

Sustainable Supply Chain Optimization: A Breakthrough in Swarm-based Artificial Intelligence

Khaoula Khlie

Department of Industrial Management, Faculty of Business, Liwa College, Abu Dhabi, UAE
khaoula.khlie@lc.ac.ae

Aruna Pugalenth

Department of Industrial Management, Faculty of Business, Liwa College, Abu Dhabi, UAE
aruna.pugalenth@lc.ac.ae

Zoubida Benmamoun

Department of Industrial and Mechanical Engineering, Faculty of Engineering, Liwa College, Abu Dhabi, UAE
zoubida.benmamoun@lc.ac.ae (corresponding author)

Widi Aribowo

Department of Electrical Engineering, Faculty of Vocational Studies, Universitas Negeri Surabaya, Surabaya, East Java, Indonesia
widiaribowo@unesa.ac.id

Mohammad Dehghani

Department of Electrical and Electronics Engineering, Shiraz University of Technology, Iran
m.dehghani@sutech.ac.ir

Received: 9 February 2025 | Revised: 24 February 2025 and 7 March 2025 | Accepted: 13 March 2025

Licensed under a CC-BY 4.0 license | Copyright (c) by the authors | DOI: <https://doi.org/10.48084/etasr.10505>

ABSTRACT

Traditional Supply Chain Management (SCM) often faces challenges such as high risks due to the lack of accountability and transparency. Optimization algorithms are essential to improve green manufacturing operations. This study introduces the Dhole Optimization Algorithm (DOA) to solve a green lot size optimization problem. DOA is mathematically modeled in two phases: (i) exploration based on simulating the attack of dholes toward prey and (ii) exploitation based on simulating the chase process between dholes and prey. Real-world data are transformed into an optimization function aimed at minimizing costs and CO₂ emissions. The DOA model was applied to determine the optimal lot size, balancing cost and sustainability. Comparative experiments with 12 established metaheuristic algorithms showed DOA's superior performance. The proposed algorithm was implemented in 10 case studies, and the results show that it performed more successfully than competing algorithms in all of them. This study demonstrates that not only does DOA optimize better but also reduces environmental impact, offering a promising solution for green manufacturing and sustainable SCM. This is a novel approach to lot-size optimization and highlights DOA's potential for future research and applications in the field.

Keywords-sustainability; supply chain management; SDGs; optimization; generative AI; bio-inspired; metaheuristic; dhole; exploration; exploitation

I. INTRODUCTION

Optimization challenges pose significant hurdles in the field of Supply Chain Management (SCM) [1]. Artificial Intelligence (AI) is a powerful tool to address these problems,

allowing optimal decisions such as selecting the most appropriate warehouse location [2], designing an efficient production layout and resource allocation strategy [3], and determining the best routes to deliver goods from factories to

customers [2]. These problems are inherently complex because they often have multiple feasible solutions. Optimization is the process of identifying the most suitable solution [4]. An optimization problem is constructed using a mathematical model with three essential components: (i) decision variables that represent the parameters to be determined, (ii) constraints that specify the limitations or conditions that must be satisfied, and (iii) an objective function that quantifies the goal to be maximized or minimized. Optimization aims at calculating the values of the decision variables to optimize the objective function while adhering to all constraints [5], requiring suitable methods for their resolution, which are classified into two main categories: deterministic and stochastic approaches [6].

Deterministic methods, including gradient-based and non-gradient-based techniques, are well-suited for solving problems that are linear, convex, continuous, differentiable, and have low dimensionality [7]. However, as optimization problems increase in complexity, especially in terms of dimensionality, deterministic methods often fail because they tend to become trapped in local optima and lose their efficiency [8].

The limitations and inefficiencies of deterministic methods in addressing real-world optimization challenges have driven the development of stochastic approaches, which rely on probabilistic mechanisms to explore and exploit the search space [9]. Among these, metaheuristic algorithms stand out as particularly effective tools [10] because they combine simplicity of implementation with the ability to tackle diverse optimization problems, including those that are nonlinear, non-convex, non-differentiable, and NP-hard. Furthermore, they perform well in problem-solving spaces that are discrete, high-dimensional, or unknown [11]. Despite their strengths, metaheuristic algorithms are not without limitations, as they are based on random search and trial-and-error processes that cannot guarantee convergence to the global optimum. Instead, they provide solutions that are considered quasi-optimal, meaning that they are close to the global optimum. The quest to achieve even better quasi-optimal solutions continues to motivate researchers to design and improve metaheuristic algorithms [12]. Recently published metaheuristic algorithms that can be used in various optimization applications are Revolution Optimization Algorithm (ROA) [13], Paper Publishing Based Optimization (PPBO) [14], Dynamic Virtual Bats Algorithm (DVBA) [15], Passing Vehicle Search (PVS) [16], Great Wall Construction Algorithm (GWCA) [17], Fire Hawk Optimizer (FHO) [18], Ali Baba and the Forty Thieves (AFT) [19], Builder Optimization Algorithm (BOA) [20], Makeup Artist Optimization Algorithm (MAOA) [21], Geometric Octal Zones Distance Estimation (GOZDE) [22], Dream Optimization Algorithm (DOA) [23], and Atomic Orbital Search (AOS) [24].

Metaheuristic algorithms must balance global search (exploration) and local search (exploitation). Exploration involves scanning the entire solution space to identify promising regions, thus preventing the algorithm from becoming stuck in local optima. Exploitation, focuses on a thorough investigation of these regions to refine solutions and move closer to the global optimum. The ability to achieve an

effective balance between these two processes is a critical factor in the success of any metaheuristic algorithm [25].

According to the No Free Lunch (NFL) theorem, no single metaheuristic algorithm can perform optimally across all types of optimization problems [26]. In other words, the success of an algorithm on one set of problems does not ensure its effectiveness on others. This inherent variability underscores the need for ongoing research into the design of innovative metaheuristic algorithms to address diverse optimization challenges. In this context, this study introduces a completely different and novel bio-inspired metaheuristic algorithm, called the Dhole Optimization Algorithm (DOA). The key contributions of this work are as follows:

- DOA draws its inspiration from the unique and cooperative hunting behaviors of dholes, wild canids known for their highly organized strategies in the wild.
- The algorithm's foundation is based on mimicking two distinct hunting phases: attacking and chasing prey.
- The implementation of DOA is systematically modeled in two phases: (i) exploration, simulating the positional dynamics of dholes during the attack, and (ii) exploitation, capturing the positional adjustments during the chase.
- The efficacy of DOA to address SCM is challenged for sustainable lot size optimization.
- Comparative analyses highlight DOA's superior performance against twelve well-established metaheuristic algorithms.

DOA introduces a novel approach to optimization problems, incorporating unique characteristics that make it particularly effective for the batch-size optimization problem. Compared to other metaheuristic algorithms, DOA offers the following advantages.

A. *Balanced Exploration and Exploitation*

One of the key challenges in optimization algorithms is maintaining a balance between exploration (global search) and exploitation (local search). DOA utilizes adaptive mechanisms to dynamically adjust this balance, leading to enhanced solution quality and reduced risk of premature convergence. This ability is particularly beneficial for batch size optimization, where finding the right balance between computational efficiency and model accuracy is crucial.

B. *Inspired by Coordinated Group Behavior and Collaborative Decision-Making*

DOA draws inspiration from collective intelligence and cooperative behavior, enhancing the efficiency of population updates and accelerating the search process. Compared to traditional methods such as Genetic Algorithm (GA) or Particle Swarm Optimization (PSO), DOA leverages more sophisticated collaborative strategies to guide the search toward optimal solutions.

C. *Higher Adaptability to Various Objective Functions*

Unlike classical approaches such as Differential Evolution (DE), DOA features a more flexible mechanism for

dynamically adjusting internal parameters based on the characteristics of the objective function. This adaptability makes it particularly suitable for problems like batch size optimization, where gradient smoothness and stability during learning are critical factors.

D. Reduced Computational Complexity Compared to Evolutionary Methods

Unlike some evolutionary algorithms that require extensive computational resources, DOA achieves faster convergence while maintaining solution accuracy through a more efficient search process. This advantage is especially useful in machine learning applications and batch size optimization, where computational speed is a priority.

E. Key Differences from Other Metaheuristic Approaches

DOA distinguishes itself from traditional algorithms such as Ant Colony Optimization (ACO) or Simulated Annealing (SA) by integrating a hybrid model that incorporates both optimization and adaptive learning mechanisms. This hybrid approach improves its effectiveness in determining the optimal batch size for deep learning and related fields.

DOA's innovative mechanisms, particularly in batch size selection for machine learning, contribute to enhanced accuracy, reduced learning fluctuations, and improved computational efficiency. These features allow the proposed algorithm to outperform existing methods in terms of convergence speed, accuracy, and overall optimization performance.

II. DHOLE OPTIMIZATION ALGORITHM (DOA)

A. Initialization

DOA is a novel swarm-based optimization technique that utilizes the collective search capability of its members to solve optimization problems iteratively. Each member, represented by a dhole, seeks an optimal solution by exploring the problem-solving space, where the position of each dhole corresponds to a potential solution. Mathematically, these positions are modeled as vectors. The DOA population consists of multiple dholes, whose positions are described by a matrix (1). Initially, dhole positions are randomly assigned using (2), ensuring a diverse population and facilitating efficient exploration of the solution space. This randomness enhances the swarm's ability to search different regions of the space effectively. In this context, the equation governing the population matrix X is:

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N \times m} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,d} & \cdots & x_{1,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & \cdots & x_{i,d} & \cdots & x_{i,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,d} & \cdots & x_{N,m} \end{bmatrix}_{N \times m} \quad (1)$$

$$x_{i,d} = lb_d + r \cdot (ub_d - lb_d) \quad (2)$$

where X_i is the i -th dhole (candidate solution), $x_{i,d}$ is its d -th dimension in search space (decision variable), N is the number of dholes, m is the number of decision variables, r is a random number in the interval $[0,1]$, and lb_d , and ub_d are the lower and upper bounds of the d -th decision variable, respectively.

Once the initial positions are established, the objective function can be evaluated for each candidate solution. The corresponding set of objective function values can be represented as a vector, as shown in (3), which allows for the assessment of the fitness of each solution in the population:

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N \times 1} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix}_{N \times 1} \quad (3)$$

Here, F represents the vector of evaluated objective function values and each F_i corresponds to the objective function value for the i -th dhole.

In the DOA design, the positions of the DOA members are updated in two separate phases in each iteration.

B. Phase 1: Attacking and Moving Toward the Prey (Exploration Phase)

In the first phase of DOA, the population members are updated based on the behavior of dholes as they move toward the prey during the initial attack. Dholes are known for their social behavior and hunt in groups, using their acute vision to locate their prey. Upon detecting the prey's position, they begin their attack. The movement of dholes in nature toward the prey is characterized by rapid and significant changes in position, enhancing the algorithm's ability to effectively explore the problem-solving space. This leads to an exploration strategy that encourages the algorithm to search globally, helping to avoid premature convergence to suboptimal solutions.

To determine the prey's position for the dholes, the algorithm combines three critical elements: the best member of the population, the worst member, and the population's mean position. The best and worst members are identified by evaluating the objective function, with the best corresponding to the lowest (or highest) value, depending on the problem. The population mean is calculated by averaging the positions of all members, as described in (4). The prey's position is then determined using (5), which incorporates the best, worst, and mean positions of the population, introducing a random factor to diversify the exploration process.

$$X_{mean} = \frac{\sum_{i=1}^N X_i}{N} \quad (4)$$

$$P = X_{best} + r \cdot (X_{mean} - X_{worst}) \quad (5)$$

where X_{mean} is the population mean, N is the number of dholes population, P is the prey position, X_{best} and X_{worst} are the best and worst population members, and r is a random number from the interval $[0, 1]$.

Once the prey's position is identified, the dholes move toward it. In DOA, the position update for each member is calculated based on the model of dholes' movement toward the prey, as shown in (6). If this new position leads to a better objective function value, it replaces the previous position, as described in (7). This update mechanism allows the population to move dynamically toward promising regions of the search space.

$$x_{i,j}^{P1} = x_{i,j} + r \cdot (P_j - I \cdot x_{i,j}) \quad (6)$$

$$X_i = \begin{cases} X_i^{P1}, & F_i^{P1} \leq F_i, \\ X_i, & \text{else,} \end{cases} \quad (7)$$

where P is the selected prey by dholes, P_j is its j -th dimension, X_i^{P1} is the new position calculated for the i -th dhole based on the exploration phase of the proposed DOA, $x_{i,j}^{P1}$ is its j -th dimension, F_i^{P1} is its objective function value, r is a random number in the interval $[0,1]$, and I is a number that is randomly selected as 1 or 2.

C. Phase 2: Chasing to Catch Prey (Exploitation Phase)

The second phase of DOA focuses on fine-tuning the positions of the population members in the problem-solving space based on the behavior of dholes as they chase and capture their prey. Unlike the initial phase, where large movements are made towards the prey, the chase phase involves smaller, more focused movements. Dholes are capable of following their prey for long periods, often in a confined area, with rapid and precise adjustments to their position. This behavior is modeled in DOA to enhance the algorithm's exploitation ability, focusing the search on promising regions of the solution space and refining the candidate solutions.

In this phase, the position of each member is updated using (8), which incorporates a random factor and adjusts the position within the bounds of the problem space. The factor $(1 - 2r)$ ensures that the movement is smaller compared to the exploration phase, promoting fine-tuning. If this new position leads to an improvement in the objective function, it replaces the previous position, as described in (9). This phase is crucial for optimizing solutions once promising areas of the search space have been identified.

$$x_{i,j}^{P2} = x_{i,j} + (1 - 2r) \cdot \frac{ub_j - lb_j}{t} \quad (8)$$

$$X_i = \begin{cases} X_i^{P2}, & F_i^{P2} \leq F_i \\ X_i, & \text{else} \end{cases} \quad (9)$$

where X_i^{P2} is the new position calculated for the i -th dhole based on the chasing phase of the proposed DOA, $x_{i,j}^{P2}$ is its j -th dimension, F_i^{P2} is its objective function value, r is a random number in the interval $[0,1]$, and t is the iteration counter. The DOA implementation process is presented in Algorithm 1.

Algorithm 1: Pseudocode of DOA.

Start DOA.

1. Input problem information: variables, objective function, and constraints.
2. Set DOA population size (N) and iterations (T).
3. Generate the initial population matrix at random using (2).
4. Evaluate the objective function.
5. For $t=1$ to T
6. For $i=1$ to N
7. Phase 1: Attacking and moving toward the prey (exploration phase)

8. Calculate the mean of the population using (4).
 9. Determine the prey location using (5).
 10. Calculate the new position of the i -th DOA member using (6).
 11. Update the i -th DOA member using (7).
 12. Phase 2: Chasing to catch prey (exploitation phase)
 13. Calculate the new position of the i -th DOA member using (8).
 14. Update the i -th DOA member using (9).
 15. End for.
 16. Save the best candidate solution so far.
 17. End for.
 18. Output the best quasi-optimal solution obtained.
- End DOA.

D. Computational Complexity of DOA

The computational complexity of the proposed DOA approach was evaluated. Preparing DOA for implementation on an optimization problem has a computational complexity equal to $O(Nm)$, where N is the number of dhole members and m is the number of decision variables of the problem. In DOA, the position of the population members is updated in each iteration based on the modeling of the hunting strategy of dholes in two phases: attack and chase. Thus, the population update process in DOA has a computational complexity equal to $O(2TNm)$, where T is the maximum number of iterations. According to this, the total computational complexity of the proposed DOA approach is equal to $O(Nm(1 + 2T))$.

III. DOA FOR SUSTAINABLE LOT SIZE OPTIMIZATION

The capability and efficiency of DOA in addressing complex optimization challenges within SCM were thoroughly analyzed and evaluated. To achieve this, DOA was utilized in the context of sustainable lot size optimization, providing a practical benchmark to assess its applicability and performance in real-world SCM scenarios.

Sustainable lot size optimization focuses on determining environmentally and socially responsible production batch sizes while balancing economic priorities within the supply chain. Unlike traditional lot-sizing models that primarily aim to minimize costs such as setup, inventory holding, and ordering expenses, sustainable lot-size optimization broadens the scope to incorporate environmental impacts, resource efficiency, and social responsibility. This approach considers factors such as energy consumption, raw material usage, waste generation, emissions, and social impacts along with economic objectives. The goal is to identify lot sizes that reduce costs while minimizing environmental harm and promoting social accountability across the supply chain. To achieve this, the

mathematical model for sustainable lot size optimization integrates both economic considerations and environmental impacts. The primary objective is to determine the optimal lot size at each supply chain stage, focusing on minimizing CO₂ emissions and overall costs. This model incorporates constraints such as production limits, inventory capacity, and demand satisfaction. Sustainability benchmarks, such as CO₂ emission caps linked to manufacturing, transportation, and warehousing, are also embedded in the model. The evaluation process adheres to a deterministic framework that prioritizes harmonizing economic efficiency with environmental stewardship.

The company aims to minimize inventory shortages, reduce surpluses, and identify the ideal lot size. When customer demand triggers inventory shortages, decisions are made to initiate production or place orders. The remaining inventory is classified as backlog, which requires careful monitoring to prevent surplus and strategize its reduction. The sustainable lot size optimization model is mathematically expressed as [27]:

$$TC = C_c \cdot \frac{D}{Q} + C_p \cdot P \cdot \frac{Q+SS}{2} + p \cdot A \cdot \frac{D}{Q} + C_E \cdot \frac{D}{Q} \quad (10)$$

In this equation, *TC* represents the total cost, serving as the objective function to be optimized. The components of this equation include *C_c*, representing the order cost per unit, *C_p*, denoting the holding cost per unit, *P*, indicating the price, *p*, representing the shortage cost per unit, *A*, representing the expected shortage per cycle, *D*, denoting the annual demand, *C_E*, representing the footprint emission cost, *Q*, indicating the quantity; and *SS*, representing the shortage. Each variable reflects different cost components in SCM, including ordering, holding, shortages, demand, emissions, and pricing. The model aims to balance these elements to achieve both economic and environmental goals.

DOA's performance in addressing the sustainable lot size optimization problem was evaluated on ten different case studies whose description is specified in Table I. The dataset used to simulate the DOA and obtain the results is a proprietary one developed specifically for this study. It is based on real-world parameters and industry-based assumptions related to sustainable lot sizing, logistics, and CO₂ emissions. The data include 10 distinct case studies reflecting various product types, demand patterns, transportation modes, and environmental impact factors. A brief and concise summary of each of the 10 case studies is presented in Table II.

The results obtained from DOA were compared with those of 12 other well-known competitor algorithms that represent a broad spectrum of metaheuristic approaches, including the Genetic Algorithm (GA) [28], Particle Swarm Optimization (PSO) [29], the Gravitational Search Algorithm (GSA) [30], Teaching–Learning–Based Optimization (TLBO) [31], Multi-Verse Optimizer (MVO) [32], Grey Wolf Optimizer (GWO) [33], the Whale Optimization Algorithm (WOA) [34], the Marine Predators Algorithm (MPA) [35], the Tunicate Swarm Algorithm (TSA) [36], the Reptile Search Algorithm (RSA) [37], the African Vultures Optimization Algorithm (AVOA) [38], and the White Shark Optimizer (WSO) [39]. It should also be noted that the penalty coefficient strategy was used to deal with the constraints of the problems.

Simulations were carried out on MATLAB R2022a using a 64-bit Core i7 processor with 3.20 GHz and 16 GB of main memory. DOA, as well as all 12 competitor algorithms, were implemented in MATLAB and tested on the same datasets (Table I). The performance results of the algorithms are reported using six statistical indicators: mean, best, worst, standard deviation (std), median, and rank. The obtained values for the mean were used as a criterion for ranking metaheuristic algorithms in handling each of the benchmark functions.

The comprehensive evaluations culminate in Table III, which meticulously documents the results of applying DOA and its alternatives to sustainable lot size optimization. These results highlight the superior optimization capabilities of DOA, as demonstrated by its exceptional performance in enhancing the objective function and delivering significantly improved Total Cost (TC) values. Additionally, the Wilcoxon rank-sum test [40] statistical analysis, the results of which are reported at the end of the table, indicates that DOA has a significant statistical advantage over the other algorithms.

TABLE I. CASE STUDIES

Case studies	<i>P</i>	<i>p</i>	<i>SS</i>	<i>A</i>	<i>C_E</i>	<i>C_p</i>	<i>C_c</i>
Part 1	38.364	30.691	184.147	1	417.456	0.12	200
Part 2	64.499	51.599	309.595	1	417.456	0.12	200
Part 3	1.729	1.383	8.299	1	15645.676	0.12	200
Part 4	4.239	3.391	20.347	1	15645.676	0.12	200
Part 5	1.046	0.837	5.021	1	417.456	0.12	200
Part 6	1.698	1.358	8.150	1	15645.676	0.12	200
Part 7	2.181	1.745	10.469	1	15645.676	0.12	200
Part 8	113.808	91.046	546.278	1	417.456	0.12	200
Part 9	73.917	59.134	354.802	1	417.456	0.12	200
Part 10	73.477	58.782	352.690	1	417.456	0.12	200

TABLE II. SUMMARY OF CASE STUDIES (PARTS 1 TO 10)

Part	Consumption Level	Unit	Country	Transportation	Fuel consumption and CO ₂	Lead time (Days)	Total cost (USD)
Part 1	High (220,500 units/period)	EA	USA	Scania V8, DSV Global	Low fuel, Low CO ₂ (0.012 kg)	65	1,015,112.62
Part 2	Medium (12,305 units/period)	EA	USA	Scania V8, DSV Global	Low fuel, Low CO ₂ (0.012 kg)	55	95,241.98
Part 3	Very High (1,889,663 ml/period)	ML	France	Scania V8, FRS FERRY	High fuel, High CO ₂ (0.372 kg)	61	392,067.83
Part 4	High (942,965 ml/period)	ML	France	Scania V8, FRS FERRY	High fuel, High CO ₂ (0.372 kg)	76	479,668.84
Part 5	Very High (8,137,249 ml/period)	ML	USA	Scania V8, DSV Global	Low fuel, Low CO ₂ (0.012 kg)	60	1,021,387.52
Part 6	Very High (8,240,394 ml/period)	ML	France	Scania V8, FRS FERRY	High fuel, High CO ₂ (0.372 kg)	71	1,679,062.77
Part 7	High (1,979,133 ml/period)	ML	France	Scania V8, FRS FERRY	High fuel, High CO ₂ (0.372 kg)	61	517,979.05
Part 8	Medium (9,200 units/period)	EA	USA	Scania V8, DSV Global	Low fuel, Low CO ₂ (0.012 kg)	55	125,647.72
Part 9	Very Low (625 units/period)	EA	USA	Scania V8, DSV Global	Low fuel, Low CO ₂ (0.012 kg)	65	5,607.99
Part 10	Medium (10,250 units/period)	EA	USA	Scania V8, DSV Global	Low fuel, Low CO ₂ (0.012 kg)	65	90,380.63

TABLE III. TOTAL COST (TC) VALUES FOR SUSTAINABLE LOT SIZE OPTIMIZATION

		DOA	PSO	TLBO	MVO	TSA	RSA	WSO	MPA	AVOA	WOA	GSA	GA	GWO
Part 1	mean	127117.8	127164.2	127164.2	127164.2	127164.2	127329.2	127329.2	127164.2	127329.2	127164.2	137363.5	128754.2	127164.2
	best	127117.8	127119.9	127119.9	127119.9	127119.9	127166	127166	127119.9	127166	127119.9	128011.6	127490.9	127119.9
	worst	127117.8	127270.1	127270.1	127270.1	127270.1	127591.7	127591.7	127270.1	127591.7	127270.1	154244.6	130785.9	127270.1
	std	2.93E-11	45.86385	45.86385	45.86385	45.86385	129.7355	129.7355	45.86385	129.7355	45.86385	8491.069	1004.224	45.86385
	median	127117.8	127146.2	127146.2	127146.2	127146.2	127296.2	127296.2	127146.2	127296.2	127146.2	135091.3	128498.6	127146.2
	rank	1	2	2	2	2	3	3	2	3	2	5	4	2
Part 2	mean	14140.58	14145.26	14145.26	14145.26	14145.26	14156.89	14156.89	14145.26	14156.89	14145.26	14873.71	14266.84	14145.26
	best	14140.58	14140.62	14140.62	14140.62	14140.62	14142.85	14142.85	14140.62	14142.85	14140.62	14154.4	14158.1	14140.62
	worst	14140.58	14157.85	14157.85	14157.85	14157.85	14213.76	14213.76	14157.85	14213.76	14157.85	17624.33	14707	14157.85
	std	3.65E-12	5.077578	5.077578	5.077578	5.077578	16.523	16.523	5.077578	16.523	5.077578	870.5435	127.8971	5.077578
	median	14140.58	14143.54	14143.54	14143.54	14143.54	14153.54	14153.54	14143.54	14153.54	14143.54	14548.13	14240.9	14143.54
	rank	1	2	2	2	2	3	3	2	3	2	5	4	2
Part 3	mean	109379.7	109379.7	109379.7	109379.7	109379.7	109420.2	109383	109379.7	109401.6	109379.7	109575.3	109404.9	109379.7
	best	109379.7	109379.7	109379.7	109379.7	109379.7	109379.7	109379.7	109379.7	109379.7	109379.7	109379.7	109379.7	109379.7
	worst	109379.7	109379.7	109379.7	109379.7	109379.7	109653.2	109394.7	109379.7	109519.1	109379.7	110282.9	109496.4	109379.7
	std	1.3E-10	0.000237	0.000237	0.000283	0.000238	75.24267	4.876363	0.000237	38.61338	0.000237	292.1715	37.74573	0.000238
	median	109379.7	109379.7	109379.7	109379.7	109379.7	109391.5	109380.2	109379.7	109385.7	109379.7	109409.9	109383.6	109379.7
	rank	1	2	6	7	5	11	8	2	9	3	12	10	4
Part 4	mean	122174.7	122181.4	122181.4	122181.4	122181.4	122217	122217	122181.4	122217	122181.4	124354.2	122501.4	122181.4
	best	122174.7	122175.4	122175.4	122175.4	122175.4	122175.9	122175.9	122175.4	122175.9	122175.4	122199.6	122183.6	122175.4
	worst	122174.7	122196.3	122196.3	122196.3	122196.3	122304.6	122304.6	122196.3	122304.6	122196.3	129400	123180.2	122196.3
	std	0	7.024142	7.024142	7.024142	7.024142	34.31714	34.31714	7.024142	34.31714	7.024142	2080.942	265.6335	7.024142
	median	122174.7	122178.2	122178.2	122178.2	122178.2	122207.8	122207.8	122178.2	122207.8	122178.2	123781.9	122430.2	122178.2
	rank	1	2	2	2	2	3	3	2	3	2	5	4	2
Part 5	mean	117984.5	118030.5	118030.5	118030.5	118030.5	118230.4	118230.4	118030.5	118230.4	118030.5	130318.7	119887.5	118030.5
	best	117984.5	117985.3	117985.3	117985.3	117985.3	118041.2	118041.2	117985.3	118041.2	117985.3	119565.7	118423.6	117985.3
	worst	117984.5	118200.1	118200.1	118200.1	118200.1	118821.8	118821.8	118200.1	118821.8	118200.1	167899.6	124465.8	118200.1
	std	4.38E-11	56.70485	56.70485	56.70485	56.70485	201.8162	201.8162	56.70485	201.8162	56.70485	12665.52	1562.169	56.70485
	median	117984.5	118007.9	118007.9	118007.9	118007.9	118143.9	118143.9	118007.9	118143.9	118007.9	123904.7	119218.5	118007.9
	rank	1	2	2	2	2	3	3	2	3	2	5	4	2
Part 6	mean	281385.5	281480.3	281480.3	281480.3	281480.3	281557.3	281557.3	281480.3	281557.3	281480.3	286732.7	282715.6	281480.3
	best	281385.5	281388.5	281388.5	281388.5	281388.5	281405.2	281405.2	281388.5	281405.2	281388.5	281431.2	281537.7	281388.5
	worst	281385.5	281694.8	281694.8	281694.8	281694.8	281912.3	281912.3	281694.8	281912.3	281694.8	304734.2	285463.9	281694.8
	std	0	78.4888	78.4888	78.4888	78.4888	117.433	117.433	78.4888	117.433	78.4888	5534.048	908.9954	78.4888
	median	281385.5	281472	281472	281472	281472	281537.3	281537.3	281472	281537.3	281472	284531.1	282560.2	281472
	rank	1	2	2	2	2	3	3	2	3	2	5	4	2
Part 7	mean	126040.9	126040.9	126040.9	126040.9	126040.9	126102.3	126048.2	126040.9	126075.3	126040.9	126480.2	126097.7	126040.9
	best	126040.9	126040.9	126040.9	126040.9	126040.9	126041	126040.9	126040.9	126041	126040.9	126040.9	126040.9	126040.9
	worst	126040.9	126040.9	126040.9	126040.9	126040.9	126190.4	126074.1	126040.9	126128.1	126040.9	128028.5	126297.7	126040.9
	std	1.97E-10	0.004668	0.004668	0.004699	0.004667	54.9226	10.79951	0.004668	29.61734	0.004668	647.0211	83.5942	0.00467
	median	126040.9	126040.9	126040.9	126040.9	126040.9	126096.8	126042.6	126040.9	126076.1	126040.9	126139.5	126053.6	126040.9
	rank	1	2	5	6	4	10	7	2	8	2	11	9	3
Part 8	mean	19931.72	19937.19	19937.19	19937.19	19937.19	19952.84	19952.84	19937.19	19952.84	19937.19	20911.39	20095.18	19937.19
	best	19931.72	19932.11	19932.11	19932.11	19932.11	19936.51	19936.51	19932.11	19936.51	19932.11	19948.16	19968.74	19932.11
	worst	19931.72	19948.11	19948.11	19948.11	19948.11	19986.23	19986.23	19948.11	19986.23	19948.11	22745.27	20353.64	19948.11
	std	3.65E-12	3.819846	3.819846	3.819846	3.819846	15.53385	15.53385	3.819846	15.53385	3.819846	871.1372	120.2405	3.819846
	median	19931.72	19936.9	19936.9	19936.9	19936.9	19948.68	19948.68	19936.9	19948.68	19936.9	20788.13	20062.98	19936.9
	rank	1	2	2	2	2	4	3	2	3	2	6	5	2
Part 9	mean	4273.017	4273.017	4273.017	4273.017	4273.017	4273.057	4273.017	4273.017	4273.036	4273.017	4273.017	4273.017	4273.017
	best	4273.017	4273.017	4273.017	4273.017	4273.017	4273.017	4273.017	4273.017	4273.017	4273.017	4273.017	4273.017	4273.017
	worst	4273.017	4273.017	4273.017	4273.017	4273.017	4273.417	4273.017	4273.017	4273.217	4273.017	4273.017	4273.017	4273.017
	std	2.25E-12	1.18E-11	1.6E-07	7.77E-07	2.2E-08	0.092888	1.06E-11	1.2E-11	0.046444	1.19E-11	1.18E-11	8.16E-11	5.26E-08
	median	4273.017	4273.017	4273.017	4273.017	4273.017	4273.02	4273.017	4273.017	4273.018	4273.017	4273.017	4273.017	4273.017
	rank	1	3	7	8	5	10	2	3	9	3	3	4	6
Part 10	mean	15223.08	15227.68	15227.68	15227.68	15227.68	15242.15	15242.15	15227.68	15242.15	15227.68	16126.2	15370.71	15227.68
	best	15223.08	15223.44	15223.44	15223.44	15223.44	15224.12	15224.12	15223.44	15224.12	15223.44	15240.71	15231.14	15223.44
	worst	15223.08	15235.64	15235.64	15235.64	15235.64	15276.06	15276.06	15235.64	15276.06	15235.64	18092.61	15633.14	15235.64
	std	7.3E-12	3.325293	3.325293	3.325293	3.325293	14.74156	14.74156	3.325293	14.74156	3.325293	833.0961	114.1078	3.325293
	median	15223.08	15228.83	15228.83	15228.83	15228.83	15238.88	15238.88	15228.83	15238.88	15228.83	15936.41	15345.41	15228.83
	rank	1	2	2	2	2	3	3	2	3	2	5	4	2
Sum rank	10	21	32	35	28	53	38	21	47	22	62	52	27	
Mean rank	1	2.1	3.2	3.5	2.8	5.3	3.8	2.1	4.7	2.2	6.2	5.2	2.7	
Total rank	1	2	6	7	5	11	8	2	9	3	12	10	4	
Wilcoxon: p-value		5.73E-18	3.26E-17	5.73E-18	2.37E-17	1.22E-17	5.73E-18	1.33E-14	2.37E-17	1.79E-17	2.85E-17	8.48E-18	1.79E-17	

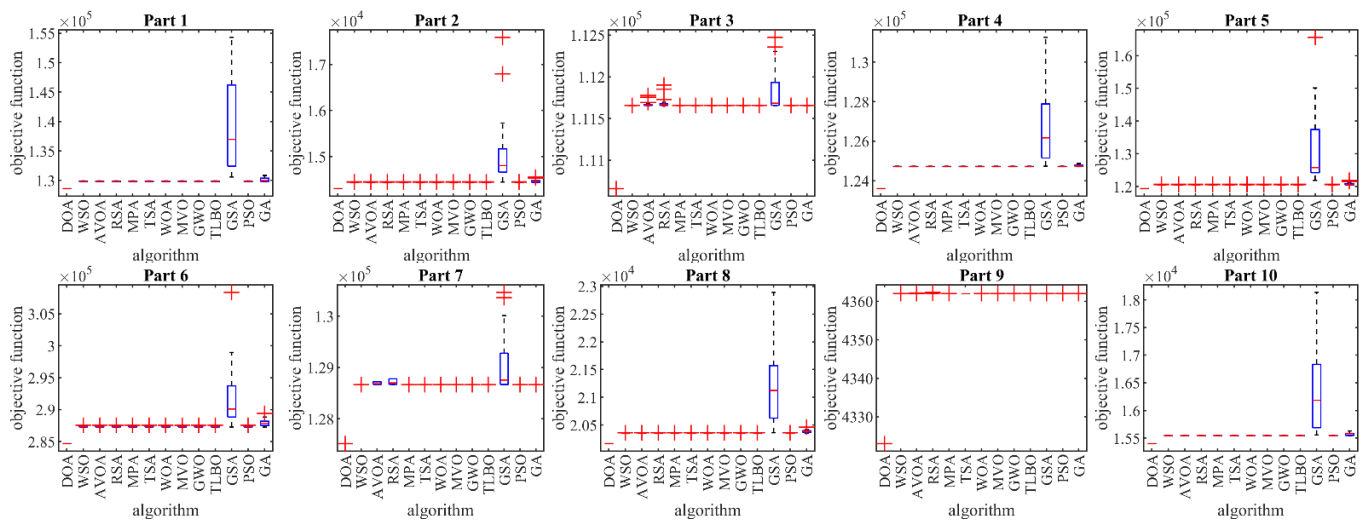


Fig. 1. Boxplot diagrams of DOA and algorithm performances on sustainable lot size optimization.

Figure 1 shows boxplots obtained from the SCM problem implementations to demonstrate the performance of DOA and competing algorithms. The simulation and statistical analysis results demonstrate the statistical superiority of DOA compared to competing algorithms. In all 10 case studies, DOA consistently was the best-performing optimizer. This advantage stems from the unique features integrated within it.

One key factor contributing to DOA's superiority is its two-phase structure, designed to balance exploration and exploitation. During the exploration phase, which emphasizes global search, DOA effectively identifies the optimal search region and avoids becoming trapped in local optima. In the subsequent exploitation phase, which focuses on local search, DOA refines its search process to guide toward better solutions and achieve convergence to the global optimum. This dual-phase approach enhances DOA's overall performance and reliability in solving complex optimization problems.

IV. CONCLUDING REMARKS AND FUTURE WORK

This study presented the DOA as an effective solution for optimizing lot sizing in green SCM. The fundamental inspiration for DOA comes from the dholes' hunting strategy during the two stages of attacking and chasing the prey. The mathematical model of DOA is divided into two phases: (i) exploration based on the simulation of the dholes' position change during the attack towards the prey, and (ii) exploitation based on the simulation of the dholes' position change during the process of chasing the prey. The performance of DOA in addressing the sustainable lot size optimization problem, aiming to balance cost and sustainability, was evaluated across 10 case studies. The findings showed that DOA achieved suitable solutions for optimization problems with its high ability to explore, exploit, and balance during the search process. The results of DOA were compared to those of 12 well-known metaheuristic algorithms, showing that DOA provided superior performance by achieving the best results for most benchmark functions. The findings confirm DOA's ability to enhance decision-making in green manufacturing by balancing cost efficiency and sustainability. Moreover, its

ability to reduce CO₂ emissions while improving operational efficiency underscores its potential as a robust tool for sustainable SCM. DOA had a statistically significant superiority compared to other algorithms. However, despite its advantages, DOA also has limitations/ As with other stochastic methods, there is no guarantee of reaching the global optimum, and it is always possible to design newer algorithms that have superior performance.

Future work could explore the design and development of binary and multi-objective versions of DOA, and employing it to solve optimization problems in various topics and real-world applications.

REFERENCES

- [1] A. Shamsuzzoha and S. Pelkonen, "A robotic process automation model for order-handling optimization in supply chain management," *Supply Chain Analytics*, vol. 9, Mar. 2025, Art. no. 100102, <https://doi.org/10.1016/j.sca.2025.100102>.
- [2] V. K. Sharma, P. Chandna, and A. Bhardwaj, "Green supply chain management related performance indicators in agro industry: A review," *Journal of Cleaner Production*, vol. 141, pp. 1194–1208, Jan. 2017, <https://doi.org/10.1016/j.jclepro.2016.09.103>.
- [3] S. F. Wamba and S. Akter, "Understanding supply chain analytics capabilities and agility for data-rich environments," *International Journal of Operations & Production Management*, vol. 39, Oct. 2019, <https://doi.org/10.1108/IJOPM-01-2019-0025>.
- [4] S. Zhao, T. Zhang, S. Ma, and M. Chen, "Dandelion Optimizer: A nature-inspired metaheuristic algorithm for engineering applications," *Engineering Applications of Artificial Intelligence*, vol. 114, Sep. 2022, Art. no. 105075, <https://doi.org/10.1016/j.engappai.2022.105075>.
- [5] Y. D. Sergeyev, D. E. Kvasov, and M. S. Mukhametzhonov, "On the efficiency of nature-inspired metaheuristics in expensive global optimization with limited budget," *Scientific Reports*, vol. 8, no. 1, Jan. 2018, Art. no. 453, <https://doi.org/10.1038/s41598-017-18940-4>.
- [6] L. Liberti and S. Kucherenko, "Comparison of deterministic and stochastic approaches to global optimization," *International Transactions in Operational Research*, vol. 12, no. 3, pp. 263–285, 2005, <https://doi.org/10.1111/j.1475-3995.2005.00503.x>.
- [7] W. G. Alshanti, I. M. Batiha, M. A. Hammad, and R. Khalil, "A novel analytical approach for solving partial differential equations via a tensor product theory of Banach spaces," *Partial Differential Equations in Applied Mathematics*, vol. 8, Dec. 2023, Art. no. 100531, <https://doi.org/10.1016/j.padiff.2023.100531>.

- [8] R. R. Mostafa, A. M. Khedr, Z. Al Aghbari, I. Afyouni, I. Kamel, and N. Ahmed, "A multi-strategy improved electric eel foraging optimization algorithm: continuous and binary variants for solving optimization problems," *International Journal of Machine Learning and Cybernetics*, Apr. 2025, <https://doi.org/10.1007/s13042-025-02609-w>.
- [9] F. A. Zeidabadi *et al.*, "Archery Algorithm: A Novel Stochastic Optimization Algorithm for Solving Optimization Problems," *Computers, Materials and Continua*, vol. 72, no. 1, pp. 399–416, Feb. 2022, <https://doi.org/10.32604/cmc.2022.024736>.
- [10] M. Q. Taha, M. K. Mohammed, and B. E. Haiba, "Metaheuristic Optimization of Maximum Power Point Tracking in PV Array under Partial Shading," *Engineering, Technology & Applied Science Research*, vol. 14, no. 3, pp. 14628–14633, Jun. 2024, <https://doi.org/10.48084/etasr.7385>.
- [11] J. de Armas, E. Lalla-Ruiz, S. L. Tilahun, and S. Voß, "Similarity in metaheuristics: a gentle step towards a comparison methodology," *Natural Computing*, vol. 21, no. 2, pp. 265–287, Jun. 2022, <https://doi.org/10.1007/s11047-020-09837-9>.
- [12] T. Hamadneh *et al.*, "Barber Optimization Algorithm: A New Human-Based Approach for Solving Optimization Problems," *Computers, Materials & Continua*, vol. 83, no. 2, pp. 2677–2718, 2025, <https://doi.org/10.32604/cmc.2025.064087>.
- [13] T. Hamadneh *et al.*, "Revolution Optimization Algorithm: A New Human-based Metaheuristic Algorithm for Solving Optimization Problems," *International Journal of Intelligent Engineering & Systems*, vol. 18, no. 2, 2025, <https://doi.org/10.22266/ijies2025.0331.38>.
- [14] T. Hamadneh *et al.*, "Paper Publishing Based Optimization: A New Human-Based Metaheuristic Approach for Solving Optimization Tasks," *International Journal of Intelligent Engineering & Systems*, vol. 18, no. 2, 2025, <https://doi.org/10.22266/ijies2025.0331.37>.
- [15] A. O. Topal and O. Altun, "A novel meta-heuristic algorithm: Dynamic Virtual Bats Algorithm," *Information Sciences*, vol. 354, pp. 222–235, Aug. 2016, <https://doi.org/10.1016/j.ins.2016.03.025>.
- [16] P. Savsani and V. Savsani, "Passing vehicle search (PVS): A novel metaheuristic algorithm," *Applied Mathematical Modelling*, vol. 40, no. 5, pp. 3951–3978, Mar. 2016, <https://doi.org/10.1016/j.apm.2015.10.040>.
- [17] Z. Guan, C. Ren, J. Niu, P. Wang, and Y. Shang, "Great Wall Construction Algorithm: A novel meta-heuristic algorithm for engineer problems," *Expert Systems with Applications*, vol. 233, Dec. 2023, Art. no. 120905, <https://doi.org/10.1016/j.eswa.2023.120905>.
- [18] M. Azizi, S. Talatahari, and A. H. Gandomi, "Fire Hawk Optimizer: a novel metaheuristic algorithm," *Artificial Intelligence Review*, vol. 56, no. 1, pp. 287–363, Jan. 2023, <https://doi.org/10.1007/s10462-022-10173-w>.
- [19] M. Braik, M. H. Ryalat, and H. Al-Zoubi, "A novel meta-heuristic algorithm for solving numerical optimization problems: Ali Baba and the forty thieves," *Neural Computing and Applications*, vol. 34, no. 1, pp. 409–455, Jan. 2022, <https://doi.org/10.1007/s00521-021-06392-x>.
- [20] T. Hamadneh *et al.*, "Builder Optimization Algorithm: An Effective Human-Inspired Metaheuristic Approach for Solving Optimization Problems," *International Journal of Intelligent Engineering & Systems*, vol. 18, no. 3, 2025.
- [21] T. Hamadneh *et al.*, "Makeup Artist Optimization Algorithm: A Novel Approach for Engineering Design Challenges," *International Journal of Intelligent Engineering & Systems*, vol. 18, no. 3, 2025.
- [22] Y. Ç. Kuyu and F. Vatansver, "GOZDE: A novel metaheuristic algorithm for global optimization," *Future Generation Computer Systems*, vol. 136, pp. 128–152, Nov. 2022, <https://doi.org/10.1016/j.future.2022.05.022>.
- [23] Y. Lang and Y. Gao, "Dream Optimization Algorithm (DOA): A novel metaheuristic optimization algorithm inspired by human dreams and its applications to real-world engineering problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 436, Mar. 2025, Art. no. 117718, <https://doi.org/10.1016/j.cma.2024.117718>.
- [24] M. Azizi, "Atomic orbital search: A novel metaheuristic algorithm," *Applied Mathematical Modelling*, vol. 93, pp. 657–683, May 2021, <https://doi.org/10.1016/j.apm.2020.12.021>.
- [25] Y. Xiao, H. Cui, R. A. Khurma, and P. A. Castillo, "Artificial lemming algorithm: a novel bionic meta-heuristic technique for solving real-world engineering optimization problems," *Artificial Intelligence Review*, vol. 58, no. 3, Jan. 2025, Art. no. 84, <https://doi.org/10.1007/s10462-024-11023-7>.
- [26] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, Apr. 1997, <https://doi.org/10.1109/4235.585893>.
- [27] N. Singh, M. Jain, and P. Singh, "Nature-Inspired Optimization for Inventory Models with Imperfect Production," in *Data Analytics and Artificial Intelligence for Inventory and Supply Chain Management*, D. K. Sharma and M. Jain, Eds. Springer Nature, 2022, pp. 23–44.
- [28] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," in *Proceedings of the sixth annual conference on Computational learning theory*, 1993, pp. 3–4.
- [29] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, Perth, Australia, 1995, vol. 4, pp. 1942–1948, <https://doi.org/10.1109/ICNN.1995.488968>.
- [30] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: A Gravitational Search Algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, Jun. 2009, <https://doi.org/10.1016/j.ins.2009.03.004>.
- [31] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems," *Computer-Aided Design*, vol. 43, no. 3, pp. 303–315, Mar. 2011, <https://doi.org/10.1016/j.cad.2010.12.015>.
- [32] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-Verse Optimizer: a nature-inspired algorithm for global optimization," *Neural Computing and Applications*, vol. 27, no. 2, pp. 495–513, Feb. 2016, <https://doi.org/10.1007/s00521-015-1870-7>.
- [33] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, Mar. 2014, <https://doi.org/10.1016/j.advengsoft.2013.12.007>.
- [34] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, May 2016, <https://doi.org/10.1016/j.advengsoft.2016.01.008>.
- [35] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine Predators Algorithm: A nature-inspired metaheuristic," *Expert Systems with Applications*, vol. 152, Aug. 2020, Art. no. 113377, <https://doi.org/10.1016/j.eswa.2020.113377>.
- [36] S. Kaur, L. K. Awasthi, A. L. Sangal, and G. Dhiman, "Tunicate Swarm Algorithm: A new bio-inspired based metaheuristic paradigm for global optimization," *Engineering Applications of Artificial Intelligence*, vol. 90, Apr. 2020, Art. no. 103541, <https://doi.org/10.1016/j.engappai.2020.103541>.
- [37] L. Abualigah, M. A. Elaziz, P. Sumari, Z. W. Geem, and A. H. Gandomi, "Reptile Search Algorithm (RSA): A nature-inspired metaheuristic optimizer," *Expert Systems with Applications*, vol. 191, Apr. 2022, Art. no. 116158, <https://doi.org/10.1016/j.eswa.2021.116158>.
- [38] B. Abdollahzadeh, F. S. Gharehchopogh, and S. Mirjalili, "African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems," *Computers & Industrial Engineering*, vol. 158, Aug. 2021, Art. no. 107408, <https://doi.org/10.1016/j.cie.2021.107408>.
- [39] M. Braik, A. Hammouri, J. Atwan, M. A. Al-Betar, and M. A. Awadallah, "White Shark Optimizer: A novel bio-inspired metaheuristic algorithm for global optimization problems," *Knowledge-Based Systems*, vol. 243, May 2022, Art. no. 108457, <https://doi.org/10.1016/j.knsys.2022.108457>.
- [40] F. Wilcoxon, "Individual Comparisons by Ranking Methods," in *Breakthroughs in Statistics: Methodology and Distribution*, S. Kotz and N. L. Johnson, Eds. Springer, 1992, pp. 196–202.