Journal of Network Intelligence ©2023 ISSN 2414-8105 (Online)
Taiwan Ubiquitous Information Volume **8**, Number **2**, May **2023**

Migration Search Algorithm: A Novel Nature-Inspired Metaheuristic Optimization Algorithm

Xinxin Zhou*

School of Computer Science
Northeast Electric Power University
Jilin, Jilin, China
zxx51@qq.com

Yuechen Guo

School of Computer Science
Northeast Electric Power University
Jilin, Jilin, China
gyc1998625@gmail.com

Yuming Yan

School of Computer Science
Northeast Electric Power University
Jilin, Jilin, China
yumingyandoffodil@163.com

Yuning Huang

School of Computer Science
Northeast Electric Power University
Jilin, Jilin, China
m18757944766@163.com

Qingchang Xue

School of Computer Science
Northeast Electric Power University
Jilin, Jilin, China
1780922632@qq.com

*Corresponding author: Xinxin Zhou

Received July 4, 2022; revised September 10, 2022; accepted November 9, 2022

Abstract. *The Migration Search Algorithm (MSA) is a novel meta-heuristic optimization algorithm put forward in this paper. This algorithm is based on the way individuals communicate with one another and the dynamic migration behavior of animal populations as they explore the world. Two behaviors of animal populations are simulated by the algorithm during migration: migration movement and station maintenance. These two behaviors can effectively ensure the survival of population as well as the survival of individuals. In addition, we included this mechanism in our algorithm because information dissemination is common among animal populations and plays an important role in achieving effective migration and foraging. There are many groups in each population, and each group has three types of individuals, namely leaders, followers, and adherents. Twenty-six (twelve multimodal and fourteen unimodal) standard norm functions are commonly used in the optimization and twelve IEEE CEC2014 test functions are used for the test norm for the MSA. Comparison analysis also verifies MSA with Gravitational Search Algorithm (GSA), Particle Swarm Optimization (PSO), and Artificial Bee Colony (ABC). Experimental results reveal that, in contrast to the above-mentioned selected algorithm, Migration Search Algorithm owns good performance and competitiveness. Further, the paper applies the algorithm to the optimization issue in text classification, the results show that the algorithm exhibits better optimization performance than other existing optimization algorithms.*
Keywords: optimization, unconstrained optimization, metaheuristic algorithm, migration search algorithm, MSA

---

1. **Introduction.** Natural world is filled with various biological behaviors existing to achieve different purposes. When Sterna paradisaea fly from their Arctic breeding grounds to Antarctica, when animals in East Africa move from Tanzania to Kenya, both biologists and climatologists recognize that animal migration is taking place and it is a significant part of the biological niche. Migratory animals are found in the main branches of the animal kingdom, and the behavior of migration occurs in a variety of media where they move by flying, walking, or swimming. The main goal of all these behaviors is to allow animals to complete their reproduction and life.

Optimization problems widely exist in the fields of engineering [1, 2], and economics [3]. In recent years, more and more complicated optimization problems existing in the real world have been proposed in these fields. To solve such complex optimization problems, researchers try to use heuristic algorithms to solve the problem. The heuristic algorithm is to solve complex computing tasks through iterative methods by imitating behavior patterns and social phenomena observed in nature, and has achieved remarkable results, overcoming many engineering optimization problems that were originally difficult to solve [4, 5, 6, 7].

In the past several decades, scholars have studied dozens of natural-heuristic optimization algorithms that simulate certain biological behaviors or physical phenomena. For instance, the Particle Swarm Algorithm [8] is designed by imitating the social behavior of birds, the Artificial Bee Swarm Algorithm [9] is designed by imitating the foraging process of bees, the Simulated Annealing Algorithm [10] is inspired by metallurgical technology, and the Gravitational Search Algorithm [11] based on the law of universal gravitation. In addition, there is Genetic Algorithm [12], Differential Evolution Algorithm [13], and the like. In addition, these algorithms are successfully applied to many domains, such as program control [14, 15], biomedicine [16, 17], signal processing [18, 19], and image processing [20]. Meta-heuristics are very simple and easy to understand, and most of their inspirations come from very simple concepts. In addition, most meta-heuristics algorithms have derivation-free mechanisms.

According to the NFL theorem [21], some heuristic algorithms may show excellent performance in specific problems, but they are mediocre on other problems. Although different meta-heuristic algorithms have differences, the common point is that the search process is divided into two stages: exploration and development. In the exploration process, it is necessary to explore the promising regional space as comprehensively as possible, so the algorithm needs a stochastic operator to search the space globally and stochastic. The development stage refers to a local search within a certain range of the area obtained in the exploration stage. Due to the stochastic nature of the heuristic algorithm, how to find a break-event point in these two stages is a huge challenge. This research proposes a new meta-heuristic algorithm for unconstrained majorization problems, the Migration Search Algorithm, which simulates the dynamic migration behavior of animal populations and effective communication methods to figure out majorization problems in the real world. We hope that through the mathematical modeling of biological population migration behavior, we propose a new meta-heuristic algorithm inspired by biological migration, and hope to use this algorithm to deal with problems in real life, which is of great significance for us to solve problems in real life, research, and engineering applications in the field of scientific research.

The details of each chapter in this article are as follows: Section 2 introduces the literature review of meta-heuristics majorization algorithms. Section 3 summarizes the inspiration sources of the MSA algorithm. Section 4 expounds the principle of the MSA algorithm detailly and gives algorithm flow. Section 5 conducted a detailed experiment on the performance of the algorithm. Section 6 gives application of the algorithm in the hyper-parameter optimization of the text classification algorithm based on transfer learning and makes a brief discussion. In section 7, we summarize the work of this paper and look forward to the future research work.

2. **Literature Review.** Meta-heuristics are generally divided into three categories: algorithm that simulates the genetic evolution of organisms, algorithms derived from population intelligence simulation, and algorithms that simulates physical phenomena. For that may be based on genetic evolution, the most representative of them must be Genetic Algorithm, that was first put forward in 1992 [12] by Holland, which had a profound impact at that time. It mainly comes from understanding of evolutionary principle. It can be understood as a computational model that imitates genetic process and world selection for Darwinian natural cycle theory. It was inspired by a simulation of evolution. The method begins from initial solutions which are stochastically generated, which are population. The issues solution is Individual, which is chromosome. That create a new generation of individuals through operations. Due to natural selection, the excellent genetic factors of higher fitness individual are likely inherited to next generation, so the new population is better than the previous generation. In this way, after several times, it will approach the best set, that may represent first and second choice to solve an issue. Another typical example is the Differential Evolution Algorithm [13], which is the same as other evolutionary algorithms. It starts with a randomly initialized population, then generates a new population through cross-mutation selection operation, and repeats the process until the stop condition is satisfied. The advantage of Differential Evolution Algorithm is that it adopts floating-point coding, so the algorithm performs better in continuous optimization.

Second category algorithms are based on physical phenomena. This algorithm draws inspiration from physical phenomena and solves optimization problems by simulating physical rules, such as Simulated Annealing Algorithms [10]. The earliest idea of the simulated annealing algorithm was proposed by Metropolis in 1983, and it was widely used in engineering later. At high temperature, interior prana for object atoms thermal motion can be stronger. With increase of temperature, the interior prana diminishes internal energy increases, and the internal structure will become chaotic. When the solid is cooled, the internal particles tend to be ordered as the temperature decreases, it will reach a steady state at the final room temperature, and energy inside becomes smallest. Another typical algorithm is the Gravitational Search Algorithm [11]. In the Gravitational Search Algorithm, each feasible solution represents a particle with mass. location's fitness can describe the quality. The particles attract each other under the action of gravity and gather around the particles with higher mass. Because the Gravitational Search Algorithm is an isolated system, this algorithm that explores the solution space through gravity can effectively ensure that it converges to an optimal position.

The third category is from group intelligent, which form a series of new ways to work out traditional complex issues by simulating biological groups. Typical examples are Particle Swarm Algorithm [8], which simulates flocking flight behavior of birds and uses information sharing between individuals to perform a collaborative search on problems. The structure is simple, the speed is quick and it's not simple to get stuck in partly optimum, so what has been widely used since it was proposed. Another typical example is the Artificial Bee Colony Algorithm (ABC) [9]. At this stage, many algorithms that were on account of group intelligent have been proposed: Artificial Fish-swarm Algorithm [22], Ant Colony Optimization [23], Firefly Algorithm [24], Bat Algorithm [25], Grey Wolf Optimizer [26], Whale Optimization [27]. In addition to the above algorithms, Sine Cosine Algorithm [28], Cache Optimization [29], and Butterfly-inspired Algorithm [30] also can be applied in many fields.

Based on the above content, it can be found that these algorithms search for the optimal solution position by simulating the intelligent behavior of biological organisms living in truffles natural. Because it is similar with the way animals obtain resources in the living space. To a large extent, this mechanism simulates the movement of animals to search for the solution space, and uses the social behavior of the population to conduct intelligent navigation.

3. **Inspiration.** Animal migration is usually a relatively long-distance movement of certain animals on a seasonal basis [31]. It is found in all major animal populations, including birds, mammals, fish, reptiles, amphibians, insects, and crustaceans [32]. The reason why animals migrate is probably because of the climate, food supply, seasonal change, or breeding [33]. Animal migration behavior is an adaptive phenomenon, which can satisfy the environmental conditions they need in a specific period of life, ensuring the individuals' survival and the race's prosperity.

In order to distinguish the general movement and migration movement of species in their habitat, we divide the movement of species into two types. The first is the movement within the population and in the habitat is called station retention [34]. The most prominent manifestation of this kind of behavior is foraging. The population presents tortuous and repetitive characteristics on a small-time scale and spatial scale. Individuals frequently change their routes during foraging, but their range of activities is still in their habitats. Another type of movement is when individuals permanently exceed the range of activity in the current habitat. This movement contains an element of exploration, and it comes to a halt when a suitable new home is discovered. It is worth mentioning that animal migration behavior has an obvious tendency of convergence, one of the most typical is the seasonal migration of birds [35]. Figure 1 shows the migration route of some birds in East Asia and Australia.

It is difficult to determine the cause of this phenomenon, but this convergence behavior is certainly caused by the spread of information. Information dissemination means that information on high-quality habitats or food sources flows from individuals with information to individuals without information. Many natural behaviors can lead to the spread of information, such as bees tell the location or number of honey sources by dancing [36]. The imprinting behavior of some newly hatched young birds and new born mammals [37] (learning to recognize and follow the first moving object they see, which is usually their mother). This kind of information transmission can effectively ensure the reproduction of the whole population. Intriguingly, many animal populations [38] have hidden social hierarchies, which will affect their foraging and migration behavior to a certain extent. In general, the population has a leader who occupies the majority of the resources, such as food, mating rights, and so on. In addition, the leader is also responsible for patrolling territory, making decisions, leading, and ensuring the population's survival and reproduction; the remaining individuals will be fallen into two categories: One is the strong individuals in the population, who help the leader to capture or obtain food; the other is the old and young individuals in the population, who choose to cling to the leader or the dominant individual in the population to obtain survival opportunities.

The phenomenon that animals change the living environment of the population through migration and other behaviors to seek a greater living quality is the main source of inspiration for the Migration Search Algorithm (MSA algorithm) proposed in this article. At the beginning of the algorithm, all individuals are evenly distributed in space and spontaneously forms different small groups. After the search process, individuals tend to approach the leader of the group, who often occupies the vast majority of resources. They also explore different areas of the living environment as they move. However, as time goes on and the environment changes, food will become scarce, forcing the population to temporarily restrain its ecological demands and start long-distances migration. In the process of migration, the population often relies on the transmission of information to determine the direction of migration.

4. **Migration Search Algorithm.**

For the sake of mathematically model migration behavior of animal populations, we divide algorithm's initial population into several groups, each containing N individuals to imitate the habitat's complex population composition. The N individuals in the group are divided into three categories: leaders, assistants, and dependents. Individuals in the population begins at a random initial position in D-dimensional space, and their motion behavior is represented by a D-dimensional vector.

4.1. **Random Initialization.**

Suppose the population is divided into m groups, each containing 3 individuals. The following matrix can be used to represent the positions of all individuals in each group:

$$X^m = \begin{pmatrix} x_{1,1}^m & \cdots & x_{n,1}^m \\ \vdots & \ddots & \vdots \\ x_{1,d}^m & \cdots & x_{n,d}^m \end{pmatrix} \tag{1}$$

In d-th dimension, $x_{n,d}^m$ represents the location of n individual in the m group in population. Equation (2) is used to allocate individual's initial position in the solution space:

$$x_{n,d}^m = LB + U(0,1) \times (UB - LB) \tag{2}$$

Lower bound and upper bound of the dimensionality in the solution spatial are represented by UB and LB, and $U(0,1)$ represents stochastic number uniformly distributed between zero and one.

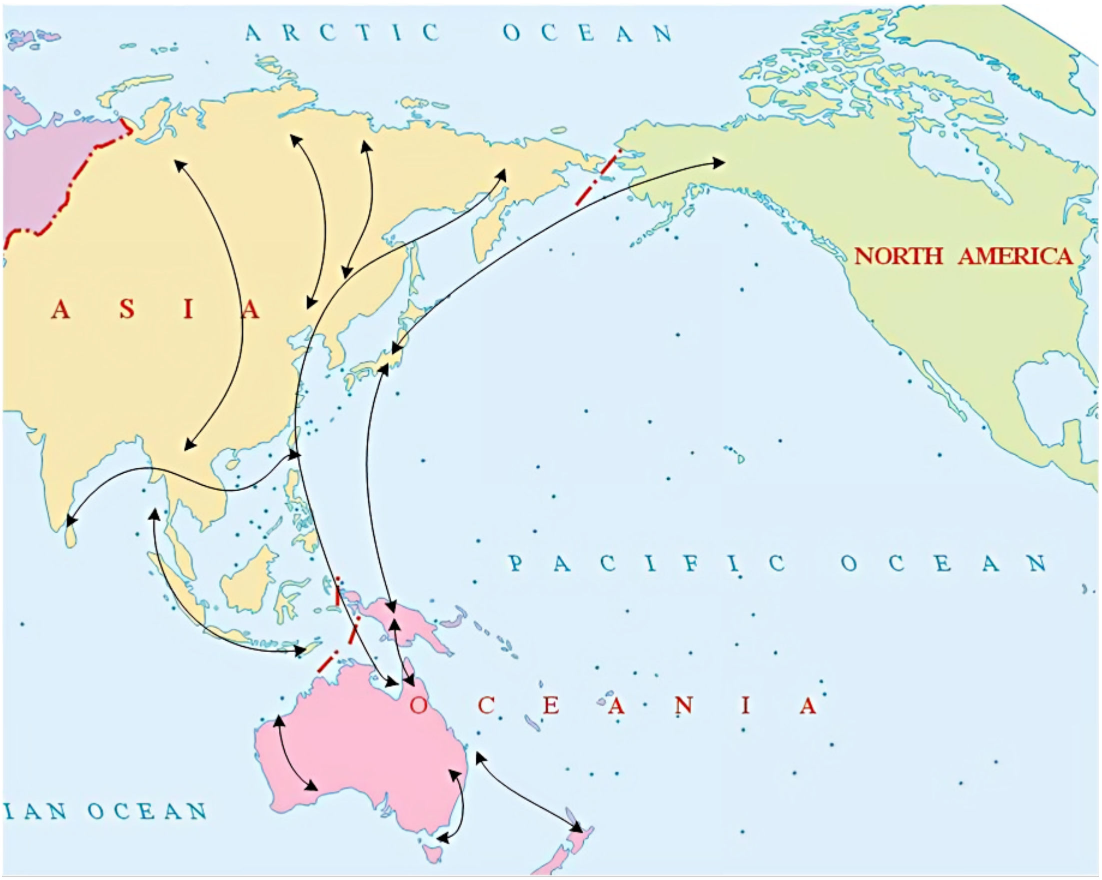

FIGURE 1. The route of Bird migration

4.2. **Adaptation Evaluation.** Suppose fitness function $\boldsymbol{f(\cdot)}$, fitness value with individuals can be calculated by putting the solution vector into the adaptation function, result is conserved in the array of Equation (3). The adaptation value for location of individuals in group describes the environmental quality of some areas of the habitat. By sorting the fitness values in the group, a numerical vector used to describe the quality of life of the group can be obtained. With the continuous exploration and development of the population, the group's quality of life will approach the upper limit of the habitat's environmental carrying capacity:

$$F_m = \begin{bmatrix} f_1\left(\left[x_{1,1}^m, x_{1,2}^m, ..., x_{1,d}^m\right]\right) \\ f_2\left(\left[x_{2,1}^m, x_{2,1}^m, \ldots, x_{2,d}^m\right]\right) \\ \cdots \\ f_n\left(\left[x_{n,1}^m, x_{n,1}^m, \ldots, x_{n,d}^m\right]\right) \end{bmatrix} \tag{3}$$

Where $f_n\left(\left[x_{n,1}^m, x_{n,1}^m, \ldots, x_{n,d}^m\right]\right)$ is fitness value for n unit in the m group's position, and F_m is a numerical vector describing the group's quality of life. It's worth noting that after each fitness value update, the numerical vector must be reordered to ensure that $f_n < \cdots < f_2 < f_1$ The reason for this is to ensure that each group's leader always has an absolute dominance position.

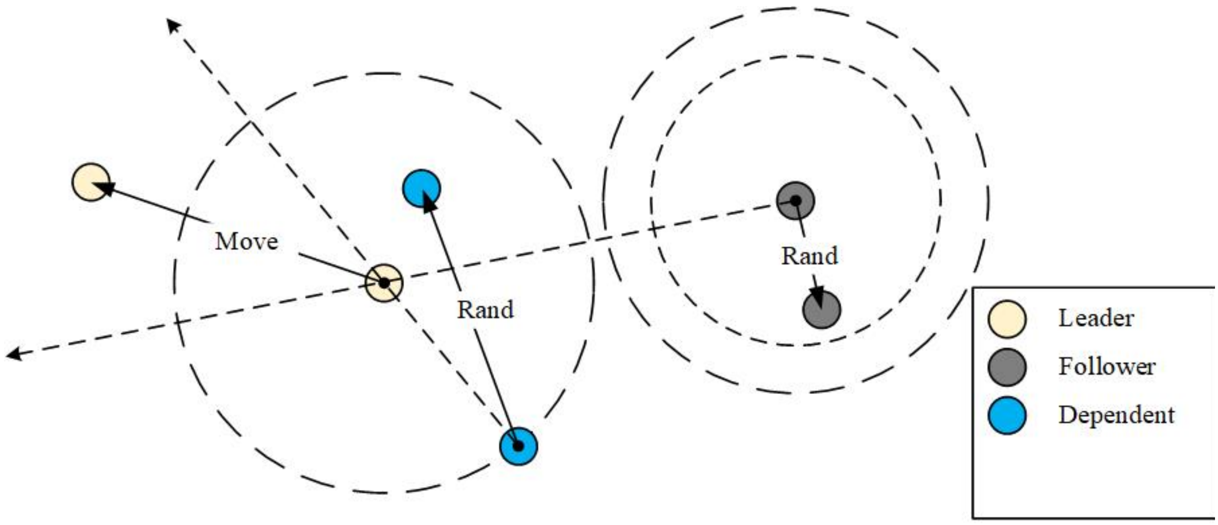

FIGURE 2. Position updating in MSA

When the numerical vector of the quality of life is obtained, the high-quality ecological locations in the entire ecological environment can be screened out according to the vector:

$$X_{Best\_Loc} = \begin{pmatrix} x_{1,1}^1 & \cdots & x_{1,d}^1 \\ \vdots & \ddots & \vdots \\ x_{1,1}^m & \cdots & x_{1,d}^m \end{pmatrix} \tag{4}$$

Where $\left[x_{1,1}^m, x_{1,2}^m, \cdots, x_{1,d}^m\right]$ represents the highest quality ecological position in the m group, that is, the position of the leader in the m group.

4.3. **Updating Positions.** As mentioned above, we divide the migration behavior of animals into two types, namely station keeping and migration movement. In each case, individuals are more inclined to find a more favorable ecological environment and follow the strong leader in the population. Individuals in the population can efficiently explore their surroundings by updating their positions through autonomous movement, thereby obtaining more survival resources. As a result, we can model this behavior mathematically.

Case 1: The animal population's behavior tends to explore randomly in the habitat and follow the population leader at times during the station keeping stage. It is worth saying that the leader in the population will patrol their territory and lead the population to obtain food, while the vulnerable groups in the population often choose to rely on the leader in exchange for survival opportunities. Therefore, the individual location information in this case will be updated as follow:

$$X_k^{m,t+1} = \begin{cases} x_k^{m,t} + U(0,1) \times \left(\left(x_k^{m,t} - x_2^{m,t}\right) + \left(x_k^{m,t} - x_2^{m,t}\right)\right), & k = 1 \\ x_k^{m,t} + x_k^{m,t} \times N(0,1), & k = 2 \\ x_1^{m,t} + \left(x_k^{m,t} - x_1^{m,t}\right), & k = 3 \\ x_k^m = LB + U(0,1) \times (UB - LB), & otherwise \end{cases} \tag{5}$$

The $x_k^{m,t}$ denotes the location of the k individual in m group at t iteration. $N(0,1)$ represents the standard deviation of one and an average value of stochastic figure of zero, which obeys a normal distribution, and $N(0,1)$ represents a stochastic figure evenly dispersal between zero and one. LB and UB can be utilized to indicate the upper and lower surpass of the dimension in the solution space, respectively. In the formula above, to scavenge for better food sources, the judgment and exploration of the leader are made based on the existing living conditions, while the followers will adopt a relatively conservative strategy, searching the surroundings under the existing living conditions. As for the dependents, they are more inclined to explore around the leader to get food.

Figure 2 shows how the group searches for and updates the position in the seek space of two-dimension. Group's leader will judge their own direction based on the position of followers and dependents. The dependents in the group will randomly update their positions around the leader-centered area. The group's followers will randomly explore around themselves according to probability.

Case 2: When the ecological environment in the habitat is no longer suitable, when means that the optimal location in the group's habitat has a continuous for "Liminte" generation. Under this, the population will spontaneously explore other survival areas under the pressure of survival and moving towards a high-quality ecological niche, and stop this movement after acquiring a more suitable ecological niche. The direction of migration for the population can be calculated as Equation (6):

$$V_1{}^{m,t} = \begin{cases} \alpha \times U(0,1) \times \left(T - x_k^{m,t}\right) + \beta \times U(0,1) \times \left(M - x_1^{m,t}\right) \\ \alpha \times U(0,1) \times \left(T - x_1^{m,t}\right) \end{cases} \tag{6}$$

T represents nowadays population's location for head group. The head group is composed of the top 5 percent of the current population fitness value, and a random selection method is adopted when selecting T. M represents the population's historical optimal position, which is spread to other populations by the first discoverer according to the message transfer model, and the population that obtains the information can formulate its migration strategy based on the information.

Case 3: Furthermore, on account of the strong mobility and autonomy of animals during migration, some will actively explore unknown areas, which may bring in a certain degree of unpredictability. In such a case, we randomly explore the position in the solution space through Equation (7), and judge whether to stay here according to Equation (8):

$$XN_n^{m,t} = (Best\_Loca - Worst\_Loca) \times N(0,1) + Worst\_Loca \tag{7}$$

$$X_n^{m,t} = \begin{cases} XN_n^{m,t}, & f\left(XN_n^{m,t}\right) > f\left(X_n^{m,t}\right) \\ X_n^{m,t}, & f\left(XN_n^{m,t}\right) < f\left(X_n^{m,t}\right) \end{cases} \tag{8}$$

$Best - Location$ and $Worst - Location$ in Equation (7) represent the locations with the best and worst fitness values, respectively, and $N(0,1)$ the standard deviation of one and an average value of stochastic figure of zero that follows a normal distribution.

4.4. **Information Dissemination.** In addition to the motion simulation mentioned above, we also need to consider the impact of information transmission between populations on migration activities. We assume that in a closed environment with a certain total population, the optimal location information is transmitted from the known individuals to the unknown individuals.

Due to the more individuals know this information, the more individuals get the optimal position every day; the more the number of individuals who don't know the optimal position, the more individuals will get the optimal position every day. Consequently, we can make the following assumptions: the number of individuals who newly obtain the optimal position information every day is proportional to the product of the number of individuals who have obtained the optimal position information and the number of individuals who have not obtained the optimal position information.

$$p_{t+1} - p_t = cp_t \times (MN - p_t) \tag{9}$$

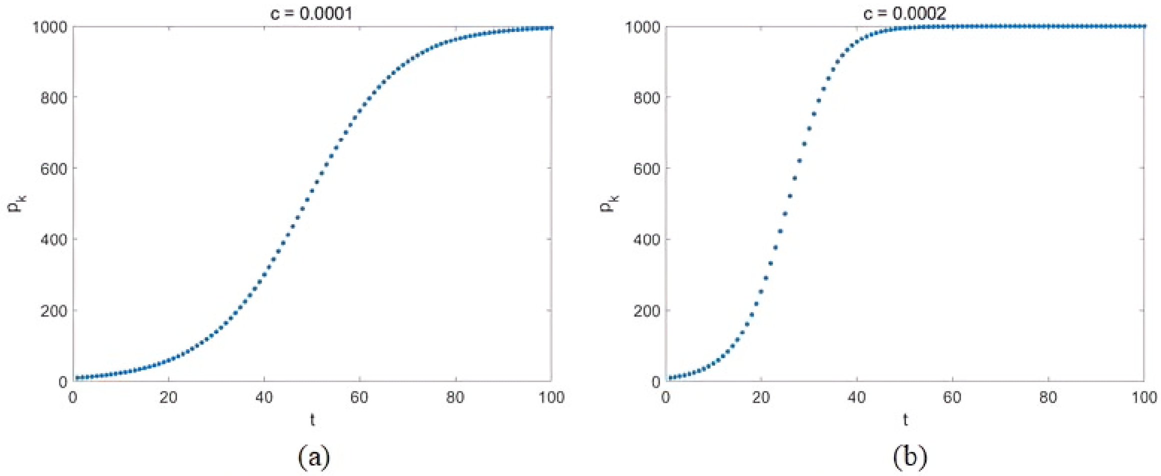

FIGURE 3. The infection of parameter c on message dissemination

In t-th iteration, p_t is the number of individuals who have known the optimal location information, and c is the propagation coefficient. The meaning of Equation (9) is the relative increase in the number of individuals who have new information every day for individual who has not obtained information. We can get the recursive form of the first-order difference equation by sorting out the above formula:

$$p_{t+1} = (1 + cMN) \times p_t \times \left(1 - \frac{c}{1 + cMN} p_t\right) \tag{10}$$

From the Equation (10), we suppose the initial state is $p_0 = 1$, $MN = 1000$, and c takes two different values of 0.0001 and 0.0002, and that the result of calculating p_t using the recursive formula is shown in Figure 3.

People's number who have learned information after 80 iterations is close to the total number MN when $c = 0.0001$, but only 40 iterations are required when $c = 0.0002$, and the propagation speed is significantly accelerated.

It can be seen from Equation (10) that an explicit expression of P_k cannot be obtained, so in Equation (11) and Equation (12) we set:

$$b = 1 + cMN \tag{11}$$

$$x_k = \frac{c}{1 + cMN} (1 - x_k) \tag{12}$$

Thus, Equation (12) is simplified to Equation (13):

$$x_{k+1} = bx_k (1 - x_k) \tag{13}$$

The non-zero equilibrium point of Equation (10) can be obtained in Equation (14):

$$x^* = 1 - 1/b \tag{14}$$

According to the stability condition of $|f'(x^*)| < 1$ we know that $1 < b < 3$, that is:

$$c < 2/MN \tag{15}$$

From Equation (10), it can be concluded that $c < 1/MN$ satisfies the equation balance condition of Equation (15).

4.5. **Stopping Criterion.** Algorithm's termination condition determines iterations. Tolerance is a common convergence criterion, which defines an allowable but small threshold between the last several consecutive results. Another common convergence criterion is the maximum execution time as the termination condition. In this study, we use the maximum iterations as stopping standard. Algorithm1 is the pseudocode of MSA.

Algorithm 1: Migration Search Algorithm

```
Define input
Generate stochastic positions by Equation (1)
Assess fitness of agent's position
Rank agents in same group on their fitness value
Proceed according to the stopping standard
  For each group do
    Update the position of agents in this group by Equation (5)
    Rank agents again
  End for
  Update the best position
  Disseminate information of the best position
  For each group do
    If the best location do not update for continuous Liminte generations then
      the group begin to migrate by Equation (6)
    End if
  End for
  For each migratory group do
    If the group found a location better than before during migration then
      the group stop migration
    End if
  End for
  Explore unknowed area randomly
End while the best location of colony found is the final optimum solution
```

5. **Experiment and Analysis.** In this section, the MSA algorithm will be tested by comparing it with 26 test functions and analyzing its performance against these benchmarks, including continuous function and discrete function, linear function and nonlinear function, single mode function and multi-modal function, separable function, and inseparable function. When a function has a mode, it refers to the number of fuzzy peaks on its surface. A function with only one fuzzy peak is called a single-modal function, and a function with multiple fuzzy peaks is called a multi-modal function. Multi-modal function with multiple fuzzy peaks contains multiple local optimums. This process of algorithm optimization can easily lead to falling into local optimum situations and getting far from the optimal solution area. Moreover, we can simply assume that inseparable functions are harder to solve than separable functions for the separability of functions because inseparable functions have complex functional relationships between the various dimensional variables. Therefore, we are more concerned about the performance of the new algorithm in the above two types of test functions. In this section, the twenty-six test functions are classified into Tables 3, 5, 7 and 9, and then these four groups of test functions allow an analysis of the performance of the MSA algorithm. The test results are recorded in Tables 4, 6, 8 and 10.

TABLE 1. Inspired optimization algorithms for Testing

Algorithm	Inspiration
Gravitational search algorithm (GSA)	Law of gravity and mass interactions
Particle swarm optimization (PSO)	Intelligent social behavior of bird flock
Artificial bee colony (ABC)	Honey Bee
Migratory search algorithm (MSA)	Animal migration behavior

TABLE 2. Inspired optimization algorithms for Testing

Name of parameter	GSA	PSO	ABC	MSA
Alpha	2			
Beta	2			
GO	0.5			
c1		2		
c2		2		
w		0.5		
Liminte			50	
Liminte				20

A numerical optimization problem proposed by the IEEE CEC 2014 special meeting [39] is another test platform we use in this research. These benchmark functions are variants of complex mathematical optimization problems involving translation, rotation, expansion, and combination. The details of these functions are information reference Table 11. Considering that the difficulty of the problem will increase with the growth of the problem size as the search space increases. Therefore, when testing on the CEC2014 function test platform, we chose two problem scales: dim10 and dim30. The test results are recorded in Tables 12 and 13.

This study compares the MSA algorithm with the GSA algorithm, PSO algorithm, and ABC algorithm to make sure to the validity of the experiment, and the details of these algorithms are listed in Table 1. This experiment had a population size of 120 and a maximum iteration count of 1000, respectively. The number of individuals in each group is 3, including one leader, one follower, and one dependent. There is a list of parameters for all algorithms under Table 2. Statistical results are presented in the table for each algorithm after 50 iterations in each experiment.

5.1. **Experiment 1.** The MSA algorithm's performance in the single-modal separable test function is evaluated in this experiment. Table 3 contains the detailed information about these functions. Table 4 records the statistical data of 50 independent runs of the MSA algorithm and other comparison algorithms, including the optimal, worst, average, and standard deviation values. The results show that only the MSA algorithm was successful in determining the optimal values of TF1 and TF2. TF3 and TF4 cannot be optimized by any algorithm. The result of the MSA algorithm is similar to other algorithms.

In addition, we also analyzed the convergence of the algorithm. We can see that the MSA algorithm got faster convergence in TF1 and TF2, while the performance is slightly

TABLE 3. Inspired optimization algorithms for Testing

Function	Name	Expression	Dim	Range	Fmin
TF1	Step	$\text{TF}\,1(x) = \sum_{j=1}^{d} (x_j + 0.5)^2$	30	$[-5.12, 5.12]$	0
TF2	Sphere	$\text{TF}\,2(x) = \sum_{j=1}^{d} x_j^2$	30	$[-100, 100]$	0
TF3	Sum Squares	$\text{TF}\,3(x) = \sum_{j=1}^{d} jx_j^2$	30	$[-10, 10]$	0
TF4	Quartic	$\text{TF}\,4(x) = \sum_{j=1}^{d} jx_j^4 + rand$	30	$[-1.25, 1.25]$	0

TABLE 4. Statistical results obtained by GSA, PSO, ABC and MSA through 30 independent runs on classical unimodal and separable benchmark functions

Function	MSA		GSA		PSO		ABC	
	Average	Std	Average	Std	Average	Std	Average	Std
TF1	**0.00E+00**	**0.00E+00**	0.00E+00	0.00E+00	3.95E+00	2.54E+00	1.00E-01	3.08E-01
TF2	**2.24E-50**	**6.76E-50**	3.06E+04	3.34E+03	1.15E+02	4.53E+01	4.89E+02	1.45E+02
TF3	6.21E+03	4.77E+03	**6.72E+01**	**2.04E+01**	6.26E+03	4.84E+03	1.72E+02	2.88E+02
TF4	9.28E+00	**3.89E-01**	1.03E+01	6.29E-01	**8.36E+00**	4.05E-01	1.18E+01	1.09E+00

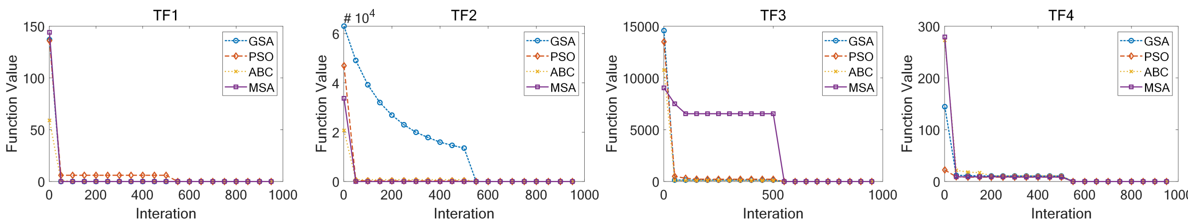

FIGURE 4. Convergence rate comparison for benchmark functions

worse in TF3 and TF4. The reason for this is that when the MSA algorithm is at the bottleneck of local search, the algorithm will selectively migrate to other regions. Although this strategy reduces the MSA algorithm's single-modal function convergence rate, the degree of reduction is still acceptable, and it will improve global exploration performance. In addition, this experiment also analyzes the convergence of the algorithm and draws the convergence curve, as shown in Figure 4.

5.2. **Experiment 2.** The purpose of this experiment is to evolute the performance of MSA in single-model independent testing (see Table 5). The complexity of this experiment is higher than the previous one because the function has inseparable characteristics. Table 6 records the statistics of the MSA algorithm and other comparison algorithms in eight benchmark functions. These results indicate that the MSA algorithm successfully found the global minimum or close to the global minimum of functions other than TF8 and TF12. There is no algorithm for TF8 to get its optimal value, but the outcome of the

TABLE 5. The description of classical unimodal and separable benchmark functions

Function	Name	Expression	Dim	Range	Fmin
TF5	Beale	$\text{TF }5(x) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2$	2	$[-4.5, 4.5]$	0
TF6	Easom	$\text{TF }6(x) = -\cos(x_1)\cos(x_2) \exp\left(-(x_1-\pi)^2 - (x_2-\pi)^2\right)$	2	$[-100, 100]$	0
TF7	Matyas	$\text{TF7}(x) = 0.26\,(x_1^2 + x_2^2) - 0.48x_1x_2$	2	$[-10, 10]$	0
TF8	Colville	$\text{TF }8(x) = 100\,(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90\,(x_3^2 - x_4)^2 +10.1\,(x_2 - 1)^2 + (x_4 - 1)^2 +19.8\,(x_2 - 1)\,(x_4 - 1)$	4	$[-10, 10]$	0
TF9	Zakharov	$\text{TF }9(x) = \sum_{j=1}^{d} x_j^2 + \left(\sum_{j=1}^{d} 0.5jx_j\right)^2 + \left(\sum_{j=1}^{d} 0.5jx_j\right)^4$	10	$[-5, 10]$	0
TF10	Schwefel 2.22	$\text{TF }10(x) = \sum_{j=1}^{d} \lvert x_j\rvert + \prod_{j=1}^{d} \lvert x_j\rvert$	30	$[-10, 10]$	0
TF11	Schwefel 1.2	$\text{TF }11(x) = \sum_{j=1}^{d}\left(\sum_{k=1}^{j} x_k\right)^2$	30	$[-100, 100]$	0
TF12	Dixon-Price	$\text{TF }12(x) = (x_1 - 1)^2 + \sum_{j=2}^{d} j\left(2x_j^2 - x_j - 1\right)^2$	30	$[-10, 10]$	0

MSA algorithm is optimal. For the TF12 algorithm, only the GSA algorithm found the global minimum.

Among the eight benchmark functions, the MSA algorithm only performed weaker than the ABC and GSA algorithms in TF6 and TF12, respectively. Clearly, MSA outperformed all the other algorithms in this collection of control experiments. The MSA algorithm's convergence is comparable to that of other algorithms.

Based on the above two experiments, we can draw consistent conclusions that for unimodal functions, the MSA algorithm performs admirably; the MSA algorithm performs better when the complexity of the function increases. In the test of convergence speed, the results are shown in Figure 5, and the convergence speed of MSA algorithm is at the same level as other algorithms.

5.3. **Experiment 3.** This experiment analyzes the performance of the MSA algorithm in the multi-modal separable function. The detailed information of these functions is depicted in Table 7. In contrast to the previous two experiments, the multi-modal function contains multiple minimum points, the algorithm is at risk of falling into minimum area and failing to reach the global optimal value. Therefore, the algorithm's exploration capabilities for multi-modal functions must be enhanced. Taking into account the experimental statistical data shown in Table 8, the MSA algorithm has discovered the global optimal value of all functions and outperforms other algorithms in terms of accuracy.

Table 6. Statistical results obtained by GSA, PSO, ABC and MSA through 30 independent runs on classical unimodal and non-separable benchmark functions

Function	MSA		GSA		PSO		ABC	
	Average	Std	Average	Std	Average	Std	Average	Std
TF5	1.36E-04	4.54E-05	9.42E-05	6.52E-08	**9.41E-05**	**7.20E-20**	**9.41E-05**	7.77E-20
TF6	1.59E-03	2.34E-03	8.50E-02	2.60E-01	2.07E-02	9.24E-02	**0.00E+00**	**0.00E+00**
TF7	**9.37E-156**	**3.48E-155**	2.21E-10	1.71E-10	3.95E-71	1.71E-70	3.27E-101	8.51E-101
TF8	**-1.29E+02**	**1.14E+01**	-5.06E+01	3.72E+01	-1.01E+02	5.66E+01	-1.27E+02	2.85E+01
TF9	**3.09E-94**	**1.01E-93**	2.10E-06	7.30E-07	8.44E-03	1.32E-02	2.29E+01	2.90E+01
TF10	**4.28E-37**	**4.67E-37**	2.86E-01	3.79E-01	7.84E+00	2.06E+00	8.80E+00	1.93E+00
TF12	**7.39E-06**	**3.29E-05**	4.84E+04	1.28E+04	2.34E+03	9.29E+02	9.03E+02	3.40E+02
TF11	2.68E+02	2.15E+01	**4.14E+01**	**2.02E+01**	5.49E+01	3.23E+01	1.72E+02	4.12E+01

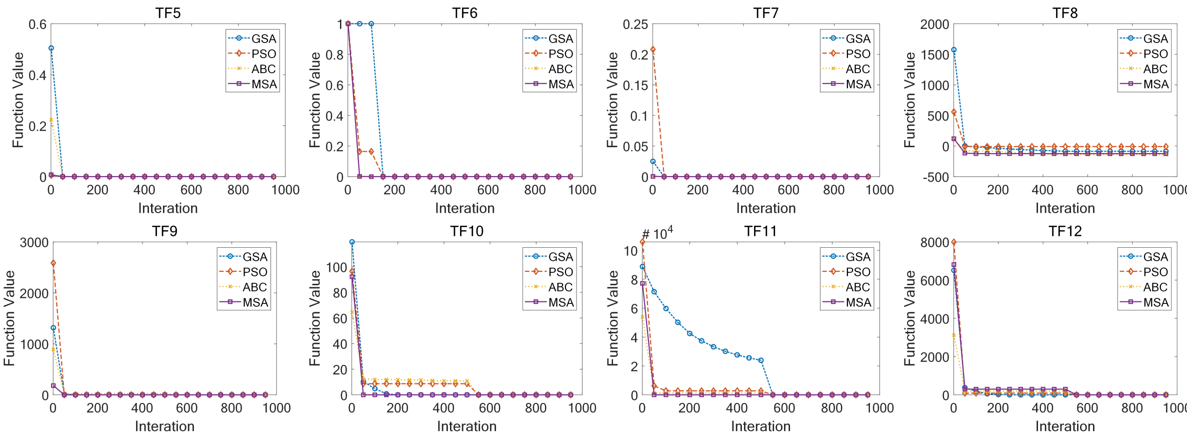

Figure 5. Convergence rate comparison for benchmark functions

Based on the results of the convergence speed test, MSA is the fastest algorithm by far. Especially in TF18, other algorithms' convergence curves have been flattened many times, which is the performance of falling into the local minimum, whereas the MSA algorithm's convergence curve is smooth, indicating that the MSA algorithm has strong exploration capabilities. In the convergence speed test, according to the results in Figure 6, we can see that the convergence speed of MSA algorithm is the fastest among all algorithms. In TF18, the convergence curves of other algorithms appear flat for many times, which is the manifestation of falling into local minimum, while the convergence curves of MSA algorithm are smooth, which indicates that MSA algorithm has strong exploration ability.

5.4. **Experiment 4.** This group of test functions is the most difficult in comparison to the previous experiment which are all multimodal inseparable functions, and the details of these eight test functions are recorded in Table 9. Table 10 shows that, MSA is better than other algorithms in this group of experiments based on seven test functions, and only weaker than the ABC algorithm in TF20, but the optimal value is also found.

In addition, this experiment also tests the convergence speed of the MSA algorithm with these eight test functions. As can be seen from Figure 7, thanks to its effective strategy for jumping out of the local optimum, the MSA algorithm's convergence speed and final accuracy are superior to other algorithms.

TABLE 7. The description of classical unimodal and separable benchmark functions

Function	Name	Expression	Dim	Range	Fmin
TF13	Bohachevsky1	$\text{TF }13(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$	2	$[-100, 100]$	0
TF14	Booth	$\text{TF }14(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	2	$[-10, 10]$	0
TF15	Michalewciz2	$\text{TF }15(x) = -\sum_{j=1}^{d} \sin(x_j)\left(\sin\left(jx_j^2/\pi\right)\right)^{20}$	2	$[-5.12, 5.12]$	0
TF16	Michalewciz5	$\text{TF }16(x) = -\sum_{j=1}^{d} \sin(x_j)\left(\sin\left(jx_j^2/\pi\right)\right)^{20}$	30	$[-9, 9]$	0
TF17	Michalewaicz10	$TF17(x) = -\sum_{j=1}^{d} \sin(x_j)\left(\sin\left(jx_j^2/\pi\right)\right)^{20}$	30	$[-1, 1]$	0
TF18	Rastrigin	$\text{TF }18(x) = \sum_{j=1}^{d} \left(x_j^2 - 10\cos(2\pi x_j) + 10\right)$	30	$[-5.12, 5.12]$	0

TABLE 8. Statistical results obtained by GSA, PSO, ABC and MSA through 30 independent runs on classical multimodal and separable benchmark functions

Function	MSA		GSA		PSO		ABC	
	Average	Std	Average	Std	Average	Std	Average	Std
TF13	**0.00E+00**	**0.00E+00**	2.50E-02	4.67E-02	0.00E+00	0.00E+00	0.00E+00	0.00E+00
TF14	**0.00E+00**	**0.00E+00**	9.49E-09	8.48E-09	0.00E+00	0.00E+00	0.00E+00	0.00E+00
TF15	**0.00E+00**	**0.00E+00**	1.45E-02	2.45E-02	0.00E+00	0.00E+00	0.00E+00	0.00E+00
TF16	**1.50E-32**	**5.62E-48**	4.05E-02	5.44E-02	1.62E+00	6.24E-01	1.47E+00	4.96E-01
TF17	**2.90E-146**	**1.30E-145**	2.44E-08	6.44E-08	9.54E-06	1.62E-05	4.03E-08	8.72E-08
TF18	**0.00E+00**	**0.00E+00**	3.49E-01	5.84E-01	5.91E+01	1.43E+01	1.02E+02	1.32E+01

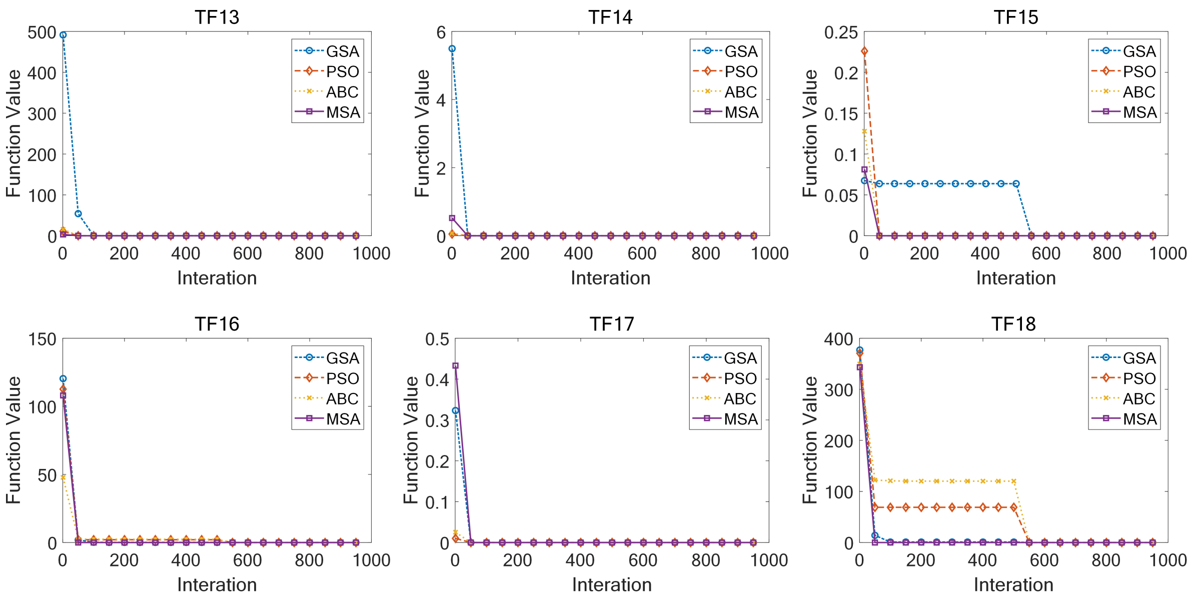

FIGURE 6. Convergence rate comparison for benchmark functions

TABLE 9. The description of classical unimodal and non-separable benchmark functions

Function	Name	Expression	Dim	Range	Fmin
TF19	Schaffer	$\text{TF }19(x) = 0.5 + \frac{\sin^2\left(\sqrt{x_1^2+x_2^2}\right)-0.5}{\left(1+0.001\left(x_1^2+x_2^2\right)\right)^2}$	2	$[-100, 100]$	0
TF20	Six Hump Cannel Back	$\text{TF }20(x) = 4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	2	$[-5, 5]$	0
TF21	Boachevsky2	$\text{TF }21(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1)\cos(4\pi x_2) + 0.3$	2	$[-100, 100]$	0
TF22	Boachevsky3	$\text{TF }22(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1 + 4\pi x_2) + 0.3$	2	$[-100, 100]$	0
TF23	Shubert	$\text{TF }23(x) = \left(\sum_{j=1}^{5} j\cos(j+1)x_1 + j\right)\left(\sum_{j=1}^{5} j\cos((j+1)x_2 + j)\right)$	2	$[-2, 2]$	0
TF24	Rosenbrock	$\text{TF }24(x) = \sum_{j=1}^{d-1} 100\left(x_{j+1} - x_j^2\right)^2 + (x_j - 1)^2$	30	$[-30, 30]$	0
TF25	Griewank	$\text{TF }25(x) = \frac{1}{4000}\left(\sum_{j=1}^{d}(x_j - 100)^2\right) - \left(\prod_{j=1}^{d}\cos\left(\frac{x_j-100}{\sqrt{j}}\right)\right) + 1$	30	$[10, 10]$	0
TF26	Ackley	$\text{TF }26(x) = -20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{j=1}^{d}x_j^2}\right) - \exp\left(\frac{1}{d}\sum_{j=1}^{d}\cos(2\pi x_j)\right) + 20 + e$	30	$[-32, 32]$	0

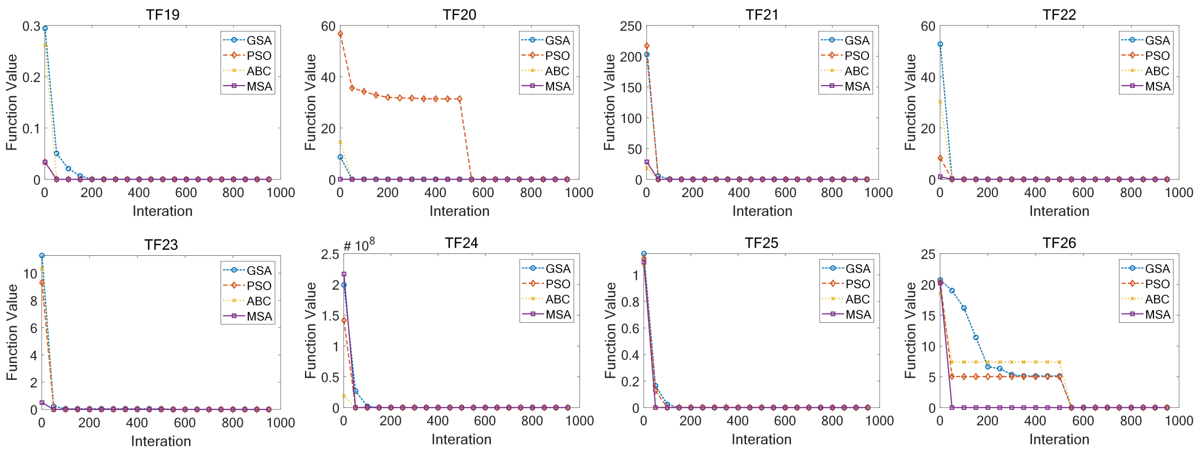

FIGURE 7. Convergence rate comparison for benchmark functions

From the above, it is possible to draw the following conclusions: the MSA algorithm outperforms other algorithms on the single-modal function and is quite competitive, with performance decreasing less as the problem complexity increases. Furthermore, the MSA algorithm performs well in multi-modal functions due to its effective migration strategy, and can effectively jump away from the local optimal solution and find the global optimal solution faster.

5.5. **Complex Function Performance.** We used the benchmark functions recommended by IEEE CEC2014 in this experiment to determine MSA's robustness and effectiveness.

Table 10. Statistical results obtained by GSA, PSO, ABC and MSA through 30 independent runs on classical multimodal and non-separable benchmark functions

Function	MSA		GSA		PSO		ABC	
	Average	Std	Average	Std	Average	Std	Average	Std
TF19	**0.00E+00**	**0.00E+00**	9.54E-04	1.86E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00
TF20	7.35E-06	1.44E-05	3.06E-02	3.97E-02	3.33E+02	5.47E+02	**9.59E-84**	**4.24E-83**
TF21	**0.00E+00**	**0.00E+00**	1.62E-02	2.75E-02	0.00E+00	0.00E+00	0.00E+00	0.00E+00
TF22	**0.00E+00**	**0.00E+00**	3.61E-03	4.22E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00
TF23	**1.32E-04**	**3.13E-04**	4.93E-02	2.01E-02	9.48E-03	6.09E-03	3.49E-03	3.08E-03
TF24	**0.00E+00**	**0.00E+00**	1.43E+05	1.01E+05	1.86E+03	2.00E+03	2.36E+04	1.31E+04
TF25	**1.11E-16**	**1.19E-16**	1.07E-07	3.23E-08	2.44E-03	2.35E-03	4.96E-03	5.07E-03
TF26	**1.42E-15**	**1.30E-15**	1.03E+01	8.03E-01	4.06E+00	5.68E-01	6.18E+00	4.80E-01

Table 11. The brief description of CEC benchmark functions

Num	Name	Range	Fmin
TF1	Rotated High Conditioned Elliptic Function (CEC1)	$[-100, 100]$	0
TF2	Rotated Bent Cigar Function (CEC2)	$[-100, 100]$	0
TF4	Shifted and Rotated Rosenbrock's Function (CEC4)	$[-100, 100]$	0
TF7	Shifted and Rotated Griewank's Function (CEC7)	$[-100, 100]$	0
TF13	Shifted and Rotated HappyCat Function (CEC13)	$[-100, 100]$	0
TF14	Shifted and Rotated HGBat Function (CEC14)	$[-100, 100]$	0
TF15	Shifted and Rotated Expandad Griewank's plus Rosenbrock's Function (CEC15)	$[-100, 100]$	0
TF19	Hybrid Function 3 (CEC19)	$[-100, 100]$	0
TF23	Composition Function 1 (CEC23)	$[-100, 100]$	0
TF24	Composition Function 2 (CEC24)	$[-100, 100]$	0
TF25	Composition Function 3 (CEC25)	$[-100, 100]$	0
TF26	Composition Function 4 (CEC26)	$[-100, 100]$	0

CEC2014 is a special conference and competition focused on optimizing single-objective real parameter values. Through rotation, translation, compounding, and other operations, these test functions are composed of some basic test functions. Finding the best is difficult. Please consult the literature [39] for additional information on CEC2014. In this experiment, which considers 12 CEC2014 test functions, the results of each algorithm are recorded after 30 independent runs.

Table 12 shows that the MSA algorithm only maintains its lead in TF4, 7, 13, 14, 23, but Table 13 shows that the MSA algorithm is superior in each test function. This shows the MSA algorithm's superiority in dealing with high-dimensional complex problems.

5.6. **Parameter Analysis.** To analyze how the parameters affect the performance of the algorithm, this experiment adjusted the parameters and recorded the experimental results. According to the results in Table 14, as the parameter Liminte increases, the algorithm's result becomes more accurate. This is since parameter Liminte determines

Table 12. Statistical results obtained by GSA, PSO, ABC and MSA through 30 independent runs on CEC 2014 benchmark functions with 10 Dim

Function	MSA		GSA		PSO		ABC	
	Average	Std	Average	Std	Average	Std	Average	Std
TF1	8.54E+06	**3.08E+06**	9.01E+07	6.26E+07	1.20E+07	1.06E+07	**5.93E+06**	5.46E+06
TF2	6.61E+08	1.96E+08	4.34E+09	1.25E+09	**7.82E+07**	**1.02E+08**	6.83E+08	5.03E+08
TF4	**5.64E+01**	**1.32E+01**	1.02E+03	4.71E+02	1.22E+02	4.23E+01	1.17E+02	6.56E+01
TF7	**1.23E+01**	**4.34E+00**	1.24E+02	3.08E+01	2.67E+01	1.44E+01	3.14E+01	1.61E+01
TF13	6.43E-01	**1.17E-01**	3.62E+00	4.30E-01	**5.32E-01**	4.70E-01	1.09E+00	8.27E-01
TF14	**1.41E+00**	**5.06E-01**	2.81E+01	6.27E+00	3.74E+00	3.73E+00	7.64E+00	4.65E+00
TF15	1.03E+01	**1.57E+00**	1.30E+03	1.16E+03	**6.04E+00**	6.34E+00	6.98E+00	6.53E+00
TF19	5.76E+00	9.25E-01	3.23E+01	1.28E+01	4.62E+00	2.17E+00	**2.59E+00**	**7.86E-01**
TF23	**2.00E+02**	**0.00E+00**	4.32E+02	3.89E+01	3.34E+02	3.29E+00	3.39E+02	5.25E+00
TF24	1.55E+02	**7.32E+00**	2.05E+02	9.87E+00	**1.32E+02**	9.22E+00	1.38E+02	7.66E+00
TF25	1.93E+02	1.03E+01	2.03E+02	**7.41E+00**	**1.86E+02**	2.16E+01	1.86E+02	1.84E+01
TF26	1.01E+02	1.64E-01	**1.03E+02**	**1.25E+00**	1.01E+02	5.00E-01	1.00E+02	1.09E-01

Table 13. Statistical results obtained by GSA, PSO, ABC and MSA through 30 independent runs on CEC 2014 benchmark functions with 30 Dim

Function	MSA		GSA		PSO		ABC	
	Average	Std	Average	Std	Average	Std	Average	Std
TF1	**4.09E+08**	**8.39E+07**	1.87E+09	5.29E+08	5.03E+08	1.86E+08	4.86E+08	1.36E+08
TF2	**2.45E+10**	**3.62E+09**	9.18E+10	1.01E+10	3.62E+10	5.71E+09	4.23E+10	5.87E+09
TF4	**1.65E+03**	**3.41E+02**	1.87E+04	3.83E+03	4.14E+03	1.18E+03	5.62E+03	1.74E+03
TF7	**2.18E+02**	3.74E+01	8.05E+02	9.65E+01	3.50E+02	**4.96E+01**	4.15E+02	9.54E+01
TF13	**3.72E+00**	**3.69E-01**	8.56E+00	7.02E-01	5.64E+00	4.83E-01	6.05E+00	6.72E-01
TF14	**6.99E+01**	**1.09E+01**	2.97E+02	3.64E+01	1.36E+02	1.67E+01	1.65E+02	3.06E+01
TF15	**2.40E+04**	**1.14E+04**	2.45E+06	1.19E+06	1.22E+04	1.25E+04	2.90E+04	2.03E+04
TF19	**1.27E+02**	**1.32E+01**	6.59E+02	1.80E+02	1.89E+02	6.87E+01	1.65E+02	6.01E+01
TF23	**2.00E+02**	1.87E-13	1.26E+03	2.31E+02	4.73E+02	**7.13E+01**	5.69E+02	1.13E+02
TF24	**2.00E+02**	**8.91E-03**	4.21E+02	1.98E+01	2.36E+02	4.92E+00	2.42E+02	7.15E+00
TF25	**2.00E+02**	**0.00E+00**	3.26E+02	3.18E+01	2.22E+02	5.15E+00	2.11E+02	2.06E+00
TF26	**1.04E+02**	**4.05E-01**	2.30E+02	7.50E+01	1.68E+02	4.36E+01	1.05E+02	9.28E-01

the duration of the population's local development a larger Liminte improves the performance of algorithm in the local development, but it will inevitably weaken the algorithm's exploration ability, increasing the probability of falling into the local optimum.

6. **Application.** The main problem studied in this part is the hyper-parameters optimization design of text classification algorithm. In fact, the text classification algorithm has had great success and made a breakthrough in many problems, but the algorithm's performance is over-reliant on the selection of hyper-parameters [40], and the fine-tuning

TABLE 14. The effect of variation of Liminte on the performance of MSA

Function	Liminte=5	Liminte=10	Liminte=15	Liminte=20	Liminte=25	Liminte=30	Liminte=35
TF2	2.09E-06	1.47E-48	1.18E-51	1.31E-51	1.32E-49	1.12E-50	2.66E-51
TF4	5.16E+01	2.45E+01	1.29E+01	9.97E+00	1.00E+01	9.51E+00	9.55E+00
TF7	1.88E-35	2.87E-89	2.95E-152	2.21E-156	7.16E-159	8.04E-157	7.52E-156
TF11	8.56E+03	2.33E+03	2.74E+02	3.80E+01	3.23E+00	5.02E-02	5.21E-03
TF17	1.46E-04	3.97E-08	2.63E-135	6.23E-134	1.64E-144	1.49E-151	3.71E-141
TF24	7.29E+03	3.09E-31	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
TF25	1.96E-01	2.73E-03	2.53E-07	1.83E-14	5.61E-16	4.05E-16	2.61E-16

hyper-parameters is time-consuming and laborious, especially requiring the experience of the algorithm designer.

In this section, the MSA algorithm is used to optimize the hyper-parameter design of the text classifier to reduce the performance loss caused by model migration. Other optimization algorithms are compared to the results of this algorithm. A text classification algorithm based on support vector machines (SVM) is used to verify the performance of the MSA algorithm in hyper-parameter selection.

6.1. **Text classification algorithm based on SVM.** Text classification is an effective means to acquire knowledge from massive amounts of information. In natural language processing and content information filtering [41], there are many applications for text classification. As a standard way of representing words and sentences in the text as vectors in the feature space, the Vector Space Model (VSM) [42] is commonly used to process the text. We classify the text using a support vector machine-based text classification method after obtaining the vector representation of the text.

The support vector machine (SVM) is a two-class classification method proposed by Vapnik [43] and Chervonenkis, and further improved by B.E.Boser [44] in 1992, and makes the SVM suitable for many non-linear classification problems. With the popularity of SVM, researchers have expanded them to include multi-classification problems [45].

In text classification, suppose $x_1, x_2, \ldots, x_n$ are partial samples belonging to class X. X is a subset of the text representation space R^n, so the binary text classification model based on support vector machine can be expressed as:

$$\begin{array}{rl} & \min_{w,b} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n} \xi_i \\ \text{s.t.} & y_i\left(w^T\phi\left(X_i\right) + b\right) \geq 1 - \xi_i, \xi_i \geq 0 \end{array} \tag{16}$$

The parameter C in Equation (16) is a hyper-parameter. When C is larger, the generalization ability of the model is stronger, and the probability of model misclassification increases. KernelScale is another algorithm hyper-parameters that is used to standardize the predictor variables by changing the parameter setting scaling ratio. The above situation is regarded as a hyper-parameter's combination optimization problem in this study, and the optimal solution can be obtained heuristically. The algorithm model with the smallest migration loss can be obtained by applying the MSA algorithm to the selection of hyper-parameters.

In addition, this research also adopted other existing heuristic optimization algorithms to solve the problem. As can be seen in Table 15, the statistical results of 30 independent operations are displayed. In order to ensure the validity of the results, the algorithms use a range of uniform parameters. In the table, it is shown that the MSA algorithm performs better than all other algorithms. The results show that the hyper-parameters chosen by the MSA algorithm can effectively reduce the migration loss of the text classification model.

TABLE 15. Statistical analysis of algorithms for application of text classification

		MSA	GSA	PSO	ABC
Loss	Best	**1.6248E-04**	8.3503E-03	3.7123E-04	2.1349E-04
	Worst	1.9611E-02	9.9445E-01	**4.0280E-03**	2.1723E-01
	Mean	**4.4871E-04**	4.6871E-01	1.9323E-03	4.2301E-02
	SD	4.2199E-02	2.9541E-01	**7.5146E-04**	4.2793E-02

7. **Conclusion.** The work presented a novel metaheuristic algorithm based on the migratory behavior of animals in nature, which can avoid local optima while fully exploring by simulating the behavior of animal migratory movements. The algorithm was tested in twenty-six test functions, and the its results showed that MSA can could achieve competitive results in comparison to other well-known metaheuristics such as PSO, GA, GSA, and ABC. First, the MSA's optimization ability was confirmed by the results of the unimodal function tests. Second, the MSA algorithm's superior exploration ability was shown by the test results on multimodal functions. Finally, the MSA showed that the corresponding competitiveness and recommended local optimums avoidance in the CEC2014 benchmark test functions.

In addition, the MSA algorithm performed admirably in the process of hyper-parameter optimization of the text classification algorithm. When compared with the current methods, MSA showed a significant improvement when dealing with practical issues, indicating the application value of the algorithm. In the future work, the main research direction is to use the algorithm to solve more practical problems, such as hyperparameter optimization problem, multi-objective optimization problem, knapsack problem and optimal parameter selection problem in natural language processing.

Funding Statement: This work was supported by Jilin Science and Technology Innovation Development Program Projects (No. 20190302202).

REFERENCES

[1] E. Osaba, E. V. Rodriguez, J. D. Ser, A. J. Nebro, D. Molina, "A tutorial on the design, experimentation and application of metaheuristic algorithms to real-world optimization problems," *Swarm and Evolutionary Computation*, vol. 64, 100888, 2021.
[2] A. S. Azad, M. S. A. Rahaman, J. Watada, P. Vasant, and J. A. G. Vintaned, "Optimization of the hydropower energy generation using metaheuristic approaches: A review," *Energy Reports*, vol. 6, pp. 2230-2248, 2020.
[3] S. A. Jalaee, A. Shakibaei, H. R. Horry, H. Akbarifard, A. GhasemiNejad, "A new hybrid metaheuristic method based on biogeography-based optimization and particle swarm optimization algorithm to estimate money demand in iran," *MethodsX*, vol. 8, 101226, 2021.
[4] L. Kang, R.-S. Chen, N. Xiong, Y.-C. Chen, Y.-X. Hu, C.-M. Chen, "Selecting Hyper-Parameters of Gaussian Process Regression Based on Non-Inertial Particle Swarm Optimization in Internet of Things," *IEEE Access*, vol. 7, pp. 59504-59513, 2019.
[5] T.-T. Nguyen, T. Dong-Nguyen, T.-G. Ngo and V.-T. Nguyen, "An Optimal Thresholds for Segmenting Medical Images Using Improved Swarm Algorithm," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 13, no. 1, pp. 12-21, 2022.
[6] P.-C. Song, S.-C. Chu, J.-S. Pan, T.-Y. Wu, "An Adaptive Stochastic Central Force Optimization Algorithm for Node Localization in Wireless Sensor Networks," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 39, no. 1-2, pp. 1-19, 2022.
[7] T.-Y. Wu, C.-W. Lin, Y. Zhang, C.-H. Chen, "The Grid-Based Swarm Intelligence Algorithm for Privacy-Preserving Data Mining," *Applied Science*, vol. 9, 774, 2019.

[8] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *MHS'95. Proc. of the Sixth International Symposium on Micro Machine and Human Science*, 1995, pp. 39-43.
[9] B. Basturk, "An artificial bee colony (abc) algorithm for numeric function optimization," in *IEEE Swarm Intelligence Symposium.* Indianapolis, 2006.
[10] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671-680, 1983.
[11] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "Gsa: A gravitational search algorithm," *Information Sciences*, vol. 179, pp. 2232-2248, 2009.
[12] J. H. Holland, "Adaptation in natural and artificial systems," Ph.D. dissertation, University of Michigan, 1975.
[13] R. Storn and K. Price, "Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341-359, 1997.
[14] J. E. Naranjo, F. Serradilla, and F. Nashashibi, "Speed control optimization for autonomous vehicles with metaheuristics," *Electronics*, vol. 9, 551, 2020.
[15] B. L. G. Costa, C. L. Graciola, B. A. Ang'elico, A. Goedtel, and M. F, Castoldi, "Metaheuristics optimization applied to pi controllers tuning of a dtc-svm drive for three-phase induction motors," *Applied Soft Computing*, vol. 62, pp. 776-788, 2018.
[16] M. Abdel-Basset, R. Mohamed, M. Abouhawwash, R. K. Chakrabortty, M. J. Ryan, Y. Nam, "An Improved Jellyfish Algorithm for Multilevel Thresholding of Magnetic Resonance Brain Image Segmentations," *CMC-Computers, Materials & Continua*, vol. 68, pp. 2961-2977, 2021.
[17] J. Sun, J. M. Garibaldi, and C. Hodgman, "Parameter estimation using metaheuristics in systems biology: a comprehensive review," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, pp. 185-202, 2011.
[18] Z. A. A. Alyasseri, A. T. Khader, M. A. Al-Betar, A. K. Abasi, and S. N. Makhadmeh, "Eeg signals denoising using optimal wavelet transform hybridized with efficient metaheuristic methods," *IEEE Access*, vol. 8, pp. 10584-10605, 2020.
[19] S. Kockanat and N. Karaboga, "The design approaches of two-dimensional digital filters based on metaheuristic optimization algorithms: a review of the literature," *Artificial Intelligence Review*, vol. 44, pp. 265-287, 2015.
[20] S. I. Bejinariu and H. Costin, "A conparison of some nature-inspired optimization metaheuristics applied in biomedical image registration," *Methods of Information in Medicine*, vol. 57, pp. 280-286, 2018.
[21] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, pp. 67-82, 1997.
[22] X. Li, "A new intelligent optimization-artificial fish swarm algorithm," Ph.D. dissertation, University of Zhejiang, 2003.
[23] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, pp 28-39, 2006.
[24] X. Yang, "Firefly algorithms for multimodal optimization," *International Symposium on Stochastic Algorithms*, vol. 5792, pp. 169-178, 2009.
[25] X. Yang, "A new metaheuristic bat-inspired algorithm," *Nature Inspired Cooperative Strategies for Optimization*, vol. 284, pp. 65-74, 2010.
[26] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances In Engineering Software*, vol. 69, pp. 46-61, 2014.
[27] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51-67, 2016.
[28] Q. Yang, S. Chu, J.-S. Pan, C.-M. Chen, "Sine Cosine Algorithm with Multigroup and Multistrategy for Solving CVRP," *Mathematical Problems in Engineering*, vol. 2020, 8184254, 2020.
[29] L. Kang, R.-S. Chen, Y.-C. Chen, C.-C. Wang, X. Li, T.-Y. Wu, "Using Cache Optimization Method to Reduce Network Traffic in Communication Systems Based on Cloud Computing," *IEEE Access*, vol. 7, pp. 124397-124409, 2019.
[30] X. Qi, Y. Zhu, and H. Zhang, "A new meta-heuristic butterfly-inspired algorithm," *Journal of Computational Science*, vol. 23, pp. 226-239, 2017.
[31] H. Dingle and V. A. Drake, "What is migration?" *Bioscience*, vol.57, pp. 113-121, 2007.
[32] D. J. Aidley, "Animal migration," Ph.D. dissertation, University of Cambridge, 1981.
[33] S. A. G. Jr, "The ecology and evolution of avian migration systems," *Elsevier*, vol. 6, pp. 93-168, 1982.

[34] E. Kaner, *Animals Migrating: How, When, where and why Animals Migrate*, Toronto, Canada: Kids Can Press Ltd, 2005.

[35] T. Alerstam, "Bird migration," Ph.D. dissertation, University of Cambridge, 1993.

[36] A. B. Barron, R. Maleszka, R. K. V. Meer, and G. E. Robinson, "Octopamine modulates honey bee dance behavior," *Proceedings of the National Academy of Sciences*, vol. 104, pp. 1703-1707, 2007.

[37] E. H. Hess, "Imprinting," *Science*, vol. 130, pp. 133-141, 1959.

[38] L. Hongqun, Z. Yuhui, and L.Baoguo, "Review on dominance hierarchy of non-human primates," *Shou Lei Xue Bao = Acta Theriologica Sinica*, vol. 24, pp. 53-60, 2004.

[39] J. J. Liang, B. Y. Qu, and P. N. Suganthan, "Problem definitions and evaluation criteria for the cec 2014 special session and competition on single objective real-parameter numerical optimization," *Zhengzhou University and Nanyang Technological University*, vol. 63, 4905, 2013.

[40] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big data*, vol. 3, pp. 1-40, 2016.

[41] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, "Text classification algorithms: A survey," *Information*, vol. 10, 150, 2019.

[42] T. Danisman and A. Alpkocak, "Feeler: Emotion classification of text using vector space model," *Interaction and Social Intelligence*, vol. 1, 53, 2008.

[43] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, 1995.

[44] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Fifth Annual Workshop on Computational Learning Theory*, 1992, pp. 144-152.

[45] C. Hsu and C. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, pp. 415-425, 2002.