The Economic Load Scheduling Based on Particle Swarm Optimization

Rongxuan Chen 1, Ning Zhao 2,a

1 Guohua Sanhe Power Co., Ltd. Langfang 065201, China
2 North China Electric Power Research Institute Co., Ltd. Beijing 100045, China
azhao.ning@jibei.sgcc.com.cn
*Corresponding author’s e-mail: zn77816@sina.com

Abstract. With the increasing cost of power generation, the economic load dispatching has been paid more and more attention. Considering the environmental temperature as the influencing factor of the economic dispatch of power plant load, it has a stronger adaptability to the working condition than the study that only considers the total load. In this paper, the application of particle swarm optimization in load economic scheduling was introduced. And combined with the actual operation data on two typical units, a comparative analysis was made on the load economic scheduling with and without environmental temperature. It was shown in the analysis results that the unit heat consumption rate decreases with the increase of total load and increases with the increase of temperature when the ambient temperature is taken into account or not. The unit heat consumption rate considering ambient temperature was always lower than that without ambient temperature. And as the temperature drops, the difference got bigger and bigger. The conclusion of the analysis has certain reference significance to the current load economic dispatching of power plants.

1. Introduction
With the introduction of a new round of power reform, reducing the cost of power generation was the lifeblood of every power generation enterprise. Economic load dispatching was an effective means to reduce the cost, so the problem had been paid attention to by power generation enterprises without exception. Economic load dispatching not only improved the economic benefits of power generation enterprises, but also reduced the emission of pollutants.

In view of the load economic dispatch, the related experts and scholars put forward a variety of schemes. It could be roughly classified into two categories: traditional methods and intelligent algorithms. The premise of the traditional method was that the change rate of energy consumption of the generator set was monotonically increasing with the load. This relation was not always true in practice, so the application of this method was limited. The calculation process of linear programming method and dynamic programming method was complicated, and it was easy to fall into "dimensional disaster" when solving multivariable problems. The artificial neural network algorithm in the intelligent algorithm was an algorithm based on gradient, which was easy to fall into the local extremum and had a slow convergence speed in the actual application process, and was not easy to be used for online load allocation adjustment Genetic algorithm was an algorithm that simulates the "survival of the fittest" in the process of biological evolution in nature. Particle swarm optimization (PSO) was an algorithm that simulates the foraging process of birds to search the solution space in
parallel. In order to prevent the phenomenon of "prematurity", this paper introduced dynamic inertia weight, contraction factor and dynamic learning factor to improve the particle swarm optimization algorithm.

There were many factors that affect the energy consumption of the generator set. The economic load dispatching aiming at energy saving was actually a multi-boundary optimization problem. In the past, most studies only considered the single variable of total load, which was bound to cause poor adaptability of the analysis results. In this paper, the environmental temperature and the total load were considered together to seek for more accurate and closer to the actual load distribution results, so as to further reduce energy consumption. The actual operation data of two typical units were used to compare and analyse the two distribution methods with and without considering the ambient temperature. The analysis results had certain reference value to the current load economic dispatching.

2. Mathematical model

The economic scheduling of power plant load was to minimize the energy consumed by the power plant under the boundary conditions of satisfying the needs of users and safe operation of equipment. Energy consumption was expressed by heat consumption rate, so the goal of load economic dispatching was to minimize the heat consumption rate under all boundary conditions. It could be seen that the mathematical model of the problem could be established with the following objective function and constraint conditions:

\[
\min F(P) = \sum_{t=1}^{T} \sum_{k=1}^{n} [U_{kt} f(P_k) + U_{kt} (1 - U_{kt}) S_k] 
\]

Where, \( F(P) \) was the heat consumption rate of all units when the total load was \( P \); \( f(P_k) \) was the heat consumption rate when the load of set \( k \) was \( P_k \); \( U_{kt} \) represents the operating state of the \( k \) and \( t \) units. The value of \( U_{kt} \) was: \( U_{kt}=1 \), indicating normal operation of the unit in time period \( t \); \( U_{kt}=0 \), indicating that the unit stops operating in time period \( t \); \( S_k \) was the energy consumed during starting and stopping of unit \( k \); \( T \) was the total number of sessions; \( n \) was the total number of units.

(1) Single unit load constraint

\[ P_{k\text{min}} \leq P_k \leq P_{k\text{max}} \]

(2) Constraint equation of total load

\[ \sum_{k=1}^{n} U_{kt} P_{k\text{max}} \leq P \leq \sum_{k=1}^{n} U_{kt} P_{k\text{max}} \]

(3) Load balance relationship

\[ \sum_{k=1}^{n} U_{kt} P_k = P \]

(4) Rotation reserve relation

\[ P + R_t \leq \sum_{k=1}^{n} U_{kt} P_{k\text{max}} \]

Where: \( P_k \) was the load of unit \( k \); \( P_{k\text{min}} \) and \( P_{k\text{max}} \) were the minimum and maximum loads of unit \( k \), respectively. \( R_t \) was the rotating reserve capacity in time period \( t \).

3. Principle of particle swarm optimization

3.1. The basic principles of particle swarm optimization

Particle swarm optimization (PSO) was a typical representative of bionic algorithm, which was accepted by many scholars for its simple operation. Particle swarm optimization (PSO) was an algorithm that simulates the foraging process of birds. Each possible solution in the solution space corresponded to a particle in the particle space. Particles in the solution space constantly adjusted their "flight" state through their accumulated experience and social information sharing, so as to find the
global optimal solution quickly. The process of particle swarm optimization was controlled by the following two formulas:

(1) Speed update formula

\[
V_{mn}(l+1) = \omega V_{mn}(l) + c_1 r_1 (P_{mn}(l) - X_{mn}(l)) + c_2 r_2 (P_{global}(l) - X_{mn}(l))
\]

Where: \(V_{mn}(l+1)\) was the \(n\) dimension velocity of particle \(m\) in the \(l+1\) generation; \(\omega\) for inertia weight; \(c_1\) was the self-learning factor; \(c_2\) was the social information sharing factor; \(r_1\) and \(r_2\) were two random sequences between \([0~1]\). \(P_{mn}(l)\) was the best \(n\) dimensional space coordinate that the \(m\) particle "leaps" before the \(L\) generation (including). \(P_{global}(l)\) was the best global space coordinate that the \(l\) generation (including) has "leap-forward" before; \(X_{mn}(l)\) was the \(N\)TH dimensional coordinate of the \(m\) particle in the \(l\) generation. The three fractions of velocity made up the particle's speed of flight. The second was the adjustment of speed by one's own experience; the third was the influence of social information on speed adjustment. The influence of these three parts on speed was determined by inertia weight \(\omega\), self-learning coefficient \(c_1\) and social information sharing factor \(c_2\).

(2) Update formula of position coordinates

\[
X_{mn}(l+1) = X_{mn}(l) + V_{mn}(l+1)
\]

Where, \(X_{mn}(l+1)\) was the \(n\) dimensional space coordinate of the \(m\) particle in the \(l+1\) generation.

3.2. Improvement measures of particle swarm optimization algorithm

Although particle swarm optimization had the advantages of simple operation, it was found that particle swarm optimization algorithm was premature in practical application. To solve this problem, the following four improvement measures were cited:

(1) Improvement of "speeding" and "crossing the line"

Traditional particle swarm optimization (PSO) algorithms forced the speed of hypervelocity particles to be the corresponding maximum or minimum speed. The space coordinates of particles beyond the boundary position were forced to be set as the maximum space coordinates or the minimum space coordinates. Such a tough approach would cause a large number of particles to cluster on the boundary, reducing the ability to search the whole space. This paper adopted the rebound rule:

\[
X_{mn} = \begin{cases} 
2X_{min,m,n} - X_{mn} (X_{mn} \leq X_{min,m,n}) \\
1.8X_{max,m,n} - X_{mn} (X_{mn} \geq X_{max,m,n})
\end{cases}
\]

Where, \(X_{min,m,n}, X_{max,m,n}\) were the minimum and maximum spatial coordinates of the \(n\) dimension of the \(m\) particle.

\[
V_{mn} = \begin{cases} 
2V_{min,m,n} - V_{mn} (V_{mn} \leq V_{min,m,n}) \\
2V_{max,m,n} - V_{mn} (V_{mn} \geq V_{max,m,n})
\end{cases}
\]

Where, \(V_{min,m,n}, V_{max,m,n}\) were respectively the minimum and maximum velocities of the \(m\) particle in the \(n\) dimension.

(2) Improvement of "flight" speed

Particles should be searched globally at high speed in the early stage and at low speed near the optimal solution in the later stage. If the late speed was too large, it will run counter to the global optimal solution. To some extent, Gaussian sequence could make particles jump out of local extremum and expand the search space. Based on the above two considerations, contraction factor and Gaussian sequence were introduced in this paper.

\[
\chi = \frac{2}{\sqrt{4\pi\phi}}
\]

Where, \(\chi\) is the contraction factor, \(\phi = c_1 + c_2\).
\[ g = [-2\ln(u_1)]^{0.5} \times \cos[2\pi(u_2)] \]  

(11)

Where, \( g \) was Gaussian sequence, and \( u_1 \) and \( u_2 \) were random sequences between [0~1].

(3) Dynamic learning factor

In the iterative process, particle swarm should be based on its own experience because of the lack of social information in previous generations. In order to find the global extremum, we should attach importance to the sharing of social information. Therefore, \( c_1 \) and \( c_2 \) remained constant throughout the optimization process, which was unfavourable to the search process. In order to enhance the global collecting ability of particles, dynamic learning factor was introduced in this paper:

\[
\begin{align*}
  c_1(k + 1) &= c_1 - k/D \\
  c_2(k + 1) &= c_2 - k/2D
\end{align*}
\]  

(12)

Where, \( k \) was the current evolutionary algebra; \( D \) was the total evolutionary algebra.

(4) Dynamic inertia weight

The inertia weight controlled the ability of particle swarm to search the space. Otherwise, the search space was small. According to the characteristics of particle swarm search process, dynamic inertia weight was introduced:

\[
W(k) = \left[1 + c \left(\frac{k}{D} - 1\right)\right] W_0.
\]  

(13)

Where, \( N \) was the particle swarm size, and \( W_0 \) was the initial inertia weight. To sum up, the improved spatial coordinate and velocity updating formulas were shown in equation (14). The algorithm flow chart was shown in FIG 1.

\[
\begin{align*}
  V_{mis}(k + 1) &= \chi [\omega V_{mis}(k) + c_1 g(P_m(k) - X_{mis}(k))] \\
  X_{mis}(k + 1) &= X_{mis}(k) + V_{mis}(k + 1)
\end{align*}
\]  

(14)

FIG. 1 Flow chart of particle swarm optimization algorithm
4. The example analysis

According to the principle of the algorithm described above, particle swarm optimization was programmed into a program by software. The particle swarm had a scale of 100, with 1000 iterations. The initial maximum and minimum inertia weights were 1.1 and 0.6 respectively; the initial self-learning factor and social information sharing factor were both 2.1. The maximum and minimum "flight" speeds of 300 MW units were 0.3 and 0.1 respectively. The corresponding speeds of 600 MW units were 0.5 and 0.3. Particle swarm optimization (PSO) algorithm needed to judge the quality of individuals according to their adaptability to solution space. According to the actual operation data of a 300 MW unit and a 600 MW unit in a power plant, the relation of load, ambient temperature and heat consumption rate was fitted to the surface. The fitting surface of 300 MW unit was shown in FIG 2. The fitting surface of 600 MW unit was shown in FIG 3.

As can be seen from FIG. 2 and FIG. 3, the heat consumption rate of two generating sets decreased with the increase of load, and the rate of change of heat consumption rate gradually decreases. With the increased of load, the unit was closer to the design condition, so the unit heat consumption rate was lower. The heat consumption rate increased with the increase of temperature, and the change of heat consumption rate first increased and then decreased. Under the same load, the higher the ambient temperature was, the higher the exhaust pressure of the steam turbine would be. In order to maintain the same power, the higher the air intake would be, and the higher the heat consumption rate would be. 300 MW units range from -10 ℃ to 30 ℃, and the heat consumption rate was sensitive to the ambient temperature. The heat consumption rate of 300 MW units was sensitive to ambient temperature between -10 ℃ and 30 ℃. Particle swarm optimization (PSO) was used to calculate the load economic scheduling with and without environmental temperature. The calculation results were shown in FIG 4.

FIG. 2 Fitting surface of 300 MW unit

FIG. 3 Fitting surface of 600 MW unit

FIG. 4 Fitting surface of the calculated results
It could be clearly seen from FIG. 4 that, regardless of the ambient temperature, the heat consumption rate decreased with the increase of load and the rate of change becomes smaller. The heat consumption rate increased with the increase of temperature. The heat consumption rate of load economic dispatch considering ambient temperature was always lower than that of load economic dispatch without considering ambient temperature. And as the temperature drops, the difference got bigger and bigger.

5. Conclusion
(1) In this paper, the particle swarm optimization algorithm and some improvement measures were described, and the particle swarm optimization algorithm was used as the optimization algorithm to compare and analyse the load economic scheduling results with and without environmental temperature.

(2) Considering the influence of ambient temperature on economic load dispatching, the heat consumption rate of the unit decreased obviously. The distribution method considering the ambient temperature was more accurate and close to reality than that considering the single variable of load. It could be seen that the ambient temperature was of great significance to economic load dispatching.

(3) There were many factors affecting economic load dispatching, such as the characteristics of coal and boiler efficiency. This paper only studied the influence of total load and ambient temperature on load economic dispatch, and the influence of other factors on load economic dispatch was still in the further study.

Acknowledgments
This paper was supported by Guohua Sanhe Power Co., Ltd and North China Electric Power Research Institute Co., Ltd.

References
[1] Ma L. Wang S.X. Liu R.J. (2005) Economic load dispatching of power plant based on improved dynamic programming method. Journal of Beijing Jiaotong University. (29): 100-103.
[2] WU J. L. Zhang B.H. Wang K. (2012) Application of improved BP neural network algorithm based on Adaboost in short-term Wind speed prediction. The grid technology. (36): 221-225.
[3] Cui C.L. Fang Y.J. (2011) Application of accelerated Genetic Algorithm in dynamic Load Allocation of power plants. Electric automation equipment. (31): 63-66.
[4] Tang Y.G. Cui Y.H. Qiao L.J. (2009) Application of a new algorithm in economic load distribution. Journal of Power System and Automation. (21): 20-26.
[5] Liu Z.G. Zeng J.J. Han Z.W. (2012) Adaptive mutation perturbed particle Swarm optimization based on individual optimal location. Journal of Southwest Jiaotong University. (47): 761–767.
[6] Qin Q.D. Li R.J. (2011) A two-population particle swarm optimization algorithm based on parasitism behaviour. Decision and control. (26): 548–557.
[7] Si F.Q. Gu H. Ye Y.L. (2011) On-line optimization of plant load distribution in thermal power plant based on chaotic particle swarm optimization algorithm. Chinese Journal of Electrical Engineering. (31): 103–109.
[8] Zhu D.L. Zhan T. Zhang Y. (2014) Cellular multi-objective particle swarm optimization based on multi-strategy differential evolution. Electronic journals. (42): 1831–1838.
[9] Qiu X.Z Zhang D.L. Huang B.H. (2012) Economic scheduling of unit load based on improved particle swarm optimization algorithm. Steam turbine technology. (54): 294–296.
[10] Zeng L.Q. Luo F.B. Ding J.M. (2011) Tabu search - Application of Particle swarm optimization in reactive power optimization. The grid technology. (35): 129-1337.
[11] Xiu C.B. Lu L.F. (2010) Improved chaos optimization algorithm and its application in load distribution of power system. Power system protection and control. (38): 109-144.
[12] Jiang S. Gao G. (2006) Research on economic load scheduling of thermal power plant based on Particle Swarm optimization. Modern power. (23): 52-56.

[13] Li S.J. Zhou R.J. Zhou S.Y. (2014) Fuzzy self-correcting particle swarm optimization for environmental economic Load distribution in power system. Power system protection and control. (42): 15-21.