Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org

Volume 25, Issues 3, pp. 2021–2033, DOI 10.12694/scpe.v25i3.2698

ECONOMIC DISPATCH OF MULTI REGIONAL POWER SYSTEMS BASED ON CMOPSO ALGORITHM

JINHUA GUO*

Abstract. With the progress of society and the aggravation of environmental pollution, the economic dispatch of the power system is developing towards multiple environmental and economic goals. To improve energy utilization efficiency, this study innovatively proposes a multi-objective particle swarm optimization algorithm based on competitive learning, and uses this algorithm to solve multi regional environmental and economic scheduling problems. In addition, the study solves static and dynamic economic (S-DE) scheduling problems in multiple regions through improved competitive group optimization algorithms. The research results show that under different testing systems, the average distribution uniformity indicators of the research algorithm built on competitive learning are 0.8058 and 0.8457, and the average anti generation distance is 67.6316 and 1664.0978. The improved competitive group optimization algorithm solves the maximum, minimum, and average fuel costs for static economic scheduling in multiple regions, which are 656.2243 \$/h, 655.8592 \$/h, and 655.9866 \$/h, respectively. Thus, the designed algorithm can effectively solve economic scheduling problems, which is of great significance for resource integration, saving power generation costs, and reducing pollution emissions.

Key words: Multi region; Economic dispatch; CMOPSO; ImCSO; Constraint condition

1. Introduction.

Economic dispatch, as one of the methods to promote high-quality development of the power system, can ensure the normal operation of the power system. Economic dispatch requires the optimal scheduling of power generation units with the best fuel cost, without violating various operational constraints [1]. The power dispatch system generally involves multiple regions, and it is necessary to dispatch electricity reasonably between regions to fully utilize the system's resources [2]. The early solutions to the problem of multi regional economic dispatch were mathematical methods, such as dynamic programming, gradient algorithms, etc. These mathematical methods have good results in economic scheduling problems with fewer constraints, but they are difficult to solve problems involving complex factors [3]. With the development of technology, more and more researchers are applying heuristic optimization algorithms to economic scheduling problems due to their ability to effectively solve complex problems. However, these studies also have certain shortcomings, such as the scheduling problems involved being relatively single and the constraints considered being incomplete [4]. Based on these issues, this study innovatively proposes the use of Competitive Multiple Objective Particle Swarm Optimization (CMOPSO) based on competitive learning to solve multi regional environmental and economic scheduling problems. The study also adopts an Improved Competitive Swarm Optimizer (ImCSO) algorithm to address S-DE scheduling issues in multiple regions. This study aims to improve energy utilization efficiency and achieve optimal scheduling of power generation units through optimal fuel costs. The study is divided into four parts. The first part is an overview of research related to economic dispatch in the power system. The second part is the design of the algorithm used to solve economic scheduling problems. The third part is the analysis of the results of the research method. The fourth part is the conclusion.

2. Related Works.

Economic dispatch is an important way to promote the high-quality development of the power system and occupies an essential position in the stable operation of the power system. At present, there are many studies on the economic scheduling of power systems. Goudarzi A and other researchers have proposed an intelligent sequence algorithm for synchronous scheduling of electricity and heat. This algorithm includes both an optimization algorithm that combines enthusiasm assistance and mathematics, as well as an improved particle swarm optimization algorithm. In addition, the study also designed constraint management,

*College of Economics and Trade, Henan Polytechnic Institute, China, Nan yang, 473000, China (Corresponding author, lwgjh2023@163.com)

indicating that the proposed method is significantly superior in performance to conventional methods [6]. Lyu C and other scholars proposed a new degradation cost model for the cyclic degradation problem in microgrid economic scheduling, and used Wasserstein fuzzy sets to describe uncertainty. In addition, this study also expresses real-time microgrids through distributed robust optimization problems. This method has good performance and can effectively solve the cyclic degradation problem during microgrid economic scheduling [7]. Wang X and other experts designed a sparse polynomial chaotic expansion proxy model based on data-driven to solve the stochastic economic scheduling problem of wind power uncertainty. This model can provide information such as mean and variance in stochastic economic scheduling solutions. This method has high accuracy and efficiency in solving stochastic economic scheduling problems [8]. Marco et al. analyzed existing power energy systems to reduce carbon dioxide emissions and proposed a modeling method for power planning tools. This method utilized computational strategies and linear programming optimization methods. This method had strong analytical ability and can effectively plan the power system [9].

Experts such as Lev K have designed a coordination mechanism between tie line power planning and regional power grid economic dispatch for cross regional power grid economic dispatch under various uncertain power sources and loads. It adopts a quality-of-service approach to analyze the service attributes in power dispatch. In addition, the study also constructed a model free hierarchical optimization method based on learning technology, and used reinforcement learning algorithms to effectively solve the economic dispatch problem of cross regional power grids [10]. Scholars such as Xu D have designed a maximum minimum two-layer optimization model and a two-stage robust optimization model to solve the problem of unpredictability during power grid backup, and used column constraint generation algorithm to solve it. This model can effectively solve the problem of unpredictability during power grid backup, and has obvious advantages in random scenarios [11]. Wang S et al. designed a multi-agent power grid control scheme to meet the requirements of the system and computing platform for autonomous control of the power grid. This scheme is data-driven and adopts deep reinforcement learning, which can be learned from scratch. This method can meet the requirements of power grid autonomous control for systems and computing platforms [12]. Shaheen A M and other experts have designed a multi-objective manta ray foraging algorithm to minimize emissions from DC AC hybrid power grids. This algorithm imitates the feeding process of manta rays and adopts fuzzy decision-making technology. It has been compared and tested on multiple systems, and the results show that the robustness of this method is significantly better than other comparison algorithms [13].

In summary, there is currently a wealth of research on economic dispatch in the power system, and the methods used are also diverse. However, these studies also have certain shortcomings, such as the scheduling problems involved being relatively single and the constraints considered being incomplete. Based on these issues, this study innovatively proposes the use of CMOPSO algorithm to solve multi regional environmental and economic scheduling problems, and uses ImCSO algorithm to solve multi regional S-DE scheduling problems.

3. Design of Economic Dispatching Method for MRPS Based on CMOPSO and ImCSO Algorithms. This study uses an improved competitive group optimization algorithm to solve the S-DE dispatch problems of multi region power systems. It optimized the competitive group optimization algorithm through ranking pairing learning and differential evolution, and designed specific steps for the algorithm in multi region S-DE scheduling problems. A MOPSO based on competitive learning was used to solve the problem of multi regional environmental and economic scheduling, and specific steps were designed for this algorithm to solve the problem.

3.1. Design of S-DE scheduling methods for multiple regions based on ImCSO algorithm. The economic dispatch of multi-regional power systems (MRPS) is mainly divided into Multi Area Static Economic Dispatch (MASED), Multi Area Dynamic Economic Dispatch (MADED), and Multi Area Environment Economic Dispatch (MAEED) [14]. The core goal of MASED is to minimize the combustion cost of the power system. To solve the MASED problem, an ImCSO algorithm was adopted for improvement through ranking pairing learning and differential evolution. The learning process of ranking paired learning is shown in Fig 3.1 [15].

In Fig. 3.1, ranking pairing learning requires sorting all particles first. The sorting is based on the fitness information of the particles, and then the particles are divided into winner and loser groups according to the sorting results. The particles of the loser group need to learn from the particles of the winner group. The

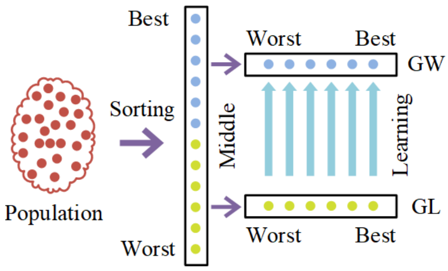


Fig. 3.1: The learning process of ranking paired learning

operation of particle sorting and grouping is equation (3.1).

$$\begin{cases} f\left(X_1^{t,W}\right) \leq f\left(X_2^{t,W}\right) \leq \cdots \leq f\left(X_{ps/2}^{t,W}\right) \leq f\left(X_1^{t,L}\right) \leq f\left(X_2^{t,L}\right) \leq \cdots \leq f\left(X_{ps/2}^{t,L}\right) \\ GW = \left\{X_1^{t,W}, X_2^{t,W}, \cdots, X_{ps/2}^{t,W}\right\} \\ GL = \left\{X_1^{t,L}, X_2^{t,L}, \cdots, X_{ps/2}^{t,L}\right\} \end{cases} \tag{3.1}$$

In equation (3.1), t means the iteration numbers. W are the particles in the winner group population. L is the particle in the loser group population. GW represents the winner group. GL represents the loser group. $X_i^t = \left[X_{i,1}^t, X_{i,1}^t, \cdots, X_{i,D}^t\right]$ represents the position item. Where i is the serial number. D is the dimension of the optimization problem. ps represents the population size. The updates of the velocity and position terms of particles in GL are shown in equation (3.2).

$$\begin{cases} V_i^{t+1,L} = R_1 \times V_i^{t,L} + R_2 \times \left(X_i^{t,W} - X_i^{t,L}\right) + R_3 \times \varphi \times \left(X_{i,center}^t - X_i^{t,L}\right) \\ X_i^{t+1,L} = X_i^{t,L} + V_i^{t+1,L} \end{cases} \tag{3.2}$$

In equation (3.2), $V_i^t = \left[v_{i,1}^t, v_{i,2}^t, \cdots, v_{i,D}^t\right]$ represents the velocity term. R_1, R_2 and R_3 represent a set of random numbers with a distribution range of $[0, 1]$. φ is a social factor that can control $\bar{X}^t$. $\bar{X}^t$ represents the average position of the entire population in the t-tph iteration. $X_{i,center}^t$ is the center position, and its calculation is equation (3.3).

$$X_{i,center}^t = a \times \bar{X}_{GW}^t + (1-a) \times \bar{X}_{GL}^t \tag{3.3}$$

In equation (3.3), a represents a random real number within $[0, 1]$. $\bar{X}_{GW}^t$ represents the average position of particles in the GW population. $\bar{X}_{GL}^t$ is the average position of particles in the GL. The update of winner particles in the GW is mainly achieved through differential evolution strategy (DES). The specific update steps include mutation, crossover, and selection. The mutation step requires the generation of mutated individuals, and the specific process is equation (3.4).

$$Z_i^{t,W} = X_{r1}^{t,W} + F \times \left(X_{r2}^{t,W} - X_{r3}^{t,W}\right) \tag{3.4}$$

In equation (3.4), $r1$, $r2$, and $r3$ are random positive numbers that are different from each other, with a value range of $\{1, 2, \cdots, ps\}$. $Z_i^{t,W}$ represents the mutant individual. F represents the variation factor. The purpose of the cross step is to generate experimental individuals, and the specific operation is equation (3.5).

$$u_{i,j}^{t,W} = \begin{cases} Z_{i,j}^{t,W}, & if \quad rand_j \leq CR \quad or \quad j = j_{rand} \\ X_{i,j}^{t,W}, & otherwise \end{cases} \tag{3.5}$$

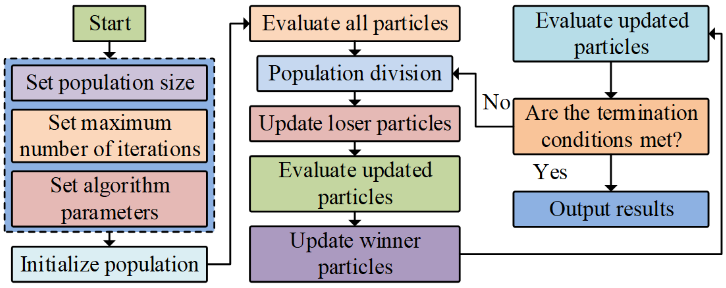


Fig. 3.2: The process of ImCSO algorithm

In equation (3.5), $U_i^{t,W}$ represents the experimental individual, and its value range is $\left[u_{i,1}^{t,W}, u_{i,2}^{t,W}, \cdots, u_{i,D}^{t,W}\right]$. CR represents the crossover factor. $rand_j$ is the dimension, with a value range of $[1, D]$. $rand_j$ represents a random real number within the range of $[0, 1]$. j_{rand} is a random integer in $[1, D]$. The selection step is to choose individuals between $X_i^{t,W}$ and $U_i^{t,W}$ to enter the next generation, grounded on fitness. The specific process of this step is equation (3.6).

$$X_{i,j}^{t+1,W} = \begin{cases} U_i^{t,W}, & if \quad f\left(U_i^{t,W}\right) < f\left(X_i^{t,W}\right) \\ X_i^{t,W}, & otherwise \end{cases} \tag{3.6}$$

The improvement of competitive swarm optimization algorithm can be achieved through ranking pairing learning and differential evolution. Fig 3.2 is the process of the ImCSO.

In Fig. 3.2 the first step of the ImCSO algorithm is to set the population size, maximum number of iterations, and algorithm parameters. The second is to initialize the population. The third is to evaluate all particles and record the globally optimal particle as *Gbest*. The fourth step is to divide the population into GW and GL based on fitness information. The fifth step is to update the particles in GL through a pairing learning strategy. The sixth step is to evaluate the updated particles and update *Gbest*. The seventh step is to use DES to update the particles in GW. The eighth step is to evaluate the updated particles and update *Gbest*. The ninth step is to determine whether the termination condition is met. If so, output the result, otherwise return to the fourth step. When using ImCSO to solve MASED problems, the position of each particle in the population corresponds to an effective solution. The position information X_i of the effective solution of the MASED system composed of M regions is equation (3.7).

$$X\_i = \Big[\underbrace{P_{11}, P_{12}, \cdots, P_{1N_1}}_{power \quad in \quad area \quad 1}, \underbrace{P_{21}, P_{22}, \cdots, P_{2N_2}}_{power \quad in \quad area \quad 2}, \cdots, \underbrace{P_{M1}, P_{M2}, \cdots, P_{MN_M}}_{power \quad in \quad area \quad M}, \underbrace{T_{12}, T_{13}, \cdots, T_{(M-1)M}}_{power \quad transmission}\Big] \tag{3.7}$$

In equation (3.7), P represents the output power (OP) of the generator set. T is the transmission power between regions. N and Mrepresent the number of generators and regions. The most crucial aspect in solving MASED problems is inequality and equality constraints. For the constraint of power generation capacity, the repair of N_{wq} OP is equation (3.8).

$$P_{wq} = \begin{cases} P_{wq}^{\min}, & if \quad P_{wq} \leq P_{wq}^{\min} \\ P_{wq}^{\max}, & if \quad P_{wq} \geq P_{wq}^{\max} \\ P_{wq}, & otherwise \end{cases} \tag{3.8}$$

In equation (3.8), N_{wq} represents the q-the generator unit in the w-the region. P_{wq} represents the actual OP of the q-the generator unit in the w-the region. $P_{wq}^{\min}$ represents the minimum OP of the unit. $P_{wq}^{\max}$ is the maximum OP of the unit. Regarding the transmission capacity constraints of the interconnection line, the

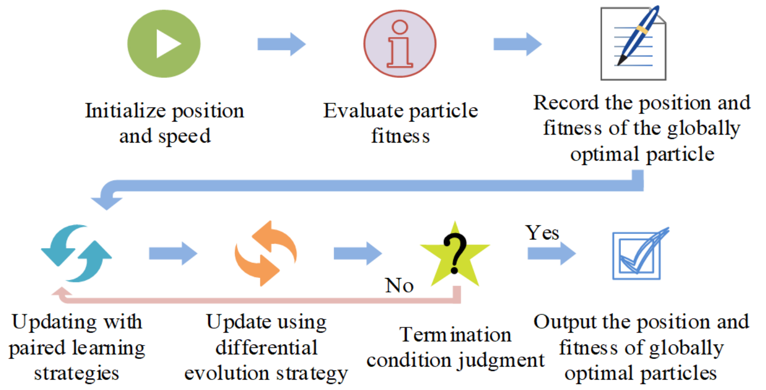


Fig. 3.3: Specific steps for solving MASED problems using ImCSO

transmission power repair between regions is equation (3.9).

$$T_{wk}=\begin{cases} T_{wk}^{\min}, & if \quad T_{wk}\leq T_{wk}^{\min} \\ T_{wk}^{\max}, & if \quad T_{wk}\geq T_{wk}^{\max} \\ T_{wk}, & otherwise \end{cases} \tag{3.9}$$

In equation (3.9), T_{wk} represents the transmission power from region w to region k. $T_{wk}^{\min}$ is the minimum value. $T_{wk}^{\max}$ is the maximum value. For the constraint of prohibited operation zone, if $P_{wq,m}^{l}<P_{wq}<P_{wq,m}^{u}$, its repair is equation (3.10).

$$P_{wq}=\begin{cases} P_{wq,m}^{l}, & if \quad P_{wq,m}^{l}<P_{wq}\leq\left(P_{wq,m}^{l}+P_{wq,m}^{u}\right)/2 \\ P_{wq,m}^{u}, & if \quad \left(P_{wq,m}^{l}+P_{wq,m}^{u}\right)/2<P_{wq}<P_{wq,m}^{u} \end{cases} \tag{3.10}$$

In equation (3.10), m represents the m-the prohibited operation zone. $P_{wq,m}^{u}$ represents the upper boundary of the m of the q-the generator unit in the w-the region. $P_{wq,m}^{l}$ represents the lower boundary. For the actual power balance constraints in the region, it is necessary to use a power balance repair operator. The repair of this constraint requires calculating the violation degree dif_w of the w-the region, as shown in equation (3.11).

$$dif_w=PG_w-\left(PD_w+PL_w+\sum_{k=1.k\neq w}^{M}T_{wk}\right) \tag{3.11}$$

In equation (3.10), PG_w represents the total electricity generation of region w. PG_w represents the load demand of region w. PG_w is the network loss of region w. If $dif_w>0$, select a generator set in region w to OP $P_{wq}=\max\left(P_{wq}-dif_w,P_{wq}^{\min}\right)$. If $dif_w<0$, then $P_{wq}=\min\left(P_{wq}+dif_w,P_{wq}^{\max}\right)$. Due to the fact that the equality constraints of the entire MASED system after repair are not fully satisfied, the study further considers the penalty function as the objective function to solve fitness, as shown in equation (3.12).

$$Fit\left(X_w\right)=FC\left(X_w\right)+factor\left(V_1+V_2+\cdots+V_M\right) \tag{3.12}$$

In equation (3.12), $FC\left(X_w\right)$ represents the total fuel cost. $factor$ represents the penalty factor. V_w is the degree of power balance constraint violation in repaired region w. The specific steps for solving the MASED problem using ImCSO are shown in Fig. 3.3.

In Fig. 3.3, the first step in solving the MASED using ImCSO is to initialize the positions and velocities of all particles in the population. The second step is to evaluate the fitness of all particles. The third step is to record the position of the globally optimal particle and its corresponding fitness information. The fourth step is to update GL with a paired learning strategy. The fifth step is to update GW using DES. The sixth step is to determine the termination conditions. If so, proceed to the next step, otherwise go back to step four. The seventh step is to output the position of the globally optimal particle and its corresponding fitness information. The research also adopts the ImCSO algorithm for solving the MADED problem, and the specific solving steps are consistent with the MASED problem.

3.2. Design of Multi regional Environmental and Economic Scheduling Method Based on CMOPSO Algorithm.. MAEED is a multi-objective optimization (MOO) that requires ensuring the comprehensive optimization of power generation costs and pollution emissions in the power system. This study first explains the solution methods for MOO, and then designs the specific steps for solving MAEED problems using the CMOPSO. MOO does not have a unique optimal solution, and can only achieve a relatively optimal overall goal. MOO is mainly solved through the concept of Pareto optimality, involving Pareto dominance, Pareto frontiers, etc. [16, 17]. The effectiveness of different algorithms in solving MOO problems varies. Therefore, this study used distribution uniformity index and comprehensive performance index inverse generation distance to evaluate the performance of the algorithm in solving MOO problems. The expression of the distribution uniformity index is shown in equation (3.13) [18].

$$\Delta(B,S)=\frac{\sum_{w=1}^{\alpha} d(E_w,B)+\sum_{X\in B}\left|d(X,B)-\bar{d}\right|}{\sum_{w=1}^{\alpha} d(E_w,B)+|S|\,\bar{d}} \tag{3.13}$$

In equation (3.13), S represents the set of points uniformly distributed on the leading edge of the real Pareto. B represents the Pareto optimal solution set. E_w is the extreme solution. Where α is the sequence number of the extreme solution, with a value range of $[1-\phi]$. ϕ represents the number of targets. X represents the solution, and d is the calculation of the minimum Euclidean distance. $\bar{d}$ represents the average value of the min Euclidean distance. $|S|$ is the number of concentration points. The calculation of the inverse generation distance of the comprehensive performance index is shown in equation (3.14) [19].

$$IGD(B,S)=\frac{\sum_{\delta\in S} d(\delta,B)}{|S|} \tag{3.14}$$

In equation (3.14), δ represents the point on the true frontier. The focus of MOO is to handle multiple constraint conditions, and common constraint processing methods can easily lead to local optimization problems in the algorithm. To avoid this issue, the study adopted the ε constraint criterion and the multi archive set method to handle the multi constraint problem of MOO. The expression of ε in the ε constraint criterion is equation (3.15).

$$\varepsilon(Gen)=\varepsilon(0)\times\left(1-\frac{Gen}{\max Gen}\right)^{cp} \tag{3.15}$$

In equation (3.15), ε is the variable value. $\varepsilon(0)$ represents the initial threshold. Gen represents the current iterations. cp is an index with a value of 2. $\max Gen$ represents the maximum iterations of the population. The core idea of multi archive constraint processing is classification, which categorizes solutions from different categories into different sets. The steps of processing MOO by combining the ε-constraint criterion and the multi archive set method are divided into three steps. The first step is to partition the solution with the support of the ε constraint. The second is to sort the infeasible solutions. The third is to determine whether the particles in the feasible solution meet the size of the population. MOPSO has the advantages of simple structure and fewer parameters that need to be adjusted, and is often used for engineering optimization problems. The process of this algorithm is Fig 3.4.

In Fig. 3.4, the first step of the MOPSO algorithm is to initialize the population, calculate the target vector, and liberate non dominated data into external files. The second step is to update the $Gbest$ and individual optimal value $Pbest$ in the population. The third step is to update the velocity and position information of the particles, calculate the target vector, and then update $Pbest$. The fourth step is to select $Gbest$. The fifth step is to determine the termination condition. If it is determined to be yes, output the external file

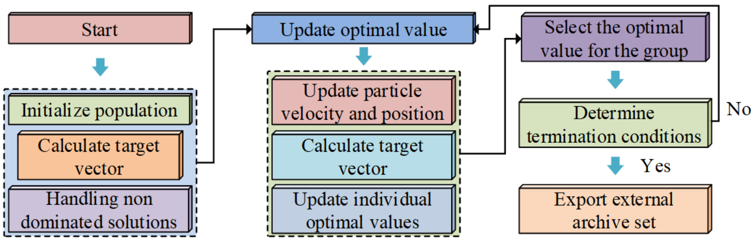


Fig. 3.4: Process of MOPSO

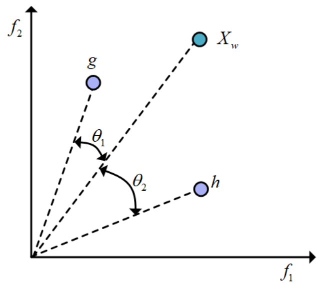


Fig. 3.5: The competitive strategy of CMOPSO algorithm

set; otherwise, return to the second step. However, the MOPSO algorithm did not solve the balance problem between population convergence and diversity, so the study introduced a competition mechanism and formed the CMOPSO algorithm. The competition mechanism mainly involves competitive learning strategies. The core idea is to let elite particles compete and then let the winning particles guide the update of the current particles. The competitive learning mechanism includes three aspects. The first aspect is elite particle selection, which is studied using the non dominated sorting genetic algorithm II. The second aspect is spatial competition, and the competition strategy of the CMOPSO algorithm is Fig 3.5.

In Fig. 3.5, both g and h are particles. θ_1 and θ_2 are both the angles between particles and X_w. If $\theta_1 > \theta_2$, h wins and is recorded as X_W. The third aspect is learning strategies. The effective solution position information expression when using the CMOPSO algorithm to solve the MAEED problem is consistent with MASED, and the repair method of constraint conditions is also the same as MASED. The process of using the CMOPSO algorithm to solve the MAEED problem is Fig 3.6.

In Fig. 3.6, the first step of the CMOPSO algorithm in solving the MAEED problem is to initialize the positions and velocities of all particles in the population. The second step is to fix the constraint violation of particles, and then calculate the target values of fuel cost and pollution emissions. The third step is to perform non dominated sorting and crowding distance sorting, and then determine the elite population. The fourth step is to select the winning particles. The fifth step is to update the velocity and position information of particles. The sixth step is to mutate and repair the particles, and calculate their objective function values. The seventh step is to determine the termination condition. If yes, output Pareto frontier particle information; otherwise, return to the third step.

4. Analysis of Economic Dispatching Results for MRPS Based on CMOPSO and ImCSO Algorithms. This study sets the simulation environment and running times for the ImCSO algorithm and CMOPSO algorithm, and verifies their effectiveness through algorithm comparison. The comparison indicators of the algorithm include fuel cost, cost convergence curve, Pareto frontier, distribution uniformity index, and comprehensive performance index inverse generation distance.

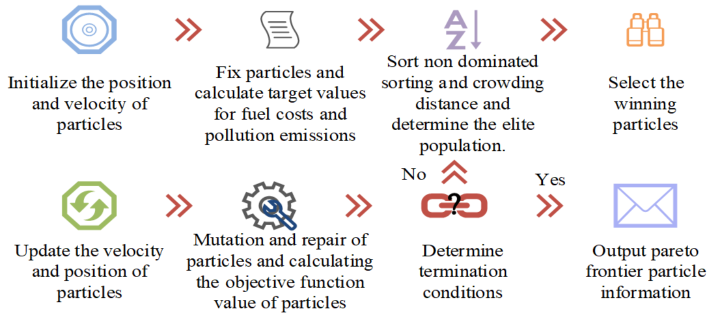


Fig. 3.6: The process of using CMOPSO algorithm to solve MAEED problems

Table 4.1: Comparison of fuel costs with different algorithms under MASED test system 1

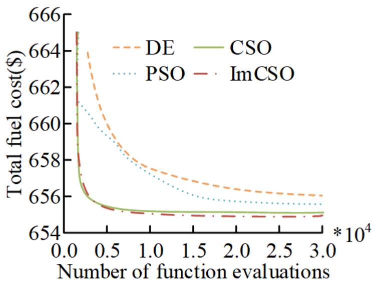
Algorithm	Minimum value($/h)	Mean value($/h)	Maximum value($/h)	Standard deviation	Time(s)
DE	656.2642	657.0810	658.5198	5.7913	1.43
PSO	655.9438	656.6521	661.2338	11.3302	1.64
CSO	655.8728	656.1905	657.0903	3.7779	1.58
ImCSO	655.8592	655.9866	656.2243	1.9642	1.60

4.1. Analysis of S-DE dispatch results in multiple regions based on ImCSO algorithm. In order to conduct simulation analysis on the ImCSO algorithm, two MASED testing systems and two MADED testing systems were selected for the study. A comparative analysis was conducted on the effectiveness verification of the ImCSO algorithm. Comparative algorithms include Competitive Swarm Optimizer (CSO), PSO, and Differential Evolution (DE). The simulation environment for comparing algorithms is MATLAB 9.6, with 10 runs. Table 1 shows the fuel cost comparison of different algorithms under MASED test system 1.

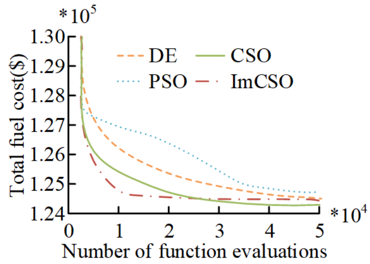
In Table 4.1, the maximum, minimum, average, and standard deviation of the fuel cost of the ImCSO algorithm are lower than those of other comparative algorithms, with values of 656.2243 $/h, 655.8592 $/h, 655.9866 $/h, and 1.9642, respectively. The running time of the ImCSO algorithm is 1.6 seconds, which is not significantly different from other comparative algorithms. The fuel cost values of the DE algorithm are 658.5198 $/h, 656.2642 $/h, 657.0810 $/h, and 5.7913, respectively, with a running time of 1.43 seconds. The four values of fuel cost for the PSO algorithm are 661.2338 $/h, 655.9438 $/h, 656.6521 $/h, and 11.3302, respectively, with a running time of 1.64 seconds. The relevant values for the fuel cost of the CSO algorithm are 657.0903 $/h, 655.8728 $/h, 656.1905 $/h, and 3.7779, respectively, with a running time of 1.58 seconds. Therefore, the performance of the ImCSO algorithm is better and more stable. The comparison of cost convergence curves of different algorithms under different MASED testing systems is Fig 4.1.

As Fig. 4.1a, with the increase of iterations, the total fuel cost of different algorithms gradually decreases. In Test System 1, the ImCSO algorithm flattened after nearly 4000 iterations, while the DE, PSO, and CSO algorithms stabilized after nearly 15000, 14000, and 5000 iterations, respectively. In test system 2 of Figure 4.1b, the ImCSO algorithm tends to flatten out after nearly 10000 iterations, while the DE, PSO, and CSO algorithms tend to flatten out after 30000, 32000, and 25000 iterations, respectively. From this, the ImCSO algorithm converges faster and has better performance. Table 2 is the fuel cost comparison of different algorithms under MADED test system 1.

In Table 4.2, the minimum fuel cost values for ImCSO, DE, PSO, and CSO algorithms are 13003.9526 $/h, 13166.7657 $/h, 13492.8771 $/h, and 13476.6407 $/h, respectively. The average fuel costs of the four algorithms are 13151.3299 $/h, 13291.1036 $/h, 14167.4435 $/h, and 16795.4673 $/h. The maximum fuel costs are 13299.2825 $/h, 13434.3578 $/h, 17826.6214 $/h, and 34830.2087 $/h. The standard fuel cost values are 78.1757, 69.2933762.1120, and 4659.4287. The running time of each algorithm is 81.427s, 179.1231s, 127.6157s,



(a) Cost convergence curve of MASED test system 1



(b) Cost convergence curve of MASED test system 2

Fig. 4.1: Comparison of cost convergence curves of different algorithms in different MASED test systems

Table 4.2: Comparison of fuel cost of different algorithms under MADED test system 1

Algorithm	Minimum value($/h)	Mean value($/h)	Maximum value($/h)	Standard deviation	Time(s)
DE	13166.7657	13291.1036	13434.3578	69.2933	179.1231
PSO	13492.8771	14167.4435	17826.6214	762.1120	127.6157
CSO	13476.6407	16795.4673	34830.2087	4659.4287	116.0529
ImCSO	13003.9526	13151.3299	13299.2825	78.1757	81.427

and 116.0529s respectively. Therefore, it can be concluded that the ImCSO algorithm has the smallest running time, and the maximum, minimum, and average fuel costs are lower than other algorithms. And this also indicates that the ImCSO algorithm performs better in solving MADED problems. In order to further validate the performance of the ImCSO algorithm, other algorithms were selected for comparison in the study. The selected algorithms for the study include Gravitational Search Algorithm (GSA), Gbest guided Artificial Bee Colony (GABC), Teaching Learning Based Optimization (TLBO), and Differential Evolution Algorithm with Strategy Adaptation (DESA). The F1 values and CPU utilization of different algorithms are compared in Table 4.3.

From Table 4.3, it can be seen that in terms of CPU utilization, the maximum value of the ImCSO algorithm is 17.5%, and the minimum value is 16.1%. The maximum values of DE, PSO, CSO, GSA, GABC, TLBO, and DESA algorithms are 28.2%, 22.8%, 21.5%, 20.8%, 24.8%, 23.6%, and 27.1%, respectively, while the minimum values are 26.7%, 21.3%, 19.2%, 18.5%, 23.6%, 22.1%, and 25.5%, respectively. On the F1 value, the maximum value of the ImCSO algorithm is 0.993 and the minimum value is 0.976. The maximum values of DE, PSO, CSO, GSA, GABC, TLBO, and DESA algorithms are 0.837, 0.938, 0.953, 0.966, 0.888, 0.924, and 0.861, respectively, while the minimum values are 0.811, 0.921, 0.938, 0.953, 0.857, 0.902, and 0.834, respectively. It can be seen that the ImCSO algorithm has advantages in CPU utilization and F1 value, indicating better performance of the algorithm.

4.2. Analysis of multi-regional environmental and economic dispatch results based on CMOPSO algorithm.. Two MAEED testing systems were also selected for the simulation analysis of the CMOPSO algorithm. To verify the effectiveness of the CMOPSO algorithm, a comparative analysis was conducted. Comparison algorithms include MOPSO, BB-MOPSO [20], and TV-MOPSO [12]. The comparison content includes Pareto Frontier, Distribution Uniformity Index, and Comprehensive Performance Index Reverse Generation Distance. The simulation environment for comparing algorithms is MATLAB 9.6, with 10 runs. The Pareto frontier comparison of different algorithms under different MAEED testing systems is shown in Fig 4.2.

From 4.2a, different algorithms have more repetitions in obtaining Pareto frontiers, but the CMOPSO

Table 4.3: Comparison of F1 values and CPU utilization of different algorithms

Algorithm	CPU utilization					F1				
	Number of experiments					Number of experiments				
	1	2	3	4	5	1	2	3	4	5
DE	26.7%	27.3%	26.8%	27.4%	28.2%	0.815	0.837	0.826	0.811	0.832
PSO	21.7%	21.3%	22.6%	22.8%	21.5%	0.925	0.934	0.938	0.921	0.933
CSO	20.8%	19.5%	21.5%	20.2%	19.2%	0.938	0.949	0.953	0.952	0.945
GSA	19.3%	20.8%	19.1%	18.5%	19.4%	0.957	0.966	0.955	0.961	0.953
GABC	23.7%	24.3%	24.5%	23.6%	24.8%	0.871	0.857	0.866	0.888	0.865
TLBO	22.5%	23.2%	23.6%	22.1%	23.4%	0.902	0.913	0.908	0.924	0.917
DESA	25.9%	26.7%	25.5%	27.1%	26.9%	0.855	0.842	0.857	0.861	0.834
ImCSO	17.2%	16.8%	17.5%	16.1%	17.1%	0.977	0.983	0.989	0.976	0.993

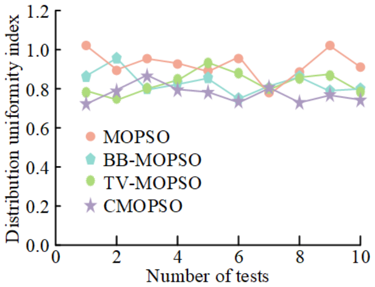
(a) Pareto frontiers of MAEED test system 1

(b) Pareto frontiers of MAEED test system 2

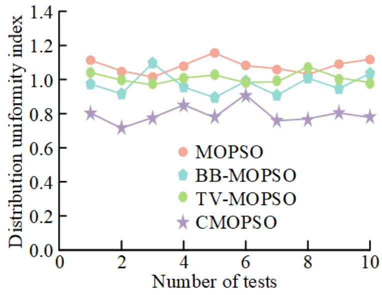
Fig. 4.2: Pareto frontier comparison of different algorithms in different MAEED test systems

algorithm has a wider distribution of Pareto frontiers than other algorithms, and the extreme solutions are also better than other algorithms. In Figure 4.2b, there is a significant difference in the Pareto frontier obtained by different algorithms. The distribution of Pareto frontiers obtained by the CMOPSO algorithm is significantly higher than other algorithms, and also higher. From this, the CMOPSO algorithm performs better and achieves better extreme solutions. The comparison of distribution uniformity indicators of different algorithms under different MAEED testing systems is Fig 4.3.

In Fig. 4.3a in MAEED test system 1, the maximum value of the MOPSO algorithm's distribution uniformity index is 1.0578, the min value is 0.7472, and the average value is 0.8669. The max values of the distribution uniformity indicators for the BB-MOPSO algorithm and TV-MOPSO algorithm are 0.9873 and 0.9625, respectively, and the min values are 0.7152 and 0.7187. The average values are 0.8358 and 0.8575. The max value of the distribution uniformity index of the CMOPSO algorithm is 0.9051, the minimum value is 0.6919, and the average value is 0.8058. According to 4.3b, in MAEED test system 2, the maximum value of the MOPSO algorithm's distribution uniformity index is 1.1872, the minimum value is 0.9815, and the average value is 1.0498. The maximum values of the distribution uniformity indicators for the BB-MOPSO algorithm and TV-MOPSO algorithm are 1.1308 and 1.1032, and the min values are 0.8617 and 0.9353. The average values are 0.9937 and 1.0149, respectively. The max value of the distribution uniformity index of the CMOPSO algorithm is 0.9605, the min value is 0.68, and the average value is 0.8457. From this, the distribution uniformity index of the CMOPSO algorithm is superior to the comparison algorithm, which also indicates that the algorithm has superiority in MAEED problems. The comparison of the comprehensive performance indicators of different algorithms under different MAEED testing systems in terms of inverse generation distance is shown in Fig 4.4.

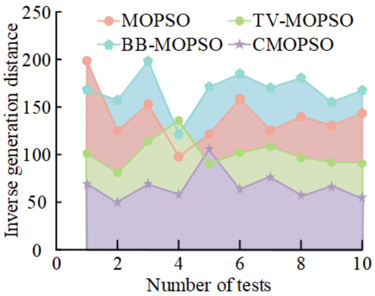


(a) Comparison of distribution uniformity indicators of different algorithms in MAEED test system 1

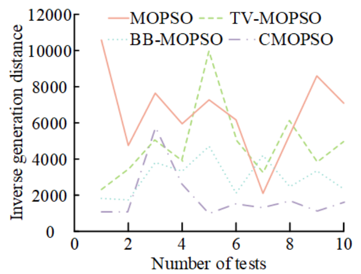


(b) Comparison of distribution uniformity indicators of different algorithms in MAEED test system 2

Fig. 4.3: Comparison of distribution uniformity indexes of different algorithms in different MAEED test systems



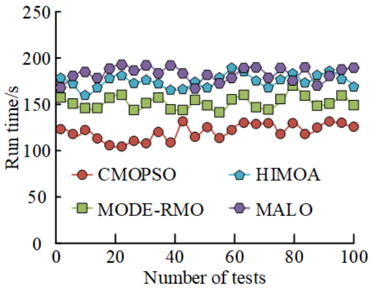
(a) Comparison of inverse generation distance between different algorithms in MAEED testing system 1



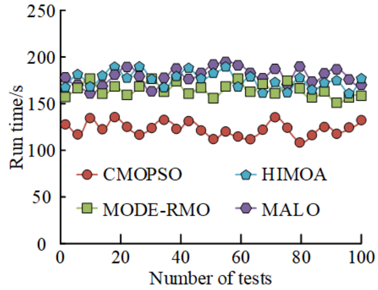
(b) Comparison of inverse generation distance between different algorithms in MAEED testing system 2

Fig. 4.4: Comparison of the comprehensive performance index of different algorithms in different MAEED test systems

From Fig. 4.4a, in MAEED test system 1, the maximum anti generation distances of the MOPSO algorithm, BB-MOPSO algorithm, and TV-MOPSO algorithm are 204.3648, 204.6281, and 143.2398, respectively. The minimum values are 93.1586, 116.6733, and 76.4959, respectively, and the average values are 123.6742, 155.883, and 95.2604, respectively. The maximum anti generation distance of the CMOPSO algorithm is 112.6341, the min value is 43.9868, and the average value is 67.6316. From Figure 4.4b, in MAEED test system 2, the maximum inverse generation distances of the three comparison algorithms are 10879.9507, 4959.483, and 10117. 1681, The minimum values are 1769.6697, 1473.3904, and 2036.2403, respectively; The average values are 6675.7363, 2898.1252, and 4685.6277, respectively. The maximum anti generation distance of the CMOPSO algorithm is 6121.3624, the minimum value is 702.5518, and the average value is 1664.0978. From this, the CMOPSO algorithm has the best comprehensive performance. In order to further verify the performance of the CMOPSO algorithm, other algorithms were selected for comparison in the study. Comparison algorithms include Multi Objective Differential Evolution with Ranking based Mutation Operator (MODE-RMO), Hybrid Immune Multi Objective Optimization Algorithm (HIMOA), and Multi Objective Ant Lion Optimization Algorithm (MALO). The experiment was conducted a total of 100 times. The comparison of runtime of different algorithms under different MAEED testing systems is shown in Figure 4.5.



(a) Comparison of runtime of different algorithms in the MAEED1 testing system



(b) Comparison of runtime of different algorithms in the MAEED2 testing system

Fig. 4.5: Comparison of runtime of different algorithms in different MAEED testing systems

From Figure 4.5a, it can be seen that in MAEED test system 1, the maximum running time of the CMOPSO algorithm is 127.8 seconds, and the minimum value is 101.3 seconds. The maximum running times of MODE-RMO, HIMOA, and MALO algorithms are 169.8 seconds, 180.2 seconds, and 188.9 seconds, respectively, while the minimum values are 147.4 seconds, 158.9 seconds, and 163.3 seconds, respectively. As shown in Figure 4.5b, in MAEED test system 2, the maximum running time of the CMOPSO algorithm is 139.6 seconds, and the minimum value is 118.4 seconds. The maximum running times of MODE-RMO, HIMOA, and MALO algorithms are 172.9s, 183.6s, and 190.8s, respectively, while the minimum values are 153.4s, 161.5s, and 165.2s, respectively. From this, it can be seen that the CMOPSO algorithm has obvious advantages in runtime and better performance.

5. Conclusion. In response to the improvement of energy utilization efficiency, this study innovatively proposes the use of CMOPSO algorithm to solve multi regional environmental and economic scheduling problems. Moreover, the ImCSO algorithm is adopted to handle S-DE scheduling matters in multiple regions. Research showed that the maximum, minimum, and average fuel costs of the ImCSO for solving MASED problems were 656.2243 $/h, 655.8592 $/h, and 655.9866 $/h, respectively. The maximum, minimum, and average fuel costs of the ImCSO algorithm for solving MADED problems were 13299.2825 $/h, 13003.9526 $/h, and 13151.3299 $/h, respectively. All values were smaller than other comparison algorithms. From this, the ImCSO algorithm performed better in solving MASED and MADED problems. The CMOPSO algorithm have a wider Pareto frontier distribution when solving MAEED problems. Under different testing systems, the maximum values of the distribution uniformity index of the CMOPSO algorithm were 0.9051 and 0.9605, the minimum values were 0.6919 and 0.68, and the average values were 0.8058 and 0.8457, respectively. Under different testing systems, the maximum anti generation distance of the CMOPSO algorithm was 112.6341 and 6121.3624, while the minimum value was 43.9868 and 702.5518, respectively. The average value was 67.6316 and 1664.0978. From this, the performance of the CMOPSO algorithm is superior to that of the comparison algorithm. However, there are also certain shortcomings in the research, which only considers multi-objective issues of the environment and economy, and does not involve many other factors, which is also an area for further research to improve.

REFERENCES

[1] Chen, S., Zhang, L., Yan, Z. & And, S. and robust security-constrained economic dispatch algorithm based on blockchain. *IEEE Transactions On Power Systems*. **37**, 691-700 (2021)

[2] Wen, G., Yu, X. & Liu, Z. Recent progress on the study of distributed economic dispatch in smart grid: an overview. *Frontiers Of Information Technology & Electronic Engineering*. **22**, 25-39 (2021)

[3] Huang, B., Li, Y., Zhan, F., Sun, Q. & Zhang, H. distributed robust economic dispatch strategy for integrated energy system considering cyber-attacks. *IEEE Transactions On Industrial Informatics.* **18**, 880-890 (2021)

[4] Younes, Z., Anharmonic, I., Makhila, S. & Reya Sudin, M. memory-based gravitational search algorithm for solving economic dispatch problem in micro-grid. *Ain Shams Engineering Journal.* **12**, 1985-1994 (2021)

[5] Lin, L., Guan, X., Peng, Y., Wang, N., Maharjan, S. & Ootsuka, T. Deep reinforcement learning for economic dispatch of virtual power plant in internet of energy. *IEEE Internet Of Things Journal.* **7**, 6288-6301 (2020)

[6] Goudarzi, A., Fahad, S., Ni, J., Ghayoor, F., Siano, P. & Achelous, H. sequential hybridization of ETLBO and IPSO for solving reserve-constrained combined heat, power and economic dispatch problem. *IET Generation, Transmission & Distribution.* **16**, 1930-1949 (2022)

[7] Lyu, C., Jia, Y. & Xu, Z. DRO-MPC-based data-driven approach to real-time economic dispatch for islanded microgrids. *IET Generation Transmission & Distribution.* **14**, 5704-5711 (2020)

[8] Wang, X., Liu, R., Wang, X., Hou, Y. & Bouffard, F. Data-Driven Uncertainty Quantification Method for Stochastic Economic Dispatch. *IEEE Transactions On Power Systems.* **37**, 812-815 (2022)

[9] Marco, A., Cecilia, M. & Guadalupe, C. power optimization model for the long-term planning scenarios: Case study of Mexico's power system decarbonization. *The Canadian Journal Of Chemical Engineering.* **99**, 884-897 (2020)

[10] Lev, K., Tang, H., Akenson, B., Pillai, J., Tan, Q. & Zhang, Q. Hierarchical learning optimization method for the coordination dispatch of the inter-regional power grid considering the quality-of-service index. *IET Generation, Transmission & Distribution.* **14**, 3673-3684 (2020)

[11] Xu, D., Liang, F., Cheng, Q., Zhou, J. & Tang, J. Research on joint optimization model and algorithm of multi-area generation and reserve with considering its availability. *IET Generation, Transmission & Distribution.* **16**, 2032-2048 (2022)

[12] Wang, S., Duan, J., Shi, D., Xu, C. & Wang, Z. Data-driven Multi-agent Autonomous Voltage Control Framework Using Deep Reinforcement Learning. *IEEE Transactions On Power Systems.* **35**, 4644-4654 (2020)

[13] Shaheen, A., Ragab, A., Elsayed, A. & Elattar, E. Multi-objective manta ray foraging algorithm for efficient operation of hybrid AC/DC power grids with emission minimization. *IET Generation, Transmission And Distribution.* **15**, 1314-1336 (2020)

[14] Shaheen, A., Elsayed, A., Gunadi, A., El-Shimmy, R., Alharthi, M. & Ghoneim, S. novel improved marine predators' algorithm for combined heat and power economic dispatch problem. *Alexandria Engineering Journal.* **61**, 1834-1851 (2022)

[15] Wang, G., Zhang, T., Yang, Y., Cheng, J., Chang, J., Liang, X. & Hou, Z. Cross-modality paired-images generation for RGB-infrared person reidentification. *Proceedings Of The AAAI Conference On Artificial Intelligence 34(7.* pp. 12144-12151 (2020)

[16] Barma, M. & Modibbo, U. Multiobjective mathematical optimization model for municipal solid waste management with economic analysis of reuse/recycling recovered waste materials. *Journal Of Computational And Cognitive Engineering.* **1**, 122-137 (2022)

[17] Abdollahzadeh, B. & Gharehchopogh, F. multi-objective optimization algorithm for feature selection problems. *Engineering With Computers.* **38**, 1845-1863 (2022)

[18] Wu, C. Li C Distribution uniformity of water-based binders in si anodes and the distribution effects on cell performance. *ACS Sustainable Chemistry & Engineering.* **8**, 6868-6876 (2020)

[19] Miguel Nacarlos, M., Wolgemuth, J., Haraf, S. & Fisk, N. Anti-oppressive pedagogies in online learning: A critical review. *Distance Education.* **41**, 345-360 (2020)

[20] Xie, Y., Du, L., Zhao, J., Liu, C. & Li, W. Multi-objective optimization of process parameters in stamping based on an improved RBM–BPNN network and MOPSO algorithm. *Structural And Multidisciplinary Optimization.* **64**, 4209-4235 (2021)

[21] Wang, S., Zhang, T., Kong, W., Wen, G. & Yu, Y. An improved MOPSO approach with adaptive strategy for identifying biomarkers from gene expression dataset. *Mathematical Biosciences And Engineering: MBE.* **20**, 1580-1598 (2022)

Edited by: Zhengyi Chai
Special issue on: Data-Driven Optimization Algorithms for Sustainable and Smart City
Received: Oct 16, 2023
Accepted: Mar 20, 2024