

Mimoun YOUNES, Khodja FOUAD, Belabbes BAGDAD

# A new technique for solving the multi-objective optimization problem using hybrid approach

© Higher Education Press and Springer-Verlag Berlin Heidelberg 2014

**Abstract** Energy efficiency, which consists of using less energy or improving the level of service to energy consumers, refers to an effective way to provide overall energy. But its increasing pressure on the energy sector to control greenhouse gases and to reduce CO<sub>2</sub> emissions forced the power system operators to consider the emission problem as a consequential matter besides the economic problems. The economic power dispatch problem has, therefore, become a multi-objective optimization problem. Fuel cost, pollutant emissions, and system loss should be minimized simultaneously while satisfying certain system constraints. To achieve a good design with different solutions in a multi-objective optimization problem, fuel cost and pollutant emissions are converted into single optimization problem by introducing penalty factor. Now the power dispatch is formulated into a bi-objective optimization problem, two objectives with two algorithms, firefly algorithm for optimization the fuel cost, pollutant emissions and the real genetic algorithm for minimization of the transmission losses. In this paper the new approach (firefly algorithm-real genetic algorithm, FFA-RGA) has been applied to the standard IEEE 30-bus 6-generator. The effectiveness of the proposed approach is demonstrated by comparing its performance with other evolutionary multi-objective optimization algorithms. Simulation results show the validity and feasibility of the proposed method.

**Keywords** economic power dispatch (EPD), firefly algorithm (FFA), real genetic algorithm (RGA), hybrid method

## 1 Introduction

The economic power dispatch (EPD) problem has been

Received November 2, 2013; accepted February 3, 2014

Mimoun YOUNES (✉), Khodja FOUAD, Belabbes BAGDAD  
Faculty of Technology, Djillali Liabes University, Sidi Bel Abbès 22000, Algeria  
E-mail: younesmimoun@gmail.com

one of the most widely studied subjects in the power system community since Carpentier first published the concept in 1962 [1]. The EPD problem is a large-scale highly constrained nonlinear non-convex optimization problem [2]. To solve it, a number of conventional optimization techniques such as nonlinear programming (NLP) [3], quadratic programming (QP) [4], linear programming (LP) [5,6], and interior point methods [7], Newton-based method [8], mixed integer programming [9], dynamic programming [10], and branch and bound [11] have been applied. All of these mathematical methods are fundamentally based on the convexity of objective function to find the global minimum. However, the EPD problem has the characteristics of high nonlinearity and nonconvexity.

The applications of the conventional optimization techniques such as the gradient-based algorithms is not good enough to solve this problem because it depends on the existence of the first and the second derivatives of the objective function and on the well computing of these derivative in large search space.

Therefore, the conventional methods based on the mathematical technique cannot guarantee the finding of the global optimum. In addition, the performance of these traditional approaches also depends on the starting points and is likely to converge to local minimum or even diverge.

Recently, many attempts to overcome the limitations of the mathematical programming approaches have been investigated such as meta-heuristic optimization methods, tabu search (TS) [12], simulated annealing (SA) [13], genetic algorithms [14], evolutionary programming (EP) [15], artificial neural networks [16], particle swarm [17], ant colony optimization (ACO) [18], and harmony search algorithm [19].

Their applications to global optimization problems become attractive because they have better global search abilities over the conventional optimization algorithms. The meta-heuristic techniques seem to be promising and evolving, and have come to be the most widely used tools for solving the EPD.

For minimization/maximization problems, the meta-heuristic methods make it possible to find solutions closer to the optimum but with high cost in time.

The problem proposed in the present paper is optimization of the fuel cost, pollutant emissions and system loss simultaneously while satisfying certain system constraints. To achieve a good design with different solutions in a multi-objective optimization problem, fuel cost and pollutant emissions are converted into a single optimization problem by introducing the penalty factor. Now the power dispatch is formulated into a bi-objective optimization problem, two objectives with two algorithms, firefly algorithm for optimization of the fuel cost, pollutant emissions and the real genetic algorithm (RGA) for minimisation of the transmission losses.

The method proposed in the present paper is tested on the electrical network IEEE 30 bus. Simulation results confirm the advantage of computation rapidity and solution accuracy of the proposed method. These results show that the proposed method is promising.

## 2 Problem formulations

The EPD problem is a nonlinear, non-convex optimization problem which determines the optimal control variables for minimizing the certain objectives subject to the several equality and inequality constraints. The EPD problem is generally formulated as

$$f(\mathbf{x}, \mathbf{u}), \quad (1)$$

subject to

$$g(\mathbf{x}, \mathbf{u}) = 0, \quad (2)$$

$$h(\mathbf{x}, \mathbf{u}) \leq 0, \quad (3)$$

where  $g(\mathbf{x}, \mathbf{u})$  is the typical equality constraint,  $h(\mathbf{x}, \mathbf{u})$  is inequality constraints,  $\mathbf{x}$  is the vector of state variables consisting of slack bus power  $P_{G1}$ , load bus voltages  $U$ , reactive power generator outputs  $Q_G$ , and transmission line loading  $S$ . Hence  $\mathbf{x}$  can be expressed as

$$\mathbf{x}^T = [P_{G1}, U_{L1}, \dots, U_{LN_{PQ}}, Q_{G1}, \dots, Q_{GN_g}, S_{11}, \dots, S_{1N_1}], \quad (4)$$

where  $N_{PQ}$ ,  $N_g$  and  $N_1$  are the number of load buses, the number of generators and the number of transmission lines, respectively.  $\mathbf{u}$  is the vector of control variables consisting of generator real power outputs except at the slack bus  $P_G$ , generator voltages  $U_G$ , transformer tap settings  $T$  and reactive power injections  $Q_C$ . Hence,  $\mathbf{u}$  can be expressed as

$$\mathbf{u}^T = [P_{G2}, \dots, P_{GN_g}, U_{G1}, \dots, U_{GN_g}, T_1, \dots, T_{N_t}, Q_{C1}, \dots, Q_{CN_c}], \quad (5)$$

where  $N_t$  is the number of regulating transformers and  $N_c$  is the number of VAR compensators.

### 2.1 Objective functions

#### 2.1.1 Minimization of fuel

The goal of the conventional EPD problem is to find an optimal allocation for generating powers in a power system. The power balance constraint and the generating power constraints for all units should be satisfied. In other words, the EPD problem (see Fig. 1) is to find the optimal combination for power generations which minimizes the total fuel cost while satisfying the power balance equality constraint and several inequality constraints in the system [20].

The total fuel cost function is formulated as

$$f(P_G) = \sum_{i=1}^{N_g} f_i(P_{Gi}), \quad (6)$$

$$f_i(P_{Gi}) = a_i P_{Gi}^2 + b_i P_{Gi} + c_i, \quad (7)$$

where  $f(P_G)$  is the total production cost in \$/h;  $f_i(P_{Gi})$  is the fuel cost function of unit  $i$  in \$/h;  $a_i$ ,  $b_i$ , and  $c_i$  are the fuel cost coefficients of unit  $i$ ; and  $P_{Gi}$  is the real power output of unit  $i$  in MW.

In minimizing total fuel cost (see Fig. 2) the following constraints should be satisfied.

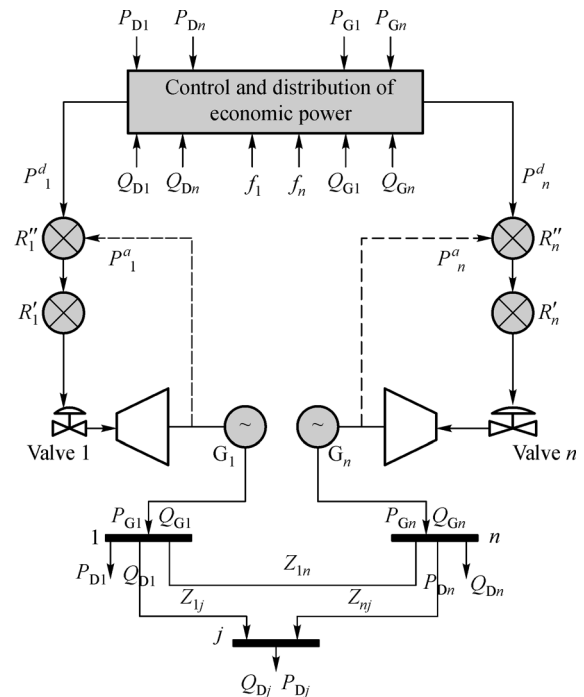


Fig. 1 Principle of EPD

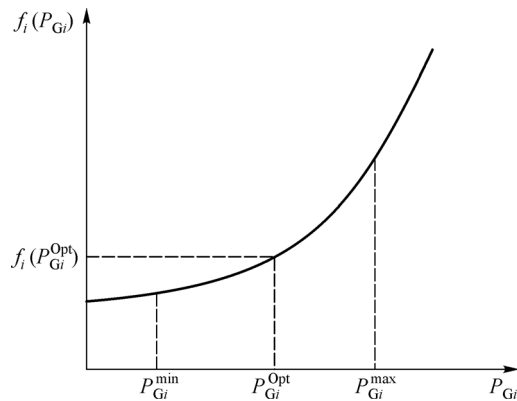


Fig. 2 Fuel cost curve of thermal generator

### 2.1.2 Minimization of real power loss

The main objective is to minimize the network active power loss while satisfying a number of operating constraints. The objective function may be expressed as

$$P_L = \sum_{k=1}^{N_l} g_k [U_i^2 + U_j^2 - 2U_i U_j \cos(\alpha_i - \alpha_j)], \quad (8)$$

where  $g_k$  is the conductance of a transmission line  $k$  connected between the  $i$ th and  $j$ th bus, while  $U_i$ ,  $U_j$ ,  $\alpha_i$ , and  $\alpha_j$  are the voltage magnitudes and phase angles of the  $i$ th and  $j$ th bus, respectively.

### 2.1.3 Total emission cost minimization

The most important emissions considered in the power generation industry due to their effects on the environment are sulfur dioxide ( $\text{SO}_2$ ) and nitrogen oxides ( $\text{NO}_x$ ). These emissions can be modeled through functions that associate emissions with power production for each unit. One approach to represent  $\text{SO}_2$  and  $\text{NO}_x$  emissions is to use a combination of polynomial and exponential terms [21]:

$$E(P_G) = \sum (\alpha_i P_{Gi}^2 + \beta_i P_{Gi} + \gamma_i) + \varepsilon_i \exp(\lambda_i P_{Gi}), \quad (9)$$

where  $\alpha_i$ ,  $\beta_i$ ,  $\gamma_i$ ,  $\varepsilon_i$  and  $\lambda_i$  are coefficients of the  $i$ th generator emission characteristics.

The bi-objective combined economic emission dispatch problem is converted into a single optimization problem by introducing the price penalty factor  $h$  as follows,

$$\text{Min} F = \text{FC} + h \cdot \text{EC},$$

$$F = \sum_{i=1}^{N_g} f_i(P_{Gi}) + h \sum_{i=1}^{N_g} E(P_{Gi}),$$

subjected to the power flow constraints of equations. The price penalty factor  $h$  blends the emission with fuel cost

and  $F$  is the total operating cost in \$/h. The price penalty factor  $h_i$  is the ratio between the maximum fuel cost and maximum emission of the corresponding generator.

$$h_i = \frac{\sum_{i=1}^{N_g} f(P_{Gi}^{\max})}{\sum_{i=1}^{N_g} E(P_{Gi}^{\max})}. \quad (10)$$

The following steps are used to find the price penalty factor for a particular load demand:

- 1) Find the ratio between the maximum fuel cost and maximum emission of each generator.
- 2) Arrange the values of the price penalty factor in ascending order.
- 3) Add the maximum capacity of each unit  $P_{Gi}^{\max}$  one at a time, starting from the smallest  $h_i$  unit until

$$\sum P_{Gi}^{\max} \geq P_D. \quad (11)$$

- 4) At this stage,  $h_i$  associated with the last unit in the process is the price penalty factor  $h$  for the given load.

The above procedure gives the approximate value of price penalty factor computation for the corresponding load demand. Hence a modified price penalty factor ( $h_m$ ) is introduced in this work to give the exact value for the particular load demand. The first two steps of  $h$  computation remain the same for the calculation of the modified price penalty factor. Then it is calculated by interpolating the values of  $h_i$  corresponding to their load demand values.

## 2.2 Problem constraints

### 2.2.1 Active power balance equation

For power balance, an equality constraint should be satisfied. The generated power should be the same as the total load demand added to the total line losses, which is represented as

$$\sum_{i=1}^{N_g} P_{Gi} = P_{\text{load}} + P_L, \quad (12)$$

where  $\sum_{i=1}^{N_g} P_{Gi}$  is the total system production,  $P_{\text{load}}$  is the total load demand,  $P_L$  is the total transmission loss of the system in MW, and  $N_g$  is the number of generator units in the system.

The exact value of the system losses can be determined by means of a power flow solution. The most popular approach for finding an approximate value of the losses is by using Kron's loss formula as given in Eq. (5), which represents the losses as a function of the output level of the system generators.

$$P_L = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} P_i B_{ij} P_j + \sum_{i=1}^{N_g} B_{0i} P_{Gi} + B_{00}, \quad (13)$$

where  $B_{ij}$  is the transmission loss coefficient,  $P_i$  and  $P_j$  are the power generation of the  $i$ th and  $j$ th units,  $B_{0i}$  is the  $i$ th element of the loss coefficient vector,  $B_{00}$  is the constant loss coefficient, and  $\sum_{j=1}^{N_D} P_{Dj}$  is the total system demand.

The equality constraints on real and reactive power at each bus are given as

$$P_{Gi} - P_{Di} - U_i \sum_{j=1}^{N_b} U_j [G_{ij} \cos(\alpha_i - \alpha_j) - B_{ij} \sin(\alpha_i - \alpha_j)] = 0, \quad (14)$$

$$Q_{Gi} - Q_{Di} - U_i \sum_{j=1}^{N_b} U_j [G_{ij} \sin(\alpha_i - \alpha_j) - B_{ij} \cos(\alpha_i - \alpha_j)] = 0, \quad (15)$$

where  $i = 1, 2, \dots, N_b$ , and  $N_b$  is the number of buses;  $Q_{Gi}$  is the reactive power generated at the  $i$ th bus;  $P_{Di}$  and  $Q_{Di}$  are the bus real and reactive load, respectively;  $G_{ij}$  and  $B_{ij}$  are the transfer conductance and susceptance between bus  $i$  and bus  $j$ , respectively;  $U_i$  and  $U_j$  are the voltage magnitudes at bus  $i$  and bus  $j$ , respectively; and  $\alpha_i$  and  $\alpha_j$  are the voltage angles of bus  $i$  and bus  $j$ , respectively.

### 2.2.2 Active power generation limits

Generation constraints: The generator voltages, real power outputs and reactive power outputs are restricted by their lower and upper bounds as follows:

$$U_{Gi}^{\min} \leq U_{Gi} \leq U_{Gi}^{\max}, \quad (16)$$

$$P_{Gi}^{\min} \leq P_{Gi} \leq P_{Gi}^{\max}, \quad (17)$$

$$Q_{Gi}^{\min} \leq Q_{Gi} \leq Q_{Gi}^{\max}. \quad (18)$$

Transformer constraints: The transformer tap settings are restricted by their minimum and maximum limits as follows:

$$T_i^{\min} \leq T_i \leq T_i^{\max}. \quad (19)$$

Shunt VAR constraints: The reactive power injections at the buses are restricted by their minimum and maximum limits as

$$Q_{Ci}^{\min} \leq Q_{Ci} \leq Q_{Ci}^{\max}. \quad (20)$$

Security constraints: The security constraints include the constraints of voltage magnitudes at load buses and transmission line loadings as follows:

$$U_{Li}^{\min} \leq U_{Li} \leq U_{Li}^{\max}, \quad (21)$$

$$S_{li} \leq S_{li}^{\max}. \quad (22)$$

## 3 Firefly algorithm (FFA)

Fireflies (lightning bugs) use their bioluminescence to attract mates or prey. Some of them live in moist places under debris on the ground, others beneath bark and decaying vegetation. FFA was developed at Cambridge University in 2008 [22] by using the following three idealized rules:

1) All fireflies are unisex so that a firefly will be attracted to other fireflies regardless of their sex [23].

2) Attractiveness is proportional to their brightness; thus for any two flashing fireflies the less bright will move toward the brighter [24]. The attractiveness is proportional to the brightness and they both decrease as their distance increases. If there is no brighter firefly than a particular one it will move randomly.

3) The brightness of a firefly is affected or determined by the landscape of the objective function. On the first rule, each firefly attracts all the other fireflies with weaker flashes [25].

The brightness of a firefly is affected or determined by the landscape of the objective function. For a maximization problem the brightness can simply be proportional to the value of the objective function. Other forms of brightness can be defined in a similar way to the fitness function in genetic algorithms based on these three rules.

### 3.1 Attractiveness

The light intensity  $I$  varies with distance  $r$  [26], which is expressed by

$$I(r) = I_0 e^{-\gamma r^2}. \quad (23)$$

As attractiveness of a firefly is proportional to the light intensity [27] seen by adjacent fireflies, the attractiveness  $\beta$  of a firefly can now be defined by

$$\beta(r) = \beta_0 e^{-\gamma r^2}, \quad (24)$$

where  $I$  is the light intensity,  $I_0$  is the original light intensity,  $\gamma$  is the light absorption coefficient, and  $\beta_0$  is the attractiveness [28].

### 3.2 Distance and movement

The distance between any two fireflies  $i$  and  $j$  at  $x_i$  and  $x_j$  is the Cartesian distance given by Ref.[29] as

$$r_{ij} = |x_i - x_j| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2}, \quad (25)$$

where  $x_{i,k}$  is the  $k$ th component of the spatial coordinate  $x_i$  of the  $i$ th firefly,  $k = 1$  to  $d$ , which  $d$  denotes the dimensionality of a problem.

The movement of a firefly  $i$  attracted to another more attractive firefly  $j$  is determined by

$$x_{i+1} = x_i + \beta_0 e^{-\gamma r_{ij}^2} (x_j - x_i) + \alpha \left( \text{rand} - \frac{1}{2} \right), \quad (26)$$

where the first term is the current position of a firefly [30], the second term is used for considering the attractiveness of a firefly to light intensity seen by adjacent fireflies, and the third term is used for the random movement of a firefly in case there are not any brighter ones.

The coefficient  $\alpha$  is a randomization parameter determined by the problem of interest, while  $\text{rand}$  is a random number generator uniformly distributed in the space  $(0, 1)$  [31]. As will be seen in this implementation of the algorithm,  $\beta_0 = 0.1$ ,  $\alpha \in (0, 1)$  and the attractiveness or absorption coefficient  $\gamma = 1.0$  which guarantees a quick convergence of the algorithm to the optimal solution will be used [32].

## 4 Genetic algorithm (GA)

For the application of the GA, the method of penalty is used:

$g_i(P_{Gi}) \geq 0$ , ( $i = 0, 1, \dots, m$ ) are the constraints of the inequality type, and  $h_j(P_{Gi}) = 0$ , ( $j = 0, 1, \dots, n$ ) are the constraints of the equality type.

By transforming the problem into a function of penalty, Eq. (27) can be obtained.

$$F(P_{Gi,r^k}) = F(P_{Gi}) + \frac{1}{r^k} \sum_{j=1}^n H(h_j(P_{Gi})) + r^k \sum_{i=1}^m G(g_i(P_{Gi})), \quad (27)$$

where  $r^k$  is the coefficient of penalization, and the functions of penalty  $H(h_j(P_{Gi}))$  and  $G(g_i(P_{Gi}))$  are determined by three methods of penalty:

1) Method of external penalty:

$$G(g_i(P_{Gi})) = 0 \text{ and } H(h_j(P_{Gi})) = [h_j(P_{Gi})]^2.$$

The function of penalty to be solved becomes

$$F(P_{Gi,r^k}) = F(P_{Gi}) + \frac{1}{r^k} \sum_{j=1}^n [h_j(P_{Gi})]^2. \quad (28)$$

2) Method of interior penalty:

In this case, only the constraints of inequality are taken into account and are defined as

$$G(g_i(P_{Gi})) = \frac{1}{g_i(P_{Gi})}.$$

The function to be minimized will be

$$F(P_{Gi,r^k}) = F(P_{Gi}) + r^k \sum_{i=1}^m \frac{1}{g_i(P_{Gi})}. \quad (29)$$

3) Method of penalty mixed:

It acts in this method of a combination of both premieres

$$G(g_i(P_{Gi})) = 1/g_i(P_{Gi}) \text{ and } H(h_j(P_{Gi})) = [h_j(P_{Gi})]^2.$$

The function to be minimized will be

$$F(P_{Gi,r^k}) = F(P_{Gi}) + \frac{1}{r^k} \sum_{j=1}^n [h_j(P_{Gi})]^2 + r^k \sum_{i=1}^m \left[ \frac{1}{g_i(P_{Gi})} \right]. \quad (30)$$

### 4.1 Representation

It turns out that there is no rigorous definition of “genetic algorithm” accepted by all in the evolutionary-computation community that differentiates GA from other evolutionary computation methods [33].

However, it can be said that most methods called “GA” have at least the following elements in common:

Populations of chromosomes, selection according to fitness, crossover to produce new offspring are random.

The chromosomes in a GA population typically take the form of bit strings. Each locus in the chromosome has two possible alleles: 0 and 1. Each chromosome can be thought of as a point in the search space of candidate solutions. The GA processes populations of chromosomes, successively replacing one such population with another. The GA most often requires a fitness function that assigns a score (fitness) to each chromosome in the current population. The fitness of a chromosome depends on the ability the chromosome solves the problem at hand [34].

### 4.2 GA operators

The simplest form of GA involves three types of operators: selection, crossover (single point), and mutation [35].

#### 4.2.1 Selection

This operator selects chromosomes in the population for reproduction. The fitter the chromosome is, the more likely it is selected to reproduce.



#### 4.2.2 Crossover

This operator randomly chooses a locus and exchanges the subsequences before and after that locus between two chromosomes to create two offspring. For example, the strings 10000100 and 11111111 could be crossed over after the third locus in each to produce the two offspring 10011111 and 11100100. The crossover operator roughly mimics biologic recombination between two single-chromosome (haploid) organisms.

#### 4.2.3 Mutation

This operator randomly flips some of the bits in a chromosome. For example, the string 00000100 might be mutated in its second position to yield 01000100. Mutation can occur at each bit position in a string with some probability, usually very small (e.g., 0.001).

#### 4.2.4 A binary-coded genetic algorithm (BCGA)

Given a clearly defined problem to be solved and a bit string representation for candidate solutions, a simple BCGA works as follows:

- 1) Start with a randomly generated population of  $n$   $l$ -bit chromosomes (candidate solutions to a problem).
- 2) Calculate the fitness  $f(x)$  of each chromosome  $x$  in the population.
- 3) Repeat the following steps until  $n$  offspring have been created:

Select a pair of parent chromosomes from the current population, the probability of selection being an increasing function of fitness. Selection is done “with replacement”, meaning that the same chromosome can be selected more than once to become a parent.

With probability  $P_c$  (the “crossover probability” or “crossover rate”), cross over the pair at a randomly chosen point (chosen with uniform probability) to form two offspring. If no crossover takes place, form two offsprings that are exact copies of their respective parents. (Note that here the crossover rate is defined as the probability that two parents will cross over in a single point. There are also “multi-point crossover” versions of the BCGA in which the crossover rate for a pair of parents is the number of points at which a crossover takes place.)

Mutate the two offsprings at each locus with probability  $P_m$  (the mutation probability or mutation rate), and place the resulting chromosomes in the new population. If  $n$  is odd, one new population member can be discarded at random.

4) Replace the current population with the new population.

5) Go to Step 2. Each iteration of this process is called a generation. BCGA are typically iterated for anywhere from 50 to 500 or more generations. The entire set of

generations is called a *run*. At the end of a run there are often one or more highly fit chromosomes in the population. Since randomness plays a large role in each run, two runs with different random-number seeds will generally produce different detailed behaviors. BCGA researchers often report statistics (such as the best fitness found in a run and the generation at which the individual with that best fitness was discovered) averaged over many different runs of the BCGA on the same problem (Fig. 2).

The simple procedure just described is the basis for most applications of BCGAs. There are a number of details to fill in, such as the size of the population and the probabilities of crossover and mutation, and the success of the algorithm often depends greatly on these details. There are also more complicated versions of BCGA (e.g., BCGA that work on representations other than strings or BCGA that have different types of crossover and mutation operators).

As a more detailed example of a simple BCGA, suppose that  $l$  (string length) is 8, that  $f(x)$  is equal to the number of ones in bit string  $x$  (an extremely simple fitness function, used here only for illustrative purposes), that  $n$  (the population size) is 4, that  $P_c = 0.7$ , and that  $P_m = 0.001$ . (Like the fitness function, these values of  $l$  and  $n$  were chosen for simplicity. More typical values of  $l$  and  $n$  are in the range of 50–1000. The values given for  $P_c$  and  $P_m$  are fairly typical.) The initial (randomly generated) population is listed in Table 1.

**Table 1** Initial (randomly generated) population

Chromosome	Chromosome string	Fitness
<i>A</i>	00000110	2
<i>B</i>	11101110	6
<i>C</i>	00100000	1
<i>D</i>	00110100	3

A common selection method in BCGA is fitness-proportionate selection, in which the number of times an individual is expected to reproduce is equal to its fitness divided by the average of fitnesses in the population. (This is equivalent to what biologists call “viability selection”)

A simple method of implementing fitness-proportionate selection is “roulette-wheel sampling” which is conceptually equivalent to giving each individual a slice of a circular roulette wheel equal in area to the individual’s fitness. The roulette wheel is spun, the ball comes to rest on one wedge-shaped slice, and the corresponding individual is selected. In the  $n = 4$  example above, the roulette wheel would be spun four times; the first two spins might choose chromosomes *B* and *D* to be parents, and the second two spins might choose chromosomes *B* and *C* to be parents. (The fact that *A* might not be selected is just the luck of the draw. If the roulette wheel were spun many times, the average results would be closer to the expected values)

Once a pair of parents is selected, with probability  $P_c$  they cross over to form two offspring. If they do not cross over, then the offsprings are exact copies of each parent. Suppose, in the example above, that parents  $B$  and  $D$  crossover after the first bit position to form offspring  $E = 10110100$  and  $F = 01101110$ , and parents  $B$  and  $C$  do not cross over, instead forming offspring that are exact copies of  $B$  and  $C$ . Next, each offspring is subject to mutation at each locus with probability  $P_m$ . For example, suppose offspring  $E$  is mutated at the sixth locus to form  $E' = 10110000$ , offspring  $F$  and  $C$  are not mutated at all, and offspring  $B$  is mutated at the first locus to form  $B' = 01101110$ . The new population is presented in Table 2 and the general structure of standard GA is depicted in Fig. 3.

**Table 2** New population

Chromosome label	Chromosome string	Fitness
$E'$	10110000	3
$F$	01101110	5
$C$	00100000	1
$B'$	01101110	5

### 4.3 Real genetic algorithm (RGA)

Using the binary coding, it is rather simple to set up all the operations. Despite everything, some disadvantages exist:

1) It can be difficult to adapt this coding to certain problems. The traditional binary representation used for the genetic algorithms causes difficulties for the large-sized problems optimization to high numerical precision. For example, with 100 variables belonging to the interval  $[-500, 500]$  and whose accuracy of 6 digits after the comma is necessary, the size of the chromosome is 3000, which, in return, generates a space of search for

approximately 101000. For such problems, the genetic algorithms based on binary representations have weak performances.

2) The distance of Hamming between two close real numbers which is the number of bits that is different from the one with the other can be large (example: 0111 which is worth 7 and 1000 which worth 8, the distance is 4) and very often creates a convergence toward a non-optimal value.

3) According to the problem, the resolution of the algorithm can be expensive in time.

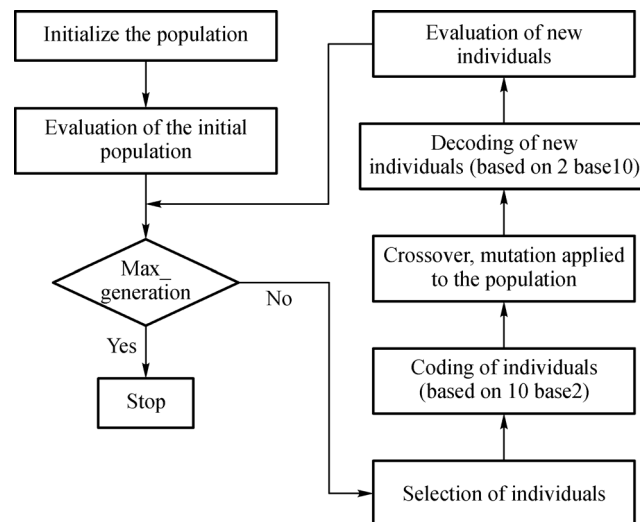
4) The crossing and the change can be unsuited (creation of individuals not belonging to the search space).

One of the major improvements consists of using real numbers directly [36]. RGA is a probabilistic search technique, which generates the initial parent vectors distributed uniformly in intervals within the limits and obtains global optimum solution over number of iterations.

The implementation of RGA is given below. The initial population is generated after satisfying Eq. (17). The elements of parent vectors ( $P_{Gi}$ ) are the real power outputs of generating units distributed uniformly between their minimum and maximum limits.

The fitness function is used to transform the cost function value into a measure of relative fitness. The cost function is given in Eq. (6).

The selection is based on the cost of parent vectors  $f(P_{Gi})$  with the corresponding cost of offspring vectors  $f(P'_{Gi})$  in this population. The best vector having the minimum cost, whether parent vector  $P_{Gi}$  or offspring vector  $P'_{Gi}$  is selected for the new parent for the next generation. A non-uniform arithmetic crossover operator is used. After the crossover is completed, non-uniform mutation is performed. In the mutation step, a random real value makes a random change in the  $m$ th element of



**Fig. 3** General structure of standard genetic algorithm

the chromosome [37]. After the mutation, all constraints are checked whether violated or not. If the solution has at least one constraint violated, a new random real value is used for finding a new value of the  $m$ th element of the chromosome. Then, the best solution so far obtained in the search is retained and used in the following generation. The RGA process repeats until the specified maximum number of generations is reached [38]. The diagram of the proposed algorithm is shown as follows:

**Step 1** Create an initial population.

**Step 2** Evaluate the adaptation of each individual.

**Step 3** Is there a convergence?

If yes, post the results.

Else go to Step 4.

**Step 4** Select the individuals.

**Step 5** Subject the population of the crossings (non-uniform arithmetic) and the random mutations (non-uniform mutation).

**Step 6** Evaluate the adaptation of the new individuals and to go to Step 3.

## 5 Firefly algorithm-real genetic algorithm (FFA-RGA)

It has been noticed that the meta-heuristic methods are very efficient for the search of global solution for complex problems better than deterministic methods. However their disadvantage is the time of convergence caused the high number of the agents and iterations. To solve this problem, two meta-heuristic methods, the FFA and the RGA are combined with a lower number of fireflies and the population as possible. This paper proposes a hybrid method which has two meta-heuristics, the FFA with the RGA are employed to solve the EPD problem including transmission losses in power system. The power dispatch is formulated into a bi-objective optimization problem, which is to minimize the fuel cost as well as transmission losses. Two objectives with two algorithms, firefly algorithm for optimization of the global cost function and RGA for minimisation of the transmission losses. The flow chart for the EPD using the FFA-RGA is demonstrated in Fig. 4.

$$B_{ij} = \begin{bmatrix} 0.1382 & -0.0299 & 0.0044 & -0.0022 & -0.0010 & -0.0008 \\ -0.0299 & 0.0487 & -0.0025 & 0.0004 & 0.0016 & 0.0041 \\ 0.0044 & -0.0025 & 0.0182 & -0.0070 & -0.0066 & -0.0066 \\ -0.0022 & 0.0004 & -0.0070 & 0.0137 & 0.0050 & 0.0005 \\ -0.0010 & 0.0016 & -0.0066 & 0.0050 & 0.0109 & 0.0005 \\ -0.0008 & 0.0041 & -0.0066 & 0.0033 & 0.0005 & 0.0244 \end{bmatrix},$$

$$B_0 = [-0.0107 \quad 0.0060 \quad -0.0017 \quad 0.0009 \quad 0.0002 \quad 0.003],$$

## 6 Simulation results

The proposed approach (FFA-RGA) for solving the EPD problem has been applied to the standard IEEE 30 bus test system, as displayed in Fig. 5. The line, bus data, generator data and the minimum and maximum limits for the control variables are given in Table 3.

The test system consists of 41 branches, 6 generator buses placed at buses 1, 2, 13, 22, 23, and 27 load-buses. Four branches, (6, 9), (6, 10), (4, 12) and (27, 28), are under load tap setting transformer branches within the interval (0.9, 1.1). The possible 9 reactive power sources within the interval (0, 0.05) are installed at bus numbers 10, 12, 15, 17, 20, 21, 23, 24 and 29. The generated voltages of the generator buses are within the range of (0.95, 1.1) and the load bus voltages are within the range of (0.9, 1.05). The total system demand is 2.834 pu at 100 MVA base [39].

All the simulations were performed on a personal computer with 3 GHz Intel Processor and 2 GB of RAM running Matlab 7.6. The results obtained by the proposed approach are compared with the results found by other heuristic methods reported in the literature recently.

### 6.1 Case 1: quadratic fuel cost minimization

In this case the objective function is a quadratic form (Eq. (6)).

The minimum total fuel cost obtained from the proposed FFA algorithm was 792.576410 \$/h. Figure 6 shows the convergence curve corresponding to the minimum total fuel cost resulted from the FFA algorithm (Tables 4 and 5). The fuel cost minimization decreased to 792.576410 \$/h (18.7600%) in Case 1 in comparison to 975.5687 \$/h in Case 2.

The results obtained from the FFA are compared with other methods reported in the literature, as shown in Tables 5 and 6. It can be seen that the minimum total active power losses obtained by the method is 792.576410 MW, which is less than those of the methods of MSFLA [39], GA-OPF [40], FGA [40], IEP [41], TS [42], EP [43], Hybrid MPSO-SFLA [44], PSO [44], SFLA [44].

The  $B$ -coefficients used in Refs. [48,49] are





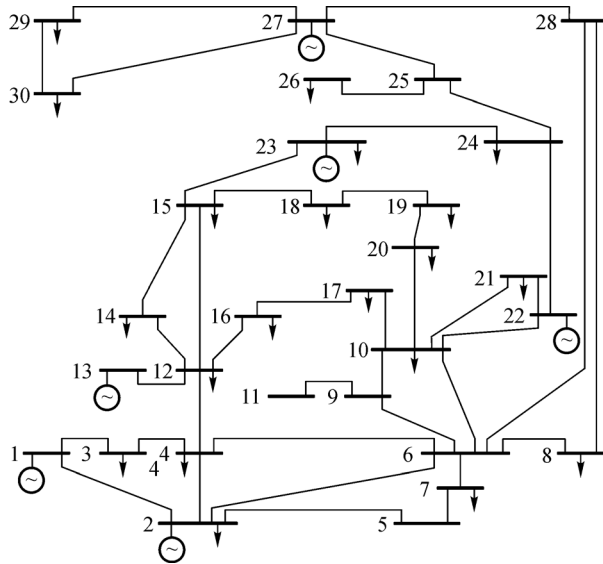


Fig. 5 Single line diagram of IEEE 30 bus test system

$$B_{00} = 9.8557 \times 10^{-4}.$$

### 6.2 Case 2: active power loss minimization

The objective function selected was the active power loss minimization (Eq. (8)). The active power loss minimization decreased to 2.75 t/h (72.2100%) in Case 2 in comparison to 9.8950 t/h in Case 1.

The result obtained from the FFA was compared with other methods reported in the literature. The results of this comparison are shown in Tables 4 and 7. It can be seen that the minimum total active power losses obtained by the proposed FFA is 2.75 MW, which is less than those of the methods of MSFLA [39], FGA [40], IEP [41], Hybrid MPSO-SFLA [44], PSO [44], SFLA [44], EGA [45], EADDE [46], DE [46], and HS [47].

Figure 7 shows the convergence of the losses minimization corresponding to the results from the FFA.

### 6.3 Case 3: Emission minimization

The objective function selected was the total emission cost

minimization  $E$  as defined in Eq. (9). The total emission cost decreased to 0.20012 t/h (43.89%) in Case 3 in comparison to 0.365141 t/h in Case 1.

Figure 8 shows the convergence of the losses minimization corresponding to the results from the FFA.

The results obtained from the FFA were compared with other methods reported in the literature. The results of this comparison are shown in Table 8. It can be seen that the total emissions are less than those of the methods of MSFLA [47], SFLA [47], GA [47], PSO [47], and PSO [44].

### 6.4 Case 4: emission, cost and losses minimization

In this case, all constraints about fuel cost, pollution emission and system transmission loss are considered. The EPD problem was considered as a multi-objective problem. The best compromise solution by using the RGA with the FFA is given in Table 4. According to Table 4, the total active power loss found by the proposed FFA-RGA was 4.7651 MW, but it is reduced to 9.8950 as in Case 1(51.8434% less)

The fuel cost in this case is reduced by as much as 18.1525 % in comparison to 975.5687MW in Case 2. The emission is reduced by as much as 42.6351% in comparison to 0.358156 in Case 1.

In this case, three competing objectives, i.e., fuel cost, losses and emission, were considered. This multi-objective optimization problem was solved by the proposed approach (FFA-RGA). The compromise solution was found using the RGA, for minimizing the losses function and integrating in the cost and emission functions. The best solution for the minimum cost, minimum loss and the compromise solution are given in Table 4.

## 7 Conclusions

This paper proposes a hybrid method with two meta-heuristic methods, the FFA with RGA. They are employed to solve the EPD problem. The transmission losses, fuel cost, pollutant emissions and system loss are minimized simultaneously while satisfying certain system constraints. The proposed algorithm has been tested on an IEEE 30-bus

Table 3 Generators parameters of the IEEE 30 bus

Bus	$P_{Gi}^{\min}$ /MW	$P_{Gi}^{\max}$ /MW	Cost coefficients		
			$a_i$	$b_i$	$c_i$
$P_{G1}$	50	200	0.00375	2.00	0.00
$P_{G2}$	20	80	0.01750	1.75	0.00
$P_{G5}$	15	50	0.06250	1.00	0.00
$P_{G8}$	10	35	0.00834	3.25	0.00
$P_{G11}$	10	30	0.02500	3.00	0.00
$P_{G13}$	12	40	0.02500	3.00	0.00

**Table 4** Optimization results of FFA-RGA approach for IEEE 30-bus

	Case1 Best cost (FFA)	Case 2 Best losses (FFA)	Case 3 Best emission (FFA)	Case 4 Best comp (FFA-GA)
$P_{G1}/MW$	183.6671	153.6546	155.7184	160.3670
$P_{G2}/MW$	35.7766	53.7402	40.3519	52.0428
$P_{G3}/MW$	24.7353	28.1859	32.8250	15.7494
$P_{G8}/MW$	11.0505	17.7983	17.3313	15.8770
$P_{G11}/MW$	10.6959	17.6231	20.1661	25.4492
$P_{G13}/MW$	27.3693	12.1365	20.7859	18.6795
$U_{G1}/pu$	1.0177	1.0528	1.0277	0.9567
$U_{G2}/pu$	1.0693	1.0965	1.0129	1.0981
$U_{G3}/pu$	1.0996	0.9654	0.9822	1.0039
$U_{G4}/pu$	1.0633	1.0684	1.0835	1.0740
$U_{G5}/pu$	0.9728	0.9762	0.9542	1.0742
$U_{G6}/pu$	1.0964	1.0334	0.9752	1.0407
$T_{CL11}$	1.0828	1.0936	1.0116	1.0383
$T_{CL12}$	1.0512	1.0995	0.9914	0.9200
$T_{CL15}$	1.0871	1.0839	0.9101	1.0260
$T_{CL36}$	1.0946	1.0882	1.0472	0.9729
$Q_{C10}/pu$	0.0277	0.0493	0.0496	0.0499
$Q_{C12}/pu$	0.0296	0.0496	0.0499	0.0416
$Q_{C13}/pu$	0.0418	0.0499	0.0499	0.0413
$Q_{C17}/pu$	0.0407	0.0488	0.0496	0.0489
$Q_{C20}/pu$	0.0553	0.0412	0.0497	0.0492
$Q_{C21}/pu$	0.0460	0.0489	0.0590	0.0498
$Q_{C23}/pu$	0.0439	0.0441	0.0404	0.0447
$Q_{C24}/pu$	0.0491	0.0499	0.0494	0.0489
$Q_{C29}/pu$	0.0428	0.0317	0.0492	0.0391
Fuel cost/(\$·h <sup>-1</sup> )	792.5764	975.56	801.5510	798.4820
Emission/(t·h <sup>-1</sup> )	0.3581	0.2028	0.2001241	0.2055
Real loss/MW	9.8950	2.7596	3.7789	4.7651
$t/s$	0.9687	0.4256	0.2656	0.9218

**Table 5** Comparison of results by different algorithms for cost objective function of IEEE 30-bus system

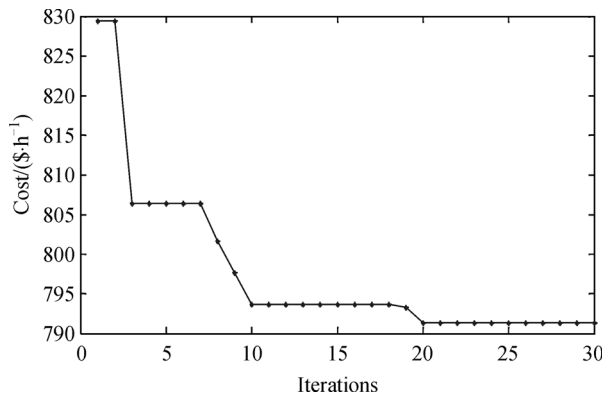
Methods	$P_{G1}/MW$	$P_{G2}/MW$	$P_{G3}/MW$	$P_{G8}/MW$	$P_{G11}/MW$	$P_{G13}/MW$	Loss/MW	Cost/(\$·h <sup>-1</sup> )	$T/S$
MSFLA [39]	179.19	48.98	20.45	20.92	11.58	11.95	9.69	802.28	–
GA-OPF [40]	174.83	48.88	23.78	20.2	13.14	12.22	–	803.92	–
FGA [40]	175.14	50.35	21.45	21.18	12.67	12.11	9.49	802	–
IEP [41]	176.24	49.01	21.5	21.81	12.34	12.01	10.87	802.47	594.08
TS [42]	176.04	48.76	21.56	22.05	12.44	12	–	802.29	–
EP [43]	173.85	50	21.39	22.63	12.93	12	–	802.62	–
Hybrid MPSO-SFLA [44]	180.53	52.09	22.78	15.49	10	12.05	9.54	801.75	18.17
PSO [44]	180.23	52.09	22.81	15.62	10	12.21	9.56	801.89	20.19
SFLA [44]	181.06	52.17	22.47	15.35	10	12.07	9.72	802.21	20.75
FFA	183.66	35.77	24.73	11.05	10.69	27.36	185.57	792.57	1.094
FFA-RGA (Case 4)	160.36	52.042	15.74	15.87	25.44	18.67	189.87	798.48	0.92

**Table 6** Comparisons of the results obtained for Case 2 of IEEE 30-bus system

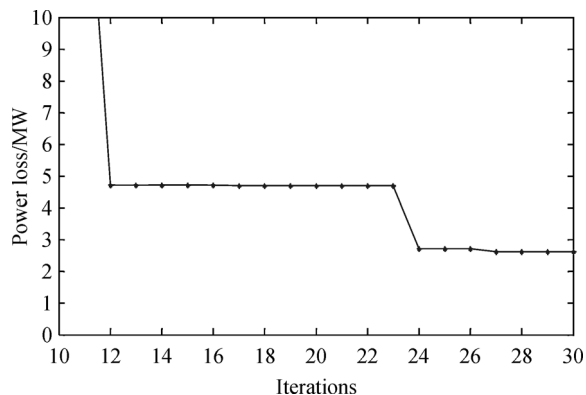
Method	Total active power losses/MW
EGA [45]	3.2008
EADDE [46]	9.13
DE [46]	9.06
HS [47]	2.9678

**Table 7** FFA-RGA method parameters

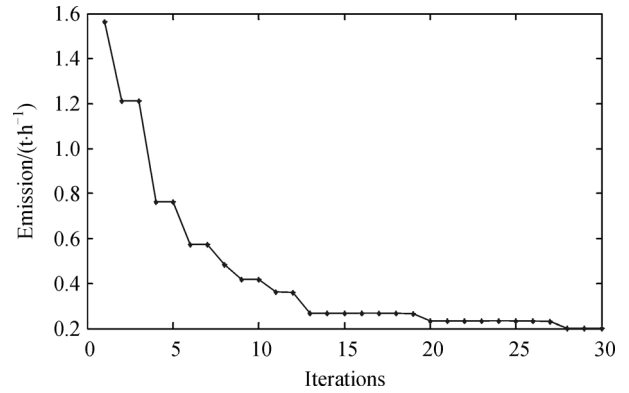
Parameter	Setting
Number of iterations FA-RGA	30
Population size for firefly( <i>n</i> )	10
Light absorption coefficient( $\gamma$ )	1.0
Randomization parameter of FFA( $\alpha$ )	0.4
Attractiveness coefficient of FFA( $\beta_0$ )	1.0
Population size	20
Crossover (non-uniform arithmetic) mutation (non-uniform mutation)	



**Fig. 6** Convergence graph of FFA, IEEE 30-bus test system (Case 1)



**Fig. 7** Convergence graph of FFA, IEEE 30-bus test system (Case 2)



**Fig. 8** Convergence graph of FFA, IEEE 30-bus test system (Case 3)

**Table 8** Comparisons of the results obtained for Case 3 of IEEE 30-bus system

Method	Total emissions/(t·h <sup>-1</sup> )
MSFLA [47]	0.2056
SFLA [47]	0.2063
GA [47]	0.21170
PSO [47]	0.2096
PSO [39]	0.37255

system to verify its efficiency. The proposed approach appears to be very efficient in particular for its fast convergence to the optimum and its interesting financial profit. The effectiveness of the proposed approach is demonstrated by comparing its performance with that of other evolutionary multi-objective optimization algorithms. The computational results reveal that the multi-objective FFA-RGA approach has excellent convergence characteristics and is superior to other multi-objective optimization algorithms. Besides, the results confirm the great potential of the proposed approach in handling the multi-objective problems. The method can be applied to other optimization problems.

## References

1. Carpentier J. Contribution to the study of economic dispatch. Bulletin of the French Society of Electricians, 1962, 3: 431–447
2. Vanderbei J R, Shanno F D. An interior-point algorithm for nonconvex nonlinear programming. Computational Optimization and Applications, 1999, 13(1–3): 231–252
3. Bottero M H, Caliana E D, Fahmideh-Vojdani A R. Economic dispatch using the reduced hessian. IEEE Transactions on Power Apparatus and Systems, 1982, 101(10): 3679–3688
4. Reid G E, Hasdorf L. Economic dispatch using quadratic programming. IEEE Transactions on Power Apparatus and Systems,

- 1973, 92(6): 2015–2023
5. Stott B, Hobson E. Power system security control calculation using linear programming. *IEEE Transactions on Power Apparatus and Systems*, 1978, 97(5): 1713–1720
  6. Stott B, Hobson E. Power system security control calculation using linear programming. *IEEE Transactions on Power Apparatus and Systems*, 1978, 97(5): 1721–1731
  7. Momoh J A, Zhu J Z. Improved interior point method for OPF problems. *IEEE Transactions on Power Systems*, 1999, 14(3): 1114–1120
  8. Sun D I, Ashley B, Brewer B, Hughes A, Tinney W F. Optimal power flow by Newton approach. *IEEE Transactions on Power Apparatus and Systems*, 1984, PAS-103(10): 2864–2880
  9. Bahiense L, Oliveira G C, Pereira M, Granville S. A mixed integer disjunctive model for transmission network expansion. *IEEE Transactions on Power Systems*, 2001, 16(3): 560–565
  10. Dusonchet Y P, El-Abiad A H. Transmission planning using discrete dynamic optimization. *IEEE Transactions on Power Apparatus and Systems*, 1997, 92: 1358–1371
  11. Haffner S, Monticelli A, Garcia A, Romero R. Specialised branch and bound algorithm for transmission network expansion planning. *IEE Proceedings-Generation, Transmission and Distribution*, 2001, 148(5): 482–488
  12. Glover F. Tabu search—Part I. *ORSA Journal on Computing*, 1986, 1(3): 190–206
  13. Kirkpatrick S, Gelatt C D, Vecchi M P. Optimisation by simulated annealing. *Science*, 1983, 220(4598): 671–680
  14. Lai L L, Ma J T, Yokoyama R, Zhao M. Improved genetic algorithms for optimal power flow under both normal and contingent operation states. *International Journal of Electrical Power & Energy Systems*, 1997, 19(5): 287–292
  15. Yuryevich J, Wong K P. Evolutionary programming based optimal power flow algorithm. *IEEE Transactions on Power Systems*, 1999, 14(4): 1245–1250
  16. Mousavian S, Valenzuela J, Wang J. Real-time data reassurance in electrical power systems based on artificial neural networks. *Electric Power Systems Research*, 2013, 96: 285–295
  17. Abido M A. Optimal power flow using particle swarm optimization. *International Journal of Electrical Power & Energy Systems*, 2002, 24(7): 563–571
  18. Song Y H, Chou C S, Stonham T J. Combined heat and power economic dispatch by improved ant colony search algorithm. *Electric Power Systems Research*, 1999, 52(2): 115–121
  19. Fesanghary M, Mahdavi M, Minary-Jolandan M, Alizadeh Y. Hybridizing harmony search algorithm with sequential quadratic programming for engineering optimization problems. *Computer Methods Applied Mechanics and Engineering*, 2008, 197(33–40): 3080–3091
  20. Abido M A. A novel multiobjective evolutionary algorithm for environmental/economic power dispatch. *Electric Power Systems Research*, 2003, 65(1): 71–81
  21. Abido M A. Multiobjective evolutionary algorithms for electric power dispatch problem. *IEEE Transactions on Evolutionary Computation*, 2006, 10(3): 315–329
  22. Yang X S. Review of meta-heuristic and generalized evolutionary walk algorithm. *International Journal of Bio-Inspired Computation*, 2011, 3(2): 77–84
  23. Amiri B, Hossain L, Crawford J W, Wigand R T. Community detection in complex networks: multi-objective enhanced firefly algorithm. *Knowledge-Based Systems*, 2013, 46: 1–11
  24. Gandomi A H, Yang X S, Talatahari S, Alavi A H. Firefly algorithm with chaos. *Communications in Nonlinear Science and Numerical Simulation*, 2013, 18(1): 89–98
  25. Senapati M R, Dash P K. Local linear wavelet neural network based breast tumor classification using firefly algorithm. *Neural Computing & Applications*, 2013, 22(7–8): 1591–1598
  26. Fister I, Yang X S, Brest J, Fister I Jr. Modified firefly algorithm using quaternion representation. *Expert Systems with Applications*, 2013, 40(18): 7220–7230
  27. Kazem A, Sharifi E, Hussain F K, Saberi M, Hussain O K. Support vector regression with chaos-based firefly algorithm for stock market price forecasting. *Applied Soft Computing*, 2013, 13(2): 947–958
  28. Yang X S. Multiobjective firefly algorithm for continuous optimization. *Engineering with Computers*, 2013, 29(2): 175–184
  29. Yang X S. *Nature-Inspired Metaheuristic Algorithms*. Bristol: Luniver Press, 2008
  30. Yang X S. *Engineering Optimization: An Introduction with Metaheuristic Applications*. Wiley, 2010: 221–230
  31. Basu B, Mahanti G K. Firefly and artificial bees colony algorithm for synthesis of scanned and broadside linear array antenna. *Progress in Electromagnetic Research B*, 2011, 32: 169–190
  32. Yazdani A, Jayabarathi T, Ramesh V, Raghunathan T. Combined heat and power economic dispatch problem using firefly algorithm. *Frontiers in Energy*, 2013, 7(2): 133–139
  33. Holland J. *Adaptation in Natural and Artificial Systems*. Ann Arbor, USA: The University of Michigan Press, 1975
  34. Goldberg D E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley Educational Publishers Inc, 1989
  35. Younes M, Rahli M, Koridak L A. Optimal power flow based on hybrid genetic algorithm. *Journal of Information Science and Engineering*, 2007, 23: 1801–1816
  36. Michalewicz Z. A survey of constraint handling techniques in evolutionary computation methods. In: McDonnell J R, Reynolds R G, Fogel D B, eds. *Proceedings of the 4th Annual Conference on Evolutionary Programming*, MIT Press, 1995, 135–155
  37. Caorsi S, Massa A, Pastorino M. A computational technique based on a real-coded genetic algorithm for microwave imaging purposes. *IEEE Transactions on Geoscience and Remote Sensing*, 2000, 38(4): 1697–1708
  38. Oyama A, Obayashi S, Nakamura T. Real-coded adaptive range genetic algorithm applied to transonic wing optimization. *Applied Soft Computing*, 2001, 1(3): 179–187
  39. Niknam T, Narimani M R, Jabbari M, Malekpour A R. A modified shuffle frog leaping algorithm for multi-objective optimal power flow. *Energy*, 2011, 36(11): 6420–6432
  40. Saini A, Chaturvedi D K, Saxena A K. Optimal power flow solution: a GA-fuzzy system approach. *International Journal of Emerging Electric Power Systems*, 2006, 5(2): 1–21
  41. Ongsakul W, Tantimaporn T. Optimal power flow by improved evolutionary programming. *Electric Power Components and*



- Systems, 2006, 34(1): 79–95
42. Abido M A. Optimal power flow using tabu search algorithm. *Electric Power Components and Systems*, 2002, 30(5): 469–483
  43. Yuryevich J, Wong K P. Evolutionary programming based optimal power flow algorithm. *IEEE Transactions on Power Systems*, 1999, 14(4): 1245–1250
  44. Narimani M R, Azizipanah-Abarghooee R, Zoghdar-Moghadam-Shahrekohne B, Gholami K. A novel approach to multi-objective optimal power flow by a new hybrid optimization algorithm considering generator constraints and multi-fuel type. *Energy*, 2013, 49: 119–136
  45. Sailaja Kunari M, Maheswarapu S. Enhanced genetic algorithm based computation technique for multi-objective optimal power flow. *International Journal of Electrical Power & Energy Systems*, 2010, 32(6): 736–742
  46. Vaisakh K, Srinivas L R. A genetic evolving ant direction DE for OPF with non-smooth cost functions and statistical analysis. *Energy*, 2010, 35(8): 3155–3171
  47. Rezaei Adaryani M, Karami A. Artificial bee colony algorithm for solving multi-objective optimal power flow problem. *International Journal of Electrical Power & Energy Systems*, 2013, 53: 219–230
  48. Perez-Guerrero R E, Cedefio-Maldonado J R. Differential evolution based economic environmental power dispatch. In: *Proceedings of the 37th Annual North American Power Symposium*. Piscataway, USA: NJ IEEE Service Center, 2005, 191–197
  49. Wang L, Singh C. Balancing risk and cost in fuzzy economic dispatch including wind power penetration based on particle swarm optimization. *Electric Power Systems Research*, 2008, 78(8): 1361–1368