# An Enhanced Whale Optimization Algorithm Based on Fibonacci Search Principle for Service Composition in the Internet of Things

Yun CUI
Hebei Chemical & Pharmaceutical College, Shi Jiazhuang 050026, China

***Abstract*—Service composition in the Internet of Things (IoT) poses significant challenges owing to the dynamics in IoT ecosystems and the exponential increase in service candidates. This paper proposes an Enhanced Whale Optimization Algorithm (EWOA) by introducing the Fibonacci search principle for service composition optimization to overcome certain shortcomings of conventional approaches, including slow convergence and being stuck in local optima, in addition to imbalanced exploration-exploitation trade-offs. The proposed EWOA combines the application of nonlinear crossover weights with a Fibonacci search to optimize the global exploration and local exploitation searches of the basic version, thereby producing a better solution. Several simulations were performed for IoT functions. Among the experiments involving different QOS-based service compositions, the results show that the EWOA achieves superior and faster convergence capability with enhanced convergence compared to recent methods.**

***Keywords*—Service composition; Internet of Things; quality of service; whale optimization; Fibonacci search**

## I. Introduction

The Internet of Things (IoT) is driving a revolution in modern technologies, enabling the world to grow further connected, and billions of devices can communicate and share data effectively [1]. This rapid growth of services has led to a vast repository of functionalities that address different application areas [2]. However, this immense size and dynamism introduce a severe challenge: efficiently composing multiple IoT services to satisfy user demands [3]. Efficient service composition guarantees the realization of desired functionalities and optimizes the quality attributes of execution time, reliability, and cost in dynamic IoT ecosystems [4].

Traditional optimization methods and swarm-intelligence-based algorithms offer great promise for solving service composition problems [5]. However, these methods typically incur severe limitations due to slow convergence rates, premature stagnation at local optima, or failure to balance exploration and exploitation efficiently [6]. These challenges impede efficient solutions, particularly for large-scale and complex IoT service composition problems; hence, advanced approaches are required to ensure their robustness and scalability [7].

This paper presents an improved QoS-based service composition using a new metaheuristic approach. The contribution of this work is to improve the solution accuracy, accelerate convergence, and overcome local optima entrapment by addressing the limitations of existing algorithms. Thus, this work aims to find a globally optimal balance between local exploitation and global exploration and ensure a more effective and efficient service composition in dynamic IoT environments.

These include the development of the Fibonacci search principle to improve the performance of the WOA on global optimization problems and applying nonlinear weights in the WOA to maintain balance within its processes. In general, this contributes to reaching the optimality of a solution by increasing exploration during the initialization stage, thus creating faster convergence, whereas adopting this modified WOA to tackle the optimal solution in IoT service composition. Intensive validation showed the excellent performance of this algorithm owing to better convergence speed and stability in results, hence providing a concrete base for more challenging tasks within IoT service optimization processes.

The remainder of this paper is organized as follows: Section II presents a comprehensive literature review, discussing previous studies and existing optimization techniques for IoT service composition. Section III details the proposed algorithm. Section IV describes the experimental setup and presents the simulation results. Section V provides a critical discussion and comparative analysis of the findings. Finally, Section VI concludes the study, summarizing key contributions and outlining potential future research directions.

## II. Literature Review

The authors in study [8] propose an Artificial Neural Network-based Particle Swarm Optimization (ANN-PSO) hybrid algorithm for cloud-edge computing to improve QoS factors. They utilized a formal verification process through functional transitions and constraint logic to ensure correctness regarding functional and non-functional aspects. Their approach shows enhanced memory, response time, availability, and price, yielding higher fitness values than others.

In study [9], a Hidden Markov Model (HMM) integrated with Ant Colony Optimization (ACO) was suggested for IoT service composition. HMM was trained for QoS prediction, and the Viterbi approach enhanced the emissions and transitions. The ACO algorithm identified optimal service paths, achieving better response time, reliability, energy consumption, and cost than prior approaches.

The authors of study [10] developed a semantic middleware to address IoT service composition challenges, incorporating contextual service search and semantic analysis. Automated, scalable service composition enhanced scalability, validated on innovative city scenarios, with improved service discovery, selection, and composition metrics versus existing methods.

In study [11], a fuzzy-driven hybrid algorithm combining ACO and Artificial Bee Colony (ABC) methods was proposed for cloud-fog IoT service composition. The approach optimized QoS metrics, including energy consumption, availability, reliability, and cost, achieving significant performance improvements over contemporary techniques.

Moreover, in study [12], a service composition technique based on Grey Wolf Optimization (GWO) within the MapReduce methodology was presented for QoS-aware IoT applications. The model achieved energy savings, reduced response times, and enhanced availability and cost metrics, with average performance gains over baseline algorithms.

The authors in study [13] proposed an enhanced ABC with a dynamic dimensionality reduction-inspired mechanism for IoT Service Composition. Further, the dimensions of disparity adjustment among solutions enable a method that has improved convergence rates and a more balanced exploration of solution exploitation. This significantly facilitates energy consumption to enhance availability and reliability with cost metrics.

In study [14], the Discrete Adjustable Lion Optimization Algorithm (DALOA) was proposed for composing IoT services, employing sub-populations and operators like roaming, mating, and migration. The approach provided a strong balance between exploitation and exploration, achieving near-optimal QoS-aware compositions in reduced execution time. A QoS-aware service discovery, developed using WOA and GA, is proposed in study [15]. This bioinspired technique has proven efficient in selecting a way of optimally utilizing energy, data access time, and cost-effectiveness in IoT service discovery.

While some existing approaches, as summarized in Table I, represent significant advances in IoT service composition, several critical gaps remain. Most of these approaches, such as HMM-based or semantic middleware-based, are not sufficiently adaptive for wider-scale IoT environments because of computational overhead or energy inefficiency in IoT devices. The majority of them experience performance bounds in highly dynamic IoT scenarios.

TABLE I. RECENT LITERATURE IN IOT SERVICE COMPOSITION

| Algorithm | Key features | Performance gains | Shortcomings |
|---|---|---|---|
| ANN-PSO [8] | Hybrid approach combining ANNs for QoS enhancement and PSO for candidate service selection. Formal verification using labeled transition systems ensures correctness. | Achieved better response time, availability, and cost efficiency. Demonstrated improved fitness function values compared to other algorithms. | High computational complexity due to the hybrid approach and formal verification methods. |
| HMM + ACO [9] | The hidden Markov Model predicts QoS metrics based on emissions and transitions optimized by the Viterbi algorithm. ACO is used for service path identification. | Enhanced QoS regarding energy usage, cost, reliability, and response time. Outperformed prior techniques in availability and efficiency. | Limited scalability for real-time large-scale IoT environments due to the computational overhead of HMM. |
| Semantic middleware [10] | Modular and context-aware semantic abstraction for discovering IoT services, semantic filtering, and lightweight automatic service composition. Validated in smart city scenarios. | Improved scalability of service discovery by 15%, selection by 20%, and composition by 40% compared to state-of-the-art methods. | It focuses primarily on scalability but lacks energy efficiency and response time optimizations. |
| ACO + ABC (Fuzzy-based hybrid) [11] | A hybrid algorithm combining ACO and ABC algorithms with a fuzzy logic system. Focuses on energy-aware and QoS-based service selection in cloud-fog architectures. | Reduced energy utilization by 17%, improved availability by 8%, enhanced reliability by 4%, and lowered cost by 21% on average. | Increased complexity due to hybridization and dependency on parameter tuning for optimal performance. |
| GWO + MapReduce [12] | Combines GWO with the MapReduce framework to enable large-scale IoT service composition optimization. Targets QoS metrics like response time, cost, and energy savings. | Achieved a 24% reduction in cost, an 11% gain in availability, a 14% drop in response time, and a 40% energy savings. | MapReduce overhead may impact performance in highly dynamic IoT scenarios. |
| ABC with dynamic reduction [13] | The enhanced ABC algorithm introduces a dynamic reduction mechanism. Adjusts dimension disparities among solutions dynamically for better exploration-exploitation balance. | Decreased energy consumption by 17%, increased availability by 10%, improved reliability by 8%, and lowered cost by 23% compared to alternatives. | Potential convergence issues if initial dimension disparities are not set optimally. |
| DALOA [14] | Discrete adaptive lion optimization algorithm with unique operators (roaming, mating, migration). Balances strong global exploration through nomad roaming and efficient local exploitation via pride searching. | Achieved the best trade-off between exploration and exploitation. Reduced execution time and provided near-optimal IoT service compositions. | Increased complexity due to multiple operators and higher execution time for larger populations. |
| WOA + GA [15] | Integrates WOA with genetic algorithm for efficient IoT service discovery and selection. Bio-inspired optimization enhances QoS awareness in dynamic environments. | Optimized energy utilization, reduced data access time, and improved cost-effectiveness compared to traditional methods. | Lacks adaptability to large-scale IoT environments due to limited scalability and high computational cost. |

In addition, the optimal balance between exploitation and exploration limits the possibility of obtaining a globally optimal solution with high efficiency. This study attempts to fill these gaps by incorporating the Fibonacci search principle into the WOA to leverage its global optimization strengths to enhance the convergence rates, stability, and QoS outcomes in IoT service composition.

## III. Proposed Methodology

### A. Problem Formulation and Statement

An IoT service refers to functional components within the IoT environment that facilitate the interaction and exchange of information between devices [16]. These services can be defined as a triple ($TDP$, $FDP$, $QoSDP$) where TDP stands for the text description of services, providing a semantic explanation of its functionality; FDP represents the functional description of services, detailing its operations and capabilities; and QoSDP refers to the Quality of Service (QoS) characteristics associated with IoT services, describing non-functional properties like execution time, cost, reliability, and trust. The QoS attributes provide measurable criteria for assessing service performance and ensuring non-functional requirements are met. These attributes are particularly crucial when selecting services for specific tasks in IoT applications.

Abstract IoT services represent a set of service instances that perform similar or identical functions. These services are abstracted into individual tasks within a requirements workflow and ensure functional compatibility but differ in their QoS values, making them key candidates in the service composition process [17].

As shown in Fig. 1, the composition of IoT services can follow various control logic structures based on user requirements. In the loop, certain tasks are repeated iteratively for a specified number of iterations (Fig. 1(a)). In the selection fashion, tasks are chosen based on specific conditions or decision points (Fig. 1(b)). In the parallel way, multiple tasks are executed simultaneously to improve performance (Fig. 1 (c)). Lastly, in the sequential approach, tasks are executed one after the other in a predefined order (Fig. 1 (d)). Given the focus on simplicity and efficiency, this paper exclusively considers sequential structures for IoT service composition.

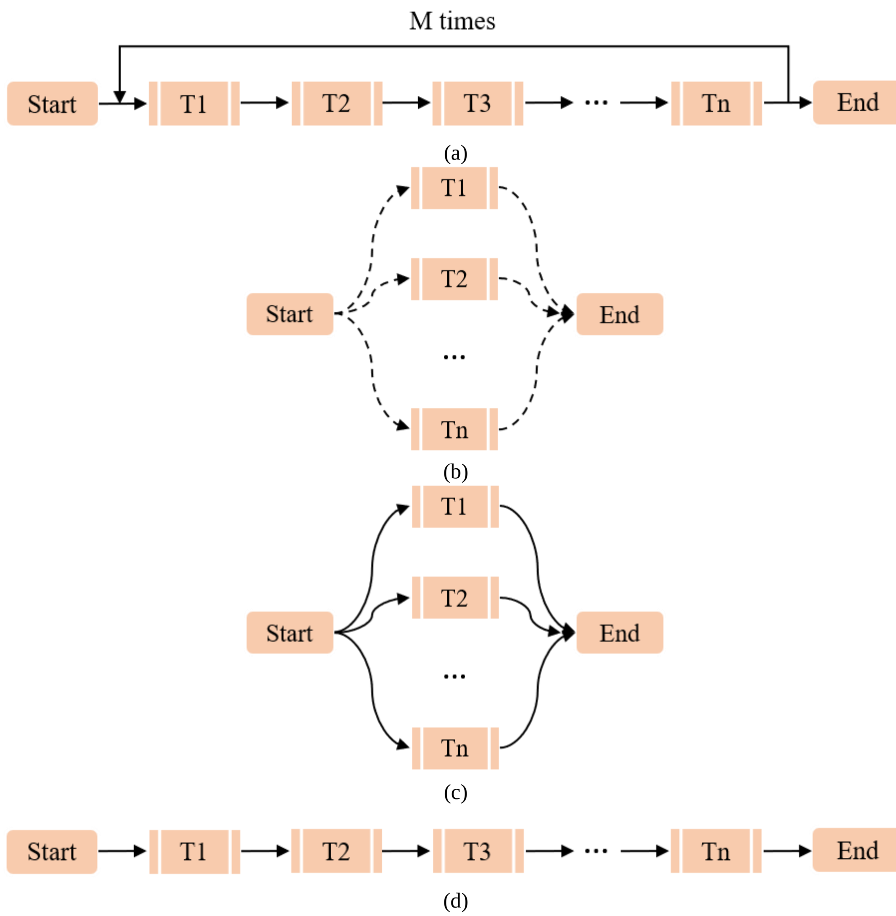



Fig. 1. IoT service composition structures: (a) Loop, (b) Selection, (c) Parallel, and (d) Sequential.

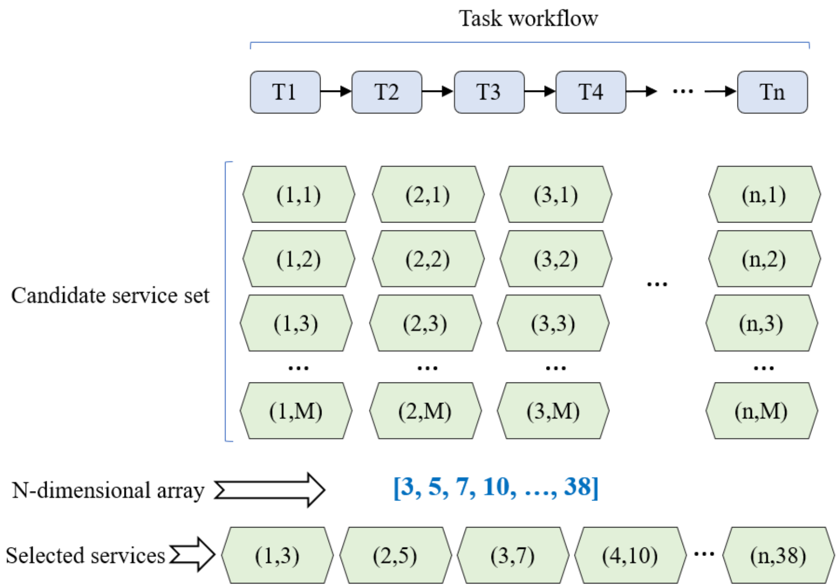



Fig. 2. Optimization process for selecting optimal IoT services from candidate sets.

The IoT service composition involves selecting specific services from a large pool of candidates to satisfy user requirements and QoS constraints. As illustrated in Fig. 2, the optimization process begins with a task workflow $T_1, T_2, \ldots, T_n$, where each task represents an abstract IoT service. For each task $T_i$, there exists a candidate service set $S(i, M)$, from which the most optimal service must be selected. Each task $Ti$ has several service options $S(i, 1), S(i, 2), \ldots, S(i, M)$. The selected services from all tasks are represented as an array [3, 5, 7, 10, …, 38], where each number corresponds to the selected service for a specific task. For a system with $n$ tasks and $M$ service candidates per task, the total number of possible combinations is $M^n$. This combinatorial complexity makes the service composition problem an NP-hard optimization challenge.

Several key QoS metrics are considered to evaluate the effectiveness of an IoT service composition, including reliability, credibility, service cost, and execution time. Reliability indicates the likelihood that the IoT service composition can complete tasks successfully without failure, calculated using Eq. (1) [18].

$$Q_r = \prod_{i=1}^{n} q_i^r \tag{1}$$

Credibility measures the user's trust level in the service composition based on factors such as reputation, expressed by Eq. (2) [19].

$$Q_c = \frac{1}{n}\sum_{i=1}^{n} q_i^c \tag{2}$$

Service cost refers to the monetary cost incurred by the user for utilizing the IoT service composition, calculated by Eq. (3) [20].

$$Q_{co} = \sum_{i=1}^{n} q_i^{co} \tag{3}$$

Execution time represents the total time required for the service composition to execute, including the processing time of all tasks, calculated by Eq. (4) [21].

$$Q_t = \sum_{i=1}^{n} q_i^t \tag{4}$$

The aggregated QoS value of an IoT service composition is calculated as a weighted sum of the above QoS metrics using Eq. (5).

$$Q = \sum_{k \epsilon \{r,c,co,t\}} Q_k \omega_k \tag{5}$$

Where $\omega_k$ denotes the weight assigned to each QoS metric. This weight reflects the relative importance of the corresponding attribute in the overall composition.

The objective is to select one service instance for each task in the workflow such that the aggregated QoS value $Q$ is maximized. This optimization ensures that the composition meets user-defined QoS requirements and achieves the best possible performance in terms of execution time, cost, credibility, and reliability.

## B. *Enhanced Whale Optimization Algorithm*

The WOA, introduced by Mirjalili and Lewis [22], is a heuristic optimization approach inspired by the hunting behavior of humpback whales. The algorithm mimics the whales' bubble-net feeding strategy as its central mechanism for solving optimization problems. The overall process of WOA is illustrated in Fig. 3. WOA consists of three main phases: encircling prey, bubble-net feeding, and searching for prey.

These phases emulate the whales' local exploitation and global exploration strategies, making WOA a versatile optimization framework.

The first phase of WOA involves encircling the prey, which represents the best solution found so far. Whales are assumed to position themselves around the prey to prepare for attack. Mathematically, this behavior is modeled using Eq. (6) and (7).

$$\vec{D} = |\vec{C}.\vec{X}_{best}(t) - \vec{X}(t)| \tag{6}$$

$$\vec{X}(t+1) = \vec{X}_{best}(t) - \vec{A}.\vec{D} \tag{7}$$

Where $\vec{A}$ and $\vec{C}$ are coefficient vectors, $\vec{D}$ stands for the distance between the whale and the prey, $\vec{X}(t)$ is the current position of the whale, and $\vec{X}_{best}(t)$ is the position vector of the best solution (prey) at iteration $t$. The vectors $\vec{A}$ and $\vec{C}$ are computed using Eq. (8) and (9).

$$\vec{A} = 2.\vec{a}.\vec{r}_1 - \vec{a} \tag{8}$$

$$\vec{C} = 2.\vec{r}_2 \tag{9}$$

Where $\vec{r}_1$ and $\vec{r}_2$ range between [0, 1], and $\vec{a}$ drops linearly from 2 to 0 with increasing iterations.

The bubble-net feeding phase simulates whales' two simultaneous strategies to capture prey: shrinking, encircling, and spiral updating. These strategies reflect both global and local search mechanisms.

Shrinking encircling reduces the distance between whales and prey over time by decreasing the range of $|\vec{A}|$. This is accomplished by progressively lowering the value of $\vec{a}$. Spiral updating mimics the spiral-shaped trajectory of whales around their prey. This phenomenon is represented mathematically using Eq. (10) and (11).

$$\vec{X}(t+1) = \vec{D}_p.e^{bl}.cos(2\pi l) + X_{best}(t) \tag{10}$$

$$\vec{D}_p = |\vec{X}_{best}(t) - \vec{X}(t)| \tag{11}$$

Where $\vec{D}_p$ refers to the distance between the whale and the prey, $b$ defines the shape of the logarithmic spiral, and $l$ signifies a random number in the range [-1, 1].

To combine these two strategies, WOA uses a probabilistic mechanism where a random number $P$ determines which behavior is applied in each iteration, calculated using Eq. (12).

$$\vec{X}(t+1) = \begin{cases} \vec{D}_p.e^{bl}.cos(2\pi l) + X_{best}(t), & P < 0.5 \\ \vec{X}_{best}(t) - \vec{A}.\vec{D}, & P \geq 0.5 \end{cases} \tag{12}$$

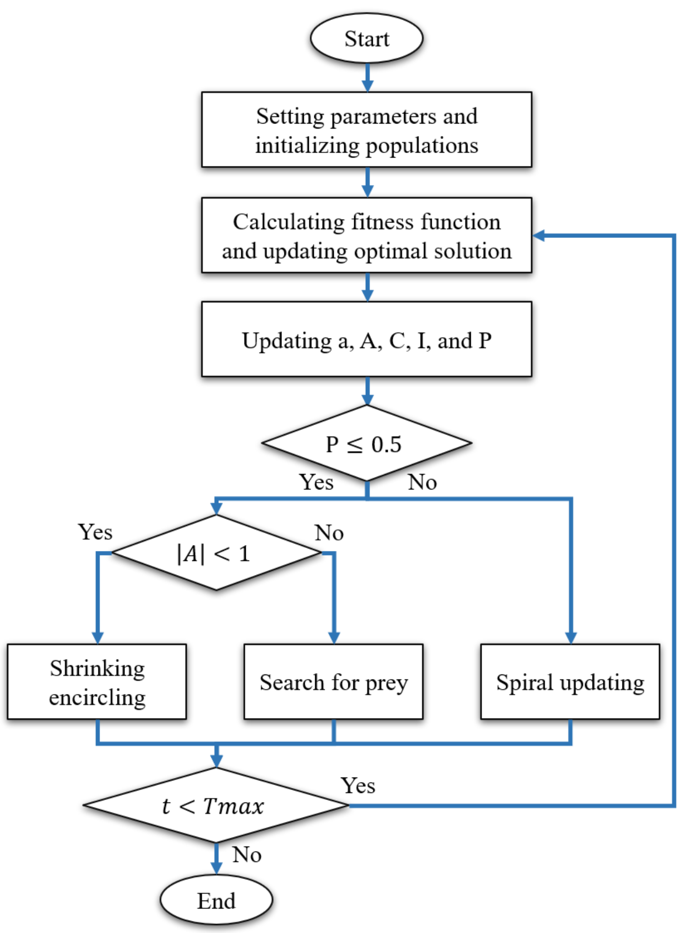



Fig. 3. WOA flowchart.

This probabilistic combination of behaviors balances global exploration and local exploitation. The final phase focuses on exploration by searching for prey. When the condition $|\vec{A}| \geq 1$ is satisfied, and whales move randomly for better solutions. This behavior is modeled using Eq. (13) and (14).

$$\vec{D} = \left|\vec{C}.\vec{X}_{rand}(t) - \vec{X}(t)\right| \quad (13)$$

$$\vec{X}(t+1) = \vec{X}_{rand}(t) - \vec{A}.\vec{D} \quad (14)$$

Where $\vec{X}_{rand}(t)$ is the position vector of a randomly selected whale. This phase prevents WOA from getting stuck on local optimum and enhances the algorithm's global search capability.

The enhanced WOA (EWOA) builds upon the original WOA by addressing its inherent shortcomings, such as slow convergence, poor accuracy, and proneness to local optima. This improvement is achieved by integrating a nonlinear cross-weight mechanism and the Fibonacci Search Method (FSM). The enhanced algorithm ensures optimal equilibrium between diversification (exploration) and intensification (exploitation), key components of any robust optimization method. Fig. 4 presents the pseudocode of the EWOA.

```
Input:
    Initialize population X = {X_1, X_2, ..., X_n}
    Define objective function f(X)
    Set algorithm parameters: maximum iterations (T_max) and control parameters (α, A, C, I, P)
Step 1: Initialize and Evaluate
    Evaluate the fitness of each solution in the population.
    Identify the best solution BEST_sol.
Step 2: Iterative optimization
    While iteration t is less than T_max
        Update control parameters.
        Exploration and exploitation decision:
            if P ≤ 0.5 (Exploration phase):
                if |A| < 1:
                    Perform shrinking encircling mechanism to update position.
                Else |A| ≥ 1:
                    Conduct random search for prey to enhance exploration.
            Else if P > 0.5 (Exploitation phase):
                    Perform spiral updating to refine solution accuracy.
Boundary Verification:
    Check if any search agents exceed the search space limits and correct their positions if necessary.
Fitness Evaluation and Update:
    Calculate the fitness of each solution.
    If a better solution is found, update BEST_sol.
Increment iteration count t = t + 1.
End While
Output:
    Return the best solution found BEST_sol
```

Fig. 4. Pseudocode of the EWOA.

Diversification explores the entire search space to identify global optima, while intensification involves focusing on local regions for fine-tuning solutions. The original WOA struggles with achieving an optimal balance between these two processes, often leading to stagnation at local optima. EWOA addresses this by introducing a nonlinear crossover weight to enhance solution diversity during exploration and incorporating the FSM for a more efficient local search, improving solution accuracy and convergence speed.

The EWOA incorporates crossover weights during location updates. The revised position update model is defined using Eq. (15).

$$\vec{X}(t+1) = \begin{cases} (\vec{X}_{best}(t) - \vec{A}.\vec{D}).CR1, & if\ a < 0.5 \\ \vec{D}'.e^{bl}.cos(2\pi l).CR2 + \vec{X}_{best}(t).(1-CR2), & if\ a \geq 0.5 \end{cases} \quad (15)$$

Where $\vec{D}'$ stands for the distance between the current and best solutions, $CR1 = exp\left(tan\left(rand(1,N)\right)\right)$, $CR2 = 3.\left(0.5 - rand(1,N)\right).f$, and $f = exp\left(-\frac{Iteration}{Max\_iteration}\right)$. This mechanism ensures that solutions generated during exploration maintain sufficient diversity while enabling convergence during exploitation.

FSM is incorporated into EWOA to improve local search efficiency. FSM minimizes the search space by applying Fibonacci sequences, which guide the selection of optimal intervals. It operates as follows:

Generate Fibonacci numbers: The Fibonacci sequence $F = [F_1, F_2, \ldots, F_n]$ is expressed by Eq. (16).

$$F_n = F_{n-1} + F_{n-2}, \;\; F_0 = 1, \;\; F_1 = 1 \quad (16)$$

Calculate initial points: Two points $t_1$ and $t_2$ are defined in the search range [LL, UL] using Eq. (17).

$$t_1 = LL + \frac{F_{n-2}}{F_n}.(UL - LL)$$
$$t_2 = UL - \frac{F_{n-2}}{F_n}.(UL - LL) \quad (17)$$

Where $UL$ and $LL$ define the upper and lower bounds of the range.

Evaluate function values: Compare the function values at $t_1$ and $t_2$:

If $(t_2) > (t_1)$, shift the range to the left.

If $(t_1) > (t_2)$, shift the range to the right.

## IV. RESULTS

The effectiveness of the proposed EWOA was evaluated for optimizing the QoS-based IoT service composition optimization problem. EWOA effectiveness was evaluated against standard WOA [22], DALOA [14], and Genetic Algorithm (GA) [23] using three key evaluation criteria: effectiveness, convergence, and stability. The tests were carried out in a Windows 10 system powered by an Intel Core i7-12700F processor, 16 GB RAM, and PyCharm Community Edition 2022.3.

The experiments used randomly generated datasets based on the QoS value ranges defined in Table II. Four QoS attributes, including execution time, service cost, credibility, and reliability, were evaluated for candidate IoT service instances. The dataset scales were represented as $A \times I$, where $A$ denotes the number of abstract service tasks, and $I$ signifies the number of candidate services per task. The datasets included the following scales: 10×50, 10×100, 20×50, 20×100, 30×50, and 30×100. Each experiment was repeated 100 times to ensure robustness, and the results were analyzed to measure the algorithm's performance under varying scales and iterations.

TABLE II. QOS VALUE RANGES

| Attributes | Reliability | Credibility | Service cost | Execution time |
|---|---|---|---|---|
| Ranges | (0.1,1] | (2,10] | (0,100] | (0,60] |

The effectiveness of the algorithms was assessed by the average fitness values obtained after 100 global iterations. EWOA demonstrated significantly higher efficacy than WOA, DALOA, and GA, as shown in Table III and Fig. 5-7. Key observations include:

- For smaller scales (e.g., 10×50), EWOA slightly outperformed other algorithms, achieving higher-quality solutions.
- For larger scales (e.g., 30×100), EWOA's advantage became more evident, delivering higher fitness values with a broader margin.

TABLE III. FITNESS VALUES FOR VARIOUS SERVICE COMPOSITION SCALES

| No. of abstract service tasks | No. of candidate services | Fitness values | | | |
|---|---|---|---|---|---|
| | | GA | DALOA | WOA | EWOA |
| 10 | 50 | 3.75 | 4.13 | 4.22 | 4.92 |
| | 100 | 4.02 | 4.21 | 4.36 | 4.85 |
| 20 | 50 | 6.49 | 6.75 | 7.06 | 9.53 |
| | 100 | 7.16 | 7.53 | 8.12 | 9.61 |
| 30 | 50 | 9.15 | 9.91 | 10.58 | 13.91 |
| | 100 | 10.13 | 11.54 | 12.02 | 14.05 |

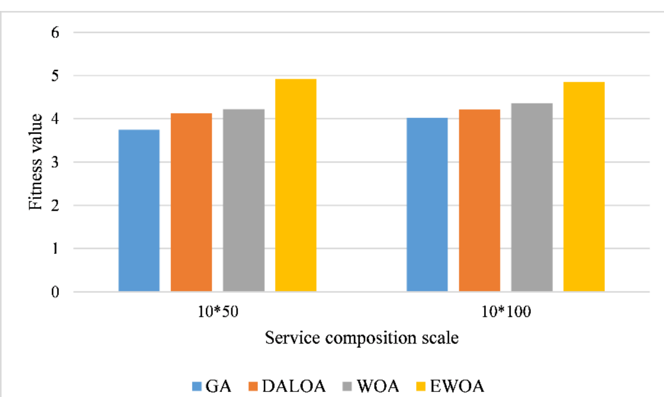



Fig. 5. Fitness value comparison for 10 service tasks.

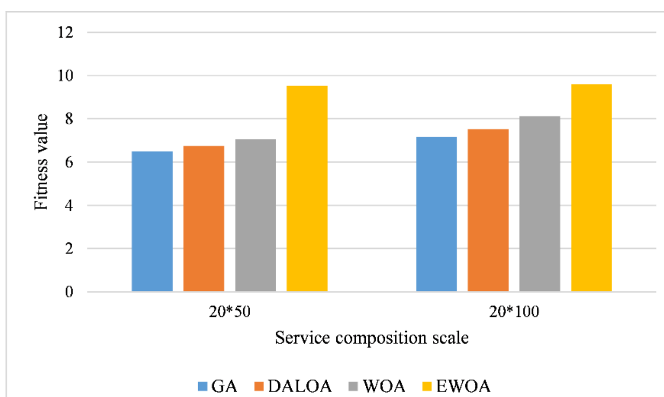



Fig. 6. Fitness value comparison for 20 service tasks.

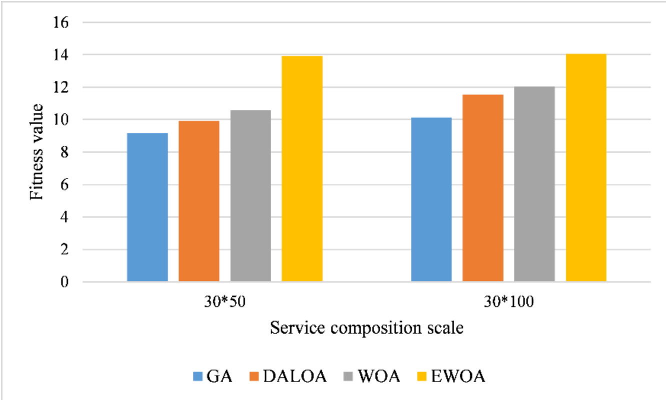



Fig. 7. Fitness value comparison for 30 service tasks.

The nonlinear crossover weights dynamically adjust exploration/exploitation, ensuring sufficient solution diversity at the beginning and accurate refinement in later iterations. The Fibonacci search strategy effectively narrowed the search space, leading to higher solution accuracy and better optimization of QoS attributes.

The convergence performance of the algorithms was analyzed based on their fitness values over iterations, as depicted in Fig. 8 – Fig. 10. EWOA consistently converged faster and to better solutions than WOA, DALOA, and GA. Notable findings include:

- At smaller scales (e.g., 10×50), EWOA achieved convergence within fewer iterations than other algorithms.
- At larger scales (e.g., 30×100), EWOA showed a significant fitness advantage and faster convergence speed.

The nonlinear crossover weights ensured efficient exploration in the early iterations, preventing premature convergence to local optima. The Fibonacci search strategy refined the best solutions in the exploitation phase, accelerating convergence toward the global optimum.

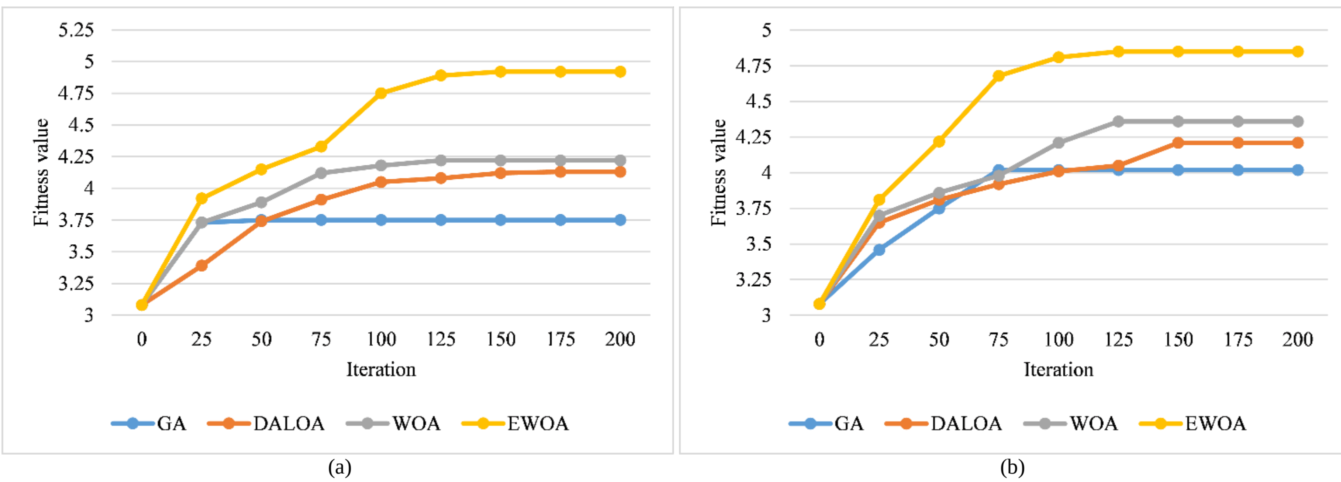



Fig. 8. Convergence performance comparison: (a) 10 service tasks and 50 candidate services, (b) 10 service tasks and 100 candidate services.

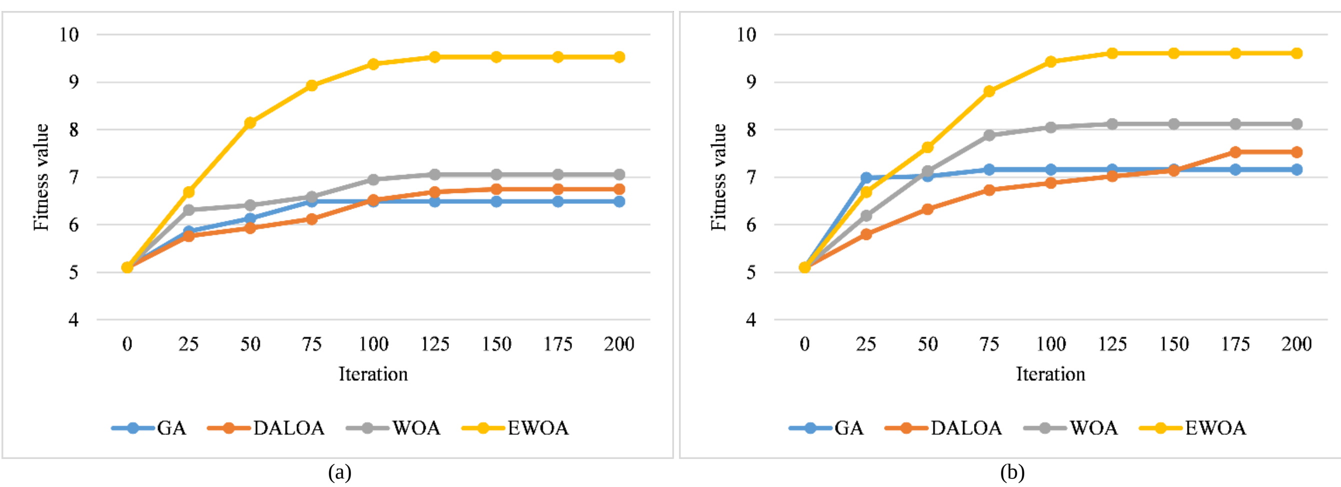
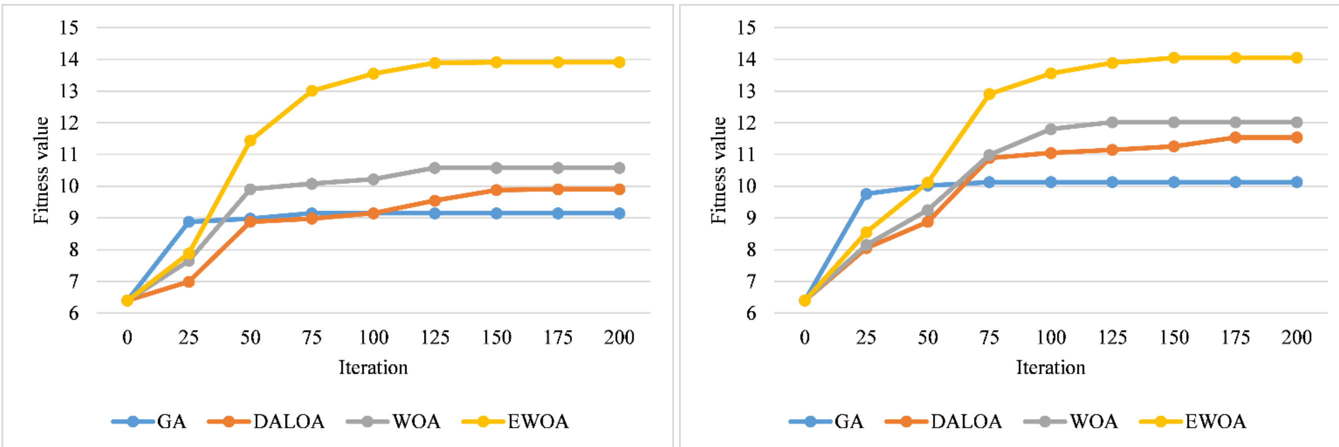



Fig. 9. Convergence performance comparison: (a) 20 service tasks and 50 candidate services, (b) 20 service tasks and 100 candidate services




(a) (b)

Fig. 10. Convergence performance comparison: (a) 30 Service tasks and 50 candidate services, (b) 30 Service tasks and 100 candidate services.

TABLE IV. STANDARD DEVIATION FOR VARIOUS SERVICE COMPOSITION SCALES

| No. of abstract service tasks | No. of candidate services | Standard deviation | | | |
|---|---|---|---|---|---|
| | | GA | DALOA | WOA | EWOA |
| 10 | 100 | 0.0831 | 0.0526 | 0.0415 | 0.0089 |
| 20 | 100 | 0.1173 | 0.1085 | 0.1019 | 0.0269 |
| 30 | 100 | 0.1569 | 0.1503 | 0.1494 | 0.0612 |

The stability of the algorithms was measured in terms of standard deviation of optimal fitness values across 100 experiments, as shown in Table IV and Fig. 11. Lower standard deviation values indicate higher stability. Key findings include:

- EWOA exhibited significantly lower standard deviation values compared to WOA, DALOA, and GA, particularly at larger scales (e.g., 30×100).
- As scales increased, the standard deviation of all algorithms rose. However, EWOA maintained superior stability, consistently delivering reliable results with minimal variation.

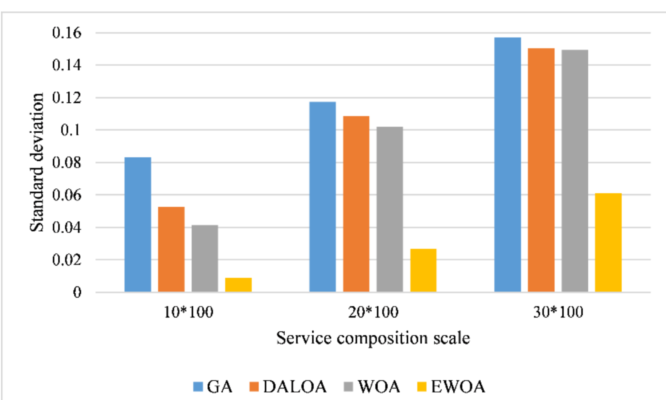



Fig. 11. Standard deviation comparison.

The nonlinear crossover weights provided adaptive adjustments, ensuring robustness against variations in initial population diversity. The Fibonacci search strategy reinforced solution refinement, minimizing the impact of random fluctuations in search trajectories.

## V. DISCUSSION

The results demonstrate that the proposed EWOA significantly improves the optimization of QoS-based IoT service composition compared to standard WOA, DALOA, and GA. One of the key advantages of EWOA is its superior performance across various dataset scales, particularly for larger problem instances, where it consistently achieved higher-quality solutions. The nonlinear crossover weights effectively balanced exploration and exploitation, preventing premature convergence and ensuring sustained search diversity. Additionally, the integration of the Fibonacci search strategy enabled precise solution refinement by efficiently narrowing the search space, ultimately leading to better fitness values and improved QoS attribute optimization. These results validate the effectiveness of the proposed modifications, particularly in handling complex IoT service composition scenarios where scalability and solution accuracy are critical.

Furthermore, the convergence analysis confirms that EWOA consistently outperforms its counterparts in both speed and solution quality. The rapid convergence observed in smaller-scale datasets indicates that EWOA is highly effective even for less complex problems. However, its performance advantage becomes more pronounced as the problem scale increases, demonstrating its robust scalability. Additionally, stability analysis reveals that EWOA maintains lower standard deviation values, signifying its ability to deliver consistent and reliable results across multiple runs. This is primarily due to the adaptive adjustments of nonlinear weights, which dynamically regulate search behavior, and the Fibonacci search refinement, which enhances exploitation precision. The findings underscore the suitability of EWOA for large-scale IoT service composition problems, offering a highly efficient, stable, and scalable optimization framework.

## VI. Conclusion

This study proposed EWOA to resolve the QoS-based IoT service composition optimization problem. EWOA mitigated some disadvantages of conventional optimization algorithms, particularly slow convergence, the tendency toward local optima, and inability to balance exploration and exploitation. With nonlinear crossover weights combined with the Fibonacci search strategy, the EWOA optimized global exploration and local exploitation, providing outstanding performance in achieving optimization. The experimental test validated the efficiency of EWOA under different composition scenarios, from small-scale to large-scale problems. EWOA had better fitness, higher convergence speed, and more substantial stability than WOA, DALOA, and GA in all cases. Dynamic adjustment of the nonlinear crossover weights regulated solution diversity and refinement during optimization, whereas the Fibonacci search strategy improved efficiency in local search and prevented falling into suboptimal solutions.

EWOA proved to be especially helpful in large-scale composition, while its ability to handle increased problem complexity led to significant gains over existing algorithms. Moreover, its computational efficiency and stability over multiple runs indicated the appropriateness of real-world IoT scenarios. Future work will focus on integrating the EWOA with dynamic composition models to handle real-time and evolving QoS requirements. This could be further enhanced by hybridizing the EWOA with other metaheuristics for optimization problems that are highly complex and multidimensional. Given its capability, the EWOA represents a promising trend toward development in the field of optimization solutions within an IoT context.

## References


[1] A. Choudhary, "Internet of Things: a comprehensive overview, architectures, applications, simulation tools, challenges and future directions," Discover Internet of Things, vol. 4, no. 1, p. 31, 2024.

[2] B. Pourghebleh, N. Hekmati, Z. Davoudnia, and M. Sadeghi, "A roadmap towards energy - efficient data fusion methods in the Internet of Things," Concurrency and Computation: Practice and Experience, vol. 34, no. 15, p. e6959, 2022.

[3] L. Camargo, J. Pauletti, A. M. Pernas, and A. Yamin, "VISO approach: A socialization proposal for the Internet of Things objects," Future Generation Computer Systems, vol. 150, pp. 326-340, 2024.

[4] V. Hayyolalam, B. Pourghebleh, M. R. Chehrehzad, and A. A. Pourhaji Kazem, "Single - objective service composition methods in cloud manufacturing systems: Recent techniques, classification, and future trends," Concurrency and Computation: Practice and Experience, vol. 34, no. 5, p. e6698, 2022.

[5] D. K. K. Reddy, J. Nayak, H. Behera, V. Shanmuganathan, W. Viriyasitavat, and G. Dhiman, "A systematic literature review on swarm intelligence based intrusion detection system: past, present and future," Archives of Computational Methods in Engineering, vol. 31, no. 5, pp. 2717-2784, 2024.

[6] Z. Zhang, L. Tan, D. Martín, L. Qian, M. Khishe, and P. Jangir, "A dual-adaptive stochastic reinforcement chimp optimization algorithm for fire detection and multidimensional problem solving," Scientific Reports, vol. 14, no. 1, p. 31226, 2024.

[7] S. Paul, S. De, and S. Bhattacharyya, "Emerging trends in computational swarm intelligence: A comprehensive overview," Recent Trends in Swarm Intelligence Enabled Research for Engineering Applications, pp. 1-40, 2024.

[8] M. Hosseinzadeh et al., "A hybrid service selection and composition model for cloud-edge computing in the internet of things," IEEE Access, vol. 8, pp. 85939-85949, 2020, doi: https://doi.org/10.1109/ACCESS.2020.2992262.

[9] S. Sefati and N. J. Navimipour, "A qos-aware service composition mechanism in the internet of things using a hidden-markov-model-based optimization algorithm," IEEE Internet of Things Journal, vol. 8, no. 20, pp. 15620-15627, 2021.

[10] S. Berrani, A. Yachir, B. Djamaa, S. Mahmoudi, and M. Aissani, "Towards a new semantic middleware for service description, discovery, selection, and composition in the Internet of Things," Transactions on Emerging Telecommunications Technologies, vol. 33, no. 9, p. e4544, 2022.

[11] M. Hamzei, S. Khandagh, and N. Jafari Navimipour, "A quality-of-service-aware service composition method in the internet of things using a multi-objective fuzzy-based hybrid algorithm," Sensors, vol. 23, no. 16, p. 7233, 2023.

[12] A. Vakili, H. M. R. Al - Khafaji, M. Darbandi, A. Heidari, N. Jafari Navimipour, and M. Unal, "A new service composition method in the cloud - based internet of things environment using a grey wolf optimization algorithm and MapReduce framework," Concurrency and Computation: Practice and Experience, vol. 36, no. 16, p. e8091, 2024.

[13] G. Xiao, "Toward Optimal Service Composition in the Internet of Things via Cloud-Fog Integration and Improved Artificial Bee Colony Algorithm," International Journal of Advanced Computer Science & Applications, vol. 15, no. 5, 2024.

[14] S. Ait Hacène Ouhadda, S. Chibani Sadouki, A. Achroufene, and A. Tari, "A Discrete Adaptive Lion Optimization Algorithm for QoS-Driven IoT Service Composition with Global Constraints," Journal of Network and Systems Management, vol. 32, no. 2, p. 34, 2024.

[15] X. Liu and Y. Deng, "A new QoS-aware service discovery technique in the Internet of Things using whale optimization and genetic algorithms," Journal of Engineering and Applied Science, vol. 71, no. 1, p. 4, 2024.

[16] B. Pourghebleh, V. Hayyolalam, and A. Aghaei Anvigh, "Service discovery in the Internet of Things: review of current trends and research challenges," Wireless Networks, vol. 26, no. 7, pp. 5371-5391, 2020.

[17] B. Pourghebleh and N. J. Navimipour, "Data aggregation mechanisms in the Internet of things: A systematic review of the literature and recommendations for future research," Journal of Network and Computer Applications, vol. 97, pp. 23-34, 2017, doi: https://doi.org/10.1016/j.jnca.2017.08.006.

[18] D. Rupanetti and N. Kaabouch, "Combining Edge Computing-Assisted Internet of Things Security with Artificial Intelligence: Applications, Challenges, and Opportunities," Applied Sciences, vol. 14, no. 16, p. 7104, 2024.

[19] L. Wei, Y. Yang, J. Wu, C. Long, and B. Li, "Trust management for Internet of Things: A comprehensive study," IEEE Internet of Things Journal, vol. 9, no. 10, pp. 7664-7679, 2022.

[20] C. Liu, Z. Su, X. Xu, and Y. Lu, "Service-oriented industrial internet of things gateway for cloud manufacturing," Robotics and Computer-Integrated Manufacturing, vol. 73, p. 102217, 2022.

[21] H. Gao, Y. Zhang, H. Miao, R. J. D. Barroso, and X. Yang, "SDTIOA: modeling the timed privacy requirements of IoT service composition: a user interaction perspective for automatic transformation from BPEL to timed automata," Mobile Networks and Applications, pp. 1-26, 2021.

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

[23] Q. Li, R. Dou, F. Chen, and G. Nan, "A QoS-oriented Web service composition approach based on multi-population genetic algorithm for Internet of things," International Journal of Computational Intelligence Systems, vol. 7, no. Suppl 2, pp. 26-34, 2014.