sciendo

Transport and Telecommunication, 2024, volume 25, no. 2, 200-208
Transport and Telecommunication Institute, Lomonosova 1, Riga, LV-1019, Latvia
DOI 10.2478/ttj-2024-0015

ENHANCING TRANSPORTATION EFFICIENCY WITH OPTIMAL CONTAINER PLACEMENT USING THE BAT ALGORITHM

Yachba Khadidja[1], Feghoul Imane Amina[2], Belalia Sif Eddine[3]

[1]Oran Graduate School of Economics
LIO Laboratory
Oran, Algeria, Bir El Djir
yachba.khedidja@ese-oran.dz
[2]Relizane University
Relizane, Algeria, Bourmadia, Relizane
feghoul.imane.amina@gmail.com
[3]Relizane University
Relizane, Algeria, Bourmadia, Relizane
belalia.sifeddine@gmail.com

The objective of this article is to provide an in-depth exploration of the complex task of container storage at seaports, a problem characterized as one of the challenging NP (Non-Deterministic Polynomial time) problems. Seaports are faced with the dilemma of accommodating a finite number of containers due to the constrained surface area available, making the management of container storage operations a formidable task.

To address this challenge, the present study leverages a meta-heuristic approach aimed at identifying an optimal storage plan for containers within a storage area. This approach is informed by insights drawn from bat swarm intelligence, commonly known as the Bat Algorithm. By integrating principles from this nature-inspired algorithm, the authors seek to develop a robust solution for optimizing container storage strategies in seaports. This approach takes into account several critical constraints, including container travel distances and considerations related to container type and departure dates.

Keywords: Storage area, container, port, stack, container placement problem, optimization, bio-inspired method, the bat algorithm

1. Introduction

The handling of containers during import or export may include many movements or unnecessary rearrangements that involve unstacking several containers to reach a specific container below the stack (Yachba *et al.*, 2016).

In operations, redesigns are crucial because they lead to unnecessary movement of containers. It is common practice to determine the optimal storage of the containers to minimize the time required to perform the rearrangements during the processing (Singgih *et al.*, 2016).

The authors of this article focus on the issue of container storage. The following questions arise as a result:

How do you determine the correct location of containers in a storage area when they arrive at a terminal? Are there any temporary containers there?

- How do you minimize the waiting time for ships at the docks?
- What can we do to reduce the cost of container storage?

Through this contribution, the authors aim to reduce the cost of container storage and increase the efficiency of transport companies by minimizing the amount of time a container spends in storage. Essentially, they are reviewing the placement of containers in the storage area (taking into account the type of each container, and taking into account the time the container leaves the storage area) while minimizing the number of unnecessary and unproductive movements.

In order to manage the stocking zone effectively, a place for the container must be located optimally. As a response, the authors propose in this paper a technique for placing containers based on an optimization algorithm: Bats. As stated above, this technique takes into account the respective contraindications: the distance travelled, the type, and the date of departure of the container.

2. Related works

One of the most widely studied problems in combinatorial optimization is the Container Storage Problem.

In this area, several searches have been conducted; the following table presents the most recent searches in chronological order.

Table 1. Comparative table of related works

Author	Proposed algorithm	Key contributions
(Singgih *et al.*, 2016)	An integrated control system that aims to coordinate the operation of different types of equipment.	At container terminals, automation ensures precise and well-organized container transport.
(Bendaoud & Yachba, 2017)	The AHP method	Minimizes the total number of unnecessary movements while respecting the dynamic constraints of space and time.
(Amrani *et al.*, 2018)	An ant colony algorithm integrated into a decision-making model.	PSCs are solved by describing a decision model that solves and optimizes the storage space available at a seaport to handle the arrivals of full containers.
(Chang *et al.*,2019)	Based on a sliding planning horizon approach, a new heuristic algorithm was devised	A new model for allocating outgoing container locations is presented in order to reduce redesign operations Stowage, containers from previous planning periods, and departure time for containers.
(Bilican *et al.*,2020)	A two-step heuristic solution methodology based on whole programming formulations (IP), then algorithms using permutation heuristics (SH).	Stowage of containers with stability constraints. The authors of this paper have developed a Linear Mixed Integer (MILP) programming formulation to generate load plans while minimizing the total cost associated with over-stowage of cutting moments.
(Hsu *et al.*, 2021)	Hybrid approaches (Genetic Algorithms (GA), optimization of particle swarms (PSO), PSO subgroups (SGPSO)).	Address the Yard Crane Planning Problem (YCSP) and Yard Truck Planning Problem (YTSP) simultaneously in the area side yard of a container terminal.
(Tahiri *et al.*, 2022), (Tahiri *et al.*, 2020)	A multi-criteria approach (Electre 2) for placing containers on a ship	Considers several types of containers. Provides a storage plan based on a multi-criteria decision support.

As a result of studying previous work, the authors discovered that the bat algorithm does not apply to the container storage problem, so this algorithm was chosen. The authors were motivated to choose this algorithm because it is bio-inspired, which they adapted according to the PSC while considering the constraints: Type, distance, and departure date of container.

This work forms a part of the research work in the field of decision support system, transport, Maritime transportation (Yachba *et al.*, 2016; Bendaoud & Yachba, 2017), logistic (Yachba *et al.*, 2021), optimization (Belayachi *et al.*, 2017; Amrani *et al.*, 2018; Yachba *et al.*, 2018) and multicretaria methods (Yachba *et al.*, 2015; Yachba *et al.*, 2018; Tahiri *et al.*, 2022; Tahiri *et al.*, 2020).

3. The proposed approach

In this section, the authors delve into the methodology employed to identify the optimal locations for container storage. The objective is to elucidate how the Bat Algorithm and the principles of bat swarm intelligence are used to determine the best container locations within the storage area.

3.1. Bat Algorithm

The modelling of the echolocation hunting behaviour of these micro bats led to the creation of the Bat Algorithm by X.S. Yang (Yang ,2021; Benmostefa & Fizazi, 2013).

Among the advantages of the standard bat algorithm is its ability to achieve rapid convergence during the initial stages of moving from exploration to exploitation. As a result, it makes for an efficient algorithm when a quick fix is required. Many changes have been made in order to improve performance (Fister *et al.*, 2013).

As previously mentioned (Bedboudi & Bouras, 2018), They aim to increase the diversity of the solution and improve the performance of the standard bat algorithm.

3.2. Initialization of the Bat Algorithm

Initially, n number of bats are randomly selected. A real-valued vector with dimension d describes each individual in the population. To generate the initial population (Bedboudi & Bouras, 2018), the following equation is used.

$$X_ij = X_minj + rand(0, 1) (X_maxj - X_minj), \quad (1)$$

where: i = 1,2, ..., n; j = 1,2, ..., d; X_maxj and X_minj are the upper limits and lower for dimension j (Bedboudi & Bouras, 2018).

3.3. Solution, Frequency, and Speed

The authors use virtual bats in simulations. It is important to define how their positions are updated.

For each iteration t, calculate the x_i and the velocity v_i in a two-dimensional search space. The current best solution is x*.

Using the following update equations [6], the previous rules can be translated into the new solutions x_i^t and velocities V_i^t at step t.

$$f_i = f_{min} + (f_{max} - f_{min})\,\boldsymbol{\beta}, \quad (2)$$

$$V_i^t = V_i^{t-1} + (X_i^t - X^*)\, f_i, \quad (3)$$

$$X_i^t = X_i^{t-1} + V_i^t, \quad (4)$$

where: $\beta \in [0, 1]$ is a random vector drawn from a uniform distribution.

X*: The current global best solution is determined by comparing all solutions among all n bats. While λ_i f_i is the speed increase, we use f_i to adjust the speed while fixing the other factor λ_i. Depending on the problem, f_i can take on a wide range of values depending on the domain, the size of the problem, etc.

Initially, each bat randomly receives a frequency which is uniformly derived from $[f_{min}, f_{max}]$. A new solution is generated locally for each bat when a solution is selected among the best common solutions using a transformation integrating a random factor (Bedboudi & Bouras, 2018),

$$X_{new} = X_{old} + \varepsilon A_t, \quad (5)$$

where $\varepsilon \in [-1, 1]$ is a random number, A_t represents the average volume of all bats at that processing step.

The update of speeds and positions of bats is similar to the procedure standard optimization of particle swarms (Induja & Eswaramurthy, 2016; Ramesh *et al.*, 2013).

Considering Controlling the range of movement of invading particles, Bat can be thought of as a combination of particle swarm optimization and intensive local search, controlled by volume and pulsation rate (Bedboudi & Bouras, 2018).

3.4. Updating the Volume and Pulse Rate

At each iteration, the volume Ai and the pulsation rate r_i must be updated. When the intensity decreases once the bat has found its prey, when the pulsation rate r_i increases, the volume can be selected as a convenience value. When the volume reaches the minimum Amin, it means the bat has found its prey and has stopped making noise. During iteration, the volume Ai and rate r are updated based on the following equations (Bedboudi & Bouras, 2018).

$$A_i^{t+1} = \boldsymbol{\alpha} A_i^t, \quad (6)$$

$$r^{t+1} = r_i^0\, [1 - \exp(-\gamma t)], \quad (7)$$

where: α and γ are Constants. The constant α is the same as the factor Simulation of annealing cooling.

For each $0 < \alpha < 1$ et $\gamma > 0$ such as $A_i^t \to 0$, $r_i^t \to r_i^0$, as $t \to \infty$. (8)

It is necessary to experiment with the parameters. The tones and pulse rates should be different for each bat. Randomization can make this possible. Volume and emission rates are only updated if the new

solutions are enhanced, which means that these bats move towards the optimal solution (Bedboudi & Bouras, 2018).

Objective Function : f(x), x = (x1, ..., xd)
Initiate the bat population xi and the velocity vi, i = 1.2, ...
Determine the pulse frequency fi of each position xi
Initiate the pulse rate ri and the volume Ai
While (t ≤ maximum number of iterations) do
Generate new solutions by adjusting the frequency and updating
speeds and positions/solutions. (equations 2, 3 ; 4).
If (rand > ri) Then
1. Select a solution from the best solutions
2. Generate a local solution around the selected best solution x* j (equation 5)
End if
If (rand < Ai and f (xi) < f(x*)) Then
1. Accept new solutions
2. Increase ri and decrease Ai (equations 6, 7)
End if
Find the best solution x*
Do
Display the results given by the best solution x*

Figure 1. Bat Algorithm

4. Adaptation of the algorithm for container storage problem

The authors present the adaptation of the container storage problem using the bat algorithm in the following table.

Table 2. Real view VS Artificial view

Real view in bat	**Artificial view of a PSC case**
Bat	A vector of n dimensions each one represents the container position
Insect	The right location (optimal location)

Standard BATs are continuous optimization algorithms, so we cannot use their standard continuous coding scheme directly to solve PSCs. In order to apply BAT to PSC, you have to modify in few steps:

4.1. Step 1: Initialize the Bat algorithm

An individual in a population is described by a vector of integer values, these values represent the identifiers of the stacks with coordinates:

- X: represents the coordinate in the x axis,
- Y: represents the coordinate in the y axis,
- Z: represents the floor identifier.

In This case, the initial population is generated randomly (an empty slot is selected at random from the available empty slots).

4.2. Step 2:

We permute the current location i by the location j equivalent to (i+1) (see figure 2) in order to respect the discontinuity of the problem (one remains on the interval of the positions (vector)).

Similarly, for equation (5), we permute x_old by j, equivalent to x_old + 1 or x_old - 1.

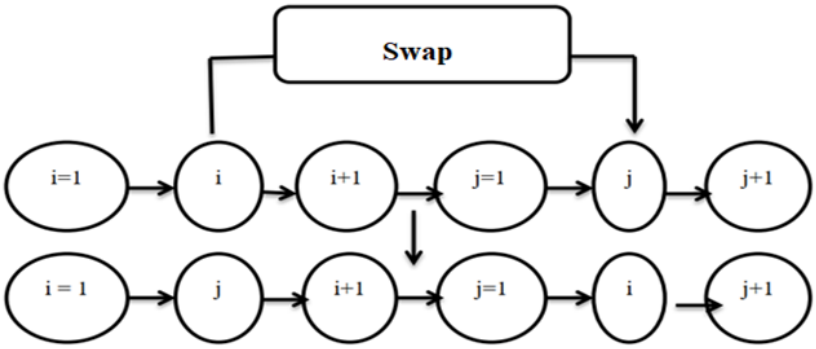


Figure 2. Location Search Operations

4.3. Step 3: Update:

This step represents updating the volume and heartbeat rate. This should be done by using the above formulas.

• **Indices:**

c = Container;
p = Stack;
e = Location in a stack;
s = Output port

Stack data:

Ne = Number of stacks;
ce = Number of free slots in p;
re = Dimension of stack e (20 feet, or 30 feet, or 40 feet);
te = Start date of the container which is already at the top of p, at start of storage operations;
X, Y = Stack coordinate;
x, y = Port coordinate;

Container data:

N = Number of containers;
Rc = Container dimension c (20 feet, or 30 feet, or 40 feet);
Tc = Departure Date of container k;

$$x_{p,e}^{c} = \begin{cases} 1 & \text{If container c is assigned to slot e of stack p} \\ 0 & else \end{cases}. \quad (9)$$

B = the number of containers with a departure date less than the departure date of container c.

$$B = \begin{cases} 0 \\ 1 \\ 2 \\ 3 \end{cases}$$

These 4 values (0,1,2,3) were set according to the number of floors in the stack.

$$\sum_{e=1}^{Ne} \sum_{p=1}^{Ce} x_{e,p}^{c} = 0, \forall\ c=1,\ldots..,\ N : R_c \neq r_e. \quad (10)$$

$$\sum_{c=1}^{N} B_c. \quad (11)$$

$$\sum_{c=1}^{N} |x_s - x_p| + |y_s - y_p|. \quad (12)$$

Stacks are only stacked if they have the same dimensions, because constraint 10 ensures this.

Constraint 11 ensures that new containers are stored in descending order of their start dates in each stack. This minimizes the need for redesigns.

Using constraint 12, you can calculate the distance between two containers that belong to the same category.

$$Minimize\ f(x) = \left(\sum_{c=1}^{N} |x_s - x_p| + |y_s - y_p| \right) + \sum_{c=1}^{N} B_c. \quad (13)$$

The objective function (13) minimizes both the number of reshuffles and the total distance between storage locations and outputs.

5. Results and Discussion

The authors present the adaptation of the container storage problem using the bat algorithm in the following table.

The dataset processed in this work was randomly generated. In order to better explain the proposed contribution, considering the following scenario:

The containers are grouped into 3 categories (Type 1, Type 2, Type 3) according to their sizes. Containers of the same type should be stored in stacks of the same type. We assume that we have the following configuration:

For a total number of stacks = 6:

Table 3. Stacks Configuration

Number of stacks type 1	2
Number of stacks type 2	3
Number of stacks type 3	1
Total number of stacks	6

For a total number of containers (to place) = 180:

Table 4. Containers Configuration

Number of containers type 1	100
Number of containers type 2	40
Number of containers type 3	40
Total number of containers	180

The number of bats (BAT) in this example is 10 and the number of iterations is 200. The parameters in this case study are:

Table 5. Bat parameters

Parameters	**Values**
R	0.001
ϒ	0.9
A	0.9

The following table illustrates the first step of the BAT algorithm (the initial BAT generation).

In the following table, we show the first step of the BAT algorithm (the initial generation of the BAT).

Table 6. An initial bat fragment

BAT Number	BAT N° 1	BAT N° 2	…….	BAT N° 10
Fitness	25.05%	20.80%	…….	24.50%
C-1	Stack 2 : (4,4,2)	Stack 3 : (2,1,2)	…….	Stack 3 : (1,2,2)
C-2	Stack 3 : (2,3,3)	Stack 3 : (5,1,2)	…….	Stack 1 : (1,1,4)
C-3	Stack 3 : (4,2,3)	Stack 3 : (3,2,1)	…….	Stack 1 : (1,4,3)
……	…….	…….	…….	…….
C-140	Stack 1 : (2,1,2)	Stack 2 : (4,1,3)	…….	Stack 1 : (5,1,3)

For this example, BAT N°2 of iteration 200 Tableau (4.7) is optimal. The container (C_1) should be stored in slot (2.3) floor 3 of stack 3.

Table 7. The optimal solution

BAT Number	BAT N° 2
Fitness	67.75%
C-1	Stack 3 : (2,3,3)
C-2	Stack 2 : (3,1,3)
C-3	Stack 3 : (5,1,2)
…..	…..
C140	Stack 1 : (2,4,4)

In the case study, we see the Bat algorithm minimized the costs of the task, the initial cost was 358950, it passed to a cost of 205600 in a time of execution of 3 seconds (a significant cost reduction).

6. Analyze Comparative

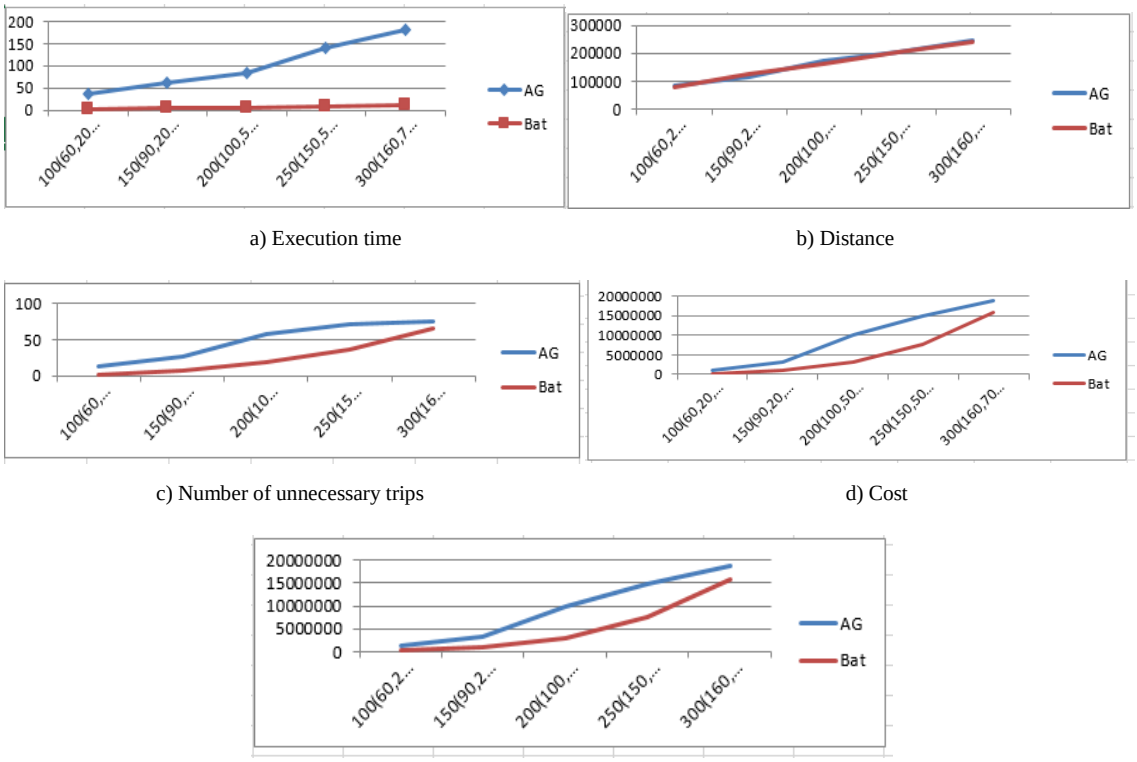
To demonstrate the effectiveness of the proposed method, it is necessary to compare the results found with another bio-inspired approach (genetic algorithm). Both algorithms were tested in the same working environment and with the same configuration.

A number of comparison parameters were used in the test (Execution time, Distance travelled, Number of useless movements, Cost of placement, Quality of the solution). The following table illustrates the results of the tests using the same parameters for the two methods (bats and genetics).

Table 8. Comparison results

Comparison parameters	**Results**		
Execution time (MS)	**Containers number**	**Genetic algorithm**	**Bats**
	100 (Type 1 : 60, Type 2 : 20, Type 3 : 20)	35.37	2.73
	150 (Type 1 : 90, Type 2 : 20, Type 3 : 20)	61.44	4.2
	200 (Type 1 : 100, Type 2 : 50, Type 3 : 50)	84.98	5.42
	250 (Type 1 : 150, Type 2 : 50, Type 3 : 50)	140.48	7.15
	300 (Type 1 : 160, Type 2 : 70, Type 3 : 70)	181.68	9.9
Distance	**Containers number**	**Genetic algorithm**	**Bats**
	100 (Type 1 : 60, Type 2 : 20, Type 3 : 20)	83340	80840
	150 (Type 1 : 90, Type 2 : 20, Type 3 : 20)	116746	128250
	200 (Type 1 : 100, Type 2 : 50, Type 3 : 50)	172712	163730
	250 (Type 1 : 150, Type 2 : 50, Type 3 : 50)	206108	206270
	300 (Type 1 : 160, Type 2 : 70, Type 3 : 70)	247090	242340
Number of unnecessary trips	**Containers number**	**Genetic algorithm**	**Bats**
	100 (Type 1 : 60, Type 2 : 20, Type 3 : 20)	14	2
	150 (Type 1 : 90, Type 2 : 20, Type 3 : 20)	27	8
	200 (Type 1 : 100, Type 2 : 50, Type 3 : 50)	58	19
	250 (Type 1 : 150, Type 2 : 50, Type 3 : 50)	72	37
	300 (Type 1 : 160, Type 2 : 70, Type 3 : 70)	76	65
Cost	**Containers number**	**Genetic algorithm**	**Bats**
	100 (Type 1 : 60, Type 2 : 20, Type 3 : 20)	1166760	161680
	150 (Type 1 : 90, Type 2 : 20, Type 3 : 20)	3152142	1026000
	200 (Type 1 : 100, Type 2 : 50, Type 3 : 50)	10017296	3110870
	250 (Type 1 : 150, Type 2 : 50, Type 3 : 50)	14839776	7631990
	300 (Type 1 : 160, Type 2 : 70, Type 3 : 70)	18778840	15752100
Quality of solution	**Containers number**	**Genetic algorithm**	**Bats**
	100 (Type 1 : 60, Type 2 : 20, Type 3 : 20)	34.98	89.17
	150 (Type 1 : 90, Type 2 : 20, Type 3 : 20)	60.91	87.01
	200 (Type 1 : 100, Type 2 : 50, Type 3 : 50)	37.82	80.99
	250 (Type 1 : 150, Type 2 : 50, Type 3 : 50)	39.34	70.82
	300 (Type 1 : 160, Type 2 : 70, Type 3 : 70)	63.74	72.05

According to the comparison results, the bat algorithm gives better results regardless of the number of containers processed. From the above table, we can see that the bat algorithm reduces the execution time, the distance traveled, the quality of the solution, or the overall placement cost. An overview of the comparison tests is shown in the following figure. According to Figure 3, the bat algorithm is faster and more efficient than the genetic algorithm.



a) Execution time

b) Distance

c) Number of unnecessary trips

d) Cost

e) Solution quality

Figure 3. Comparisons between the Bats algorithm and the genetic algorithm

7. Conclusions and Perspectives

The work presented in this article is part of the container placement problem, a problem that has been studied extensively. This problem has been addressed previously by the authors using several optimization and multi-criteria decision support methods.

The problem has been solved in this article using the bat algorithm, which has never been used before for this kind of problem. The proposed approach has proven effective in solving container placement problems regardless of the size of the problem (number of containers).

The authors plan on combining the bat algorithm with another meta-heuristic in the future to test the contribution of hybridization to the cost of placing containers, as well as taking other constraints into account and seeing the behavior of the method.

Acknowledgements

The authors would like to thank the Directorate General for Scientific Research and Technological Development (DGRSDT), an institution of the Algerian Ministry of Higher Education and Scientific Research, for their support on this work.

References

1. Amrani, F., Yachba, K., Belayachi, N., & Hamdadou, D. (2018) A decision-making system for container storage management in a seaport using the ant-colony optimisation algorithm. *International Journal of Management and Decision Making*, 17(3), 348-367.

2. Belayachi, N., Gelareh, S., Yachba, K., & Bouamrane, K. (2017) The Logistic of Empty Containers′ Return in the Liner-Shipping Network. *Transport and Telecommunication Journal*, 18(3), 207-219.
3. Bendaoud, Z., & Yachba, K. (2017) Towards A Decision Support System for Optimization of Container Placement in a Container Terminal. *International Journal of Strategic Information Technology and Applications (IJSITA)*, 8(3), 59-72.
4. Benmostefa, S., & Fizazi, H. (2013) Classification automatique des images satellitaires optimisée par l'algorithme des chauves-souris. *Revue française de photogrammétrie et de télédétection*, (203), 11-17.
5. Bilican, M. S., Evren, R., & Karatas, M. (2020) A Mathematical model and two-stage heuristic for the container stowage planning problem with stability parameters. *IEEE Access*, 8, 113392-113413.
6. Boulouard, Z., El Haddadi, A., Bouhafer, F., El Haddadi, A., Koutti, L., & Dousset, B. (2018) Bat-cluster: A bat algorithm-based automated graph clustering approach. *International Journal of Electrical and Computer Engineering*, 8(2), 1122.
7. Bedboudi, A., & Bouras, M. C. (2018) Amélioration de l'algorithme de chauve-souris par modification de règles d'évolution et introduction de mécanisme de croisement. *Synthèse: Revue des Sciences et de la Technologie*, 36, 1-8.
8. Chang, Y., Zhu, X., & Haghani, A. (2019) The outbound container slot allocation based on the stowage plan in rail–water intermodal container terminals. *Measurement and Control*, 52(5-6), 509-525.
9. Fister Jr, I., Fister, D., & Yang, X. S. (2013) A hybrid bat algorithm. *arXiv preprint arXiv*: 1303.6310.
10. Hsu, H. P., Tai, H. H., Wang, C. N., & Chou, C. C. (2021) Scheduling of collaborative operations of yard cranes and yard trucks for export containers using hybrid approaches. *Advanced Engineering Informatics*, 48, 101292.
11. Induja, S., & Eswaramurthy, V. P. (2016) Bat algorithm: an overview and its applications. *International Journal of Advanced Research in Computer and Communication Engineering*, 5(1), 448-451.
12. Ramesh, B., Mohan, V. C. J., & Reddy, V. V. (2013) Application of BAT algorithm for combined economic load and emission dispatch. *Journal of Electrical Engineering*, 13(2), 6-6.
13. Singgih, I. K., Jin, X., Hong, S., & Kim, K. H. (2016) Architectural design of terminal operating system for a container terminal based on a new concept. *Industrial Engineering and Management Systems*, 15(3), 278-288.
14. Tahiri, H., Khadidja, Y., & Bouamrane, K. (2022) Towards a Multi-Criteria Decision Approach to Solving the Container Storage Problem: Container Ship Loading Component. *International Journal of Organizational and Collective Intelligence (IJOCI)*, 12(1), 1-21.
15. Tahiri, H., Yachba, K., & Bouamrane, K. (2020, December). A container placement approach in a container-ship based on the electre 2 method. In: *4th International Symposium on Informatics and its Applications (ISIA)*, 1-5. IEEE.
16. Yachba, K., Gelareh, S., & Bouamrane, K. (2016) Storage management of hazardous containers using the genetic algorithm. *Transport and Telecommunication Journal*, 17(4), 371-383.
17. Yachba, K., Bendaoud, Z., & Bouamrane, K. (2018) A Technique for Resolution of the Assign-ment Problem Containers in a Container Terminal. In: *Hanbok of Research on Biomimicry in Information Retrieval and Knowledge Management*, 120-132. IGI Global.
18. Yachba, K., Bouamrane, K., & Gelareh, S. (2015, October) Containers storage optimization in a container terminal using a multimethod multi-level approach. In: *The International Conference on Computers & Industrial Engineering (CIE45)*, 28-30.
19. Yachba, K., Bendaoud, Z., & Bouamrane, K. (2018) Toward a Decision Support System for Regulation in an Urban Transport Network. *International Journal of Strategic Information Technology and Applications (IJSITA)*, 9(2), 1-17.
20. Yachba, K., Chaabane A., Benadda, M.A (2021) *Un système d'aide à la décision pour l'optimisation de processus de distribution des produits finaux.* Séminaire international sur les mathématiques et l'informatique, Février, Oran .
21. Yang, X. S. (2011) Bat algorithm for multi-objective optimization. *International Journal of Bio-Inspired Computation*, 3(5), 267-274.