Applied Mathematics & Information Sciences
*An International Journal*

http://dx.doi.org/10.18576/amis/200114

# A Novel Enhancement of Bird Swarm Algorithm for Efficient Web Service Composition

*Fadl Dahan*

Department of Management Information Systems, College of Business Administration - Hawtat Bani Tamim, Prince Sattam bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia

Received: 1 Oct. 2025, Revised: 21 Nov. 2025, Accepted: 28 Dec. 2025
Published online: 1 Jan. 2026

**Abstract:** We propose an optimized version of Bird Swarm Algorithm (BSA), termed Improved Bird Swarm Algorithm (IBSA) to solve Web Service Composition (WSC) problem. WSC plays a key role in fulfilling complex user requests, especially Quality of Services (QoS) categories. By its nature, WSC is a multi-objective optimization problem and is NP-hard in computing. To deal with these challenging problems, optimization algorithms inspired by swarm intelligence have been investigated. Our proposed algorithm improves the BSA performance using three mechanisms Firstly, the Sine Chaos Theory has been proposed to initiate the population of BSA and to amplify the diversity within the initialization phase. Secondly, a Lévy flight mechanism has been introduced to improve the dual aspects of exploitation and exploration within BSA by preserving the diversity of the avian agents. Furthermore, a neighborhood search mechanism has been employed to mitigate the trade-off between exploration and exploitation within searching methodologies. The efficacy of the proposed approach is assessed utilizing both real and synthetic web service composition datasets (encompassing a cumulative total of 72,000 web services) and compared with standard BSA and three state-of-the-art algorithms, with findings demonstrating its superiority relative to other contemporary methodologies in terms of composition quality and computational efficiency. Consequently, the suggested methodology offers a pragmatic and efficient resolution to the web service composition dilemma, signifying a substantial progression in the domain of service-oriented computing.

**Keywords:** Web Service Composition, Bird Swarm Algorithm, Improved Bird Swarm Algorithm, Service-Oriented Computing

## 1 Introduction

Swarm intelligence (SI) algorithms are being widely acknowledged for their usefulness in solving various optimization problems. Such solutions are based on the idea of multiple agents who can share information and together search for the optimal solution of problems. By exploring and exploiting the solution space, such algorithms try to optimize some fitness function [1]. Several methods of swarm intelligence have been used for solving Web Service Composition (WSC) problems and also like the Bat Algorithm, Artificial Bee Colony (ABC) algorithm. Though these methods offer several benefits, a primary obstacle in their development centers around enhancing the convergence speed (reducing computational time) [2], and the convergence rate, which are associated with solution quality and accuracy they provide while approaching near-optimum solutions. In this work we present an enhanced variant of the Bird Swarm Algorithm (BSA) [3] targeted to overcome issues related to slow convergence rate that affect precision and quality of obtained solutions. To address these problems, the improved algorithm consists of some strategies based on sine chaos theory, Lévy flight mechanism and a neighborhood search strategy, for more effective and accurate searching.

Service-Oriented Architecture (SOA) seeks to gather Web Services (WSs) for addressing complex user request. This architectural style advocates the reuse of pre-existing Web services with composition techniques, such that complex workflows can be realized cost-effectively by combining modular and reusable sub-services [4]. The increasing adoption of Service-Oriented Architecture SOA has made Web Services (WSs) abundant with alternative services offering the same set of functionalities but different in their non-functional aspects, e.g., Quality of Service QoS. Web Service Composition (WSC) has a goal to satisfy customer requirements by assembling many available services in a unified one solution. As

* Corresponding author e-mail: f.naji@psau.edu.sa

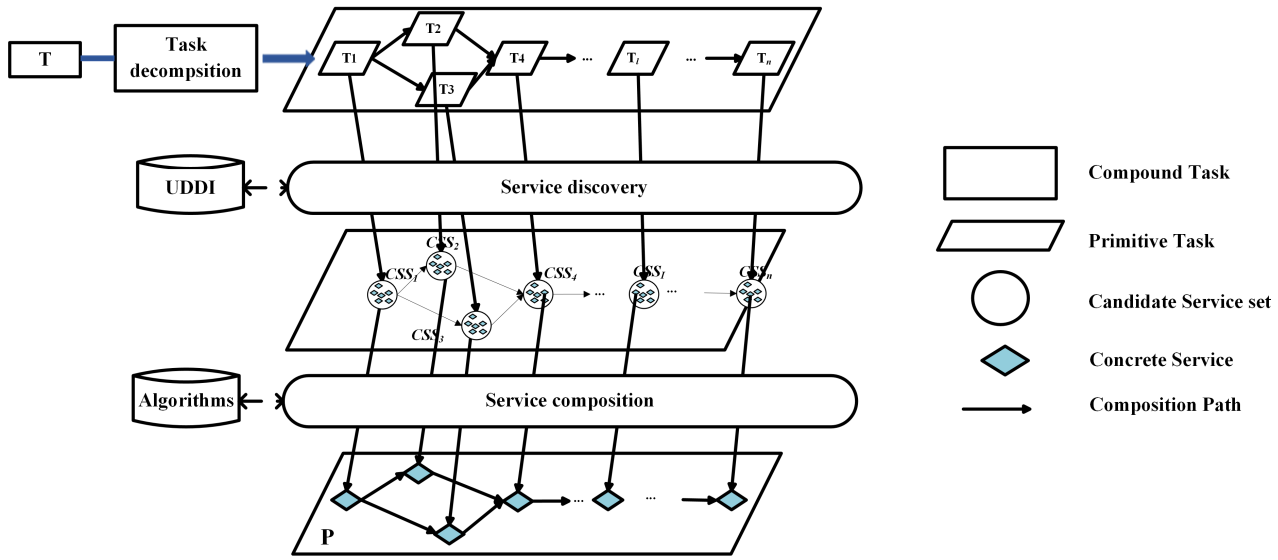


**Fig. 1:** The conceptualization of the WSC problem

functionally similar services could be provided with different QoS characteristics [5], there may exist alternative ways to compose. This combination of services is highly variable, and it is the core of the WSC problem.

One of the key issues to WSC is from the wide variety of service providers in terms of functionally similar web services with different QoS parameters. Choosing the most suitable services that meet user needs, which has certain levels of QoS, is a key feature within the WSC problem. In practice, the user functionality requirement is modelled as a business process that is composed of several tasks to achieve its function. There may be several candidate services for each activity that satisfy the functional requirements, with different associated QoS attributes. If the total number of tasks is n and there are m services that can be used per task, then the number of possible compositions could become mn , which indicates the NP-hard complexity of composition problem [6]. A representative sample of this WSC scenario is sketched in fig. 1. Here, the four QoS (Quality of Service) parameters such as Response Time (RT), Cost (C), Reliability (R) and Throughput (T) are considered.

In the dynamic and heterogeneous digital era of today, a simple one-stop (web) service is no longer enough to cover the wide range of users' needs as it used to be. Consequently, there emerges a need of the discovery of best optimal composition of Web services that can collaborate and cooperate with each other in providing more complex functionality. This tendency has highly increased the demand for developing/composing multiple web services to have effective response to new needs.

We organize the structure of this paper as follows: Section 2 reviews some related literatures. Section 3 gives a summary of the Bird Swarm Algorithm (BSA). The IBSA is described in Section 4, and the developed Modifications are presented in Section 5. Section 5 presents the experiments and an extensive discussion of the results. Finally, the paper is ended by Section 6 with concluding remarks.

## 2 Related Work

Web services as part of service-oriented architectures are state-of-the-art systems in recent years. One of the critical issues in this field is to compose web services effectively to fulfill the user needed functionality. Because of the exponential increase in available service compositions, classical (e.g. brute-force) optimization methods are not suited for solving this problem. To remedy this problem, swarm intelligence-based algorithms have been studied for the web service composition. Swarm intelligence algorithms are a family of metaheuristic optimization techniques, which mimic the collective behavior of natural systems like ant colonies, bird flocks and fish schools. In this paper, we review the literature about the use of swarm intelligence algorithms, such as particle swarm optimization ant colony optimization and bee colony optimization, for solving web service composition problem.

Du and Miao [7] develop an interesting approach for service composition optimization with the hybrid beetle antenna search and ACO algorithm. This hybrid algorithm, called Be-ACO by the authors, achieves a good trade-off between exploration and exploitation in order to produce an evolved solution. A new approach for QoS-aware composition of cloud services was proposed

by Dahan [8] using the multi-agent ant colony optimization. The weakness of the paper is that it lacks a more in-depth comparison with other existing works on QoS-aware cloud service composition. Wang et al. [9] proposed an improved genetic and ant colony optimization algorithm for resource service composition in cloud manufacturing. The purpose of this algorithm is to improve the utilization of resources for cloud manufacturing by finding the most cost-effective service composition with respect to a set of user requests. El-allali et al. [10]proposed a composite model that integrates the ant colony and cooperative agent to explore and conquer proximate-optimal solutions. Zhang et al. [11] improved the ABC through incorporating a move scheme of neighborhood search to enhance the bees' searching capability and opposition-based learning for improving diversity at the initialization. Seghir et al. [12] have improved neighborhood selection with an interval-based approach. An improvement to this approach was proposed in [13], where an exchange operator is introduced to enhance the algorithm. A fuzzy ABC algorithm was presented in [14], which introduced fuzzy distance measures and ranking schemes to maintain the diversity of artificial bees during optimization. Arunachalam and Amuthan [15] presented a rule-based acceptance criterion augmented with incorporated probability to enhance the search mechanism of the ABC algorithm, leading to updating of exploration-exploitation trade off. Chandra and Niyogi [16] have similarly introduced to enhancements through a search operator, which contributes exploration, as well as differential evolution techniques, which contribute increased exploitation. Li et al. [42] also used a genetic algorithm within the ABC algorithm to enhance its search process, which was investigated in another study [18]. Moreover, a hybridization was also presented in [19], which combined ABC with cuckoo search and the cuckoo factor is employed to assist underachieving bees.

Jin et al. [20] proposed an enhanced version of the WOA method called MWOA, which integrates uniform mutation with the WOA. The objective of this integration is to achieve a compromise between exploration and exploitation using the best aspects from both strategies. A modified version of WOA called HWOA was proposed by [21], and employed mutation strategies, a nonlinear convergence factor and chaotic initialization to improve performance. Chen et al. [22] introduced a WOA-based algorithm named iWOA for solving service composition optimization. It uses the AHP and SAW for fitness parameters calculation, as well as tent map and OBL to initialize population. Teng et al. [23] introduced the Logarithmic Energy Particle Swarm Optimizer (LEPSO) algorithm based on a potential energy aggregate and logarithmic convergence factor, which had better optimization efficiency and faster speed of convergent when compared with the other counterparts of swarm Intelligence algorithms. In [24], Kumar Mohit et al. introduced the Fine-tuned Sunflower Whale Optimization Algorithm (FSWOA) as the Qos oriented resource scheduling in order to optimally deploy the microservices for serving both the service provider and user perspectives. Dahan Fadl [25] introduced an IWOA based on solving multi-objective optimization problems with few control parameters and being easy to implement.

A robust prediction model of QoS values was introduced in [26] for web services. Such a method integrates reputation and trust to compute credibility and applies an automatic way of estimating the weights of QoS attributes. In [27], a modified Bat Algorithm was suggested by adding local search routine for enhancing its global efficiency. This method involves the fuzzy logic dependent WSC model called as F3L-WSCM. In [28], a QoS-aware optimization framework was proposed, where both the methodologies and quantification of QoS data were studied. With the help of variable precision rough set, evaluation indexes were mined from history data and weighted by the model, a full range ranking of service quality was achieved. In [29], the combination of FAA and FOA was proposed in cloud computing with locality aware optimization feature.

In [30], Yang et al. during which three enhancements on the Grey Wolf Optimizer were proposed to more effectively balance between exploration and exploitation. These consist of a backward learning mechanism to initialize the population, an improved search step in order to reinforce the exploration capacity of leading wolves and a nonlinear adaptation value searching to improve exploitation stage of its method. hybrid method of using agent-based systems and Particle Swarm Optimization (PSO) utilizing the approach to identify QoS parameters by agent-based and afterwards PSO is applied to choose the most suitable services in [31]. In another work, Ahmed and Majid [32] presented an agent-based methodology which integrated multi-criteria decision-making with Petri net simulation. Bouzary and Chen [33] have suggested a hybrid metaheuristic-based optimization technique by integrating GWO with Genetic Algorithms to reduce the effect of premature convergence in Grey Wolf algorithm by employing crossover and mutation operation. Li et al. [34] proposed a trust-based service composition model that is divided into a three-tier structure, i.e., customers, providers and brokers. Cloud-based composition is a breakthrough toward the evolution of Service Composition within the Manufacturing Cloud.

The aforementioned swarm-based algorithms suffer that they are difficult to make the optimal routing path when routing web services since those swarming mechanisms are random in nature. Furthermore, the No-Free-Lunch (NFL) theorem [35] asserts that in practice no single optimizer can solve well a wide array of optimization problems effectively. Accordingly, these approaches tend to lead to poor performance in handling large-scale datasets. This constraint has led to the search of novel optimization approaches for Web Service Composition (WSC), which involve better fitness in terms

of response time and overall throughput [35]. This has facilitated the generation of new optimization algorithms for WSC meant to cover increased evaluation measures, namely optimized performance and decreasing execution time.

## 3 Bird Swarm Algorithm

Bird Swarm Algorithm (BSA) is one of the new approaches in the family members of swarm intelligence for solving searching and optimization problems. This approach is known as Birds Inspired Optimization based upon behaviors observed in birds, such as foraging, vigilance behaviour and flight. Through the emulation of these natural acts and their social transactions, BSA integrates five basic laws as well as four types foraging behaviors to direct the optimization process [36].

Rule 1: Foraging–vigilance (one forager and one vigilance behavior) When for a bird, the behavior at time is to search for food or watch, with the choice between these being determined by a random mechanism.

Rule 2: Birds re-parametrize their best experiences as individuals and monitor the swarm's collective best direction while foraging. This knowledge also increases foraging efficiency and spreads literally instantly to all members of the group.

Rule 3: When vigilant, birds move to the center of the swarm. Here, the movement is driven by intraspecific competition – birds with more energy are likely to be able to remain closer to the center (where they should feed easily), while those with less energy will mostly likely stick to the periphery.

Rule 4: Birds conditionally change their direction on regular basis, switching between two types of roles depending on the food resources abundant (producers and scroungers). Farmers tend to be species with the lowest fuel reserves, whilst producers are those that carry the most. Birds of intermediate reserves are assigned equiprobable either role.

Rule 5: Producers actively forage for food on their own, while scroungers find food by tracking a random producer.

At each time step $t$, all $N$ virtual birds, with their positions $x_i^t$ in a d-dimensional space, are simulated to fly and search for food. The foraging response in Equation (1) is the mathematical expression of Rule 2.

$$x_{ij}^{(t+1)} = x_{ij}^t + (p_{ij} - x_{ij}^t) \times c \times \text{rand}(0,1) + (g_j - x_{ij}^t) \times s \times \text{rand}(0,1) \tag{1}$$

Here, $j \in [1,\ldots,d]$, where $d$ is the dimensionality of the search space. You can generate sequences of uniformly distributed random numbers between 0 and 1 with the "$rand$"$(0,1)$. Here, the index $i$ runs over birds and $t$ is time during iteration. The constants $c$ and $s$ are positive numbers known as cognitive and social acceleration factors in Equation (1). $x_i j^t$ is the position of the $i-th$ bird in the $j-th$ dimension at time $t$, $p_i j$ is the personal best solution (position), and $g_j$ indicates quasi-best value among all birds.

The vigilance behavior of the birds is controlled by Rule 3, which compensates the movement towards the swarm's center among internal competition. The mathematical description of this dynamic process is given by Equations (2)-(4) [36].

$$x_{ij}^{(t+1)} = x_{ij}^t + A_1 (\text{mean}_j - x_{ij}^t) \,\text{rand}(0,1) + A_2 (p_{kj} - x_{ij}^t) \,\text{rand}(-1,1) \tag{2}$$

$$A_1 = a_1 \times \exp\left(-\frac{p\text{Fit}_i}{(\text{sumFit} + \varepsilon)} \times N\right) \tag{3}$$

$$A_2 = a_2 \times \exp\left(-\frac{(P\text{Fit}_i - P\text{Fit}_k)}{|P\text{Fit}_k - P\text{Fit}_i| + \zeta} \times N\right) \tag{4}$$

Here, $k(wherek \neq i)$ is a positive random integer from 1 to $N$, and constants $a_1$ and $a_2$ are two specific fixed positive numbers in the domain [0,2]. where $PF_(it_i)$ is the best fitness value of the $i-th$ bird, and $sumFit$ is the summation of each bird's best fitness values in particles. To prevent division by zero, a small positive constant $\varepsilon$ is added to the denominator. $mean_j$ is the mean position in the $j-th$ dimension for the entire swarm.

During flight behavior, in line with Rule 4, both producers and scroungers may temporarily leave the swarm. The respective movement strategies for producers and scroungers are described mathematically in Equations (5) and (6) [36].

$$x_{ij}^{(t+1)} = x_{ij}^t + \text{rand}(0,1) \times x_{ij}^t \tag{5}$$

$$x_{ij}^{(t+1)} = x_{ij}^t + (x_{kj}^t - x_{ij}^t) \times FL \times \text{rand}(0,1) \tag{6}$$

Where "$rand$"$(0,1)$ is the Gaussian (normal) random variable with zero mean and variance of 1. The variable $k$ is an integer randomly chosen at random between $[1,2,3,\ldots,N]$, such that $k \neq i$. The parameter $FL$, represents the intensity of scroungers following producers for finding food.

## 4 Improved Bird Swarm Algorithms

This paper is in a sense a sequel to our research (which was started in [25]) The rest of this paper is organized as follows: from section 3 onwards some main contributions of the current work, for an improved BSA-based optimization algorithm to solve WSC problem will be described.

### *4.1 Sine Mapping for Initialization*

Population initialization diversity may have a big impact on the performance and rate of convergence of swarm-based algorithms [37]. This method based on a stochastic approach for population initialization does not guarantee these important dimensions. Sine mapping constitutes a one-dimensional chaotic mapping [38], and various mappings can be employed in optimization algorithms to produce chaotic numbers enumerated in [1]. The Sine map is mathematically articulated as follows:

$$X_{n+1} = \sin\left(\frac{2}{X_n}\right), \qquad n = 0, 1, \ldots, N \tag{7}$$

Where the $X_n$ denotes the current solution and must not equate to 0.

In this study the sine mapping formula has been modified to handle the WSC problem. The proposed new Sine Mapping formula is presented as follows

$$X^b_{i+1,j+1}(t+1) =$$

$$\begin{cases} \sin\left(\dfrac{2}{X_{i,j}}\right) m, & \text{if } 0 < \sin\left(\dfrac{2}{X_{i,j}}\right) m \le m, \\ \text{Randomised method,} & \text{otherwise.} \end{cases} \tag{8}$$

Where $b$ is the $b^{th}$ of the bird for which you want to calculate the fitness or flight length $n$ refers to total number of birds $t$ shows the current iteration $i$ indicates $i^{th}$ task, and j stands for $j^{th}$ WS in the same task. $m$ is the WS size in each task.

This can be allowed for by the formal of the Sine mapping formula if this value is within *0 to m* then we are permitted due to the sine equation to have a web service index, otherwise it will force us identical randomization mechanism which would help in generating our existing position.

### *4.2 Lévy Flight Mechanism*

Researchers in biology have identified that the Lévy flight pattern is a favored foraging strategy among numerous organisms [37]. Consequently, the Lévy flight has been integrated into various heuristic algorithms to address random search challenges with enhanced efficacy. This approach aids heuristic algorithms in circumventing the stagnation issue and augmenting diversity by alternating between high-frequency short-range and low-frequency long-range explorations.

Lévy flight constitutes a specific category of stochastic movement strategy, wherein the statistical distribution governing the movement is characterized by a power function distribution that encapsulates the properties of heavy-tailed probability distributions. The mathematical formulation of Lévy flight can be expressed as follows:

$$L(s, \gamma, \mu) =$$

$$\begin{cases} \sqrt{\dfrac{\gamma}{2\pi}} \exp\left[-\dfrac{\gamma}{2(s-\mu)}\right] \dfrac{1}{(s-\mu)^{3/3}}, & 0 < \mu < s < \infty, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

In this equation, $s$ denotes the step size, $\mu$ represents the minimum step size, $\gamma$ represents the scale parameter.

When $s \to \infty$ the Equation 9 can be reformulated as:

$$L(s, \gamma, \mu) \approx \sqrt{\frac{\gamma}{2\pi}} \frac{1}{s^{3'/3}} \tag{10}$$

The magnitude of the Lévy flight step size can be derived utilizing the following expression:

$$s = \frac{u}{|v|^{1/\beta}} \tag{11}$$

In this context, $\beta$ signifies a random variable constrained between 0 and 2, while $u$ and $v$ are random variables that exhibit a normal distribution, defined as follows:

$$\begin{cases} u \sim N(0, \sigma_u^2) \\ v \sim N(0, \sigma_v^2) \end{cases} \tag{12}$$

Where $\sigma_u$ and $\sigma_v$ are defined as:

$$\begin{cases} \sigma_u = \left( \dfrac{\Gamma(1+\beta) \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \beta\, 2^{(\beta-1)/2}} \right)^{1/\beta}, \\ \sigma_v = 1 \end{cases} \tag{13}$$

Where $\Gamma$ is the gamma function.

In the present study, the Lévy-flight model is integrated into the BSA to enhance the localization of birds and to optimize the efficacy of target searching.

$$X^b_{i+1,j+1}(t+1) = X^b_{i,j}(t) + (2r-1) \oplus \left( X^b_{i,j}(t) + r \oplus s \right) \tag{14}$$

Where $r$ signifies a stochastic variable confined within the interval, $\oplus$ is the scalar product of the elements. $b$ denotes the $b^{th}$ bird, $t$ indicates the iteration count, $i$ represents the $i^{th}$ task, and $j$ corresponds to the $j^{th}$ web service associated with the same task.

The implementation of Lévy-flight mechanisms, characterized by stochastic searches across diverse ranges, facilitates the ability of the BSA to extricate itself from local optima. Concurrently, this approach ensures an optimal equilibrium between the exploratory and exploitative components, thereby enhancing the efficacy of the proposed algorithm.

### 4.3 Neighborhood Search Strategy

This technique provides an adaptation of the neighborhood search approach described in our earlier work. The adaptive search policy developed here successfully balances the diversity and concentration in swarm-based algorithms [27], [19] . At the same time, it is also a powerful technique to escape local optima in the optimization process [39]. In [27], [19], the neighborhood search was performed around the best WSs. This method similarly increases time complexity according to the number of neighboring WSs. This process is enhanced in the context of IBSA to compensate for inadequacies within specific time-frame:

First: At each iteration, a test is performed on the quality of the local best solution for possible enhancement; if there is improvement in the solution quality, the neighborhood search will start; otherwise, Iterative IBSA process continues.

Second: a random number (*RN*) ranging $1 \leq RN \leq B\text{-}1$ (where $B$ is referred the size of population) which selects whales on its family group level.

Third: For every bird $i$ in population, chooses a task $j$ randomly, save this value for remember task selection processes and mutation based on best iteration solution will be applied for $WS_j$.

## 5 Experimental Settings and Results Discussion

Our evaluations compare the performance of IBSA algorithm with respect to traditional BSA and state-of-the-arts though systematic comparison study. The main objective of this analysis is to offer experimental evidence to justify the better performance of the IBSA approach across different experimentation conditions. The details of the experimental configurations and the resultant findings will be presented in what follows

### 5.1 Experimental Settings

This study utilized two separate datasets to confirm the competitive efficacy and demonstrate superior performance of IBSA over other approaches. The base dataset was drawn from the real-world sources, i.e., QWS 2.0 [40] a well-known dataset containing 2507 WSs by utilizing the real or authentic quality of service (QoS) metrics. This dataset contains smaller-sized corpora to keep task size consistent and vary the number of WSs, of them in all (selected randomly with replacement).

The second dataset was randomly created using the method detailed in [41] with values of QoS constraint between 1 and 1000. This particular dataset is moderate as well as large-sized datasets (keeping a fixed number of WSs) by varying the diversity in services provided by individual tasks resulting an overall 72,000 WSs. Table 1 presents corresponded summarization that has a large amount of data about the datasets and also will give details on how many tasks to how many WSs each task was assigned. It is important to note that the current work considers four different QoS bounds.

**Table 1:** Summary of dataset characteristics.

| Size | Dataset | No. Tasks | No. WSs/task |
|---|---|---|---|
| Small | *DS*1 | 10 | 200 |
| | *DS*2 | 10 | 400 |
| | *DS*3 | 10 | 600 |
| | *DS*4 | 10 | 800 |
| Medium | *DS*5 | 30 | 100 |
| | *DS*6 | 40 | 100 |
| | *DS*7 | 50 | 100 |
| | *DS*8 | 60 | 100 |
| Large | *DS*9 | 70 | 100 |
| | *DS*10 | 80 | 100 |
| | *DS*11 | 90 | 100 |
| | *DS*12 | 100 | 100 |

Performance is measured based on throughput (T), reliability (R), cost (C) and response time (RT). Where the objective is to maximize the values of throughput and reliability, in other hand we aim to minimize the values of cost and response time. These performance metrics are well used for performance evaluation. Equation 15 describes the mathematical set up that was used in order to evaluate each candidate solution given these constraints, these cases include:

$$F_i = \left( \prod_{j=1}^{n} T_{jb} + \prod_{j=1}^{n} R_{jb} - \sum_{j=1}^{n} C_{jb} - \sum_{j=1}^{n} (RT)_{jb} \right) \quad (15)$$

In the equation, $F$ is the fitness of $i^{th}$ based on proposed solution (i.e., scheduling), $n$ is the total number of tasks, and $x$ stands for web services.

All the algorithms were coded using Java running on an operation platform with uniform hardware configuration (Intel(R) Core(TM) i7-1355U CPU at 1.70GHz and 16.0 GB RAM). Effectiveness comparison We also compared the efficiency of the IBSA with that of the Standard BSA, a number of recent swarm-based algorithms BSA [36], MWOA [20], LEWOA [23], and HWOA [21].

To be consistent with experiment, the IBSA parameters were set to be identical BSA values whereas the LEWOA, MWOA and HWOA were all rescaled according to literature containing their respective information. In order to achieve uniformity among the algorithms, we normalised universal parameters using maximum iteration (Z) of 500 including a population size (P) of 100.

Moreover, the performance evaluation criteria used were the best fitness values (BFV), average execution time (AET) and mean value of the fitness function and its

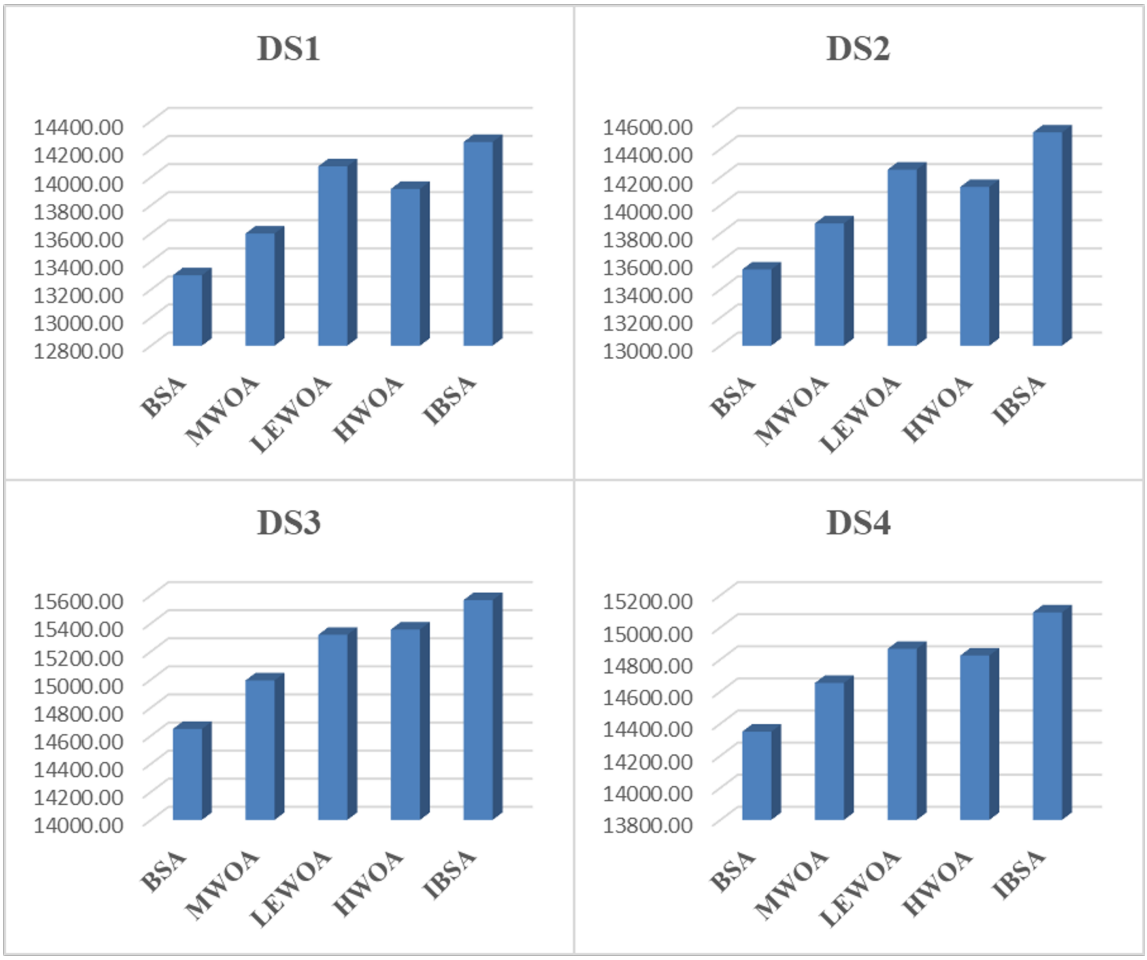


**Fig. 2:** Small size datasets representation of the average fitness value for all algorithms

standard deviation (STD) for each algorithm. Each experiment was independently conducted 30 times for all the datasets. The average fitness was obtained at every generation, and the time taken to determine best fitness values were noted.

## *5.2 Experimental Results and Analysis*

The performance of the IBSA algorithm is reported in this section, where three other methods are involved. fig. 2 through fig. 4 show the average fitness value of all algorithms. fig. 2 in particular explores scenarios where workflows are of size 10 and each task can be executed by either 200, 400, 600 or 800 candidate services. fig. 3 and fig. 4 extend this assessment to from 30 to 100 abstract tasks (in steps of ten) for workflows with each task having up to 100 candidate services. The values are presented in Table 2, which shows the results for all algorithms using three metrics: BFV, STD and AET. The entries of the highest performance values for each class are boldfaced in the table. We did two sets of experiments to observe the performance of IBSA related to local exploitation and global search against other algorithms. The first set was based on small-scale data to demonstrate the strength of IBSA in local exploitation, quantifying it over smaller search spaces and maintaining a fixed number of tasks size [42]. The second pair, on the contrary, was used to test IBSA in global search: those datasets were from moderate or relatively large size. This choice was motivated by the wide range of services and the large search space that characterizes these bigger datasets [42].

### 5.2.1 Local Exploitation Validation Experiments

We conducted these experiments to evaluate the efficiency of IBSA in promoting local exploitation, especially when dealing small-sized datasets. The average fitness values of 30 independent runs are shown in fig. 4 for each algorithm. It is crucial to observe that IBSA is better than all other algorithms on average fitness for all datasets.

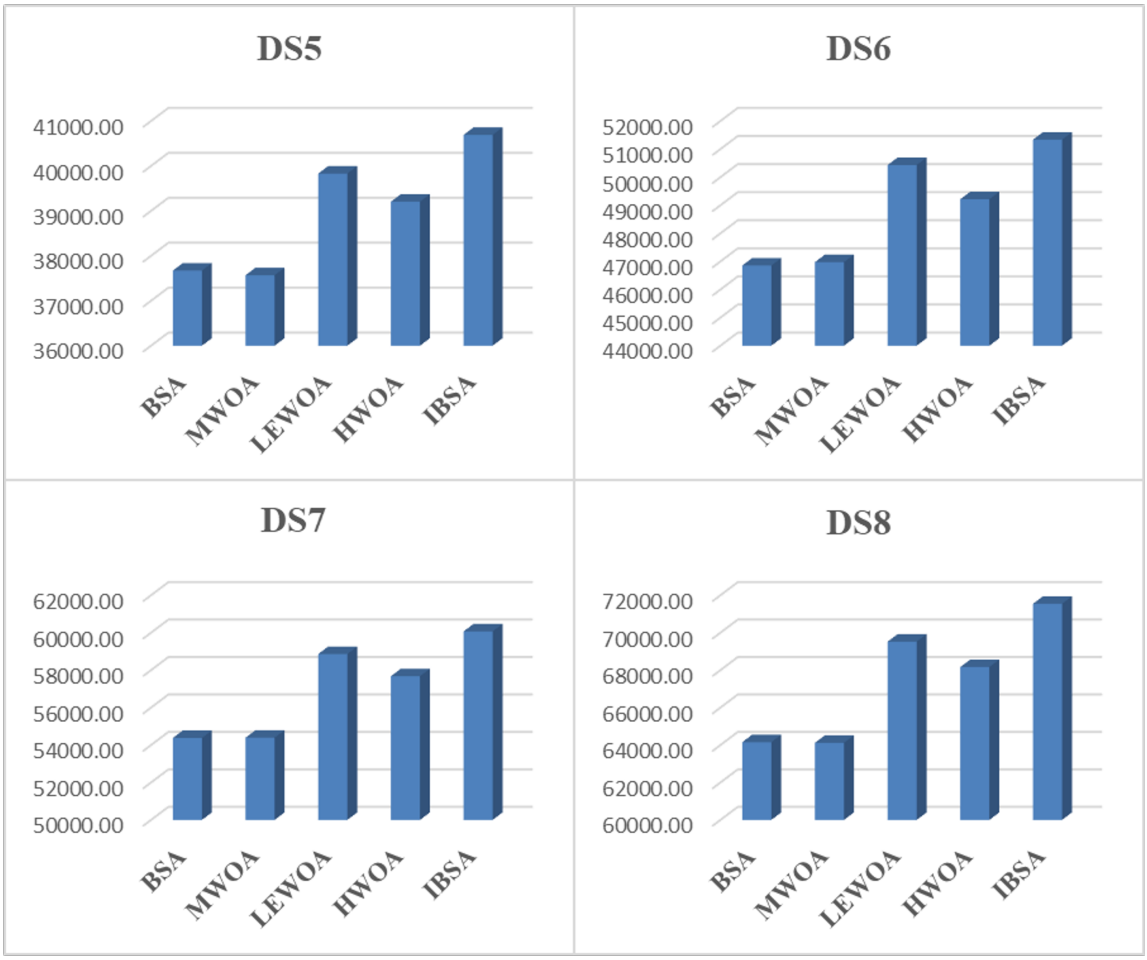


**Fig. 3:** Medium size datasets representation of the average fitness value for all algorithms

Such a persistent superiority indicates that IBSA might have better capability in fine tuning solutions over local search space, hence resulting in more optimal results. Higher mean fitness's manifest that peak search is not random, but well established and steady process. This result can be explained by the good tradeoff between exploration and exploitation of IBSA to prevent premature convergence yet focusing on promising areas of search space.

Table 2 apparently shows that the IBSA algorithm is more competitive and robust in comparison with other models for all small datasets. IBSA generally resulted to better best fitness values and lower standard deviations compared with its competitors, with the exception of the standard deviation value for DS2. These results emphasis the superiority of IBSA regarding convergence and its stability in keeping solution quality constant from one run to another. From a computational time perspective, BSA also had the smallest average running time and was followed by MWOA and HWOA; however, IBSA ranked fourth. This balance between increased computational cost and improved solution quality clearly highlights the operational efficacy of IBSA, on which we can rely suitably when it is more important to have high accuracy/results than fast execution.

### 5.2.2 Global Search Validation Experiments

These experiments were performed to analyses the global search properties of the IBSA algorithm on medium to large datasets, and are depicted in fig. 3 and fig. 4. These results are showing the average fitness values for each algorithm over 30 independent runs. We notice that, in practice, IBSA consistently gets better results than the other two solutions when we test with both scales of data sets. Additionally, the solutions found by IBSA are consistent rather well as is reflected in a tight distribution of fitness values.

As can be seen from Table 2, in the medium- and large-scaled datasets, IBSA algorithm presents a better

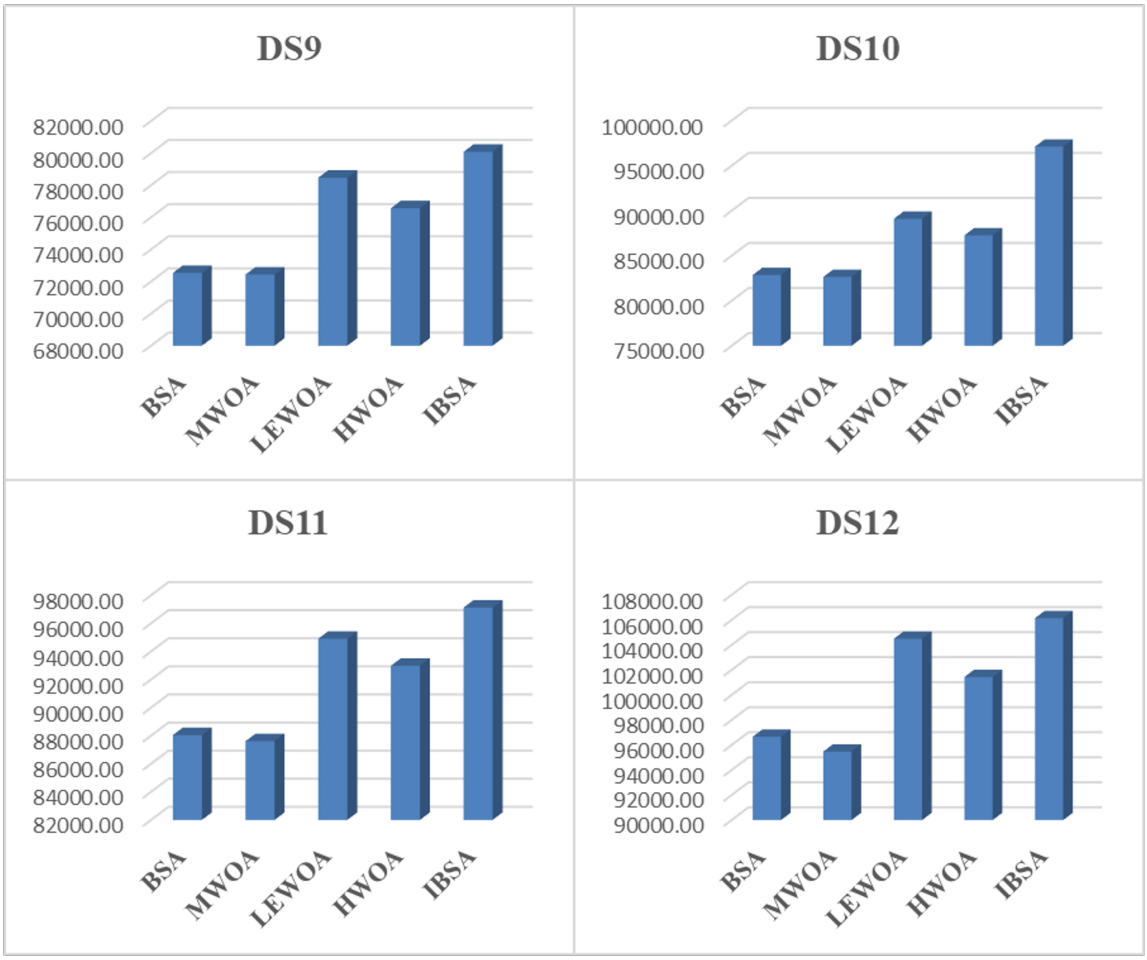


**Fig. 4:** Large size datasets representation of the average fitness value for all algorithms

best fitness value and standard deviation than other algorithms with little exceptions for the improvement in DS6 and DS12. In general, IBSA is better than the compared algorithms which demonstrate its power of optimization and steady convergence process. These observations indicate that IBSA is really competitive and robust toward the handling of harder problem instances. Since IBSA has the fourth average execution times (except BSA, MWOA, and HWOA), such trade-off is tolerable with respect to its superior performance. The little extra load overhead can be well compensated by incorporating the proposed schemes due to its better solution quality and search stability.

### 5.2.3 Wilcoxon Rank-Sum Test Analysis

Optimization-based studies use an algorithm whose performance may fluctuate owing to inherent randomness. To test the differences of these fluctuations, we use Wilcoxon signed-rank test to compare IBSA with other algorithms. For both datasets, 30 independent runs are performed and their associated p-values are calculated using Wilcoxon two-sample rank-sum test at significance level of 5%. If the $p$-value is less than 0.05, it means that IBSA average fitness value significantly differs with the compared algorithm. The statistical analysis results of the Wilcoxon signed-rank test for small-, medium– and large–scale datasets are shown in Table 3, IBSA with other methods. In this table "–" stands for IBSA performs significantly different from the competitor; "+" is used when the performance of the competitor is significantly different comparing to IBSA and there is no significant difference (=) when very close convergence results obtained by both methods.

For the small datasets, as given in Table 3, differences of IBSA with respect to the MWOA, LEWO and BSA are statistically significant in all datasets (except for DS2 where LEWOA is statistically significant). For the medium and large datasets, differences of IBSA with respect to the MWOA, LEWO and BSA are statistically

**Table 2:** : BFV, STD, and AET values for all algorithms.

| Dataset | Evaluation | BSA | MWOA | LEWOA | HWOA | IBSA |
|---|---|---|---|---|---|---|
| DS1 | *BFV* | 13,693.39 | 13,987.59 | 14,307.40 | 14,221.22 | 14,307.40 |
| | *STD* | 240.03 | 189.56 | 127.66 | 161.98 | 76.42 |
| | *AET* | 30.63 | 35.77 | 42.67 | 233.67 | 82.62 |
| DS2 | *BFV* | 14,068.79 | 14,513.93 | 14,548.65 | 14,529.89 | 14,841.17 |
| | *STD* | 262.88 | 226.93 | 148.27 | 162.75 | 155.94 |
| | *AET* | 29.17 | 36.43 | 44.30 | 470.73 | 95.30 |
| DS3 | *BFV* | 14,953.94 | 15,408.35 | 15,829.33 | 15,943.02 | 16,031.64 |
| | *STD* | 223.34 | 217.31 | 220.69 | 242.33 | 196.95 |
| | *AET* | 29.53 | 38.63 | 51.27 | 701.47 | 90.49 |
| DS4 | *BFV* | 14,898.23 | 14,988.04 | 15,266.71 | 15,317.69 | 15,482.66 |
| | *STD* | 297.36 | 157.65 | 150.78 | 188.29 | 134.16 |
| | *AET* | 30.87 | 34.53 | 47.20 | 910.13 | 131.17 |
| DS5 | *BFV* | 38,607.81 | 38,953.02 | 40,695.91 | 40,052.55 | 41,635.39 |
| | *STD* | 518.46 | 680.46 | 383.31 | 417.88 | 363.53 |
| | *AET* | 67.83 | 66.37 | 74.93 | 206.00 | 131.23 |
| DS6 | *BFV* | 47,696.07 | 48,415.68 | 52,108.82 | 51,218.96 | 53,292.01 |
| | *STD* | 502.89 | 662.76 | 721.92 | 704.85 | 764.92 |
| | *AET* | 83.47 | 75.77 | 95.53 | 225.53 | 151.69 |
| DS7 | *BFV* | 56,709.17 | 56,101.42 | 60,199.06 | 59,613.38 | 63,541.84 |
| | *STD* | 931.82 | 1,017.30 | 728.03 | 699.56 | 587.96 |
| | *AET* | 93.03 | 91.83 | 115.50 | 256.87 | 269.97 |
| DS8 | *BFV* | 66,358.39 | 66,431.15 | 71,479.49 | 70,993.59 | 73,239.07 |
| | *STD* | 971.48 | 1,011.98 | 930.94 | 1,231.58 | 679.52 |
| | *AET* | 138.50 | 113.20 | 138.47 | 286.67 | 121.16 |
| DS9 | *BFV* | 75,683.29 | 74,062.45 | 80,701.89 | 78,680.82 | 81,895.96 |
| | *STD* | 1,498.37 | 939.15 | 1,132.73 | 1,349.60 | 929.75 |
| | *AET* | 122.40 | 119.22 | 151.15 | 312.00 | 196.32 |
| DS10 | *BFV* | 84,712.80 | 86,189.22 | 92,080.27 | 93,034.48 | 93,272.12 |
| | *STD* | 1,219.82 | 1,592.84 | 1,524.44 | 2,056.95 | 1,209.02 |
| | *AET* | 149.00 | 135.55 | 171.96 | 440.94 | 235.85 |
| DS11 | *BFV* | 90,997.58 | 90,301.45 | 98,802.44 | 97,056.98 | 100,085.85 |
| | *STD* | 1,473.70 | 1,277.71 | 1,473.71 | 1,857.72 | 1,244.68 |
| | *AET* | 155.39 | 163.59 | 186.32 | 384.44 | 235.24 |
| DS12 | *BFV* | 99,726.74 | 98,506.73 | 109,208.03 | 105,201.62 | 110,229.42 |
| | *STD* | 1,738.10 | 1,460.46 | 1,774.66 | 1,917.87 | 1,480.13 |
| | *AET* | 172.89 | 178.62 | 229.70 | 443.44 | 259.39 |

significant in all datasets (except for DS12 where LEWOA is statistically significant).

In general, the experimental results indicate that IBSA enjoys better performance, which demonstrates that the designed improvements are effective. The enhancements effectively overcome the exploration limitation of the conventional BSA and achieve a considerable equilibrium between exploration and exploitation. Hence the improved IBSA algorithm always shows better performance with different problem sizes.

**Table 3:** : IBSA compared to other algorithms obtained from Wilcoxon's rank-sum test.

| Dataset | | BSA | MWOA | LEWOA | HWOA |
|---|---|---|---|---|---|
| Small datasets | - | 4 | 4 | 3 | 4 |
| | + | 0 | 0 | 1 | 0 |
| | = | 0 | 0 | 0 | 0 |
| Medium datasets | - | 4 | 4 | 4 | 4 |
| | + | 0 | 0 | 0 | 0 |
| | = | 0 | 0 | 0 | 0 |
| Large datasets | - | 4 | 4 | 3 | 4 |
| | + | 0 | 0 | 1 | 0 |
| | = | 0 | 0 | 0 | 0 |

## 6 Conclusion and Future Research

An improved bat swarm algorithm (IBSA) was introduced in this paper for the WSC problem to alleviate some constraints of the original BSA, such as low precision of solution and local optimum trap. The improved IBSA includes three new mechanisms: Sine chaotic mapping is used to improve the population initialization, a Lévy

flight algorithm is used to compromise exploration and exploitation, and the neighborhood search method is used as a heuristic technique for local search. To verify the effectiveness of IBSA, it was applied to 12 benchmark data sets and compared with standard BSA and three classical optimization algorithms under the same experimental conditions. Results clearly show that IBSA outperforms state-of-the-art in optimization of the models on various problem instances, furthermore, offering more accurate and stable solutions. These results reaffirm the stability of the algorithm, its stronger global exploring capability and competitive performance when tackling challenging optimization tasks.

Building on its strong performance in the WSC problem, future work will focus on extending IBSA to large-scale and real-world optimization scenarios, particularly those characterized by high dimensionality and complex constraints. In addition, the proposed improvement strategies will be systematically integrated into selected swarm intelligence algorithms to evaluate their effectiveness in mitigating premature convergence. Further investigations will also examine hybrid IBSA-based frameworks within other metaheuristic paradigms to assess scalability and robustness.

## Acknowledgement

The authors extend their appreciation to Prince Sattam bin Abdulaziz University for funding this research work through the project number (PSAU/ 2025/01/34076)

## References

[1] E. Varol Altay and B. Alatas, "Bird swarm algorithms with chaotic mapping," *Artificial Intelligence Review*, vol. 53, no. 2, pp. 1373–1414, 2020.

[2] V. Guliashki, L. Kirilov, and G. Marinova, "Methods and Algorithms for Flexible Job Shop Scheduling A State of the Art," *Cybernetics and Information Technologies*, vol. 25, no. 2, pp. 3–30, 2025.

[3] X.-B. Meng, X. Z. Gao, Y. Liu, and H. Zhang, "A novel bat algorithm with habitat selection and Doppler effect in echoes for optimization," *Expert Systems with Applications*, vol. 42, no. 17–18, pp. 6350–6364, 2015.

[4] P. Podili, K. K. Pattanaik, and P. S. Rana, "BAT and hybrid BAT meta-heuristic for quality of service-based web service selection," *Journal of Intelligent Systems*, vol. 26, no. 1, pp. 123–137, 2017. doi: 10.1515/jisys-2015-0032.

[5] M. J. A. Alsammarraie, H. M. Noman, A. H. H. Hussein, and A. A. Abdulrazzaq, "An Analytical Study to Justify the Transformation from Traditional to Software-Defined Network in Terms of QoS Parameters," *Cybernetics and Information Technologies*, vol. 25, no. 2, 2025.

[6] G. Canfora, M. Di Penta, R. Esposito, and M. L. Villani, "A lightweight approach for QoS-aware service composition," in *Proceedings of the 2nd International Conference on Service Oriented Computing (ICSOC'04) – Short Papers*, Citeseer, 2004, pp. 1–2.

[7] Z. Du and H. Miao, "An Optimization Method Based on Be-ACO Algorithm in Service Composition Context," *Computational Intelligence and Neuroscience*, vol. 2022, 2022.

[8] F. Dahan, "An effective multi-agent ant colony optimization algorithm for QoS-aware cloud service composition," *IEEE Access*, vol. 9, pp. 17196–17207, 2021.

[9] Z. Wang, "Optimization of resource service composition in cloud manufacture based on improved genetic and ant colony algorithm," in *Smart Innovation, Systems and Technologies*. Springer, 2022, pp. 183–198.

[10] N. El Allali, M. Fariss, H. Asaidi, and M. Bellouki, "Semantic web services composition model using ant colony optimization," in *2020 Fourth International Conference on Intelligent Computing in Data Sciences (ICDS)*. IEEE, 2024, pp. 1–5.

[11] S. Zhang, Y. Shao, and L. Zhou, "Optimized artificial bee colony algorithm for web service composition problem," *International Journal of Machine Learning and Computing*, vol. 11, no. 5, 2021.

[12] F. Seghir, A. Khababa, and F. Semchedine, "An interval-based multi-objective artificial bee colony algorithm for solving the web service composition under uncertain QoS," *Journal of Supercomputing*, vol. 75, no. 9, pp. 5622–5666, 2019.

[13] F. Dahan, H. Mathkour, and M. Arafah, "Two-step artificial bee colony algorithm enhancement for QoS-aware Web service selection problem," *IEEE Access*, vol. 7, pp. 21787–21794, 2019.

[14] F. Seghir, "FDMOABC: fuzzy discrete multi-objective artificial bee colony approach for solving the non-deterministic QoS-driven web service composition problem," *Expert Systems with Applications*, vol. 167, p. 114413, 2021.

[15] N. Arunachalam and A. Amuthan, "Integrated probability multi-search and solution acceptance rule-based artificial bee colony optimization scheme for web service composition," *Natural Computing*, vol. 20, no. 1, pp. 23–38, 2021.

[16] M. Chandra and R. Niyogi, "Web service selection using modified artificial bee colony algorithm," *IEEE Access*, vol. 7, pp. 88673–88684, 2019.

[17] T. Li, Y. Yin, B. Yang, J. Hou, and K. Zhou, "A self-learning bee colony and genetic algorithm hybrid for cloud manufacturing services," *Computing*, vol. 104, pp. 1–27, 2022.

[18] P. Karthikeyan and G. Preethi, "Artificial bee colony and genetic algorithms in selecting and combining web services for enhancing QoS," *Design Engineering*, vol. 2021, pp. 6009–6021, 2021.

[19] F. Dahan and A. Alwabel, "Artificial bee colony with cuckoo search for solving service composition," *Intelligent Automation & Soft Computing*, vol. 35, no. 3, pp. 3385–3402, 2023, doi: 10.32604/iasc.2023.030651.

[20] H. Jin, S. Lv, Z. Yang, and Y. Liu, "Eagle strategy using uniform mutation and modified whale optimization algorithm for QoS-aware cloud service composition," *Applied Soft Computing*, vol. 114, p. 108053, 2022.

[21] C. Ju, H. Ding, and B. Hu, "A hybrid strategy improved whale optimization algorithm for web service composition," *The Computer Journal*, vol. 66, no. 3, pp. 662–677, 2023.

[22] Y. Ye, S. Chen, K. Cheng, and H. Zhang, "A Web Service composition Method Based on Improved Whale Optimization

Algorithm," in *2022 IEEE 12th International Conference on Electronics Information and Emergency Communication (ICEIEC)*. IEEE, 2022, pp. 85–88.
[23] X. Teng, Y. Luo, T. Zheng, and X. Zhang, "An improved whale optimization algorithm based on aggregation potential energy for QoS-driven web service composition," *Wireless Communications and Mobile Computing*, vol. 2022, 2022.
[24] M. Kumar, J. K. Samriya, K. Dubey, and S. S. Gill, "QoS-aware resource scheduling using whale optimization algorithm for microservice applications," *Software: Practice and Experience*, vol. 54, no. 4, 2024, doi: 10.1002/spe.3211.
[25] F. Dahan, "An improved whale optimization algorithm for web service composition," *Axioms*, vol. 11, no. 12, p. 725, 2022.
[26] S. Subbulakshmi, K. Ramar, A. E. Saji, and G. Chandran, "Optimized web service composition using evolutionary computation techniques," in *Intelligent Data Communication Technologies and Internet of Things: Proceedings of ICICI 2020*. Springer Singapore, 2021, pp. 457–470.
[27] F. Dahan, "Neighborhood search based improved bat algorithm for web service composition," *Computer Systems Science and Engineering*, vol. 45, no. 2, pp. 1343–1356, 2023, doi: 10.32604/csse.2023.031142.
[28] W. Ma, Y. Xu, J. Zheng, and S. U. Rehman, "QoS-Aware Cloud Service Optimization Algorithm in Cloud Manufacturing Environment," *Intelligent Automation and Soft Computing*, vol. 37, no. 2, 2023, doi: 10.32604/iasc.2023.030484.
[29] P. Rajeswari and K. Jayashree, "Hybrid metaheuristics web service composition model for QoS aware services," *Computer Systems Science and Engineering*, vol. 41, no. 2, 2022, doi: 10.32604/csse.2022.020352.
[30] Y. Yang, B. Yang, S. Wang, T. Jin, and S. Li, "An enhanced multi-objective grey wolf optimizer for service composition in cloud manufacturing," *Applied Soft Computing*, vol. 87, p. 106003, 2020.
[31] A. Naseri and N. J. Navimipour, "A new agent-based method for QoS-aware cloud service composition using particle swarm optimization algorithm," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 5, pp. 1851–1864, 2019.
[32] F. D. Ahmed and M. A. Majid, "Towards agent-based petri net decision making modelling for cloud service composition: A literature survey," *Journal of Network and Computer Applications*, vol. 130, pp. 14–38, 2019.
[33] H. Bouzary and F. F. Chen, "A hybrid grey wolf optimizer algorithm with evolutionary operators for optimal QoS-aware service composition and optimal selection in cloud manufacturing," *The International Journal of Advanced Manufacturing Technology*, vol. 101, no. 9–12, pp. 2771–2784, 2019.
[34] W. Li, J. Cao, K. Hu, J. Xu, and R. Buyya, "A trust-based agent learning model for service composition in mobile cloud computing environments," *IEEE Access*, vol. 7, pp. 34207–34226, 2019.
[35] 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.
[36] X.-B. Meng, X. Z. Gao, L. Lu, Y. Liu, and H. Zhang, "A new bio-inspired optimisation algorithm: Bird Swarm Algorithm," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 28, no. 4, pp. 673–687, 2016.
[37] W. Yang *et al.*, "A multi-strategy Whale optimization algorithm and its application," *Engineering Applications of Artificial Intelligence*, vol. 108, p. 104558, 2022.
[38] L. Liu and R. Zhang, "Multistrategy Improved Whale Optimization Algorithm and Its Application," *Computational Intelligence and Neuroscience*, vol. 2022, 2022.
[39] A. Kaur and Y. Kumar, "Neighborhood search based improved bat algorithm for data clustering," *Applied Intelligence*, vol. 52, pp. 1–35, 2022.
[40] E. Al-Masri and Q. H. Mahmoud, "Discovering the best web service," in *Proceedings of the 16th International Conference on World Wide Web*, 2007, pp. 1257–1258.
[41] X. Wang, Z. Wang, and X. Xu, "An improved artificial bee colony approach to QoS-aware service selection," in *Proceedings of the IEEE 20th International Conference on Web Services (ICWS 2013)*, 2013, pp. 395–402.
[42] J. Li, H. Ren, C. Li, and H. Chen, "A novel and efficient salp swarm algorithm for large-scale QoS-aware service composition selection," *Computing*, vol. 104, no. 1, pp. 1–21, 2022.

**Fadl Dahan** Associated Professor of Artificial Intelligence, Department of Management Information System, College of Business Administration - Hawtat Bani Tamim, Prince Sattam Bin Abdulaziz University, Al-Kharj, Saudi Arabia. Email: f.naji@psau.edu.sa. B.Sc: Thamar University, Yemen. M.Sc: King Saud University, Ph.D: Department of Computer Science, King Saud University. He is also a Faculty Member with the Department of Computer Science, Faculty of Computer Science, Taiz University, Taiz, Yemen. His research interests include optimization and swarm intelligence.