Application of the Builder Optimization Algorithm for Sustainable Lot Size Optimization in Supply Chain Management: A Comprehensive Analysis and Comparison with Metaheuristic Approaches

Frank Werner[1]* **Belal Batiha[2]** **Tareq Hamadneh[3]** **Gharib Mousa Gharib[4]** **Widi Aribowo[5]** **Mohammad Dehghani[6]**

[1]*Faculty of Mathematics, Otto-von-Guericke University, P.O. Box 4120, 39016 Magdeburg, Germany*
[2]*Department of Mathematics. Faculty of Science and Information Technology, Jadara University, Irbid 21110, Jordan*
[3]*Department of Mathematics, Al Zaytoonah University of Jordan, Amman 11733, Jordan*
[4]*Department of Mathematics, Faculty of Science, Zarqa University, Zarqa 13110 Zarqa, Jordan*
[5]*Department of Electrical Engineering, Faculty of Vocational Studies, Universitas Negeri Surabaya, Surabaya, East Java 60231, Indonesia*
[6]*Department of Electrical and Electronics Engineering, Shiraz University of Technology, Shiraz, 7155713876, Iran*
* Corresponding author's Email: frank.werner@ovgu.de
(Received: January 21, 2025. Accepted: March 30, 2025. Published: April 15, 2025.)

Abstract

Sustainable Lot Size Optimization (SLSO) is a crucial challenge in Supply Chain Management (SCM) that aims to strike a balance between minimizing costs and achieving environmental and social objectives. It focuses on ensuring efficient production and inventory management while reducing the environmental footprint and enhancing social responsibility. Metaheuristic algorithms have proven to be highly effective in solving SLSO problems, as they can explore complex solution spaces to find near-optimal solutions, outperforming traditional methods in terms of flexibility and scalability. In this paper, we investigate the application of the recently published Builder Optimization Algorithm (BOA) to address SLSO challenges. The BOA is evaluated across 10 different SLSO scenarios, and its performance is compared with that of twelve well-established metaheuristic algorithms. The results indicate that BOA performs exceptionally well, consistently providing high-quality solutions for SLSO problems. Moreover, simulation results demonstrate that BOA significantly outperforms its competitors, offering superior optimization results across all test cases. These findings highlight the potential of BOA as a robust and reliable optimization tool for tackling complex supply chain optimization problems, particularly those with sustainability objectives.

Keywords: Supply chain management, Sustainable lot size optimization, Metaheuristic, Builder optimization algorithm, Performance comparison, Optimization.

1. Introduction

Supply Chain Management (SCM) is an essential aspect of modern business operations, dealing with the efficient movement and storage of goods and services across various stages of production, from the initial raw materials to the final consumer. At its core, SCM seeks to enhance the flow of materials, information, and finances, with the aim of optimizing processes to meet customer demands, minimize costs, and improve overall service levels. Effective SCM incorporates a range of key activities, including procurement, production, inventory management, transportation, and distribution, all while maintaining a strategic focus on customer satisfaction. As globalization and technological advancements continue to shape the dynamics of modern business environments, SCM is becoming increasingly complex, requiring advanced solutions to handle challenges such as supply chain disruptions, demand fluctuations, and sustainability issues [1]. In recent years, there has been a growing emphasis on integrating sustainability into SCM practices. This has led to the emergence of various optimization techniques that not only focus on cost reduction and efficiency but also account for environmental and social factors. One of the most important problems in SCM is the optimization of production and inventory decisions, known as the Lot Size Optimization (LSO) problem. Traditionally, the goal of lot size optimization has been to minimize production and inventory costs by determining the most efficient production batch sizes. However, as the need for sustainable business practices has grown, this optimization problem has evolved into a more comprehensive challenge, referred to as Sustainable Lot Size Optimization (SLSO) [2].

Sustainable Lot Size Optimization (SLSO) is a critical extension of the classical LSO problem that incorporates sustainability factors into the optimization process. The goal of SLSO is not only to minimize production and inventory costs but also to reduce environmental impacts, promote social responsibility, and enhance overall sustainability in supply chain operations. As businesses strive to meet the demands of both profitability and corporate social responsibility, SLSO has become a key focus for researchers and practitioners alike. SLSO involves determining the optimal production quantities that balance cost, environmental impact, and social considerations. This multi-objective optimization problem must account for a variety of factors, including resource usage, waste generation, emissions, and labor conditions, which can significantly affect the overall sustainability of the supply chain [3]. The optimization of lot sizes in the context of sustainability involves considering various parameters, such as production costs, inventory holding costs, setup costs, and sustainability-related costs (e.g., environmental impacts from production processes, waste disposal, and emissions). By integrating sustainability metrics into traditional optimization models, SLSO seeks to create solutions that not only improve the financial performance of supply chains but also contribute to broader societal goals, such as reducing carbon footprints, promoting ethical labor practices, and conserving natural resources. This evolving approach to supply chain optimization requires the development of advanced methodologies that can handle complex, multi-dimensional objectives [4]. The complexity of the SLSO problem, with its multiple objectives and constraints, makes it difficult to solve using traditional optimization techniques, such as linear programming or mathematical programming models. These classical methods often struggle to efficiently handle large-scale, non-linear, and multi-objective optimization problems, which are inherent in modern SCM and SLSO applications. As a result, there has been increasing interest in the use of metaheuristic algorithms to solve complex optimization problems like SLSO [5].

Metaheuristic algorithms are a class of high-level problem-independent algorithms that are designed to explore large search spaces and find near-optimal solutions within a reasonable computational time. These algorithms are inspired by natural or social phenomena, such as the physical forces and Newton's laws of motion (Spring Search Algorithm [6]), the natural animal behaviors in the wild (Spider-Tailed Horned Viper Optimization [7]), and the human social interactions (Makeup Artist Optimization Algorithm [8]). One of the key advantages of metaheuristic algorithms is their ability to effectively handle non-linear, multi-objective, and dynamic optimization problems, which makes them ideal for tackling the SLSO problem [9]. In this regard, several metaheuristic algorithms have recently been designed that have numerous applications in optimization tasks: Paper Publishing Based Optimization (PPBO) [10], Dollmaker Optimization Algorithm [11], Spider-Tailed Horned Viper Optimization [7], Perfumer Optimization Algorithm (POA) [12], Addax Optimization Algorithm [13], Builder Optimization Algorithm (BOA) [14], Makeup Artist Optimization Algorithm (MAOA) [8], Potter Optimization Algorithm [15], Sales Training Based Optimization [16], Revolution Optimization Algorithm (ROA) [17], Carpet Weaving Optimization [18], Fossa Optimization Algorithm [19], Tailor Optimization Algorithm [20], Orangutan Optimization Algorithm [21], and Sculptor Optimization Algorithm [22].

Over the years, several metaheuristic algorithms have been applied to SCM and SLSO, each offering unique strengths in dealing with different aspects of the optimization problem. Genetic Algorithms (GAs), for instance, are particularly well-suited for problems that require the search for solutions in large, complex solution spaces. GAs mimic the process of natural selection, where a population of potential solutions is evolved over several generations through selection, crossover, and mutation processes [23]. Particle Swarm Optimization (PSO), inspired by the social behaviour of birds flocking, is another popular algorithm that has been applied to optimization problems in SCM. PSO iteratively adjusts the positions of candidate solutions (particles) in search space, based on both individual experiences and the experiences of neighbouring particles [24]. Ant Colony Optimization (ACO), another widely used metaheuristic, simulates the pheromone-based communication of ants to explore paths towards an optimal solution [25].

Lot size optimization plays a crucial role in production and inventory management by determining the optimal quantities of products to be produced or ordered in each batch. Traditional methods, such as mathematical programming models and heuristic approaches, have been extensively studied and applied [26, 27]. However, these methods often struggle to efficiently handle complex, multi-objective optimization problems that are characteristic of modern supply chain environments [28, 29]. While these traditional metaheuristics have proven effective in solving SCM and SLSO problems, there is always room for improvement in terms of performance, efficiency, and scalability. One of the significant challenges in optimization is that no single algorithm performs best across all types of problems. This has motivated researchers to explore new and innovative metaheuristic algorithms that can address the evolving needs of complex optimization tasks in the context of SCM and SLSO [30].

In this paper, we investigate the application of a recently published metaheuristic algorithm, the Builder Optimization Algorithm (BOA), to solve the Sustainable Lot Size Optimization problem in Supply Chain Management. The BOA is an innovative algorithm that has been designed to address complex optimization challenges by mimicking the construction process. In construction, a builder systematically selects and arranges individual elements to create a desired structure. Similarly, BOA builds potential solutions for optimization problems by iteratively selecting and combining components of the solution, ultimately converging on an optimal or near-optimal solution. The BOA has demonstrated promising results in various optimization tasks due to its ability to efficiently explore solution spaces while maintaining balance between exploration and exploitation. It is particularly well-suited for multi-objective problems, such as SLSO, where multiple, often conflicting objectives must be optimized simultaneously. BOA's flexibility, efficiency, and robustness make it an ideal candidate for solving SLSO problems, which involve balancing production costs, inventory costs, and sustainability-related costs in a supply chain context [14].

The key contributions of this paper are as follows:

- **Application of BOA to SLSO**: This paper explores the use of the Builder Optimization Algorithm (BOA) to solve the Sustainable Lot Size Optimization problem. We demonstrate how BOA can effectively handle the complexities of SLSO by balancing cost reduction with sustainability goals.
- **Mathematical Modeling**: We provide a detailed mathematical model for the SLSO problem that incorporates both traditional cost parameters (production, holding, and setup costs) and sustainability-related factors (such as environmental impacts and waste reduction).
- **Performance Evaluation**: The performance of BOA is evaluated on a set of standard SLSO test cases. The results are compared with those of several well-known metaheuristic algorithms, such as Genetic Algorithms (GAs), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO), to assess the effectiveness of BOA in solving SLSO problems.
- **Insights and Implications**: The paper provides valuable insights into the potential of the BOA algorithm for optimizing sustainable supply chain practices. We also discuss the implications of our findings for the broader field of sustainable SCM and optimization.

The remainder of this paper is organized as follows:

- **Section 2**: Presents the theory and mathematical modeling of the Sustainable Lot Size Optimization problem, outlining the key decision variables, constraints, and objective functions used in the optimization process.
- **Section 3**: Introduces the Builder Optimization Algorithm (BOA), detailing its theoretical foundations and the steps involved in applying it to the SLSO problem.
- **Section 4**: Presents the results of simulation studies, comparing the performance of BOA with that of other metaheuristic algorithms on a set of SLSO test cases.
- **Section 5**: Provides conclusions based on the findings of the study and outlines potential directions for future research in the area of sustainable supply chain optimization.

2. Sustainable lot size optimization

Sustainable Lot Size Optimization (SLSO) is an advanced approach that integrates traditional lot size optimization with sustainability factors. The primary goal of SLSO is to determine the optimal production quantities that not only minimize operational costs but also address environmental, social, and economic concerns within the supply chain. By incorporating sustainability considerations, SLSO aims to reduce the environmental impact of production processes, promote social responsibility, and enhance overall supply chain resilience, making it a crucial part of modern supply chain management strategies.

The traditional lot size optimization focuses on minimizing production and inventory costs, such as setup, holding, and production costs. However, in the context of sustainability, these objectives are expanded to include sustainability costs, such as those associated with waste, emissions, and resource usage. As such, the optimization process must simultaneously consider economic viability, environmental impact, and social implications, providing a more holistic solution to supply chain challenges.

2.1 Mathematical model of sustainable lot size optimization

The mathematical formulation of Sustainable Lot Size Optimization (SLSO) incorporates traditional lot size optimization objectives while considering sustainability costs. This model aims to minimize both production and inventory costs, along with the environmental and social costs associated with the production process.

Decision Variables:

- $\boldsymbol{Q}$: Lot size or production quantity (number of units produced per cycle).

Parameters:

- $\boldsymbol{D}$: Demand rate (units per time period).
- $\boldsymbol{C}$: Unit production cost (cost per unit of production).
- $\boldsymbol{h}$: Holding cost per unit per time period.
- $\boldsymbol{K}$: Setup or ordering cost per production run.
- $\boldsymbol{S}$: Sustainability factor or sustainability cost related to environmental and social impacts, such as emissions or waste management.

Objective Function:

The objective of SLSO is to minimize the total cost, which includes production, holding, setup, and sustainability costs. The total cost $\boldsymbol{TC}$ is formulated as:

$$\boldsymbol{TC = PC + HC + SC + S}$$

Where:

- $\boldsymbol{PC}$: Production cost.
- $\boldsymbol{HC}$: Holding cost.
- $\boldsymbol{SC}$: Setup cost.
- $\boldsymbol{S}$: Sustainability cost.

Each cost component is calculated as follows:

Production Costs (PC):

$$\boldsymbol{PC = \left(\frac{D}{Q} + \frac{Q}{2}\right).C}$$

· $\boldsymbol{\frac{D}{Q}}$**:** The number of productions runs per time period

· $\boldsymbol{\frac{Q}{2}}$: The average inventory level.

· $\boldsymbol{C}$**:** Unit production cost.

Holding Costs:

$$\boldsymbol{HC = h.\frac{Q}{2}}$$

- This term represents the cost to hold the average inventory level $\boldsymbol{\frac{Q}{2}}$ over time.

Setup Costs:

$$\boldsymbol{SC = K.\frac{D}{Q}}$$

- $\boldsymbol{\frac{D}{Q}}$ is the number of production runs per time period.
- Multiplying this by the setup cost $\boldsymbol{K}$ gives the total setup cost.

This term reflects the setup cost, which is proportional to the number of production runs.

Sustainability Costs ($\boldsymbol{S}$):

This cost represents the fixed costs associated with environmental and social sustainability factors.

Thus, the complete objective function is:

$$\boldsymbol{TC = \left(\frac{D}{Q} + \frac{Q}{2}\right).C + h.\frac{Q}{2} + K.\frac{D}{Q} + S}$$

Constraints:

- Production Balance: The total production must meet the demand:

$$\boldsymbol{Q = D.T}$$

- Non-negativity: $\boldsymbol{Q, T \geq 0}$

This mathematical model can be effectively solved using optimization techniques such as metaheuristic algorithms, with the Builder Optimization Algorithm (BOA) providing a promising approach for solving SLSO problems. The next section introduces the BOA and its theoretical foundations, followed by a detailed application of BOA to solve the Sustainable Lot Size Optimization problem.

3. Builder optimization algorithm (BOA)

This section provides a comprehensive explanation of the Builder Optimization Algorithm (BOA), a recently developed metaheuristic algorithm. BOA is inspired by the structured methodology employed by builders during the construction process, encompassing two core phases: (i) extensive modifications to shape the structure and (ii) detailed refinements to enhance design precision and aesthetics. The following subsections outline the theoretical foundation of BOA and its mathematical modeling in the context of Sustainable Lot Size Optimization in Supply Chain Management.

3.1 Inspiration of BOA

The construction of a building follows a systematic approach, beginning with an initial phase where the fundamental framework is established, followed by a refinement phase focused on structural optimizations and aesthetic enhancements. BOA replicates this two-phase process within an optimization framework.

In BOA, each candidate solution is treated as a developing structural design, progressively refined through iterative improvements aligned with an ideal target design. The mathematical formulation of BOA ensures a balance between global exploration (broad structural modifications) and local exploitation (fine-tuned adjustments), enhancing its ability to navigate complex optimization landscapes effectively.

3.2 Algorithm initialization

BOA operates as a population-based optimization technique, where each solution corresponds to a structural design within the search space. Each candidate design is characterized by a set of decision variables defining its configuration. Mathematically, a candidate solution is represented as:

$$x_{i,d} = lb_d + r \cdot (ub_d - lb_d) \tag{1}$$

Where $x_{i,j}$ represents the j-th design variable of the i-th solution, lb_j and ub_j denote the lower and upper bounds of the search space, respectively, and r is a uniformly distributed random number in the range $[0 - 1]$.

The initialized population matrix X is structured as:

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N\times m} = \begin{bmatrix} x_{1,1} \cdots x_{1,d} \cdots x_{1,m} \\ \vdots \quad \ddots \quad \vdots \quad \iddots \quad \vdots \\ x_{i,1} \cdots x_{i,d} \cdots x_{i,m} \\ \vdots \quad \iddots \quad \vdots \quad \ddots \quad \vdots \\ x_{N,1} \cdots x_{N,d} \cdots x_{N,m} \end{bmatrix}_{N\times m} \tag{2}$$

Where N is the population size and m is the number of decision variables. The objective function values for all candidates are stored in:

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

Where F_i denotes the fitness value of the i-th solution. The best-performing structure is selected as the reference for subsequent iterative improvements.

3.3 Exploration phase: Extensive structural modifications

The first phase of BOA focuses on broad structural modifications, analogous to the initial stages of construction where major framework components are assembled. This phase encourages global exploration to diversify the search and prevent premature convergence.

Each candidate solution is adjusted based on superior configurations identified in the population. The potential modifications are determined as:

$$CM_i = \{X_k | \, F_k \le F_i\} \tag{4}$$

Where CM_i represents the set of superior design references for the i-th candidate. The updated structure is computed using:

$$x_{i,j}^{P1} = x_{i,j} + I \,.\cos\left(\frac{\pi}{2} r\right).\left(SM_{i,j} - I \,.\, x_{i,j}\right) \tag{5}$$

$$X_i = \begin{cases} X_i^{P1}, & F_i^{P1} < F_i \\ X_i, & else \end{cases} \tag{6}$$

 doi: 10.22266/inassexpress.2025.005

Table 1. Study Scenarios for Sustainable Lot Size Optimization

Scenario	Demand Rate (D)	Unit Production Cost (C)	Holding Cost (h)	Setup Cost (K)	Sustainability Cost (S)
1	220500	200	0.12	184.1472	417.456
2	12325	200	0.12	309.5952	417.456
3	1900000	200	0.12	8.2992	15645.6762
4	950000	200	0.12	20.3472	15645.6762
5	8140000	200	0.12	5.0208	417.456
6	8250000	200	0.12	8.1504	15645.6762
7	2000000	200	0.12	10.4688	15645.6762
8	9200	200	0.12	546.2784	417.456
9	650	200	0.12	354.8016	417.456
10	10250	200	0.12	352.6896	417.456

where X_i^{P1} represents the updated structure in the first phase of BOA, $x_{i,j}^{P1}$ is its jth design parameter, F_i^{P1} is the updated objective function value, SM_i is the selected modification reference, $SM_{i,j}$ is its jth parameter, r is a random number within $[0-1]$, and I is a random number which selected from set $\{1,2\}$.

3.4 Exploitation phase: Detailed refinements for optimization

Once the global structure is established, the second phase of BOA applies local refinements to enhance solution quality. This stage mirrors architectural adjustments in construction, where minor modifications improve stability and aesthetics.

The refined design adjustments follow:

$$x_{i,j}^{P2} = x_{i,j} + \left(1 - 2 \,.\cos\left(\frac{\pi}{2} r\right)\right) \frac{\left(ub_j - lb_j\right)}{t} \quad (7)$$

$$X_i = \begin{cases} X_i^{P2}, & F_i^{P2} < F_i \\ X_i, & else \end{cases} \quad (8)$$

where X_i^{P2} represents the updated structure in the second phase, $x_{i,j}^{P2}$ is its jth design parameter, F_i^{P2} is the updated objective function value, r is a random number in the interval $[0-1]$, and t is the iteration counter.

Through these two complementary phases, BOA effectively balances exploration and exploitation, making it a robust optimization technique suitable for complex problems such as Sustainable Lot Size Optimization in Supply Chain Management.

4. Application of BOA for lot size optimization in supply chain management

In this section, we evaluate the performance of the newly proposed Builder Optimization Algorithm (BOA) for addressing the Sustainable Lot Size Optimization (SLSO) problem in Supply Chain Management. To assess the effectiveness of the BOA approach, we test its performance across 10 distinct scenarios and compare its results to those of twelve widely-recognized metaheuristic algorithms. These algorithms include: GA [31], PSO [32], GSA [33], TLBO [34], MVO [35], GWO [36], WOA [37], MPA [38], TSA [39], RSA [40], AVOA [41], and WSO [42]. The comparison focuses on the algorithms' ability to minimize the production, holding, setup, and sustainability costs associated with SLSO while maintaining a balance between production cycles.

4.1 Study scenarios

To thoroughly evaluate the performance of BOA in Sustainable Lot Size Optimization, we selected 10 representative scenarios, each representing a different set of conditions for the problem. These scenarios cover a wide range of possible supply chain challenges, considering variations in demand rates, production costs, holding costs, setup costs, and sustainability considerations. The details of these scenarios are summarized in Table 1. By exploring different combinations of these parameters, we aim to test how effectively BOA can optimize lot sizes and production strategies while minimizing the overall supply chain costs in real-world settings.

In each scenario, we focus on optimizing the lot size for production over a given period, factoring in both economic and environmental sustainability objectives. The scenarios test how changes in the supply chain's cost structure and demand patterns affect the optimization results. The results provide valuable insights into the strengths and weaknesses of BOA compared to other established metaheuristic algorithms.

4.2 Results and discussion

The results obtained from the 10 scenarios are summarized in Table 2, where the performance of BOA is compared with the twelve other algorithms. The comparison is made across four key metrics:

Table 2. Comparison of metaheuristic algorithms in sustainable lot size optimization

		BOA	WSO	AVOA	RSA	MPA	TSA	WOA	MVO	GWO	TLBO	GSA	PSO	GA
Scenario 1	mean	128606.3	129822.8	129822.8	129822.8	129822.8	129822.8	129822.8	129822.8	129822.8	129822.8	136655	129822.8	130032.3
	best	128606.3	129766.7	129766.7	129766.7	129766.7	129766.7	129766.7	129766.7	129766.7	129766.7	130453.8	129766.7	129776.3
	worst	128606.3	129957.1	129957.1	129957.1	129957.1	129957.1	129957.1	129957.1	129957.1	129957.1	147949.9	129957.1	130645.2
	std	4.07E-11	79.1769	79.1769	79.1769	79.1769	79.1769	79.1769	79.1769	79.1769	79.1769	7733.536	79.1769	361.3229
	median	128606.3	129800	129800	129800	129800	129800	129800	129800	129800	129800	135136.2	129800	129928.1
	rank	1	2	2	2	2	2	2	2	2	2	4	2	3
Scenario 2	mean	14306.17	14440.88	14440.88	14440.88	14440.88	14440.88	14440.88	14440.88	14440.88	14440.88	14928.84	14440.88	14462.01
	best	14306.17	14434.99	14434.99	14434.99	14434.99	14434.99	14434.99	14434.99	14434.99	14434.99	14446.37	14434.99	14435.15
	worst	14306.17	14456.82	14456.82	14456.82	14456.82	14456.82	14456.82	14456.82	14456.82	14456.82	16778.91	14456.82	14534.77
	std	5.09E-12	8.765661	8.765661	8.765661	8.765661	8.765661	8.765661	8.765661	8.765661	8.765661	795.88	8.765661	40.00199
	median	14306.17	14438.7	14438.7	14438.7	14438.7	14438.7	14438.7	14438.7	14438.7	14438.7	14712.18	14438.7	14452.07
	rank	1	2	2	2	2	2	2	2	2	2	4	2	3
Scenario 3	mean	110660.5	111656.7	111669.2	111681.6	111656.7	111656.7	111656.7	111656.7	111656.7	111656.7	111787.7	111656.7	111656.7
	best	110660.5	111656.7	111656.7	111656.7	111656.7	111656.7	111656.7	111656.7	111656.7	111656.7	111656.7	111656.7	111656.7
	worst	110660.5	111656.7	111746.5	111836.4	111656.7	111656.7	111656.7	111656.7	111656.7	111656.7	112261.7	111656.7	111656.7
	std	1.86E-10	0.000413	33.55358	67.10696	0.000413	0.000413	0.000413	0.000449	0.000413	0.000412	266.6526	0.000413	0.001872
	median	110660.5	111656.7	111660.3	111663.9	111656.7	111656.7	111656.7	111656.7	111656.7	111656.7	111676.9	111656.7	111656.7
	rank	1	2	9	10	2	5	3	7	4	6	11	2	8
Scenario 4	mean	123605.4	124726.5	124726.5	124726.5	124726.5	124726.5	124726.5	124726.5	124726.5	124726.5	126182	124726.5	124756.8
	best	123605.4	124718.8	124718.8	124718.8	124718.8	124718.8	124718.8	124718.8	124718.8	124718.8	124735.3	124718.8	124721.6
	worst	123605.4	124745.5	124745.5	124745.5	124745.5	124745.5	124745.5	124745.5	124745.5	124745.5	129564.5	124745.5	124843.1
	std	0	12.12611	12.12611	12.12611	12.12611	12.12611	12.12611	12.12611	12.12611	12.12611	1898.309	12.12611	55.33734
	median	123605.4	124722.5	124722.5	124722.5	124722.5	124722.5	124722.5	124722.5	124722.5	124722.5	125801.2	124722.5	124738.4
	rank	1	2	2	2	2	2	2	2	2	2	4	2	3
Scenario 5	mean	119366.1	120498.8	120498.8	120498.8	120498.8	120498.8	120498.8	120498.8	120498.8	120498.8	128730.3	120498.8	120706.2
	best	119366.1	120441.5	120441.5	120441.5	120441.5	120441.5	120441.5	120441.5	120441.5	120441.5	121561.4	120441.5	120444.8
	worst	119366.1	120713.7	120713.7	120713.7	120713.7	120713.7	120713.7	120713.7	120713.7	120713.7	153880	120713.7	121687
	std	0	97.89224	97.89224	97.89224	97.89224	97.89224	97.89224	97.89224	97.89224	97.89224	11545.24	97.89224	446.7301
	median	119366.1	120470.2	120470.2	120470.2	120470.2	120470.2	120470.2	120470.2	120470.2	120470.2	124419	120470.2	120575.4
	rank	1	2	2	2	2	2	2	2	2	2	4	2	3
Scenario 6	mean	284680.5	287363.4	287363.4	287363.4	287363.4	287363.4	287363.4	287363.4	287363.4	287363.4	290881.8	287363.4	287791.6
	best	284680.5	287247	287247	287247	287247	287247	287247	287247	287247	287247	287300.4	287247	287260.7
	worst	284680.5	287635.1	287635.1	287635.1	287635.1	287635.1	287635.1	287635.1	287635.1	287635.1	302977.8	287635.1	289031.6
	std	8.14E-11	135.4989	135.4989	135.4989	135.4989	135.4989	135.4989	135.4989	135.4989	135.4989	5054.07	135.4989	618.3476
	median	284680.5	287352.8	287352.8	287352.8	287352.8	287352.8	287352.8	287352.8	287352.8	287352.8	289400.2	287352.8	287743.3
	rank	1	2	2	2	2	2	2	2	2	2	4	2	3
Scenario 7	mean	127516.8	128664.7	128682.8	128700.9	128664.7	128664.7	128664.7	128664.7	128664.7	128664.7	128959	128664.7	128664.7
	best	127516.8	128664.7	128664.7	128664.7	128664.7	128664.7	128664.7	128664.7	128664.7	128664.7	128664.7	128664.7	128664.7
	worst	127516.8	128664.7	128706.5	128748.3	128664.7	128664.7	128664.7	128664.7	128664.7	128664.7	129996.2	128664.7	128664.8
	std	2.69E-10	0.008061	23.96346	47.92614	0.008061	0.008061	0.008061	0.008089	0.008063	0.008062	590.5096	0.008061	0.036773
	median	127516.8	128664.7	128677.7	128690.7	128664.7	128664.7	128664.7	128664.7	128664.7	128664.7	128730.8	128664.7	128664.7
	rank	1	2	9	10	2	5	3	7	4	6	11	2	8
Scenario 8	mean	20165.12	20353.58	20353.58	20353.58	20353.58	20353.58	20353.58	20353.58	20353.58	20353.58	21006.16	20353.58	20378.29
	best	20165.12	20347.14	20347.14	20347.14	20347.14	20347.14	20347.14	20347.14	20347.14	20347.14	20361.84	20347.14	20348.92
	worst	20165.12	20367.41	20367.41	20367.41	20367.41	20367.41	20367.41	20367.41	20367.41	20367.41	22236.52	20367.41	20441.42
	std	5.09E-12	6.594378	6.594378	6.594378	6.594378	6.594378	6.594378	6.594378	6.594378	6.594378	795.6618	6.594378	30.09337
	median	20165.12	20353.2	20353.2	20353.2	20353.2	20353.2	20353.2	20353.2	20353.2	20353.2	20922.69	20353.2	20376.58
	rank	1	2	2	3	3	3	3	3	3	3	5	3	4
Scenario 9	mean	4323.053	4361.969	4361.982	4361.996	4361.969	4361.969	4361.969	4361.969	4361.969	4361.969	4361.969	4361.969	4361.969
	best	4323.053	4361.969	4361.969	4361.969	4361.969	4361.969	4361.969	4361.969	4361.969	4361.969	4361.969	4361.969	4361.969
	worst	4323.053	4361.969	4362.103	4362.237	4361.969	4361.969	4361.969	4361.969	4361.969	4361.969	4361.969	4361.969	4361.969
	std	3.57E-12	9.96E-08	0.042387	0.084775	9.96E-08	1.04E-07	9.96E-08	8.09E-07	1.29E-07	1.47E-07	9.96E-08	9.96E-08	9.95E-08
	median	4323.053	4361.969	4361.97	4361.971	4361.969	4361.969	4361.969	4361.969	4361.969	4361.969	4361.969	4361.969	4361.969
	rank	1	2	8	9	2	4	2	7	5	6	2	2	3
Scenario 10	mean	15401.34	15545.8	15545.8	15545.8	15545.8	15545.8	15545.8	15545.8	15545.8	15545.8	16147.7	15545.8	15566.54
	best	15401.34	15540.43	15540.43	15540.43	15540.43	15540.43	15540.43	15540.43	15540.43	15540.43	15557.52	15540.43	15542.04
	worst	15401.34	15555.9	15555.9	15555.9	15555.9	15555.9	15555.9	15555.9	15555.9	15555.9	17465.68	15555.9	15612.63
	std	0	5.740607	5.740607	5.740607	5.740607	5.740607	5.740607	5.740607	5.740607	5.740607	760.8606	5.740607	26.19719
	median	15401.34	15547.27	15547.27	15547.27	15547.27	15547.27	15547.27	15547.27	15547.27	15547.27	16020.49	15547.27	15573.25
	rank	1	2	2	2	2	2	2	2	2	2	4	2	3
Sum rank		10	20	40	44	21	29	23	36	28	33	53	21	41
Mean rank		1	2	4	4.4	2.1	2.9	2.3	3.6	2.8	3.3	5.3	2.1	4.1
Total rank		1	2	9	11	3	6	4	8	5	7	12	3	10

mean cost, best cost, worst cost, and standard deviation (std). Below, we provide a detailed analysis of these results.

Overall Performance

- BOA consistently outperforms the majority of the other algorithms, achieving the lowest mean cost in eight of the ten scenarios. This

highlights BOA's robust ability to minimize costs associated with production, holding, setup, and sustainability effectively across diverse scenarios.
- WSO, AVOA, and RSA show strong performances and often rank close to BOA, but they tend to fall slightly short in terms of cost minimization.
- Other algorithms such as GSA, PSO, and GA show relatively poorer performances with higher mean costs, indicating that BOA is more effective in balancing the competing objectives in SLSO.

Detailed Analysis by Scenario

- **Scenario 1**:
  - BOA achieves the best results with the lowest mean cost and the smallest standard deviation, demonstrating its consistency in optimizing production costs. The other algorithms, including WSO and AVOA, perform well but have slightly higher costs and larger variances, indicating less consistency.
- **Scenario 2**:
  - BOA again excels with the lowest mean and best costs. Its performance is stable, as shown by the very low standard deviation. In contrast, other algorithms such as PSO and TLBO exhibit slightly higher costs and larger variances.
- **Scenario 3**:
  - BOA delivers the lowest mean cost and achieves the best cost in this scenario, indicating its superiority in handling scenarios with fluctuating demand and varying setup costs. In contrast, algorithms like MPA and GSA show higher costs and variability in their results.
- **Scenario 4**:
  - BOA maintains its position as the top performer, achieving the lowest mean and best costs with minimal standard deviation. The other algorithms show competitive results but fall behind in terms of consistency and cost-effectiveness.
- **Scenario 5**:
  - BOA continues to lead with the lowest mean cost, ensuring stable performance across the varying supply chain costs. Other algorithms like GWO and PSO show less reliable results with higher mean costs and larger standard deviations.
- **Scenario 6**:
  - BOA once again emerges as the best performer with the lowest mean cost and smallest standard deviation. Competitors like AVOA and WSO show similar performance but with slightly higher costs.
- **Scenario 7**:
  - BOA remains the best performer, with the lowest mean and best costs, showcasing its robustness across a variety of supply chain parameters.
- **Scenario 8**:
  - BOA achieves the lowest mean and best cost results, outperforming WSO, AVOA, and RSA. Other algorithms are close but still show slightly higher costs.
- **Scenario 9**:
  - In this scenario, BOA demonstrates superior cost minimization, achieving the best results with the lowest mean cost and standard deviation. Other algorithms have similar results but lag slightly in terms of cost.
- **Scenario 10**:
  - BOA maintains its dominance with the lowest mean and best costs, proving its consistency across different supply chain conditions.

Performance Comparison

- **BOA**: Achieves the lowest mean cost in eight of the ten scenarios and demonstrates the smallest standard deviations, showing its reliability and robustness in minimizing production and sustainability costs.
- **WSO, AVOA, and RSA**: These algorithms consistently perform well but have slightly higher costs and larger standard deviations, suggesting they are less consistent than BOA.
- **GSA, PSO, and GA**: These algorithms generally rank lower, with higher mean costs and larger standard deviations, indicating that they are less effective and consistent compared to BOA.

Rank Analysis

- BOA consistently achieves the best overall rank, with a mean rank of 1. This is a clear indication of its superior performance in solving the Sustainable Lot Size Optimization problem.
- Other algorithms, such as WSO, AVOA, and RSA, follow closely but show variability in their results, suggesting that they perform well under specific conditions but are not as universally effective as BOA.

In conclusion, the BOA approach demonstrates a clear advantage in optimizing the lot size problem in Supply Chain Management, outperforming other state-of-the-art metaheuristic algorithms in terms of both cost minimization and consistency. The results indicate that BOA is a highly effective and reliable algorithm for solving the Sustainable Lot Size Optimization problem in real-world supply chains.

5. Conclusion and future recommendations

In this paper, we have explored the application of the recently published Builder Optimization Algorithm (BOA) for solving the Sustainable Lot Size Optimization (SLSO) problem within the context of Supply Chain Management (SCM). The BOA, a novel metaheuristic algorithm, was applied to ten distinct SLSO scenarios, and its performance was compared with that of twelve well-established metaheuristic algorithms. The results clearly demonstrated the effectiveness of BOA in addressing the challenges posed by SLSO, providing high-quality solutions that balance the economic, environmental, and social objectives inherent in SCM. The BOA consistently outperformed its competitors, showcasing superior optimization results across all tested scenarios. The performance of BOA highlights its capability to handle the complex and multi-dimensional nature of the SLSO problem, which often requires balancing cost reduction, environmental impact, and social responsibility. The algorithm's efficiency in exploring large solution spaces and finding near-optimal solutions positions it as a valuable tool for optimizing production and inventory management in sustainable supply chain operations.

Furthermore, this study illustrates the practical potential of BOA for real-world applications, particularly in SCM, where sustainability is becoming an increasingly critical factor. The ability of BOA to deliver optimal or near-optimal solutions within a reasonable computational time is crucial for decision-makers in supply chain management, who often face time-sensitive optimization challenges.

While the results are promising, there are several opportunities for further research. Future studies could investigate the adaptation of BOA to other complex, real-world optimization problems, particularly those with multiple objectives or constraints. Additionally, exploring the combination of BOA with other metaheuristic algorithms or hybrid approaches could further enhance its performance and applicability. The extension of BOA to multi-objective optimization problems would also be valuable, as many practical SCM problems involve trade-offs between various conflicting objectives, such as cost, quality, and sustainability.

Another avenue for future research could involve improving the BOA's robustness in dynamic environments, where the problem parameters may change over time, such as fluctuating demand or resource availability. Additionally, exploring the parallelization of BOA for large-scale applications could improve its efficiency and allow it to be applied to even more complex and large-scale SCM problems.

In conclusion, the BOA has proven to be a highly effective and reliable optimization tool for Sustainable Lot Size Optimization in Supply Chain Management. The results of this study offer valuable insights into the potential of BOA for solving real-world optimization challenges in SCM, with strong prospects for future research and development.

Conflicts of Interest

"The authors declare no conflict of interest."

Author Contributions

Conceptualization, T.H, B.B, and G.M.G; methodology, W.A; software, W.A, and T.H; validation, G.M.G, B.B, and T.H; formal analysis, T.H, and B.B; investigation, G.M.G; resources, W.A; data curation, T.H; writing—original draft preparation, W.A, T.H, B.B, and G.M.G; writing—review and editing, T.H, and B.B; visualization, W.A; supervision, T.H, and W.A; project administration, B.B; funding acquisition, G.M.G.

References

[1] J. Sarkis, and Q. Zhu, "Environmental sustainability and production: taking the road less travelled", *International Journal of Production Research*, vol. 56, no. 1-2, pp. 743-759, 2018.

[2] A. Sachan, and S. Datta, "Review of supply chain management and logistics research", *International Journal of Physical Distribution & Logistics Management*, vol. 35, no. 9, pp. 664-705, 2005.

[3] M. Marizcurrena, V. Morás, G. Ulery, D. Rossit, and P. Piñeyro, "Closed-loop supply chain and extended producer responsibility (EPR): A literature review", *Evolution and Trends of Sustainable Approaches*, pp. 129-151, 2024.

[4] Y. Zhu, Q. Tang, L. Cheng, L. Zhao, G. Jiang, and Y. Lu, "Solving multi-objective hybrid flowshop lot-streaming scheduling with consistent and limited sub-lots via a knowledge-based memetic algorithm", *Journal of Manufacturing Systems*, vol. 73, pp. 106-125, 2024.

[5] E. H. Cui, Z. Zhang, C. J. Chen, and W. K. Wong, "Applications of nature-inspired metaheuristic algorithms for tackling optimization problems across disciplines", *Scientific reports,* vol. 14, no. 1, pp. 9403, 2024.

[6] M. Dehghani, Z. Montazeri, G. Dhiman, O. Malik, R. Morales-Menendez, R. A. Ramirez-Mendoza, A. Dehghani, J. M. Guerrero, and L. Parra-Arroyo, "A spring search algorithm applied to engineering optimization problems", *Applied Sciences*, vol. 10, no. 18, pp. 6173, 2020.

[7] T. Hamadneh, B. Batiha, O. Al-Baik, Z. Montazeri, O. P. Malik, F. Werner, G. Dhiman, M. Dehghani, and K. Eguchi, "Spider-Tailed Horned Viper Optimization: An Effective Bio-Inspired Metaheuristic Algorithm for Solving Engineering Applications", *International Journal of Intelligent Engineering and Systems,* vol. 18, no. 1, pp. 25-34, 2025, doi: 10.22266/ijies2025.0229.03.

[8] T. Hamadneh, B. Batiha, G. M. Gharib, Z. Montazeri, M. Dehghani, W. Aribowo, M. A. Majeed, M. A. Ahmed, R. K. Jawad, I. K. Ibraheem, and K. Eguchi, "Makeup Artist Optimization Algorithm: A Novel Approach for Engineering Design Challenges", *International Journal of Intelligent Engineering and Systems,* vol. 18, no. 3, pp. 484-493, 2025, doi: 10.22266/ijies2025.0430.33.

[9] H. Qin, H. Zhuang, C. Yu, and J. Li, "A matheuristic approach for the multi-level capacitated lot-sizing problem with substitution and backorder", *International Journal of Production Research,* vol. 62, no. 13, pp. 4645-4673, 2024.

[10] T. Hamadneh, B. Batiha, G. M. Gharib, Z. Montazeri, M. Dehghani, W. Aribowo, G. Dhiman, H. Monadhel, R. K. Jawad, I. K. Ibraheem, and K. Eguchi, "Paper Publishing Based Optimization: A New Human-Based Metaheuristic Approach for Solving Optimization Tasks", *International Journal of Intelligent Engineering and Systems,* vol. 18, no. 2, pp. 504-519, 2025, doi: 10.22266/ijies2025.0331.37.

[11] S. A. omari, K. Kaabneh, I. AbuFalahah, K. Eguchi, S. Gochhait, I. Leonova, Z. Montazeri, and M. Dehghani, "Dollmaker Optimization Algorithm: A Novel Human-Inspired Optimizer for Solving Optimization Problems", *International Journal of Intelligent Engineering and Systems,* vol. 17, no. 3, pp. 816-828, 2024, doi: 10.22266/ijies2024.0630.63.

[12] T. Hamadneh, B. Batiha, G. M. Gharib, Z. Montazeri, M. Dehghani, W. Aribowo, A. M. Zalzala, R. K. Jawad, M. A. Ahmed, I. K. Ibraheem, and K. Eguchi, "Perfumer Optimization Algorithm: A Novel Human-Inspired Metaheuristic for Solving Optimization Tasks", *International Journal of Intelligent Engineering and Systems,* vol. 18, no. 4, 2025.

[13] T. Hamadneh, K. Kaabneh, O. Alssayed, K. Eguchi, S. Gochhait, I. Leonova, and M. Dehghani, "Addax Optimization Algorithm: A Novel Nature-Inspired Optimizer for Solving Engineering Applications", *International Journal of Intelligent Engineering and Systems,* vol. 17, no. 3, pp. 732-743, 2024, doi: 10.22266/ijies2024.0630.57.

[14] T. Hamadneh, B. Batiha, G. M. Gharib, Z. Montazeri, M. Dehghani, W. Aribowo, H. M. Noori, R. K. Jawad, M. A. Ahmed, I. K. Ibraheem, and K. Eguchi, "Builder Optimization Algorithm: An Effective Human-Inspired Metaheuristic Approach for Solving Optimization Problems", *International Journal of Intelligent Engineering and Systems,* vol. 18, no. 3, pp. 928-937, 2025, doi: 10.22266/ijies2025.0430.62.

[15] T. Hamadneh, B. Batiha, O. Alsayyed, G. Bektemyssova, Z. Montazeri, M. Dehghani, and K. Eguchi, "On the Application of Potter Optimization Algorithm for Solving Supply Chain Management Application", *International Journal of Intelligent Engineering & Systems,* vol. 17, no. 5, pp. 88-99, 2024, doi: 10.22266/ijies2024.1031.09.

[16] T. Hamadneh, B. Batiha, O. Al-Baik, G. Bektemyssova, Z. Montazeri, F. Werner, G. Dhiman, M. Dehghani, and K. Eguchi, "Sales Training Based Optimization: A New Human-inspired Metaheuristic Approach for Supply Chain Management", *International Journal of Intelligent Engineering & Systems,* vol. 17, no. 6, pp. 1325-1334, 2024, doi: 10.22266/ijies2024.1231.96.

[17] T. Hamadneh, B. Batiha, G. M. Gharib, Z. Montazeri, M. Dehghani, W. Aribowo, H. M. Noori, R. K. Jawad, I. K. Ibraheem, and K. Eguchi, "Revolution Optimization Algorithm: A New Human-based Metaheuristic Algorithm for Solving Optimization Problems", *International Journal of Intelligent Engineering and Systems,* Vol. 18, No. 2, pp.

520-531, 2025, doi: 10.22266/ijies2025.0331.38.

[18] S. Alomari, K. Kaabneh, I. AbuFalahah, S. Gochhait, I. Leonova, Z. Montazeri, M. Dehghani, and K. Eguchi, "Carpet Weaver Optimization: A Novel Simple and Effective Human-Inspired Metaheuristic Algorithm", *International Journal of Intelligent Engineering & Systems*, vol. 17, no. 4, pp. 230-242, 2024, doi: 10.22266/ijies2024.0831.18.

[19] T. Hamadneh, B. Batiha, F. Werner, Z. Montazeri, M. Dehghani, G. Bektemyssova, and K. Eguchi, "Fossa Optimization Algorithm: A New Bio-Inspired Metaheuristic Algorithm for Engineering Applications", *International Journal of Intelligent Engineering & Systems*, vol. 17, no. 5, 2024, doi: 10.22266/ijies2024.1031.78.

[20] T. Hamadneh, B. Batiha, O. Alsayyed, Z. Montazeri, H. J. Ashtiani, M. Jafarpour, M. Dehghani, and K. Eguchi, "On the Application of Tailor Optimization Algorithm for Solving Real-World Optimization Application", *International Journal of Intelligent Engineering and Systems*, vol. 18, no. 1, pp. 1-12, 2025, doi: 10.22266/ijies2024.1031.09.

[21] T. Hamadneh, B. Batiha, G. M. Gharib, Z. Montazeri, F. Werner, G. Dhiman, M. Dehghani, R. K. Jawad, E. Aram, I. K. Ibraheem, and K. Eguchi, "Orangutan Optimization Algorithm: An Innovative Bio-Inspired Metaheuristic Approach for Solving Engineering Optimization Problems", *International Journal of Intelligent Engineering and Systems*, vol. 18, no. 1, pp. 47-57, 2025, doi: 10.22266/ijies2025.0229.07.

[22] T. Hamadneh, K. Kaabneh, O. AlSayed, G. Bektemyssova, Z. Montazeri, M. Dehghani, and K. Eguchi, "Sculptor Optimization Algorithm: A New Human-Inspired Metaheuristic Algorithm for Solving Optimization Problems", *International Journal of Intelligent Engineering & Systems*, vol. 17, no. 4, 2024, doi: 10.22266/ijies2024.0831.43.

[23] R. Sarker, and C. Newton, "A genetic algorithm for solving economic lot size scheduling problem", *Computers & Industrial Engineering*, vol. 42, no. 2-4, pp. 189-198, 2002.

[24] H. Ben Ammar, O. Ayadi, and F. Masmoudi, "An effective multi-objective particle swarm optimization for the multi-item capacitated lot-sizing problem with set-up times and backlogging", *Engineering Optimization*, vol. 52, no. 7, pp. 1198-1224, 2020.

[25] R. Pitakaso, C. Almeder, K. F. Doerner, and R. F. Hartl, "A MAX-MIN ant system for unconstrained multi-level lot-sizing problems", *Computers & Operations Research*, vol. 34, no. 9, pp. 2533-2552, 2007.

[26] H. Qawaqneh, "New contraction embedded with simulation function and cyclic (α, β)-admissible in metric-like spaces", *International Journal of Mathematics and Computer Science*, vol. 15, no. 4, pp. 1029-1044, 2020.

[27] T. Hamadneh, N. Athanasopoulos, and M. Ali, "Minimization and positivity of the tensorial rational Bernstein form," *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, Amman, Jordan, 2019, pp. 474-479, doi: 10.1109/JEEIT.2019.8717503.

[28] T. Hamadneh, A. Hioual, O. Alsayyed, Y. A. Al-Khassawneh, A. Al-Husban, and A. Ouannas, "The FitzHugh–Nagumo Model Described by Fractional Difference Equations: Stability and Numerical Simulation", *Axioms*, vol. 12, no. 9, pp. 806, 2023.

[29] T. Hamadneh, M. Ali, and H. AL-Zoubi, "Linear Optimization of Polynomial Rational Functions: Applications for Positivity Analysis", *Mathematics*, vol. 8, no. 2, pp. 283, 2020.

[30] 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, 1997.

[31] D. E. Goldberg, and J. H. Holland, "Genetic Algorithms and Machine Learning", *Machine Learning*, vol. 3, no. 2, pp. 95-99, 1988/10/01, 1988.

[32] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proceedings of ICNN'95 - International Conference on Neural Networks*, Perth, WA, Australia, 1995, pp. 1942-1948 vol.4, doi: 10.1109/ICNN.1995.488968.

[33] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: a gravitational search algorithm", *Information sciences*, vol. 179, no. 13, pp. 2232-2248, 2009.

[34] R. V. Rao, V. J. Savsani, and D. Vakharia, "Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems", *Computer-Aided Design*, vol. 43, no. 3, pp. 303-315, 2011.

[35] 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, 2016.

[36] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey Wolf Optimizer”, *Advances in Engineering Software*, vol. 69, pp. 46-61, 2014/03/01/, 2014.

[37] S. Mirjalili, and A. Lewis, “The whale optimization algorithm”, *Advances in engineering software*, vol. 95, pp. 51-67, 2016.

[38] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, “Marine Predators Algorithm: A nature-inspired metaheuristic”, *Expert Systems with Applications*, vol. 152, pp. 113377, 2020.

[39] 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, pp. 103541, 2020.

[40] L. Abualigah, M. Abd Elaziz, P. Sumari, Z. W. Geem, and A. H. Gandomi, “Reptile Search Algorithm (RSA): A nature-inspired meta-heuristic optimizer”, *Expert Systems with Applications*, vol. 191, pp. 116158, 2022.

[41] 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, pp. 107408, 2021.

[42] M. Braik, A. Hammouri, J. Atwan, M. A. Al-Betar, and M. A. Awadallah, “White Shark Optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems”, *Knowledge-Based Systems*, pp. 108457, 2022.