

Improved Particle Swarm Optimization Algorithm for Mobile Robot Path Planning

Zhiwei Yang

Xiangyang Auto Vocational Technical College, Xiangyang Hubei, China

Abstract: In solving the path planning problem of multi-path robots, an improved particle swarm optimization algorithm is proposed to address the drawbacks of premature convergence and low search accuracy of particle swarm optimization algorithm. Firstly, the improved Sine chaotic mapping is used to initialize the population, making it more evenly distributed in the search space and increasing population diversity. Then, the concept of quantum mechanics is introduced, which cancels the original particle movement speed and sets a new innovative parameter a instead. While reducing the parameters, the randomness of the particles is increased. Finally, the Levy flight strategy is used to improve the global search ability and convergence speed of the algorithm. The experimental results show that improving the particle swarm optimization algorithm for path planning enhances both local and global search capabilities. While minimizing algorithm complexity, it maximizes search accuracy and plans the shortest path that meets practical needs.

Keywords: Mobile Robot; Path Planning; Particle Swarm Optimization Algorithm; Sine Chaotic Mapping; Levy Flight Strategy.

1. Introduction

With the rapid development of modern intelligent unmanned technology and the gradual maturity of mobile robot technology, it has been widely applied in many industries such as military, transportation, logistics, and industry. In short, mobile robot path planning refers to finding a collision free path from the starting point to the target point when the mobile robot moves in an environment with obstacles without human intervention. Path planning technology is one of the key technologies in the field of intelligent robot research, which to some extent marks the level of robot intelligence. Many scholars at home and abroad have conducted research on path planning problems for intelligent robots. Traditional path planning algorithms can be divided into search-based path planning and sampling-based path planning algorithms. Search based algorithms such as BFS, Astar algorithm[1], Dijkstra algorithm, etc. have completeness and optimality, but they also have the problem of falling into local optimal traps when facing complex obstacle environments, making it difficult to generate optimal paths. Sampling based algorithms such as PRM and RRT have probabilistic completeness, but there is also a problem of generating routes that are not the shortest or optimal.

In recent years, with the development and application of swarm intelligence algorithms, there have been more solutions to the path planning problem of mobile robots. Compared with traditional path planning methods, intelligent algorithms make up for many shortcomings of traditional methods and perform well in solving complex environmental path planning problems. PSO was proposed by Kennedy and Eberhart in 1995[2]. In the algorithm, candidate solutions are treated as a group of particles, each with its own position and velocity, and the optimal solution found is recorded. Particles update their position and velocity by following the best particles in the group, ultimately finding the global optimal solution. PSO has a fast convergence speed and strong generality, but it also has the characteristics of premature convergence, low search accuracy, and low efficiency in later

iterations. Feng et al[3]. used nonlinear inertial weights to balance exploration and exploitation, and improved the convergence speed and resolution quality of the algorithm by dynamically adjusting the inertial weights. Tian et al[4]. improved the acceleration coefficient of standard PSO to make the individual learning factor and social learning factor asymmetric and change over time, thus better balancing exploration and exploitation, improving the convergence speed and quality of the algorithm. Although the improved PSO mentioned above has shown good performance in solving path planning problems, there are also many parameters that need to be set, including c_1 , c_2 , ω , etc., which are not conducive to training optimization and the algorithm particles lack randomness, making it easy to fall into local optimal solutions.

This article proposes an improved particle swarm optimization algorithm, which introduces a re initialization mechanism based on global information feedback to address the possible stagnation phenomenon in PSO algorithm. On this basis, the concept of quantum dynamic kinematics is introduced, and the original concept of particle velocity as the direction of movement is eliminated. The concept of potential well is adopted, so that the next motion of particles has no relationship with their previous motion, greatly increasing the randomness of particle motion. Finally, path planning simulation experiments were conducted in different grid environments, and the experimental results proved the superiority of our algorithm in path planning problems.

2. Particle Swarm Optimization

The PSO algorithm first initializes a group of particles in the solvable space, where each particle represents a potential optimal solution to the extreme value optimization problem. The characteristics of the particle are represented by three indicators: position, velocity, and fitness value. The fitness value is calculated by the fitness function, and the quality of its value indicates the superiority or inferiority of the particle. During each iteration, particles update their velocity and

position through individual extremum and global extremum,

$$v_{in}(t+1) = \omega \cdot v_{in}(t) + c_1 \text{rand}() [pbest(t) - x_{in}(t)] + c_2 \text{rand}() [gbest(t) - x_{in}(t)] \quad (1)$$

$$x_{in}(t+1) = x_{in}(t) + v_{in}(t+1) \quad (2)$$

In the formula, v and x represent the velocity and position of particles; ω is the inertia weight; r_1 and r_2 are random numbers distributed between $[0,1]$; c_1 and c_2 are acceleration factors; $pbest(t)$ represents the current optimal particle position, while $gbest(t)$ represents the global optimal particle position; $d=1,2,\dots,D$; $i=1,2,\dots,n$

3. Improved Particle Swarm Optimizations

The main reasons for the poor search ability of standard PSO and the tendency to fall into local optima are: (1) there are many parameters that need to be set, such as acceleration weight, inertia weight, etc., which are not conducive to finding the optimal parameters of the model to be optimized. (2) The lack of randomness in particle position changes can easily lead to the trap of local optima. In order to solve the above problems, this article improves the PSO algorithm, and the specific measures are as follows.

3.1. Improved Sine Chaotic Mapping

The initialization of the basic population is a random

with the following update formula:

distribution throughout the entire space, which has strong randomness and uneven distribution, which can cause problems such as insufficient population diversity and low search efficiency. This article uses chaos mapping mechanism to increase the diversity of the population and improve the performance of the algorithm. Its nonlinear characteristics and periodic properties enable it to generate more complex and random sequences, which helps to enhance the diversity of the population and avoid falling into the situation of local optimal solutions. The sequence generated by the iteration of traditional one-dimensional Sine chaotic mapping is unevenly distributed in phase space, and its parameter space in chaotic state is narrow. Therefore, by introducing an improved Sine chaotic mapping [5], the distribution is more uniform and the chaotic effect is better. The expression is as follows:

$$\begin{cases} d_{i+1} = \sin(\mu\pi d_i) \\ e_{i+1} = \sin(\mu\pi e_i) \\ w_{i+1} = d_{i+1} + e_{i+1} \bmod 1 \end{cases} \quad (3)$$

Among them, w_{i+1} , d_{i+1} and e_{i+1} are iterative sequence values, i is a non negative integer, $w_0 \in (0,1)$ 0; μ is a system parameter, $\mu=1500$.

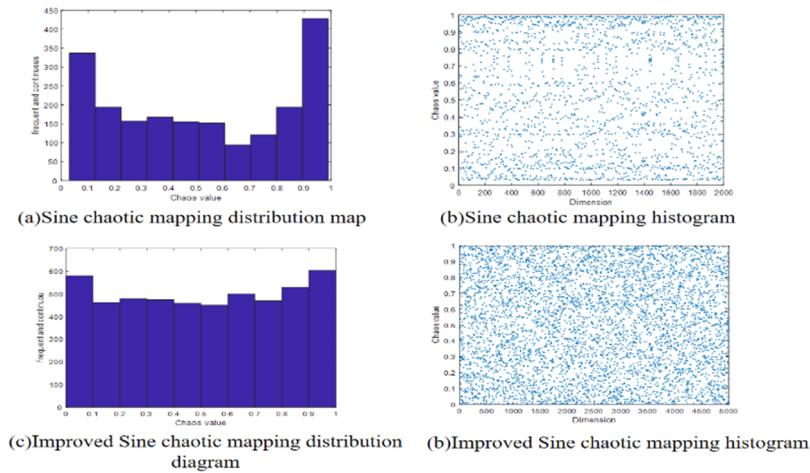


Fig 1. Comparison of distribution of improved Sine and Sine chaotic maps

From Figure (1), it can be observed that both the improved Sine and Sine mapping distributions are between $[0,1]$, and the improved Sine chaotic mapping distribution is more uniform. Its chaos can replace random initialization and make the population more evenly distributed in the search space.

3.2. Optimizing PSO Algorithm based on Quantum Computing

Although particle swarm optimization algorithm is easy to use, it requires a large number of parameters to be set, which is not conducive to finding the optimal parameters. Moreover, the changes in particle positions lack randomness and are prone to falling into the trap of local optima. The derivation process of PSO algorithm based on quantum computing optimization[6] is based on quantum dynamics formula, and the specific position of particles is determined through Monte

Carlo inverse method. Only one parameter needs to be adjusted to update the equation, making the implementation and parameter selection of the algorithm easier. At the same time, in order to increase the randomness of particle position, the movement direction attribute of particles is eliminated.

Assuming the population size is M and the index i represents the i -th particle, the average value of the best position of the particles in the current population, M_{best} , is calculated as follows:

$$M_{best} = \frac{1}{M} \sum_{i=1}^M Pbest_i \quad (4)$$

Among them, $pbest_i$ represents the current optimal position of the i -th particle.

The potential field formula for the i -th particle is as follows:

$$P_i = \lambda \cdot pbest_i + (1 - \lambda) \cdot gbest \quad (5)$$

Among them, λ is a random number with a value range of [0,1]; $gbest_i$ represents the globally optimal particle.

The Monte Carlo method is used to update the particle position formula as follows:

$$x_i = P_i \pm \alpha \left| Mbest - x_i \right| \ln\left(\frac{1}{\mu}\right) \quad (6)$$

Among them, the value range of μ is [0,1], and α is the innovation parameter with a value less than 1, which performs contraction and expansion operations on the position of the particles.

3.3. Levy Flight Strategy

The PSO algorithm based on quantum computing optimization only targets non optimal particles during each evolution, and cannot solve the problem of optimal particle evolution. To improve the above shortcomings, the Levy flight strategy[7] is introduced to update the global optimal particles. The specific idea of the Levy flight particle swarm algorithm is that when the particle swarm algorithm performs each iteration, all particles will continue to perform a Levy flight, utilizing the characteristics of Levy flight's small-scale search and long-distance migration to increase the search range, avoid falling into local optima, and enhance the convergence accuracy of the particle swarm algorithm.

The formula for Levy's flight strategy step size is shown in equation (7):

$$\text{Levy}(\beta) = \frac{\mu}{|\nu|^{-\beta}} \quad (7)$$

Among them, $\text{Levy}(\beta)$ follows a Levy distribution with parameter β , $0 < \beta < 2$, and μ follows an $N(0,1)$

distribution.

The formula for updating the position using Levy flight is:

$$x_i^{t+1} = x_i^t + \alpha L(\beta)(g_{best}(i) - x_i^t) \quad (8)$$

In the formula, x_i^{t+1} and x_i^t are the positions of the t+1st and t th generations of x_i , respectively; α is the scaling factor, taken as 0.01; $g_{best}(i)$ is the optimal position of the current population; L is the step size factor that follows the Levy flight strategy, as shown in the following equation:

$$L \sim \frac{\lambda \Gamma(\beta) \sin\left(\frac{\pi\lambda}{2}\right)}{\pi \cdot s^{1+\beta}} \quad (9)$$

In the equation, $\beta=1.5$; $\Gamma(\beta)$ is the gamma function. Due to the complexity of the Levy distribution, Monte Carlo algorithm is often used for simulation, and s is calculated using equation (10):

$$s = \frac{\mu}{|\nu|^{\frac{1}{\beta}}} \quad (10)$$

In the formula, μ follows an $N(0, \sigma^2)$ distribution; ν follows $N(0, 1)$; σ can be calculated by equation (11):

$$\sigma = \frac{\Gamma(1+\beta) \sin\left(\frac{\pi\beta}{2}\right)}{\beta \Gamma\left(\frac{1+\beta}{2}\right) 2^{\frac{\beta-1}{2}}} \quad (11)$$

3.4. Construction of MPSO Algorithm Path Planning Model

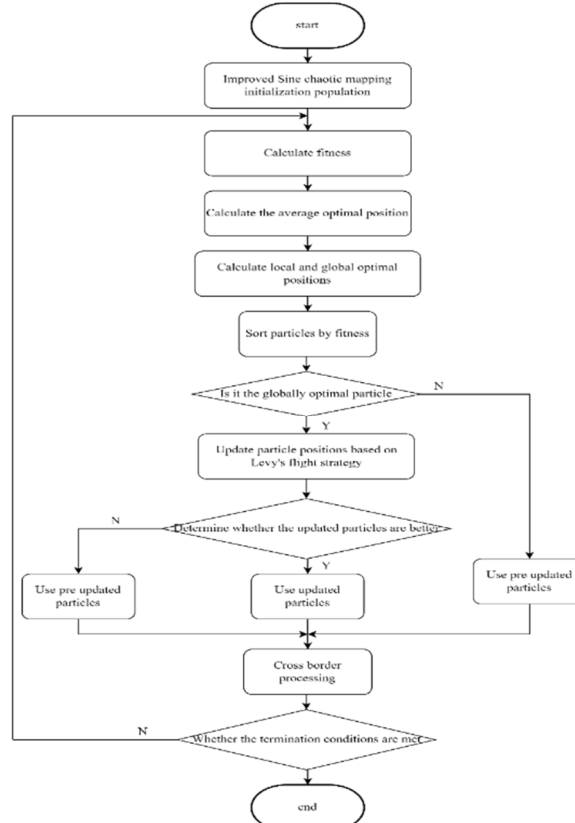
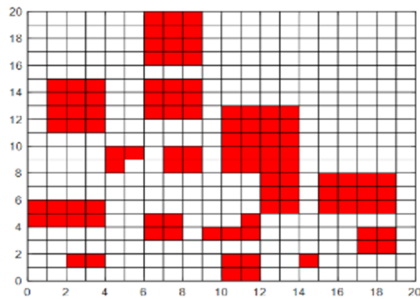


Fig 2. MPSO algorithm path planning flowchart

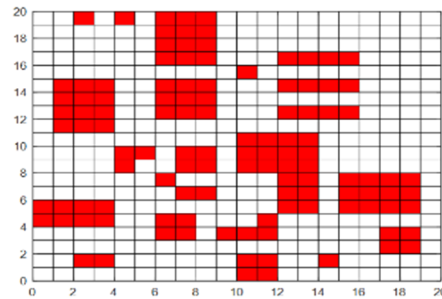
The current problems in path planning mainly include difficulty in environmental modeling, slow convergence speed of algorithms, and susceptibility to getting stuck in local optimal solutions [8]. The first step is to initialize the population and various parameters using an improved Sine chaotic map; The second step is to calculate the average of the optimal positions of particles in the current population; Step three, calculate the potential field of the i -th particle; Step 4: Use Monte Carlo method to update particle positions and incorporate Levy flight strategy for optimization; Step five, calculate the particle position fitness and compare it with the best position fitness of individuals and groups to update the optimal position. Step six, determine whether the iteration termination condition is met. If not, return to step two. The MPSO algorithm path planning flowchart is shown in Figure 2:

4. Simulation Experiments and Result Analysis

The experimental platform selected is MATLAB 2022b,



(a)Simple map model



(b)Complex map model

Fig 3. Grid maps

4.2. Analysis of Path Planning Experiment Results

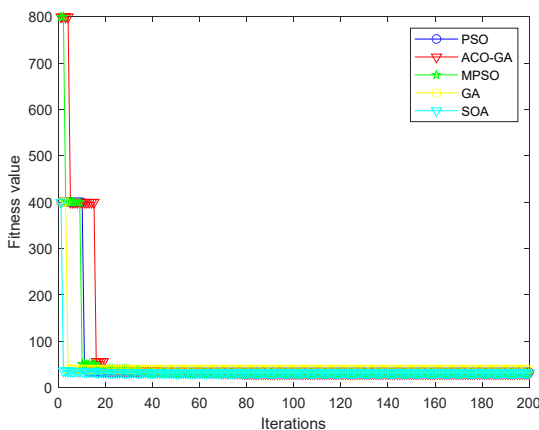


Fig 4. Comparison and optimization of iterative curves using multiple intelligent algorithms

with a computer CPU model of i7-13700K and a running memory of 32GB. In order to verify the effectiveness of the algorithm proposed in this paper, path planning for mobile robots was conducted in two different grid environments, and simulation comparative experiments were conducted using the algorithm proposed in this paper with PSO, ACO_GA, SOA, and GA algorithms.

4.1. Environmental Map

This article maps the environment to a grid map[9], which equates the environment to a grid matrix composed of cells of the same size. Build two maps of different complexities using MATLAB, encode each grid with real numbers, fill the grids corresponding to actual obstacles with 1, represent them in red on the map, fill feasible areas with 0, and represent them in white. The side length of each grid is 1, where the point with a real number code of 0 represents the starting point of the robot, and the point with a real number code of 399 represents the target point[10]. The grid map is shown in Figure 3.

In order to verify the performance of the proposed MPSO algorithm, during the simulation comparison experiment, the algorithm proposed in this paper was used in the iterative process of robot path planning with PSO, ACO_GA, SOA, and GA[11]. The fitness iteration process of each algorithm was compared as shown in Figure 4. From the figure, it can be seen that the MPSO algorithm has a faster number of iterations. Compared to a single intelligent algorithm, it is slightly slower due to the integration of multiple improvement strategies, which increases complexity. However, for the fusion of multiple intelligent algorithms, it is significantly faster.

Path planning simulation experiments were conducted 20 times on the grid maps shown in Figure 3 (a) and (b), with an initial particle count of 50 and 200 iterations. Each algorithm was compared under the same parameters, and the shortest path, average time, and average path were recorded.

The path planning results in Figure 3 (a) are shown in Table 1, and the planned path is shown in Figure 5.

Table 1. Comparison of path planning results

Algorithm	Shortest Path /cm	Average Path /cm	Average Time/s
MPSO	28.6.025	28.5827	1.3476
PSO	30.05	31.668	0.18252
GA	32.054	34.9618	2.7102
ACO_GA	27.4663	28.4327	2.1453
SOA	29.3832	31.7202	0.66871

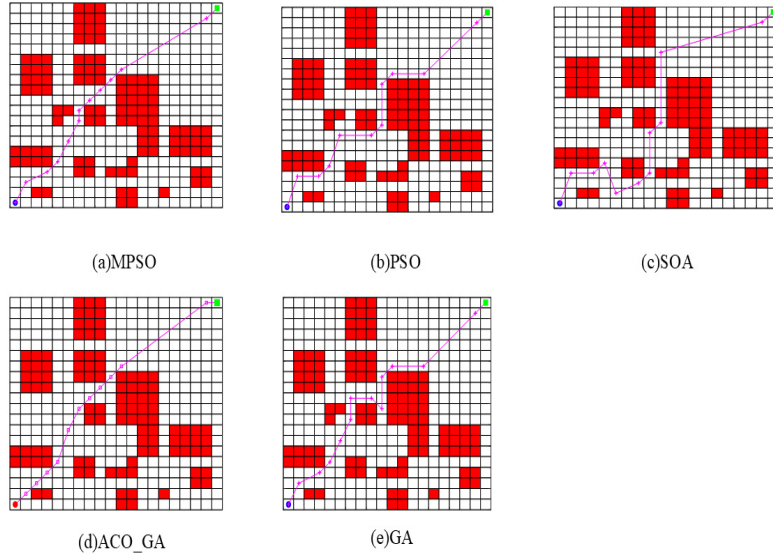


Fig 5. Simple map model with multiple algorithm path planning results

From Table 1, it can be observed that each algorithm underwent 20 rounds of planning, and the average time recorded the efficiency of each algorithm. Among them, MPSO algorithm has a slower convergence speed and longer time compared to PSO algorithm and SOA algorithm, but the planned shortest path and average path are much higher than these two algorithms, demonstrating better search accuracy and optimization ability. Compared to MPSO algorithm, AOC_GA algorithm has better solution quality, but its optimization efficiency is lower than MPSO algorithm[12]. Therefore, the MPSO algorithm proposed in this article not only plans shorter paths, but also has high efficiency and good robustness.

Figure 5 shows the optimal routes planned by various algorithms in the same grid map. Combining Figures 4 and 5,

it can be clearly observed that using standard swarm intelligence algorithms for path planning is prone to getting stuck in local optima, particularly evident in the PSO algorithm; The simple fusion of two intelligent algorithms for path planning can improve the problem of weak local search, but the search takes a long time, occupies a large amount of memory, and has weak real-time performance[13]. This article introduces multiple optimization strategies to optimize standard intelligent algorithms, aiming to minimize algorithm complexity while maximizing search accuracy and planning the shortest path that meets practical needs.

Table 2 and Figure 6 show experiments conducted in the environment of Figure 1 (b), which is more complex and has more obstacles.

Table 2. Comparison of path planning results

Algorithm	Shortest Path /cm	Average Path /cm	Average Time/s
MPSO	30.8743	31.629	3.0223
PSO	37.1263	42.2978	0.68269
GA	30.6143	36.2148	3.5334
ACO_GA	29.4203	30.2661	3.1433
SOA	36.6224	42.476	1.1332

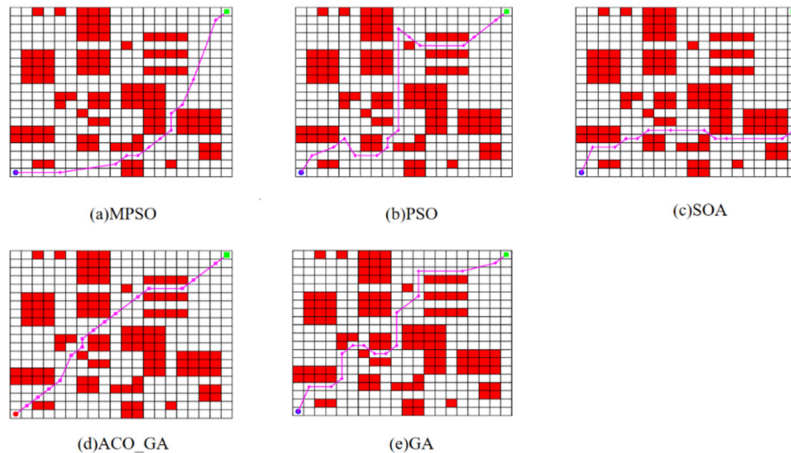


Fig 6. Multiple algorithm path planning results for complex map models

When the environmental map becomes more complex, the advantages of the algorithm in this article can be better reflected. In Table 2, the performance of the improved particle swarm algorithm is significantly better than other algorithms, and the performance of the algorithm has been improved. The length of the optimal path planned by the MPSO algorithm is smaller than the path length planned by the basic PSO algorithm, which solves the problem of PSO algorithm easily getting stuck in local optima. Figure 6 shows the optimal route planned in a complex grid map. It can be seen that the improved algorithm in this paper has always maintained excellent planning path capability and better algorithm stability.

5. Conclusion

The MPSO algorithm proposed in this article first initializes the population by introducing an improved Sine chaotic map to enhance its diversity. Then, the quantum concept is introduced to improve the global search ability and convergence of the traditional PSO algorithm by adding two probability density functions. Finally, the Levy flight strategy is introduced to update the global optimal particles, solving the evolution problem of the optimal particles and enhancing the convergence accuracy of the particle swarm algorithm. The algorithm proposed in this article was applied to robot path planning in two different grid maps, namely PSO, AOC_GA, SOA, and GA. The experimental results showed that the MPSO algorithm performed well in shortest path, average path, and convergence speed, outperforming a single intelligent algorithm in accuracy and slightly inferior to the fusion of multiple intelligent algorithms in searching for the best path, demonstrating good performance and providing a feasible solution for mobile robots to plan effective shortest paths.

References

- [1] Dai Xiaolin, Shuai Long, Zhiwen Zhang, et al. "Mobile robot path planning based on ant colony algorithm with A* heuristic method[J]." *Frontiers in neurorobotics*, 2019, 16(4):13-15.
- [2] Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks* (Vol. 4, pp. 1942-1948). IEEE.
- [3] Feng, Y., Teng, G. F., Wang, A. X., & Yao, Y. M. (2007, August). Chaotic inertia weight in particle swarm optimization. In *Second International Conference on Innovative Computing, Informatio and Control (ICICIC 2007)* (pp. 475-475). IEEE.
- [4] Tian, D., Zhao, X., & Shi, Y. (2019). Chaotic particle swarm optimization with sigmoid-based acceleration coefficients for numerical function optimization. *Swarm and Evolutionary Computation*, 51, 100552.
- [5] Yuanzhou Z, Lei L ,Long Q , et al.Sine-SSA-BP Ship Trajectory Prediction Based on Chaotic Mapping Improved Sparrow Search Algorithm[J].*Sensors*,2023,23(2):704-704.
- [6] Zhang, J., Xiao, M., Gao, L., & Pan, Q. (2019). Queuing search algorithm: A novel metaheuristic algorithm for solving engineering optimization problems. *Applied Mathematical Modelling*, 63, 464-490.
- [7] Kamaruzaman, Anis Farhan, et al. "Levy flight algorithm for optimization problems-a literature review." *Applied mechanics and materials* 421 (2013): 496-501.
- [8] Gao, Fang, Qiang Zhao, and Guixian Li. "Modelling and control of vehicle magnetorheological seats based on bond graph." *China Sciencepaper/Zhongguo Keji Lunwen* 7.11 (2012): 818-821.
- [9] Dogar F R, Karagiannis T, Ballani H, et al. Decentralized task-aware scheduling for data center networks[J]. *ACM SIGCOMM Computer Communication Review* 2014,44(4): 431-442.
- [10] Wang L, Wang W. Fair coflow scheduling without prior knowledge[C]. *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, 2018:22-32.
- [11] Qiu Z, Stein C, Zhong Y. Minimizing the total weighted completion time of coflows in datacenter networks[C] // *27th ACM symposium on Parallelism in Algorithms and Architectures*, 2015:294-303.
- [12] XIN-SHE Yang, SUASH DEB. Multiobjective cuckoo search for design optimization[J]. *Computers and Operations Research*, 2013,40(6).
- [13] Hüseyin Haklı,, and Harun Uğuz. "A novel particle swarm optimization algorithm with Levy flight." *Applied Soft Computing* 23 (2014): 333-345.