Enhanced Krill Herd Algorithm Using Shuffled Frog Leaping and Meme Grouping for Multi-Objective Optimization Problems

Ruijie Wang
School of Mathematics and Statistics, Ankang University, Ankang 725000, China
E-mail: wrj6213@163.com

Keywords: sfla, meme grouping, krill herd algorithm, multi-peak optimization, logistics scheduling

Received: July 25, 2024

To solve the low-performance problem of the krill herd algorithm in the face of multi-modal optimization problems, this study proposes an improved krill herd algorithm based on a hybrid frog leaping algorithm and meme grouping method. This study analyzes the global optimization and local distribution behavior characteristics of the krill herd algorithm. Then, combined with the hybrid frog leaping algorithm, the krill individuals are optimized through meme grouping to enhance the algorithm's global and local search capabilities. This study conducted MATLAB simulation experiments to test the Schaffer and Griebank functions and compared the results with traditional krill herd algorithms. The results demonstrated that the enhanced algorithm commenced convergence at the 32nd iteration of the Schaffer function search and reached a minimum error of 3% at the 64th iteration. The conventional Krill foraging optimization algorithm reached convergence at the 72nd iteration with a minimum error of 5%. The convergence of the improved algorithm was improved by 11.1% and the error was reduced by 2%. In the search for the Griewank function, convergence commenced at the 68th iteration and was largely completed at the 130th iteration, with a minimum error of 5%. In comparison, the traditional krill foraging optimization algorithm was completed at the 143rd iteration, with a minimum error of 8%. The convergence of the enhanced algorithm was enhanced by 9.1%, and the error was diminished by 3%. This study further validated the algorithm through logistics scheduling and showed that the optimized algorithm shortened the completion time of scheduling tasks by 3 hours and reduced costs by 13,500 yuan. Research has shown that the proposed method performs outstandingly in improving global optimization capability and computational efficiency, and has practical application value.

Povzetek: Raziskava izboljšuje algoritem jat krila z združitvijo pristopa žabjih skokov in meme gručenja, kar povečuje učinkovitost pri večciljni optimizaciji in logističnem razporejanju.

1 Introduction

Global Optimal Solution (GOS) has always been an important and complex problem in optimization. Traditional optimization algorithms often fall into Local Optimal Solutions (LOS) when facing multi-objective optimization [1]. With the development of computational intelligence and swarm intelligence technologies, bionic algorithms have received widespread attention for their excellent performance in solving optimization problems [2-3]. Among them, the Krill Herd (KH) algorithm for optimizing krill foraging has demonstrated strong exploration and global optimization capabilities by simulating the foraging behavior of krill populations and has been widely applied by scholars. For example, Hamad R K et al. conducted a comprehensive analysis of the application of the KH algorithm in medicine and health and found that the algorithm has good feasibility in this field and can complete tasks such as medicine recognition and classification [4]. Neelamkavel PS used adaptive methods to improve the KH algorithm to optimize multi-objective problems in wind power generation, including optimizing various power generation costs, actual power consumption, etc. In the experiment, this method had significant advantages compared to the artificial bee colony algorithm, Particle Swarm Optimization algorithm (PSO), and flower pollination algorithm [5]. Gupta et al. improved the KH algorithm to address reactive power issues in power systems. This method considered traditional control parameters, as well as flexible AC transmission systems, combined with the objective function to minimize energy loss and operating costs. This method demonstrated superior performance in solving the reactive power problem of AC transmission system equipment used in power systems [6]. Bhatti et al. proposed a multi-objective fuzzy KH algorithm. To minimize network congestion by achieving fast convergence, this algorithm combined five objectives. In the simulation results, the algorithm showed significant improvements in transmission rate, throughput, fairness, and friendliness indicators. In addition, it also reduced packet loss, latency, queue size, energy consumption, and congestion [7]. The research progress of optimization algorithm in recent years is summarized. Many scholars have explored the application of KH algorithm and its variants in different fields, and the summarized results are shown in Table 1.

As shown in Table 1, existing optimization algorithms have obvious defects in solving complex multi-modal problems, especially in terms of convergence speed and dependence on the initial population. Although certain algorithms demonstrate proficiency in terms of accuracy and error rates, they frequently fail to identify a GOS in a shorter timeframe or exhibit constraints when confronted with more intricate and dynamic environments. Given these shortcomings, the Shuffled Frog Leaping Algorithm (SFLA) is adopted to improve the global convergence of the traditional KH algorithm and overcome the strong dependence of the algorithm. By strengthening global search and local optimization, this algorithm has significant advantages in solving the problems of slow convergence and initial population sensitivity. Specifically, SFLA's grouping strategy can effectively improve the diversity of the population and reduce the dependence on the initial population, thereby speeding up the convergence rate and reducing the error rate.

Table 1: Algorithm performance comparison table

Algorithm name	Goal problem	Convergence time	Accuracy	Error rate
Krill feeding Optimization Algorithm (KH)	Multi-modal problem	Relatively slow	0.95	0.05
Improved Genetic Algorithm (IGA)	Combination optimization, continuous optimization	Intermediate	0.96	0.04
Particle swarm Optimization (PSO)	Multi-objective problem	Fast	0.94	0.06
Leapfrog Algorithm (SFLA)	Multi-modal problem	Intermediate	0.95	0.05
Fuzzy KH algorithm	Network congestion optimization	Quicker	0.97	0.03

The innovation of this method lies in enhancing the global search and local optimization capabilities of KH algorithm through the grouping and jumping mechanism of SFLA. By combining Meme Grouping Method (MGM), the distribution and interaction process of krill are optimized, improving the performance of handling complex Multi-peak Optimization Problems (MPOPs).

2 Methods and materials

2.1 Construction of KH algorithm

KH is an algorithm that mimics the foraging behavior of krill populations. This algorithm provides a new swarm intelligence strategy to solve optimization problems during the search for GOS [8-9]. Due to its simulation of the natural behavior of krill populations, this algorithm can effectively avoid LOS and find GOS when dealing with MPOP. This algorithm can be applied in fields such as engineering optimization, machine learning and data mining, control system design, scheduling problems, etc. The KH algorithm mainly exhibits three behaviors, namely movement, foraging, and group effects [10]. In terms of mobile behavior, this study assumes that the induced movement velocity of neighbors around krill is B, which is expressed as formula (1) [11].

$$\begin{cases} B_i = B^{\max} a_i + w_n B_i^{old} \\ a_i = a_i^{local} + a_i^{t\arg er} \end{cases} \tag{1}$$

In formula (1), a_i is the induced orientation. a_i^{local} and $a_i^{t\arg er}$ are the induced orientations of neighboring krill and the current global optimal individual. w is the induced inertia weight. Formula (1) determines the direction and intensity of krill movement in the search space, which is influenced by the location of neighbors and the global best individual. For a_i^{local}, this study requires calculating the sensitive interval between krill, as shown in formula (2).

$$d_{s,t} = \frac{1}{5L}\sum_{j=1}^{L}\left\| x_i - x_j \right\| \tag{2}$$

In formula (2), L is the population size, x is the individual's location information, and d is the sensitive interval. Formula (2) determines the spatial distance at which krill form neighbor relationships. Logically, this means that if other individuals are within this interval range, they will be considered neighbors to each other and can affect the individual's movement [12]. Based on formula (3) and the distribution of krill positions, the neighbor positions of krill are circular areas generated with the particle as the center and the sensitive interval as the radius. The specific distribution is shown in Figure 1.

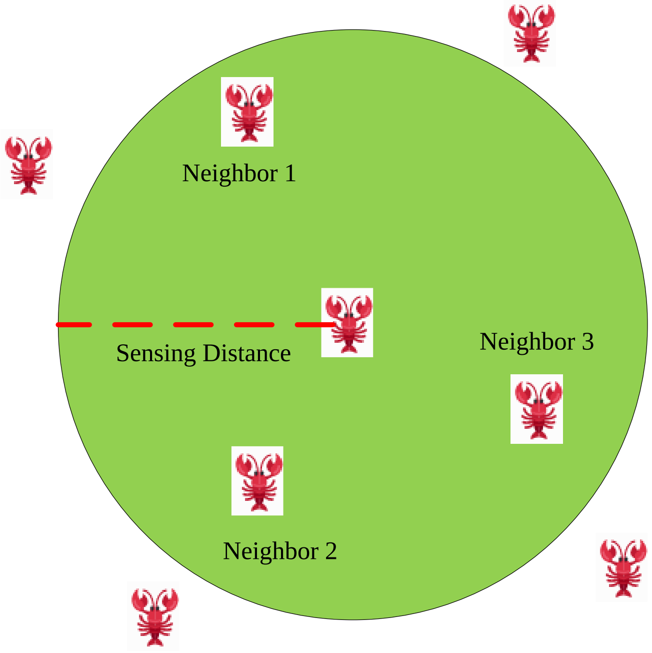


Figure 1: Neighbor distribution of krill particles

In Figure 1, each krill particle has a certain number of neighbors or companions. The location and behavior of these neighbors will affect the decisions made by the krill particles. In KH, the distribution information of neighbors helps determine how krill particles interact socially, thereby affecting the direction and distance of their movement. When other krill particles move within the sensitive interval of a particle, it will be more sensitive to changes in the position of its surrounding companions and adjust its position accordingly. The induced orientation of neighboring particles can be represented by formula (3) [13].

$$a_i^{local} = \sum_{j=1}^{L} \widehat{K}_{i.j}\widehat{x}_{i,j} \tag{3}$$

In formula (3), $\widehat{K}_{i.j}$ is the influence generated by neighboring particles. $\widehat{x}_{i,j}$ is the direction of the current particle towards its neighbors. By normalizing $\widehat{K}_{i.j}$ and $\widehat{x}_{i,j}$ in this study, particles can undergo certain movements, which helps determine their social interaction patterns and movement directions. The expression for the unitization of $\widehat{K}_{i.j}$ and $\widehat{x}_{i,j}$ is shown in formula (4).

$$\begin{cases} \widehat{K}_{i.j} = \dfrac{K_i - K_j}{K^{worst} - K^{best}} \\ \widehat{x}_{i,j} = \dfrac{x_j - x_i}{\left\| x_j - x_i \right\| + \varepsilon} \end{cases} \tag{4}$$

In formula (4), K and $\widehat{K}_{i.j}$ are the fitness values of particles and units. ε is a small positive integer that serves to prevent the calculation formula from being meaningless. Formula (4) unitizes the fitness values of krill particles, which can to some extent eliminate the differences between different fitness value scales and provide a relative measure in the algorithm. The induction direction of the globally optimal individual in the algorithm is shown in formula (5).

$$\begin{cases} a_i^{target} = Z^{best}\widehat{K}_{i,best}\widehat{x}_{i,best} \\ Z^{best} = 2(rand + t / t_{\max}) \end{cases} \tag{5}$$

In formula (5), Z^{best} is the perturbation variable. $rand$ is a random function with a value range of [0,1]. t_m is the number of iterations. Formula (5) considers perturbation variables and iteration times to calculate the induced direction of the globally optimal individual, providing a target direction for particles to move towards GOS. The individual movement speed of krill is shown in formula (6).

$$\begin{cases} V_i = v_f \beta_i + w_f V_i^{old} \\ \beta_i = \beta_i^{food} + \beta_i^{ibest} \end{cases} \tag{6}$$

In formula (6), V is the movement speed of krill. v_f is the maximum foraging speed of krill. β_i is in the KH direction. Formula (6) defines the movement speed of individual krill, which combines foraging direction and foraging inertia weights to provide speed for krill foraging activities in the search space. The individual diffusion rate of krill is shown in formula (7) [14].

$$D_i = D^{\max}(1 - t / t_{\max})\gamma \tag{7}$$

In formula (7), D_i and γ are the speed and orientation of arbitrary diffusion of krill. Formula (7) is the random diffusion rate of an individual while exploring the environment. It allows particles to perform random searches in the search space, which can help algorithms escape LOS and increase the possibility of exploring potential solutions. Finally, the particle velocity is composed of induced migration velocity, individual migration velocity, and diffusion migration velocity, as shown in formula (8).

$$\frac{dx_i}{dt} = B_i + V_i + D_i \tag{8}$$

Formula (8) combines induced velocity, individual velocity, and diffusion velocity to calculate the total velocity of particles at a specific moment. The expression for updating the position of krill particles is shown in formula (9).

$$\begin{cases} x_i(t+\Delta t) = x_i(t) + \Delta t \dfrac{dx_i}{dt} \\ \Delta t = R\sum_{j=1}^{u}(\eta - \mu) \end{cases} \tag{9}$$

In formula (9), R is the step size scaling factor. Δt is the time increment. η and μ are the upper and lower limits of decision variables. u is the dimensionality of the variable. Formula (9) updates the position of krill based on the velocity and time increment of particles, and also includes a step size scaling factor and upper and lower bounds for decision variables to ensure that particles do not exceed the search space. After the above steps, the KH algorithm performs crossover, mutation, and selection operations, as shown in formula (10).

$$\begin{cases} x_{i,m} = \begin{cases} x_{r,m} & rand_{i,m} < \sigma \\ x_{i,m} & else \end{cases} \\ x_{i,m} = \begin{cases} x_{gbest,m} + \xi(x_{p,m} - x_{q,m}) & rand_{i,m} < \zeta \\ x_{i,m} & else \end{cases} \\ x_i^{t+1} = \begin{cases} x_i^{t+1} & f(x_i^{t+1}) \geq f(x_i^t) \\ x_i^t & else \end{cases} \end{cases} \tag{10}$$

In formula (10), σ is the crossover probability, ζ is the mutation probability, ξ is the coefficient of variation, and f is the fitness function. Formula (10) allows for information exchange between solutions through crossover operations. Mutation operation introduces new features and selects individuals with high fitness as potential solutions for the algorithm based on the fitness function. The specific process of the KH algorithm mentioned above is shown in Figure 2.

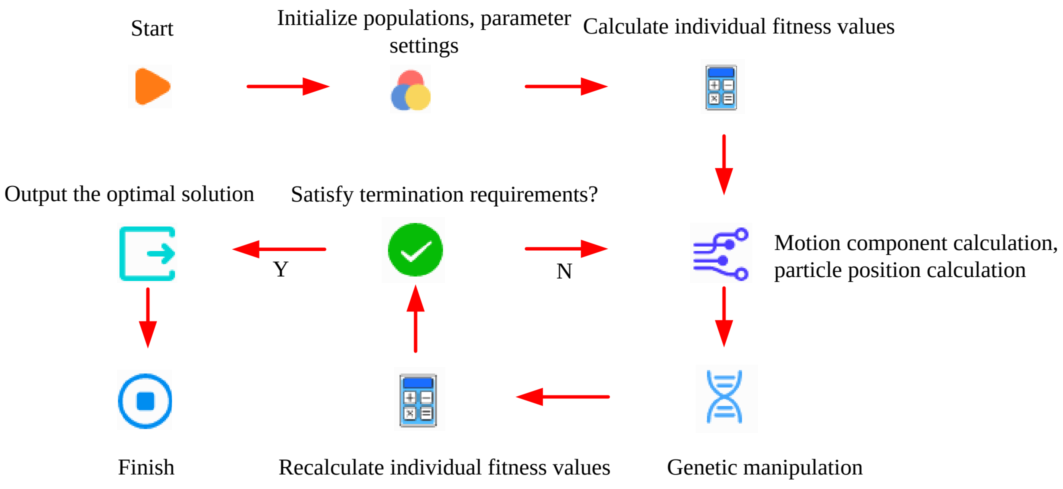

Start
Initialize populations, parameter settings
Calculate individual fitness values
Output the optimal solution
Satisfy termination requirements?
Y
N
Motion component calculation, particle position calculation
Finish
Recalculate individual fitness values
Genetic manipulation

Figure 2: KH algorithm flowchart

In Figure 2, KH first randomly generates a certain number of krill, which represents a potential solution in the problem space. Then KH calculates the fitness value of krill, and updates the position of krill based on the current fitness value of each krill and the location information of other krill in the population. Next, it adopts improvement strategies to optimize the position of krill, including local search, crossover operation, mutation operation, etc. Then, based on the position and fitness value of each krill, the inertia weight is updated. Finally, whether the termination condition is met is checked, such as reaching the maximum number of iterations or meeting specific convergence conditions.

2.2 Improved KH algorithm based on frog leaping algorithm and meme grouping

In the construction of KH, this algorithm has certain advantages. For example, it has a fast global convergence speed, can find solutions close to the optimal solution in a short time, can effectively deal with optimization problems in high-dimensional space, and has good search ability. However, the algorithm still has some limitations. The KH algorithm is sensitive to the quality and quantity of the initial population. The selection of the initial population may have a significant impact on the results. To address the shortcomings of the KH algorithm, this study uses SFLA and MGM for improvement. MGM is a meta-heuristic algorithm that combines the advantages of multiple heuristic algorithms. MGM introduces heuristic rules when solving complex problems, which can better utilize the advantages of different algorithms and improve their performance. SFLA is a heuristic optimization algorithm that simulates the behavior of frogs in searching for food. It explores and optimizes the solution space by simulating the jumping of frogs [15]. The search process of the SFLA algorithm in Figure 3.

In Figure 3, the position of each frog represents a feasible solution, and there are a certain number of stones in the frog's search range, to which each generation of frogs will be assigned. Frogs will adjust according to their position, first jumping towards the optimal position on the same stone. If the new position is worse than the original position, the frog jumps towards the global optimal position. If the position is still worse than the original position, it randomly jumps once in the solution space. Each frog has only two attributes: one is the location attribute and the other is the current location fitness value attribute. Each generation of frogs is sorted according to their position, as shown in Figure 4. In Figure 4, there are frogs on stones M1 to M5, and the frog with the worst position in each generation will jump towards the frog with the best position on the current stone, that is, F21 will jump towards F1, and F12 will jump towards F2. If frogs with poor positions do not find a better solution, they will all jump towards the global optimal position. If the new position is still poor, the frog will randomly jump to the selected position. This study assumes that in a D -dimensional space, the expression for the frog's positional fitness value is shown in formula (11) [16].

$$F_i = (f_{i1}, f_{i2}, ..., f_{iD}) \quad (11)$$

In formula (11), F_i is the frog's positional fitness value. In the initial frog population, its

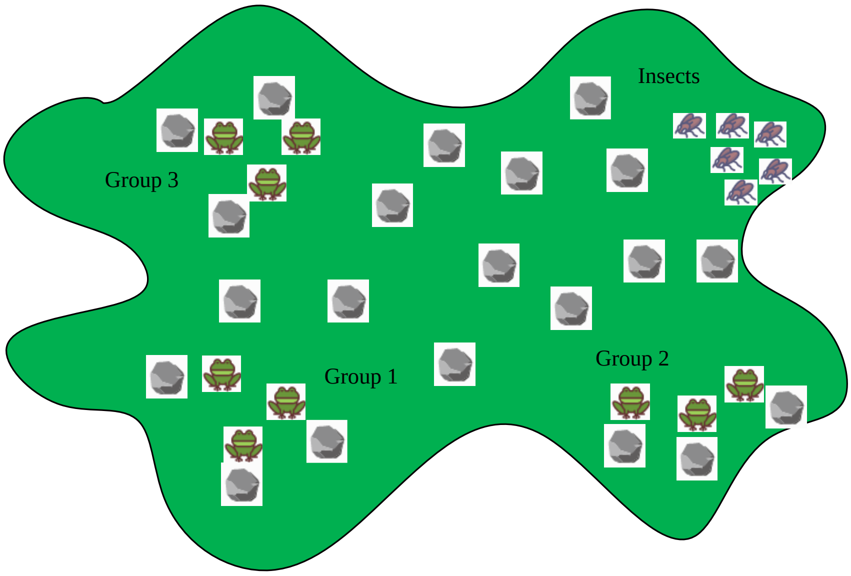


Figure 3: SFLA process

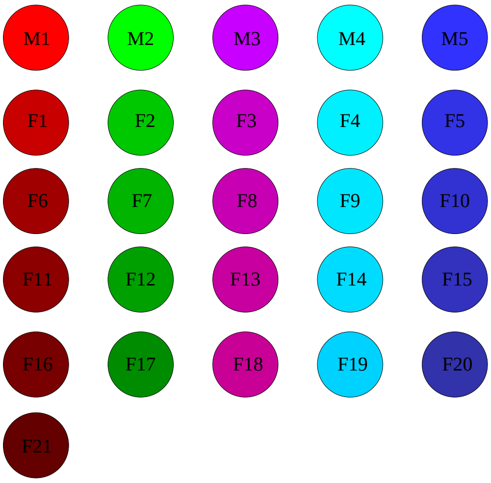


Figure 4: Position arrangement diagram of SFLA

number was set to S. Each frog is sorted according to its degree of adaptation, and the ranking decreases gradually. The entire group is split again into m subgroups, each containing n frogs, satisfying formula (12).

$$S = m \times n \tag{12}$$

Regarding the allocation method of frogs, this study starts from the first subgroup and arranges frogs in sequence, starting from the frog with the highest fitness. Then, the second frog in the second subgroup is placed, and this process is repeated until the m-th subgroup is assigned a frog with the m-th level of adaptability. After all subgroups have placed frogs in order, if there are still a number of frogs, they will be relocated from the first subgroup until all frogs have been relocated. The specific allocation expression for frogs is shown in formula (13).

$$M^k = \left\{ X_{k+m(l-1)} \in S \right\} \quad 1 \le l \le n \quad 1 \le k \le m \tag{13}$$

In formula (13), M^k is the set of frogs in the module. Assuming that the best fitness frog in different subgroups is F_b and the worst fitness frog is F_w, then in the process of population evolution, F_w will be updated, as shown in formula (14).

$$\begin{cases} C_i = rand() \times (F_b - F_w) \\ F_w^{'} = F_w + C_i (\|C_i\| \le C_{\max}) \end{cases} \tag{14}$$

In formula (14), C is the step size. In formula (14), if the best solution is obtained, the worst solution is replaced. If no better solution is obtained, the best position in the entire population is used to replace the best position in the sub-population, and then the calculation is performed. If a better solution cannot be obtained, generate a completely new random solution to replace the worst individual. Once the local search reaches the maximum number of iterations, all frogs within the sub-population are remixed. All frogs are sorted according to their fitness values and their memes are reclassified based on these sorting results. This study aims to accelerate the convergence efficiency of the KH algorithm and improve its performance. The SFLA algorithm mentioned above has been applied to the KH algorithm, and its process is shown in Figure 5.

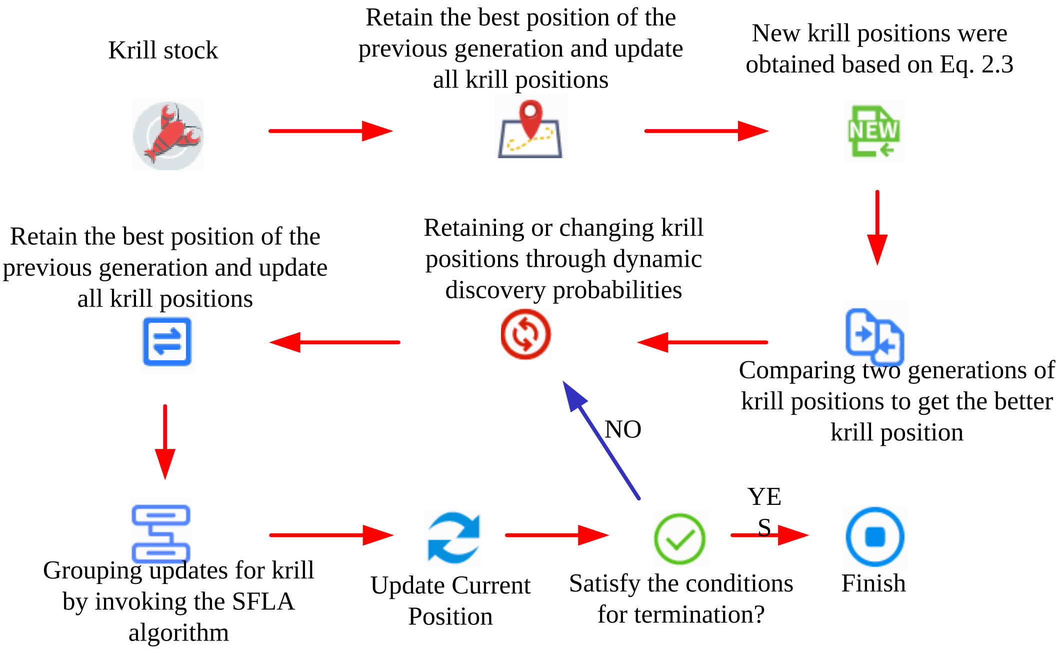

Krill stock
Retain the best position of the previous generation and update all krill positions
New krill positions were obtained based on Eq. 2.3
NEW
Retain the best position of the previous generation and update all krill positions
Retaining or changing krill positions through dynamic discovery probabilities
Comparing two generations of krill positions to get the better krill position
NO
YES
Grouping updates for krill by invoking the SFLA algorithm
Update Current Position
Satisfy the conditions for termination?
Finish

Figure 5: Flowchart of SFLA-KH algorithm

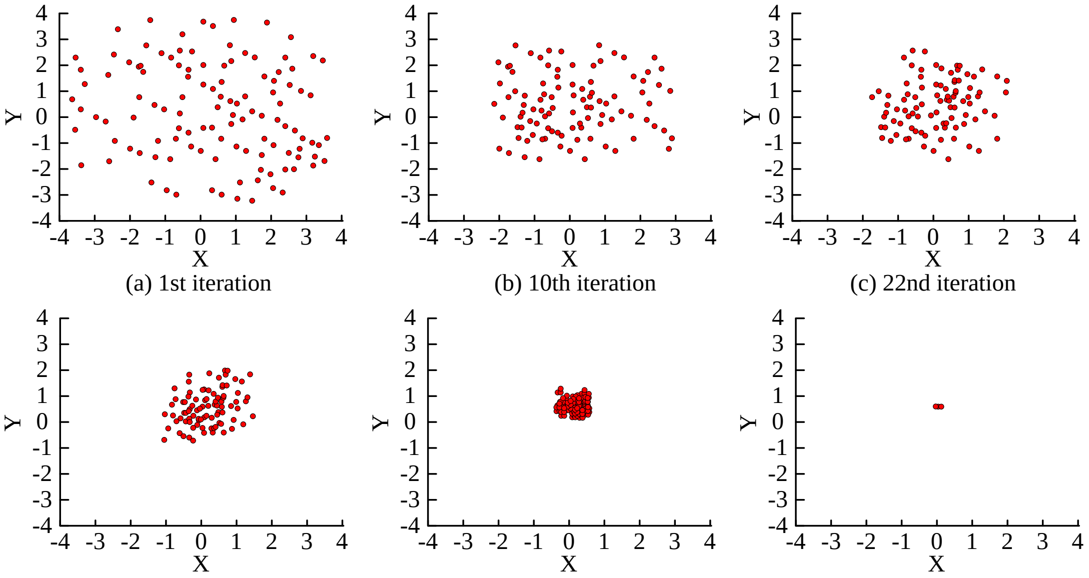
In Figure 5, the SFLA-KH mainly improves and optimizes the offspring search of KH, grouping krill through SFLA and updating their positions. This further expands the search range of the SFLA-KH algorithm and enables it to quickly jump out of LOS.

3 Results

3.1 Performance analysis of SFLA-KH algorithm

To verify the performance of the proposed model, this study conducts simulations using MATLAB. The specific experimental environment is as follows: the CPU clock speed is 2.80GHz, the memory is 8GB, and the operating system is Windows 10. The experiment conducts testing and analysis using the Schaffer function and Griebank function. Schaffer is a binary test function with a Local Minimum Point (LMP) and multiple peaks and valleys. Griebank is a commonly used multivariate testing function that also has multiple LMP functions. By testing and analyzing these two functions, the performance of the optimization algorithm in handling challenging problems with multiple LMPs can be evaluated, and the effectiveness and robustness of the proposed model can be verified. Therefore, this study sets the population size to 25 and the algorithm iteration times to 200. Schaffer is solved using an Improved Genetic Algorithm (IGA), as shown in Figure 6.

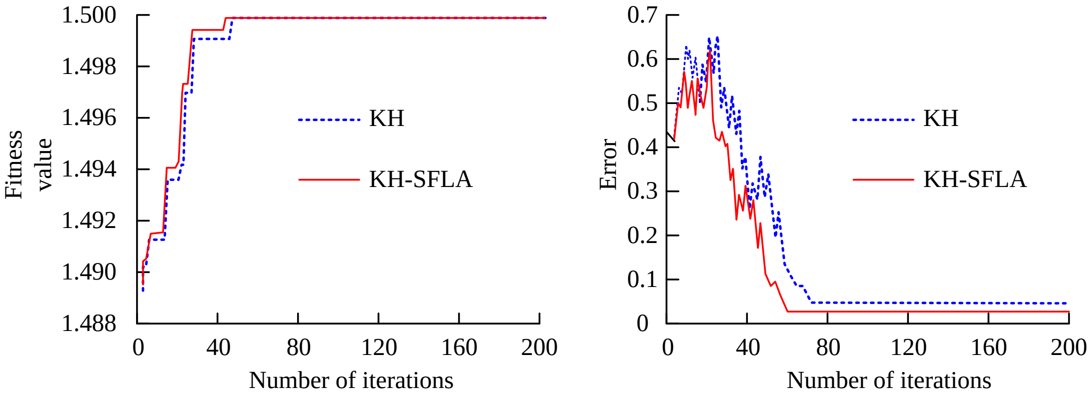
Figures 6 (a) to 6 (f) show the results of the first, 10th, 22nd, 32nd, 64th, and 98th iterations of the algorithm. The research algorithm gradually shows convergence in the 32nd iteration and basically completes convergence in the 64th iteration. In this study, Schaffer's fitness curve and iterative error curve are used as evaluation indicators of algorithm performance and are compared with the KH algorithm to verify the effectiveness and progressiveness of the improved algorithm. The results are shown in Figure 7.



(a) 1st iteration (b) 10th iteration (c) 22nd iteration

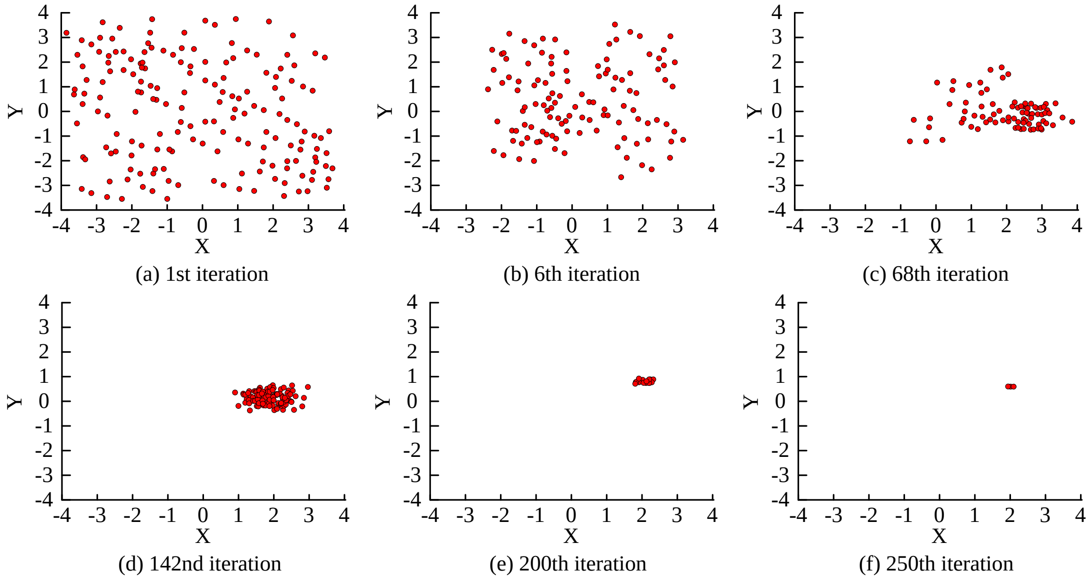
(d) 32nd iteration (e) 64th iteration (f) 98th iteration

Figure 6: Test results of schaffer function



(a) Fitness evolution curve (b) Error curve

Figure 7: The fitness and iteration error of schaffer function



(a) 1st iteration (b) 6th iteration (c) 68th iteration

(d) 142nd iteration (e) 200th iteration (f) 250th iteration

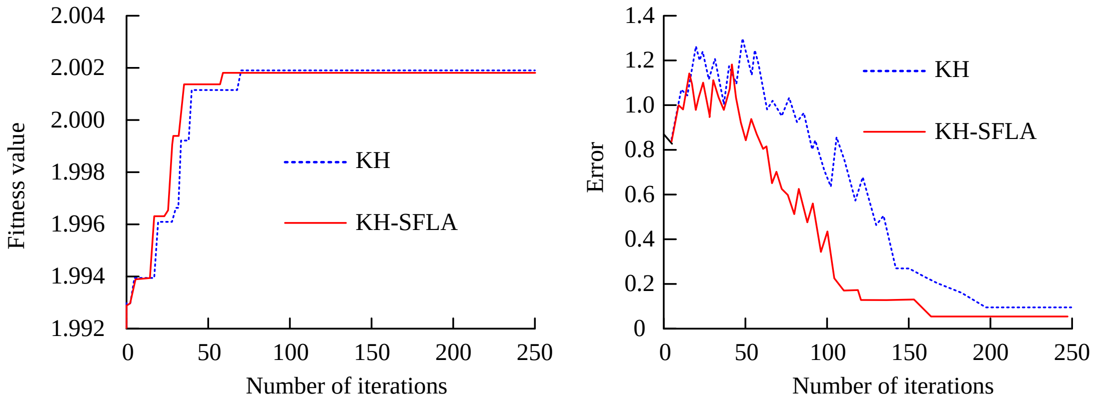
Figure 8: Test results of griebank function

In the fitness curve of Figure 7 (a), the KH algorithm begins to converge at the 48th iteration, the research algorithm begins to converge at the 42nd iteration, and the algorithm has a solution value of 1.5 for the multivariate unimodal function. In the iteration error curve of Figure 7 (b), KH reaches the minimum error at the 60th iteration, with a minimum error of 5%. The research algorithm reaches the minimum error at the 53rd iteration, with a minimum error of 3%. This indicates that the SFLA-KH algorithm can have high computational efficiency while ensuring computational accuracy. In Griebank, a function has two extrema in its domain, and the local minima have a regular arrangement. This study sets the population size to 100 and the maximum number of iterations to 250. The results of solving the Griebank function through IGA are shown in Figure 8.

Figures 8 (a) to (f) show the results of the first, sixth, 68th, 142nd, 200th, and 250th iterations of the algorithm. The SFLA-KH algorithm gradually converges at the 68th iteration and basically completes convergence at the 130th iteration. SFLA-KH can effectively find GOS when dealing with complex optimization problems containing multiple LMPs, demonstrating its superior performance. This study uses Griebank's fitness and iteration error curves as evaluation metrics for algorithm performance, as shown in Figure 9.

In Figure 9, the KH algorithm begins to converge at the 72nd iteration and reaches its minimum error of 8% at the 200th iteration. The solution value of the research algorithm for multivariate unimodal functions is 2.0. The SFLA-KH algorithm begins to converge at the 65th iteration and reaches its minimum error of 5% at the 164th iteration. Based on Figures 8 and 9, the SFLA-KH algorithm has good global optimization ability and fast convergence, and its effectiveness has been verified. To further evaluate the model, the real data set is used to compare and analyze the SVA-KH, GA, and PSO algorithms. The real data set is mainly the actual delivery order data of a large logistics company, including the number of orders, delivery routes, and estimated delivery time. The dataset consists of a total of 1,000 shipping orders with order characteristics including order ID, start and end point, estimated distance, estimated time, and priority. The evaluation results are shown in Table 2.

In Table 2, the convergence speed of the SVA-KH algorithm on the real data set is good, with 85 iterations, which is significantly lower than GA and PSO algorithms. Fewer iterations mean faster convergence. When the FLA-KH algorithm solves the logistics scheduling problem, the minimum error of the final solution is 4.5%, which is significantly better than GA, PSO, and First-Come, First-Served (FCFS) algorithms. This shows that the proposed method can provide higher solution quality in practical applications. The CPU running time of the SVA-KH algorithm is 45.7 seconds, showing high efficiency, which is lower than the 58.3 seconds of GA and 50.1 seconds of PSO. In terms of memory usage, the 150 MB required by the SVA-KH algorithm is better than that of GA and PSO, which require 200 MB and 180 MB, respectively. This also shows the advantages of SVA-KH in resource utilization. Through the evaluation on real data set, the SVA-KH algorithm shows good performance in iteration times, minimum error, and CPU running time, which proves its effectiveness in practical



(a) Fitness evolution curve (b) Error curve

Figure 9: The fitness and iteration error results of the griebank function

Table 2: Performance evaluation of the model in the real data set

Algorithm name	Number of iterations	Minimum error (%)	CPU runtime (s)	Memory Requirements (MB)
SFLA-KH	85	4.5	45.7	150
Genetic Algorithm (GA)	120	6.2	58.3	200
PSO	100	5.5	50.1	180

application.

3.2 Application performance analysis based on SFLA-KH algorithm

To further validate the performance of the research algorithm, this study analyzes SFLA-KH through logistics scheduling problems. There are the related parameter settings of SFLA-KH. The initial population size is 100; The probability of crossing is 0.6; The mutation probability usually takes a value below 0.1. However, the individual samples and model iterations in the study are relatively small, so the mutation probability is appropriately increased and set at 0.3, with a maximum genetic iteration of 700 times. Comparative experiments are conducted using the FCFS algorithms for comparative analysis [17-18]. Assuming there are 2 main stations and 12 task points in the logistics distribution scenario, with 10 logistics vehicles at the main station and the same vehicle load capacity. The site settings are shown in Figure 10.

In Figure 10, the task point information table is composed of the work vehicle, execution sequence, and task number. The homework vehicle is represented by Q. The number represents the order of tasks in the assignment task. The task number is represented by T. This study first solves the scheduling task problem using two algorithms and automatically generates a Gantt Chart for the Work Vehicles Scheduling (GC-WVS) based on the results, as shown in Figure 11.

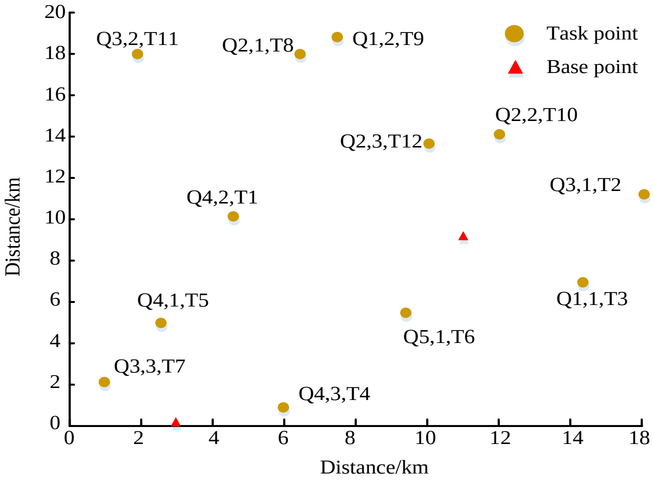
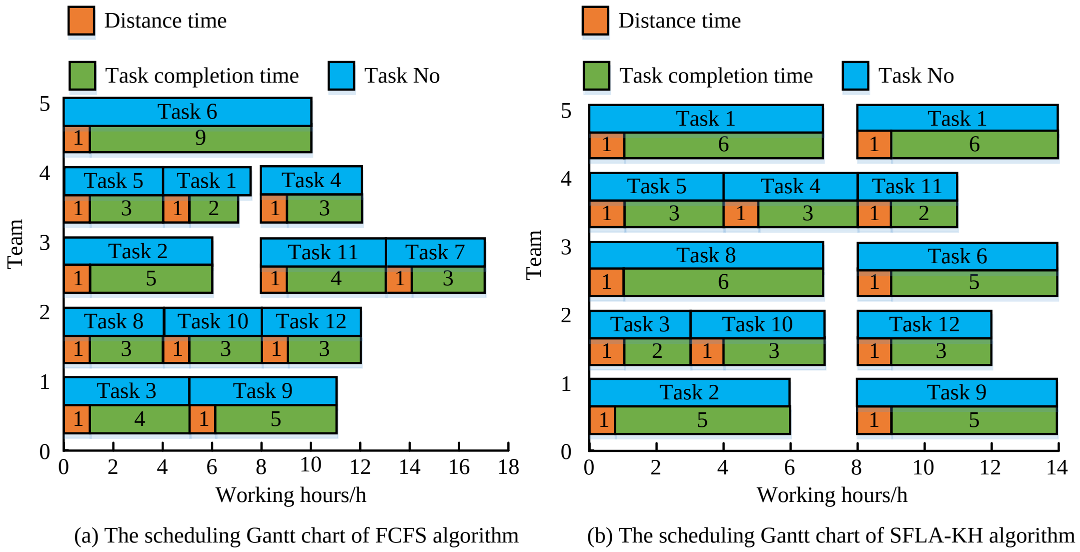
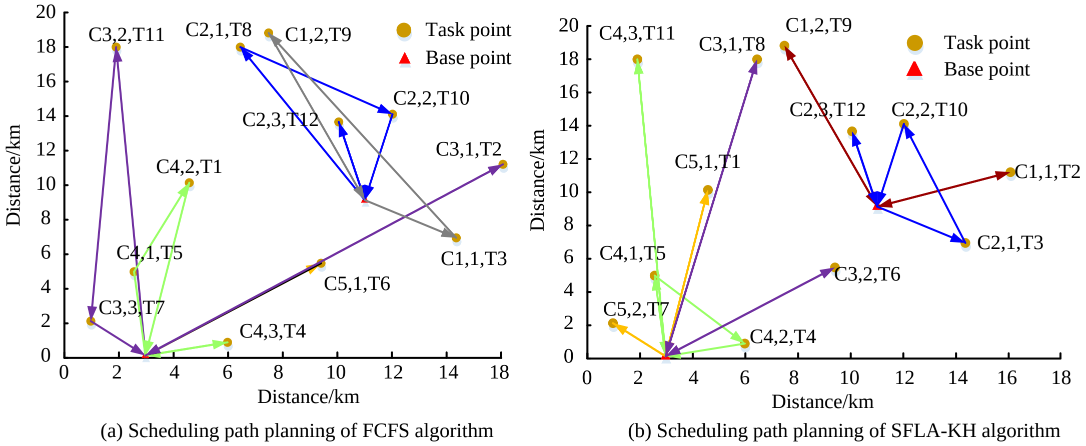


Figure 10: Visualization of site and task point plan



(a) The scheduling Gantt chart of FCFS algorithm (b) The scheduling Gantt chart of SFLA-KH algorithm

Figure 11: GC-WVS generated by different algorithms



(a) Scheduling path planning of FCFS algorithm (b) Scheduling path planning of SFLA-KH algorithm

Figure 12: Scheduling path plan of work vehicles generated by different algorithms

Figure 11 (a) shows the GC-WVS results generated by the FCFS algorithm. The Gantt chart mainly consists of three parts, including task number, travel time, and work completion time. In Figure 11 (a), each group’s layout distribution is messy, as team levels of 1 or 2 have higher execution efficiency. To minimize the cost produced during this period, most tasks are allocated to the 1st 4 groups, while the fifth group only has one task. Figure 11 (b) shows the GC-WVS result generated by the SFLA-KH algorithm. The task allocation is relatively even, and the priority order of task scheduling generally meets the requirements. Compared with the former, the scheduling model that takes into account the overall time for task-finishing and idle hourly wages ensures the total journey’s distance and handles the issue of extremely uneven distribution. The scheduling path plan generated based on the Gantt chart results is shown in Figure 12.

In the FCFS algorithm shown in Figure 12 (a), the route planning is complex and the driving path is inadvisable. After simulation, overall, the time is about 2min and 20s. The delay loss is 421,000 yuan, showing that there is a delay phenomenon. The overdue is 125,000 yuan, proving the overdue phenomenon in the overall plan. The whole time needed to finish the task is 16.8 hours, resulting in a travel cost of 25,400 yuan, for a gross of 571,400 yuan. The SFLA-KH results in Figure 12 (b) show that the optimization process completed the scheduling path in 2min and 30s. The delay loss is 408000 yuan, and the overdue loss is 124,000 yuan. There are also delays and overdue phenomena. The time to deliver the planned task is 14 hours, which is 3 hours earlier than the former. The generated driving expenditure is 25,900 yuan, which is a certain degree of reduction compared to the former. The speeding of idle hourly pay is 1,500 yuan, and the cost calculated for the former's idle hourly wages is 5,800 yuan. In contrast, the human resource cost is significantly lower in this optimization. The total cost is 557,900 yuan.

4 Discussion

The proposed SVA-KH algorithm shows remarkable performance advantages in solving multi-modal optimization problems. To fully evaluate this algorithm, it is compared with the State-of-the-art (SOTA) in terms of convergence speed, minimum error, computational complexity, and applicability. Convergence speed is an important index to evaluate the performance of optimization algorithms. In the experiment of the Schaffer function, the SVA-KH algorithm begins to converge at the 32nd iteration, while the traditional KH algorithm begins to converge at the 48th iteration. In contrast, IGA also shows relatively slow convergence in solving similar problems, usually requiring around the 55th iteration to approach the convergence state [19]. For the Griewank function, the SVA-KH algorithm converges at the 68th iteration, significantly faster than the 72nd iteration of the KH algorithm. The PSO algorithm shows the same slow performance on the same problem, and the convergence time is generally after the 80th iteration [20]. In addition, the convergence rate of the fuzzy KH algorithm is relatively slow, failing to reach convergence in a short number of iterations. Therefore, the proposed algorithm shows obvious advantages in convergence speed. In the performance of dealing with the minimum error, the SVA-KH algorithm is also superior to other methods. On the Schaffer function, the FLA-KH algorithm can achieve a minimum error of 3%, which is significantly lower than 5% of KH and 6% of PSO. The IGA also shows better accuracy, with an error of about 4%. In the test of the Griewank function, the SVA-KH algorithm again shows its advantage, with a minimum error of 5%, while the KH algorithm reaches 8%. The results show that the SVA-KH algorithm can provide higher-quality solutions when dealing with complex multi-modal optimization problems. Although the SVA-KH algorithm is superior in performance, its computational complexity is relatively high. Compared with the KH algorithm and PSO algorithm, the SVA-KH algorithm introduces an MGM algorithm and an MPOP algorithm, which leads to an increase in complexity to a certain extent. Especially in high-dimensional problems, the computational burden can be even more significant. However, this increase in computational cost is relatively acceptable compared to the increase in convergence speed and error rate. The scope of application of different algorithms varies. KH algorithm is suitable for dealing with simple multi-modal problems, but its performance deteriorates when the problem size and complexity increase. In contrast, the SVA-KH algorithm effectively improves the search ability of the algorithm in complex high-dimensional space through the introduction of meme grouping. PSO and SFLA also perform well in some cases, tending to fall into local optimal when faced with dynamic problems. Although the fuzzy KH algorithm performs well on the network congestion problem, its universality is poor. Therefore, the SVA-KH algorithm provides a more flexible and efficient solution, which is especially suitable for solving high-dimensional complex optimization problems.

The convergence speed and accuracy of the proposed method exceed other SOTA algorithms mainly due to the following two points: the introduction of MGM and the advantage of the leapfrog mechanism. The MGM effectively enhances the diversity of the population and enables the algorithm to explore the solution space more comprehensively. This avoids the early convergence of individuals to the LOS to some extent. By simulating the hopping behavior among frogs, the SVA-KH algorithm can effectively search around the LOS, thus improving the quality of understanding.

5 Conclusion

To overcome the shortcomings of the KH algorithm in terms of initial population selection sensitivity and global convergence speed, this study proposed an improved KH algorithm based on SFLA. SFLA-KH enhanced the global search and local optimization capabilities of the KH algorithm by combining SFLA's grouping strategy. Experiments have shown that SFLA-KH exhibited faster convergence speed and lower error values compared to the original KH in handling Schaffer and Griebank functions. In practical logistics scheduling problems, this algorithm significantly shortened the completion time of optimization tasks and reduced the overall scheduling cost. Although the algorithm has achieved good results, there are still some shortcomings. For example, by introducing SFLA and MGM, the algorithm improves its search performance on complex problems but also increases its complexity, which may affect its efficiency and practicality in practical applications. The dynamism and stability of this algorithm in different fields and application scenarios still need to be verified. Future research can focus on simplifying algorithms and reducing computational costs. For dynamic optimization problems, intelligent optimization algorithms that can dynamically adjust their structure and parameters can be designed to maintain efficient and stable performance in changing environments.

Funding

The research is supported by Theory and Application of several kinds of non-smooth generalized convex multi-objective programming (N0.2020AYQN06).

References

[1] P. Kaliraj, and B. Subramani (2024) Intrusion detection using krill herd optimization based weighted extreme learning machine, *Journal of Advances in Information Technology*, vol. 15, no. 1, pp. 147-154. https://doi.org/10.12720/jait.15.1.147-154

[2] T. Mahmood, and Z. Ali (2022) Prioritized muirhead

mean aggregation operators under the complex single-valued neutrosophic settings and their application in multi-attribute decision-making, *Journal of Computational and Cognitive Engineering*, vol. 1, no. 2, pp. 56-73. https://doi.org/10.47852/bonviewJCCE2022010104

[3] N. C. Cruz, S. Puertas-Martín, J. L. Redondo, and P. M. Ortigosa (2023) An effective solution for drug discovery based on the tangram meta-heuristic and compound filtering, *Informatica*, vol. 34, no. 4, pp. 743-769. https://doi.org/10.15388/23-infor535

[4] R. K. Hamad, and T. A. Rashid (2023) Current studies and applications of krill herd and gravitational search algorithms in healthcare, *Artificial Intelligence Review*, vol. 56, no. 1, pp. 1243-1277. https://doi.org/10.1007/s10462-023-10559-4

[5] P. S. Neelamkavil (2023) Development of optimal placement and sizing of FACTS devices in power system integrated with wind power using modified krill herd algorithm, *COMPEL-The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, vol. 42, no. 6, pp. 1408-1433. ttps://doi.org/10.1108/compel-12-2021-0502

[6] V. K. Gupta, S. K. Mishra, and R. Babu (2024) Solution of reactive power planning with TCSC and UPFC using improved krill herd algorithm, *Transactions of the Indian National Academy of Engineering*, vol. 9, no. 1, pp. 87-99. https://doi.org/10.1007/s41403-023-00428-5

[7] K. A. Bhatti, S. Asghar, and S. Naz (2024) Multi-objective fuzzy krill herd congestion control algorithm for WSN, *Multimedia Tools and Applications*, vol. 83, no. 1, pp. 2093-2121. https://doi.org/10.1007/s11042-023-15200-8

[8] A. O. Abdalrahman, D. Pilevarzadeh, S. Ghafouri, and A. Ghaffari (2023) The application of hybrid krill herd artificial hummingbird algorithm for scientific workflow scheduling in fog computing, *Journal of Bionic Engineering*, vol. 20, no. 5, pp. 2443-2464. https://doi.org/10.1007/s42235-023-00389-z

[9] E. Bas, and A. Ihsan (2023) Gray wolf and krill herd optimizations: performance analysis and comparison, *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, vol. 29, no. 7, pp. 711-736. https://doi.org/10.5505/pajes.2023.38739

[10] Z. Gao (2023) A novel long and short-term memory Network-Based Krill Herd Algorithm for explainable art sentiment analysis in interior decoration environment, *Journal of Cases on Information Technology (JCIT)*, vol. 25, no. 1, pp. 1-13. https://doi.org/10.4018/JCIT.324602

[11] P. S. Apirajitha, and R. R. Devi (2023) A novel blockchain framework for digital forensics in cloud environment using multi-objective krill Herd Cuckoo search optimization algorithm, *Wireless Personal Communications*, vol. 132, no. 2, pp. 1083-1098. https://doi.org/10.1007/s11277-023-10649-0

[12] K. Parthiban, Y. V. Rao, B. Harika, R. Kumar, A. Shaik, and S. Shankar (2023) Diagnose crop disease using Krill Herd optimization and convolutional neural scheme, *International Journal of Information Technology*, vol. 15, no. 8, pp. 4167-4178. https://doi.org/10.1007/s41870-023-01417-1

[13] S. Sivamohan, S. S. Sridhar, and S. Krishnaveni (2023) TEA-EKHO-IDS: An intrusion detection system for industrial CPS with trustworthy explainable AI and enhanced krill herd optimization, *Peer-to-Peer Networking and Applications*, vol. 16, no. 4, pp. 1993-2021. https://doi.org/10.1007/s12083-023-01507-8

[14] Y. Li, and L. Zheng (2023) An optimisation method of urban road green space landscape layout based on leapfrog algorithm, *International Journal of Environmental Technology and Management*, vol. 26, no. 6, pp. 457-469. https://doi.org/10.1504/ijetm.2022.10052203

[15] B. Zhang, and X. Wang (2024) A wireless sensor network node redeployment method based on improved leapfrog algorithm, *International Journal of Information and Communication Technology*, vol. 24, no. 1, pp. 33-47. https://doi.org/10.1504/ijict.2024.135313

[16] J. Zheng, Y. Zeng, Z. Zhao, W. Liu, H. Xu, and S. Ji (2023) A semi-implicit parallel leapfrog solver with half-step sampling technique for FPGA-based real-time HIL simulation of power converters, *IEEE Transactions on Industrial Electronics*, vol. 71, no. 3, pp. 2454-2464. https://doi.org/10.1109/TIE.2023.3265042

[17] M. I. Abdillah, and M. D. Irawan (2023) Implementation of the first come first served algorithm in the futsal field booking application using extreme programming, *ZERO: Jurnal Sains, Matematika dan Terapan*, vol. 7, no. 2, pp. 182-191. https://doi.org/10.30829/zero.v7i2.19163

[18] H. A. Shehadeh, H. M. J. Mustafa, and M. Tubishat (2022) A hybrid genetic algorithm and sperm swarm optimization (HGASSO) for multimodal functions, *International Journal of Applied Metaheuristic Computing (IJAMC)*, vol. 13, no. 1, pp.1-33. https://doi.org/10.4018/ijamc.292507

[19] S. T. Shishavan, and F. S. Gharehchopogh (2022) An improved cuckoo search optimization algorithm with genetic algorithm for community detection in complex networks, *Multimedia Tools and Applications*, vol. 81, no. 18, pp. 25205-25231. https://doi.org/10.1007/s11042-022-12409-x

[20] J. Popper, and M. Ruskowski (2022) Using multi-agent deep reinforcement learning for flexible job shop scheduling problems, *Procedia CIRP*, vol. 112, no. 1, pp. 63-67. https://doi.org/10.1016/j.procir.2022.09.039