J. COMBIN. MATH. COMBIN. COMPUT. 127b (2025) 9595--9608

Journal of Combinatorial Mathematics and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Combinatorial Press

Research on Constructing Task Scheduling and Processing Models for Complex Data Sets Using Multi-Objective Optimization Algorithm

Zhiqiang Liu[1,✉]

[1] *Shanghai Nanyang Wanbang Software Technology Co., Ltd., Shanghai, 200233, China*

ABSTRACT

In the process of increasing the service capacity of digital infrastructure, the complex data generated by data terminals grows rapidly, which puts forward higher requirements for complex data task scheduling preprocessing. In this paper, based on particle swarm algorithm and improved artificial fish swarm algorithm, a hybrid particle swarm multi-objective optimization scheduling algorithm applicable to task scheduling and processing of complex data sets is designed. Then we design a reasonable expression method for the particle position and adaptation value algorithm in the multi-objective optimization algorithm, and put forward the pre-search strategy of the particle swarm algorithm to improve the search performance of the particles in the algorithm. Finally, the algorithm is equipped to construct a task scheduling and processing model for complex data sets. The results show that the hybrid particle swarm optimization algorithm established in this paper outperforms the comparison model in terms of load balancing and processing time, and is able to keep the system CPU utilization between 0.350-0.491 in the simulation experimental environment. It is also found that the application of the task scheduling and processing model in this paper can increase the power of photovoltaic and wind power generation in the grid system and reduce the operating cost of the grid system. This study provides an effective reference method for the processing of data and task scheduling in various types of complex systems, and brings new ideas and directions for research in related fields.

Keywords: particle swarm algorithm; artificial fish swarm algorithm; multi-objective optimization algorithm; search strategy; task scheduling

1. Introduction

Multi-objective optimization algorithm is an algorithm to solve the optimal solution of multiple objective functions. It is a fuzzy function which can be used to measure the relationship between

✉ Corresponding author.
E-mail address: shlzq2025@163.com (Z. Liu).
Received 05 February 2024; Revised 12 April 2024; Accepted 05 December 2024; Published Online 16 April 2025.
DOI: 10.61091/jcmcc127b-525

multiple objective functions [1-2]. Multi-objective optimization algorithms are also known as multi-objective optimization techniques, which are very different from single-objective optimization algorithms, such as the size of the optimization problem, the relationship between multiple objective functions, and the monotonicity of the objective function [3-5]. Multi-objective optimization algorithms are used to solve a variety of different optimization problems, such as mixed integer optimization, minimum path optimization, optimal dynamic programming, and so on. Multi-objective optimization algorithms can combine the optimal solutions of multiple objective functions to obtain better results [6-9].

Data task scheduling refers to organizing, processing, and analyzing data, and then going through multiple processes for calculation and result output. In this process, reasonable task scheduling can ensure the reasonable operation of the process and improve the processing efficiency [10-13]. At present, there are many different data task scheduling systems, these systems have their own characteristics and applicable scenarios, which can be selected according to the needs. The hardware architecture of the task scheduling center mainly includes servers, storage devices, network devices and so on [14-17]. Server is the core equipment of the task scheduling center, which is used to store and manage task scheduling data and perform task scheduling operations. Storage devices are used to store task scheduling data and ensure data security and reliability. Network devices are used to connect servers and storage devices to ensure data transmission and communication. The servers are deployed in a clustered way, including the main server and the backup server [18-21]. The main server is used to perform task scheduling operations, and the backup server is used to backup task scheduling related data to ensure data security. The storage device adopts disk arrays and network storage devices to ensure high data availability and high transmission speed. The network equipment adopts switches and routers, which are used to connect servers and storage devices to ensure data transmission and communication [22-25].

In this paper, the original random behavior in the artificial fish swarm algorithm is changed to jumping behavior, which enhances the global optimization ability of the artificial fish search, and introduces the survival period and survival index to further improve the optimization efficiency of the algorithm, so as to obtain the improved artificial fish swarm algorithm. The improved artificial fish swarm algorithm is combined with particle swarm algorithm to propose a multi-objective hybrid particle swarm optimization algorithm. The Z-Score standardization method is used to standardize the task execution time and load balancing degree of the algorithm to realize the reasonable encoding of the particle position and the adjustment of the adaptation value calculation in the algorithm. The pre-search solution space strategy is also proposed to improve the search range and quality of the particle population in the multi-objective optimization algorithm, so as to improve the efficiency of task scheduling and processing in complex data sets. Based on this optimization algorithm, a task scheduling and processing model for complex data sets is constructed. In this study, we first compare and analyze the differences in performance between the particle swarm algorithm and the hybrid particle swarm optimization algorithm to verify the improvement effect of the hybrid optimization algorithm. Then, simulation experiments and grid application experiments are used to investigate the effect of the model on task scheduling and processing in complex data sets.

2. Multi-objective hybrid particle swarm optimization algorithm design

2.1. Multi-objective optimization algorithm establishment

The idea of particle swarm algorithm [26] originates from the study of the predatory behavior of birds in the biological world, the particle population is a flock of birds, and each particle represents an individual in the flock. The particle swarm algorithm is characterized by simple ideas, wide range of application and fast convergence speed. This paper combines the idea of improving the artificial fish swarm algorithm to use different position update strategies for population classification to improve the position update process of the particle swarm algorithm, and designs a hybrid particle swarm algorithm.

2.1.1. IAFA algorithm construction. In order to make the artificial fish-schooling algorithm (FASA) [27] better adapted to the problem of task scheduling in complex datasets, an improved artificial fish-schooling algorithm (IAFA) is proposed in this paper. The algorithm cancels the random behavior of the original algorithm, adds the jumping behavior, and introduces the survival period and the survival index, which improves the algorithm's global optimization seeking ability and efficiency. Through the above improvements, the IAFA algorithm can better solve the problem of slow convergence of PSO algorithm when the population size is large.

1) Jumping behavior. In order to further enhance the ability of the artificial fish to search for the global optimum, this paper introduces the jumping behavior to the artificial fish. Firstly, the jump value of the fish group is set to $jump$. After n iterations, it is judged whether the difference in the value of the objective function of the artificial fish is less than $jump$. Here, the difference in the value is the difference between the current optimal value and the optimal value before the iteration, and if the condition is satisfied, some artificial fish are randomly selected and the parameters of these objects are set to make them jump out of the current local optimal state, and the mathematical expression is as follows:

$$X_i(t+1) = X_i(t) + \beta \cdot Visual \cdot Rand(\ ) \tag{1}$$

where β can be set as a mutation parameter as desired.

2) Survival cycle and survival index. The survival time of artificial fish is the survival cycle T, and the mathematical expression is as follows:

$$T = \varepsilon \frac{E_{\max}}{\lambda} \tag{2}$$

where $E_{\max}$ is the optimal food concentration in the fish population, ε is the scale factor, and λ represents the consumption of food per unit time, referred to as the consumption factor.

The survival index h is described as follows:

$$h = \frac{E}{\lambda T}; \begin{cases} h \geq 1 :: AF\_movement \\ h < 1 :: AF_initialize \end{cases} \tag{3}$$

Where, T is the survival cycle of the artificial fish; E is the food concentration at the current location of the artificial fish; and λ is the consumption factor.

With the introduction of the survival cycle, the survivability of the artificial fish changes with its location. When near the global extreme, the artificial fish has the longest life cycle. When located at the local extremes, the artificial fish will gradually die out and respawn in its initial state. This has the advantage of giving the algorithm a wider search space, as well as improving the efficiency of the optimization search and saving the huge storage space brought about by the increase in the number of populations.

2.1.2. IAFA-PSO hybrid algorithm establishment. Assuming that the number of complex data sets is N, the population size of the particle swarm is m, the search space of the particle swarm is the number of complex data sets N, the coordinates of particle $i(i=1,2,\ldots,n)$ in the D-dimensional space can be expressed as $X_i = (X_{i1}, X_{i2}, \cdots, X_{in})$, and the speed can be expressed as $V_i = (V_{i1}, V_{i2}, \cdots, V_{in})$. The most important thing in the multi-objective optimization algorithm is the design of the fitness function [28], which is a criterion for the algorithm to carry out the evaluation, and the multi-objective optimization algorithm fitness function f_i is the effective coverage rate of the data $\left((x_1,x_2,\cdots,x_n)',(y_1,y_2,\cdots,y_n)',(\vec{v}_1,\vec{v}_2,\cdots,\vec{v}_n)'\right)$ under the data set of complex data. Effective coverage, the fitness function f_i of particles i can be defined as:

$$f_i = \eta\left(X_{i1}, X_{i2}, \cdots, \mathrm{X}_{in}\right) \tag{4}$$

Then the global optimum experienced by the population is:

$$P_g = \max\left\{f_1, f_2, \cdots f_m\right\} \tag{5}$$

The artificial fish school size is n, the state of each artificial fish $x_i\left(i=1,2,\ldots,n\right)$ can be represented as vector $X'=\left(x_1,x_2,\cdots,x_n\right)$, and the food concentration at the current location of the artificial fish is $Y_i = f\left(X_i'\right)$, which corresponds to the task scheduling coverage of the complex dataset. The maximum step size for the artificial fish to move is step; the field of view is visual, the crowding factor is δ, and the maximum number of tries per foraging trip is try_number.

As mentioned above, the food concentration of the artificial fish population can be defined as:

$$f_i = \eta\left(X_{i1}', X_{i2}', \cdots, X_{in}'\right) \tag{6}$$

Then the current food concentration of the fish is the largest:

$$Y_i = \max\left\{f_1, f_2, \cdots f_n\right\} \tag{7}$$

Based on the above analysis, the Improved Artificial Fish Swarm and Particle Swarm Hybrid Multi-objective Optimization Algorithm is divided into three phases, firstly, the network is initialized, and then the particle positions and directions are adjusted with the Improved Artificial Fish Swarm Algorithm, and each artificial fish represents a task scheduling method. Finally, the result of optimization when the improved artificial fish swarm algorithm reaches the maximum number of iterations is assigned as input to the particles in the corresponding particle swarm, and the network continues to be optimized with the particle swarm algorithm, and each particle also represents a task scheduling method.

2.2. Particle Position Encoding and Adaptation Value Calculation

Before the execution of the multi-objective optimization algorithm, it is necessary to determine the maximum number of computational resources to be used by the parallelization algorithm based on the number of resources available in the complex dataset. In this paper, all computational resources are encoded in binary form, and the encoded data is represented as the computational resources allocated to each task.

In this paper, we denote whether computational resources are allocated to the current algorithm by 0 and 1 in binary, respectively. Let the total number of resources be M and the number of subtasks in the set of data processing tasks to be executed be N, then there exists a sequence of tasks $T=\left\{T_1,T_2,\ldots,T_N\right\}$. The position information of the population individuals in the scheduling algorithm needs to represent the resource allocation status of each subtask, then the position information of particle D will be in the form of $D=\left\{D_1,D_2,\ldots,D_N\right\}$ of the same length as the sequence of tasks, where D_i represents the computational resources allocated for subtask T_i. D_i will be converted to a binary sequence of the following form to represent the resource allocation:

$$D_i = \left(B_{i1}B_{i2}\ldots B_{iM}\right) \tag{8}$$

Where D_i is the decimal value corresponding to the binary sequence B, which represents the computational resource assigned to task T_i, and its value range is $1 \sim 2^N - 1$ when the total number of resources is N. The length of the binary sequence B represents the maximum number of computational resources, and the number of the value 1 is the current parallelism of the task. B_{ij} is a binary value, and a value of 1 means that task T_i is assigned to computing resource No. j, and a value of 0 means that computing resource No. j is not used for the computation of the task T_i Let the computing resource is always 8, and the position D_1 corresponding to task T_1 may exist as follows $D_1 = 10101010$ represents that computing resources No. 2, 4, 6, and 8 are assigned to the task T_1, and the position information of the particles under this resource allocation strategy is $D = \{170, D_2, \dots, D_N\}$, 170 is the decimal form of the binary sequence 10101010. By controlling the range of values of particle positions can make the scheduling algorithm applicable to task scheduling under different numbers of computing resources in the platform with good robustness.

Hybrid particle swarm algorithm calculates the adaptation value corresponding to the current position of the particle as the criterion for evaluating the advantages and disadvantages of the current position and as the basis for the subsequent iterative update of the population. In this paper, the shortest task execution time and resource load balancing are taken as the optimization objectives, and the adaptation values of particles should reflect the information of task execution time and load balancing degree. In the multi-objective task scheduling problem, there is almost no optimal solution that satisfies different objectives at the same time, in this paper, when calculating the adaptation value, we standardize the task execution time and load balancing degree through the Z-Score standardization method [29]. The normalization formula is as follows:

$$X' = \frac{X - \mu}{\sigma} \tag{9}$$

Where μ is the mean value of the data and σ is the standard deviation of the data, the Z-Score standardized data conforms to the standard normal distribution, satisfying the mean value of 0 and the standard deviation of 1. By standardizing the execution time and the degree of load balancing, the target values of different magnitudes are converted to the same magnitude, ensuring that each value has the same level of influence on the particle adaptation value. The final particle adaptation formula is as follows:

$$Adp = -\left(Makespan' * \lambda_1 + L' * \lambda_2\right) \tag{10}$$

Where, λ_1 and λ_2 are the weights occupied by the running time and resource load respectively to satisfy $\lambda_1 + \lambda_2 = 1$. To satisfy the logic of thinking that the larger the adaptation value the better the particle position, the opposite number is taken for the result after calculating the adaptation.

2.3. Algorithmic pre-generation of search particle strategies

Before the hybrid particle swarm algorithm searches the multi-objective relative optimal solution according to the problem modeling, this paper proposes a strategy of pre-searching the solution space in order to improve the search range and quality of the particle population within a finite number of iterations. Assuming that the target solution space is represented by a two-dimensional space, there exists an optimal solution region D_{best} in the solution space centered on the optimal solution (x_{best}, y_{best}). In addition to the optimal solution region D_{best}, there also exists a poor solution region D_{bad} and a region D_{better} that produces a local optimal solution in the solution space of a complex problem. Based on this situation, in the process of particle optimization, the following judgment can be made on the position (x, y) obtained after each iteration. When the particle position (x, y) is closer to the region D_{best}, it means the position is more optimal. When the particle position (x, y) is closer to the region D_{bad}, it indicates the worse position. The following relationship exists between the population particles:

$$\left(X_i \in D_{best}, X_j \in D_{bad}\right) \Rightarrow P\left(X_i\right) > P\left(X_j\right) \tag{11}$$

Where P is the probability that the particle finds a better solution. In this paper, we restrict the particle position generation and iterative updating process by labeling the D_{bad}-region to ensure that the search particles are always far away from the D_{bad}-region to improve the search efficiency and accuracy, and the particle position updating needs to satisfy the following conditions:

$$X=\begin{cases} f\left(X_i^t\right), f\left(X_i^t\right)\notin D_{bad} \\ reload, f\left(X_i^t\right)\in D_{bad} \end{cases} \tag{12}$$

where f is the particle position generating function. For the population as a whole, the overall optimization process of the population will be more stable after optimizing the position of the population in each generation.

Particles that appear in region D_{bad} during initialization and iteration will repeatedly update the particle position information until the particle is far away from region D_{bad}, ensuring that the population can perform a larger search within the number of iterations. By introducing the strategy of pre-searching the target solution space before generating the formal search particles, the target solution space can be simply divided, which ensures that the formal search particles can stay away from the region where the probability of the existence of the relative optimal solution is low. In the case of limited population size and number of iterations, the population can search for regions where the optimal solution is more likely to occur, which effectively improves the quality of individuals in the population during the position updating process.

3. Complex dataset task scheduling and processing model architecture

In this paper, the above multi-objective optimization algorithm is used to design a task scheduling and processing model for complex datasets, and the specific architecture of the model is shown in Fig. 1. The multi-objective optimization module in the model can adopt different optimization strategies according to different optimization objectives, and the scheduling module can also construct different scheduling models according to different scheduling requirements. The scheduling model is constructed by two optimization objectives of user requirements and dynamic situation of data resources in complex dataset processing, and the scheduling model and strategy can be designed and extended by the model developer. In order to ensure the model's load balance and improve resource utilization in complex dataset task scheduling and processing, resource pools of different application types, including computation-intensive, storage-intensive, and application-intensive, are designed, and the scheduling system carries out scheduling of different resource pools according to the user's submission of tasks. Finally, the results of resource requests and task processing are returned to users or applications. On the other hand, the physical resource scheduling module of the second-level scheduling also adopts a multi-objective optimization approach to generate resource placement or migration strategies. In this way, the whole scheduling model fully considers the characteristics of multiple optimization objectives of users, application types and complex datasets themselves, and adopts a multi-objective optimization approach to improve the efficiency of task scheduling processing and resource utilization.

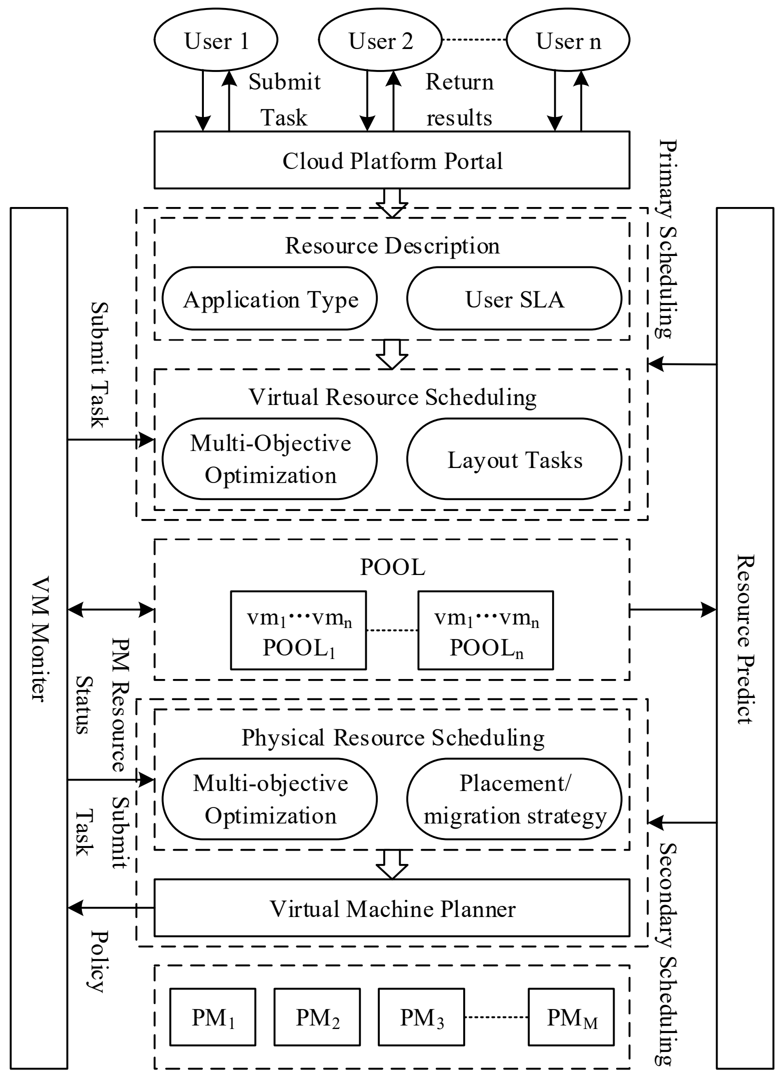
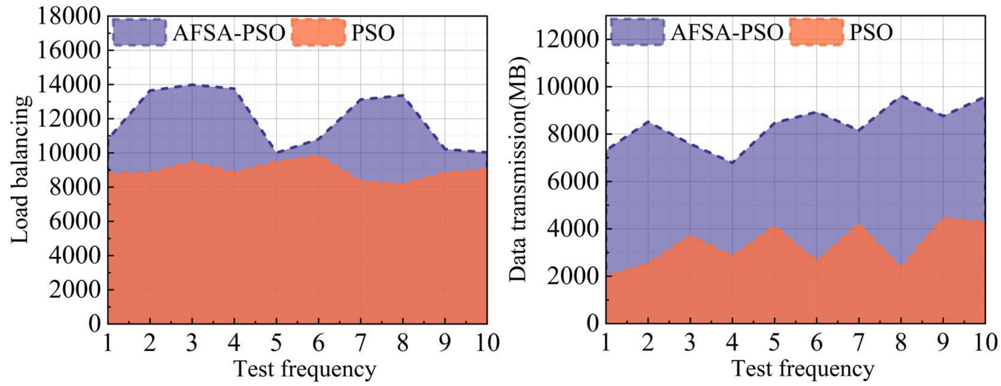


Fig. 1. Task scheduling and processing model architecture

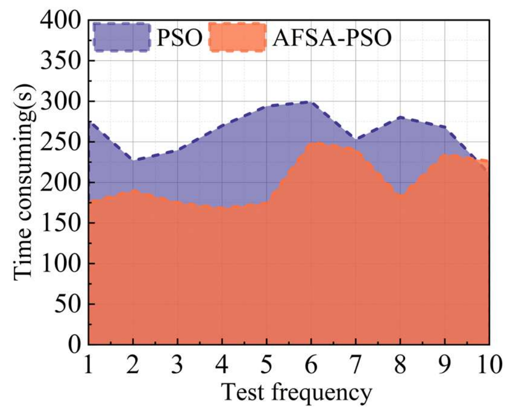
4. Task scheduling and processing model application analysis

4.1. Performance optimization analysis of hybrid particle swarm algorithm

In order to verify the effectiveness of the hybrid particle swarm algorithm improved based on the artificial fish swarm algorithm to improve the performance of the multi-objective optimization algorithm, this paper constructs a complex dataset based on an industrial process data. The experimental results obtained from the data processing and task scheduling multi-objective optimization algorithm based on particle swarm algorithm and the new method proposed in this paper are compared respectively. Under the same experimental environment, ten sets of experiments were done, and the results of task scheduling and processing performance analysis of complex datasets under different algorithms are shown in Fig. 2, with (a)-(c) representing the results of load balancing, data transmission and elapsed time analysis, respectively. It can be seen that the multi-objective optimization algorithm for task scheduling based on particle swarm algorithm is more stable and can ensure to come up with a better allocation scheme. However, the method based on particle swarm algorithm compared to the method based on hybrid particle swarm optimization algorithm proposed in this paper, the former requires many particles for iteration. In order to ensure that a better result can be obtained, the number of iterations is an unknown larger value, the calculation with the new method will be unstable, and in the current environment will sometimes be better than the particle swarm algorithm allocation scheme. But sometimes it will be worse, but this situation can be improved by taking other measures, for example, it can be done several times more program allocation test, and then carry out the selection of the optimal as the final program. Of course, such an improvement strategy will sacrifice some extra time, but the average time spent (201.49s) will still be much less than that of the particle swarm algorithm-based method (261.78s). From the experimental evaluation value comparison graph, we can see that the new hybrid particle swarm optimization method is better than the particle swarm algorithm, and the number of iterations of the data task scheduling algorithm based on the particle swarm algorithm to get this solution is much more than the number of iterations needed by the hybrid particle swarm optimization method proposed in this paper.



(a) Load balancing (b) Data transmission



(c) Time consuming

Fig. 2. The algorithm of particle swarm and the performance analysis of this paper

4.2. Simulation Scheduling and Processing Analysis of Complex Data Sets

4.2.1. Experimental simulation environment setup. In this paper, we use CloudSim 2.1.1 to establish a virtual scheduling simulation environment for experiments, and the development environment is Eclipse. The experimental machine is equipped with CPU Intel Core2 dual-core T5750@2.0GHz, 2GB of memory, 160 GB of hard disk, and operating system Windows 7 Ultimate. In the simulation, 1 complex data center is created, including 100 physical hosts, and then virtual machines are generated according to the ratio of 1:5. The tasks in the experiment are randomly generated with the size of 1000-4000 MI, and since this paper investigates the scheduling and processing of tasks in the environment of complex datasets, the tasks are sequentially bound to the virtual machines, and when the number of tasks, n, is less than 500, then the first n virtual machines are selected to corresponding to it. When the number of tasks n is 500, the tasks correspond to the virtual machines one by one. When the number of tasks n is greater than 500, the tasks are corresponded by the rotation method. Meanwhile, in order to verify the simulation task scheduling performance of the multi-objective optimization algorithm proposed in this paper, this paper extends the DatacenterBroker class in the CloudSim platform, and adds three methods, NSGA, BFH and MVRA, to implement the NSGA-based task scheduling model, the best-fit heuristic scheduling model BFH and the virtual resource allocation optimization model MVRA.

4.2.2. Experimental results and analysis. In the following, the total task completion time, load balancing coefficient and its task execution energy consumption of the scheduling algorithm proposed in this paper, the NSGA-based VM scheduling algorithm, the best-fit heuristic VM scheduling algorithm and the virtual resource allocation optimization algorithm are compared, respectively. The experiments set the population size of this paper's multi-objective optimization algorithm to 50, the number of evolutionary iterations to 100, the number of tasks to be gradually increased from 100 to 1000 with a step size of 100, and 10 experiments are conducted respectively, and each experiment eventually generates multiple Pareto solution sets. The final 10 Pareto optimal solutions of the algorithm when the number of tasks for the complex dataset is 500 are shown in Table 1, which shows that the multi-objective optimization algorithm proposed in this paper can obtain several different solutions by executing them at one time. Therefore, corresponding to the three objectives of this paper, the shortest task execution time, load balancing and the lowest energy consumption for task execution, can be selected according to different needs in practical use, and when these three objectives need to be coordinated, the solution with the smallest sum of the three objective values can be selected. Due to the contradictory nature between the objective functions, it is less likely to make them get the optimal solution at the same time, so in the practical application, the decision maker should choose the optimal scheme configuration from the set of Pareto solutions according to the optimization focus, while the simple single-objective algorithm can not meet such requirements, such as choosing the scheme 6 when the minimum of the task execution time (200.52s) is the main objective function. When load balancing (0.12) is the main objective function, option 7 is chosen. When the lowest energy consumption (398.00 KJ) is the main focus, option 5 is chosen. When balancing these three objectives and choosing a better configuration, option 7 can be chosen.

Table 1. Pareto optimal solution set analysis results

Scheme	Task execution time (s)	Load balancing factor	Energy consumption (KJ)
Solution 1	209.41	0.28	475.37
Solution 2	203.05	0.22	461.68
Solution 3	201.42	0.30	461.02
Solution 4	206.67	0.23	427.72
Solution 5	212.92	0.24	398.00
Solution 6	200.52	0.16	535.78
Solution 7	217.05	0.12	417.18
Solution 8	215.45	0.26	512.32
Solution 9	222.36	0.32	451.26
Solution 10	211.48	0.25	487.23

The execution time of a task reflects the execution efficiency of the scheduling algorithm. In this paper, we compare the simulation results of task scheduling and processing completion time of each model for a data center with 100 physical hosts and 500 virtual machines when the tasks are varied from 100 to 1000 as shown in Fig. 3. As the number of tasks grows, the matching of complex data resources with physical nodes becomes more complicated, and the task execution time of the four models grows and the difference between them increases. The task completion times of the MVRA and BFH models are always higher, reaching 406.70s and 381.74s for a task number of 1000, respectively. This is due to the fact that the MVRA model achieves energy saving by reducing the CPU frequency and supply voltage to achieve energy saving, thus prolonging the task execution time. The BFH model is scheduling the current complex dataset to a feasible physical node with the minimum amount of remaining resources, and multiple tasks jointly compete for fewer resources, which directly leads to an increase in the total task execution time. The task execution time of this paper's model (41.23s-239.82s) is lower than that of the NSGA model (58.95s-278.88s), which is due to the fact that the multi-objective optimization algorithm improves the generation of the initial population and solves the problem that the original algorithm misses the elite individuals, which makes its individuals more excellent, thus shortening the execution time of the task.

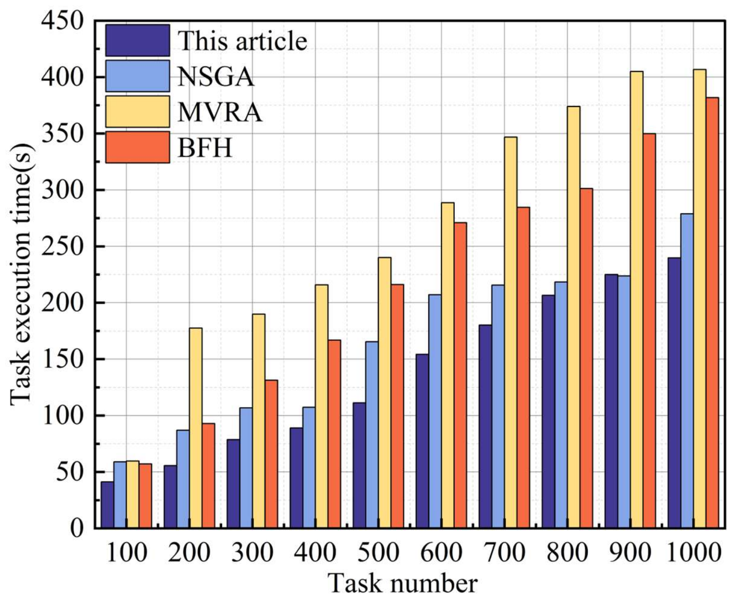


Fig. 3. The execution time is compared by the number of different tasks

Since load balancing reflects the stability of the system, the experiment selects the load balancing of the model as the goal to verify the performance of the model in this paper, and takes the CPU utilization as an example for the load balancing experiment, which compares the CPU utilization of 10 virtual machines after the execution of four scheduling models when the number of processing tasks for complex datasets is 500, and the simulation results of the load balancing performance are shown in Fig. 4. Since load balancing is taken as the goal, the CPU utilization of the four scheduling algorithms fluctuates between 0.350-0.491 when the tasks are deployed to the virtual machines for execution, and all of them have achieved some optimization in terms of load balancing. However, in terms of the degree of ups and downs, the scheduling model based on the multi-objective optimization algorithm proposed in this paper makes the CPU utilization of the VMs stay more within a certain value range (0.350-0.378). And the average CPU utilization (0.367) is much lower than the other models (0.406-0.414). This is due to the fact that the accuracy of the excellent solution of the algorithm is improved, thus reflecting that when different objective functions are selected, the optimal solution obtained by the task scheduling and processing model of complex dataset proposed in this paper is better than other models, and it is the first choice of task scheduling model under complex dataset processing.

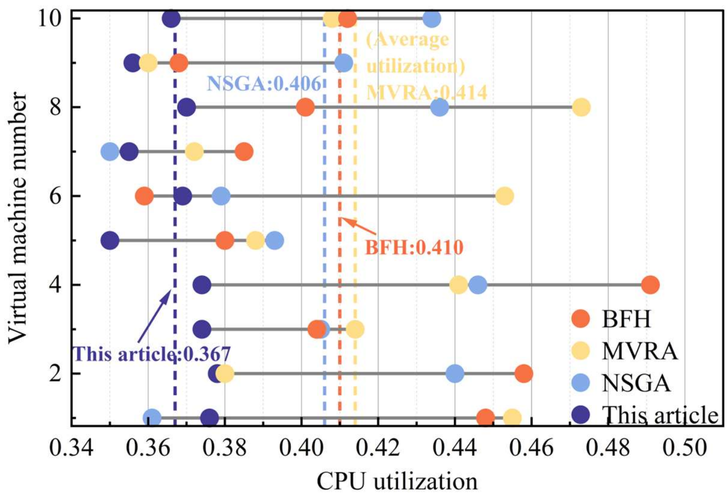


Fig. 4. The utilization of the CPU of 10 virtual machines

4.3. Example Application of a Multi-Objective Task Scheduling and Processing Model

In order to explore the practical application effect of the complex data set task scheduling and processing model constructed on the basis of the multi-objective optimization algorithm, this paper applies the model to a power grid scheduling system, and realizes the task scheduling and processing of various types of complex data sets in the power grid system through the model, so as to realize the optimized scheduling of the power grid and the improvement of the power generation effect of each module. The grid system in the empirical analysis is a grid-connected microgrid system, which contains a photovoltaic power generation system, a wind power generation system, a distributed energy system and a micro gas turbine. The original scheduling method in this grid and the optimized task scheduling model proposed in this paper are used to schedule the data in the grid system. In the grid system scheduling scheme obtained from both the original scheduling method and the optimized scheduling model in this paper, the use of batteries to assist in the regulation of the microgrid can effectively address the power demand at peak loads of the grid, thus effectively improving the stability and security of the grid. Batteries are charged when electricity prices are low and discharged when electricity prices are high, thus reducing economic and environmental costs. If solar and wind power generation cannot meet the load requirements of the power system, the microgrid can only provide power through the main grid or other distributed power sources. If the combined benefit of micro gas turbine and diesel power generation is greater than the power purchased from the main grid, gas and diesel power generation is preferred. When the microgrid itself generates power to satisfy the load and has surplus, then the microgrid provides additional power to the main grid. Before and after the improvement of the scheduling model to minimize the cost of environmental protection of the microgrid scheduling program in the photovoltaic output and wind turbine output comparison results are shown in Table 2. It can be seen that the peak PV and wind turbine outputs in the scheduling scheme achieved by applying the task scheduling and processing model constructed based on the multi-objective optimization algorithm in this paper (17.56kw and 7.95kw) are significantly larger than those obtained by the original scheduling method of the grid (9.66kw and 5.65kw). This indicates that the model proposed in this paper is able to realize effective processing and task scheduling of various types of complex power data in the grid system, thus improving the power generation and consumption cost of the grid system.

Table 2. Improvement of wind and photovoltaic output

Time(h)	PV output (kw)		Wind turbine output (kw)	
	Primary method	This method	Primary method	This method

1	0	0	1.23	0.82
2	0	0	1.23	0.8
3	0	0	1.36	0.05
4	0	0	0.7	0.3
5	0	0	0.26	1.66
6	0	0	1.17	1.41
7	0.98	0.75	0.8	1.97
8	1.79	1.23	1.36	2.04
9	4.26	5.02	0.55	1.13
10	6.61	5.16	0.12	1.45
11	7.86	9.66	1.29	1.64
12	8.8	9.83	5.42	7.85
13	9.59	15.25	5.65	7.95
14	9.66	17.56	3.62	3.95
15	7.44	13.55	1.32	1.4
16	3.35	10.23	0.8	0.96
17	1.74	5.06	1.31	1.25
18	0.16	2.69	1.24	1.91
19	0.14	0.58	1.33	1.48
20	0	0	1.12	1.51
21	0	0	1.48	1.3
22	0	0	1.2	1.87
23	0	0	1.33	0.89
24	0	0	1.21	1.16

5. Conclusion

This paper combines the improved artificial fish swarm algorithm and particle swarm algorithm to construct a multi-objective optimization algorithm, and based on this algorithm, constructs a task scheduling and processing model for complex data sets. The application effect of the model is explored through simulation experiments and empirical analysis experiments, and the results show that:

1) The hybrid particle swarm optimization algorithm established in this paper outperforms the particle swarm algorithm in terms of load balancing and processing time. The task execution time of the model in the simulation experiments (41.23s-239.82s) is lower than that of the NSGA model (58.95s-278.88s), which has better performance, and the average CPU utilization rate (0.367) is much lower than that of other models (0.406-0.414).

2) After being applied in the microgrid system, the model proposed in this paper is able to improve the output power of wind and PV in the grid system, and the peak output power of PV and wind turbine in the model-implemented scheduling scheme (17.56kw and 7.95kw) is significantly larger than that of the original scheduling scheme of this grid (9.66kw and 5.65kw).

3) The complex dataset task scheduling and processing model based on the multi-objective optimization algorithm proposed in this paper shows excellent performance in both simulation and practical application, and is able to achieve effective processing and scheduling of tasks in the dataset while taking into account multiple objectives such as load balancing and task loading time.

Funding

This work was supported by Shanghai Development and Reform Commission Service Industry Development Guidance Fund Project, Project Title: An Integrated Data Asset Governance Platform with Data Intelligence at its Core (Project Number: 06162021595).

References

[1] Mirjalili, S. Z., Mirjalili, S., Saremi, S., Faris, H., & Aljarah, I. (2018). Grasshopper optimization algorithm for multi-objective optimization problems. Applied Intelligence, 48, 805-820.

[2] Abdollahzadeh, B., & Gharehchopogh, F. S. (2022). A multi-objective optimization algorithm for feature selection problems. Engineering with Computers, 38(Suppl 3), 1845-1863.

[3] Deb, K., Sindhya, K., & Hakanen, J. (2016). Multi-objective optimization. In Decision sciences (pp. 161-200). CRC Press.

[4] Sharma, S., & Kumar, V. (2022). A comprehensive review on multi-objective optimization techniques: Past, present and future. Archives of Computational Methods in Engineering, 29(7), 5605-5633.

[5] Laumanns, M., & Ocenasek, J. (2002, September). Bayesian optimization algorithms for multi-objective optimization. In International Conference on Parallel Problem Solving from Nature (pp. 298-307). Berlin, Heidelberg: Springer Berlin Heidelberg.

[6] Cui, Y., Geng, Z., Zhu, Q., & Han, Y. (2017). Multi-objective optimization methods and application in energy saving. Energy, 125, 681-704.

[7] Prina, M. G., Cozzini, M., Garegnani, G., Manzolini, G., Moser, D., Oberegger, U. F., ... & Sparber, W. (2018). Multi-objective optimization algorithm coupled to EnergyPLAN software: The EPLANopt model. Energy, 149, 213-221.

[8] Afshari, H., Hare, W., & Tesfamariam, S. (2019). Constrained multi-objective optimization algorithms: Review and comparison with application in reinforced concrete structures. Applied Soft Computing, 83, 105631.

[9] Pereira, J. L. J., Oliver, G. A., Francisco, M. B., Cunha Jr, S. S., & Gomes, G. F. (2022). A review of multi-objective optimization: methods and algorithms in mechanical engineering problems. Archives of Computational Methods in Engineering, 29(4), 2285-2308.

[10] Soualhia, M., Khomh, F., & Tahar, S. (2017). Task scheduling in big data platforms: a systematic literature review. Journal of Systems and Software, 134, 170-189.

[11] Rjoub, G., Bentahar, J., & Wahab, O. A. (2020). BigTrustScheduling: Trust-aware big data task scheduling approach in cloud computing environments. Future Generation Computer Systems, 110, 1079-1097.

[12] Govindarajan, K., Kamburugamuve, S., Wickramasinghe, P., Abeykoon, V., & Fox, G. (2017, December). Task scheduling in big data-review, research challenges, and prospects. In 2017 Ninth International Conference on Advanced Computing (ICoAC) (pp. 165-173). IEEE.

[13] Liu, H., Liu, B., Yang, L. T., Lin, M., Deng, Y., Bilal, K., & Khan, S. U. (2017). Thermal-aware and DVFS-enabled big data task scheduling for data centers. IEEE Transactions on Big Data, 4(2), 177-190.

[14] Islam, T., & Hashem, M. M. A. (2018, December). Task scheduling for big data management in fog infrastructure. In 2018 21st International Conference of Computer and Information Technology (ICCIT) (pp. 1-6). IEEE.

[15] Yuan, H., Bi, J., Zhou, M., Liu, Q., & Ammari, A. C. (2020). Biobjective task scheduling for distributed green data centers. IEEE Transactions on Automation Science and Engineering, 18(2), 731-742.

[16] Zhu, Z., Zhang, J., Zhao, J., Cao, J., Zhao, D., Jia, G., & Meng, Q. (2019). A hardware and software task-scheduling framework based on CPU+ FPGA heterogeneous architecture in edge computing. IEEE Access,

7, 148975-148988.

[17] Khan, B., Goodman, D., Khan, S., Toms, W., Faraboschi, P., Luján, M., & Watson, I. (2015). Architectural support for task scheduling: hardware scheduling for dataflow on NUMA systems. The Journal of Supercomputing, 71, 2309-2338.

[18] Deng, Y., Chen, Z., Yao, X., Hassan, S., & Wu, J. (2019). Task scheduling for smart city applications based on multi-server mobile edge computing. IEEE Access, 7, 14410-14421.

[19] Lin, W., Wang, W., Wu, W., Pang, X., Liu, B., & Zhang, Y. (2018). A heuristic task scheduling algorithm based on server power efficiency model in cloud environments. Sustainable computing: informatics and systems, 20, 56-65.

[20] Javanmardi, S., Shojafar, M., Mohammadi, R., Nazari, A., Persico, V., & Pescape, A. (2021). FUPE: A security driven task scheduling approach for SDN-based IoT–Fog networks. Journal of information security and applications, 60, 102853.

[21] Liu, H., Pu, J., Yang, L. T., Lin, M., Yin, D., Guo, Y., & Chen, X. (2017). A holistic optimization framework for mobile cloud task scheduling. IEEE Transactions on Sustainable Computing, 4(2), 217-230.

[22] Pan, F., Xiong, J., Shen, Y., Wang, T., & Jiang, D. (2018, December). H-scheduler: Storage-aware task scheduling for heterogeneous-storage spark clusters. In 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS) (pp. 1-9). IEEE.

[23] Schäfer, D., Edinger, J., Eckrich, J., Breitbach, M., & Becker, C. (2018, March). Hybrid task scheduling for mobile devices in edge and cloud environments. In 2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops) (pp. 669-674). IEEE.

[24] Bu, T., Huang, Z., Zhang, K., Wang, Y., Song, H., Zhou, J., ... & Liu, S. (2024). Task scheduling in the internet of things: challenges, solutions, and future trends. Cluster Computing, 27(1), 1017-1046.

[25] Zhou, J. (2020). Real-time task scheduling and network device security for complex embedded systems based on deep learning networks. Microprocessors and Microsystems, 79, 103282.

[26] Ning Ding,Wenting Jiang,Yanheng Xi,Hongwei Li,Jun Su & Krishnamachar Prasad. (2025). Strategic layout planning of carbon emission monitoring devices in the open dustrial area based on an improved particle swarm optimization algorithm. Measurement116436-116436.

[27] Yeshen Lan,Chuchu Rao,Qike Cao,Bingyu Cao,Mingan Zhou,Bo Jin... & Wei Chen. (2024). An improved energy saving clustering method for IWSN based on Gaussian mutation adaptive artificial fish swarm algorithm. Scientific reports(1),27040.

[28] Chongyang Jiao & Qinglei Zhou. (2024). Path test data generation using adaptive simulated annealing particle swarm optimization. Soft Computing(17-18),9587-9607.

[29] Eric Coy,William Santo,Bonnie Jue,Helen Betts,Francisco Ramos Gomez & Stuart A. Gansky. (2024). Among Artificial Intelligence/Machine Learning Methods, Automated Gradient-Boosting Models Accurately Score Intraoral Plaque in Non-Standardized Images. Journal of the California Dental Association(1).