A Novel Multi-Objective Service Scheduling Framework for Fog-Driven Agricultural Internet of Things

Mengya NIE[1]*, Fengmin JIAO[2]
School of Drones and Intelligent Industry, Jiangsu Aviation Technical College, Zhenjiang City, Jiangsu Province, 212000, China[1]
School of Aviation Engineering Institute, Jiangsu Aviation Technical College, Zhenjiang City, Jiangsu Province, 212000, China[2]

***Abstract*—Timely, context-aware service recommendations are most significant for maximizing system efficiency and resource utilization in fog-enabled Internet of Things (IoT) for smart agriculture. Classical Grey Wolf Optimization (GWO) algorithms are prone to premature convergence and lack robustness for multi-objective and dynamic environments. This study presents an Enhanced Grey Wolf Optimization (EGWO) algorithm that incorporates adaptive weighting and nonlinear exploration-exploitation control. The EGWO algorithm uses a dynamic alpha-beta-delta hierarchy for weighting and a cosine-based decay for the control parameter, ensuring better convergence and global search efficiency. In the fog-based service recommendation environment, EGWO optimizes multiple conflicting objectives, including latency, energy consumption, trust, and service utility. A comparison is made between standard GWO, PSO, and DE using a simulated agricultural fog-IoT network. The results demonstrate that EGWO achieves faster convergence speed, higher recommendation quality, and uniform behavior across all test scenarios. The method is suitable for practical application in real-time agricultural use cases, ensuring guaranteed optimal service allocation and improved decision support at the network's edge.**

Keywords—Grey wolf; fog computing; internet of things; smart agriculture; service recommendation

I. Introduction

The convergence of Internet of Things (IoT) technology and farming has made intelligent farming feasible, enabling real-time monitoring, automation, and information-based decision-making [1]. For such environments, fog computing places computational resources closer to the field, reducing dependencies on central cloud infrastructure [2]. Proximity of this nature guarantees rapid response times and higher responsiveness, essential in time-sensitive operations such as equipment maintenance, pest control, and irrigation management [3]. Controlling and service recommendations for an IoT architecture based on fog introduce new challenges. Real-time recommendations for services are necessary to accommodate dynamic resources, IoT device-at-the-edge energy limitations, and to preserve service trust and reliability [4]. With more connected gadgetry and service providers, request mapping to the most fitting services is an intricate multi-objective challenge [5].

Also, based on the learning, communication reliability and efficiency are improving. For instance, neural coding frameworks like DeepPolar have used a reliability-aware loss function to enhance frame error rate performance, which suggests the possibilities of smart solutions in future communication systems [6]. Metaheuristic algorithms are being considered as promising approaches to find the optimal solutions for various IoT problems such as service allocation and scheduling. The Grey Wolf Optimization technique (GWO) is one of the top metaheuristic algorithms owing to its flexible structure and good performance. However, the conventional GWO has some drawbacks. It suffers from premature convergence in high-dimensional search spaces because of its low exploration ability and non-divergent nature [7]. These deficiencies make it inadequate for use in rapidly changing, memory-constrained fog computing networks where fast convergence and thorough search are required. In addition, the fixed roles of the α, β, and δ wolves in the standard GWO restrict its potential for adaptively searching complex, multimodal optimization problems, such as in real-time service recommendation search landscapes of intelligent agriculture.

To address these challenges, an Enhanced Grey Wolf Optimization (EGWO) algorithm is proposed for recommendations in fog-based IoT agricultural networks. Adopting a dynamic saliency between the leaders, the influence of the lead wolves on the other ones changes as the optimization proceeds. Furthermore, a suitable parameter determination in line with an arctan mapping and cosine decay is proposed to gradually adapt the balance of global searching and local searching over time scale. These improvements enhance convergence and eliminate stagnation in local optima. EGWO is capable of selecting an optimal service that meets multiple constraints, e.g., latency requirements, energy consumption, trustworthiness, and usefulness, in a fog computing environment with an efficient multi-objective service recommendation under fog computing constraints. The salient contributions of this study are:

- Proposing a novel EGWO algorithm with adaptive hierarchical weights and nonlinear convergence control.
- Formulating the fog-based service recommendation problem in agricultural IoT as a multi-objective optimization task incorporating latency, energy, trust, and utility.
- Designing a simulation environment to evaluate EGWO and benchmark it against standard GWO, Differential

*Corresponding author.

Evolution (DE), and Particle Swarm Optimization (PSO).

- Proving that EGWO outperforms traditional methods in convergence rates, stability, and solution quality across a range of agricultural service scenarios.

The study is organized as follows. Section II presents the related work on service scheduling in IoT. Section III presents the problem and the system model used in this study. In Section IV, the proposed GWO algorithm enhancements and the details of the algorithm design components and working principles are described. Section V presents the experimental results, comparative analysis, and performance evaluation of the proposed technique. Section VI presents the conclusion of the study with possible future considerations.

II. Related Work

Ravi, et al. [8] proposed the Cluster-Based Reliable Data Aggregation (CRDA) for IoT scenarios to achieve energy consumption and reliability balance. This approach applies the Monarch and Sine-Cosine (MSC) algorithm to form clusters and enhance the data aggregation process with an Enhanced Sunflower Optimization (ESFO) algorithm. The most reliable IoT sensors are selected as cluster heads to transmit data efficiently. A novel Optimal–Learning-Based Deep Neural Network (ROL-DNN) calculates the optimal paths for reliable data aggregation and delivery. The simulation results show the superiority of the proposed system against traditional routing protocols.

Liu, et al. [9] proposed the application of Unmanned Aerial Vehicles (UAVs) as aerial base stations for gathering data in agricultural IoT, which mitigates the constraints of conventional ground-based stations. They formulated a multi-objective optimization problem to maximize the minimum data rate and minimize the energy consumption of both the UAV and the IoT devices. To handle this complicated problem, an Improved Multi-Objective Artificial Hummingbird Algorithm (IMOAHA) based on hybrid initialization, Cauchy mutation, and discrete mutation operators is proposed.

Çavdar, et al. [10] A task scheduling technique in Wireless Sensor Networks (WSNs), called the Shared-Data Approach, is introduced, which considers the presence of multiple applications with different requirements sharing the same infrastructure and addresses sensing and communication load balancing among them. Their method uses a genetic algorithm and also three greedy heuristics (LTSF, LMSF, and LMPF) for data request scheduling. These approaches are designed to maximize resource utilization by minimizing the makespan and the waiting time, while maintaining a good success rate in terms of task execution. The experimental results show that our proposed algorithm outperforms the other scheduling algorithms in a number of performance metrics such as turnaround time and execution throughput.

Vakili, et al. [11] proposed a GWO-based service composition method in IoT and cloud environments. Their work aims at maximizing Quality of Service (QoS) factors (e.g., energy, response time, availability, and cost) using service composition in a MapReduce environment. The simulation results demonstrate a 40% enhancement in energy savings with significant reduction in cost and response time, which outperforms three benchmark algorithms with respect to energy and QoS optimization.

Abohamama, et al. [12] A semi-dynamic real-time scheduling framework is introduced for IoT applications within cloud–fog environments. The algorithm, based on a modified genetic algorithm, models task scheduling as a permutation-based optimization challenge. In this method, tasks are allocated to virtual machines with sufficient resources to reduce processing time.

Shankar and Anbarasan [13] introduced the Improved Multi-objective Grey Wolf Optimization (IMGWO) algorithm for dynamic virtual machine allocation and resource allocation in cloud infrastructure. The method leverages evolutionary strategies combined with fuzzy logic to optimize power consumption and resource allocation. Their approach significantly reduces power consumption, achieving notable improvements over traditional optimization methods such as MOOERA and CRASVM.

To overcome the challenges of cloud-based execution of IoT applications such as network latency and high service cost, Naik [14] put forward the CoS_FRLM model. Their solution is to cluster IoT devices in logical fog nodes, which are led by fog agents to process application requests at the local level. The method incorporates a pheromone-based metric to assess and suggest the best fog node for performing each IoT task, which leads to the best load sharing.

Lin, et al. [15] proposed a weighted interest degree-based recommendation using association rule mining to improve content dissemination in Internet of Vehicles (IoV) networks. Their approach mines user data to generate association rules and predicts user interest scores by considering the similarity between various user preferences. This context-aware recommendation scheme can accurately recommend content in intelligent vehicle systems over IoV scenarios.

Altaleb, et al. [16] proposed an extension of the Cross-Industry Standard Process for Data Mining (CRISP-DM), specifically adapted for agricultural applications. Their framework acknowledges the increasing importance of big data in optimizing machinery performance and enhancing decision-making within the sector. Their approach refines the generic CRISP-DM model by introducing domain-specific guidelines for structuring and handling agricultural datasets.

Despite the fact that the above-stated works pitch different approaches for enhancing IoT systems in agriculture, a considerable number of these works concentrate on optimizing only one objective, e.g., energy or communication efficiency, while ignoring the multi-objective aspect of practical agricultural IoT applications. Further, some solutions neglect the need for dynamic adaptability to environmental changes, e.g., weather conditions and farm topography. In this work, we aim to fill this gap and develop an EGWO algorithm that addresses multiple conflicting objectives, including energy consumption, latency, trust, and service utility in a dynamic fog-enabled agricultural IoT environment. With the integration of the adaptive weighting scheme with the dynamic EEC control strategy, it is expected to enhance the convergence performance, solution quality, and robustness of the proposed method in different agricultural environments. Table I presents a comparative summary of the related research and points out the key contributions, strengths, and weaknesses of each method.

TABLE I. RECENT EFFORTS ON SERVICE SCHEDULING IN IOT

Study	Main contribution	Pros	Cons
[8]	Developed a cluster-based reliable data aggregation scheme using MSC and ISFO algorithms for IoT networks to optimize data aggregation and routing.	Efficient data aggregation, trust-based cluster head selection, and improved routing.	Complexity in implementation due to the integration of multiple algorithms.
[9]	Introduced UAV-assisted data collection for agricultural IoT, optimizing UAV positioning, speed, and transmit power via the improved multi-objective artificial hummingbird algorithm.	An effective solution for flexible data collection in agricultural IoT, with significant performance improvement.	UAV deployment might be costly, and optimal UAV positioning is challenging in irregular farm topologies.
[10]	Developed a shared-data approach for application scheduling in WSNs with a genetic algorithm and greedy algorithms to optimize resource usage.	Reduces communication and sensing load, effective in multi-application environments.	May not scale well with very large networks or highly dynamic conditions.
[11]	Suggested a service composition technique using the MapReduce framework and GWO for IoT and cloud integration, optimizing multiple QoS attributes like energy and cost.	Significant QoS improvement, particularly in energy savings and response time reduction.	Does not account for the dynamic nature of service availability in real-time environments.
[12]	Proposed a semi-dynamic task scheduling algorithm for IoT in cloud-fog environments, based on an enhanced genetic algorithm to optimize task execution and reduce failure rates.	Effective for real-time task scheduling, a balance between execution cost and reliability.	May require fine-tuning to meet specific real-time constraints and support diverse IoT applications.
[13]	Introduced an improved multi-objective grey wolf optimization algorithm for dynamic virtual machine allocation and resource allocation in cloud systems, incorporating fuzzy logic for power efficiency.	Significant power consumption reduction and performance enhancement.	Focuses primarily on cloud VM optimization, not directly applicable to agricultural IoT systems.
[14]	Proposed a co-scheduling framework using fog-nodes and a pheromone indicator for optimal IoT application placement.	Reduces makespan and service cost; enhances resource utilization.	Complexity in dynamic fog-node selection and PI recalculations.
[15]	Suggested a weighted interest recommendation algorithm using association rules for intelligent navigation in IoV.	Improves recommendation accuracy by analyzing user interests.	Scalability challenges with large-scale user data in real-time scenarios.
[16]	An extended CRISP-DM process model tailored for agricultural data to guide structured data mining in farming.	Provides a domain-specific methodology for agricultural analytics.	Lacks automation in dynamic environments; relies on static process stages.

III. PROBLEM DEFINITION AND SYSTEM MODEL

Fog-based IoT architecture consists of three layers: cloud, fog, and IoT. The IoT part consists of the physical devices with electronics and/or electronic components to gather and send data. The cloud tier receives data from the fog tier, after two data transformations and a trust evaluation, and executes advanced analytics to extract value from the raw data. IoT in farming is well known for the massive improvements it can bring to production and environmental impacts. As shown in Fig. 1, the IoT applied in agriculture brings a number of positive impacts: monitoring soil moisture, controlling crop diseases, and reducing fertilizer use, which help in enhancing crop yield and reducing the cost of operation.

Metaheuristic methods have also shown promising results in practical rural applications. Saeidi and Kaisar [17], for example, studied a rural logistics network in North Florida, where routing and inventory decisions considered both transportation costs and accident risk. Their case study covered 26 rural locations and showed improvements of about 7–14% when alternative routes were considered. The study also showed that metaheuristic methods can handle larger problems in considerably less computational time than exact optimization methods, which is particularly useful when decisions need to be made within a limited time.

The problem is to recommend services to users based on their specific needs and each service's capabilities. The decision for service recommendation is modeled using a binary variable y_{ik}, as expressed in Eq. (1). Service allocation is subject to capacity constraints, as described in Eq. (3). In this context, m stands for the total number of users, and n refers to the total number of available services. The parameter μ_i reflects how many inquiries service i is capable of processing within a given time frame, while C_k represents the upper limit of requests that user k is permitted to issue.

$$y_{ik} = \begin{cases} 1, & when\ service\ i\ is\ recommended\ to\ user\ k \\ 0, & otherwise \end{cases} \quad (1)$$

$$\sum_{k=1}^{m} y_{ik} = 1 \quad \forall i = 1, \dots, n \quad (2)$$

$$\sum_{i=1}^{n} \mu_i y_{ik} \le C_k \quad \forall k = 1, \dots, m \quad (3)$$

To account for the exchange cost between services, a new variable x_{ij} is introduced. This variable indicates whether services i and j are connected to the same user. When two services are assigned to the same user, $x_{ij} = 1$; otherwise, it is 0, as defined in Eq. (4). The exchange cost becomes part of the objective function, where the cost is zero if $x_{ij} = 0$, as detailed in Eq. (5). h_{ij} represents service exchange costs between services i and j.

$$x_{ij} = \sum_{k=1}^{m} y_{ik} y_{jk} \quad \forall i, j = 1, \dots, n, \ \ i \ne j \quad (4)$$

$$\sum_{i=1}^{n} \sum_{j=1}^{n} h_{ij} \left(1 - x_{ij}\right) \quad (5)$$

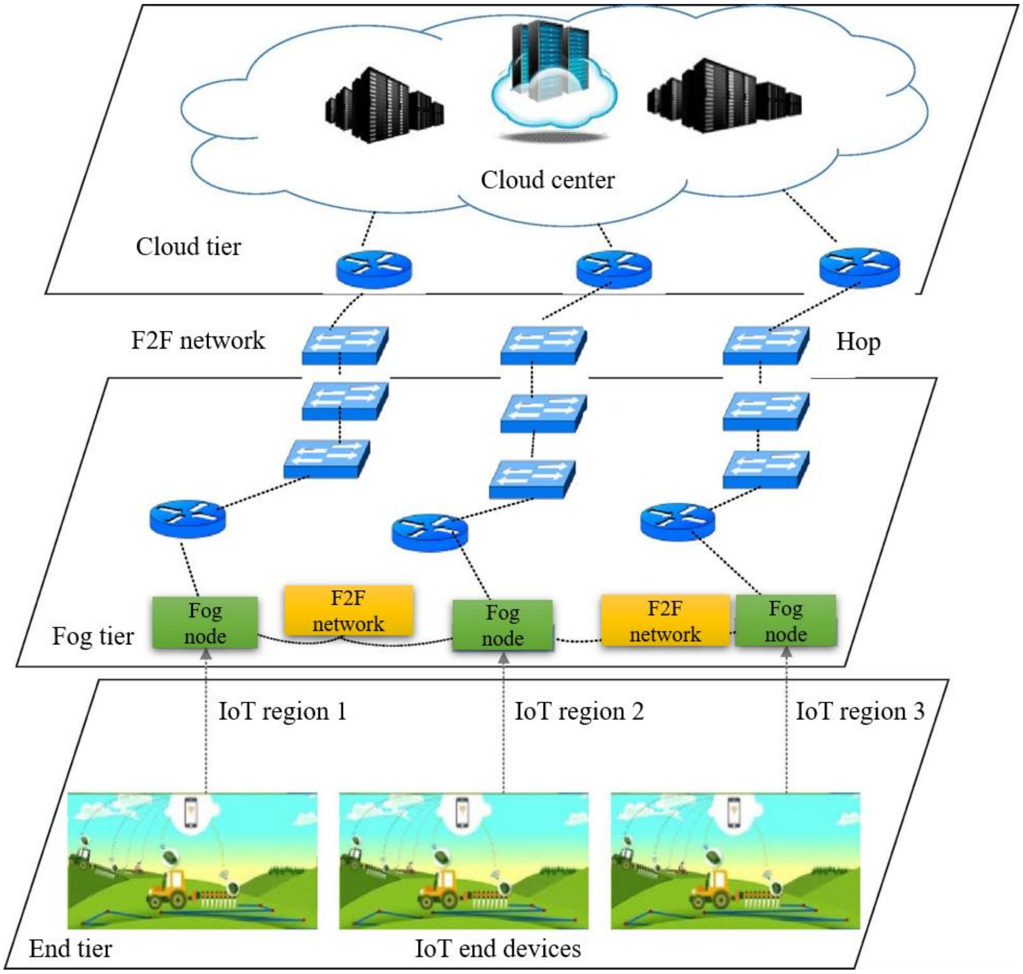


Fig. 1. An overview of the fog-based IoT architecture in agricultural environments.

Additionally, the cost of connecting user k and service i is calculated using a distance measurement between the user and the service. This is captured by Eq. (6), where c_{ik} is the connection cost and d_{ik} denotes the calculated distance between the corresponding service and user.

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ik} d_{ik} y_{ik} \tag{6}$$

Finally, the overall objective function for minimizing both connection and exchange costs is represented in Eq. (7). The objective function aggregates the connection cost, exchange cost, and user-service assignments:

$$FF_1 = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ik} d_{ik} y_{ik} + \sum_{i=1}^{n}\sum_{j=1}^{n} h_{ij}\left(1 - x_{ij}\right) \tag{7}$$

The random key method is used to represent the service recommendation results, formatted as an n×1 matrix. Each matrix entry is a floating-point number, where the whole-number component indicates the user to whom a service is assigned. When multiple services are mapped to the same user (i.e., have the same integer part), the one with the lowest decimal value is prioritized as the recommended option. For example, consider a case involving 7 services (n=7) and 3 users (m=3), producing a matrix such as (2.42, 2.10, 1.21, 3.32, 3.59, 3.80, 1.43). In this case, services 1 and 2 would be linked to user 2, services 3 and 7 to user 1, and services 4, 5, and 6 to user 3. Fig. 2 illustrates how the allocation is derived from these coordinates, emphasizing the efficiency of the random key method in optimizing service recommendations.

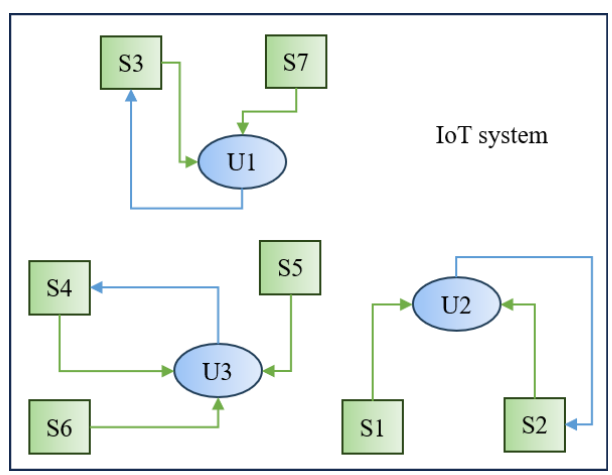


Fig. 2. Service allocation to users based on the random key method.

IV. Enhanced Grey Wolf Optimization Algorithm

GWO is a nature-inspired optimization algorithm that mimics the predatory behavior of grey wolves. The best individual is designated as the alpha (α) wolf, the second-best as the beta (β) wolf, and the third-best as the delta (δ) wolf, with the remaining individuals labeled as omega (ω) wolves. The GWO simulates the hunting behavior of the wolf pack, in which the wolves' movements are directed toward the prey's location, as mathematically expressed in Eq. (8).

$$X(t+1) = X_p(t) - A \cdot \left|C \cdot X_p(t) - X(t)\right| \tag{8}$$

where, t denotes the current iteration number, X is the position vector of the wolves, X_p is the prey's position, A and C are coefficient vectors, and $|\cdot|$ represents the absolute value. The values of the vectors A and C are determined as:

$$\begin{gathered} A = 2 \cdot \alpha . r_1 - \alpha \\ C = 2 \cdot \alpha \cdot r_2 \\ \alpha = 2 - \frac{2 \cdot t}{t_{max}} \end{gathered} \tag{9}$$

where, r_1 and r_2 represent random values within the range [0, 1], and t_{max} denotes the total number of allowable iterations. The positions of the top three wolves are updated as follows:

$$X_\alpha = X_\alpha - A_\alpha \cdot |C_\alpha \cdot X_\alpha - X| \tag{10}$$

$$X_\beta = X_\beta - A_\beta \cdot \left|C_\beta \cdot X_\beta - X\right| \tag{11}$$

$$X_\delta = X_\delta - A_\delta \cdot |C_\delta \cdot X_\delta - X| \tag{12}$$

$$X(t+1) = X_\alpha(t) + X_\beta(t) + X_\delta(t) \tag{13}$$

where, A_α, A_β, and A_δ are the vectors for the top three wolves, and C_α, C_β, and C_δ control the wolves' exploration and exploitation abilities.

Although the GWO algorithm is very effective in many optimization fields, it has difficulties in stricter cases. These include a tendency to converge prematurely to a non-optimal solution, especially in the presence of multiple local minima, combined with a slow convergence rate, which may result in high computational overhead in real-time and large-scale applications. Furthermore, the performance of GWO depends heavily on several parameters, such as population size, number of iterations, and control coefficients, which makes parameter tuning a difficult task, in particular for end users with less experience.

Focusing only on the three best agents in the hierarchy limits the capacity of the algorithm to fully investigate the search space, and consequently, the diversity can be low, and the chance of convergence to local optima can be quite high. Moreover, the basic GWO algorithm is not scalable to high-dimensional problems, is ineffective for noisy or inaccurate objective functions, and is unstable in problems with time-varying constraints or objectives. Such limitations emphasize the opportunity for superior variants to tackle such challenges and enhance the algorithm's applicability in evolving, intricate problem spaces.

To overcome the identified limitations, the enhanced EGWO framework incorporates multiple modifications to strengthen both search diversification and intensification. The weight of the α-wolf's position in Eq. (13) is adjusted to account for the fact that, during the initial search, the α-wolf may not be close to the optimal solution. This leads to a higher weight for the α-wolf during the early iterations, as shown in Eq. (14).

$$\alpha = 2 - \frac{t}{t_{max}} \cdot cos(r) \tag{14}$$

where, t_{max} is the maximum number of iterations, and r is a random number between 0 and 1. This adjustment allows the algorithm to explore more thoroughly in the early iterations and then shift to exploitation as the search progresses.

Additionally, the weights for the wolves ($\sigma_1, \sigma_2, and\ \sigma_3$) are calculated to adaptively control the transition between exploration and exploitation. These weights are formulated as:

$$\sigma_1 = cos(\theta) \tag{15}$$

$$\sigma_2 = 0.5 \cdot sin(\theta) \cdot cos(\varphi) \tag{16}$$

$$\sigma_3 = 1 - \sigma_1 - \sigma_2 \tag{17}$$

where, θ is the angle associated with the search process, and φ is a supplementary variable to control the distribution of weights. By using adaptive weights, the α-wolf initially steers the search process more strongly. However, as iterations proceed, the influence of the β- and δ-wolves gradually increases, playing a more prominent role in refining the solution.

The proposed EGWO algorithm introduces an adaptive weighting mechanism that enhances the efficiency of the baseline GWO. By allowing the wolves' weights to adapt during the search, the algorithm can smoothly switch between exploitation and exploration, thereby avoiding premature convergence and improving overall solution quality. EGWO offers a more effective approach to solving complex optimization problems and is less likely to get trapped in local optima. The algorithm's performance is demonstrated in Algorithm 1.

Algorithm 1: Pseudocode of EGWO

1: Set the maximum number of iterations t_{max} and initialize the population size N.
2: Generate an initial population randomly and set the corresponding solutions.
3: Define the parameters for the GWO, including the values for a, A, and C.
4: Select the top three individuals: $X_\alpha, X_\beta, and\ X_\delta$.
5: Repeat for each iteration from $t = 1\ to\ t_{max}$:
6: For each individual i in the population N.
7: Update the position of the current agent using Eq. (15).
8: End loop over individuals.
9: Update the coefficient a using Eq. (14), and adjust A and C using Eq. (9).
10: Calculate the fitness values of all search agents.
11: Update the positions of $X_\alpha, X_\beta, and\ X_\delta$ Eq. (10)-(12).
12: Increment the iteration count by 1: $t = t + 1$.
13: Repeat until the termination condition is met.
14: Return the best solution found.

The time complexity of the standard GWO algorithm can be analyzed based on its iterative operations, which consist of three core components: calculating the fitness of each member of the population, adjusting their positions using the best three solutions found so far, and maintaining a proper trade-off between exploitation and exploration. Assuming a population of size N, a maximum iteration count t_{max}, and a dimensionality of dim, the position updating process in each iteration requires $O(N \cdot dim)$ computations. Therefore, the total computational complexity over all iterations is $O(t_{max} \cdot N \cdot dim)$.

To enhance the performance of the proposed EGWO, a novel adaptive weighting approach is introduced, which slightly increases the computational load. This mechanism dynamically recalibrates the weight parameters σ_1, σ_2, and σ_3 by leveraging trigonometric and inverse tangent operations, and also alters the equilibrium between the exploitation and exploration stages. These modifications contribute an additional computational complexity of $O(N \cdot log\, t_{max})$ per iteration. Consequently, the entire complexity of EGWO becomes $O(t_{max} \cdot N \cdot dim + log\, t_{max})$. Although the added logarithmic component slightly increases cost, it yields noticeable gains in search performance and solution precision.

V. Results and Discussion

The performance of the proposed EGWO algorithm was examined on a collection of 10 conventional test problems, each with both fixed- and variable-dimensional unimodal and multimodal features. These problems were chosen for an in-depth exploration of the EGWO algorithm's exploring-exploiting abilities across a diverse set of complexes. The effectiveness of the EGWO algorithm was evaluated against several conventional algorithms, namely PSO, DE, Gravitational Search Algorithm (GSA), basic GWO, GWO with Crossover and Mutation (GWO-CM), GWO with Local Search (LSGWO), and Improved GWO with Dynamic Search (IGWO-DS). For a more objective comparison, a separate run for each algorithm was conducted 30 times independently, with a population of 30 individuals, for up to 500 iterations.

The selected benchmark functions corresponding to various difficulty levels are given in Table II, where F1-F4 are unimodal, and F5-F7 are multimodal. The unimodal test functions with 30 dimensions can be well exploited and explored to find global optimal solutions, while the multimodal test functions, with attention to F5–F7, have more local optima, so the exploration should be cautious to avoid possible premature convergence. Also, the functions F8–F10 present more challenging multimodal features, with a smaller differentiability set, which requires the algorithm to achieve a good balance between exploration and exploitation.

As shown in Table III, the EGWO algorithm can solve unimodal and multimodal test problems very well. In the unimodal test problems, through the dynamic variation of weight factors, the EGWO method can effectively exploit the search space with high precision and also increase the exploration and exploitation capabilities. Applying the EGWO algorithm to multimodal functions also manifests strong exploration capability with overall better performance than other algorithms. Considering specific functions F5 through F8 with several local optima, EGWO efficiently coordinates global and local searches for significantly better performance. Statistics show that, for accuracy, the standard deviation (STD) measures indicate better performance than other algorithms implemented for comparison. Through a robust exploration mechanism, with a boost from the adaptive weight factors implemented in the exploration-exploitation mechanism with EGWO, tackling challenging multimodal optimization problems can be more efficient for the algorithm.

TABLE II. SUMMARY OF BENCHMARK FUNCTIONS

Function	Formula	Dimensions	Domain	Optimal Value (fmin)
F1	$\max(y_k), \quad 1 \le k \le d$	30	[-100, 100]	0
F2	$\sum_{k=1}^{d}\sum_{l=1}^{k} y_l^2$	30	[-100, 100]	0
F3	$\sum_{k=1}^{d} y_k + \prod_{k=1}^{d} y_k$	30	[-10, 10]	0
F4	$\sum_{k=1}^{d} y_k^2$	30	[-100, 100]	0
F5	$\frac{1}{4000}\sum_{k=1}^{d} y_k^2 - \prod_{k=1}^{d} \cos\left(\frac{y_k}{\sqrt{k}}\right) + 1$	30	[-600, 600]	0
F6	$-20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{k=1}^{d} y_k^2}\right) - \exp\left(\frac{1}{d}\sum_{k=1}^{d}\cos(2\pi y_k)\right) + 20 + e$	30	[-32, 32]	0
F7	$\sum_{k=1}^{d}[y_k^2 - 10\cos(2\pi y_k) + 10]$	30	[-5.12, 5.12]	0
F8	$-\sum_{k=1}^{10}[(z - b_k)^T(z - b_k) + d_k]^{-1}$	4	[0, 10]	-10.536
F9	$5.1 \times 4\pi^2 z_1^2 + z_1 - 6 + 101 - \frac{1}{8}\cos(z_1) + 10$	2	[-5, 5]	0.398
F10	$4z_1^2 - 2.1z_1^4 + \frac{1}{3}z_1^6 + z_1^2 + z_1^2 - 4z_2^2 + 4z_2^2 + 4z_4^2$	2	[-5, 5]	-1.031

TABLE III. PERFORMANCE COMPARISON OF DIFFERENT ALGORITHMS ON BENCHMARK FUNCTIONS

Benchmark functions	Performance metrics	PSO	GWO-CM	GSA	DE	LSGWO	GWO	EGWO
F1	Minimum	5.83E+01	6.19E-01	2.33E+01	5.28E+00	7.18E-02	1.08E-07	2.91E-57
	Maximum	8.25E+01	8.46E+01	5.33E+01	1.11E+01	1.31E+00	3.06E-06	4.83E-52
	Average	7.06E+01	4.28E+01	3.97E+01	7.67E+00	5.41E-01	1.07E-06	6.23E-53
	FRT	7.62	6.79	6.31	5.43	4.05	3.13	1.18
	RT	0.27	2.53	0.21	0.22	0.18	0.09	0.14
	STD	6.48E+00	3.51E+01	9.87E+00	1.81E+00	3.28E-01	1.12E-06	1.51E-52
F2	Minimum	9.92E+03	1.62E+04	1.42E+03	3.19E+04	8.62E-01	1.25E-08	1.66E-103
	Maximum	5.17E+04	6.66E+04	1.85E+04	4.62E+04	3.89E+01	2.78E-04	6.69E-93
	Average	2.93E+04	4.95E+04	7.46E+03	3.93E+04	2.31E+01	3.03E-05	6.74E-94
	FRT	6.22	7.53	5.09	7.11	3.89	3.01	1.26
	RT	0.26	2.65	0.28	0.26	0.23	0.17	0.21
	STD	1.41E+04	1.68E+04	5.02E+03	4.62E+03	1.09E+01	8.74E-05	2.09E-93
F3	Minimum	5.55E+00	4.67E-57	7.53E-03	2.27E-03	1.13E-01	2.36E-17	2.69E-65
	Maximum	5.31E+01	3.16E-50	5.81E-02	7.07E-03	3.25E+00	1.85E-16	3.54E-59
	Average	2.62E+01	3.24E-51	1.18E-02	4.45E-03	1.05E+00	1.01E-16	4.21E-60
	FRT	7.89	2.47	5.71	5.43	7.17	3.86	1.09
	RT	0.15	2.61	0.12	0.18	0.16	0.08	0.10
	STD	1.35E+01	9.94E-51	1.71E-02	1.21E-03	1.03E+00	6.34E-17	1.11E-59
F4	Minimum	4.36E+02	4.21E-87	2.55E-01	1.71E-04	8.44E-04	2.47E-29	4.62E-126
	Maximum	4.41E+04	83.31E-75	6.85E+01	6.42E-04	1.51E+00	1.82E-27	6.78E-115
	Average	4.38E+03	1.54E-75	1.01E+01	3.02E-04	5.46E-01	7.66E-28	6.77E-116
	FRT	8.10	3.53	7.05	5.51	6.33	4.72	1.04
	RT	0.17	2.62	0.11	0.27	0.15	0.08	0.11
	STD	5.66E+03	3.02E-75	2.06E+01	1.62E-04	5.87E-01	5.35E-28	2.13E-115
F5	Minimum	6.29E+00	0.00E+00	5.66E-01	2.87E-04	5.26E-06	0.00E+00	0.00E+00
	Maximum	2.33E+01	0.00E+00	1.20E+00	4.65E-02	4.38E-02	1.72E-02	0.00E+00
	Average	1.15E+01	0.00E+00	9.24E-01	7.01E-03	1.60E-02	2.93E-03	0.00E+00
	FRT	7.75	4.82	7.14	3.23	4.01	5.69	1.15
	RT	0.17	2.56	0.18	0.14	0.18	0.12	0.10
	STD	5.01E+00	0.00E+00	2.01E-01	1.48E-02	1.51E-02	6.27E-03	0.00E+00
F6	Minimum	1.74E+01	4.42E-16	7.38E-02	3.62E-03	1.22E-02	9.27E-14	4.28E-17
	Maximum	2.01E+01	7.51E-15	2.01E+01	7.96E-03	3.51E+00	1.38E-13	4.28E-17
	Average	1.93E+01	3.60E-15	1.38E+01	5.01E-03	1.02E+00	1.06E-13	4.28E-17
	FRT	7.91	2.38	6.94	5.22	5.67	2.90	2.11
	RT	0.26	4.45	0.22	0.18	0.16	0.11	0.14
	STD	7.01E-01	2.60E-15	9.01E+00	1.33E-03	1.19E+00	1.27E-14	4.28E-17
F7	Minimum	1.05E+02	0.00E+00	1.12E+01	1.41E+02	6.99E-03	5.67E-14	0.00E+00
	Maximum	1.71E+02	0.00E+00	5.89E+01	1.68E+02	4.91E+01	1.11E+01	0.00E+00
	Average	1.42E+02	0.00E+00	3.61E+01	1.55E+02	1.86E+01	3.96E+00	0.00E+00
	FRT	7.32	2.80	7.45	4.93	6.12	3.95	1.60
	RT	0.16	2.59	0.18	0.19	0.16	0.08	0.13
	STD	2.05E+01	0.00E+00	1.89E+01	8.81E+00	1.91E+01	4.21E+00	0.00E+00
F8	Minimum	-10.5168	-10.5166	-7.6043	-10.5168	-10.5168	-10.5166	-10.5168
	Maximum	-1.8492	-5.1228	-0.9321	-10.5168	-5.1192	-10.5148	-5.1105
	Average	-7.4204	-8.6168	-4.7578	-10.5168	-8.9134	-10.5159	-8.3631
	FRT	0.14	5.11	6.82	1.17	3.73	3.92	3.44

	RT	4.53	2.52	0.15	0.18	0.13	0.06	0.11
	STD	4.08E+00	2.52E+00	2.01E+00	1.85E-15	2.558E+00	8.01E-04	2.68E+00
F9	Minimum	0.3971	0.3971	0.3975	0.3971	0.3971	0.3971	0.3971
	Maximum	0.3971	0.3971	0.4060	0.3971	0.3971	0.3971	0.3971
	Average	0.3971	0.3971	0.4009	0.3971	0.3971	0.3971	0.3971
	FRT	2.84	6.43	7.52	1.36	3.84	6.72	2.28
	RT	0.11	2.61	0.11	0.11	0.09	0.02	0.06
	STD	0.00E+00	7.25E-06	2.48E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F10	Minimum	-1.0308	-1.0308	-1.0308	-1.0308	-1.0308	-1.0308	-1.0308
	Maximum	-1.0308	-1.0308	-1.0305	-1.0308	-1.0308	-1.0308	-1.0308
	Average	-1.0308	-1.0308	-1.0308	-1.0308	-1.0308	-1.0308	-1.0308
	FRT	2.52	5.91	8.02	2.51	2.49	6.19	2.47
	RT	0.13	2.52	0.13	0.14	0.11	0.02	0.08
	STD	7.41E-17	8.13E-10	3.22E-05	0.00E+00	2.21E-16	1.04E-08	1.01E-16
Average FRT		5.83	4.77	6.80	4.19	4.73	4.40	1.76

Concerning computational complexity, although the runtime (RT) for EGWO is slightly higher because of the adaptive weight mechanism, the growth is negligible. This rise in RT is offset by better algorithm performance and the effective solving of complex problems. EGWO's adaptive weight mechanism boosts its robustness while minimizing computational burden, proving efficient and effective for practical applications in multimodal and unimodal optimization problems.

The proposed EGWO algorithm is simulated and compared with some popular benchmark techniques such as User Characteristics-Collaborative Filtering (UCCF), CRISP-DM, Association rule, CoS_FRLM, and K-Nearest Neighbors (KNN)to analyze recommendations on the simulation platform MATLAB 2020b on a 64-bit Intel Core i7-6700 processor computer with 16GB DDR3 RAM. Several parameters like Mean Absolute Error (MAE), Hamming distance, intra-list similarity, recall, precision, and ranking scores were used to assess these methods.

The MAE measure approximates the average magnitude of errors between the predicted outputs and actual user ratings. It is determined using the average of absolute errors between each prediction P_j and the corresponding actual value R_j over n samples, respectively, defined using Eq. (18). Lower MAE indicates better prediction accuracy for more reliable recommendations. MAE is computed over various sparsity levels and compared for diverse algorithms in this work.

$$MAE = \frac{1}{n}\sum_{j=1}^{n}\left|P_j - R_j\right| \tag{18}$$

For a user U_i, the ranking score S_{ij} is determined by the position Pos_{ij} of a recommended item Obj_j within the recommendation list L_i. It is calculated by normalizing Pos_{ij} over the list length, as shown in Eq. (19). An efficient recommendation system will ensure that relevant items appear in higher ranks, resulting in a lower S_{ij} value, signifying higher accuracy.

$$S_{ij} = \frac{Pos_{ij}}{L_i} \tag{19}$$

Precision measures the proportion of relevant items among the recommended set. For a user group, it is computed as the ratio of successfully recommended items A to the total recommendations T, described in Eq. (20). An increase in precision corresponds to a higher concentration of relevant suggestions within the top-ranked recommendations.

$$Precision = \frac{A}{T} \tag{20}$$

Recall calculates the system's capacity for retrieving all relevant items for a user. It is defined as the number of retrieved relevant items A divided by the number of pertinent actual items M using Eq. (21). High recall indicates the system efficiently contains most of the appropriate items for a user, without compensating for the irrelevant recommendations.

$$Recall = \frac{A}{M} \tag{21}$$

The intra-list similarity examines how similar the recommended items are within a user's list, utilizing cosine similarity as a measure. For two items Obj_m and Obj_n, their similarity Sim_{mn} is evaluated according to Eq. (22).

$$Sim_{mn} = \frac{\sum_{i=1}^{K}\alpha_{im}\cdot\alpha_{in}}{\sqrt{k(Obj_m)}\cdot\sqrt{k(Obj_n)}} \tag{22}$$

where, K represents the total number of users, and α_{im}, α_{in} are binary indicators if user i has interacted with Obj_m and Obj_n . The average similarity across all pairs within a recommendation list is used to assess diversity, with lower values being preferable for diverse suggestions.

Hamming distance predicts the disparity between recommendation lists across diverse users. It is calculated from the quantity of similar items Q between the lists of user H_x and user H_y concerning the total list size L through Eq. (23). Greater Hamming distances indicate more unique lists with minimal user similarity, encouraging diverse recommendations.

$$H_{xy} = 1 - \frac{Q}{L} \tag{23}$$

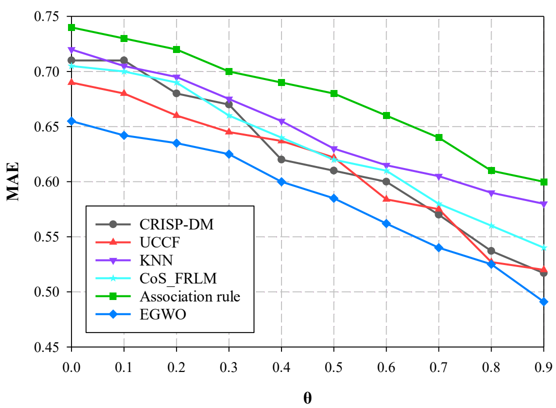
Three thresholds, namely neighborhood similarity (θ), predicted rating reliability (α), and user rating confidence (β), are used for fine-tuning recommendations. Neighborhood N_θ is determined with the help of Eq. (24), unreliable ratings are filtered out using Eq. (25), and Eq. (26) identifies reliable neighbors for making final predictions. These thresholds fine-tune the trade-off between accuracy, reliability, and diversity in the recommendation scheme.

$$N_\theta = \{U \in \mathcal{U} \mid f(U) > \theta\} \tag{24}$$

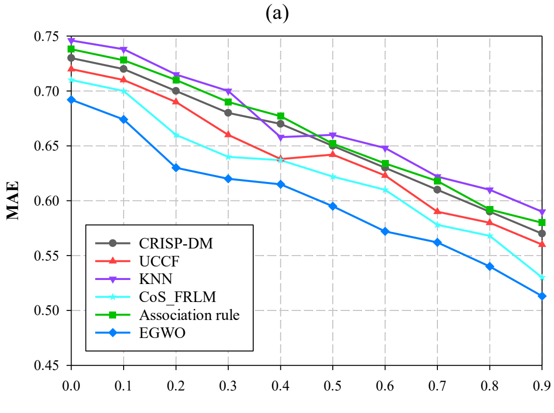
$$UR = \{R_i \mid Reliability(R_i) < \alpha\} \tag{25}$$

$$N_\beta^{\mathrm{new}} = \{U \in N_0 \mid Confidence(U) > \beta\} \tag{26}$$

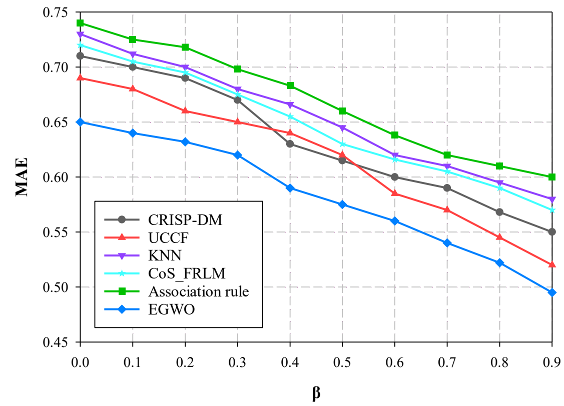
Regarding the MAE, Fig. 3(a)-3(c) illustrate the outcome, which confirms that the MAE gradually diminishes with an increase in parameters α, β, and θ. This indicates that high levels of these parameters result in more precise recommendations, rendering EGWO a reliable method of enhancing forecast accuracy. The average ranking score, shown in Fig. 4, also illustrates that EGWO outperforms as the number of customers increases, generating better recommendations. Likewise, regarding precision, Fig. 5 shows that EGWO achieves a high level compared to other methods, with an increase in the number of customers indicating a better outcome.



(a)



(b)



(c)

Fig. 3. Evaluation of MAE performance for the proposed method in comparison with existing algorithms by varying (a) threshold parameter θ, (b) reliability factor α, and (c) confidence level β, highlighting the impact of each parameter on recommendation accuracy.

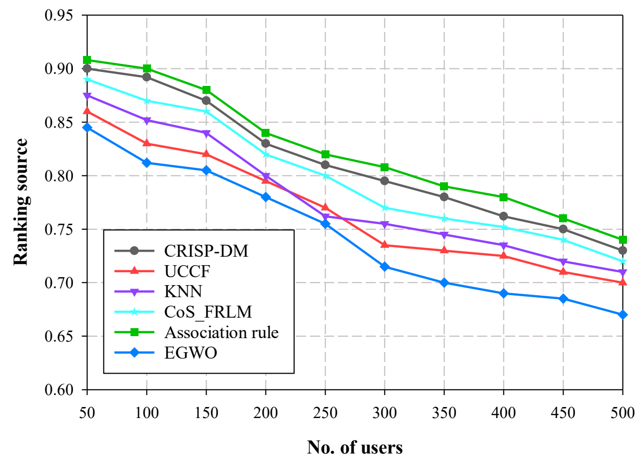


Fig. 4. Analysis of ranking score variations across different user group sizes.

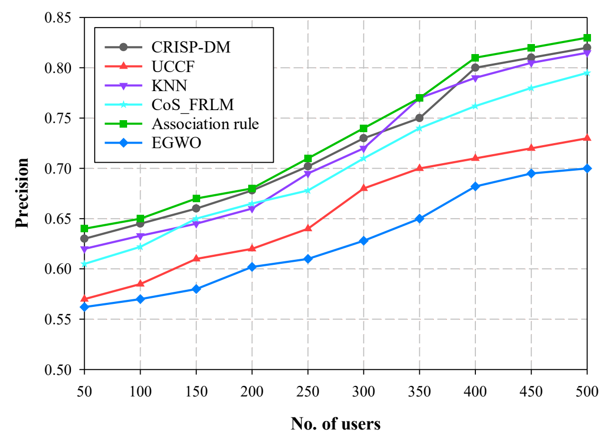


Fig. 5. Precision evaluation across different user group sizes.

The recall result in Fig. 6 also demonstrates the superiority of EGWO. The higher the number of users, the greater the advantage of benefiting from EGWO recall compared with other methods, except the UC-CF method. This suggests EGWO can successfully fetch relevant recommendations regardless of how large the collection of users is. Regarding the intra-similarity of recommendations, Fig. 7 shows that EGWO obtains the highest intra-similarity between recommended items compared with other methods.

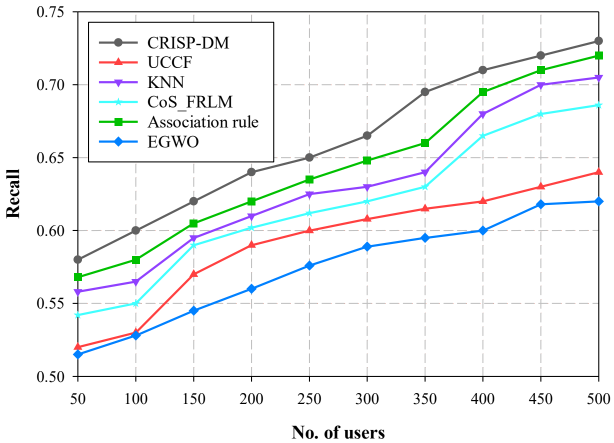


Fig. 6. Recall evaluation across different user group sizes.

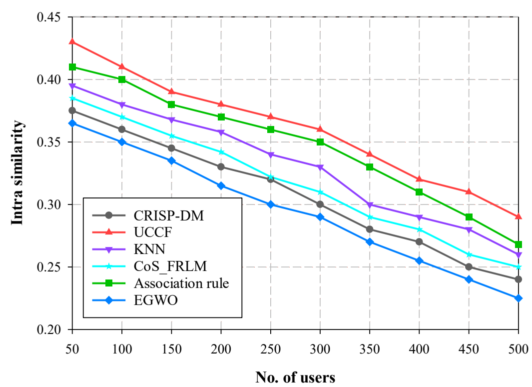


Fig. 7. Intra-similarity distance evaluation across different user group sizes.

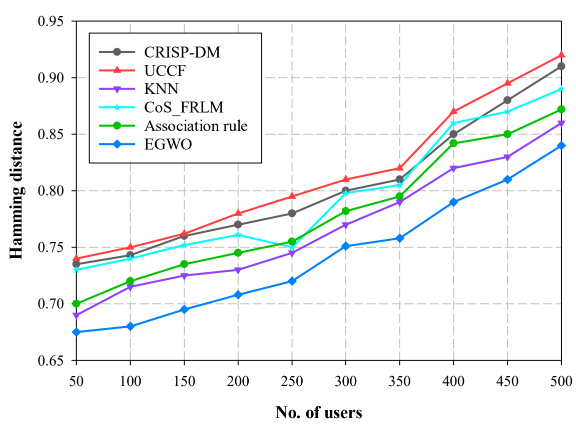


Fig. 8. Hamming evaluation across different user group sizes.

Additionally, the Hamming distance results in Fig. 8 suggest that EGWO achieves better personalization than its competitors. With the growth in the number of users, EGWO obtains better performance with a smaller Hamming distance, which proves that EGWO is more competent at personalizing recommendations than other methods. The improvements in results for all metrics imply that EGWO is an effective and efficient service recommendation algorithm for a dynamic environment.

The improvements of EGWO are due to an adaptive weighting factor, which can bring a more flexible and faster convergence process as compared to the original GWO. In the standard GWO, the exploration and exploitation phases are managed through a set of static parameters that often lead to premature convergence to a local optimum, especially in high-dimensional and multi-modal problems. By using adaptive weights for alpha, beta, and delta wolves, the EGWO modifies the effect strength of these wolves along the search process to keep a better balance between explorative (broad search) and exploitative (narrow search) behavior throughout the search process. This active correction drastically enhances convergence speed, solution quality, and robustness, with notable effectiveness for multimodal and dynamic optimization problems.

VI. Conclusion

In this study, we proposed the EGWO algorithm as a new version of the conventional GWO to solve service scheduling problems in agricultural IoT. With the inclusion of an adaptive weighting strategy, the proposed EGWO algorithm is capable of balancing the exploration and the exploitation phases within the searching process and thus exhibits better and faster convergence, higher quality of solutions, and stronger ability in solving dynamic and complex multi-objective optimization problems. EGWO also dynamically adjusts the contributions of alpha, beta, and delta wolves to escape local optima and to search a wider area, which generates best-quality service recommendations and allocation solutions in agricultural IoT applications. The comparative analysis shows that EGWO surpasses conventional optimization methods in convergence speed, stability, and general service allocation quality. However, although EGWO provides much better solutions, its computational complexity is also increased since more calculations are needed to perform the adaptive weight updates, and this can affect the scalability of the method to large- scale systems. The algorithm's efficiency and avenues for applications in large-scale real-time agricultural IoT networks should be proactively pursued in the future. Further research may be carried out on hybrid techniques with integrations of EGWO with other metaheuristic algorithms to enhance solution performance and tackle varying challenges in smart agriculture.

Acknowledgment

This work was supported by the project Research on Embedded Agricultural Internet of Things (JATC250115), 2025 Jiangsu Aviation Technical College Institute-level Project.

References

[1] 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, doi: https://doi.org/10.1002/cpe.6959.

[2] S. N. Srirama, "A decade of research in fog computing: Relevance, challenges, and future directions," Software: Practice and Experience, vol. 54, no. 1, pp. 3-23, 2024, doi: https://doi.org/10.1002/spe.3243.

[3] A. Satouf, A. Hamidoğlu, O. M. Gul, A. Kuusik, S. N. Kadry, and A. Elghirani, "A survey on task scheduling and optimization techniques for IoT-enabled UAV with Edge/Fog computing: A. Satouf et al,"

Telecommunication Systems, vol. 88, no. 3, p. 89, 2025, doi: https://doi.org/10.1007/s11235-025-01320-z.

[4] S. Rani and G. Srivastava, "Secure hierarchical fog computing-based architecture for industry 5.0 using an attribute-based encryption scheme," Expert Systems with Applications, vol. 235, p. 121180, 2024, doi: https://doi.org/10.1016/j.eswa.2023.121180.

[5] 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.

[6] N. J. Kankanamge, S. E. Mahdimahalleh, N. H. Tran, and K. Pham, "Improving DeepPolar Neural Codes via Reliability-Weighted FER-Centric Loss Functions," IEEE Communications Letters, vol. 30, pp. 867-871, 2026, doi: https://doi.org/10.1109/LCOMM.2026.3653871.

[7] Y. Yuting and G. Yuelin, "An adaptive hybrid differential Grey Wolf Optimization algorithm for WSN coverage," Cluster Computing, vol. 28, no. 4, p. 229, 2025, doi: https://doi.org/10.1007/s10586-024-04856-y.

[8] G. Ravi, M. S. Das, and K. Karmakonda, "Reliable cluster based data aggregation scheme for IoT network using hybrid deep learning techniques," Measurement: Sensors, vol. 27, p. 100744, 2023, doi: https://doi.org/10.1016/j.measen.2023.100744.

[9] L. Liu, A. Wang, G. Sun, J. Li, H. Pan, and T. Q. Quek, "Multi-objective Optimization for Data Collection in UAV-assisted Agricultural IoT," IEEE Transactions on Vehicular Technology, 2024, doi: https://doi.org/10.1109/TVT.2024.3514664.

[10] M. C. Çavdar, I. Korpeoglu, and Ö. Ulusoy, "Application scheduling with multiplexed sensing of monitoring points in multi-purpose IoT wireless sensor networks," IEEE Transactions on Network and Service Management, vol. 21, no. 1, pp. 729-744, 2023, doi: https://doi.org/10.1109/TNSM.2023.3317758.

[11] 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, doi: https://doi.org/10.1002/cpe.8091.

[12] A. S. Abohamama, A. El-Ghamry, and E. Hamouda, "Real-time task scheduling algorithm for IoT-based applications in the cloud–fog environment," Journal of Network and Systems Management, vol. 30, no. 4, p. 54, 2022, doi: https://doi.org/10.1007/s10922-022-09664-6.

[13] S. Shankar and M. Anbarasan, "An Intelligent Approach for Cloud Infrastructure With Improved Multi‐Objective Graywolf Optimization and Resource Allocation for Dynamic Virtual Machine Placement," Transactions on Emerging Telecommunications Technologies, vol. 36, no. 6, p. e70172, 2025, doi: https://doi.org/10.1002/ett.70172.

[14] K. J. Naik, "A co-scheduling system for fog-node recommendation and load management in cloud-fog environment (CoS_FRLM)," in 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), 2020: IEEE, pp. 1-6, doi: https://doi.org/10.1109/ICDABI51230.2020.9325619.

[15] F. Lin, Y. Zhou, I. You, J. Lin, X. An, and X. Lü, "Content recommendation algorithm for intelligent navigator in fog computing based IoT environment," IEEE Access, vol. 7, pp. 53677-53686, 2019, doi: https://doi.org/10.1109/ACCESS.2019.2912897.

[16] M. Altaleb, H. Deeken, and J. Hertzberg, "A data mining process for building recommendation systems for agricultural machines based on big data," in 42. GIL-Jahrestagung, Künstliche Intelligenz in der Agrar-und Ernährungswirtschaft, 2022: Gesellschaft für Informatik eV, pp. 27-32.

[17] S. Saeidi and E. Kaisar, "Rural multi-period inventory pickup and delivery problem with multi-attribute alternative routes: a case study," Transportation Planning and Technology, pp. 1-28, 2026.