Research Article

Application of the Firefly Algorithm for Solving the Economic Emissions Load Dispatch Problem

Theofanis Apostolopoulos and Aristidis Vlachos

Department of Informatics, University of Piraeus, 80 Karaoli and Dimitriou Street, 18534 Piraeus, Greece

Correspondence should be addressed to Aristidis Vlachos, avlachos@unipi.gr

Efficient and reliable power production is necessary to meet both the profitability of power systems operations and the electricity demand, taking also into account the environmental concerns about the emissions produced by fossil-fuelled power plants. The economic emission load dispatch problem has been defined and applied in order to deal with the optimization of these two conflicting objectives, that is, the minimization of both fuel cost and emission of generating units. This paper introduces and describes a solution to this famous problem using a new metaheuristic nature-inspired algorithm, called firefly algorithm, which was developed by Dr. Xin-She Yang at Cambridge University in 2007. A general formulation of this algorithm is presented together with an analytical mathematical modeling to solve this problem by a single equivalent objective function. The results are compared with those obtained by alternative techniques proposed by the literature in order to show that it is capable of yielding good optimal solutions with proper selection of control parameters.

1. Introduction

Biology-inspired metaheuristic algorithms have recently become the forefront of the current research as an efficient way to deal with many NP-hard combinatorial optimization problems and non-linear optimization constrained problems in general. These algorithms are based on a particular successful mechanism of a biological phenomenon of Mother Nature in order to achieve optimization, such as the family of honey-bee algorithms, where the finding of an optimal solution is based on the foraging and storing the maximum amount of flowers' nectar [1]. A new algorithm that belongs in this category of the so-called nature inspired algorithms is the firefly algorithm which is based on the flashing light of fireflies.

Although the real purpose and the details of this complex biochemical process of producing this flashing light is still a debating issue in the scientific community, many

researchers believe that it helps fireflies for finding mates, protecting themselves from their predators and attracting their potential prey [1–4]. In the firefly algorithm, the objective function of a given optimization problem is associated with this flashing light or light intensity which helps the swarm of fireflies to move to brighter and more attractive locations in order to obtain efficient optimal solutions.

In this research paper we will show how the recently developed firefly algorithm can be used to solve the famous economic emissions load dispatch optimization problem. This hard optimization problem constitutes one of the key problems in power system operation and planning in which a direct solution cannot be found and therefore metaheuristic approaches, such as the firefly algorithm, have to be used to find the near optimal solutions. This optimization problem deals with allocating loads to power generators of a plant for minimum total fuel cost and emissions while meeting the power demand and transmission losses constraints. There are numerous variations of this problem which model the two objective functions and the constraints in many different ways.

Moreover, we will demonstrate how the firefly algorithm works and how this method can be easily adapted in order to solve this multiobjective optimization problem. Therefore, we will discuss why this method is sufficiently accurate and easy to implement for real-time operation and control of power systems. For the efficiency and validation of this algorithm, we will use, as an example, a sample realistic test system having six power generators. We will also compare the solutions obtained with the ones obtained by alternative optimization techniques that have been successfully applied by many scientists in order to solve these types of problems, such as the goal attainment SQP method, ant colony optimization, and particle swarm optimization [5–10].

The remainder of this paper is organized as follows: Section 2 gives a brief description of the multiobjective optimization and why this is important in our case. In Section 3, a mathematical formulation and description of the economic emissions load dispatch problem is given. Section 4 gives a brief description of the goal attainment SQP method which was used as an alternative way to solve an example test system of the economic emission load dispatch problem, while Section 5 briefly describes the Firefly algorithm in general. Section 6 presents the computational results simulated in Matlab and acquired when we applied both the firefly algorithm and the goal attainment SQP method to solve the economic emission load dispatch problem. Moreover, the efficiency of the firefly algorithm is measured by comparing its results with those obtained by other stochastic algorithms proposed by the researchers for different test systems of this problem. Finally, Section 7 provides some conclusions concerning the solutions obtained by the firefly algorithm and some suggestions and ideas for further research.

2. Multiobjective Optimization and Problem Formulation

Multiple conflicting objectives arise naturally in most real-world combinatorial optimization problems, such as the economic emissions load dispatch problem. Several principles and strategies have been developed and proposed for over a decade in order to solve these problems, some of which will be discussed in this section. In the multiobjective optimization problem, as its name implies, we have multiple objective functions with a possibility of conflict with each other. The aim is to find a vector of decision variables that satisfies constraints and optimizes (minimizes or maximizes) these functions. In such cases, we have to construct an overall objective function as a linear combination of the conflicting multiple objective functions using a weighting factor for each function. In a more precise mathematical

way, the multiobjective optimization problem can be defined as follows [1, 7, 10–14]:

$$\text{Find vector} \quad X = (x_1, x_2, \ldots, x_n)^T \in \Omega$$

$$\text{which optimizes} \quad f(x) = (f_1(x), f_2(x), \ldots, f_i(x)) \quad\quad\quad (2.1)$$

$$\text{subject to} \quad g_j(x) \geq 0 \quad \text{or} \quad g_j(x) \leq 0, \quad j = 1, 2, \ldots m,$$

where $f_1, f_2, \ldots, f_i$ denote the objective functions to be optimized simultaneously, X is the vector of discrete decision variables or search/decision space, Ω is the finite set of feasible solutions and $g_j(x)$ denotes the inequality constraints. The functions $f_i(x)$ and $g_i(x)$ may be linear or non-linear. The multiobjective optimization problem is sometimes called vector minimization problem.

2.1. Pareto Optimal Solutions

For any given problem having more than one objective functions, any two solutions x and y of this problem can have one of two distinct possibilities: one solution may dominate over the other or none of them dominates over the other, since there can be no solution vector X that minimizes all the i objective functions simultaneously. Therefore, we introduce the well known Pareto optimum solution concept in multiobjective optimization problems. A feasible solution X is called Pareto optimal solution if there exists no other feasible solution Y such that $f_i(Y) \leq f_i(X)$ for all $i = 1, 2, \ldots, k$ with $f_j(Y) < f_i(X)$ for at least one j, which means that the solution X is no worse than Y in all objective functions, and the solution X is strictly better than Y in at least one objective function [10, 11, 13, 14]. In other words, the solutions that are nondominated within the entire search space are denoted as Pareto optimal solutions and constitute the Pareto optimal set or Pareto optimal frontier (i.e., the image of the Pareto set in objective space). The knowledge of this set is crucial in many multiobjective optimization problems, as this enables the decision maker choose the best compromise solution [11, 13, 14].

As it is very difficult to effectively handle with all the conflicting objective functions, several methods have been developed for this purpose, such as the utility function method and the goal attainment method. In most of these methods, the multiobjective problem is transformed into a single-objective problem, then a set of Pareto optimal solutions is generated, and some additional criterion or rule to select one particular Pareto optimal solution is used as a solution of the multiobjective problem.

2.2. The Utility Function Method

In this paper, in order to apply and solve the economic emissions load dispatch problem to the firefly algorithm, we use the utility function method or weighting function method as an efficient way to deal with conflicting discrete goals and combine the two conflicting objective functions of the problem into one objective function. In this method, the problem is transformed into a single-objective function problem by using a utility function $U_i(f_i)$ for

each objective function based on its importance compared to the other objective functions. The overall utility function of the problem is defined as [12–14]

$$U = \sum_{i=1}^{k} U_i(f_i). \tag{2.2}$$

Thus, the solution vector X is found by maximizing the total utility function U subject to the constraints $g_j(x) \geq 0$, $j = 1, 2, \ldots m$. Then, the previous equation can be rewritten as

$$U = \sum_{i=1}^{k} U_i = \sum_{i=1}^{k} w_i f_i(X), \tag{2.3}$$

where w_i is a scalar weighting factor associated with the ith objective function and $w_1 + w_2 + \cdots + w_k = 1$. In case of $g_i(x) \leq 0$, then $w_1 + w_2 + \cdots + w_k = -1$.

Since the economic emissions load dispatch problem has two conflicting objective functions, the problem can be defined as $f(x) = a * f_1(x) + (1 - a) * f_2(x)$, where $f_1(x)$ and $f_2(x)$ represent these two conflicting objectives, and a is the weighting coefficient, as we will see in Section 6. However, in all the tables of this paper, we will not use the weights in the presentation of the values of the two objectives.

3. The Economic Emissions Load Dispatch Problem

The economic emission load dispatch problem is considered a hard optimization problem which normally renders classical exact optimization methods ineffective. For this reason, many approximate algorithms and particularly evolutionary methods such as artificial immune systems, genetic algorithms, cultural algorithms, particle swarm optimization, and differential evolution, have gained a lot of popularity for efficiently solving this problem [5, 6, 9, 10, 13, 15–18]. Furthermore, many hybrid methods in which two or more methodologies are combined together in order to enhance the final model have been introduced and attracted much attention by many researchers in the last few years [6, 8–10, 13, 15–17]. In fact, some of these proposed hybrid approaches are considered very effective, robust, and well promising so as to deal with other hard combinatorial optimization problems.

This problem is a multiple, conflicting objective function problem. Its primary objective is to determine the optimal quantity of both energy and emission objectives to reliably serve consumers. In particular, the main objective is the minimization of both the fuel cost of operation and emission of each power unit, subject to satisfying the equality and inequality constraints concerning operational limits of generations, the load demand, and the transmission losses of the facilities. Notably, in the recent years, many countries in the world have agreed to decrease the amount of pollutants fossil fuel power generating units taking into account the environmental concerns and protection policies about the emissions produced by these plants. However, this decrease of the total emission from a system plant implies an increase in the system operating cost, justifying the high applicability and paramount importance of this optimization problem. Therefore, the objective of this problem is to find out an operating point, which keeps a balance between cost and emission. This can be formulated mathematically as a minimization multiobjective problem as follows [5, 6, 8–10, 13, 15–19].

3.1. The Fuel Cost Objective

The aim is to minimize the total fuel cost (operating cost) of all committed plants can be stated as follows:

$$\min f_1(X) = \sum_{i=1}^{n} C_i(P_{Gi}) = \sum_{i=1}^{n} \left(\alpha_i + b_i P_{Gi} + c_i P_{Gi}^2 \right), \tag{3.1}$$

where n is the number of units power generators of a power plant, C_i is the fuel cost of the ith generator, P_{Gi} is the out power of generator i and a_i, and b_i and c_i are the fuel cost coefficients of the ith generator. Normally, the fuel cost equation $f_1(X)$ is expressed as continuous quadratic (higher order) equation, as we have here, but sometimes it can be expressed in linear form, when the coefficient c_i is equal to zero. However, in both cases, the equation expresses the variation of fuel cost ($ or Rs) with generated power or time (MW or hr). In our paper we use the $/hr as the only unit of measurement of the fuel cost function.

3.2. The Emissions Objective

The aim is to minimize the sum of all types of emissions from fossil power plants, namely, the gaseous particle pollutants, such as NOx, SOx, and CO_2, thermal and any other chemical emissions with suitable emission weights/coefficients on each pollutant power generator. In our research, without loss of generality, we will consider only one type of emission, the amount of NOx, which is formulated in the following quadratic equation:

$$\min f_2(X) = \sum_{i=1}^{n} \left[10^{-2} * \left(\alpha_i + \beta_i P_{Gi} + \gamma_i P_{Gi}^2 \right) + \zeta_i \exp(\lambda_i P_{Gi}) \right], \tag{3.2}$$

where, α_i, b_i, γ_i, ζ_i, and λ_i are the emission coefficients of NOx for the ith power generator. In our research, we use the unit of measurement of ton/hr for the emissions function (some research papers use kg/hr).

3.3. The Necessary Constraints of the Problem

The total power generation must satisfy the total required demand (power balance) and transmission losses. This can be formulated as follows:

$$\sum_{i=1}^{n} P_{Gi} = D + P_{\text{loss}}, \tag{3.3}$$

where D is the real total load demand of the system, P_{Gi} is the ith generator's power, and P_{loss} is the transmission losses. These can be determined from either the load/power flow or the matrix B_{ij} of coefficients. In this paper, only the B_{ij} coefficients are considered

$$P_{\text{loss}} = \sum_{i}^{n} \sum_{j}^{n} B_{ij} P_i P_j, \tag{3.4}$$

where, B_{ij} are the elements of the loss coefficient matrix B and P_i and P_j are the out powers of the ith and jth generator; respectively. In our paper, we use the MW as the only unit of measurement of the power balance constrain.

Apart from the total demand and transmission loss constrain, there is also the generator capacity constrain in which the power limits of each generator are formulated in order to have a stable operation of a plant. The upper and lower limits are defined as follows:

$$P_{Gi}^{\text{MIN}} \le P_{Gi} \le P_{Gi}^{\text{MAX}}, \quad \text{for } i = 1 \dots n, \tag{3.5}$$

where P_{Gi}^{MIN} and P_{Gi}^{MAX} are the lower and upper limit of the ith generator's out power P_{Gi}, respectively. The power load of each generator unit is measured in MW. As we will see later, in order incorporate these conditions into one equation, we will use an auxiliary (Lagrange) function $f_3(x)$.

4. The Goal Attainment SQP Method

The goal attainment SQP method is actually a hybrid method which combines the Goal Attainment (GA) method with the Sequential Quadratic Programming (SQP). In particular, a multiobjective non-linear optimization problem is initially reformulated by the goal attainment method so as it can be used by the Sequential Quadratic Programming method (SQP) in order to optimize it. This hybrid method has been successfully applied by many scientists as an effective way to solve many optimization problems. It constitutes a highly effective and state-of-the-art technique to obtain the best compromise solution in a multiobjective problem [7]. This method is already implemented in Matlab 2008 and belongs to the standard optimization toolbox multiobjective solver of Matlab. We will apply this method in a realistic test system, described in Section 6, in order to compare and contrast the solutions obtained by this alternative method and those obtained by the firefly algorithm. By this way, we will prove the effectiveness and robustness of the proposed algorithm.

4.1. The Goal Attainment Method

The main idea behind this method is to reduce a set of non-linear functions $f_i(X)$ below a set of goals of f_i^*. However, since all goals cannot be achieved, this set of functions is reformulated in an appropriate, well-defined way and enabling the designer to be relatively imprecise about the initial goals [6, 8, 14, 16]. In particular, in this method, we define a set of desired design goals $b = [b_1, b_2, \dots, b_n]^T$ which is associated with a set of objective functions $f(x) = \{f_1(x), f_2(x), \dots, f_n(x)\}$. The relative degree of underachievement or overachievement of the desired goals is controlled by a vector of weighting coefficients, $w = [w_1, w_2, \dots, w_n]$ for each objective function, which express the importance of the ith objective function relative to the other objective functions in meeting the respective ith goal. The goal b_i is found by first solving the single-objective function optimization problem [14]

$$\begin{aligned} \text{minimize} \quad & f_i(X), \ i = 1, 2, \dots, n \\ \text{subject to} \quad & g_j(X) \ge 0 \quad \text{or} \quad g_j(X) \le 0, \quad j = 1, 2, \dots m, \end{aligned} \tag{4.1}$$

where $X = (x_1, x_2, \ldots, x_n)^T \in \Omega$ are the design variables. In this case, the goal b_i is taken as the optimum value of the objective $f_i(X)$, that is, $b_i = f_i^* = f(X_i^*)$, where X_i^* denotes the solution of the problem defined in (4.1).

In order to find the best compromise solution of the problem, we introduce a scalar variable γ as a design variable in addition to n design variables x_i, for $i = 1, 2, \ldots, n$. Then, the problem is solved as a standard optimization problem using the following equation [6–8, 14]:

$$\text{Find} \quad \{x_1, x_2, \ldots, x_n\} \in \Omega, \quad \gamma \in R$$

$$\text{to minimize} \quad f(x_1, \ x_2, \ldots, \ x_{n,\gamma}) = \gamma$$

$$\text{subject to} \quad g_j(X) \geq 0 \quad \text{or} \quad g_j(X) \leq 0, \quad j = 1, 2, \ldots m, \quad f_i(X) - \gamma w_i \leq b_i$$

$$\text{(for } g_j(X) \leq 0) \quad \text{or} \quad f_i(X) + \gamma w_i \leq b_i \quad \text{(for } g_j(X) \geq 0), \quad i = 1, 2, \ldots, n,$$

$$(4.2)$$

where $X = (x_1, x_2, \ldots, x_n)^T \in \Omega$ are the design variables (parameters), γ is a scalar variable unrestricted in sign, and the weights $w = [w_1, w_2, \ldots, w_n] \in R^n$ satisfy the necessary normalization condition which in case of $g_j(X) \geq 0$ the condition is $w_1 + w_2 + \cdots + w_k = 1$ or in case of $g_i(x) \leq 0$ the condition is $w_1 + w_2 + \cdots + w_k = -1$.

Moreover, the term γw_i introduces the degree of slackness into the problem, which otherwise imposes the goals to be rigidly met. The weighting vector w enables the decision maker to quantitatively express a measure of the relative tradeoffs between the objectives [6–8]. For example, if we set the weighting vector equal to the initial goals, we assume equal underachievement and overachievement of the goals b_i. We can also include hard constraints into the design by setting a particular weighting factor to zero. In particular, if a component of vector w is equal to zero (i.e., some $w_i = 0$), this means that the maximum limit of an objective $f_i(X)$ is equal to b_i. We can easily see that a set of solutions can be generated by varying w over R^n even for nonconvex problems, and therefore the multiobjective optimization involves generating and selecting solutions that characterize the objectives, where the improvement of one objective causes a declination in another objective.

During this optimization procedure, the scalar variable γ is varied changing the size of the feasible region. It is interesting to mention that the optimum value of γ, which is also called attainment factor, will inform the decision maker whether the decision goals are attainable or not. A negative value of γ implies that the goal defined by the decision maker is attainable and an improved solution can be obtained. If the value of γ is positive, then the goal defined by the decision maker is unattainable, and no improvement of the solution is possible.

4.2. The Sequential Quadratic Programming Method

The Sequential Quadratic Programming (SQP) method constitutes one of the most popular, efficient, and accurate algorithms for non-linear continuous optimization which has been implemented and tested over a large number of several combinatorial optimization problems [7]. This method is very closely related to Newton's method for unconstrained optimization. In particular, at each iteration, an approximation local model of the problem is constructed and solved providing the necessary direction for finding the solution of the originally defined problem. In the case of an unconstrained minimization, only the objective function must be

approximated, then the local model is quadratic and the algorithm reduces to the famous Newton's method.

Given a problem defined as in (4.1), both the objective function and constraints have to be modeled. The objective function is formulated as a quadratic model while constraints are formulated as a linear model. Both of these models are based on the quadratic approximation of the Langragian function of the problem which is updated using a quasi-Newton updating method. The Langragian function of the problem defined in (4.1) is given as follows [1, 7, 14]:

$$L(X, \lambda) = f_i(X) - \sum_{j=1}^{m} \lambda_j * g_j(X), \tag{4.3}$$

where, λ_j is a Langrage multiplier, $g_j(X) \geq 0$ and for $i = 1, 2, \ldots, n$. In case of $g_j(X) \leq 0$, we have a plus sign instead of minus before the sum.

Now, we can define the quadratic programming model as follows [1, 7, 14]:

$$\min_{d} \quad L(x_k, \lambda_k) + \nabla L(x_k, \lambda_k)^T d + \frac{1}{2} d^T \nabla^2 L(x_k, \lambda_k) d,$$

$$\text{s.t.} \quad g(x_k) + \nabla g(x_k)^T d \geq 0 \quad \text{for } g(x_k) \geq 0, \tag{4.4}$$

where, x_k is the current estimate of a solution x^*, and d is the solution of the quadratic programming model which is used as a search direction for finding the appropriate solution of the given problem.

5. The Firefly Algorithm

5.1. Description

The Firefly Algorithm (FA) is a metaheuristic, nature-inspired, optimization algorithm which is based on the social (flashing) behavior of fireflies, or lighting bugs, in the summer sky in the tropical temperature regions [1–3, 20]. It was developed by Dr. Xin-She Yang at Cambridge University in 2007, and it is based on the swarm behavior such as fish, insects, or bird schooling in nature. In particular, although the firefly algorithm has many similarities with other algorithms which are based on the so-called swarm intelligence, such as the famous Particle Swarm Optimization (PSO), Artificial Bee Colony optimization (ABC), and Bacterial Foraging (BFA) algorithms, it is indeed much simpler both in concept and implementation [2–4, 20]. Furthermore, according to recent bibliography, the algorithm is very efficient and can outperform other conventional algorithms, such as genetic algorithms, for solving many optimization problems; a fact that has been justified in a recent research, where the statistical performance of the firefly algorithm was measured against other well-known optimization algorithms using various standard stochastic test functions [1–3, 20]. Its main advantage is the fact that it uses mainly real random numbers, and it is based on the global communication among the swarming particles (i.e., the fireflies), and as a result, it seems more effective in multiobjective optimization such as the economic emissions load dispatch problem in our case.

The firefly algorithm has three particular idealized rules which are based on some of the major flashing characteristics of real fireflies [2–4, 20]. These are the following: (1) all

fireflies are unisex, and they will move towards more attractive and brighter ones regardless their sex. (2) The degree of attractiveness of a firefly is proportional to its brightness which decreases as the distance from the other firefly increases due to the fact that the air absorbs light. If there is not a brighter or more attractive firefly than a particular one, it will then move randomly. (3) The brightness or light intensity of a firefly is determined by the value of the objective function of a given problem. For maximization problems, the light intensity is proportional to the value of the objective function.

5.2. Attractiveness

In the firefly algorithm, the form of attractiveness function of a firefly is the following monotonically decreasing function [2, 3, 20]:

$$\beta(r) = \beta_0^* \exp(-\gamma r^m), \ \text{with} \ m \geq 1, \tag{5.1}$$

where, r is the distance between any two fireflies, β_0 is the initial attractiveness at $r = 0$, and γ is an absorption coefficient which controls the decrease of the light intensity.

5.3. Distance

The distance between any two fireflies i and j, at positions x_i and x_j, respectively, can be defined as a Cartesian or Euclidean distance as follows [2, 3, 20]:

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^2}, \tag{5.2}$$

where $x_{i,k}$ is the kth component of the spatial coordinate x_i of the ith firefly and d is the number of dimensions we have, for $d = 2$, we have

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \tag{5.3}$$

However, the calculation of distance r can also be defined using other distance metrics, based on the nature of the problem, such as Manhattan distance or Mahalanobis distance.

5.4. Movement

The movement of a firefly i which is attracted by a more attractive (i.e., brighter) firefly j is given by the following equation [2, 3, 20]:

$$x_i = x_i + \beta_0 * \exp\left(-\gamma r_{ij}^2\right) * (x_j - x_i) + a * \left(\text{rand} - \frac{1}{2}\right), \tag{5.4}$$

where the first term is the current position of a firefly, the second term is used for considering a firefly's attractiveness 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 α is a randomization parameter determined by the problem of interest, while rand is a random number generator uniformly distributed in the space [0,1]. As we will see in this implementation of the algorithm, we will use $\beta_0 = 1.0$, $\alpha \in [0,1]$ and the attractiveness or absorption coefficient $\gamma = 1.0$, which guarantees a quick convergence of the algorithm to the optimal solution.

5.5. Convergence and Asymptotic Behavior

The convergence of the algorithm is achieved for any large number of fireflies (n) if $n \gg m$, where m is the number of local optima of an optimization problem [1, 3]. In this case, the initial location of n fireflies is distributed uniformly in the entire search space. The convergence of the algorithm into all the local and global optima is achieved, as the iterations of the algorithm continue, by comparing the best solutions of each iteration with these optima. However, it is under research a formal proof of the convergence of the algorithm and particularly that the algorithm will approach global optima when $n \rightarrow \infty$ and $t \gg 1$ [3]. In practice, the algorithm converges very quickly in less than 80 iterations and less than 50 fireflies, as it is demonstrated in several research papers using some standard test functions [1–3, 20]. Indeed, the appropriate choice of the number of iterations together with the γ, β, α, and n parameters highly depends on the nature of the given optimization problem as this affects the convergence of the algorithm and the efficient find of both local and global optima. Note that the firefly algorithm has computational complexity of $O(n^2)$, where n is the population of fireflies. The larger population size becomes the greater the computational time is [1–3].

5.6. Special Cases

There are two important special cases of the firefly algorithm based on the absorption coefficient γ; that is, when $\gamma \rightarrow 0$ and $\gamma \rightarrow \infty$ [1, 3, 20]. When $\gamma \rightarrow 0$, the attractiveness coefficient is constant $\beta = \beta_0$, and the light intensity does not decrease as the distance r between two fireflies increases. Therefore, as the light of a firefly can be seen anywhere, a single local or global optimum can be easily reached. This limiting case corresponds to the standard Particle Swarm Optimization (PSO) algorithm. On the other hand, when $\gamma \rightarrow \infty$, the attractiveness coefficient is the Dirac delta function $\beta(r) \rightarrow \delta(r)$. In this limiting case, the attractiveness to light intensity is almost zero, and as a result, the fireflies cannot see each other, and they move completely randomly in a foggy place. Therefore, this method corresponds to a random search method.

5.7. Hybridization

In a recent bibliography, a new metaheuristic algorithm has been developed and formulated based on the concept of hybridizing the firefly algorithm. In particular, the new Levy flight Firefly algorithm was developed by Dr. Xin-She Yang at Cambridge University in 2010 and it combines the firefly algorithm with the Levy flights as an efficient search strategy [4]. It combines the three idealized rules of the firefly algorithm together with the characteristics of Levy flights which simulate the flight behavior of many animals and insects. In this algorithm,

the form of the attractiveness function and the calculation of distance between two fireflies are the same as in firefly algorithm, but in the movement function, the random step length is a combination of the randomization parameter together with a Levy flight. In particular, the movement of a firefly is a random walk, where the step length is drawn by the Levy distribution [4].

6. Case Study

6.1. Application Example

In order to apply the firefly algorithm to the economic emissions load dispatch problem, we need to effectively deal with the necessary constraints of the problem and the fact that the firefly algorithm can directly solve only maximum optimization problems, not minimization problems. This is expected as the light intensity or brightness of a firefly at a particular location x can be chosen as analogous or equal to the value of the objective function at location x.

For this reason, in order to avoid the violation of constraints, which could cause infeasible solutions, we have converted the constrained optimization problem into an unconstrained problem by penalizing infeasible solutions of the objective function, instead of repairing them, which is very computationally expensive in our case [1, 11, 14]. Therefore, we used the method of Lagrange multipliers as an efficient strategy for finding the maxima (orminima) of the given function subject to the constraints of total required demand and transmission losses (Section 3.3). The Lagrange multiplier we used as a weighting factor in the Lagrange (constraints) function $f_3(X)$ is 1000. This value was chosen experimentally as a parameter that maximizes $f_3(X)$. Furthermore, without loss of generality, in an optimization problem, the minimum of a function can be found by seeking the maximum of the negative of the same function, or alternatively, we can subtract from that function a large positive number which is usually the global optimum of this function [1, 11, 12, 14]. In this problem we adopt the first choice, where we maximize the negative form of the objective function. Finally, we use $w_1 = w_2 = a = 0.5$ as a weighting factor for each objective function, since both of the objectives (i.e., fuel cost and emissions) have equal importance in the given problem, and therefore, they have to contribute equally to the formulation of the objective function. Moreover, the choice of equal weights is also justified by some experiments we conducted with different weights, in which we did not achieve better results, as the ranges of the values of the two objective functions $f_1(X)$ and $f_2(X)$ are very different (Figures 3 and 6). Now the problem can be formulated as follows:

$$\max -f(X) = \max -0.5 * f_1(X) - 0.5 * f_2(X) - f_3(X), \tag{6.1}$$

where $f_1(X)$ is the fuel cost objective function defined in the following equation (in \$/hr):

$$f_1(X) = \sum_{i=1}^{6} \left(\alpha_i + b_i P_{Gi} + c_i P_{Gi}^2 \right), \tag{6.2}$$

Table 1: Generator cost, emission, and capacities coefficients of the test problem we will solve with the firefly algorithm.

		G1	G2	G3	G4	G5	G6
	a	100	120	40	60	40	100
cost	b	200	150	180	100	180	150
	c	10	10	20	10	20	10
	α	4.091	2.543	4.258	5.326	4.258	6.131
	β	−5.554	−6.047	−5.094	−3.55	−5.094	−5.555
emission	γ	6.490	5.638	4.586	3.38	4.586	5.151
	ζ	$2e-4$	$5e-4$	$1e-6$	$2e-3$	$1e-6$	$1e-5$
	λ	2.857	3.333	8	2	8	6.667
power	$P_{\min}$	0.05	0.05	0.05	0.05	0.05	0.05
	$P_{\max}$	0.5	0.6	1.0	1.2	1.0	0.6

where $f_2(X)$ is the emissions objective function defined in the following equation (in ton/hr):

$$f_2(X) = \sum_{i=1}^{6} \left[10^{-2} * \left(\alpha_i + \beta_i P_{Gi} + \gamma_i P_{Gi}^2 \right) + \zeta_i \exp(\lambda_i P_{Gi}) \right], \tag{6.3}$$

and $f_3(X)$ is the constraints equation which is defined as follows (in MW):

$$f_3(X) = 1000 * \mathrm{abs} \left(\sum_{i=1}^{6} P_{Gi} - D - \sum_{i=1}^{6} \sum_{j=1}^{6} B_{ij} P_{Gi} P_{Gj} \right), \tag{6.4}$$

It can be easily seen that the generalized form of the objective function given in (6.1) is non-linear, and the number of equality and inequality constraints of the problem increase with the size of the power test system we use. Therefore, the application of conventional straightforward optimization techniques, such as the gradient-based methods, is highly unsuccessful as the solution of the non-linear objective functions with many inequality constraints of the problem is very computationally expensive and the search space inefficiently large to explore.

In this test problem, we consider a plant with 6 power generators. The test system was derived from [10]. The generation data are given in Tables 1 and 2. The total power system load demand D is 1 MW. In particular, the values of the fuel cost and emission coefficients together with the power limits of each generator are given in Table 1 of the next page.

In order to test and demonstrate the efficiency of the algorithm and how it deals with different problem complexities and tradeoff surfaces we have considered two different cases. In case A the test problem is considered as lossless; that is, the transmission power loss matrix B is not considered. In case B, the power loss is taken into account and the transmission power loss coefficients matrix B is defined in Table 2 of the following page. Notice that the power loss coefficients matrix B is not symmetric (i.e., $B_{46} \neq B_{64}$), which depicts a realistic sample system for application. The reported results of both of these cases are discussed and analyzed in the following Section 6.3.

Table 2: The transmission losses matrix B of the test problem we apply.

Power loss coefficients					
0.001382	−0.000299	0.000044	−0.000022	−0.000010	−0.000008
−0.000299	0.000487	−0.000025	0.000004	0.000016	0.000041
0.000044	−0.000025	0.000182	−0.000070	−0.000066	−0.0066
−0.000022	0.000004	−0.000070	0.000137	0.000050	0.0033
−0.000010	0.000016	−0.000066	0.000050	0.000109	0.0005
−0.000008	0.000041	−0.000066	0.000033	0.000005	0.02444

6.2. Application of the Proposed Firefly Algorithm

In order to solve the economic emissions load dispatch problem, we have implemented the firefly algorithm in Matlab 2008 and it was run on a portable computer with an Intel Core2 Duo (1.8 GHz) processor, 2 GB RAM memory and MS Windows 7 as an operating system. Mathematical calculations and comparisons can be done very quickly and effectively with Matlab and that is the reason that the proposed Firefly algorithm was implemented in Matlab 2008 programming environment. In this proposed method, we represent and associate each firefly with a valid power output (i.e., potential solution) encoded as a real number for each power generator unit, while the combined fuel cost and emission objective (i.e., the objective function of the problem) is associated and represented by the light intensity of the fireflies. In this simulation, the values of the control parameters are: $\alpha = 0.2$, $\gamma = 1.0$, $\beta_0 = 1.0$, and $n = 12$, and the maximum generation of fireflies (iterations) is 50. The values of the fuel cost, the emission coefficients, the power limits of each generator, the power loss coefficients, and the total power load demand are supplied as inputs to the firefly algorithm. The power output of each generator, the total system power, the fuel cost, and the emissions with/without transmission losses are considered as outputs of the proposed Firefly algorithm. The pseudocode of this algorithm can be summarized in Pseudocode 1 of the next page [1, 3, 20].

Pseudocode 1 gives a brief description and a practical application of the proposed algorithm for the economic emission load dispatch problem. Initially, the objective function of the given problem is formulated as defined in (6.1) and it is associated with the light intensity of the swarm of the fireflies. The initial solution of the given problem is generated based on the mathematical formulation given below:

$$x_j = \text{rand} * (\text{upper-range} - \text{lower-range}) + \text{lower-range}, \tag{6.5}$$

where x_j is the new solution of jth firefly, that is, created, rand is a random number generator uniformly distributed in the space [0,1], while upper range and lower range are the upper range and lower range of the jth firefly (variable), respectively.

After the evaluation of the initial population/generation (i.e., solution), the firefly algorithm enters its main loop which represents the maximum number of generations of the fireflies. This is actually the termination criterion that needs to be satisfied for the termination of the loop.

Input: $\alpha, \gamma, \beta_0, n$, MaxGeneration, B, cost-emission-capacities coefficients
Output: P_{Gi} for $i = 1, \ldots 6$, $f(X)$, $f_1(X)$, $f_2(X)$

Begin of algorithm
Define the multiobjective function: max $-f(P_{Gi})$, with $i = 1, \ldots, 6$
Generate initial population of fireflies $n = 1, \ldots, 12$ %generate $n = 12$ initial solutions
Light Intensity of firefly n is determined by objective function, $I_n \sim f(P_{Gi})$
Define $\alpha = 0.2$, $\beta_0 = 1.0$ and $\gamma = 1.0$ %necessary algorithm's parameters
While ($t \leq$ MaxGeneration = 50)
   For $i = 1 : 12$ %for all fireflies (solutions)
     For $j = 1 : 12$ %for all fireflies (solutions)
       If ($I_i < I_j$)
       Then move firefly i towards firefly j (move towards brighter one)
       Attractiveness varies with distance r_{ij} via $\exp(-\gamma r_{ij})$
       Generate and evaluate new solutions and update Light Intensity
     End for j loop
   End for i loop
Check the ranges of the given solutions and update them as appropriate
Rank the fireflies, find and display the current best %max solution for each iteration
End of while loop
% Post-process results and visualization
Find the firefly with the highest Light Intensity among all fireflies %optimal solution
Plot the increase of the Light Intensity with time\iterations
Plot the two objectives with time %best solution with time
End of algorithm

Pseudocode 1: The proposed implemented firefly algorithm.

Repeat for each firefly i
If $P_{Gi} < P_i$ min
   Then $P_{Gi} = P_i$ min

If $P_{Gi} > P_i$ max
   Then $P_{Gi} = P_i$ max

End of Repeat

Pseudocode 2: Repair procedure for the generator capacity constraints.

The generation of a new solution (i.e., the movement of a firefly) of the given problem is made based on the following mathematical formulation:

$$x_i = x_i + \beta_0 * \exp\left(-\gamma * \sum_{i,j=1}^{6} (x_i - x_j)^2\right) * (x_j - x_i) + a * \left(\text{rand} - \frac{1}{2}\right), \qquad (6.6)$$

where x_i is the current solution of the ith firefly and x_j is the current (optimal) solution of jth firefly. The values of the algorithm's control parameters is $\alpha = 0.2$, $\gamma = 1.0$, $\beta_0 = 1.0$, and rand is a random number which is uniformly distributed in the space $[0,1]$. As we can see the distance between two fireflies is calculated using the Euclidean distance (Section 5.3) and the

generation of a new solution is actually a sum of the current solution (x_i), the metric of the evaluation of the current solution based on the current optimal solution (Euclidian metric), and a random step/move of the algorithm (Section 5.4).

After the generation of the new solutions, we have to apply the generator capacity constraints so as the new solutions are within the given operational power ranges. To avoid such violation, a repair process is applied to each solution (firefly) in order to guarantee that the generated power outputs are feasible. P_{Gi}, P_i min and P_i max denote the current, the minimum, and the maximum power outputs of the ith unit, which is associated with the ith firefly. The implemented procedure is shown in Pseudocode 2 of the previous page.

Finally, it is notable that for each generation (iteration), the swarm of 12 fireflies is ranked based on their light intensity, and the firefly with the maximum light intensity (i.e., the solution with the higher objective function value) is chosen as the brighter one (i.e., it is a potential optimal solution), while the others are updated based on (6.6). In the final iteration, the firefly with the brighter light intensity among the swarm of 12 fireflies is chosen as the brightest one which represents the optimal solution of the problem.

6.3. Results and Discussion

The main characteristic feature of the firefly algorithm is the fact that it simulates a parallel independent run strategy, where in every iteration, a swarm of n fireflies (12 in our case) has generated n solutions. Each firefly works almost independently, and as a result the algorithm, will converge very quickly with the fireflies aggregating closely to the optimal solution [1, 3, 20]. In our case, we can see that the total number of 600 function evaluations (12 fireflies ∗ 50 generations) is sufficient, and as a result, the algorithm stably converges to the optimal solution very quickly (approximately from the 10th generation/iteration). This algorithm also differs from other alternative approaches in the selection procedure in which each firefly constructs its own solution based on a weighted sum of the objective functions, where the weight attached to multiobjective criteria is constant. It is also observed that the proposed implementation of the firefly algorithm is very fast and predicts accurate results while satisfying inequality generator capacity constraints at various power load levels. It also offers a considerable saving in computer memory. The algorithm generated the optimal solution in less than 5 seconds, on the personal computer we used with specifications discussed in the previous section. From the results we obtained, it is clear that the firefly algorithm gives a global optimum in a computation time of less than 3 seconds, which is much lower than the time of other alternative techniques, such as the goal attainment SQP method, where the computation time was approximately 5 seconds. From the simulation results, as we show in Table 3 of the next page, we can conclude that the algorithm is very fast, efficient, and computational inexpensive in finding the Pareto optimal solutions of the given combinatorial optimization problem.

6.3.1. Application to the Test System

The results in terms of values of the power losses, the best fuel cost, and the best emission objectives of both the proposed firefly algorithm and the goal attainment SQP method can be summarized in the following Table 3 of the next page. Two different cases (A and B) have been considered as discussed earlier, and in both cases, the proposed algorithm has been demonstrated through a sample test system consisting of six power generators with load

Table 3: Solutions compromising minimum fuel cost and emission for $D = 1\,\text{MW}$ and $a = 0.5$.

	Firefly algorithm		Goal attainment SQP method	
	Case A	Case B	Case A	Case B
PG1 (MW)	0.050000	0.050000	0.05000	0.05000
PG2 (MW)	0.147177	0.147177	0.23670	0.23770
PG3 (MW)	0.099979	0.099979	0.07250	0.07270
PG4 (MW)	0.434936	0.434936	0.32520	0.33010
PG5 (MW)	0.214481	0.214481	0.07250	0.07190
PG6 (MW)	0.052670	0.052670	0.24310	0.23910
Obj. function	302.552853	302.709382	301.633115	301.625363
Fuel cost	603.354361	603.354361	603.034045	603.018459
Emission	0.236339	0.236339	0.232184	0.232266
PLoss	0.000000	0.000158	0.000000	0.001596

demand D of $1\,\text{MW}$ and equal weights of the two objectives ($w_1 = w_2 = a = 0.5$) with and without transmission losses B.

The Obj. function in Table 3 represents the value of the single-objective function of the problem, as defined in (6.1), which is used by the firefly algorithm in order to find the true Pareto optimal solution. The values of the fuel cost and emission objectives have been calculated directly from (6.2) and (6.3) without using weights.

According to the analysis of the simulation results, the firefly algorithm generated a good quality of Pareto set of solutions in contrast to the goal attainment SQP method. Using this algorithm, we can find the global minimum of the problem in about 10 iterations for a population of 12 fireflies. Thus, the total number of function evaluations is just 120. This is indeed very efficient and robust.

It is also worth to mention that in Table 3, the amount of transmission losses PLoss is very low, and as a result, it does not seem to affect the finding of the Pareto optimal solution using the proposed firefly algorithm since both the Pareto optimal solutions and the fuel cost and emission objectives remain the same. The small increase in the objective function of the problem (Obj. function) is expected as we have penalized infeasible solutions, based on the problem constraints, into the single-objective function of the problem. It is notable, however, that the transmission losses slightly affect the Pareto optimal solution in the goal attainment method.

In the goal attainment SQP method, we gave the following input parameters: $X_0 = [0.1, 0.1, 0.1, 0.1, 0.1, 0.1]$ as initial values of power generators, weight = [0.5 0.5] as a weight of the two objectives, and goal = [603 0.2] as an initial solution of the two objectives for the algorithm. We should consider that the goals are usually found empirically by first solving every single objective of the optimization problem individually, which constitutes higher initial values than those given here. However, in this case, we give as goals the solutions we obtained by the firefly algorithm as an effective way to test the robustness of these solutions and whether these could be further improved. It is also notable that the goal attainment method gave positive attainment factor ($\gamma = 0.0644$ and 0.0645 for transmission losses) which implies that the goal defined is unattainable, and as a result, no further improvement of the solution is possible. Therefore, the solution obtained by the firefly algorithm is both robust and efficient.

A graphical representation of the convergence of the objective function of the problem (with equal weights $w_1 = w_2 = a = 0.5$) given in (6.1) with time (iterations) is shown
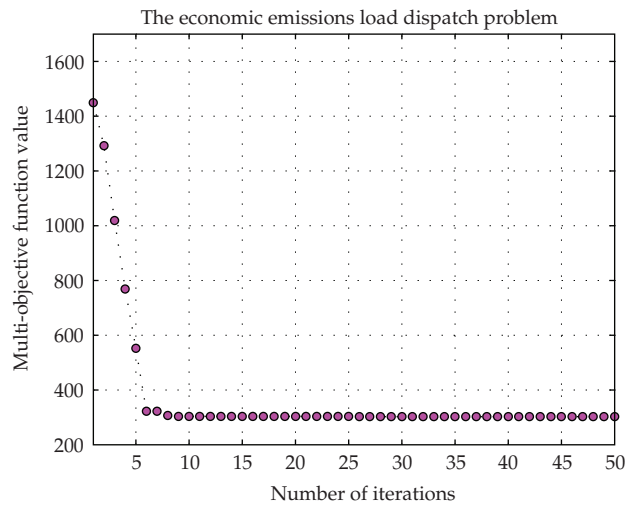
The economic emissions load dispatch problem

Figure 1: Convergence of the single-objective function of the problem with time (iterations).
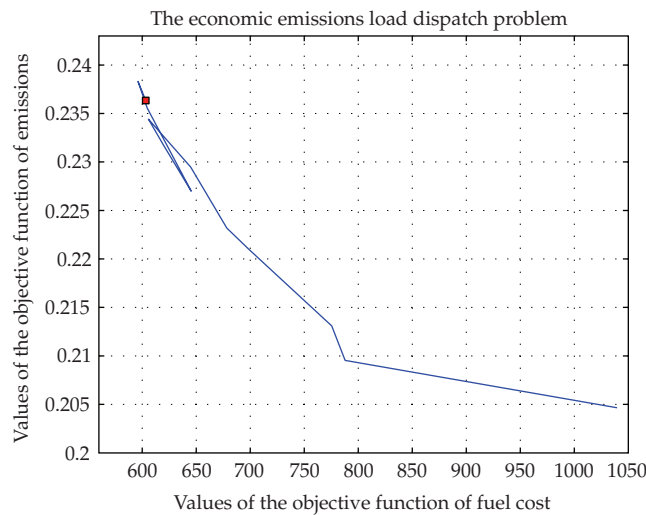
The economic emissions load dispatch problem

Figure 2: Convergence of fuel cost and emission objectives.

above in Figure 1. Figures 2 and 3 show the convergence of fuel cost and emission objectives when optimized and their convergence through time (iterations) respectively. The red point in Figure 2 represents the Pareto optimal solution found by the firefly algorithm in Table 3. Moreover, in Figure 3 we can see that the decrease of fuel cost causes the increase of emission and vice versa, since these are conflicting objectives and cannot be minimized simultaneously. The aim of this optimization problem is to find a good balance between them by converting the biobjective function into a single-objective function with different weights for each objective.
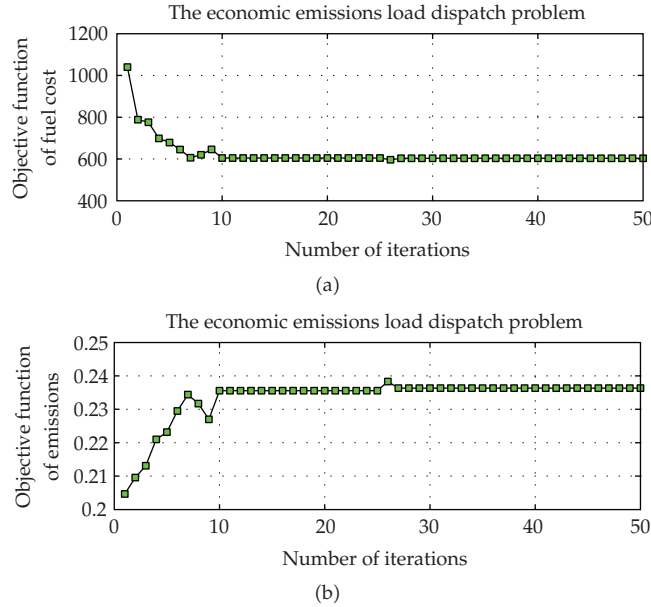
(a)

(b)

Figure 3: Convergence of the fuel cost and emission objectives with time (iterations).

6.3.2. Comparison with Other Optimization Algorithms

Recently, Abd El-Wahed et al. [10] used an efficient hybrid ant colony optimization and modified simplex method in order to solve the economic emissions load dispatch problem with random weighting factors [10]. The test system they used is the same as those described in this paper, with the same emission/cost coefficients as defined in Tables 1 and 2 and total load demand D = 2.8 MW. The best Pareto optimal solution found is at $x_* \approx$ (0.135, 0.411, 0.467, 1.00, 0.462, 0.357), which corresponds to $f_{1*} \approx$ 891.1966 and $f_{2*} \approx$ 0.2192. This means that the lowest fuel cost is about 891.1966 \$/hr, while the emissions are 0.2192 ton/hr. The transmission losses are PLoss = 0.0035. However, it is notable a minor mistake which gives erroneously that the lowest fuel cost is about 603.3 \$/hr instead of 891.1966 \$/hr which is the correct value. The fuel cost and emission values can be easily verified if we apply the given solutions to (6.2) and (6.3) or if we use the same equations given in [10]. Applying the firefly algorithm to the same problem, we have found a slightly better solution but with higher power losses. The Pareto optimal solution we obtained with 12 fireflies after 50 iterations is at $x_* \approx$ (0.050000, 0.511433, 0.215548, 0.935979, 0.581323, 0.507463), which corresponds to $f_{1*} \approx$ 881.533506 and $f_{2*} \approx$ 0.222158 and transmission losses PLoss = 0.007614. This clearly shows that the firefly algorithm is very efficient and effective. However, obviously, further applications are highly needed to see how it may behave for solving various similar tough engineering optimization problems, such as the economic, rapid and environmental power dispatch problem [13]. The following three Figures show the stable convergence of the objective function of the problem given in (6.1) with time (Figure 4), the convergence of fuel cost and emission objectives when optimized (Figure 5) and their quick convergence through time (Figure 6). Notice again the red point in Figure 5, which represents the Pareto optimal solution found and the fact that the two objectives are conflicting and they cannot be minimized simultaneously (Figure 6).
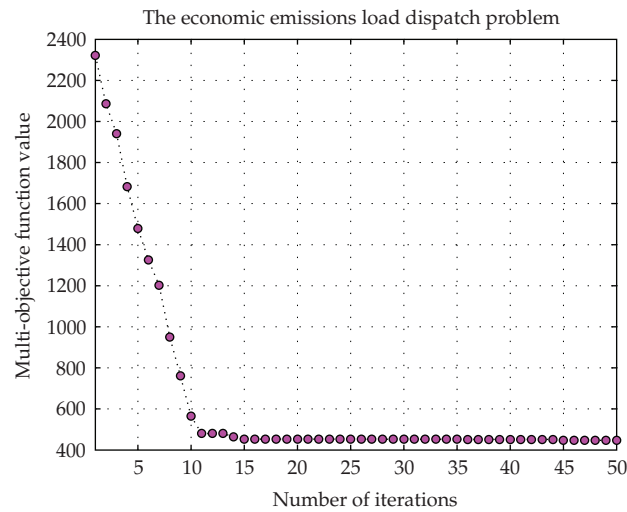
Figure 4: Convergence of the single-objective function of the problem with time (iterations).
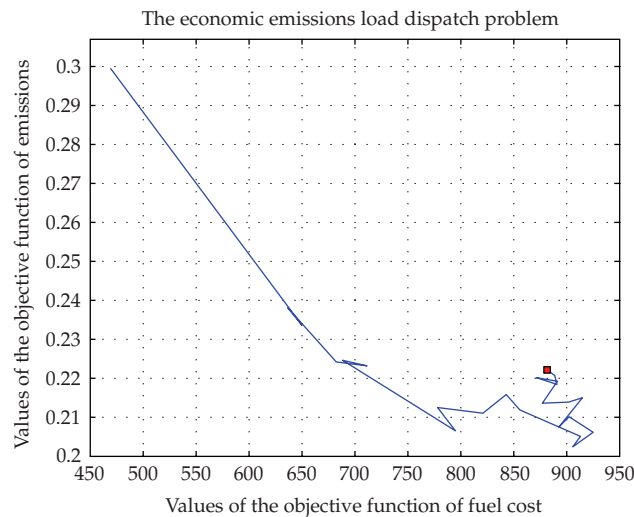

Figure 5: Convergence of fuel cost and emission objective.

In another research article, Kumar et al. [5] used an improved version of particle swarm optimization algorithm in order to solve the economic emissions load dispatch problem for a test system of 6 power generators, for several load demands, with transmission losses, using a price penalty factor h [5]. The equations used in that article for the problem formulation constitute a slight variation of those defined in this paper.

In order to apply the firefly algorithm to this example problem, we have slightly changed the firefly algorithm. In particular, there is a change of the generation of the initial solution since the difference of the ranges of each solution (power generator) is relatively large. For the case of load demand equal to 900 MW, the initial solution is generated using
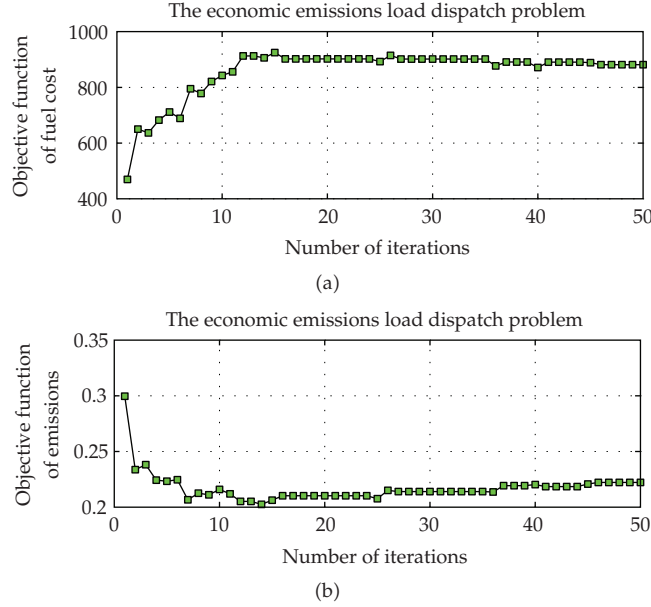
(a)

(b)

Figure 6: Convergence of the fuel cost and emission objectives with time (iterations).

the same equation (6.5), while in the other two cases the initial solution is created based on the following two equations:

$$\text{for } D = 500\,\text{MW}; \quad x_j = \text{rand} * \frac{\text{up-range} - \text{low-range}}{4} + \text{low-range}, \tag{6.7}$$

$$\text{for } D = 700\,\text{MW}; \quad x_j = \text{rand} * \frac{(\text{up-range} - \text{low-range})}{2} + \frac{(\text{up-range})}{3}, \tag{6.8}$$

where x_j is the new solution of jth firefly that is created, rand is a random number generator uniformly distributed in the space [0,1], while up range and low range are the upper range and lower range of the jth firefly (power generator), respectively.

The results of that research paper including the generation cost, emission level, and power losses can be summarized in Table 4, while Table 5 presents the results obtained when we applied the firefly algorithm to the same test system.

From Tables 4, 5 (next page) we can see that the firefly algorithm gives good quality of Pareto optimal solutions for the given test problem and in some cases it outperforms the improved version of particle swarm optimization. In particular, in the case of $D = 500\,\text{MW}$ the Firefly algorithm obtains lower fuel cost, emissions and power losses than those found by the particle swarm optimization. In the other two cases the firefly algorithm managed to obtain a lower value in one of the two objectives, while in all three cases the algorithm achieved lower power losses.

It is also notable that the firefly algorithm solves this test problem by converting the multiobjective problem to a single-objective problem by a linear combination of different objectives as a weighted sum, while the particle swarm optimization introduces a price penalty factor h for the same purpose. Moreover, by using a population of solutions (fireflies)

Table 4: Minimum fuel cost and emission for various load demands using new particle swarm optimization [5].

	New particle swarm optimization algorithm [5]		
	$D = 500\,\text{MW}$	$D = 700\,\text{MW}$	$D = 900\,\text{MW}$
PG1 (MW)	34.5469	62.1411	90.8027
PG2 (MW)	28.1660	61.8393	100.0541
PG3 (MW)	90.4942	120.3419	153.8375
PG4 (MW)	91.4414	119.1079	148.0631
PG5 (MW)	130.000	178.2963	220.7988
PG6 (MW)	134.2356	175.3215	214.3293
Fuel cost	27640	37504	48358
Emission	262.6220	439.5522	693.8518
PLoss	8.8848	17.0481	27.8856

Table 5: Minimum fuel cost and emission for various load demands using the firefly algorithm.

	Firefly algorithm		
	$D = 500\,\text{MW}$	$D = 700\,\text{MW}$	$D = 900\,\text{MW}$
PG1 (MW)	46.940406	72.997450	102.24677
PG2 (MW)	55.668856	56.099051	149.58352
PG3 (MW)	63.889521	164.74354	156.58499
PG4 (MW)	73.894725	129.65927	142.96344
PG5 (MW)	130.000000	151.09698	200.92360
PG6 (MW)	125.000000	125.00000	145.97127
Fuel cost	27414.977	37072.477	48652.824
Emission	259.488314	447.59928	664.78570
PLoss	7.910586	14.865361	23.263366

in its search, multiple Pareto optimal solutions can be found more quickly, even in one run, in contrast to particle swarm optimization algorithm where each agent (particle) corresponds to one single solution of the problem. Finally, it is important to point out that the firefly algorithm converges in an acceptable time, which for this test system was approximately 3 seconds.

In general, the analysis of the experimental results has demonstrated that the firefly algorithm performs better than other methods used for the same problem, or at least it obtains good quality Pareto optimal solutions in significantly low computing times. It is characterized by a stable and fast convergence compared to other conventional methods and good computation efficiency, as it has been demonstrated by its application. This much improved speed of computation allows for additional searches and improvements that could be made in order to increase the confidence and efficiency of the generated solutions. As we have seen, one such improvement is the generation of the initial solution. Overall, the firefly algorithm is shown to be very helpful and promising in solving optimization problems in power systems although more example problems are necessary in order to prove its efficiency and superiority towards other alternative optimization algorithms used in the same domain, such as the particle swarm optimization and ant colony optimization algorithms.

7. Conclusions and Future Work

There is no doubt that the firefly algorithm, developed by Dr. Xin-She Yang in 2007, is a very powerful novel population-based method for solving constrained optimization problems and particularly NP-hard problems. The idea behind this algorithm is that the social behavior and especially the flashing light of fireflies can be easily formulated and associated with the objective function of a given optimization problem [1, 3, 20].

In this paper, we have proposed, presented, and tested the recently developed Firefly algorithm for application to the multiobjective minimization problem of economic emissions load dispatch. Focus was on the reduction of a single pollutant nitrogen oxide, NOx, while the transmission losses, the equality constraint of power balance, and the inequality generator capacity constraints are also considered. The results of the implementation of this proposed method clearly showed the efficiency and effectiveness of the firefly algorithm for solving the particular optimization problem and finding the true Pareto optimal solutions. The algorithm achieved good results comparable to those achieved by other stochastic nature-inspired algorithms as these reported in the literature.

For various load demands of two different test systems consisting of 6 power generators the simulation results of the algorithm compared favorably with other competitive state-of-the-art algorithms, and as a result, we can say that the firefly algorithm is very efficient and accurate in obtaining global optima with high success rates for the given constrained optimization problem. In particular, the results indicate that the algorithm obtained good quality Pareto optimal solutions which are better or at least equal to the solutions obtained by other stochastic alternative optimization algorithms such as the goal attainment SQP method, the particle swart optimization, and the Genetic algorithms, in terms of efficiency and success rates.

In all cases, the algorithm converged to the optimal solution very quickly (mainly from the 10th iteration), and its computation time in the simulations we run was less than 3 seconds. Moreover, the fact that the proposed algorithm is very simple in concept and easy to implement clearly implies that it could also be effectively applied to other multiobjective optimization problems and especially to NP-hard combinatorial optimization problems, such as the vehicle-routing problem.

However, from the simulation results, it seems that the proper selection of the population size, the number of generations (iterations), and the absorption coefficient is of paramount importance for the convergence of the algorithm as this heavily depends on the nature of the applied problem. Moreover, a refinement and improvement of the generation of the initial solution and the decreasing random step size seem very promising and beneficial to further increase the algorithm's performance, while it might also be possible to hybridize the algorithm together with other heuristic search methods for better results, such as the Levy flight firefly algorithm [4]. Furthermore, future comparison studies with other methods are necessary so as to identify the strengths and weaknesses of the current metaheuristic algorithm and prove its superiority and effectiveness. In future works, we intend to analyze the behavior of the proposed algorithm in other variations of the economic emissions load dispatch problem, such as the environmental power load dispatch problem with more security constraints [13]. Ultimately, even better optimazation algorithms may emerge as a promising way to solve NP-hard optimization problems.

We hope that this paper will provide the necessary background for further improvement of optimization algorithms and will be the milestone conducive to further research and

development in the domain of analyzing and applying evolutionary optimization algorithms to NP-hard combinatorial optimization problems.

References

[1] X. S. Yang, *Nature-Inspired Meta-Heuristic Algorithms*, Luniver Press, Beckington, UK, 2008.

[2] S. Lukasik and S. Zak, "Firefly algorithm for con-tinuous constrained optimization tasks," in *Proceedings of the International Conference on Computer and Computational Intelligence (ICCCI '09)*, N. T. Nguyen, R. Kowalczyk, and S.-M. Chen, Eds., vol. 5796 of *LNAI*, pp. 97–106, Springer, Wroclaw, Poland, October 2009.

[3] X. S. Yang, "Firefly algorithms for multimodal optimization," in *Proceedings of the Stochastic Algorithms: Foundations and Applications (SAGA '09)*, vol. 5792 of *Lecture Notes in Computing Sciences*, pp. 178–178, Springer, Sapporo, Japan, October 2009.

[4] X. S. Yang, "Firefly algorithm, Levy flights and global optimization," in *Research and Development in Intelligent Systems XXVI*, pp. 209–218, Springer, London, UK, 2010.

[5] K. S. Kumar, V. Tamilselvan, N. Murali, R. Rajaram, N. S. Sundaram, and T. Jayabarathi, "Economic load dispatch with emission constraints using various PSO algorithms," *WSEAS Transactions on Power Systems*, vol. 3, no. 9, pp. 598–607, 2008.

[6] M. Basu, "A simulated annealing-based goal-attainment method for economic emission load dispatch of fixed head hydrothermal power systems," *International Journal of Electrical Power and Energy Systems*, vol. 27, no. 2, pp. 147–153, 2005.

[7] On-line Matlab Tutorials, The MathWorksTM Support web-page, *Documentation of Optimization Toolbox*, Topics: Goal Attainment Method, Sequential Quadratic Pro-gramming & Multi-objective Optimization http://www.mathworks.com.au/help/toolbox/optim/.

[8] P. C. Chen and C. M. Huang, "Biobjective power dispatch using goal-attainment method and adaptive polynomial networks," *IEEE Transactions on Energy Conversion*, vol. 19, no. 4, pp. 741–747, 2004.

[9] T. Bouktir, R. Labdani, and L. Slimani, "Economic power dispatch of power system with pollution control using multi-objective particle swarm optimization," *Journal of Pure & Applied Sciences*, vol. 4, no. 2, pp. 57–77, 2007.

[10] W. F. Abd El-Wahed, A. A. Mousa, and M. A. Elsisy, "Solving economic emissions load dispatch problem by using hybrid ACO-MSM approach," *The Online Journal on Power and Energy Engineering*, vol. 1, no. 1, 2009.

[11] E. Zitzler, M. Laumanns, and S. Bleuler, "A Tuto-rial on Evolutionary Multi-objective Optimization," Swiss Federal Institute of Technology (ETH) Zurich, Computer Engineering and Networks Laboratory (TIK), Zurich, Switzerland.

[12] J. Cooper, *A MATLAB® Companion for Multivariable Calculus*, A Harcourt Science and Technology Company/Academic Press, 2001.

[13] L. Xie, S. Wang, and Z. Wu, "Study on economic, rapid and environmental power dispatch based on fuzzy multi-objective optimization," *Modern Applied Science*, vol. 3, no. 6, pp. 38–44, 2009.

[14] S. S. RAO, *Engineering Optimization Theory and Practice*, John Wiley & Sons, New York, NY, USA, 4th edition, 2009.

[15] J. S. Alsumait and J. K. Sykulski, "Solving economic dispatch problem using hybrid GA-PS-SQP method," in *Proceedings of the International Conference on Computer as a Tool (EUROCON '09)*, pp. 351–356, Saint Petersburg, Russia, May 2009.

[16] M. Sudhakaran, S. M. R. Slochanal, R. Sreeram, and N. Chandrasekhar, "Application of refined genetic algorithm to combined economic and emission dispatch," *Journal of the Institution of Engineers*, vol. 85, no. 2, pp. 115–119, 2004.

[17] R. Gonçalves, C. Almeida, J. Kuk, and M. Delgado, "Solving economic load dispatch problem by natural computing intelligent systems," in *Proceedings of the 15th International Conference on Intelligent System Applications to Power Systems (ISAP '09)*, Curitiba, Brazil, November 2009.

[18] Y. P. Verma and A. Kumar, "Economic load dispatch solutions using new particle swarm intelligence," in *Proceedings of the 5th National Power Systems Conference (NPSC '08)*, pp. 220–225, IIT, Bombay, India, December 2008.

[19] C. Palanichamy and N. S. Babu, "Analytical solution for combined economic and emissions dispatch," *Electric Power Systems Research*, vol. 78, no. 7, pp. 1129–1137, 2008.

[20] X. S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," *International Journal of Bio-Inspired Computation*, vol. 2, no. 2, pp. 78–84, 2010.

Advances in
Operations Research

Advances in
Decision Sciences

Mathematical Problems
in Engineering

Algebra

Journal of
Probability and Statistics

The Scientific
World Journal

International Journal of
Differential Equations

International Journal of
Combinatorics

Advances in
Mathematical Physics

Journal of
Complex Analysis

Journal of
Mathematics

International Journal of
Stochastic Analysis

Abstract and
Applied Analysis

Discrete Dynamics in
Nature and Society

International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces

Journal of
Applied Mathematics

Journal of
Optimization