

Dynamic Data Migration Strategy for Cloud Computing Based on Improved Mayfly Algorithm

Bao-Fu Gong^{1,*}, Jie Yang¹, Cheng-Qing Feng²

¹Library, Criminal Investigation Police University of China, Shenyang 110854, China
gongbaofu525@163.com yangjie18940177832@163.com

²Faculty of Social Sciences and Humanities, Thammasat University, Bangkok 10330, Thailand
ax6399@163.com

*Corresponding author: Bao-Fu Gong

Received July 10, 2024, revised December 4, 2024, accepted March 9, 2025.

ABSTRACT. Existing dynamic migration algorithms for cloud computing data are inefficient, time-consuming, and the migration cost is too high, for this reason, a dynamic migration strategy for cloud computing data based on Improved Mayfly Algorithm (IMA) is suggested. Firstly, to address issues of uneven distribution of the standard MA population and poor optimization performance, a chaotic inverse learning strategy is used to build up the diversity of the population, and in the mayfly updating stage, adaptive inertia weights are used to improve the mayfly position updating formula to enhance the convergence speed of the algorithm; in order to ensure that the range of the optimization search is expanded, the Kersey variant of the individual is perturbed with the elitist strategy, to prevent the MA from falling under the local optimum. Based on the above, a dynamic data migration framework for cloud computing is constructed to transform the issue of realizing a dynamic data migration strategy into a multi-objective optimization issue constrained by performance, migration cost and bandwidth. The IMA is used to update the location of individuals within the data center area and find the optimal solution, and the optimality search range is expanded by the fitness function and Cauchy's variation perturbation to minimize the cost of dynamic data migration in cloud computing. In this article, the CloudSim platform is applied for simulation validation, and the results indicate that the suggested strategy reduces the time consumption to about 1/3 of the traditional method, and the data migration efficiency is the highest.

Keywords: Mayfly algorithm; Cloud computing; Data dynamic migration; Adaptive inertia weight; Cauchy variation

1. Introduction. Cloud computing, a research direction, is getting more and more widespread attention. With the explosive growth of cluster size and data volume, large-scale cloud data centers have successively appeared in various industries [1]. In cloud data centers, the soaring information volume makes network bandwidth a serious bottleneck problem. Data migration is a key part to ensure the efficient operation, smooth upgrade and update system of cloud data center, and it also occupies a pivotal position in terms of cloud computing [2]. Data movement approach is an important prerequisite for data movement and a strong insure for constant and effective operation of the system in the future [3, 4]. When a server is overloaded, the data can be shared to the proper node equipment in terms of the dynamic migration approach to gain burden balancing. When some data is too heavily accessed for some reason, network bandwidth congestion can

cause the cloud data core to be unable to offer effective services [5]. At this point it is important to design an efficient dynamic data migration strategy to solve this problem.

1.1. Related work. Dynamic data migration for cloud computing has become a growing concern. Zhang et al. [6] suggested a process-oriented data migration scheme for cloud data-intensive demands. Laili et al. [7] offered a scheduling method relied on time period and expense for a single type of task. Aggarwal et al. [8] proposed a data grid for accessing, moving and modifying massive datasets in cloud storage resources using structured data. Terefe et al. [9] used cloud computing model and Markov process theory to give a probabilistic algorithm for data migration to solve the load imbalance problem. Kanniga Devi et al. [10] suggested a bi-qualified value scheduling strategy for dynamic migration of VMs for load balancing with a high time consumption. Li et al. [11] give a correlation-based two-phase effective data placement approach and task scheduling approach. Periola et al. [12] propose an algorithm to migrate big data in mixed cloud storage on the basis of the ocean's life cycle, which effectively reduces the cost of data management. Gao and Wu [13] solved for the time consumption, amount of network accesses and burden balancing in the cloud dynamic data migration approach, respectively, and achieved good results in the CloudSim platform. Mansouri et al. [14] partitioned the dataset collection into intervals based on the dynamic data clustering matrix to achieve the data migration, but it was time-consuming and inefficient.

Population intelligence optimization methods have gradually been a significant study direction for the optimal solution of complex engineering problems due to their parallelism and distributed advantages. Tseng et al. [15] solved the problem by using the number of physical hosts and the number of VM migrations as the objective parameters through Genetic Algorithm (GA), but ignored the loading of the physical hosts. Chawla et al. [16] obtained the dynamic data through the GA's cross-variance operation, and migrated the load data from the main region of the loaded system to the region with low load area. Liu et al. [17] proposed an improved multiobjective VM placement algorithm based on ant colony algorithm. Dynamic data is obtained to migrate the load data from the main region of the loaded system to the region where the load region is low. Khan and Santhosh [18] suggested an adaptive Particle Swarm Optimization (PSO) approach to achieve efficient deployment of VMs in cloud environments. Suja and Booba [19] designed a heuristic dynamic migration strategy for VMs based on Simulated Annealing (SA) idea, ignoring the cost spend of migration. The aforementioned population intelligent optimization algorithms GA, PSO, etc., have their own limitations. Mayfly Algorithm (MA) achieves population search in solution space by simulating the flight and mating behaviours of mayflies, and combines the benefits of PSO, GA, and SA, which has the features of high optimization search accuracy and fast convergence [20]. Justin and Karuppusamy [21] applied the MA to the load seeking problem among multiple servers, and set a reasonable objective function to perform global search, but the exploration ability is insufficient at the later stage and falls into the local optimum.

1.2. Contribution. Existing research does not consider migration cost and bandwidth pressure, so it fails to solve the bandwidth bottleneck problem fundamentally. In this paper, we aim to maximize the performance and save the migration cost by finding a destination server to achieve the migration of cloud computing data through dynamic data migration strategy. To this end, a dynamic migration strategy for cloud computing data based on Improved Mayfly Algorithm (IMA) is suggested. Firstly, the traditional MA is optimized to increase the population diversity by using chaotic mapping, and the adaptive inertia weights are used to improve the mayfly position update formula to balance the global exploration of the algorithm. To ensure the expansion of the range of

the search for optimality, the Cauchy’s variance perturbation strategy is used to rank the offspring mayflies, avoiding the MA falling under the local optimum. Then, relied on the IMA, a dynamic data migration framework is constructed to transform the dynamic data migration strategy issue into a multi-objective optimization issue subject to performance, migration cost and bandwidth constraints. The IMA is used to update the position of individuals and find the optimal solution to minimize the cost of dynamic data migration in cloud computing. From the actual test results, the designed strategy has lower hardware resource consumption, and the real-time migration efficiency and cost are better than the three traditional migration algorithms under the same conditions.

2. Theoretical analysis.

2.1. Big data migration in the cloud. The architecture of the vast majority of cloud computing paradigm systems will be divided into a four-tier structure, i.e., resource, virtualization, management and service tiers [22], as implied in Figure 1.

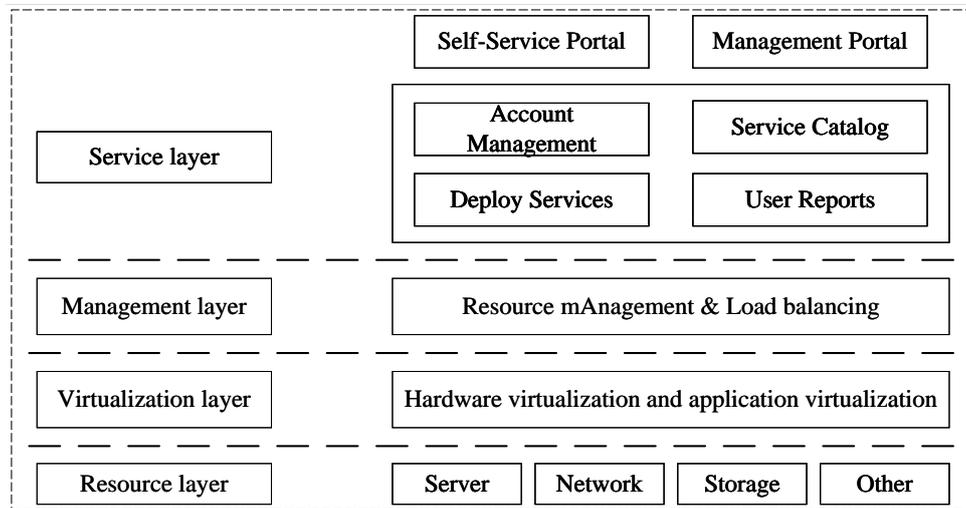


Figure 1. Architecture of cloud computing system

The big data migration in this architecture involves load balancing and bandwidth utilisation of the cloud platform. As the objective of this paper, data migration is a technology that integrates offline storage with online storage, and its process can be roughly divided into three steps: extraction, conversion, and loading. The big data migration problem based on cloud computing architecture can be regarded as the optimal solution of n data migrations between m servers. Let Sum be the sum of n data sizes to be migrated, i.e., $Sum = \sum^n M_{ij}$, and T be the data transfer time, $T = \frac{Sum}{bandwidth}$, where M_{ij} denotes the j -th data to be moved in the i -th server.

2.2. Mayfly algorithm. MA is a novel class of population intelligence optimization method, which is with the advantages of good optimality searching performance and fast convergence [23], and is of high study value.

(1) Motion of a male mayfly. Let t be the existing location of the mayfly in the seek space, and the position at the next moment is x_i^{t+1} , with velocity v_i^{t+1} , having $x_i^{t+1} = x_i^t + v_i^{t+1}$. The velocity of the population-optimal male is defined as $v_i^{t+1} = v_i^t + dr$, d is the dance coefficient, and r is a arbitrary number among $[-1, 1]$. The velocity of a male mayfly with a poor fitness value is defined as follows:

$$v_{ij}^{t+1} = v_{ij}^t + \beta_1 e^{-\alpha r_p^2} (pbest_{ij} - x_{ij}^t) + \beta_2 e^{-\alpha r_g^2} (gbest_{ij} - x_{ij}^t) \tag{1}$$

where v_{ij}^t is the speedlight of mayfly i in dimension j at moment t , x_{ij}^t represents the location at moment t , β_1 and β_2 are the positive attraction coefficients of social effects, $pbest_{ij}$ represents the historical ideal location of the mayfly group, $gbest_{ij}$ represents the historical ideal location of the individual mayfly, α is the visibility coefficient of the mayfly, r_p stands for the distance of the existing location from $pbest$, r_g stands for the distance of the existing location from $gbest$.

(2) Motion of a female mayfly. Let the location of female mayfly i at moment t be y_i^t , and the velocity at the next moment be v_{ij}^{t+1} , then having $y_i^{t+1} = y_i^t + v_{ij}^{t+1}$. When the fitness value of female mayfly i is greater than that of male mayfly i , the definition of velocity adopts Equation (2), otherwise using Equation (3):

$$v_{ij}^{t+1} = v_{ij}^t + \beta_2 e^{-\alpha r_{ij}^2} (x_{ij}^t - y_{ij}^t) \quad (2)$$

$$v_{ij}^{t+1} = fl \times r \quad (3)$$

where v_{ij}^t represents the velocity, fl represents the position of the mayfly, rmf represents the distance of the female mayfly i from the male mayfly i , and fl is the arbitrary wandering coefficient.

(3) Mayfly mating. The crossover operator stands for the process of mating two mayflies: one parent is chosen from each of the populations. The result of the crossover produces two offspring, which are produced as bellow.

$$\begin{cases} \text{offspring}_1 = L \cdot \text{male} + (1 - L) \cdot \text{female} \\ \text{offspring}_2 = L \cdot \text{female} + (1 - L) \cdot \text{male} \end{cases} \quad (4)$$

where male is the parent, female is the mother, L is a random number lying in the interval $[-1, 1]$; \cdot denotes the number multiplication.

3. Optimization of MA incorporating adaptive weighting and elite reverse learning. For the goal of enhancing the convergence speed of the MA and enhance its optimization ability, this article adopts the chaotic mapping sequence to optimize the original population and improve the population diversity. The gravity coefficient in the standard MA is improved from a fixed constant to an inertia weight with a cosine function to regulate the search and capability of the algorithm. Finally, an elite back-learning center of gravity perturbation strategy is used to perturb the generated offspring mayflies to avoid the MA falling under a local optimum.

(1) **Chaotic mapping.** This article adopts chaotic mapping to achieve the initialization of the population, the commonly used chaotic mapping to initialize the population are Logistic mapping and Tent mapping, the relevant research outcome implies that Circle chaotic mapping can not only overcome the shortcomings of Logistic mapping and Tent mapping, but also the chaotic sequences generated by it are more uniform [24]. Thus, the Circle chaotic mapping adopted to generate the initial population is defined as follows.

$$x_{i+1} = x_i + 0.2 - \text{mod}(0.5 \sin(2\pi x_i), 1) \quad (5)$$

where x_{i+1} denotes the position after mapping, x_i denotes the original position of the target, and i denotes the dimension.

(2) **Adaptive weights.** Inertia weight is an important parameter in the MA velocity update formula, while the inertia weight is set as a constant in the standard MA, and the iterative process cannot make a good adjustment of the mayfly velocity according to the actual situation. In this chapter, adaptive inertia weights based on cosine function [25]

are suggested to better remain the universal search capability and local victimization capability. The expression of the adaptive weighting factor is as follows.

$$g(t) = \sin\left(\frac{\pi}{2}\left(1 - \frac{t}{\text{MaxIt}}\right)\right) \tag{6}$$

where t is the existing amount of iterations and MaxIt is the maximum amount of iterations.

The equation for calculating the speed of a male mayfly is as bellow.

$$v_{ij}^{t+1} = g \times v_{ij}^t + a_1 e^{-\alpha r_1^2} (pbest_{ij} - x_{ij}^t) + a_2 e^{-\alpha r_2^2} (gbest_j - x_{ij}^t) \tag{7}$$

The equation for computing the speed of a female mayfly is as bellow.

$$v_{ij}^{t+1} = \begin{cases} g \times v_{ij}^t + a_2 e^{-\alpha r_{ij}^2} (x_{ij} - y_{ij}), & f(y_i) > f(x_i) \\ g \times v_{ij}^t + fl \times r, & f(y_i) \leq f(x_i) \end{cases} \tag{8}$$

(3) **Elite reverse learning based center of gravity perturbation strategy.** For the goal of improving the ability of MA to jump out of the local optimum, the elite center of gravity perturbation strategy [26] is used, which firstly solves for the center of gravity of the population individuals, and secondly sorts the offspring produced by mating male and female mayflies, and selects the top n elite individuals to be perturbed. The offspring mayfly individual of the current iteration number is $cd = (cd_1, cd_2, \dots, cd_n)$, and its i -th dimensional post-perturbation solution is $cd^* = (cd_1^*, cd_2^*, \dots, cd_n^*)$, where cd_i^* is implied below.

$$\begin{cases} cd_i^* = C + RD \times \vartheta \times (C - cd) \\ RD = \frac{F(cd_i) - F(gbest)}{F(C) - F(gbest) + \mu} \end{cases} \tag{9}$$

where C is the center of gravity of the population, ϑ is a arbitrary number on $(0, 1)$, F is the fitness function, RD is the perturbation factor, and μ is a very small positive number.

The center of gravity perturbation strategy can expand the search space, increase the diversity of the population and improve the convergence capability of the MA. For the current population, the first m mayflies with the best fitness values are selected for Cauchy's mutation [27], and if the fitness values of mayflies after the mutation are better than those before the mutation, they are replaced; otherwise, the mayflies before the mutation are still used.

4. Dynamic data migration strategy for cloud computing based on improved mayfly algorithm.

4.1. **Dynamic data migration framework construction in cloud computing.** Based on the above optimized MA, this article designs an efficient dynamic data migration approach for cloud computing, as implied in Figure 2. Firstly, the cloud computing dynamic data migration framework is constructed, then mathematical modelling is carried out for the load balancing problem, and finally, the IMA is used to update the location of individuals within the data center area and search for the optimal solution, introduce the fitness function and nonlinear inertia weight optimization algorithm, and expand the range of the search for the optimal solution, so as to achieve the minimization of the cost of cloud computing data dynamic migration.

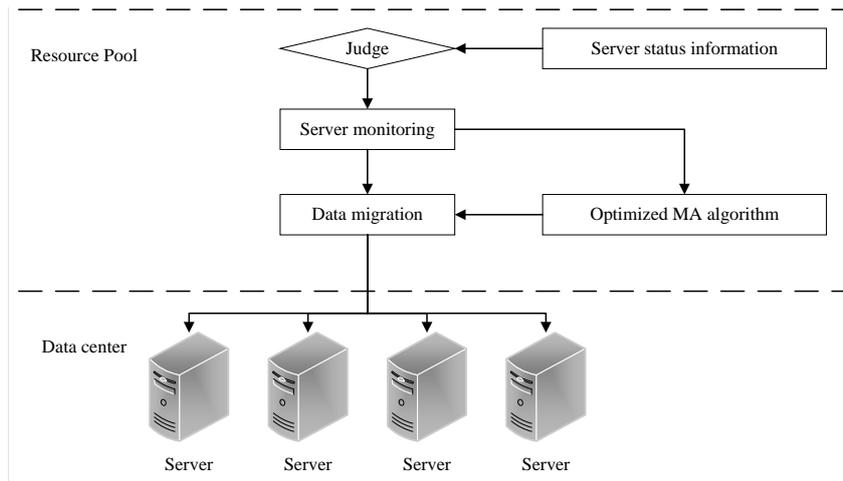


Figure 2. The model of the suggested dynamic data migration strategy for cloud computing

The performance of different calculating devices for the cloud calculating environment has a big difference, so it is essential to make appropriate scheduling for the dynamic movement procedure of data to guarantee that various calculating devices can accomplish the related amount of courses [28]. In a cloud calculating environment, the entire migrated data is assigned to some parts and saved in various scheduling cores to enhance the effectiveness and dependability of data movement. When many users require resources in the meantime, the IMA are required to balance the migration work between occupied servers and leisure servers under this framework. The dynamic migration framework for cloud computing data designed in this article is implied in Figure 3.

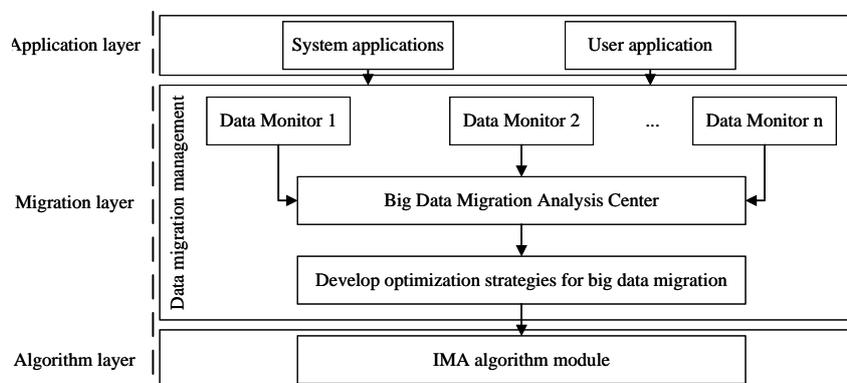


Figure 3. The framework of the dynamic migration framework

There are n data monitors in the framework, which collect real-time data on the load conditions of the network nodes and transmit them to the data analysis center, which formulates scheduling commands and optimization policies according to the current data flow of the nodes and shares the commands to the algorithmic layer. It is assumed that the cloud environment consists of a total of m data centers $c_i, i = 1, 2, \dots, n$. This is denoted as $C = \{c_1, c_2, \dots, c_n\}$, where C is the data center. The network topology of the nodes is analyzed based on the center C , a framework model is built, and the dynamic migration of cloud data is load-balanced and scheduled based on the IMA.

4.2. Description and analysis of dynamic data migration problem in cloud computing. In a migration framework system with m data centers, some of the frequently

accessed nodes will be overburdened, leading to a degradation of the overall system performance. In this paper, the problem is formalized as an issue of migrating n data to m servers of the cloud data core. The solved way is able to be expressed as the n -dimensional vector, that every element stands for the position of the goal server of the migrated data. Define a quaternion $X = \{D_i, DC, B, BS\}$. D_i denotes the n migrated data in the i -th node, represented as $D_i(n, \phi) = \{D_{i1}, D_{i2}, \dots, D_{in}\}$, $i \in [1, m]$, ϕ denotes the point in time, and D_{ij} denotes the j -th to-be-migrated data in the i -th server. DC denotes the m accessible server, represented as $DC(m, \phi) = \{DC_1, DC_2, \dots, DC_m\}$. $B(m, \phi)$ denotes the network bandwidth available to m nodes at time ϕ , denoted as $B(m, \phi) = \{B_1, B_2, \dots, B_m\}$. The chief purpose of the issue is to discover candidate server positions, represented as $BS[D_{ij}, DC_p, \xi_j, \varphi]$, ($i \neq p$), that maximize performance and minimize migration cost, and then select the position server that stores the most bandwidth in the whole data center.

Sum stands for the sum of n data sizes to be migrated, as follows.

$$Sum = \sum_{i=1}^n S(D_{ij}) \quad (10)$$

T denotes the time needed for data transmittal and is denoted as follows.

$$T = \frac{Sum}{Bandwidth} \quad (11)$$

ρ denotes the current bandwidth usage of the server node, denoted as follows.

$$\rho = \frac{T_{waited}}{T_{waited} + \sin \theta T_{background}} \times 100\% \quad (12)$$

To design an efficient and reliable algorithm for finding the dynamic data migration location policy, it is first necessary to determine the solution representation problem. In this paper, there are n data to be migrated to m servers, and each generation of mayflies is represented as an array, denoted as $Z^g[n]$, and each element in the array represents the solution found by each mayfly, i.e., $Z^g[i] = (a, b)$. a stands for the position of the node where the data to be migrated is located, and b stands for the position of the selected goal node.

4.3. A dynamic migration strategy for cloud data based on IMA. On the basis of the above problem analysis, this article suggests a strategy based on the IMA to deal with the dynamic migration problem of cloud computing data. Each server is generated as a population in the outline of a mayfly individual, and by designing a target operation that can illustrate the cloud computing data resource employment, migration cost and bandwidth as the fitness function of the IMA process, the optimal solution is converged through an iterative approach.

(1) Determine the data to be migrated. This policy sets up that when the cloud data is accessed for some reason, the data is selected for migration and a migration request is sent.

(2) Monitor obtains data migration requests: The monitor obtains migration requests and information about the server's resource utilization in cycle θ . This information is represented by the quaternion $X = \{D_i, DC, B, BS\}$ mentioned above. At the end of a cycle θ , the monitor passes the relevant parameters needed for the policy to the IMA module in the architecture.

(3) Selection of the optimal goal server node for the to-be-migrated data based on the IMA. The IMA module receives the required parameter information and starts running the algorithm.

Step 1: Initialization parameters, population size N , iteration number $Iter$, dance coefficient $dance$, flight coefficient β . Initialization population is generated by Circle chaotic mapping according to Equation (13), and mayfly position and velocity are initialized.

$$\begin{cases} x_n = 2y'_n(1 - y'_n) \\ y_n = \frac{1}{\pi} \arcsin(2x_{n+1} - 1) - \frac{1}{2} \end{cases} \quad (13)$$

To further expand the mayfly population and increase its diversity, the initial position calculation of the mayfly population was carried out as implied below.

$$Z_i = a_i + (b_i - a_i)x_n \quad (14)$$

where a_i and b_i stand for the lower and upper bounds of the seek space.

Step 2: Update the velocities of male and female mayflies according to Equation (1) and Equation (2), and calculate the iterative update positions of female and male mayflies by $x_i^{t+1} = x_i^t + v_i^{t+1}$ and $x_i - X_i = \sum_{j=1}^n (x_{ij} - X_{ij})^2$ respectively, meanwhile, introduce the location of female mayflies in the target operation, and finally calculate the fitness value τ of female mayflies.

$$\tau = \begin{cases} g \times v_{ij}^t + \beta e^{-\alpha r_p^2} x_{ij}^t + TD(t)^d \cdot [g \times v_{ij}^t + a_1 e^{-\alpha r_p^2} (pbest_{ij} - x_{ij}^t)] , if(x_i) > f(gbest) \\ g \times v_{ij}^t + d' * r + TD(t)^d \cdot [g \times v_{ij}^t + d' \times r] , if(x_i) \leq f(gbest) \end{cases} \quad (15)$$

where t is the existing amount of iterations; $TD(t)$ is a t-distribution with a degree of freedom parameter of t ; and $TD(t)d$ is the random number generated by the t-distribution variational operator in the d -th dimension.

Step 3: Establish the link between the fitness τ_i of mayfly i and the value of the target operation, which is always described by the target operation τ_i . The better the target operation is, the larger τ is, i.e., $\tau_i = f(x_i^g)$, $f(x_i^g)$ are defined as bellow.

$$f(x_i^g) = acrDC_i - \xi_{ij} \quad (16)$$

where $acrDC_i$ denotes the resources available to the i -th node, $\xi_{ij} = \eta \times t_{ij}$ denotes the cost of migration from server i to server j , η denotes the migration spend per minute, and t_{ij} denotes the time required for migration.

Step 4: According to Equation (4) mayfly mating produces offspring.

Step 5: Combine the mayflies $pbest$ and $gbest$, rank the fitness, select the elite individuals, and generate the elite inverse solution according to Equation (17).

$$\bar{x}_i^e = \begin{cases} k * (za_i + zb_i) - x_i^{best}; a_i \leq \bar{x}_i^e \leq b_i \\ \frac{1}{N} \sum_{i=1}^N x_i * \lambda; \bar{x}_i^e \leq a_i \text{ or } \bar{x}_i^e \geq b_i \end{cases} \quad (17)$$

Step 6: If the current mayfly meets the Equation (17), it can be judged as an inferior mayfly individual, and then randomly generate random $w\%$, $w\% \in [0, 1]$. If $w\% > p$, then Cauchy mutation is performed according to Equation (18). Use the n -th line of Equation (19) to filter the updated population and eliminate the individuals with low fitness values.

$$x_i^{t+1} = \begin{cases} x_i^t + \text{cauchy}(0, 1) \oplus v_i^{t+1}; & w\% > p \\ x_i^t; & w\% \leq p \end{cases} \quad (18)$$

$$f(x_i^{t+1}) - f(x_i^t) < \frac{1}{N} \sum_{x_i \in N} (f(x_i^{t+1}) - f(x_i^t)) \quad (19)$$

Step 7: When the maximum amount of iterations is achieved, end the procedure, at this time output the solution corresponding to the optimal fitness, that is, the final destination server for each data to be migrated, otherwise go to Step 2 for the next iteration.

(4) Send migration commands to the migrator: The optimal solution from the previous process is sent to the migrator to migrate the cloud data, thus enhancing the bandwidth usage of the whole data center.

5. Experiments and analysis of results.

5.1. Comparison of the efficiency of dynamic data migration. In this article, Cloud Sim, an open-source cloud computing simulation platform for virtualization services, is used to verify and analyze the dynamic data migration strategy of cloud computing relied on the improved MA algorithm. Experimental hardware environment: Windows 7 operating system, Intel(R) Core (TM) i5 CPU, 4 GB RAM, 500G hard disc. The experimental software environment: Eclipse 8.5, Cloud Sim 3.0.3. The population size of MA is 200, the number of female mayflies and male mayflies are 100, the positive attraction constant $a_1 = 1$, $a_2 = 1.5$, the visibility coefficient $\alpha = 2$, the initial random travelling coefficient $fl_0 = 0.1$, and the mutation rate 0.25.

For the goal of evaluating the performance of the proposed strategy in data dynamic migration, the proposed strategy DMIMA is compared and analysed with the CDMGA strategy in literature [16], the AVMCA strategy in literature [17] and the CMPSO strategy in literature [18]. The methods used in the experiments are all iterated 1500 times, and when the data size is 15GB, the comparison of the data migration consumption time of the four methods is shown in Figure 4. CDMGA and CMPSO methods require the longest migration time due to the high time complexity of multiple iterations. AVMCA's data migration consumption time is 77s. The suggested strategy consumes the least amount of time, which is about 1/3 of the time consumed by the traditional methods, and the data migration efficiency is the highest.

5.2. Cost analysis of dynamic data migration. In the experiments, the migration cost is calculated by comparing the positions of cloud computing data randomly moved into the cloud, and the product of the data migration time and the migration overhead per unit of time (set as 1 s) is defined as the migration cost. The dynamic migration algorithms and scheduling schemes will determine the final migration cost, and the migration costs of four different migration algorithms are shown in Table 1 for a migration period of 0-100 s. The migration costs of four different migration strategies are implied in Table 1. The DMIMA strategy has a lower cost per time period than the other three methods, and the rate of increase of the cost overhead remains low as the time period increases.

The upload and download times of the four strategies for data of different sizes are implied in Table 2. CDMGA consumes a long time in the load balancing process, resulting in low efficiency of file uploading. AVMCA improves the efficiency of load balancing and reduces a certain amount of time consumed, but has a larger overall performance loss. CMPSO has a slower upload and download speed of data. While achieving data load balancing, DMIMA can greatly reduce the overall data load balancing time without any impact on system performance, which has obvious advantages.

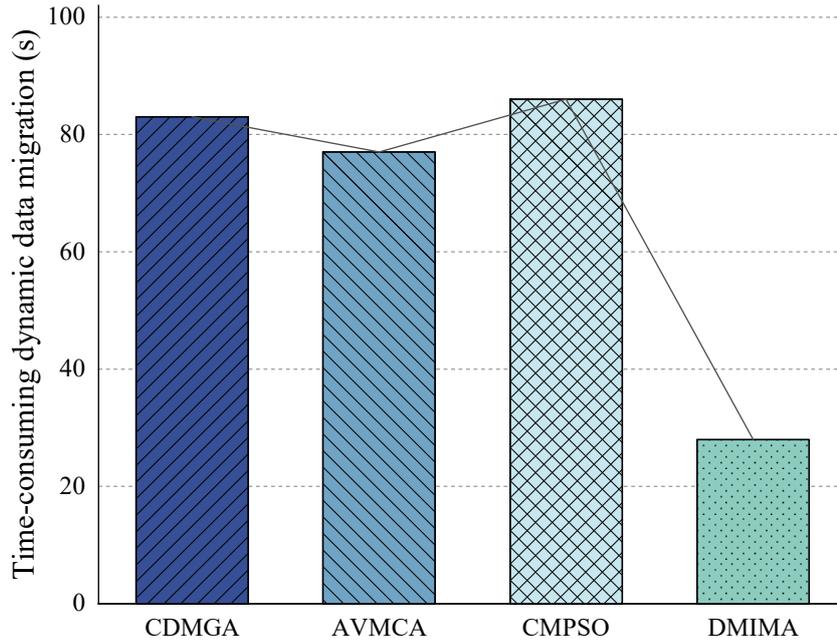


Figure 4. 15.0 GB data size migration elapsed time for each strategy

Table 1. Comparative analysis of migration costs of various strategies

Time period (s)	CDMGA	AVMCA	CMPSO	DMIMA
1–10	10.98	9.47	10.18	6.59
11–20	11.64	9.95	12.42	7.41
21–30	12.54	10.45	13.38	7.94
31–40	13.08	11.41	14.46	8.32
41–50	14.19	12.68	16.58	8.43
51–60	15.39	13.51	17.48	8.61
61–70	15.92	14.92	18.29	8.79
71–80	16.71	15.66	18.94	8.93
81–90	17.65	17.75	22.59	9.15
91–100	18.86	19.61	24.92	9.36

Table 2. Cost control of data upload and download time

Strategy	Data upload time	Data download time	File download processing efficiency (%)
CDMGA	231.7	197.1	67.8
AVMCA	258.4	154.4	74.9
CMPSO	328.1	187.3	63.1
DMIMA	148.9	79.8	93.5

5.3. Occupancy of node hardware resources. For nodes in cloud computing, frequent user accesses and high load on hardware resources of nodes will increase the occupancy of hardware resources, such as CPU and memory, and if the load on hardware is too high, it will reduce the performance of data storage and scheduling of nodes. With the same data size, we observe the CPU and memory consumption of nodes under different migration algorithms, as shown in Figure 5 and Figure 6.

From the results of hardware resource statistics of nodes in real-time dynamic migration of cloud computing data, it can be seen that IMA performs well in terms of both CPU occupancy rate and memory occupancy rate. In the dynamic migration of cloud

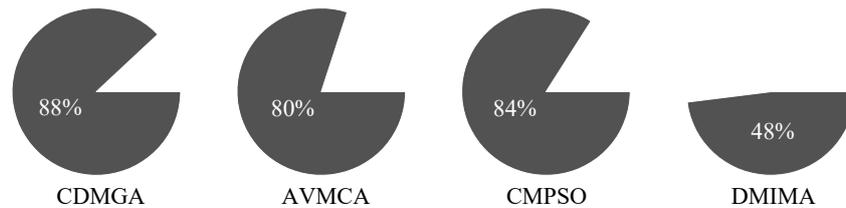


Figure 5. CPU usage of the node

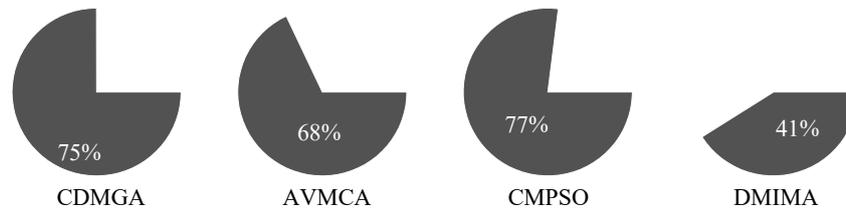


Figure 6. RAM usage of the node

computing data, the hardware resources of the nodes are reserved for a larger space, so that the performance of the system can be more fully utilized, and the computing tasks of some high-load nodes can be afforded to improve the efficiency of the dynamic migration of data.

6. Conclusion. Big data applications in cloud computing environments encounter various problems in dynamic data migration, mainly in terms of how to reduce global time consumption during the migration process. To this end, a dynamic migration strategy for cloud computing data based on an IMA is proposed. Intending to the issues of slow convergence speed and poor optimality seeking performance of the traditional MA, adaptive inertia weights and elite backward learning are fused to optimize MA. The IMA is adopted to update the location of individuals and find the optimal solution within the data center area, and the optimality seeking range is expanded by the fitness function and Cauchy's variation perturbation to minimize the cost of dynamic migration of cloud computing data. The test results indicate that the suggested strategy is more efficient in migrating with larger data sizes, and at the same time has lower hardware resource occupancy.

Acknowledgment. This work is supported by the Economic and Social Development Research Project of Liaoning Province (No. 2024lslybkt-071).

REFERENCES

- [1] A. Katal, S. Dahiya, and T. Choudhury, "Energy efficiency in cloud computing data centers: a survey on software technologies," *Cluster Computing*, vol. 26, no. 3, pp. 1845-1875, 2023.
- [2] R. Amin, S. Vadlamudi, and M. M. Rahaman, "Opportunities and challenges of data migration in cloud," *Engineering International*, vol. 9, no. 1, pp. 41-50, 2021.
- [3] L. Gkatzikis, and I. Koutsopoulos, "Migrate or not? Exploiting dynamic task migration in mobile cloud computing systems," *IEEE Wireless Communications*, vol. 20, no. 3, pp. 24-32, 2013.
- [4] A. Gupta, and S. Namasudra, "A novel technique for accelerating live migration in cloud computing," *Automated Software Engineering*, vol. 29, no. 1, 34, 2022.
- [5] U. Mandal, P. Chowdhury, M. Tornatore, C. U. Martel, and B. Mukherjee, "Bandwidth provisioning for virtual machine migration in cloud: Strategy and application," *IEEE Transactions on Cloud Computing*, vol. 6, no. 4, pp. 967-976, 2016.
- [6] X. Zhang, Z. Hu, M. Zheng, and J. Li, "A novel cloud model based data placement strategy for data-intensive application in clouds," *Computers & Electrical Engineering*, vol. 77, pp. 445-456, 2019.

- [7] Y. Laili, F. Tao, F. Wang, L. Zhang, and T. Lin, "An iterative budget algorithm for dynamic virtual machine consolidation under cloud computing environment," *IEEE Transactions on Services Computing*, vol. 14, no. 1, pp. 30-43, 2018.
- [8] V. Aggarwal, M. Mathur, and N. Saraswat, "Comprehensive Cloud Incremental Data-Application Migration-A Proposed Model for Cloud Migration," *International journal of Computer Applications in Engineering Sciences*, vol. 3, no. 1, 20, 2013.
- [9] M. B. Terefe, H. Lee, N. Heo, G. C. Fox, and S. Oh, "Energy-efficient multisite offloading policy using Markov decision process for mobile cloud computing," *Pervasive and Mobile Computing*, vol. 27, pp. 75-89, 2016.
- [10] R. Kanniga Devi, G. Murugaboopathi, and M. Muthukannan, "Load monitoring and system-traffic-aware live VM migration-based load balancing in cloud data center using graph theoretic solutions," *Cluster Computing*, vol. 21, no. 3, pp. 1623-1638, 2018.
- [11] C. Li, J. Liu, W. Li, and Y. Luo, "Adaptive priority-based data placement and multi-task scheduling in geo-distributed cloud systems," *Knowledge-Based Systems*, vol. 224, 107050, 2021.
- [12] A. Periola, A. A. Alonge, and K. A. Ogudo, "Architecture and system design for marine cloud computing assets," *The Computer Journal*, vol. 63, no. 6, pp. 927-941, 2020.
- [13] R. Gao, and J. Wu, "Dynamic load balancing strategy for cloud computing with ant colony optimization," *Future Internet*, vol. 7, no. 4, pp. 465-483, 2015.
- [14] Y. Mansouri, A. N. Toosi, and R. Buyya, "Cost optimization for dynamic replication and migration of data in cloud data centers," *IEEE Transactions on Cloud Computing*, vol. 7, no. 3, pp. 705-718, 2017.
- [15] F.-H. Tseng, X. Wang, L.-D. Chou, H.-C. Chao, and V. C. Leung, "Dynamic resource prediction and allocation for cloud data center using the multiobjective genetic algorithm," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1688-1699, 2017.
- [16] N. Chawla, D. Kumar, and D. K. Sharma, "Improving cost for data migration in cloud computing using genetic algorithm," *International Journal of Software Innovation (IJSI)*, vol. 8, no. 3, pp. 69-81, 2020.
- [17] F. Liu, Z. Ma, B. Wang, and W. Lin, "A virtual machine consolidation algorithm based on ant colony system and extreme learning machine for cloud data center," *IEEE Access*, vol. 8, pp. 53-67, 2019.
- [18] M. S. A. Khan, and R. Santhosh, "Hybrid optimization algorithm for vm migration in cloud computing," *Computers and Electrical Engineering*, vol. 102, 108152, 2022.
- [19] T. L. Suja, and B. Booba, "A Hybrid Metaheuristic approach on Virtual Machine Allocation using Intelligent Backtracking and Simulated Annealing in cloud computing," *Journal of Algebraic Statistics*, vol. 13, no. 3, pp. 3021-3030, 2022.
- [20] D. Chen, and X. Liu, "Mayfly Taylor Optimization-Based Graph Attention Network for Task Scheduling in Edge Computing," *Journal of Grid Computing*, vol. 21, no. 4, 53, 2023.
- [21] S. Justin, and S. K. Karuppusamy, "Mayfly optimization with chaos-based image encryption technique for cloud computing environment," *Journal of Electronic Imaging*, vol. 32, no. 5, pp. 052303-052303, 2023.
- [22] T.-Y. Wu, A. Shao, and J.-S. Pan, "CTOA: Toward a Chaotic-Based Tumbleweed Optimization Algorithm," *Mathematics*, vol. 11, no. 10, p. 2339, 2023.
- [23] T.-Y. Wu, H. Li, and S.-C. Chu, "CPPE: An Improved Phasmatodea Population Evolution Algorithm with Chaotic Maps," *Mathematics*, vol. 11, no. 9, p. 1977, 2023.
- [24] T.-Y. Wu, H. Li, S. Kumari, and C.-M. Chen, "A Spectral Convolutional Neural Network Model Based on Adaptive Fick's Law for Hyperspectral Image Classification," *Computers, Materials & Continua*, vol. 79, no. 1, pp. 19-46, 2024.
- [25] M. Taherkhani, and R. Safabakhsh, "A novel stability-based adaptive inertia weight for particle swarm optimization," *Applied Soft Computing*, vol. 38, pp. 281-295, 2016.
- [26] S. A. Chvatal, G. Torres-Oviedo, S. A. Safavynia, and L. H. Ting, "Common muscle synergies for control of center of mass and force in nonstepping and stepping postural behaviors," *Journal of Neurophysiology*, vol. 106, no. 2, pp. 999-1015, 2011.
- [27] M. Dzherbashian, and A. Nersesian, "Fractional derivatives and Cauchy problem for differential equations of fractional order," *Fractional Calculus and Applied Analysis*, vol. 23, no. 6, pp. 1810-1836, 2020.
- [28] S. Kashyap, and A. Singh, "Prediction-based scheduling techniques for cloud data center's workload: a systematic review," *Cluster Computing*, vol. 26, no. 5, pp. 3209-3