Five Phases Algorithm: A Novel Meta-heuristic Algorithm and Its Application on Economic Load Dispatch Problem

Xiaopeng Wang[1,2], Shu-Chuan Chu[1], Václav Snášel[2], Hisham A. Shehadeh[3], Jeng-Shyang Pan[1,4]*

[1] *College of Computer Science and Engineering, Shandong University of Science and Technology, China*
[2] *Faculty of Electrical Engineering and Computer Science, VŠB-Technical University of Ostrava, Czech Republic*
[3] *Departments of Computer Information System and Technology and Computer Science, Amman Arab University, Jordan*
[4] *Department of Information Management, Chaoyang University of Technology, Taiwan*
wangxp1993@163.com, scchu0803@gmail.com, vaclav.snasel@vsb.cz, h.shehadeh@aau.edu.jo, jengshyangpan@gmail.com

Abstract

A new meta-heuristic algorithm named the five phases algorithm (FPA) is presented in this paper. The proposed method is inspired by the five phases theory in traditional Chinese thought. FPA updates agents based on the generating and overcoming strategy as well as learning strategy from the agent with the same label. FPA has a simple structure but excellent performance. It also does not have any predefined control parameters, only two general parameters including population size and terminal condition are required. This provides flexibility to users to solve different optimization problems. For global optimization, 10 test functions from the CEC2019 test suite are used to evaluate the performance of FPA. The experimental results confirm that FPA is better than the 6 state-of-the-art algorithms including particle swarm optimization (PSO), grey wolf optimizer (GWO), multi-verse optimizer (MVO), differential evolution (DE), backtracking search algorithm (BSA), and slime mould algorithm (SMA). Furthermore, FPA is applied to solve the Economic Load Dispatch (ELD) from the real power system problem. The experiments give that the minimum cost of power system operation obtained by the proposed FPA is more competitive than the 14 counterparts. The source codes of this algorithm can be found in https://ww2.mathworks.cn/matlabcentral/fileexchange/118215-five-phases-algorithm-fpa.

Keywords: Meta-heuristic algorithm, Optimization problem, Five phases algorithm, Economic load dispatch

1 Introduction

Optimization problems are very general in academic and engineering fields. Thus, researchers need to develop optimization techniques and apply them to practical applications. Deterministic algorithms and meta-heuristic algorithms are two effective ways to solve optimization problems. Deterministic algorithms are also called conventional algorithms, which are limited by gradient descent, impose differentiability and convexity restrictions. For simple unimodal optimization problems, deterministic algorithms can obtain reliable accuracy and significant convergence speed while they lack randomness and easily fall into local optima for complex multimodal optimization problems. Different from deterministic algorithms, meta-heuristic algorithms as random optimization strategies do not impose any additional restrictions, which makes them more flexible to solve real-world optimization problems [1-2]. The meta-heuristic algorithms randomly generate agents in the search space to begin the optimization process. In each iteration, the agents are guided to move based on the cooperation of the agents. When the termination condition is met, the global optimal solution or approximate solution is output. In the last two decades, scholars' interest in meta-heuristic algorithms has been growing. At present, many meta-heuristic algorithms have been used successfully to solve the engineering optimization problems, such as image segmentation [3-5], parameter extraction of solar photovoltaic models [6-9], feature selection [10-11], and wireless sensor networks [12-13]. Meta-heuristic algorithms can be divided into four categories: evolution-based algorithms, swarm-based algorithms, physics-based algorithms, and human-based algorithms. They are mainly inspired by principles of natural evolution, social behavior of animal groups, physical principles, and human social behavior. Evolution-based algorithms include genetic algorithm (GA) [14], genetic programming (GP) [15], differential evolution (DE) [16], QUasi-Affine Transformation Evolutionary (QUATRE) [17] and backtracking search optimization algorithm (BSA) [18]. Swarm-based algorithm includes particle swarm optimization (PSO) [19], ant colony optimization (ACO) [20], cat swarm optimization (CSO) [21], grey wolf optimizer (GWO) [22] and slime mould algorithm (SMA) [23]. Physics-based algorithms include multi-verse optimizer (MVO) [24], gravitational search algorithm (GSA) [25], ray optimization (RO) [26], black hole (BH) [27] and central force optimization (CFO) [28]. Human-based algorithms include teaching-learning-based optimization (TLBO) [29], soccer league competition (SLC) [30], league championship algorithm (LCA) [31], seeker optimization algorithm (SOA) [32] and preaching optimization algorithm (POA) [33].

It is imperative to highlight the no free lunch (NFL) [34] theorem proposed by Wolpert and Macready in 1997. The NFL theory opens up the field of meta-heuristic algorithms for researchers. It logically verifies that there is no algorithm

*Corresponding Author: Jeng-Shyang Pan; E-mail: jengshyangpan@gmail.com
DOI: 10.53106/160792642023072404002

for solving all optimization problems. This indicates that a meta-heuristic algorithm can obtain accurate results on a specific set of problems, but the same algorithm may also show poor performance on optimization problems of different types and natures. Therefore, proposing novel meta-heuristic algorithms or improving existing algorithms are essential to solve optimization problems in different fields. Moreover, the adjustable parameters in meta-heuristic algorithms usually are divided into two classes: general parameters and special parameters. General parameters refer to the population size and terminal conditions (the maximum number of function evaluations or the maximum number of iterations) required by each meta-heuristic algorithm. Special parameters are also called control parameters, which reflect the features of each meta-heuristic algorithm. For example, the crossover rate of the differential evolution algorithm is a predefined constant in the range (0, 1), which controls the fraction of parameter values copied from the mutation vector. The difficulty is how to set special parameter values to solve different optimization problems. Proper parameter settings will give promising results, and vice versa. Therefore, when we contribute a new or improved algorithm, special parameters should be avoided as much as possible to achieve a parameter-free model. However, most current meta-heuristic algorithms still contain special parameters.

In recent years, human life's demand for electricity has increased, but the existing resources are limited. How to use electric energy effectively and rationally is a challenge to human development. Many scholars have paid attention to solving the urgent optimization problems of modern power systems. Economic load dispatch (ELD) is one of the important optimization problems in power system operation and planning. It solves how to reasonably distribute the power supply of distributed generations. Minimizing operating costs is the target of the economic load dispatch problem. How to properly plan the power output of the generating units to meet the load demand is to solve the optimization problem, but also must meet certain power system equality and inequality constraints. Today, many meta-heuristic optimization algorithms have been used to solve load economic dispatch, such as genetic algorithm (GA) [35], differential evolution (DE) [36-37], particle swarm optimization (PSO) [38-39], harmony search (HS) [40], grey wolf optimization (GWO) [41], firefly algorithm (FA) [42], grasshopper optimization algorithm (GOA) [43], search and rescue (SAR) [44], slime mould algorithm (SMA) [45], Arithmetic optimization algorithm (AOA) [46]. The above algorithms are useful for load economic dispatch, but the optimization operation cost is still very high, so it is necessary to propose a new competitive meta-heuristic algorithm in this field.

The above three reasons motivated this work to propose a novel evolution algorithm named five phases algorithm (FPA). The most notable feature is the simple structure of the proposed five phases algorithm, which is modeled based on generating and overcoming strategy and learning strategy from the agent with the same label. In addition, five phases algorithm does not include any special parameters, only population size and terminal condition parameters are required. This provides flexibility for people in solving different types of optimization problems. Furthermore, the performance of five phases algorithm is evaluated under the CEC2019 test suite. The experimental results verify the effectiveness of the proposed algorithm. Finally, the five phases algorithm is applied to solve the economic load dispatch problem, and the simulation results confirm that it can find more competitive results than its counterparts.

The main contributions included in this work are as follows:

(1) We propose a new meta-heuristic algorithm named five phases algorithm (FPA). FPA is modeled based on the generating and overcoming strategy as well as learning strategy from the agent with the same label.

(2) The CEC2019 test suite is used to evaluate the performance of FPA. The experimental results confirm that FPA is better than the 6 state-of-the-art algorithms.

(3) FPA is also applied to solve the economic load dispatch problem in the power system. The experiments give that the minimum cost of power system operation obtained by the proposed FPA is more competitive than the 14 counterparts.

The remainder of this paper is organized as follows:

Section 2 introduces the proposed five phases algorithm including inspiration and mathematical model. Section 3 presents the mathematical models to solve the economic load dispatch problem. In Section 4, the experimental results are given under the CEC2019 test suite and economic load dispatch problem. Finally, Section 5 concludes this paper.

2 Five Phases Algorithm

In this section, we will present the inspiration and mathematical model of five phases algorithm.

2.1 Inspiration

The five phases theory is a philosophical thought created by the ancient Chinese people. The ancient Chinese used metal, wood, water, fire, and earth five elements as the basis for the formation of all things in the universe and the changes in various natural phenomena. They relied on five phases theory to understand the world and believed in the unity of heaven and human. The five phases theory has been widely applied to account for existences and happenings in the conceptual domains of nature, human, and society. Its influence can still be felt in all walks of life in modern China [47-48].

Shangshu is a collection of documentary materials written in the Zhou dynasty. It is the earliest record found about the five phases theory. Wood, fire, earth, metal, and water are five basic concepts in the five phases theory. The ancient Chinese believed that the relationship between five phases was not isolated from each other, but closely intertwined. According to observations in daily life, the ancient Chinese found that, on the one hand, metal becomes liquid once melted, water is a necessary condition for the growth of trees, boring wood can create fire, burnt wood becomes ashes, and metallic ore exists in earth. On the other hand, metal-made tools can cut trees, wood-made tools can plow earth, earth-made dams can stop water, water can extinguish fire, and fire can melt metal. Therefore, ancient thinkers integrated five phases into a system and contributed to five phases theory. Wood generates fire, fire generates earth, and earth generates metal, metal generates water, and water generates wood. Wood overcomes earth, earth overcomes water, water overcomes fire, and fire overcomes metal, metal overcomes wood. The mutual generation and overcome interaction of five phases theory is drawn in Figure 1.

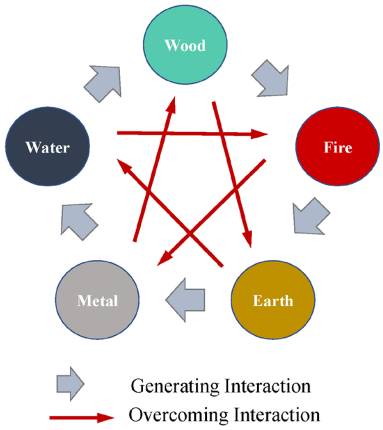


Figure 1. The mutual generation and overcome interaction of five phases theory

2.2 Mathematical Model

This subsection presents the two mathematical models based on the generating and overcoming strategy and learning strategy from the agent with the same label. The details are as follows.

2.2.1 Generating and Overcoming Strategy

Inspired by the five phases theory, the proposed generating and overcoming strategy updates the position of agents and completes the information exchange among agents in different phase labels. The specific operation is shown in Eq. (1).

$$X_{G+1}^{l,i,d} = X_G^{l,i,d} + r1 \times \left(X_G^{g,d} - X_G^{l,i,d} \right) + r2 \times \left(X_G^{o,d} - X_G^{l,i,d} \right), \quad \textbf{(1)}$$

where $X_G^{l,i,d}$ and $X_{G+1}^{l,i,d}$ denote the *dth* dimension of *ith* agent at the *Gth* and *(G + 1)th* generations and the label of *ith* agent is *l*. $X_G^{g,d}$ and $X_G^{o,d}$ mean the *dth* dimension of generating and overcoming agents of the *ith* agent with label *l* at the *Gth* generation, respectively. The parameters $r1$ and $r2$ denote random numbers in [0, 1]. For solving the minimization problem, assuming that the label of $X_G^{l,i,d}$ is wood, then $X_G^{g,d}$ represents the *dth* dimension of the generating agent, and the generating agent is the agent with the smallest fitness value in the water label. $X_G^{o,d}$ is the *dth* dimension of the overcoming agent, and the overcoming agent is the agent with the largest fitness value in the metal label. For solving the maximization problem, the $X_G^{g,d}$ and $X_G^{o,d}$ are respectively selected as the agent with the largest fitness and the agent with the smallest fitness value under the corresponding label. The agents with other labels are also updated based on the mutual generation and overcome relationship in the five phases theory.

2.2.2 Learning Strategy

The learning strategy presents the mutual communication mechanism between agents in the same label. The updated method of guiding agents is to learn from better agents with the same label. For the minimization problem, the updated method for the agent is as Eq. (2).

$$X_{G+1}^{l,i,d} = \begin{cases} X_G^{l,i,d} + r3 \times \left(X_G^{l,j,d} - X_G^{l,i,d} \right), & \text{if } f\left(X_G^{l,j}\right) < f\left(X_G^{l,i}\right) \\ X_G^{l,i,d} - r4 \times \left(X_G^{l,j,d} - X_G^{l,i,d} \right), & \text{otherwise} \end{cases}, \quad \textbf{(2)}$$

where $X_G^{l,j}$ represents the randomly selected *jth* agent with label *l*. $f\left(X_G^{l,i}\right)$ and $f\left(X_G^{l,j}\right)$ are the fitness values of $X_G^{l,i}$ and $X_G^{l,j}$. The parameters $r3$ and $r4$ are two random numbers in [0, 1]. If the fitness value of *jth* agent is less than *ith* agent, it indicates that the *jth* agent is superior to the *ith* agent, and *ith* agent will move towards the selected *jth* agent. Otherwise, the *ith* agent moves in the reverse direction.

Figure 2 shows the flowchart of five phases algorithm. First, initialize the optimization process: set two general parameters including the population size and the maximum number of iterations, then evaluate the fitness values of the agents, and select the current optimal agent. Further, randomly separate the agents into five phases. For wood, fire, earth, and metal, the subpopulation size is set to *floor* (population size/5), and the number of remaining agents is set to water. Eq. (1) and Eq. (2) are executed to evolve the agents based on the parameters *p* and *q*, where *p* and *q* are random numbers from 0 to 1. If *p* is less than *q*, then Eq. (1) is executed, otherwise, Eq. (2) is executed. Moreover, evaluate the fitness values of all agents and update the global optima. Finally, all agents are merged. If the current iteration number meets the termination condition, the global optima will be output. Otherwise, the agents will continue to be randomly separated to search the optima. Algorithm 1 gives the pseudo-code of five phases algorithm.

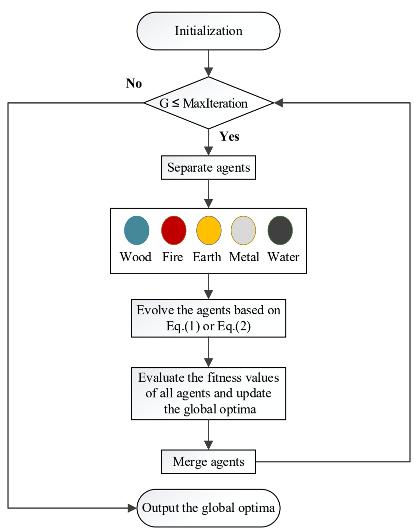


Figure 2. The flowchart of five phases algorithm

Algorithm 1. The pseudo-code of FPA algorithm

Initialization: population size ps, $MaxIteration$, evaluate the fitness values of all agents X, and calculate X_{gbest}.
While $G \leq MaxIteration$
  Randomly separate the agents into five phases, and their labels are wood, fire, earth, metal, and water. For wood, fire, earth, and metal, the size is set to $floor(ps/5)$, and the number of remaining agents is water.
  For $i = 1 : 5$
    Update agents by Eq. (1) or Eq. (2), and evaluate the fitness values of all agents
    If $f(X_{G+1}^{l,i}) < f(X_{G}^{l,i})$ // For the minimization problem
      $X_{G+1}^{l,i} = X_{G+1}^{l,i}$
    Else $X_{G+1}^{l,i} = X_{G}^{l,i}$
    End if
  End for
  Merge all the agents X
  $X_{gbest} = opt\{X\}$
  $G = G + 1$
End while
Output: The global optima X_{gbest} and $f(X_{gbest})$

3 Mathematical Model of Economic Load Dispatch

Economic load dispatch is a mathematical optimization problem to solve the output of generating units. The main purpose is to save the fuel cost consumed by the generating units, and it is subject to the equality and inequality constraints of the actual power system.

3.1 Objective Function

There are two generally used versions of the objective function for the economic load dispatch problem. The first version is that the objective function is modeled as a single quadratic function, which is shown in Eq.(3). The second version takes into account the valve point effect (VPE). Due to the sudden opening of the turbine intake valve, a wire drawing will occur, and a pulsation effect will be added to the consumption characteristic curve of the unit. Compared with the first version of the quadratic objective function, the second objective function adds a sinusoidal function with VPE to model the fuel cost of the generating units. The objective function of the second version is shown in Eq.(4).

$$Min\ F_{total} = \sum_{i=1}^{N} \left(a_i + b_i P_i + c_i P_i^2 \right), \tag{3}$$

$$Min\ F_{total} = \sum_{i=1}^{N} \left(a_i + b_i P_i + c_i P_i^2 + | e_i \sin\left(f_i \left(P_i^{\min} - P_i \right) \right) | \right), \tag{4}$$

where F_{total} represents the total fuel cost of the generating units, N is the number of generators, P_i is the electrical output power of ith generator, $P_i^{\min}$ is the lower limit of ith generator output power, a_i , b_i and c_i mean the fuel cost coefficients of ith generator, e_i and f_i are the loading coefficients of the valve points for ith generator.

3.2 Constraints

The economic load dispatch problem is subject to two constraints: generator output limits and power balance constraints. The constraints are as follows.

3.2.1 Generator Output Limits

The actual output power of each generator has upper and lower limits. When optimizing the objective function for fuel cost, the output power of each generator should be within the constrained interval.

$$P_i^{\min} \leq P_i \leq P_i^{\max}, \tag{5}$$

where $P_i^{\min}$ and $P_i^{\max}$ are the lower and upper limits of ith generator output power.

3.2.2 Power Balance Constraint

The power balance constraint of the system is that the output power of the generating units should be equal to the power demand plus the power loss on the transmission line network.

$$\sum_{i=1}^{N} P_i = P_D + P_L, \tag{6}$$

where P_i is the electrical output power of ith generator. P_D and P_L denote the total power demand and the power loss on the transmission line network, respectively. Furthermore, P_L can be calculated approximately by Kron's loss formula as follows.

$$P_L = \sum_{i=1}^{N} \sum_{j=1}^{N} P_i B_{ij} P_j + \sum_{i=1}^{N} B_{0i} P_i B_{00}, \tag{7}$$

where B_{ij}, B_{0i} and B_{00} denote the transmission loss coefficients.

4 Experimental Analysis

In this section, the CEC2019 test suite is firstly used to evaluate the performance of five phases algorithm, and then the proposed five phases algorithm is used to solve the economic load dispatch problem. The algorithms are coded in MATLAB. The experiments are tested on a personal laptop with Windows 10 OS, CPU of Intel(R) Core (TM) i7-8750H CPU @ 2.20 GHz 2.21 GHz, and 16 GB RAM.

4.1 Global Optimization

The CEC2019 test suite includes 10 benchmark functions is used to evaluate the proposed algorithm in this subsection.

The detailed benchmarks' definitions are given in [49]. We have shifted all the benchmark functions to the global minimum of "0" in the simulation. The proposed five phases algorithm is compared with the 6 outstanding algorithms including particle swarm optimization, grey wolf optimizer, multi-verse optimizer, differential evolution, backtracking search algorithm, and slime mould algorithm.

Considering the randomness of the meta-heuristic algorithm, many tests are performed. Each algorithm is independently run 51 times and the test results are recorded. Set the initial population size to 50, and the maximum of function evaluations is 5000 × dimension. Table 1 presents the special parameters of each comparison algorithm, which refer to the values of the corresponding references. The simulation results are evaluated under Tied rank [50] and Wilcoxon signed-rank [51] test statistics.

Table 1. Parameter settings of counterparts

Algorithm	Parameters settings
PSO	$V_{min} = -10$, $V_{max} = 10$, $w = [0.9, 0.4]$, c_1, $c_2 = 2$
GWO	$a = [2,0]$
MVO	$WEP_{min} = 0.2$, $WEP_{max} = 1$, $p = 0.6$
DE	$F = 0.7$, $C_r = 0.1$
BSA	$Mixrate = 1.00$
SMA	$z = 0.03$

Table 2. Experimental simulations of 7 meta-heuristic algorithms under the CEC2019 test suite

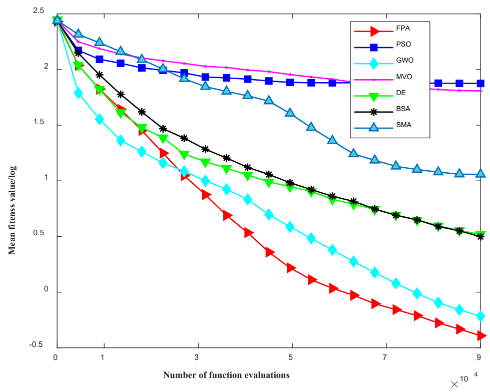
Function		PSO	GWO	MVO	DE	BSA	SMA	FPA
F1	Mean	2.972E+07	8.068E+03	8.486E+05	1.631E+06	1.039E+05	**0.000E+00**	7.495E+04
	Std	4.003E+07	2.533E+04	6.348E+05	9.132E+05	7.560E+04	**0.000E+00**	6.915E+04
	TR	7	2	5	6	4	1	3
	Mark	-	+	-	-	-	+	/
F2	Mean	7.497E+03	2.690E+02	3.982E+02	9.876E+02	2.096E+02	**3.882E+00**	1.862E+02
	Std	3.239E+03	1.789E+02	1.239E+02	2.237E+02	7.199E+01	**2.599E-01**	7.931E+01
	TR	7	4	5	6	3	1	2
	Mark	-	-	-	-	=	+	/
F3	Mean	6.520E+00	8.053E-01	6.095E+00	1.681E+00	1.646E+00	2.881E+00	**6.762E-01**
	Std	2.108E+00	7.376E-01	2.361E+00	**4.048E-01**	5.140E-01	2.594E+00	5.288E-01
	TR	7	2	6	4	3	5	1
	Mark	-	=	-	-	-	-	/
F4	Mean	3.293E+01	1.264E+01	1.723E+01	5.626E+00	8.659E+00	1.436E+01	**5.396E+00**
	Std	1.314E+01	7.063E+00	8.412E+00	**1.313E+00**	2.249E+00	5.408E+00	3.248E+00
	TR	7	4	6	2	3	5	1
	Mark	-	-	-	-	-	-	/
F5	Mean	1.402E-01	5.041E-01	3.058E-01	6.938E-02	5.563E-02	2.545E-01	**1.232E-02**
	Std	6.587E-02	4.078E-01	1.299E-01	3.087E-02	2.770E-02	9.375E-02	**1.101E-02**
	TR	4	7	6	3	2	5	1
	Mark	-	-	-	-	-	-	/
F6	Mean	3.385E+00	1.502E+00	1.718E+00	**2.046E-02**	1.058E+00	3.335E+00	1.259E-01
	Std	1.483E+00	1.078E+00	1.382E+00	**7.987E-02**	5.223E-01	1.570E+00	2.454E-01
	TR	7	4	5	1	3	6	2
	Mark	-	-	-	+	+	-	/
F7	Mean	1.137E+03	6.426E+02	7.376E+02	**2.378E+02**	4.503E+02	6.679E+02	8.498E+02
	Std	2.976E+02	2.542E+02	3.055E+02	1.685E+02	**9.931E+01**	2.402E+02	3.109E+02
	TR	7	3	5	1	2	4	6
	Mark	-	+	+	+	-	+	/
F8	Mean	3.128E+00	2.552E+00	2.687E+00	**2.182E+00**	2.673E+00	2.660E+00	2.564E+00
	Std	4.741E-01	4.404E-01	5.749E-01	2.952E-01	**2.387E-01**	4.887E-01	2.710E-01
	TR	7	2	6	1	5	4	3
	Mark	-	=	=	+	-	=	/
F9	Mean	1.850E-01	1.432E-01	2.116E-01	2.022E-01	2.132E-01	2.014E-01	**1.294E-01**
	Std	1.004E-01	5.393E-02	6.649E-02	3.411E-02	4.453E-02	6.354E-02	**3.055E-02**
	TR	3	2	6	5	7	4	1
	Mark	-	=	-	-	-	-	/
F10	Mean	**1.930E+01**	1.996E+01	2.003E+01	1.992E+01	2.008E+01	1.970E+01	2.038E+01
	Std	3.508E+00	2.295E+00	**4.108E-02**	1.154E+00	4.250E-02	2.802E+00	9.412E-02
	TR	1	4	5	3	6	2	7
	Mark	+	=	+	+	+	+	/
Over-Rank		7	3	6	2	5	4	1
-/=/+		9/0/1	4/4/2	7/1/2	6/0/4	7/1/2	5/1/4	compared

Table 2 presents the experimental results of the 7 meta-heuristic algorithms under the CEC2019 test suite. The mean value and standard deviation obtained by each algorithm are denoted as "Mean" and "Std", respectively. The best mean and standard deviation have been shown in bold. "TR" indicates the tied rank of the test algorithms, and "Mark" means the wilcoxon signed-rank test. The mark "–" denotes that the proposed FPA is better than the compared algorithm. The mark "=" denotes that the proposed FPA is similar to the compared algorithm. The mark "+" denotes that the proposed FPA is worse than the compared algorithm. For wilcoxon signed-rank test, the level of significance α is set to 0.05 in the simulation.

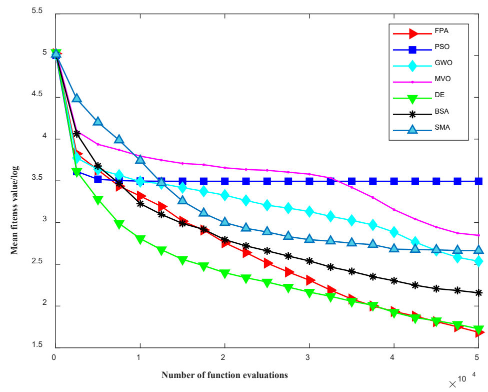
For example, when the benchamark function F5 is solved, the proposed FPA algorithm can search for the smallest mean value 1.232E-02 and standard deviation 1.101E-02 among the 7 algorithms. Tired rank statistic give the rank of 7 algorithm, the proposed FPA is first, BSA is second, DE is third, PSO is fourth, SMA is fifth, MVO is sixth, and GWO is seventh. Wilcoxon signed-rank test statistic indicates that the proposed FPA can obtan better performance than the 6 compared algorithms under the F5 benchmark function. For the 10 test functions, the ranking of FPA is third, second, first, first, first, second, sixth, third, first, and seventh. The overall performance under the CEC2019 test suite is the proposed FPA is ranked first, DE is ranked second, GWO is ranked third, SMA is ranked fourth, BSA is ranked fifth, MVO is ranked sixth, and PSO is ranked seventh. Compared to PSO, FPA achieves 9 better performances, 0 similar performances, and 1 worse performance. Compared to GWO, FPA achieves 4 better performances, 4 similar performances, and 2 worse performances. Similarly, the comparison results with MVO, DE, BSA and SMA can be found in Table 2. The average time of the proposed FPA algorithm and the comparison algorithms are shown in Table 3. Figure. 3 plots the convergence curves of mean value for 7 algorithms. As shown in the figure, the proposed FPA outperforms the 6 compared algorithms under functions F3, F4, F5 and F9. Therefore, the proposed FPA algorithm is competitive and is superior to the PSO, GWO, MVO, DE, BSA and SMA algorithms under the CEC2019 test suite.

Table 3. Average time under the CEC2019 test suite

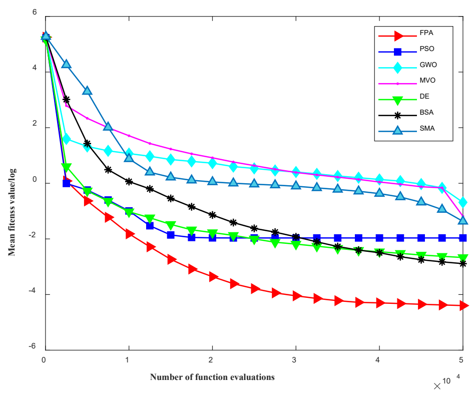
Algorithm	Average time (s)
PSO	0.198
GWO	0.503
MVO	0.575
DE	0.363
BSA	0.296
SMA	1.816
FPA	0.834



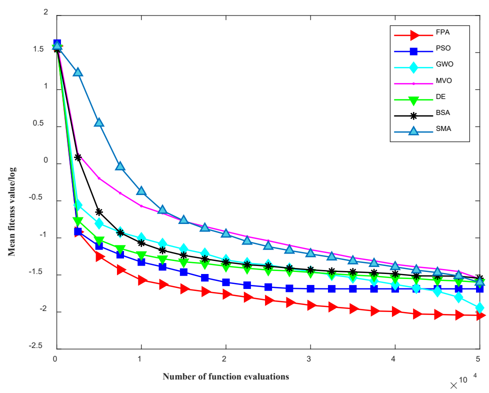
(a) F3



(b) F4



(c) F5



(d) F9

Figure 3. The convergence curves obtained by FPA and 6 compared algorithms for the selected functions

4.2 Economic Load Dispatch Problem

In this subsection, the proposed FPA is applied to solve the economic load dispatch problem in practical application.

For solving the economic load dispatch problem, we tested two cases including 20-unit with transmission line loss, P_D = 2500 MW and 40-unit with valve point effect, P_D = 10500 MW. The first case is 20 generating units, considering the line loss, the power demand is 2500MW. The second case is 40 generating units, considering the valve point effect, the power demand is 10500MW. The detailed data of the simulation are shown in the appendix. The parameters of the two test models refer to the reference [52] and [53]. The initial parameters are as follows: the population size is set 50, the maximum of function evaluations is set to 15000 for 20-unit with transmission line loss and 50000 for 40-unit with valve point effect refer to the corresponding references, respectively. To fairly compare the experimental results, each algorithm is set to run 31 times.

For the 20-unit with transmission line loss, Table 4 shows the best results searched by the 7 algorithms and the solved output power for 20 generators. The minimal electrical power loss and minimal fuel cost have been highlighted in bold. The proposed FPA obtains the smallest electrical power loss 89.9054 MW and obtains the smallest fuel cost 62443.9509 $/h. To evaluate the performance of the proposed FPA algorithm, we compare the minimum fuel cost results obtained by the other 9 counterparts. Figure 4 shows the minimal cost obtained by 10 algorithms. The cost value of PSO is 62448.252 $/h, the cost value of GWO is 62443.4563 $/h, the cost value of MVO is 62438.1534 $/h, the cost value of DE is 62469.2092 $/h, the cost value of BSA is 62473.7261 $/h, the cost value of SMA is 62446.4223 $/h, the cost value of BBO is 62456.7793 $/h, the cost value of LI is 62456.6391 $/h, the cost value of HM is 62443.6341 $/h, the cost value of ALO is 62456.6331 $/h. Therefore, the experimental result obtained by FPA outperforms other comparison algorithms under the 20-unit with transmission line loss.

For the 40-unit with valve point effect, Table 5 shows the best results obtained by the 7 algorithms and the solved output power of the 40 generators. The minimal fuel cost has been highlighted in bold. The proposed FPA obtains the smallest fuel cost at 121474.4714 $/h. To evaluate the performance of the proposed FPA algorithm, we compare the minimum fuel cost results obtained by the other 9 counterparts. Figure 5 gives the minimal cost obtained by 10 algorithms. The cost value of PSO is 126541.0877 $/h, the cost value of GWO is 123522.7778 $/h, the cost value of MVO is 122395.227 $/h, the cost value of DE is 122577.6866 $/h, the cost value of BSA is 125758.8805 $/h, the cost value of SMA is 121872.2817 $/h, the cost value of PPSO is 125503.09 $/h, the cost value of SSA is 123565.75 $/h, the cost value of MPA is 123180.98 $/h, the cost value of MGMPA is 122634.69 $/h. Therefore, the experimental result obtained by FPA outperforms other comparison algorithms under the 40-unit with valve point effect. Table 6 shows the detailed results (Best, Mean, Median, Worst, Std, Average time) of FPA in two cases. The simulation experiment proves that the proposed FPA is an effective algorithm to solve the economic load dispatch problem and it is superior to the 14 algorithms, including PSO, GWO, MVO, DE, BSA, SMA, BBO, LI, HM, ALO, PPSO, SSA, MPA and MGMPA.

Table 4. The experimental results of 7 algorithms for the 20-unit with transmission line loss, P_D = 2500 MW

Unit	PSO	GWO	MVO	DE	BSA	SMA	FPA
1	545.442	492.693	482.109	541.7115	467.7405	466.17	490.2405
2	200	161.888	171.506	140.0765	159.413	136.8815	155.1275
3	138.551	122.879	115.841	118.07	120.257	117.011	111.9605
4	91.5305	98.8235	104.4515	72.59	136.0445	117.863	84.158
5	106.2705	122.9795	96.8545	100.732	110.2239	107.9964	117.8799
6	77.0568	58.2624	94.128	79.2872	56.4176	55.9616	80.3176
7	96.993	85.851	117.406	97.153	83.439	96.542	104.138
8	125.195	102.747	113.373	100.477	110.464	104.263	130.182
9	72.6065	116.78	108.911	114.0815	123.0245	120.2945	110.486
10	75.0144	106.8564	92.0736	79.5264	86.0304	85.8132	110.0604
11	156.144	165.084	148.494	140.998	172.24	159.118	141.87
12	276.1785	294.27	298.148	262.574	297.938	303.125	289.58
13	128.3752	126.862	98.8888	146.4256	105.0916	128.5732	117.5608
14	38.909	65.397	60.3326	54.6863	91.9169	63.406	67.0558
15	138.1136	177.008	164.832	181.2384	146.7792	181.0496	165.7216
16	42.8336	35.6528	38.9342	40.2752	34.9946	36.8426	38.4338
17	65.9513	43.2264	51.6128	75.5862	76.16205	55.1119	65.6059
18	89.7726	91.1712	76.0989	50.2698	74.1162	75.3789	58.7289
19	72.7608	93.8264	83.5768	115.208	87.4272	96.7128	86.9648
20	52.4294	31.21751	72.8008	80.1599	52.9502	82.6904	63.4334
P_L	90.1277	93.47511	90.3725	91.1265	92.67035	90.8046	**89.5054**
Cost	62448.252	62443.4563	62438.1534	62469.2092	62473.7261	62446.4223	**62433.9509**

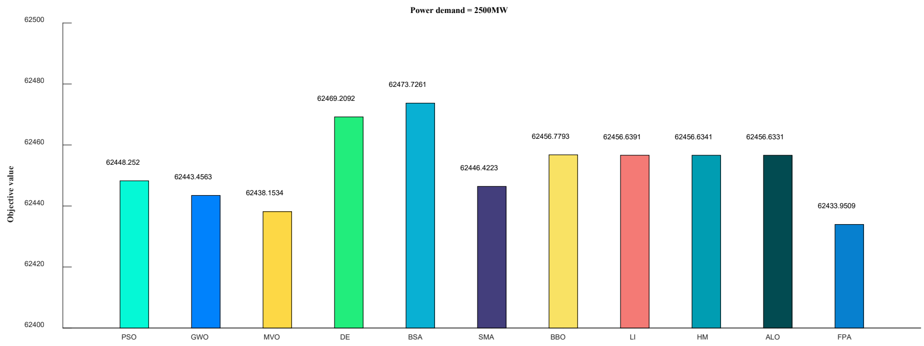


Figure 4. The minimal cost obtained by FPA and the compared algorithms for the 20 units

Table 5. The experimental results of 7 algorithms for 40-unit with valve point effect, P_D = 10500 MW

Unit	PSO	GWO	MVO	DE	BSA	SMA	FPA
1	72.93924	111.0851	113.1989	110.8894	65.08464	112.0461	112.26996
2	100.4686	83.30154	112.7192	112.883	57.90864	111.8612	111.52506
3	71.1984	68.2158	99.3432	100.7388	77.8788	99.3618	97.392
4	141.9916	136.8117	137.6818	181.5212	146.6281	178.3906	179.7854
5	94.04	93.7255	93.7945	90.8175	83.801	88.436	91.8315
6	116.8736	68.61109	136.6311	105.6574	107.137	105.4832	139.99928
7	300	299.2666	272.2923	259.815	296.2228	299.9696	260.5389
8	281.4788	285.9222	287.7867	290.1644	261.1871	288.1085	284.70945
9	284.9207	289.242	294.8388	287.2373	222.3395	284.9817	284.7738
10	130.0212	130	132.234	197.8113	205.1553	130	130.1285659
11	375	170.8114	168.7601	170.4404	296.9017	167.7063	94
12	315.8888	98.25827	170.3674	94.45983	232.2351	167.8524	169.04386
13	394.31	394.5913	215.6525	218.0563	299.4613	125.0117	125.054585
14	214.7188	305.495	304.6438	395.57	309.5938	304.5163	394.34
15	394.28	394.1338	394.3138	396.3913	389.6788	304.55	394.28
16	471.62	394.2763	394.6213	387.5975	393.0838	483.41	394.29875
17	500	489.6652	489.9368	495.9176	489.094	489.4356	489.4132
18	500	493.252	491.9248	490.1076	491.6056	489.318	489.346
19	518.698	516.622	512.5472	513.9702	510.3666	511.1766	511.27516
20	421.5394	511.8758	514.4476	509.8245	514.4876	513.0924	511.4076
21	550	547.4041	527.2494	518.0971	536.5468	523.3541	523.45472
22	523.3926	532.8646	523.7685	528.0871	532.909	523.3718	523.41032
23	523.2801	531.7486	525.8494	523.4932	530.8044	523.3126	523.33928
24	521.6906	531.2099	524.917	523.9994	510.6942	523.2238	523.42808
25	435.1964	524.8696	527.3294	519.1538	517.8485	523.2209	523.53168
26	550	525.0827	532.3643	525.953	521.5929	523.0788	523.29784
27	10	10.40449	10.59227	11.89518	10.06959	10	10.0602028
28	10	15.89918	11.39658	10.5409	10.67547	10.00002	10.0177548
29	10	10.00342	11.11933	13.23288	10.16806	10.01565	10.00728
30	82.0235	88.6845	74.222	96.571	83.617	92.6425	87.823
31	151.6071	189.9155	189.7491	168.6228	188.9899	189.9974	190
32	160.4081	190	189.9805	167.0225	190	189.9961	189.9259
33	172.7737	190	190	186.7474	188.2424	189.9935	189.9714
34	127.5573	166.549	170.894	167.5049	181.9721	199.3829	199.978
35	150.137	182.6541	164.7978	164.9815	162.3052	199.9912	165.5161
36	200	170.3748	167.913	172.5792	136.1802	193.9665	199.9637
37	89.02455	25.01844	104.7521	99.14465	57.35185	90.26045	109.6158
38	91.09005	109.4254	105.4423	91.7488	104.628	109.9465	109.9864
39	110	109.7518	97.1276	91.5873	57.5601	108.2932	109.99575
40	331.8344	513.0154	512.8028	509.5565	518.5255	511.3368	511.29672
Cost	126541.0877	123522.8	122395.2	122577.7	125758.9	121872.3	121474.4714

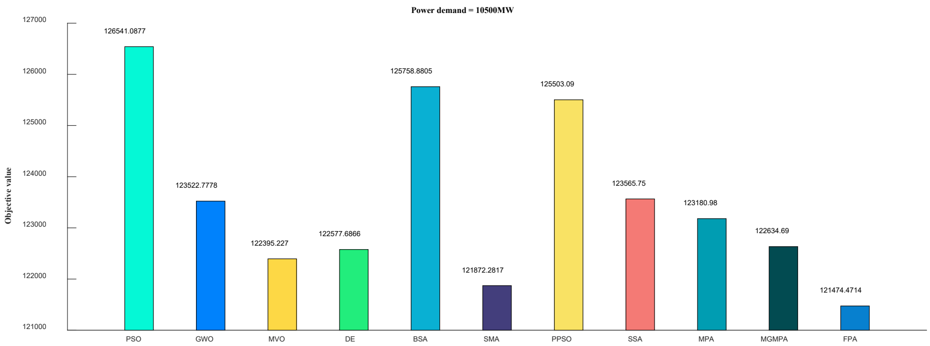


Figure 5. The minimal cost obtained by FPA and the compared algorithms for the 40 units

Table 6. The cost simulation results of FPA for 20 units and 40 units

Case	Best	Mean	Median	Worst	Std	Average time (s)
20 units	62433.9509	62467.3967	62466.2986	62506.1205	15.7884	0.4196
40 units	121474.4714	121962.16	121866.8977	123250.2606	390.1537	1.5481

5 Conclusion

This paper proposes a new five phases algorithm (FPA) inspired by the five phases theory. FPA updates agents based on two models: generating and overcoming strategy, and learning strategy from the agent with the same label. The CEC2019 test suite and economic load dispatch problem are used to evaluate the proposed FPA. For the CEC2019 test suite, the tied rank and wilcoxon signed-rank test statistics prove that FPA outperforms the outstanding PSO, GWO, MVO, DE, BSA, and SMA algorithms. For solving the economic load dispatch problem, FPA is also superior to the 14 state-of-the-art algorithms. The simulation results indicate that the proposed FPA is a competitive algorithm for solving optimization problems.

In the next work, the proposed FPA will be used to solve more actual optimization problems.

Appendix

See Tables A.1–A.3.

Table A.1. *B* loss matrix values for 20-unit with transmission line loss, $P_D = 2500$ MW

B	20-unit																			
B =	8.7	0.43	-4.61	0.36	0.32	-0.66	0.96	-1.6	0.8	-0.1	3.6	0.64	0.79	2.1	1.7	0.8	-3.2	0.7	0.48	-0.7
(1e – 5)	0.43	8.3	-0.97	0.22	0.75	-0.28	5.04	1.7	0.54	7.2	-0.28	0.98	-0.46	1.3	0.8	-0.2	0.52	-1.7	0.8	0.2
	-4.61	-0.97	9	-2	0.63	3	1.7	-4.3	3.1	-2	0.7	-0.77	0.93	4.6	-0.3	4.2	0.38	0.7	-2	3.6
	0.36	0.22	-2	5.3	0.47	2.62	-1.96	2.1	0.67	1.8	-0.45	0.92	2.4	7.6	-0.2	0.7	-1	0.86	1.6	0.87
	0.32	0.75	0.63	0.47	8.6	-0.8	0.37	0.72	-0.9	0.69	1.8	4.3	-2.8	-0.7	2.3	3.6	0.8	0.2	-3	0.5
	-0.66	-0.28	3	2.62	-0.8	11.8	-4.9	0.3	3	-3	0.4	0.78	6.4	2.6	-0.2	2.1	-0.4	2.3	1.6	-2.1
	0.96	5.04	1.7	-1.96	0.37	-4.9	8.24	-0.9	5.9	-0.6	8.5	-0.83	7.2	4.8	-0.9	-0.1	1.3	0.7	1.9	1.3
	-1.6	1.7	-4.3	2.1	0.72	0.3	-0.9	1.2	-0.96	0.56	1.6	0.8	-0.4	0.23	0.75	-0.56	0.8	-0.3	5.3	0.8
	0.8	0.54	3.1	0.67	-0.9	3	5.9	-0.96	0.93	-0.3	6.5	2.3	2.6	0.58	-0.1	0.23	-0.3	1.5	0.74	0.7
	-0.1	7.2	-2	1.8	0.69	-3	-0.6	0.56	-0.3	0.99	-6.6	3.9	2.3	-0.3	2.8	-0.8	0.38	1.9	0.47	-0.26
	3.6	-0.28	0.7	-0.45	1.8	0.4	8.5	1.6	6.5	-6.6	10.7	5.3	-0.6	0.7	1.9	-2.6	0.93	-0.6	3.8	-1.5
	0.64	0.98	-0.77	0.92	4.3	0.78	-0.83	0.8	2.3	3.9	5.3	8	0.9	2.1	-0.7	5.7	5.4	1.5	0.7	0.1
	0.79	-0.46	0.93	2.4	-2.8	6.4	7.2	-0.4	2.6	2.3	-0.6	0.9	11	0.87	-1	3.6	0.46	-0.9	0.6	1.5
	2.1	1.3	4.6	7.6	-0.7	2.6	4.8	0.23	0.58	-0.3	0.7	2.1	0.87	3.8	0.5	-0.7	1.9	2.3	-0.97	0.9
	1.7	0.8	-0.3	-0.2	2.3	-0.2	-0.9	0.75	-0.1	2.8	1.9	-0.7	-1	0.5	11	1.9	-0.8	2.6	2.3	-0.1
	0.8	-0.2	4.2	0.7	3.6	2.1	-0.1	-0.56	0.23	-0.8	-2.6	5.7	3.6	-0.7	1.9	10.8	2.5	-1.8	0.9	-2.6
	-3.2	0.52	0.38	-1	0.8	-0.4	1.3	0.8	-0.3	0.38	0.93	5.4	0.46	1.9	-0.8	2.5	8.7	4.2	-0.3	0.68
	0.7	-1.7	0.7	0.86	0.2	2.3	0.76	-0.3	1.5	1.9	-0.6	1.5	-0.9	2.3	2.6	-1.8	4.2	2.2	0.16	-0.3
	0.48	0.8	-2	1.6	-3	1.6	1.9	5.3	0.74	0.47	3.8	0.7	0.6	-0.97	2.3	0.9	-0.3	0.16	7.6	0.69
	-0.7	0.2	3.6	0.87	0.5	-2.1	1.3	0.8	0.7	-0.26	-1.5	0.1	1.5	0.9	-0.1	-2.6	0.68	-0.3	0.69	7
B_0 =	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B_{00} =																				

Table A.2. Data of 20-unit with transmission line loss, P_D = 2500 MW

Unit	a_i	b_i	c_i	P^{min}	P^{max}	Unit	a_i	b_i	c_i	P^{min}	P^{max}
1	0.00068	18.2	1000	150	600	11	0.005	16.69	800	100	300
2	0.00071	19.3	970	50	200	12	0.003	16.76	970	150	500
3	0.0065	19.8	600	50	200	13	0.009	17.36	900	40	160
4	0.005	19.1	700	50	200	14	0.005	18.7	700	20	130
5	0.00738	18.1	420	50	160	15	0.004	18.7	450	25	185
6	0.00612	19.3	360	20	100	16	0.071	14.26	370	20	80
7	0.0079	17.1	490	25	125	17	0.009	19.14	480	30	85
8	0.00813	18.9	660	50	150	18	0.007	18.92	680	30	120
9	0.00522	18.3	765	50	200	19	0.006	18.47	700	40	120
10	0.00573	18.9	770	30	150	20	0.008	19.79	850	30	100

Table A.3. Data of 40-unit with valve point effect, P_D = 10500 MW

Unit	a_i	b_i	c_i	e_i	f_i	P^{min}	P^{max}	Unit	a_i	b_i	c_i	e_i	f_i	P^{min}	P^{max}
1	0.0069	6.73	94.705	100	0.084	36	114	21	0.00298	6.63	785.96	300	0.035	254	550
2	0.0069	6.73	94.705	100	0.084	36	114	22	0.00298	6.63	785.96	300	0.035	254	550
3	0.02028	7.07	309.54	100	0.084	60	120	23	0.00284	6.66	794.53	300	0.035	254	550
4	0.00942	8.18	369.03	150	0.063	80	190	24	0.00284	6.66	794.53	300	0.035	254	550
5	0.0114	5.35	148.89	120	0.077	47	97	25	0.00277	7.1	801.32	300	0.035	254	550
6	0.01142	8.05	222.33	100	0.084	68	140	26	0.00277	7.1	801.32	300	0.035	254	550
7	0.00357	8.03	278.71	200	0.042	110	300	27	0.52124	3.33	1055.1	120	0.077	10	150
8	0.00492	6.99	391.98	200	0.042	135	300	28	0.52124	3.33	1055.1	120	0.077	10	150
9	0.00573	6.6	455.76	200	0.042	135	300	29	0.52124	3.33	1055.1	120	0.077	10	150
10	0.00605	12.9	722.82	200	0.042	130	300	30	0.0114	5.35	148.89	120	0.077	47	97
11	0.00515	12.9	635.2	200	0.042	94	375	31	0.0016	6.43	222.92	150	0.063	60	190
12	0.00569	12.8	654.69	200	0.042	94	375	32	0.0016	6.43	222.92	150	0.063	60	190
13	0.00421	12.5	913.4	300	0.035	125	500	33	0.0016	6.43	222.92	150	0.063	60	190
14	0.00752	8.84	1760.4	300	0.035	125	500	34	0.0001	8.95	107.87	200	0.042	90	200
15	0.00708	9.15	1728.3	300	0.035	125	500	35	0.0001	8.62	116.58	200	0.042	90	200
16	0.00708	9.15	1728.3	300	0.035	125	500	36	0.0001	8.62	116.58	200	0.042	90	200
17	0.00313	7.97	647.85	300	0.035	220	500	37	0.0161	5.88	307.45	80	0.098	25	110
18	0.00313	7.95	649.69	300	0.035	220	500	38	0.0161	5.88	307.45	80	0.098	25	110
19	0.00313	7.97	647.83	300	0.035	242	550	39	0.0161	5.88	307.45	80	0.098	25	110
20	0.00313	7.97	647.81	300	0.035	242	550	40	0.00313	7.97	647.83	300	0.035	242	550

Acknowledgments

This work is supported by the National Natural Science Foundation of China (61872085), the Natural Science Foundation of Fujian Province (2018J01638) and the Fujian Provincial Department of Science and Technology (2018Y3001).

References

[1] S. Ruder, *An overview of gradient descent optimization algorithms*, June, 2017. https://arxiv.org/abs/1609.04747

[2] S. C. Chu, H. C. Huang, J. F. Roddick, J. S. Pan, Overview of algorithms for swarm intelligence, *International Conference on Computational Collective Intelligence*, Gdynia, Poland, 2011, pp. 28-41.

[3] P. Mesejo, O. Ibáñez, O. Cordón, S. Cagnoni, A survey on image segmentation using metaheuristic-based deformable models: state of the art and critical analysis, *Applied Soft Computing*, Vol. 44, pp. 1-29, July, 2016.

[4] X. Wang, J. S. Pan, S. C. Chu, A parallel multi-verse optimizer for application in multilevel image segmentation, *IEEE Access*, Vol. 8, pp. 32018-32030, February, 2020.

[5] S. Sarkar, S. Das, S. S. Chaudhuri, A multilevel color image thresholding scheme based on minimum cross entropy and differential evolution, *Pattern Recognition Letters*, Vol. 54, pp. 27-35, March, 2015.

[6] Y. Liu, A. A. Heidari, X. Ye, G. Liang, H. Chen, C. He, Boosting slime mould algorithm for parameter identification of photovoltaic models, *Energy*, Vol. 234, Article No. 121164, November, 2021.

[7] S. Jiao, G. Chong, C. Huang, H. Hu, M. Wang, A. A. Heidari, H. Chen, X. Zhao, Orthogonally adapted harris hawks optimization for parameter estimation

of photovoltaic models, *Energy*, Vol. 203, Article No. 117804, July, 2020.

[8] X. Chen, K. Yu, W. Du, W. Zhao, G. Liu, Parameters identification of solar cell models using generalized oppositional teaching learning based optimization, *Energy*, Vol. 99, pp. 170-180, March, 2016.

[9] X. Wang, S. C. Chu, V. Snášel, L. Kong, J. S. Pan, H. A. Shehadeh, A two-phase quasi-affine transformation evolution with feedback for parameter identification of photovoltaic models, *Applied Soft Computing*, Vol. 113, Article No. 107978, December, 2021.

[10] P. Hu, J. S. Pan, S. C. Chu, Improved binary grey wolf optimizer and its application for feature selection, *Knowledge-Based Systems*, Vol. 195, Article No. 105746, May, 2020.

[11] G. I. Sayed, A. Tharwat, A. E. Hassanien, Chaotic dragonfly algorithm: An improved metaheuristic algorithm for feature selection, *Applied Intelligence*, Vol. 49, No. 1, pp. 188-205, January, 2019.

[12] J. S. Pan, L. Kong, T. W. Sung, P. W. Tsai, V. Snášel, A clustering scheme for wireless sensor networks based on genetic algorithm and dominating set, *Journal of Internet Technology*, Vol. 19, No. 4, pp. 1111-1118, July, 2018.

[13] T. T. Nguyen, J. S. Pan, T. K. Dao, An improved flower pollination algorithm for optimizing layouts of nodes in wireless sensor network, *IEEE Access*, Vol. 7, pp. 75985-75998, June, 2019.

[14] J. Holland, *Adaptation in natural and artificial systems*, Ann Arbor, MI: University of Michigan Press, 1975.

[15] J. Koza, J. Rice, Automatic programming of robots using genetic programming, *Proceedings of the tenth national conference on artificial intelligence*, San Jose, California, America, 1992, pp. 194-207.

[16] K. Price, R. M. Storn, J. A. Lampinen, *Differential evolution: a practical approach to global optimization*, Springer Science & Business Media, 2006.

[17] Z. Meng, J. S. Pan, H. Xu, Quasi-affine transformation evolutionary (quatre) algorithm: a cooperative swarm based algorithm for global optimization, *Knowledge-Based Systems*, Vol. 109, pp. 104-121, October, 2016.

[18] P. Civicioglu, Backtracking search optimization algorithm for numerical optimization problems, *Applied Mathematics and Computation*, Vol. 219, No. 15, pp. 8121-8144, April, 2013.

[19] J. Kennedy, R. Eberhart, Particle swarm optimization, *Proceedings of ICNN'95-International Conference on Neural Networks*, Perth, WA, Australia, 1995, pp. 1942-1948.

[20] M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization, *IEEE Computational Intelligence Magazine*, Vol. 1, No. 4, pp. 28-39, November, 2006.

[21] S. C. Chu, P. W. Tsai, J. S. Pan, Cat swarm optimization, *Pacific Rim international conference on artificial intelligence*, Guilin China, 2006, pp. 854-858.

[22] S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey wolf optimizer, *Advances in Engineering Software*, Vol. 69, pp. 46-61, March, 2014.

[23] S. Li, H. Chen, M. Wang, A. A. Heidari, S. Mirjalili, Slime mould algorithm: A new method for stochastic optimization, *Future Generation Computer Systems*, Vol. 111, pp. 300-323, October, 2022.

[24] S. Mirjalili, S. M. Mirjalili, A. Hatamlou, Multi-verse optimizer: a nature-inspired algorithm for global optimization, *Neural Computing and Applications*, Vol. 27, No. 2, pp. 495-531, February, 2016.

[25] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: a gravitational search algorithm, *Information Sciences*, Vol. 179, No. 13, pp. 2232-2248, June, 2009.

[26] A. Kaveh, M. Khayatazad, A new meta-heuristic method: ray optimization, *Computers & structures*, Vol. 112-113, pp. 283-294, December, 2012.

[27] A. Hatamlou, Black hole: A new heuristic optimization approach for data clustering, *Information sciences*, Vol. 222, pp. 175-184, February, 2013.

[28] R. A. Formato, Central force optimization, *Prog Electromagn Res*, Vol. 777, No. 1, pp. 425-491, 2007.

[29] R. V. Rao, V. J. Savsani, D. P. Vakharia, Teaching-learning-based optimization: an optimization method for continuous non-linear large scale problems, *Information sciences*, Vol. 183, No. 1, pp. 1-15, January, 2012.

[30] N. Moosavian, B. K. Roodsari, Soccer league competition algorithm: A novel meta-heuristic algorithm for optimal design of water distribution networks, *Swarm and Evolutionary Computation*, Vol. 17, pp. 14-24, August, 2014.

[31] A. H. Kashan, League championship algorithm: a new algorithm for numerical function optimization, *2009 International Conference of Soft Computing and Pattern Recognition*, Malacca, Malaysia, 2009, pp. 43-48.

[32] C. Dai, Y. Zhu, W. Chen, Seeker optimization algorithm, *International Conference on Computational and Information Science*, Guangzhou, China, 2006, pp. 167-176.

[33] D. Wei, Z. Wang, L. Si, C. Tan, Preaching-inspired swarm intelligence algorithm and its applications, *Knowledge-Based Systems*, Vol. 211, Article No. 106552, January, 2021.

[34] D. H. Wolpert, W. G. Macready, No free lunch theorems for optimization, *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, pp. 67-82, April, 1997.

[35] C. L. Chiang, Improved genetic algorithm for power economic dispatch of units with valve-point effects and multiple fuels, *IEEE Transactions on Power Systems*, Vol. 20, No. 4, pp. 1690-1699, November, 2005.

[36] L. S. Coelho, V. C. Mariani, Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect, *IEEE Transactions on Power Systems*, Vol. 21, No. 2, pp. 989-996, May, 2006.

[37] Q. Zhang, D. Zou, N. Duan, X. Shen, An adaptive differential evolutionary algorithm incorporating multiple mutation strategies for the economic load dispatch problem, *Applied Soft Computing*, Vol. 78 pp. 641-669, May, 2019.

[38] J. B. Park, K. S. Lee, J. R. Shin, K. Y. Lee, A particle swarm optimization for economic dispatch with nonsmooth cost functions, *IEEE Transactions on Power Systems*, Vol. 20, No. 1, pp. 34-42, February, 2005.

[39] M. Gholamghasemi, E. Akbari, M. B. Asadpoor, M.

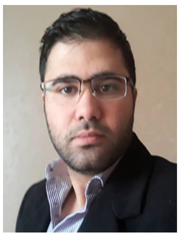
Ghasemi, A new solution to the non-convex economic load dispatch problems using phasor particle swarm optimization, *Applied Soft Computing*, Vol. 79, pp. 111-124, June, 2019.
[40] B. Y. Qu, J. J. Liang, Y. S. Zhu, Z. Y. Wang, P. N. Suganthan, Economic emission dispatch problems with stochastic wind power using summation based multi-objective evolutionary algorithm, *Information Sciences*, Vol. 351, pp. 48-66, July, 2016.
[41] D. Singh, J. S. Dhillon, Ameliorated grey wolf optimization for economic load dispatch problem, *Energy*, Vol. 169, pp. 398-419, February, 2019.
[42] X. S. Yang, S. S. S. Hosseini, A. H. Gandomi, Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect, *Applied Soft Computing*, Vol. 12, No. 3, pp. 1180-1186, March, 2012.
[43] M. Sulaiman, Masihullah, Z. Hussain, S. Ahmad, W. K. Mashwani, M. A. Jan, R. A. Khanum, Implementation of improved grasshopper optimization algorithm to solve economic load dispatch problems, *Hacettepe Journal of Mathematics and Statistics*, Vol. 48, No. 5, 1570-1589, May, 2019.
[44] M. Said, E. H. Houssein, S. Deb, R. M. Ghoniem, A. G. Elsayed, Economic load dispatch problem based on search and rescue optimization algorithm, *IEEE Access*, Vol. 10, pp. 47109-47123, April, 2022.
[45] T. Singh, Chaotic slime mould algorithm for economic load dispatch problems, *Applied Intelligence*, Vol. 52, No. 13, pp. 15325-15344, October, 2022.
[46] W. K. Hao, J. S. Wang, X. D. Li, M. Wang, M. Zhang, Arithmetic optimization algorithm based on elementary function disturbance for solving economic load dispatch problem in power system, *Applied Intelligence*, Vol. 52, No. 10, pp. 11846-11872, August, 2022.
[47] C. Lan, D. Jia, Conceptual metonymies and metaphors behind the five phases: A case study of mu (wood), tu (earth) and jin (metal), *Chinese Language and Discourse*, Vol. 7, No. 1, pp. 66-104, September, 2016.
[48] C. Lan, D. Jia, Conceptual metonymies and metaphors behind shui (water) and huo (fire) in ancient and modern chinese, *Applied Linguistics Review*, Vol. 11, No. 2, pp. 281-310, May, 2020.
[49] K. V. Price, N. H. Awad, M. Z. Ali, P. N. Suganthan, *Problem definitions and evaluation criteria for the 100-digit challenge special session and competition on single objective numerical optimization*, Technical Report, November, 2018.
[50] H. Rakhshani, A. Rahati, Snap-drift cuckoo search: A novel cuckoo search optimization algorithm, *Applied Soft Computing*, Vol. 52, pp. 771-794, March, 2017.
[51] R. F. Woolson, Wilcoxon signed-rank test, *Wiley Encyclopedia of Clinical Trials*, John Wiley & Sons, 2007, pp. 1-3.
[52] M. M. Nischal, S. Mehta, Optimal load dispatch using ant lion optimization, *International Journal of Engineering Research and Applications*, Vol. 5, No. 8, pp. 10-19, August, 2015.
[53] J. S. Pan, J. Shan, S. C. Chu, S. J. Jiang, S. G. Zheng, L. Liao, A multigroup marine predator algorithm and its application for the power system economic load dispatch, *Energy Science & Engineering*, Vol. 10, No. 6, pp. 1840-1854, June, 2022.

Biographies

Xiaopeng Wang is currently pursuing the Ph.D. degree with the Faculty of Electrical Engineering and Computer Science, VŠB – Technical University of Ostrava, Czech Republic. His research interests are mainly in meta-heuristic algorithms and intelligent computing.

Shu-Chuan Chu received the Ph.D. degree in 2004 from the School of Computer Science, Engineering and Mathematics, Flinders University of South Australia. She is currently the Research Fellow in Shandong University of Science and Technology. Her research interests are mainly in swarm intelligence, intelligent computing and data mining.

Václav Snášel received the Ph.D. degree from Masaryk University, Czech Republic, in 1991. He is currently a Full Professor with the VŠB - Technical University of Ostrava. His research interests include artificial intelligence, social networks, conceptual lattice, information retrieval, semantic web, knowledge management, data compression, machine intelligence, and intelligent computing.



Hisham A. Shehadeh received the Ph.D. degree from the Department of Computer System and Technology, University of Malaya, in 2018. He is currently an Assistant Professor of Amman Arab University, Jordan. His research interests are intelligent computing, meta-heuristic algorithms and algorithmic engineering applications of wireless networks.

Jeng-Shyang Pan received the Ph.D. degree in electrical engineering from the University of Edinburgh, U.K., in 1996. He is currently the Professor of Shandong University of Science and Technology. His current research interests include the information hiding, artificial intelligence and wireless sensor networks.