Modified SFO algorithm

At the last stage of the algorithm when the value of AP is less than 0.5, a local search technique is applied to generate two new solutions x_i^v and $y^v{}_i$, where x_i^v is a feasible solution at neighborhood of the elite sailfish x_{eli}^k and y_i^v is a feasible solution at neighborhood of injured sardine y_{inj}^k. Then, if $F(x_i^v) < F(x_{eli}^k)$ ($F(y_i^v) < F(y_{inj}^k)$), $x_{eli}^k(y_{inj}^k)$ become $x_i^v(y_i^v)$. Two permutations are performed as shown in Figure 5. The first one, swap two vessels within the same berth with the objective to change their service order see Fig.5 (a). While the second one, swap two vessels between two berths with the objective to change their berth see Fig.5 (b). A flowchart describing the proposed MSFO is given in Fig. 6.

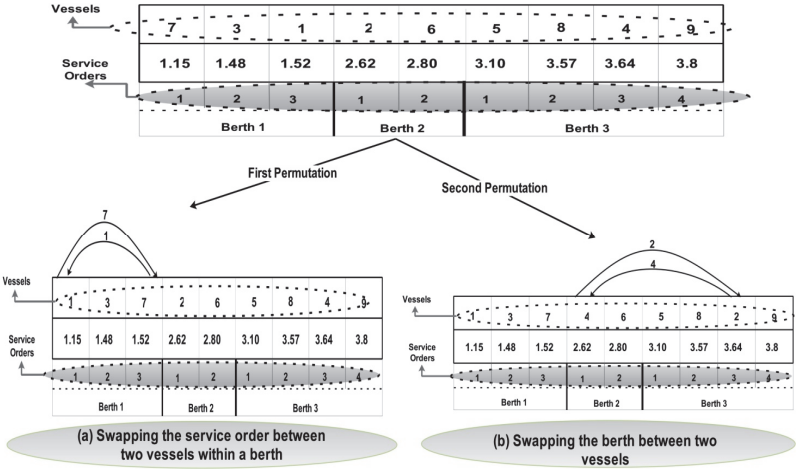

Fig. 5. Local solution

6. Computational results

This section includes two parts: the first one contains the computational results to show the effectiveness of the proposed MSFO resolution approach and the last one presents an experimental analysis of the container terminal performance based on a set of scenarios that often happens in a container terminal. The parameter-setting plays a primordial role in the good performance of the algorithm. In our case, After the preliminary tests, the following parameter-setting guarantee best solution in short running time: $A=6$ and $\varepsilon = 0.001$,Maximum iteration(K_{max}) = 500, initial population for the sailfish= 30 and initial population for the sardine= 60.

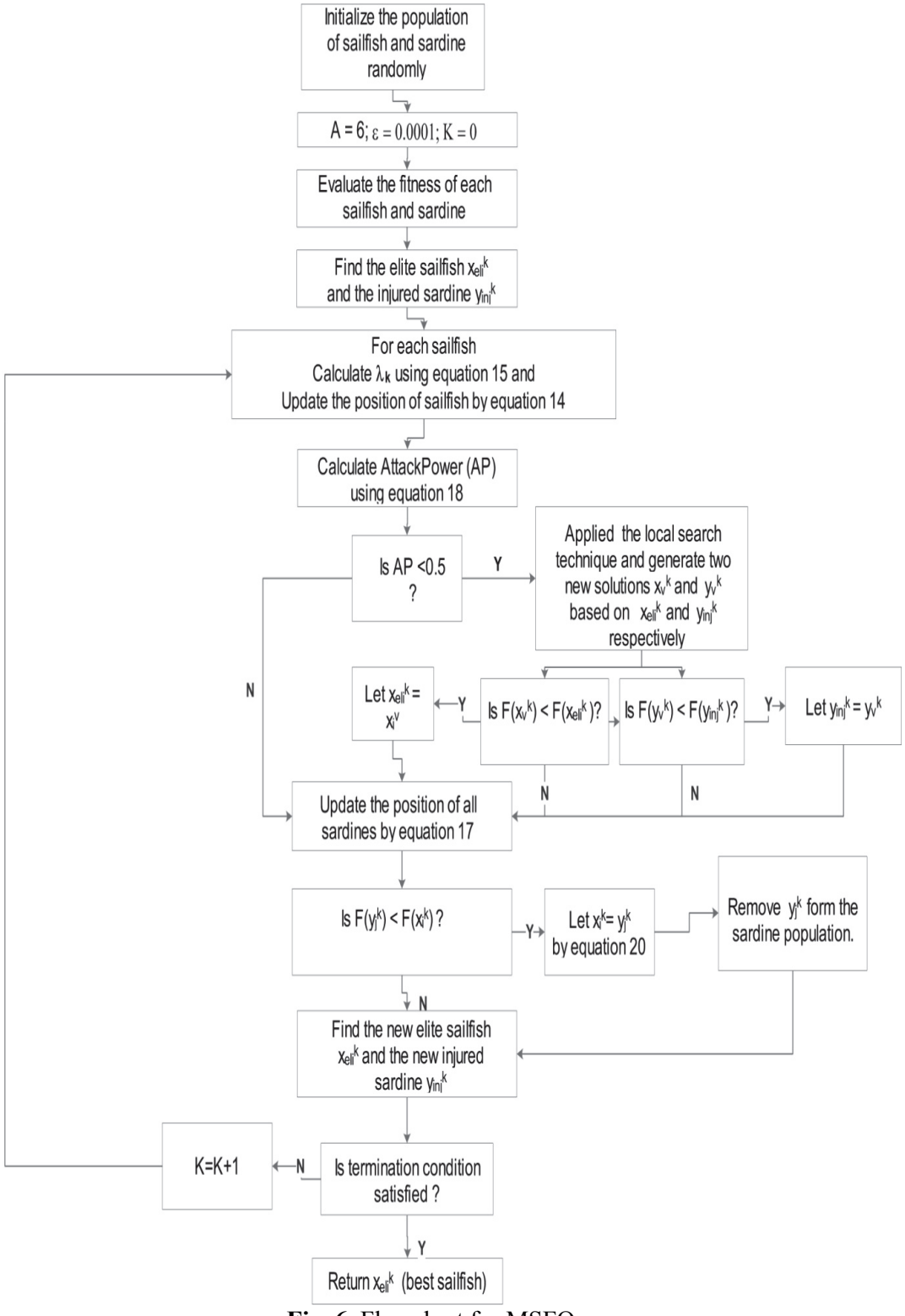

Fig. 6. Flowchart for MSFO

6.1. Effectiveness of the proposed MSFO

The proposed MSFO meta-heuristic was implemented with language java on Acer computer with an Intel core i3 (2.30 GH) and 4 GB RAM. To test the effectiveness of our method, first we did not take into account the simultaneous service at berth, i.e. all constraints which ensure the simultaneous service in

the mathematical model presented in section 3 have been eliminated. In this way, the new mathematical model is equivalent to DDBAP model. Then, a comparison was carried out between the objective function value and the running time found with our approach and those found using other existing methods, namely, Tabu Search(T^2S) see Cordeau et al. (2005), Generalized Set-Partitioning Problem mathematical model (GSPP) by Buhrkal et al. (2011), Clustering Search with Simulated Annealing for generating initial solutions (CS-SA) see Oliveira et al. (2011) and Decentralized Cooperative Met heuristic (DCM) by López-Plata et al. (2015). This comparison is based on the set I_3 of the benchmarking instances used by Cordeau et al.(2005) which includes 30 instances with 60 vessels and 13 berths.

Table 1
Comparison of the results found with T^2S, GSPP, CS-SA, DCM and MSFO

	T^2S	GSPP		CS-SA		DCM		MSFO	
Vessels×Berth	Objective Value(hours)	Objective Value (hours)	Time (sec)	Objective Value (hours)	Time (sec)	Objective Value (hours)	Time (sec)	Objective Value (hours)	Time (sec)
60×13 (1)	1415	1409	17.92	1409	12.47	1409	5.95	1409	**4.25**
60×13 (2)	1263	1261	15.77	1261	12.59	1261	4.15	1261	**4.04**
60×13 (3)	1139	1129	13.54	1129	12.64	1129	4.18	1129	**4.10**
60×13 (4)	1303	1302	14.48	1302	12.59	1302	4.25	1302	**4.18**
60×13 (5)	1208	1207	17.21	1207	12.68	1207	3.21	1207	**4.12**
60×13 (6)	1262	1261	13.85	1261	12.56	1261	4.04	1261	4.08
60×13 (7)	1279	1279	14.60	1279	12.63	1279	3.36	1279	4.21
60×13 (8)	1299	1299	14.21	1299	12.57	1299	4.96	1299	**4.23**
60×13 (9)	1444	1444	16.51	1444	12.58	1444	5.25	1444	**4.24**
60×13 (10)	1213	1213	14.16	1213	12.61	1213	3.46	1213	4.18
60×13 (11)	1378	1368	14.13	1368	12.58	1368	5.21	1368	**4.26**
60×13 (12)	1325	1325	15.60	1325	12.56	1325	4.62	1325	**4.21**
60×13 (13)	1360	1360	13.87	1360	12.61	1360	3.76	1360	4.15
60×13 (14)	1233	1233	15.60	1233	12.67	1233	4.14	1233	4.10
60×13 (15)	1295	1295	13.52	1295	13.80	1295	4.31	1295	**4.29**
60×13 (16)	1375	1364	13.68	1364	14.46	1364	4.89	1364	**4.19**
60×13 (17)	1283	1283	13.37	1283	13.73	1283	3.09	1283	4.23
60×13 (18)	1346	1345	13.51	1345	12.72	1345	4.14	1345	4.19
60×13 (19)	1370	1367	14.49	1367	13.39	1367	5.93	1367	**4.26**
60×13 (20)	1328	1328	16.64	1328	12.82	1328	5.60	1328	**4.21**
60×13 (21)	1346	1341	13.37	1341	12.68	1341	5.54	1341	**4.20**
60×13 (22)	1332	1326	15.24	1326	12.62	1326	4.97	1326	**4.23**
60×13 (23)	1266	1266	13.65	1266	12.62	1266	4.01	1266	**4.20**
60×13 (24)	1261	1260	15.58	1260	12.64	1260	4.90	1260	**4.19**
60×13 (25)	1379	1376	15.80	1376	12.62	1376	5.54	1376	**4.25**
60×13 (26)	1330	1318	15.38	1318	12.62	1318	4.92	1318	**4.12**
60×13 (27)	1261	1261	15.52	1261	12.64	1261	4.00	1261	**4.10**
60×13 (28)	1365	1359	16.22	1359	12.71	1359	5.56	1359	**4.16**
60×13 (29)	1282	1280	15.30	1280	12.62	1280	5.82	1280	**4.12**
60×13 (30)	1351	1344	16.52	1344	12.58	1344	5.76	1344	**4.19**
average	1309.7	1306.8	14.98	1306.8	12.79	1306.8	4.65	1306.8	**4.18**

In Table 1, the column 1 presents the size of the problem. While other columns expose the objective function value and running time found by T^2S, GSPP, CS-SA, DCM and MSFO, respectively. According to the results presented in the Table 2, we can observe clearly that the proposed MSFO is capable of obtaining an optimal solution for all instances tested as the case of GSPP, CS-SA and DCM. Furthermore, at the level of running time our method overcomes all others except some instances for DCM namely (6, 7, 10, 13, 14, 17, and 18). However, our method improves the DCM in 23 instances of 30 presented and the average of running time of MSFO is 4.18 while the average of DCM's running time is 4.78. Therefore, we can conclude that MSFO is a new good alternative compared with other methods to solve the BAP.

6.2. Experimental analysis on the performance of a conventional and discrete terminal layout

6.2.1. Presentation of the case-study and generation of problem instances

To make our performance analysis, we opted to take the Tangier Med I container terminal as the basis of our study, given our ability to access to port data. However, this study can be generalized to other

container terminals. The Tangier Med I container terminal is equipped with a 1600 meter of quay divided into 4 berths with a length of 400 meter. So, according to the traffic observed in the terminal during a planning horizon of 7 days, the average arrival interval of vessels is 4 hours which corresponds to an average of 42 vessels calling at the port in a week. The number of cranes assigned to each vessel depends on its size (3 or 4 cranes for vessels of size between 200 and 320 (meter) and 5 cranes for vessels of size is (> 320 meter) with the estimated efficiency of each crane is 30 containers per hour. In addition, the time took in handling a vessel is calculated according to the following rule:

(Handling time = number of container to be loaded and unloaded for the vessel / number of cranes affected to the Vessel×efficiency of each crane)

The data presented above represent the scenario of the standard conditions (S.C) for the container terminal Med 1 but usually in a container terminal there are dynamic factors as the breakdown of cranes, congested traffic and incapability of a berth to serve vessels for maintenance reasons that may generate other scenarios and disturb the normal performance of the terminal especially at the level of ships staying time in the port. Therefore, the objective of our analysis is to determine the factor that can significantly disturb the performance of the port in comparison with the performance obtained at standard conditions. So for this reason, three types of scenarios have been considered, namely scenario 1 which represents the case of congested traffic, scenario 2 represents the case when there is a breakdown in the crane and scenario 3 defines the case when some berths are incapable to serve vessels for reasons of maintenance. In addition, for each scenario three cases of studies are carried as we can see in detail in the Table 2.

Table 2
Presentation of data used in each scenario

Scenarios	Case 1	Case 2	Case 3
Scenario 1: Congested traffic	The Average arrival interval of vessels : 3h	The Average arrival interval of vessels is : 2h	The Average arrival interval of vessels is: 1h
Scenario 2: The breakdown of the cranes	Assigned 2 crane instead 3 for the ship with size between 200 and 250 meter	Assigned 3 crane instead 4 for the ship with size between 251 and 319 meter	Assigned 4 crane instead 5 for the ship with size is (>=320)
Scenario 3: Incapability of berths for reasons of maintenance.	Close 200 m of berth 1	Close 200 m of berth 1 and 3	Close 200 m of berth 1,2 and 3

6.2.2. Discussion and results

In our analysis, we have adapted the mathematical model presented at the section 3 to all scenarios mentioned previously to study the case of the conventional terminal. Then, at both the conventional and the discrete terminal, a comparison of the staying time of vessels between all scenarios and the scenario at S.C has been realized. In all experiences we have used the MSFO meta-heuristic as a method of resolution.

Table 3
Comparison of the results found in all scenarios

Scenarios	Objective Value in hours (conventional terminal)				Objective Value in hours Discrete terminal			
	Case 1	Case 2	Case 3	Average	Case 1	Case 2	Case 3	Average
Scenario 1	18053	43090	168644	76595	20920	50283	198612	89938
Scenario 2	12003	11201	11824	11676	13822	12594	13488	13301
Scenario 3	12262	17368	34527	21385	12262	17368	34527	21385

Table 3 represents the value of the objective function determined in all cases of each tested scenario and the Fig.7 shows the value of the objective function in both conventional and discrete terminals in S.C.

According to the Fig.7, it is obvious that the conventional terminal is more efficient than the discrete at the level of staying time of vessels. Because of the policy of multiple ships served simultaneously at the conventional terminal we can optimally exploit the space in the quay and therefore minimize the staying time of vessels. In addition, according to Fig.7, we can conclude that in both terminals (conventional and discrete) the objective value tends to increase in the three scenarios with respect to the objective value in S.C the fact that we can intuit before, but among all the scenarios the first one is the one that marks a highest long-stay of the vessels therefore is the factor that most disturb the normal performance of the terminal.

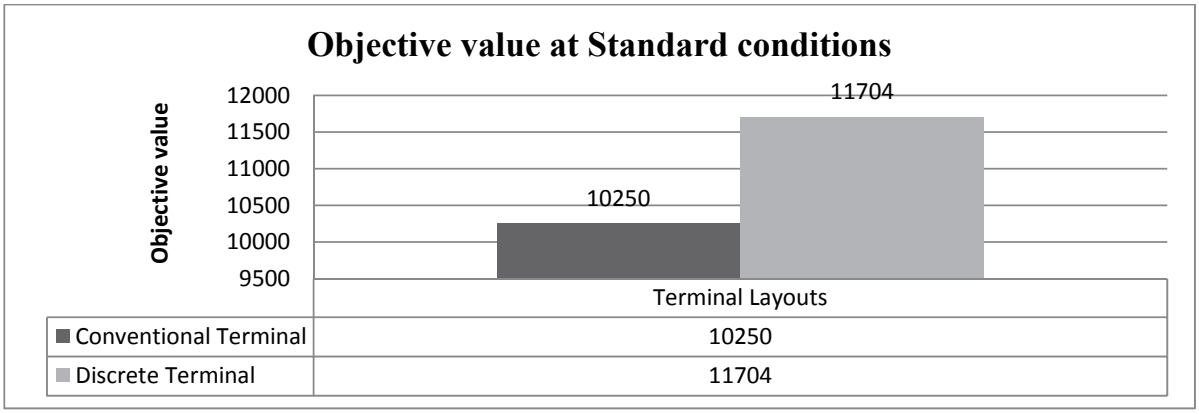

Fig. 7. Objective value at standard conditions in both conventional and discrete terminal

7. Conclusion

In this work, we have approached the berth allocation problem in the dynamic case at a conventional terminal where multiple vessels are served simultaneously in the same berth. Then, to solve the problem a new modified sailfish optimizer meta heuristic, based on hunting sailfish behavior has been developed. Building on a set of instances from the literature, a comparison of our results with other recent methods found in the literature has been accomplished in order to test and show the affectivity of our algorithm. Computational experiences have demonstrated the capacity of the proposed method to solve the BAP problem optimally and in a short time. In addition, From the performance analysis of Tangier Med I container terminal, we can conclude that the congested traffic is the main factor which disturbs the normal performance of a container terminal and increases the total staying time of vessels in the port. In the future works on Dynamic Berth Allocation Problem, other type of ports will be taken into account such as the bulk terminals which combine the waterway problem with the berth allocation problem.

References

Avriel, M., Penn, M., Shpirer, N. and Witteboon, S. (1998). Stowage planning for container ships to reduce the number of shifts. *Annals of Operations Research*, *76*, 55-71.

Arango, C., Cortés, P., Muñuzuri, J. and Onieva, L. (2011). Berth allocation planning in Seville inland port by simulation and optimisation. *Advanced Engineering Informatics*, *25*(3), 452-461.

Budipriyanto, A., Wirjodirdjo, B., Pujawan, N., & Gurning, S. (2015). Berth allocation problem under uncertainty: A conceptual model using collaborative approach. *Procedia Manufacturing*, *4*, 429-437.

Budipriyanto, A., Wirjodirdjo, B., Pujawan, I. N. and Gurning, S. (2017). A Simulation Study of Collaborative Approach to Berth Allocation Problem under Uncertainty. *The Asian Journal of Shipping and Logistics, 33*(3), 127-139.

Buhrkal, K., Zuglian, S., Ropke, S., Larsen, J. and Lusby, R. (2011). Models for the discrete berth allocation problem:A computational comparison. *Transportation Research Part E Logistics and Transportation Review*, *47*(4), 461-473.

Cordeau, J.F., Laporte, G., Legato, P. and Moccia, L. (2005). Models and tabu search heuristics for the berth-allocation problem. *Transportation Science, 39*, 526-538.

de Oliveira, R. M., Mauri, G. R., & Lorena, L. A. N.(2012). Clustering search for the berth allocation problem. *Expert Systems with Applications*, *39*(5), 5499-5505.

El hammouti, I., Lajjam, A. and El merouani, M. (2018). Solving the hybrid berth allocation problem using a bat-inspired algorithm. *In Optimization and Applications (ICOA)*, 4th International Conference on (pp. 1-6). IEEE.

Imai, A., Nishimura, E., & Papadimitriou, S. (2001). The dynamic berth allocation problem for a container port. *Transportation Research Part B: Methodological*, *35*(4), 401-417.

Imai, A., Nishimura, E., & Papadimitriou, S. (2003). Berth allocation with service priority. *Transportation Research Part B: Methodological*, *37*(5), 437-457.

Imai, A., Nishimura, E., & Papadimitriou, S. (2013). Marine container terminal configurations for efficient handling of mega-containerships. *Transportation Research Part E: Logistics and Transportation Review*, *49*(1), 141-158.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). *Optimization by simulated annealing*. *Science, 220*(4598), 671-680.

Lajjam, A., El Merouani, M., Tabaa, Y., & Medouri, A. (2014). A new approach for sequencing loading and unloading operations in the seaside area of a container terminal. *International Journal of Supply and Operations Management*, *1*(3), 328.

Lai, K. K., & Shih, K. (1992). A study of container berth allocation. *Journal of Advanced Transportation*, *26*(1), 45-60.

Lalla Ruiz, E., Izquierdo, C. E., Batista, B. M. and MorenoVega, J. M. (2015). Decentralized Cooperative Metaheuristic for the Dynamic Berth Allocation Problem. *Inteligencia Artificial, 18*(55), 1-11.

Lin, S. W., Ting, C. J., & Wu, K. C. (2018). Simulated annealing with different vessel assignment strategies for the continuous berth allocation problem. *Flexible Services and Manufacturing Journal*, *30*(4), 740-763.

Liu, C., Xiang, X. and Zheng, L. (2017). Two decision models for berth allocation problem under uncertainty considering service level. *Flexible Services and Manufacturing Journal, 29*(3-4), 312-344.

López-Plata, I., Expósito-Izquierdo, C., Lalla-Ruiz, E., Melián-Batista, B., & Moreno-Vega, J. M. (2015, February). A Greedy Randomized Adaptive Search Procedure for Solving the Uncapacitated Plant Cycle Problem. *In International Conference on Computer Aided Systems Theory* (pp. 263-270). Springer, Cham.

Meisel, F. (2009). *Seaside operations planning in container terminals*. Berlin: Physica-Verlag

Nishi,T., Okura, T., Lalla-Ruiz, E. and Voß, S. (2016). A dynamic programming-based matheuristic for the dynamic berth allocation problem. *Annals of Operations Research*, pp 1-20.

Nishimura, E., Imai, A., & Papadimitriou, S. (2001). Berth allocation planning in the public berth system by genetic algorithms. *European Journal of Operational Research, 131*(2), 282-292.

Shadravan, S., Naji, H. R. and Bardsiri, V. K. (2019). The Sailfish Optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems. *Engineering Applications of Artificial Intelligence*, *80*, 20-34

Theofanis, S., Boile, M. and Golias, M. (2007). An optimization based genetic algorithm heuristic for the berth allocation problem. *IEEE international conference one Evolutionary Computation*. pp. 4439-4445.