

Enhanced Harris Hawks Optimization Algorithm with Integrated Tabu Search for 3D Bin Packing Problems: Application in Logistics Industries

Yasser M. Ayid^{1*}, Mohamed Meselhy Eltoukhy², Alaa Mokhtar³, Rabie Mosaad⁴

Department of Computer Science and Artificial Intelligence-College of Computer Science and Engineering,
University of Jeddah, 23890 Jeddah, Saudi Arabia¹

Department of Information Technology-College of Computing and Information Technology at Khulais,
University of Jeddah, 23890 Jeddah, Saudi Arabia²

Canadian International College, School of Business Technology, Zayed Campus, 6th of October 12585, Egypt³

Department of Operations Research and Management-Faculty of Graduate Studies for Statistical Research,
Cairo University, Giza 12613, Egypt⁴

Abstract—The Harris Hawks Optimization (HHO) algorithm has proven to be an effective metaheuristic tool for addressing complex optimization challenges. This study outlines significant modifications and enhancements proposed for the standard HHO algorithm to improve its performance on complex three-dimensional bin packing problems (3D-BPP). The proposed modifications introduce a hybrid metaheuristic framework that synergistically merges the global search capabilities of an enhanced HHO with the local intensification strength of Tabu Search (TS), along with several additional algorithmic improvements. In this study, we present an enhanced Harris Hawks Optimization algorithm integrated with Tabu Search for 3D bin packing problems (EHHITS). This problem is particularly significant in logistics and warehousing, where improving space utilization directly contributes to cost reduction. The primary objective is to maximize the occupied volume of the container by efficiently packing smaller rectangular items (boxes) without overlaps. The proposed EHHITS algorithm employs an inner heuristic, a heuristic local search, to construct an optimized sequence of boxes, which is then used to determine their placement within the container. To evaluate the proposed algorithm, we benchmarked its performance against standard test cases from the literature. The experimental results demonstrate its effectiveness and validity.

Keywords—Logistics; container loading problems; harris hawks optimization; tabu search; wall-building

I. INTRODUCTION

The 3D bin packing problem is a critical challenge in logistics and transportation. It focuses on efficiently arranging a collection of items within a container to achieve maximum utilization of its available volume. This optimization is essential for companies seeking to minimize shipping costs and enhance customer satisfaction in a globalized market. The 3D bin packing problem becomes increasingly difficult to solve optimally as the number of items grows [1]. The goal is to pack a set of n items, each with defined dimensions, into a single container of known size. The core constraints require that no item exceeds the container's dimensions and that items do not overlap in space. The objective is to achieve the highest possible container utilization rate [2], [3].

HHO is a population-based metaheuristic that emulates the collective predatory behavior of Harris hawks. HHO outperforms many other algorithms due to its inherent strengths [4]. A key advantage is its strong balance between global exploration and local exploitation, exhibiting high optimization efficiency in both phases. This robust search strategy gives HHO strong solution-finding capability, making it particularly effective for challenging NP-hard problems [5]. Moreover, the algorithm dynamically adjusts its search behavior throughout the optimization process [6]. As a result, these superior characteristics have led to the widespread and recent adoption of HHO and its variants in solving numerous real-world problems.

In addition to its strong theoretical foundation, the versatility of the HHO algorithm is demonstrated through its successful application across a wide range of fields. Recent studies highlight its effectiveness in critical domains such as estimating parameters of single-diode and three-diode photovoltaic models [7], [8]. Beyond renewable energy, HHO has been employed in various image segmentation tasks including digital mammogram and satellite image segmentation [9], as well as in fault detection for anti-friction bearings [10]. Recent publications further confirm the effectiveness and continued relevance of the HHO algorithm.

The motivations of this study are as follows:

- To develop an enhanced HHO algorithm integrated with Tabu Search for solving the 3D bin packing problem.
- To introduce and demonstrate an enhanced version of the algorithm, namely EHHITS, through the integration of effective search mechanisms and improved solution-generation strategies, leading to robust and competitive performance on a comprehensive set of three-dimensional packing instances.

This research presents an improved variant of the HHO algorithm integrated with Tabu Search. The proposed algorithm is evaluated against well-established approaches in the literature within the context of 3D bin packing. The structure

*Corresponding author.

of this study is organized to thoroughly address the 3D bin packing problem using an HHO-based methodology. The study begins with a comprehensive literature review in Section II, evaluating existing heuristic and metaheuristic algorithms for 3D bin packing and comparing them with alternative solution techniques. Section III explains the HHO algorithm. Section IV introduces the core contribution of this work. Finally, Section V presents a detailed analysis of the computational results, followed by concluding remarks that summarize the findings and validate the effectiveness of the proposed approach.

II. LITERATURE REVIEW

The Harris Hawks Optimization algorithm has attracted considerable interest in the academic community since its conception, thanks to its structural flexibility and strong performance in tackling complicated optimization problems. As shown in Table I, the literature shows a strong trend toward hybridizing and improving the underlying HHO architecture to address domain-specific problems.

Researchers addressing the 3D bin packing problem primarily rely on three categories of optimization algorithms [34]. Although exact methods guarantee optimal solutions, they quickly become computationally infeasible as the number of boxes increases. In contrast, metaheuristic algorithms cannot guarantee a global optimum, but they can produce near-optimal and practically acceptable solutions within reasonable computation time [35]. For this reason, Metaheuristic Algorithms (MHAs) have become the preferred choice among researchers tackling 3D bin packing. There are three main strategies for applying MHAs to solve 3D bin packing:

1) *Direct application*: In this basic approach, MHAs are applied "as-is" without modification. It is the simplest strategy and often serves as a baseline for performance comparison.

2) *Modification*: This strategy involves tailoring an existing MHA to better address the specific characteristics of the 3D bin packing problem. Adjustments may include parameter tuning or incorporating new search mechanisms to improve efficiency and solution quality.

3) *Hybridization*: The most complex approach, hybridization combines two or more MHAs or integrates an MHA with a different type of algorithm. The objective is to exploit the complementary strengths of each component to achieve superior performance.

The three-dimensional container loading problem (3D-CLP) deals with placing rectangular cargo items inside a container while making the best possible use of the available space. The objective is to maximize the proportion of the container volume occupied by the loaded items, as expressed in Eq. (1).

$$\max f = \frac{\sum_{i=1}^n v_i \cdot k_i}{L \times W \times H} \quad (1)$$

where, n denotes the number of cargo types, v_i represents the volume of each cargo item of type i , and k_i denotes the number of items of type i loaded into the container. The parameters L , W , and H represent the length, width, and height of the container, respectively.

TABLE I. SEVERAL KEY STUDIES CONCERNING HHO

Approach	Application area	Ref.
HHO-Multi-Layer Perceptron	Two-layer foundation soils	[11]
HHO and Simulated Annealing	Feature selection	[12]
HHO	Design of microchannel heat sinks	[13]
Vibrational HHO	Fault diagnosis of rolling bearing	[14]
Enhanced HHO	Breast Cancer Detection	[15]
Whippy HHO	PV solar cells	[16]
Artificial Neural Network	Stability of the soil slopes	[17]
HHO Microgrid	Design optimized	[18]
Sine-Cosine HHO	Feature selection CEC2017	[19]
Deep Neural Network - HHO	Evaluating slope stability	[20]
Improved Harris Hawks Optimizer	Global optimization	[21]
Improved HHO	Collaborative optimization	[22]
Multi-Objective Binary HHO	Gene selection	[23]
Deep Learning - HHO	Skin lesion analysis	[24]
HHO with Chaotic Sequence	Engineering problems	[25]
Hybrid HHO with Chaotic Maps	Mechanical design problems	[26]
Improved HHO	Path planning for mobile robots	[27]
HHO	Machine learning	[28]
Enhanced HHO	Task scheduling in cloud computing	[29]
HHO	Power systems and energy management	[30]
Improved HHO	Optimizing deep learning models	[31]
Enhancing HHO	Global optimization problems	[32]
Hybrid Aquila optimizer HHO	CNN hyper parameter	[33]

In practical container loading, several constraints must be considered in addition to the dimensional and volume limitations. These include the available space for the cargo, cargo stability, the center of gravity of the container, and the prevention of overlap between cargo items. Such constraints can have a direct impact on the quality of the obtained packing solution. Therefore, the characteristics of the specific packing problem should be carefully considered. Based on the test dataset, the following constraints are considered for the 3D-CLP:

- The cargo items and the container are rectangular or square in shape.
- The cargo items are placed within the positive coordinate directions of the container.
- The available cargo can be loaded without considering crushing of the cargo items in the lowest layer.
- The cargo items are placed parallel to the axes of the container, and overlapping between cargo items is not allowed.
- The dimensions of each cargo item must be smaller than the corresponding dimensions of the container.

These assumptions provide a simplified yet suitable representation of the 3D-CLP considered in this study. The complexity of the 3D bin packing problem can vary significantly depending on the number and characteristics of the items to be loaded. These variations range from the homogeneous case as shown in Fig. 1, where all items share the same orientation and dimensions, to the strongly heterogeneous case, which involves a wide variety of item sizes. The problem becomes more challenging to solve as item diversity increases [36].

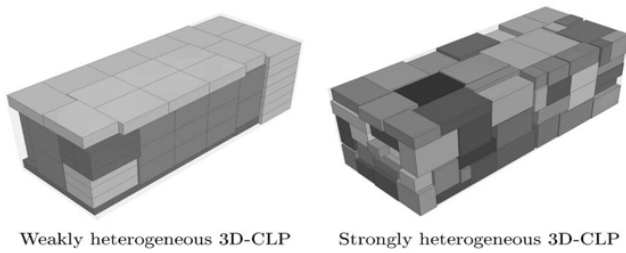


Fig. 1. Weakly and strongly heterogeneous 3D bin packing [37].

A. Packing Heuristics

Fanslau and Bortfeldt [38] classified the wide range of approaches for solving the 3D bin packing problem based on both the packing heuristic and the type of solution method employed. Packing heuristics define the specific strategies used to place items within the container, these strategies are categorized into several groups.

1) *Wall-building*: Bortfeldt and Gehring [2] introduced a heuristic wall-building within a hybrid Genetic Algorithm (GA) designed for packing a single container with boxes of varying sizes. In their approach, the stowage plan is represented using a complex, problem-specific data structure. This structure enables the construction of loading plans of several vertical layers, each containing multiple boxes. GA then optimizes these layer-based arrangements to produce highly efficient packing solutions.

2) *Stack-building*: Bischoff and Ratcliff [39] identified significant gaps in the current theoretical framework on Container Loading Problems. They argued that current methods are inadequate for many practical situations and that fundamentally new approaches are required. Moreover, they emphasized that the development of such methods should include extensive experimentation to assess their strengths and weaknesses.

3) *Horizontal layer-building*: Terno et al. [40] implemented their approach as a foundation for further research, particularly in the context of the Multi-Product Lot-Sizing Problem (MPLP). They also considered a more general scenario where the demand for a certain type of item is not a fixed quantity but is defined by a lower and upper bound. This adds another layer of complexity to the problem, making it more representative of real-world logistics.

4) *Block-building*: Eley [41] developed a homogeneous block arrangement to solve single-container problems. The use of these blocks offers several advantages for practical applications. When tested against standard benchmarks, it performed well in terms of volume utilization in heterogeneous problems. This demonstrates the effectiveness of the block-building approach in complex real-world scenarios.

5) *Guillotine cutting*: Morabito [42] addressed two-dimensional unconstrained complex cutting problems. Where a rectangular plate is divided into smaller rectangles using cuts perpendicular to a single edge. Unlike traditional guillotine methods, this approach allows for more complex cutting patterns. Morabito's method uses a graph-search technique to find an efficient cutting plan, which can be applied to container loading as a way to organize and place items.

As illustrated in Fig. 2 the evolution of container loading and packing methodologies demonstrates a clear progression from basic geometric partitioning to sophisticated heuristic optimization. Early techniques, such as the guillotine-cutting approach, relied on a structured slicing tree to divide space, whereas subsequent developments introduced more practical organizational logic through vertical stack-building and horizontal layer-building. As the field matured, methodologies shifted toward vertical wall-building and complex block-building strategies, which utilize tree-search algorithms to maximize spatial density. This transition reflects a shift in industrial research from solving simple rectangular constraints toward achieving high-volume utilization and structural stability in complex logistical environments.

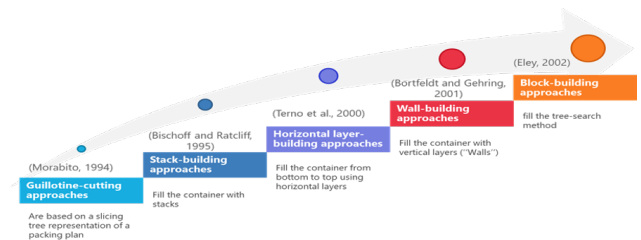


Fig. 2. 3D bin packing according to packing heuristic.

B. Metaheuristics Search Strategies

Metaheuristics search strategies have become the preferred approach for solving complex optimization problems over the past decade. Akl et al. [43] illustrate the workings of these population-driven techniques as in Fig. 3. The process begins with a diverse initial population that undergoes continuous refinement through fitness evaluations. At the heart of the strategy is a dynamic balance between exploration seeking new possibilities and exploitation focusing on the most promising solutions. This cycle iterates intelligently, progressively narrowing down options until the stopping criteria are fulfilled and the optimal solution is achieved. A notable example is the Tabu Search Approach (TSA), which has been effectively applied to the Container Loading Problem. Bortfeldt and Gehring [2] developed a parallel Tabu Search algorithm for solving the single-container loading problem, where multiple cooperating search threads are used to enhance computational efficiency. Their extensive comparative experiments demonstrated that even the sequential version of TSA can achieve high container volume utilization, underscoring the strength of this metaheuristic method in tackling difficult combinatorial optimization tasks.

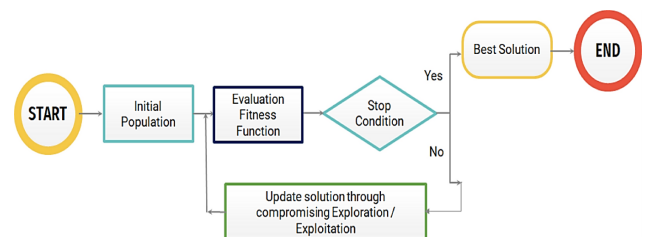


Fig. 3. Conceptual flowchart of population-driven metaheuristic optimization techniques [43].

C. Tree-Search or Graph-Search Methods

Arenales and Morabito [44] proposed a graph-search approach for 2D nonguillotine cutting problems involving orthogonal cuts. Their method represents first-order cutting patterns using an AND/OR graph, which allows different spatial configurations to be explored systematically. As shown in Fig. 4, a binary-tree search is used to evaluate alternative configurations at each decision node, where each cutting operation may produce either two or five sub-rectangles. The search combines backtracking with hill-climbing to explore promising solutions while allowing alternative configurations to be revisited when needed. To reduce unnecessary exploration, symmetry-elimination rules were applied to remove redundant cutting patterns.

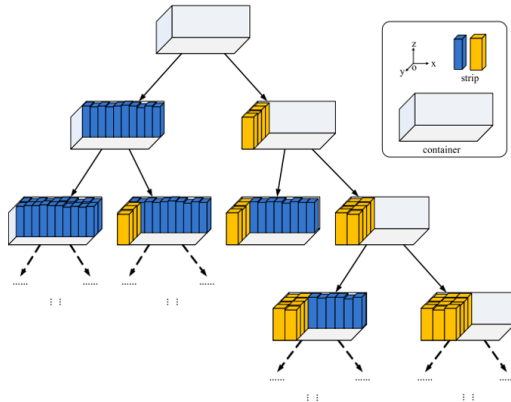


Fig. 4. Binary-tree-based search for efficient container loading plans [45].

D. Conventional Heuristics

Bischoff and Ratcliff [39] identified significant limitations in the existing literature on container loading, arguing that conventional constructive and improvement methods are insufficient for many real-world applications. They emphasized that fundamentally new approaches are required to address practical challenges effectively. Furthermore, they highlighted the necessity of rigorous testing and evaluation to determine the strengths and weaknesses of these new methods, ensuring meaningful comparisons. To support this, they recommended that testing procedures should be standardized as much as possible.

III. HARRIS HAWKS OPTIMIZATION ALGORITHM

Heidari et al. [46] introduced the Harris Hawks Optimization (HHO) algorithm, inspired by the cooperative hunting strategies of Harris' hawks in nature. HHO is an innovative, swarm-based metaheuristic, renowned for both its computational efficiency and effectiveness. It utilizes a set of deceptively simple yet highly efficient exploration and exploitation operators, all governed by a dynamically adaptive structure, which makes it particularly well-suited for solving continuous optimization problems.

The HHO algorithm is structured around two core phases, reflecting the hunting tactics of Harris' hawks: exploration and exploitation. The exploration phase emulates the birds'

initial search patterns, where candidate solutions are generated through stochastic processes to ensure comprehensive coverage of the search space. This transitions into the exploitation phase, which reflects the hawks' refined pursuit strategies, concentrating computational resources on regions exhibiting promising fitness values while progressively refining solutions. Fig. 5 provides a detailed schematic representation of these interconnected optimization phases, illustrating their respective operational mechanisms and the dynamic transition between broad exploration and focused exploitation in the HHO algorithm [46].

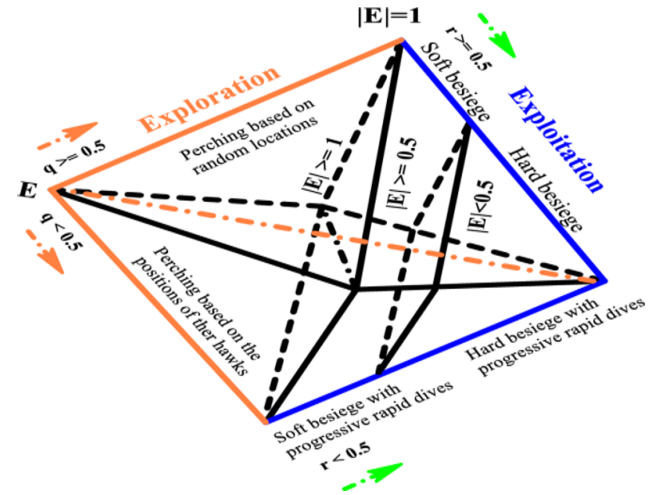


Fig. 5. Phases of HHO [46].

A. Exploration Phase

During this phase, Harris' hawks demonstrate random perching and patient observation to locate prey. Two perching strategies are selected based on a random value q , considering the positions of both family members and the prey. In the exploration phase, q is equally likely for either strategy. When $q < 0.5$, hawks adjust their perch according to the relative positions of their specifics and the rabbit. Alternatively, Harris's hawks perch on randomly selected locations within their home range. The exploration phase's updated position is represented in Eq. (2):

$$X(t+1) = \begin{cases} X_{\text{rand}}(t) - r_1 |X(t) - 2r_2 X_i(t)|, & q \geq 0.5 \\ X_{\text{best}}(t) - X_{\text{mean}}(t) - r_3 (L + r_4 (U - L)), & q < 0.5 \end{cases} \quad (2)$$

The variables and terms used in the model are defined as follows:

- q : A random number $\in [0, 1]$.
- $X(t)$: Position of i -th hawk at the t -th iteration.
- $X_{\text{best}}(t)$: The position of the rabbit at the t -th iteration.
- r_1, r_2, r_3, r_4 : Random numbers $\in [0, 1]$, updated in every iteration.
- L, U : Lower and upper bounds of variables.
- $X_{\text{rand}}(t)$: Selected random hawk from the population.

Eq. (3) represents the mean position of the current hawk population, with N indicating the total number of hawks:

$$X_{\text{mean}}(t) = \frac{1}{N} \sum_{i=1}^N X_i(t) \quad (3)$$

B. Transition from Exploration to Exploitation Phase

The Harris Hawks Optimization algorithm incorporates a transfer from an exploration phase to an exploitation phase, depending on the prey's escaping energy. In this algorithm, the prey's energy is represented as gradually decreasing during its escape behavior:

$$E = 2E_0 \left(1 - \frac{t}{T}\right) \quad (4)$$

As in Eq. (4), the energy of the escaping prey, defined as E and initialized as E_0 , controls the selection of either exploration or exploitation in the HHO algorithm. In case $|E| \geq 1$, the algorithm is in the exploration phase, while $|E| < 1$ indicates exploitation. E_0 is applied using Eq. (5):

$$E_0 = 2 \times \text{rand} - 1 \quad (5)$$

C. Exploitation Phase

In the exploitation stage, four unique chasing and attacking tactics are applied according to the prey's escaping energy and the hawks' pursuit patterns. The parameter r is used to select a chasing strategy depending on whether the prey successfully escapes ($r < 0.5$) or not ($r \geq 0.5$) before an attack.

D. Soft Besiege

When the probability of escape ($r \geq 0.5$) and the escaping energy ($|E| \geq 0.5$), the prey still possesses sufficient energy and attempts to escape. In response, the Harris' hawks softly surround the prey to deplete its remaining energy before launching an attack. The behavior of the Harris' hawks in this phase is modeled as follows:

$$X(t+1) = \Delta X(t) - E|JX_{\text{best}}(t) - X(t)| \quad (6)$$

$$\Delta X(t) = X_{\text{best}}(t) - X(t) \quad (7)$$

$$J = 2(1 - r_5) \quad (8)$$

Here, $\Delta X(t)$ explains the difference between the current prey's position and its position. J refers to a variable defined as random jump strength.

E. Hard Besiege

When the probability of escape ($r \geq 0.5$) and the escaping energy ($|E| < 0.5$), the prey's energy is low and the Harris' hawks readily encircle it before launching an attack. The positions of the hawks and prey are recalculated using the equation below:

$$X(t+1) = X_{\text{best}}(t) - E|\Delta X(t)| \quad (9)$$

Fig. 6 illustrates this process, modeled through the interaction between a single hawk and the prey.

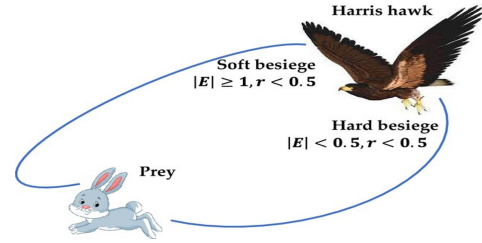


Fig. 6. Vector representation of the hard besiege mechanism [47].

F. Soft Besiege with Progressive Rapid Dives

When the prey's escape energy is high ($|E| \geq 0.5$) and ($r < 0.5$), the hawks carry out a gentle siege, conducting successive quick dives to incrementally refine the prey's location and movement direction. This behavior is modeled using the following equations:

$$Y = X_{\text{best}}(t) - E|JX_{\text{best}}(t) - X(t)| \quad (10)$$

$$Z = Y + S \times LF(D) \quad (11)$$

Where, D is the problem dimension, S is a $1 \times D$ random vector, and LF is the Lévy flight function, calculated via Eq. (12):

$$LF(x) = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}}, \quad \sigma = \left(\frac{\Gamma(1 + \beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\frac{\beta-1}{2}}} \right)^{\frac{1}{\beta}} \quad (12)$$

where, u and v are random values uniformly distributed in $(0, 1)$, and β is a constant with a default value of 1.5.

$$X(t+1) = \begin{cases} Y, & \text{if } F(Y) < F(X(t)) \\ Z, & \text{if } F(Z) < F(X(t)) \end{cases} \quad (13)$$

In this context, S represents a random vector, and the superior position between Y and Z is chosen as the subsequent location.

G. Hard Besiege with Progressive Rapid Dives

If $(|E| < 0.5)$ and $(r < 0.5)$, implying that the prey lacks sufficient energy to evade capture, the hawks execute a hard encirclement by reducing the distance between their mean position and the prey. This action is represented by the following equations:

$$Y = X_{\text{best}}(t) - E|JX_{\text{best}}(t) - X_{\text{mean}}(t)| \quad (14)$$

$$Z = Y + S \times LF(D) \quad (15)$$

$$X(t+1) = \begin{cases} Y, & \text{if } F(Y) < F(X(t)) \\ Z, & \text{if } F(Z) < F(X(t)) \end{cases} \quad (16)$$

Here, only the more optimal position between Y and Z is adopted as the next point. Algorithm 1 presents the standard HHO pseudocode, summarizing its main optimization steps and position-updating process. Fig. 7 depicts the overall HHO optimization framework, highlighting the exploration and exploitation phases and the associated hawk position-update mechanisms.

Algorithm 1 Pseudocode of HHO algorithm

Require: The population size N and maximum number of iterations T

Ensure: The location of rabbit and its fitness value

```

1: Initialize the random population  $X_i = (i = 1, 2, \dots, N)$ 
2: while (stopping condition is not met) do
3:   Calculate the fitness values of hawks
4:   Set  $X_{\text{best}}$  as the location of rabbit (best location)
5:   for (each hawk ( $X_i$ )) do
6:     Update the initial energy  $E_0$  and jump strength  $J$ 
7:      $E_0 = 2\text{rand}() - 1, J = 2(1 - \text{rand}())$ 
8:     Update the  $E$  using Eq. (4)
9:     if  $(|E| \geq 1)$  then ▷ Exploration phase
10:      Update the location vector using Eq. (2)
11:    end if
12:    if  $(|E| < 1)$  then ▷ Exploitation phase
13:      if  $(r \geq 0.5 \text{ and } |E| \geq 0.5)$  then ▷ Soft
14:        besiege
15:        Update the location vector using Eq. (6)
16:      else if  $(r \geq 0.5 \text{ and } |E| < 0.5)$  then ▷ Hard
17:        besiege
18:        Update the location vector using Eq. (9)
19:      else if  $(r < 0.5 \text{ and } |E| \geq 0.5)$  then ▷ Soft
20:        besiege with progressive rapid dives
21:        Update the location vector using Eq. (13)
22:      else if  $(r < 0.5 \text{ and } |E| < 0.5)$  then ▷ Hard
23:        besiege with progressive rapid dives
24:        Update the location vector using Eq. (16)
25:      end if
26:    end if
27:  end for
28: end while
29: return  $X_{\text{rabbit}}$ 

```

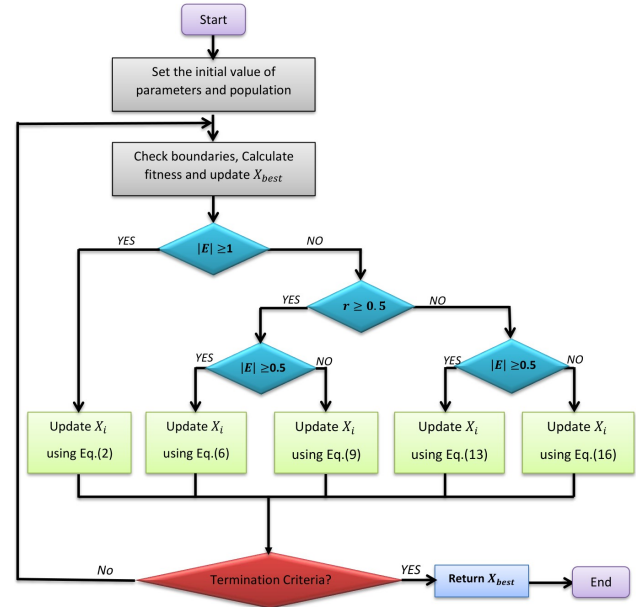


Fig. 7. HHO flowchart.

IV. THE PROPOSED ALGORITHM (EHHITS)

The complexity of the 3D Container Loading Problem necessitates a robust optimization approach capable of balancing global exploration with precise local refinement. This section details the proposed hybrid framework, which strategically integrates the HHO with an intensive Tabu Search (TS) mechanism, specifically tailored to maximize space utilization. The optimization process initiates with a pre-processing stage where the physical constraints of the container (L, W, H) and box dimensions are established. To ensure computational efficiency, the algorithm pre-calculates all unique spatial orientations for each box type, effectively pruning redundant search paths. To push the boundaries of packing density, the global search is augmented by an intensive Tabu Search, which acts as a high-precision intensification engine. Triggered periodically (every five iterations), this stage focuses exclusively on refining the current global best by navigating its immediate neighborhood through two primary structured moves:

1) *Sequence swapping*: Modifies the relative priority of items by exchanging their positions within the current packing sequence. This neighborhood operation enables the algorithm to explore alternative item-order configurations, which may lead to more efficient spatial arrangements, improved container utilization, and reduced unused space.

2) *Gaussian mutation*: Introduces subtle perturbations to fine-tune box orientations and placement coordinates, ensuring no potential spatial optimization is overlooked. To maintain search efficiency, the TS employs a Tabu List, which serves as a short-term memory to prevent the algorithm from cycling back to recently explored, unproductive configurations. However, this restriction is governed by an aspiration criterion: if a “tabu” move yields a utilization higher than the current fitness value.

A. The Robust Multi-Decoder and Extreme Point Logic

The proposed framework centers on a Robust Multi-Decoder—a specialized mechanism that bridges continuous optimization with discrete spatial constraints. Unlike conventional decoders that rely on a single heuristic, this approach evaluates each candidate solution through two competing packing paradigms to identify the most efficient configuration:

1) *Vertical layering strategy (column-based packing)*: Focuses on vertical stratification, prioritizing the stacking of items to maximize layered stability and container height utilization.

2) *Horizontal floor-filling strategy (wall-based packing)*: Focuses on floor-area coverage using a foundation-first approach to promote stable weight distribution.

The fitness value is the maximum utilization achieved by the evaluated packing strategies. An Extreme Point (EP) strategy manages the 3D packing space by dynamically updating feasible insertion points after each box placement, adding new candidate points and removing blocked or infeasible ones. The detailed execution flow and operational steps of the EHHITS algorithm are presented in Algorithm 2, while the pseudocode for the Intensive Tabu Search used during the intensification phase is outlined in Algorithm 3.

Algorithm 2 Pseudocode of the proposed EHHITS algorithm

Require: The population size N and maximum number of iterations T

Ensure: The location of rabbit and its fitness value (Max utilization)

```
1: Initialize  $X_i = (i = 1, 2, \dots, N)$ 
2: Calculate the fitness values of hawks using Robust Multi Decoder:
3: fitness values = max(core packer Z order, core packer X order)
4: while (stopping condition is not met) do
5:   Calculate the fitness values of hawks
6:   Set  $X_{\text{best}}$  as the location of rabbit (best location)
7:   for (each hawk ( $X_i$ )) do
8:     Update the initial energy  $E_0$  and jump strength  $J$ :
9:      $E_0 = 2\text{rand}() - 1$ ,  $J = 2(1 - \text{rand}())$ 
10:    Update  $E$  using Eq. (4)
11:    if ( $|E| \geq 1$ ) then ▷ Exploration phase
12:      Update the location vector using Eq. (2)
13:    end if
14:    if ( $|E| < 1$ ) then ▷ Exploitation phase
15:      Update location vector using Soft/Hard besiege or Levy flight (Eq. 6, 9, 13, 16)
16:    end if
17:  end for
18:  Apply boundary control and Evaluate new fitness value using Robust Multi Decoder
19:  if ( $t \bmod 5 = 0$ ) then ▷ Intensification phase
20:    Apply Intensive Tabu Search to the location vector  $X_{\text{best}}$  and fitness value
21:    Update  $X_{\text{best}}$  and fitness value
22:  end if
23: end while
24: return  $X_{\text{best}}$  (Rabbit Location)
```

TABLE II. PARAMETER SETTING

Parameter	Value	Description
β (Beta)	1.5	Control factor for the search mechanism
Rabbit Energy	[0,2]	The transition between exploration and exploitation
Iterations	300	The maximum number of cycles the algorithm runs for each problem instance
Population Size	30	The number of individual solutions, or “hawks,” used in the search process

Algorithm 3 Pseudo-code of Intensive Tabu Search

Require: Best location X_{best} , Current fitness, local search iterations (T_{iter})

Ensure: Refined X_{best} and its fitness value (Rabbit Energy)

```
1: Set  $\text{current}_x = X_{\text{best}}$ 
2: for  $j = 1$  to  $T_{\text{iter}}$  do
3:   Generate Neighbor hawk by applying Swap and Mutation on  $\text{current}_x$ 
4:   Calculate fitness neighbor using Robust Multi Decoder
5:   if (fitnessneighbor > Rabbit Energy) then
6:     Update  $X_{\text{best}} = X_{\text{Neighbor}}$  and Rabbit Energy = fitnessneighbor
7:   end if
8: end for
9: return  $X_{\text{best}}$ , fitness value (Rabbit Energy)
```

V. COMPUTATIONAL RESULTS

To evaluate the performance of the proposed method, this study employs the standard 15 LN instances (LN1 to LN15) introduced by Loh and Nee [48] as well as the 700 instances from Bischoff and Ratcliff [49]. These test instances vary in the number of box types and 3D bin packing dimensions, spanning from weakly to strongly heterogeneous problem characteristics. The proposed EHHITS algorithm was implemented in MATLAB® 2018 and executed on a laptop computer equipped with an Intel® Core i5 CPU (4 GHz) and 16 GB of RAM. The performance of the EHHITS algorithm for the test cases was governed by the following critical parameters as detailed in Table II:

A. Results for Test Cases LN01–LN15

For the performance evaluation, EHHITS is applied to a 3D bin packing data set. The Loh and Nee (LN) data set [48] is known for its heterogeneous nature, meaning it contains a wide variety of different box sizes and types. This makes the packing problem particularly difficult, as algorithms must be flexible enough to handle many different shapes and quantities of items, rather than just a few uniform sizes. The LN data set for the 3D container loading problem consists of 15 problems. This data set is notable for its heterogeneous nature, meaning each problem involves a variety of different box sizes and types, which makes it a challenging benchmark for packing algorithms.

The instances in the LN data set typically involve:

- A single container of a defined size.

TABLE III. VOLUME UTILIZATION (%) FOR TEST CLASSES LOH AND NEE (LN)

Problem	[48]	[50]	[51]	[49]	[40]	[2]	[41]	[3]	[52]	[53]	[54]	EHHITS (This work)
LN1	78.1	62.5	62.5	62.5	62.5	62.5	62.5	62.5	62.5	62.5	62.5	62.5
LN2	76.8	80.7	89.7	90.0	93.9	89.8	90.8	80.4	96.4	93.2	94.6	96.6
LN3	69.5	53.4	53.4	53.4	53.4	53.4	53.4	53.4	53.4	53.4	53.4	53.4
LN4	59.2	55.0	55.0	55.0	55.0	55.0	55.0	55.0	55.0	55.0	55.0	55.0
LN5	85.8	77.2	77.2	77.2	77.2	77.2	77.2	76.7	77.2	77.2	77.2	77.2
LN6	88.6	88.7	89.5	83.1	91.6	92.4	87.9	84.8	93.5	93.0	92.6	95.4
LN7	78.2	81.8	83.9	78.7	84.7	84.7	84.7	77.0	84.7	84.7	84.7	84.7
LN8	67.6	59.4	59.4	59.4	59.4	59.4	59.4	59.4	59.4	59.4	59.4	59.4
LN9	84.2	61.9	61.9	61.9	61.9	61.9	61.9	61.9	61.9	61.9	61.9	61.9
LN10	70.1	67.3	67.3	67.3	67.3	67.3	67.3	67.3	67.3	67.3	67.3	67.3
LN11	63.8	62.2	62.2	62.2	62.2	62.2	62.2	62.2	62.2	62.2	62.2	62.2
LN12	76.3	78.5	76.5	78.5	78.5	78.5	78.5	69.5	78.5	78.5	78.5	78.5
LN13	77.0	84.1	82.3	78.1	85.1	85.6	85.6	73.3	84.9	85.6	85.6	85.6
LN14	69.1	62.8	62.8	62.8	62.8	62.8	62.8	62.8	62.8	62.8	62.8	62.8
LN15	65.6	59.6	59.5	59.5	59.5	59.5	59.5	59.5	59.5	59.5	59.5	59.5
Average	74.2	69.0	69.5	68.6	70.3	69.9	69.9	67.05	70.6	70.4	70.5	70.8
Number of optimal problems	11	11	10	11	12	13	13	11	12	13	13	13

TABLE IV. VOLUME UTILIZATION (%) FOR TEST CLASSES BR1–BR7

Problem	Maximum	Minimum	Average	SD
BR1(3)	95.58	86.42	91.24	2.14
BR2(5)	95.36	86.57	91.86	1.82
BR3(8)	94.10	87.10	92.14	1.54
BR4(10)	93.42	86.50	92.72	1.43
BR5(12)	93.44	87.30	91.57	1.56
BR6(15)	93.23	88.46	91.42	1.39
BR7(20)	92.40	86.23	91.12	1.42

TABLE V. AVERAGE VOLUME UTILIZATION (%) FOR TEST CLASSES BR1–BR7

Approach	Volume utilization (%)
[40]	88.51
[41]	88.75
[51]	84.94
[52]	91.81
[54]	84.56
[49]	83.45
[55]	88.30
[56]	91.26
[57]	92.86
[58]	89.73
[59]	93.82
[60]	84.19
EHHITS (this work)	91.72

- Multiple types of rectangular boxes, with each type having different dimensions (length, width, height) and a specified quantity to be packed.
- The objective is to optimize the placement of boxes in the container for highest volume efficiency.

The constraints of the 3D bin packing are as follows:

- The items consist of rectangular boxes characterized by their known width, depth, and height.
- The boxes can be reoriented if permitted by problem constraints.
- Each box is allowed to be stacked over another.

To preserve structural orientation in these particular situations, rotations along the horizontal axes were restricted. The dataset used is made up of 100-250 instances that are divided into 6-10 different kinds. The EHHITS method was executed for a fixed duration of 10 minutes for each instance. To ensure a reliable evaluation and account for the stochastic nature of the metaheuristic, each instance was independently solved in 30 runs. This repeated evaluation helps reduce the effect of random initialization and provides a more robust assessment of the algorithm's performance.

Table III presents the experimental results for the 15 Loh/Nee test cases together with those found in the literature. The obtained volume utilization is provided for every method except L/N. According to Bischoff and Ratcliff [49], the packing density expressed in L/N is an overestimation of the volume utilization. Lastly, for each strategy, the average volume utilization and the number of ideal solutions for every box packed are provided. Our algorithm produced the highest average volume utilization for the 15 test scenarios out of all the methods. Fig. 8 presents samples of the packing results obtained by EHHITS on the LN benchmark dataset (LN2, LN6, LN12 and LN13).

The experimental results demonstrate that the EHHITS algorithm is a highly effective metaheuristic for solving the 3D packing problem. The algorithm successfully packed all available items in 13 out of 15 benchmark instances. For the remaining two instances, EHHITS achieved volume utilization

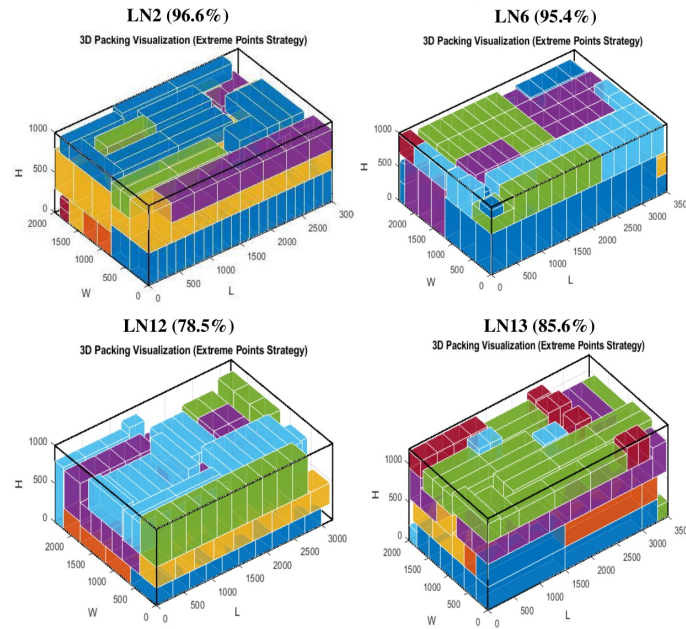


Fig. 8. LN dataset packing results.

rates of 96.6% and 95.4%, respectively, yielding an average utilization rate of 70.8% across the evaluated benchmark instances.

B. Results for Test Classes BR1–BR7

The 700 Bischoff/Ratcliff test cases are organized into seven test classes (BR1–BR7), with each class consisting of 100 test cases. For all test scenarios, the container dimensions are $L = 587$, $W = 233$, and $H = 220$. The number of box kinds for BR1 through BR7 is 3, 5, 8, 10, 12, 15, and 20. Table IV summarizes the experimental results for the 700 Bischoff/Ratcliff test cases. For every test class, the minimum, average, and maximum volume utilizations of 100 test instances are displayed together with the instances' standard deviations. The EHHITS algorithm is executed for ten minutes and 30 times with various random numbers to solve each problem. The total average volume use for the 700 test instances is shown in Table V and contrasted with the utilizations attained using alternative techniques. Other methods' volume utilizations are derived from the literature and updated with the latest findings. The results of EHHITS indicate that for all test classes, the average volume utilization is 91.72%.

Several factors intrinsic to the EHHITS design contribute to this robust performance. Firstly, the hybridization strategy successfully balances global exploration and local intensification. The multi-strategy initialization generates a diverse and high-quality initial population, allowing the HHO component to effectively search for optimal regions in the solution space. Subsequently, the Intensive Tabu Search acts as a powerful intensifier, refining these solutions by navigating complex local optima and escaping from suboptimal regions through its diverse neighborhood operators. This synergistic interaction is likely the primary driver behind the algorithm's consistency

and reliability, as reflected in its high success rate across most test instances.

VI. CONCLUSION

This research has introduced and empirically validated a novel hybrid metaheuristic for addressing the 3D container packing problem, combining Enhanced Harris Hawks Optimization with Tabu Search (EHHITS). The proposed algorithm addresses fundamental challenges in complex combinatorial optimization by integrating global exploration capabilities with intensive local search techniques. The EHHITS framework integrates an Intensive Tabu Search with HHO to strategically balance local precision and global exploration to increase packing density. Comprehensive evaluation on the standard LN benchmark instances demonstrates the strong and competitive performance of EHHITS. The proposed algorithm achieved optimal solutions in 13 of the 15 test cases and attained an average volume utilization of 70.8%, with notable improvements observed on the LN2 and LN6 instances. Furthermore, evaluation on the complete set of 700 Bischoff/Ratcliff test cases demonstrates the robustness and scalability of EHHITS across packing instances with varying problem dimensions. For all test classes ranging from BR1 to BR7, the average volume utilization consistently remains above 91%. These findings, supported by the detailed metrics of minimum, maximum, and standard deviation values, suggest that the integration of Tabu Search into the Enhanced Harris Hawks framework significantly strengthens global search capabilities. Ultimately, the EHHITS algorithm proves to be a reliable and high-performing strategic framework for achieving superior global optimization in diverse packing environments. The experimental outcomes confirm that the strategic hybridization of population-based and trajectory-based methods offers significant advantages for tackling NP-hard optimization problems. Future research directions include developing dynamic parameter adaptation

mechanisms, incorporating additional problem-specific constraints, and extending the approach to other combinatorial optimization domains.

ACKNOWLEDGMENT

This work was funded by the University of Jeddah, Jeddah, Saudi Arabia, under grant No. (UJ-23-DR-275). Therefore, the authors thank the University of Jeddah for its technical and financial support.

REFERENCES

- [1] G.Scheithauer, "Algorithms for the Container Loading Problem," in *Operations Research Proceedings 1991*, W.Gaul *et al.*, Eds. Berlin, Heidelberg: Springer, 1992, pp. 445–452, doi:10.1007/978-3-642-46773-8-112.
- [2] A.Bortfeldt and H.Gehring, "A hybrid genetic algorithm for the container loading problem," *European Journal of Operational Research*, vol. 131, no. 1, pp. 143–161, 2001, doi: 10.1016/S0377-2217(00)00055-2.
- [3] T.Dereli and G.S.Das, "A hybrid 'bee(s) algorithm' for solving container loading problems," *Applied Soft Computing*, vol. 11, no. 2, pp. 2854–2862, 2011, doi: 10.1016/j.asoc.2010.11.017.
- [4] S.K.ElSayed and E.E.Elattar, "Hybrid Harris hawks optimization with sequential quadratic programming for optimal coordination of directional overcurrent relays incorporating distributed generation," *Alexandria Engineering Journal*, vol. 60, no. 2, pp. 2421–2433, 2021, doi: 10.1016/j.aej.2020.12.028.
- [5] Z. Yu, X. Shi, J. Zhou, X. Chen, and X. Qiu, "Effective Assessment of Blast-Induced Ground Vibration Using an Optimized Random Forest Model Based on a Harris Hawks Optimization Algorithm," *Applied Sciences*, vol. 10, no. 4, p. 1403, 2020, doi: 10.3390/app10041403.
- [6] M.A.Al-Betar, M.A.Awadallah, A.A.Heidari, H.Chen *et al.*, "Survival exploration strategies for Harris Hawks Optimizer," *Expert Systems with Applications*, vol. 168, p. 114243, 2021, doi: 10.1016/j.eswa.2020.114243.
- [7] M.H.Qais, H.M.Hasanien, and S.Alghuwainem, "Parameters extraction of three-diode photovoltaic model using computation and Harris Hawks optimization," *Energy*, vol. 195, p. 117040, 2020, doi: 10.1016/j.energy.2020.117040.
- [8] S.Jiao, C.Wang, R.Gao, Y.Li, and Q.Zhang, "Harris Hawks Optimization with Multi-Strategy Search and Application," *Symmetry*, vol. 13, no. 12, p. 2364, 2021, doi: 10.3390/sym13122364.
- [9] E. Rodríguez-Esparza *et al.*, "An efficient Harris hawks-inspired image segmentation method," *Expert Systems with Applications*, vol. 155, p. 113428, 2020, doi: 10.1016/j.eswa.2020.113428.
- [10] W.Fu, K.Shao, J.Tan, and K.Wang, "Fault Diagnosis for Rolling Bearings Based on Composite Multiscale Fine-Sorted Dispersion Entropy and SVM With Hybrid Mutation SCA-HHO Algorithm Optimization," *IEEE Access*, vol. 8, pp. 13086–13104, 2020, doi: 10.1109/ACCESS.2020.2966582.
- [11] H.Moayedi, D.Bui, and P.ThiNgo, "Neural Computing Improvement Using Four Metaheuristic Optimizers in Bearing Capacity Analysis of Footings Settled on Two-Layer Soils," *Applied Sciences*, vol. 9, no. 23, p. 5264, 2019, doi: 10.3390/app9235264.
- [12] M.Abdel-Basset, W.Ding, and D.El-Shahat, "A hybrid Harris Hawks optimization algorithm with simulated annealing for feature selection," *Artificial Intelligence Review*, vol. 54, no. 1, pp. 593–637, 2021, doi: 10.1007/s10462-020-09860-3.
- [13] A.Abbasi, B.Firouzi, and P.Sendur, "On the application of Harris hawks optimization (HHO) algorithm to the design of microchannel heat sinks," *Engineering with Computers*, vol. 37, no. 2, pp. 1409–1428, 2021, doi: 10.1007/s00366-019-00892-0.
- [14] K.Shao, W.Fu, J.Tan, and K.Wang, "Coordinated approach fusing time-shift multiscale dispersion entropy and vibrational Harris hawks optimization-based SVM for fault diagnosis of rolling bearing," *Measurement*, vol. 173, p. 108580, 2021, doi: 10.1016/j.measurement.2020.108580.
- [15] N.Kaur, L.Kaur, and S.S.Cheema, "An enhanced version of Harris Hawks Optimization by dimension learning-based hunting for Breast Cancer Detection," *Scientific Reports*, vol. 11, no. 1, p. 21933, 2021, doi: 10.1038/s41598-021-01018-7.
- [16] M.Naeijian, A.Rahimnejad, S.M.Ebrahimi, N.Pourmousa, and S.A.Gadsden, "Parameter estimation of PV solar cells and modules using Whippy Harris Hawks Optimization Algorithm," *Energy Reports*, vol. 7, pp. 4047–4063, 2021, doi: 10.1016/j.egyr.2021.06.085.
- [17] L.K.Foong and H.Moayedi, "Slope stability evaluation using neural network optimized by equilibrium optimization and vortex search algorithm," *Engineering with Computers*, vol. 38, no. S2, pp. 1269–1283, 2022, doi: 10.1007/s00366-021-01282-1.
- [18] İ. Çetinbaş, B. Tamyürek, and M. Demirtaş, "The Hybrid Harris Hawks Optimizer-Arithmetic Optimization Algorithm: A New Hybrid Algorithm for Sizing Optimization and Design of Microgrids," *IEEE Access*, vol. 10, pp. 19254–19283, 2022, doi: 10.1109/ACCESS.2022.3151119.
- [19] A. G. Hussien and M. Amin, "A self-adaptive Harris Hawks optimization algorithm with opposition-based learning and chaotic local search strategy for global optimization and feature selection," *International Journal of Machine Learning and Cybernetics*, vol. 13, no. 2, pp. 309–336, 2022, doi: 10.1007/s13042-021-01326-4.
- [20] H. Zhang *et al.*, "A generalized artificial intelligence model for estimating the friction angle of clays in evaluating slope stability using a deep neural network and Harris Hawks optimization algorithm," *Engineering with Computers*, vol. 38, no. S5, pp. 3901–3914, 2022, doi: 10.1007/s00366-020-01272-9.
- [21] R. M. Rabie *et al.*, "An Improved Harris Hawks Optimizer (IHHO) with Adaptive Hill Climbing and Elite Mutation for Global Optimization," *Results in Control and Optimization*, 2026, doi: 10.1016/j.rico.2026.100815.
- [22] R.M.Rabie, H.Zaher, N.R.Saied, and H.Sayed, "A Modified Harris Hawks Algorithm to Solving Optimization Problems," *International Journal of Industrial Engineering & Production Research*, vol. 35, no. 1, 2024, doi: 10.22068/ijiepr.35.1.1853.
- [23] A. Dabba, A. Tari, and S. Meftali, "A new multi-objective binary Harris Hawks optimization for gene selection in microarray data," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 4, pp. 3157–3176, 2023, doi: 10.1007/s12652-021-03441-0.
- [24] H. M. Balaha, A. E.-S. Hassan, E. M. El-Gendy, H. ZainEldin, and M. M. Saafan, "An aseptic approach towards skin lesion localization and grading using deep learning and harris hawks optimization," *Multimedia Tools and Applications*, vol. 83, no. 7, pp. 19787–19815, 2023, doi: 10.1007/s11042-023-16201-3.
- [25] D. Dhawale, V. K. Kamboj, and P. Anand, "An improved Chaotic Harris Hawks Optimizer for solving numerical and engineering optimization problems," *Engineering with Computers*, vol. 39, no. 2, pp. 1183–1228, 2023, doi: 10.1007/s00366-021-01487-4.
- [26] S. Almotairi, E. Badr, M. Abdul Salam, and A. Dawood, "Three Chaotic Strategies for Enhancing the Self-Adaptive Harris Hawk Optimization Algorithm for Global Optimization," *Mathematics*, vol. 11, no. 19, p. 4181, 2023, doi: 10.3390/math11194181.
- [27] C.Li, Q.Si, J.Zhao, and P.Qin, "A robot path planning method using improved Harris Hawks optimization algorithm," *Measurement and Control*, vol. 57, no. 4, pp. 469–482, 2024, doi: 10.1177/00202940231204424.
- [28] Q.V. Vu, V.T. Pham, D.N. Le, Z. Kong, G. Papazafeiropoulos *et al.*, "Hybrid machine learning with HHO method for estimating ultimate shear strength of both rectangular and circular RC columns," *Steel and Composite Structures*, vol. 52, no. 2, pp. 145–163, 2024, doi: 10.12989/SCS.2024.52.2.145.
- [29] F. Wang, "Enhanced Task Scheduling Algorithm Using Harris Hawks Optimization Algorithm for Cloud Computing," *International Journal of Advanced Computer Science and Applications*, vol. 16, no. 1, 2025, doi: 10.14569/IJACSA.2025.0160189.
- [30] M. reza Behnamfar, H. Barati, M. Abasi, M. Alaleh, and G. A. Torabi, "Multi-Objective Optimization for Day-Ahead HT-WP-PV-PSH with LS-EVs Systems Self-Scheduling Unit Commitment Using HHO-PSO Algorithm," *Journal of Operation and Automation in Power Engineering*, Online First, 2025, doi: 10.22098/joape.2024.12419.1934.
- [31] R. Ghadami and J. Rahebi, "Alzheimer's Prediction Methods with Harris Hawks Optimization (HHO) and Deep Learning-Based Approach Using an MLP-LSTM Hybrid Network," *Diagnostics*, vol. 15, no. 3, p. 377, 2025, doi: 10.3390/diagnostics15030377.
- [32] S. Al-Adwan *et al.*, "A multi-strategy framework for enhancing Harris hawks optimization for global optimization problems," *Scientific Reports*, 2026.

- [33] M. Kumar, N. Mohd, G. Shivam, A. Goyal, D. Parashar *et al.*, "Hybrid Aquila optimizer–Harris Hawks optimization for CNN hyperparameter tuning in brain tumor classification," *Scientific Reports*, 2026.
- [34] A. Aliano Filho, H. De Oliveira Florentino, M. V. Pato, S. C. Poltroniere, and J. F. Da Silva Costa, "Exact and heuristic methods to solve a bi-objective problem of sustainable cultivation," *Annals of Operations Research*, vol. 314, no. 2, pp. 347–376, 2022, doi:10.1007/s10479-019-03468-9.
- [35] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Computing Surveys*, vol. 35, no. 3, pp. 268–308, 2003, doi: 10.1145/937503.937505.
- [36] J. F. Gonçalves and M.G.C. Resende, "A parallel multi-population biased random-key genetic algorithm for a container loading problem," *Computers & Operations Research*, vol. 39, no. 2, pp. 179–190, 2012, doi: 10.1016/j.cor.2011.03.009.
- [37] R. Zeineldin and A. Morsy, "A Modified Artificial Bee Colony for Solving the Container Loading Problem," *International Journal of Computer Applications*, vol. 114, no. 3, pp. 19–24, 2015, doi:10.5120/19958-1786.
- [38] T. Fanslau and A. Bortfeldt, "A Tree Search Algorithm for Solving the Container Loading Problem," *INFORMS Journal on Computing*, vol. 22, no. 2, pp. 222–235, 2010, doi: 10.1287/ijoc.1090.0338.
- [39] E. E. Bischoff and M. S. W. Ratcliff, "Issues in the development of approaches to container loading," *Omega*, vol. 23, no. 4, pp. 377–390, Aug. 1995, doi: 10.1016/0305-0483(95)00015-G.
- [40] J. Terno, G. Scheithauer, U. Sommerweiß, and J. Riehme, "An efficient approach for the multi-pallet loading problem," *European Journal of Operational Research*, vol. 123, no. 2, pp. 372–381, 2000, doi: 10.1016/S0377-2217(99)00263-5.
- [41] M. Eley, "Solving container loading problems by block arrangement," *European Journal of Operational Research*, vol. 141, no. 2, pp. 393–409, 2002, doi: 10.1016/S0377-2217(02)00133-9.
- [42] R. Morabito, "An AND/OR-graph approach to the container loading problem," *Int. Trans. Oper. Res.*, vol. 1, no. 1, pp. 59–73, Jan. 1994, doi: 10.1016/0969-6016(94)90046-9.
- [43] D. T. Akl, M. M. Saafan, A. Y. Haikal, and E. M. El-Gendy, "IHHO: an improved Harris Hawks optimization algorithm for solving engineering problems," *Neural Computing and Applications*, vol. 36, no. 20, pp. 12185–12298, 2024, doi:10.1007/s00521-024-09603-3.
- [44] M. Arenales and R. Morabito, "An AND/OR-graph approach to the solution of two-dimensional non-guillotine cutting problems," *Eur. J. Oper. Res.*, vol. 84, no. 3, pp. 599–617, Aug. 1995, doi: 10.1016/0377-2217(95)00026-M.
- [45] S. Liu, W. Tan, Z. Xu, and X. Liu, "A tree search algorithm for the container loading problem," *Comput. Ind. Eng.*, vol. 75, pp. 20–30, Sep. 2014, doi: 10.1016/j.cie.2014.05.024.
- [46] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja *et al.*, "Harris hawks optimization: Algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019, doi: 10.1016/j.future.2019.02.028.
- [47] H. M. Alabool, D. Alarabiat, L. Abualigah, and A. A. Heidari, "Harris hawks optimization: a comprehensive review of recent variants and applications," *Neural Comput. Appl.*, vol. 33, no. 15, pp. 8939–8980, Aug. 2021, doi: 10.1007/s00521-021-05720-5.
- [48] T. Loh and A. Nee, "A packing algorithm for hexahedral boxes," in *Proc. Conf. Ind. Automation*, Singapore, 1992, pp. 115–126.
- [49] E. E. Bischoff and M. S. W. Ratcliff, "Loading Multiple Pallets," *Journal of the Operational Research Society*, vol. 46, no. 11, pp. 1322–1336, 1995, doi:10.1057/jors.1995.181.
- [50] B. K. A. Ngoi, M. L. Tay, and E. S. Chua, "Applying spatial representation techniques to the container packing problem," *International Journal of Production Research*, vol. 32, no. 1, pp. 111–123, 1994, doi:10.1080/00207549408956919.
- [51] E. E. Bischoff, F. Janetz, and M. S. W. Ratcliff, "Loading pallets with non-identical items," *European Journal of Operational Research*, vol. 84, no. 3, pp. 681–692, 1995, doi:10.1016/0377-2217(95)00031-K.
- [52] A. Lim, H. Ma, J. Xu, and X. Zhang, "An iterated construction approach with dynamic prioritization for solving the container loading problems," *Expert Systems with Applications*, vol. 39, no. 4, pp. 4292–4305, 2012, doi:10.1016/j.eswa.2011.09.103.
- [53] T. A. M. Toffolo, E. Esprit, T. Wauters, and G. Vanden Berghe, "A two-dimensional heuristic decomposition approach to a three-dimensional multiple container loading problem," *European Journal of Operational Research*, vol. 257, no. 2, pp. 526–538, 2017, doi:10.1016/j.ejor.2016.07.033.
- [54] Ö. Şafak and G. Erdoğan, "A Large Neighbourhood Search Algorithm for Solving Container Loading Problems," *Computers & Operations Research*, vol. 154, p. 106199, 2023, doi:10.1016/j.cor.2023.106199.
- [55] H. Gehring and A. Bortfeldt, "A Genetic Algorithm for Solving the Container Loading Problem," *International Transactions in Operational Research*, vol. 4, no. 5–6, pp. 401–418, 1997, doi:10.1111/j.1475-3995.1997.tb00095.x.
- [56] A. Bortfeldt and H. Gehring, "Applying Tabu Search to Container Loading Problems," in *Operations Research Proceedings 1997*, P. Kischka *et al.*, Eds. Berlin, Heidelberg: Springer, 1998, pp. 533–538, doi:10.1007/978-3-642-58891-4-84.
- [57] D. Mack, A. Bortfeldt, and H. Gehring, "A parallel hybrid local search algorithm for the container loading problem," *Int. Trans. Oper. Res.*, vol. 11, no. 5, pp. 511–533, Sep. 2004, doi: 10.1111/j.1475-3995.2004.00474.x.
- [58] A. Moura and J. F. Oliveira, "A GRASP Approach to the Container-Loading Problem," *IEEE Intell. Syst.*, vol. 20, no. 4, pp. 50–57, Jul. 2005, doi: 10.1109/MIS.2005.57.
- [59] F. Parreño, R. Alvarez-Valdes, J. M. Tamarit, and J. F. Oliveira, "A Maximal-Space Algorithm for the Container Loading Problem," *Inf. J. Comput.*, vol. 20, no. 3, pp. 412–422, Aug. 2008, doi: 10.1287/ijoc.1070.0254.
- [60] A. P. Davies and E. E. Bischoff, "Weight distribution considerations in container loading," *Eur. J. Oper. Res.*, vol. 114, no. 3, pp. 509–527, May 1999, doi: 10.1016/S0377-2217(98)00139-8.