British Journal of Mathematics & Computer Science
4(20): 2978-2999, 2014

SCIENCEDOMAIN *international*
www.sciencedomain.org

Genetic Algorithm and Rough Sets Based Hybrid Approach for Economic Environmental Dispatch of Power Systems

Mohamed A. Hussein[1*], Ahmed A. EL-Sawy[1,2], EL-Sayed M. Zaki[1] and A. A. Mousa[1,3]

[1]*Department of Basic Engineering Sciences, Faculty of Engineering, Menoufia University, Shibin El-Kom, Egypt.*
[2]*Department of Mathematics, Faculty of Science, Qassim University, Saudi Arabia.*
[3]*Department of Mathematics, Faculty of sciences, Taif University, Saudi Arabia.*

Original Research Article

Received: 20 May 2014
Accepted: 03 July 2014
Published: 07 August 2014

Abstract

In this paper we present a new optimization algorithm for Economic environmental dispatch EED of power systems. The purpose of EED problem is to compute the optimal generation for individual units of the power system by minimizing the fuel cost and emission levels simultaneously, subject to various equality and inequality constraints. The proposed algorithm is population based an evolutionary algorithm which operates in two phases: in the first one, genetic algorithm is implemented as search engine in order to generate approximate true Pareto front. This algorithm based on concept of co-evolution and repair algorithm for handling nonlinear constraints. Also it maintains a finite-sized archive of non-dominated solutions which gets iteratively updated in the presence of new solutions based on the concept of ε -dominance. Then, in the second phase, rough sets theory is adopted as local search engine in order to improve the spread of the solutions found so far. Optimization using multiobjective evolutionary algorithms yields not a single optimal solution. However, for practical applications, we need to select one solution which will satisfy the different goals to some extent. TOPSIS method has the ability to identify the best alternative from a finite set of alternatives. The proposed approach is carried out on the standard IEEE 30-bus 6-generator test system. The results demonstrate the capabilities of the proposed approach to generate true and well-distributed Pareto-optimal nondominated solutions of the multiobjective EED problem. Also the comparison with the exiting well-known algorithms demonstrates the superiority of the proposed approach and confirms its potential to solve the multiobjective EED problem.

Keywords: Economic environmental dispatch; multiobjective optimization; genetic algorithms; rough sets; TOPSIS.

**Corresponding author: moh_abdelsameea_h@yahoo.com;*

1 Introduction

The generation of electricity from fossil fuel releases several contaminantsinto the atmosphere. The problem that has attracted much attention is pollution minimization due to the pressing public demand for clean air. Operating at absolute minimum cost can no longer be the only criterion for dispatching electric power due to increasing concern the environmental consideration. The purpose of Economic Environmental Dispatch (EED) problem is to compute the optimal generation for individual units of the power systemby minimizing the fuel cost and emission levels simultaneously, subject to various equality and inequality constraints including the security measures of the power transmission/distribution.

To optimize economic environmental dispatch problem, different techniques have been reported in the literature. In past decades, the multiobjective EED problem was converted to a single objective problem by linear combination of different objectives as a weighted sum [1- 4]. The important aspect of this weighted sum method is that a set of Pareto-optimal solutions can be obtained by varying the weights factors. This requires multiple runs as many times as the number of desired Pareto-optimal solutions. Also, this method can be only used to find Pareto-optimal solutions in problems having a convex Pareto-optimal front. In addition, there is no rational basis of determining adequate weights and the objective function so formed may lose significance due to combining noncommensurable objectives.

In other research direction [5-7] the multiobjective EED problem was reduced to a single objective optimization problem by treating the emission as a constraint with a permissible limit. This formulation, however, has a severe difficulty in getting the trade-off relations between cost of generation and emission.

Goal programming method was also proposed for multiobjective EED problem [8]. In this method, a target or a goal to be achieved for each objective is assigned and the objective function will then try to minimize the distance from the targets to the objectives. The main drawback of this method is that it requires a priori knowledge about the shape of the problem search space.

Recently, the direction is to handle both objectives simultaneously as competing objectives instead of simplifying the multiobjective problem to a single objective problem [9-15].The use and development of heuristics-based multiobjective optimization techniques (Evolutionary Algorithms) have significantly grown. Since they use a population of solutions in their search, multiple Pareto-optimal solutions can, in principle, be found in one single run. Moreover the studies on evolutionary algorithms over the past few years have shown that these models can be efficiently used to eliminate most of the difficulties of classical methods [16-19]. A genetic Algorithms (GA) optimization method is one evolutionary algorithms technique which was successfully applied in real optimization problems.

In this paper we present a new multiobjective optimization algorithm to optimize Economic environmental dispatch EED of power systems. The proposed algorithm operates in two phases: in the first one, multiobjective version of genetic algorithm is used as search engine in order to generate approximate true Pareto front. Then in the second phase, rough set theory is adopted as local search engine in order to improve the spread of the solutions found so far. Optimization using multiobjective evolutionary algorithms yields not a single optimal solution. However for practical applications, we need to select one solution which will satisfy the different goals to some extent. TOPSIS method [20,21] has the ability to identify the best alternative from a finite set of alternatives. The proposed algorithm is implemented to the standard IEEE 30-bus 6-generator test system to investigate the effectiveness of the proposed approach and the results are compared to different well-known algorithms reported in literature.

2 Principle of Multiobjective Optimization

A multiobjective Optimization Problem (MOP) can be defined as determining a vector of design variables within a feasible region to minimize a vector of objective functions that usually conflict with each other. Such a problem takes the form:

$$\begin{aligned} Min \quad F(x) &= (f_1(x), f_2(x), \ldots, f_m(x))^T \\ s.t. \quad & x \in S \\ x &= (x_1, x_2, \ldots, x_n)^T \end{aligned} \tag{1}$$

Where $(f_1(x), f_2(x), \ldots, f_m(x))$ are the m objectives functions, $(x_1, x_2, \ldots, x_n)$ are the n decision variables, and $S \in R^n$ is the solution (parameter) space.

Definition 1.(Pareto optimal solution): x^* is said to be a Pareto optimal solution of MOP if there exists no other feasible x (i.e., $x \in S$) such that, $f_j(x) \le f_j(x^*)$ for all $j = 1, 2, \ldots, m$ and $f_j(x) < f_j(x^*)$ for at least one objective function f_j.

Definition 2.[22]. (ε-dominance) Let $f : x \to R^m$ and $a, b \in X$. Then a is said to ε-dominate b for some ε > 0, denoted as $a \succ_{\varepsilon} b$, if and only if for $i \in \{1, \ldots, m\}$

$$(1-\varepsilon) f_i(a) \le f_i(b)$$

Definition 3. (ε -approximate Pareto set) Let X be a set of decision alternatives and $\varepsilon > 0$. Then a set x_{ε} is called an ε-approximate Pareto set of X , if any vector $a \in x$ is ε-dominated by at least one vector $b \in x_{\varepsilon}$, i.e

$$\forall a \in x : \exists b \in x_{\varepsilon} \ \text{such that} \ b \succ_{\varepsilon} a$$

According to definition 2, the ε value stands for a relative allowed for the objective values as declared in Fig. 1. This is especially important in higher dimensional objective spaces, where the concept of ε -dominance can reduce the required number of solutions considerably. Also, makes the algorithms practical by allowing a decision maker to control the resolution of the Pareto set approximation by choosing an appropriate ε-value

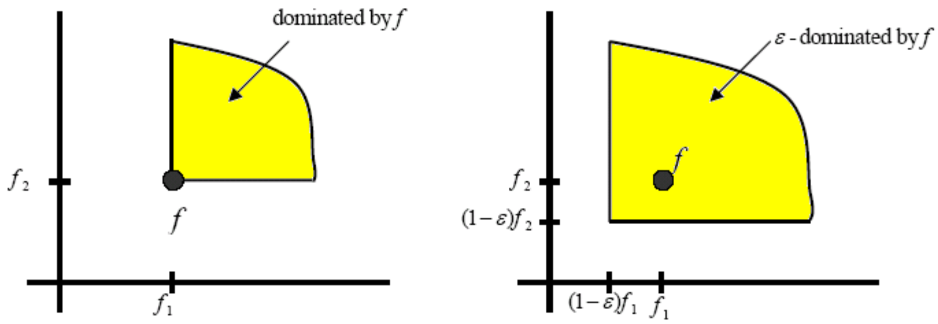

Fig. 1. Graphs visualizing the concepts of dominance (left) and ε-dominance (right)

3 Economic Environmental Dispatch (EED) of Power Systems

The Economic Environmental Dispatch EED problem is formulated as a nonlinear constrained multiobjective optimization problem which attempts to minimize fuel cost and emission simultaneously which are conflicting with each other, while satisfying various equality and inequality constraints. The problem is formulated as described below.

3.1 Problem Objectives

3.1.1 Minimization of fuel cost objective

The classical economic dispatch problem of finding the optimal combination of power generation, which minimizes the total fuel cost while satisfying the total required demand can be mathematically stated as follows [23]:

$$f_1(P_{Gi}) = C_t = \sum_{i=1}^{n} C_i(P_{Gi}) = \sum_{i=1}^{n} (a_i + b_i P_{Gi} + c_i P_{Gi}^2)\$ / hr \tag{2}$$

Where:

C_t: total fuel cost (\$/hr), C_i: is fuel cost of generator i

a_i, b_i, c_i: fuel cost coefficients of generator i, P_{G_i}: power generated (p.u)by generator i,

n: number of generators.

3.1.2 Minimization of emission objective

The emission function can be presented as the sum of all types of emission considered, such as NO_x, SO_2, thermal emission, etc., with suitable pricing or weighting on each pollutant emitted.

In the present study, only one type of emission NO_x is taken into account without loss of generality. The amount of NO_x emission is given as a function of generator output, that is, the sum of a quadratic and exponential function:

$$f_2(P_{Gi}) = E_{NO_x} = \sum_{i=1}^{n}[10^{-2}(\alpha_i + \beta_i P_{Gi} + \gamma_i P_{Gi}^2) + \xi_i \exp(\lambda_i P_{Gi})] \; ton/hr \quad (3)$$

Where, $\alpha_i, \beta_i, \gamma_i, \xi_i, \lambda_i$: coefficients of the *ith* generator's NO_x emission characteristic.

3.2 Problem Constraints

3.2.1 Power balance constraint

The total power generated must supply the total load demand and the transmission losses:

$$\sum_{i=1}^{n} P_{Gi} - P_D - P_{Loss} = 0 \quad (4)$$

Where:

P_D : total load demand (*p.u.*), and P_{loss} : transmission losses (*p.u.*).

The transmission loss is given by[24]:

$$P_{Loss} = \sum_{i=1}^{n}\sum_{i=1}^{n}[A_{ij}(P_i P_j + Q_i Q_j) + B_{ij}(Q_i P_j - P_i Q_j)] \quad (5)$$

Where:

$$P_i = P_{Gi} - P_{Di}, \quad \mathrm{Q}_i = Q_{Gi} - Q_{Di} \quad , \mathrm{A_{ij}} = \frac{R_{ij}}{V_i V_j}\cos(\delta_i - \delta_j), \quad \mathrm{B_{ij}} = \frac{R_{ij}}{V_i V_j}\sin(\delta_i - \delta_j)$$

n : number of buses

R_{ij} : series resistance connecting buses *i* and *j*

V_i : voltage magnitude at bus *i*

δ_i : voltage angle at bus *i*

P_i : real power injection at bus *i*

Q_i : reactive power injection at bus *i*

3.2.2 Power generation constraint

The power generated P_{Gi} by each generator is constrained between its minimum and maximum limits

$$P_{Gi\min} \le P_{Gi} \le P_{Gi\max}, \quad Q_{Gi\min} \le Q_{Gi} \le Q_{Gi\max}, \quad V_{i\min} \le V_i \le V_{i\max}, \qquad i = 1, \ldots\ldots, n$$

Where:

$P_{Gi\min}$: minimum power generated, and

$P_{Gi\max}$: maximum power generated.

3.2.3 Security constraint

A mathematical formulation of the security constrained EED problem would require a very large number of constraints to be considered. However, for typical systems the large proportion of lines has a rather small possibility of becoming overloaded. The EED problem should consider only the small proportion of lines in violation, or near violation of their respective security limits which are identified as the critical lines. We consider only the critical lines that are binding in the optimal solution. The detection of the critical lines is assumed done by the experiences of the DM. An improvement in the security can be obtained by minimizing the following objective function.

$$S = f(P_{Gi}) = \sum_{j=1}^{k} (|\, T_j(P_G) \,| / T_j^{\max}) \tag{6}$$

Where, $T_j(P_G)$ is the real power flow $T_j^{\max}$ is the maximum limit of the real power flow of the j th line and k is the number of monitored lines. The line flow of the j th line is expressed in terms of the control variables P_{Gs}, by utilizing the generalized generation distribution factors (GGDF) [24] and is given below.

$$T_J(P_G) = \sum_{i=1}^{n} (D_{ji} P_{Gi}) \tag{7}$$

where, D_{ji} is the generalized GGDF for line j, due to generator i

For secure operation, the transmission line loading S_l is restricted by its upper limit as

$S_\ell \le S_{\ell\max}, \ell = 1, \ldots, n_\ell$

Where n_ℓ is the number of transmission line.

4 Problem Formulation of Economic Environmental Dispatch (EED)

The multiobjective economic emission load dispatch optimization problem is therefore formulated as:

$$
\begin{aligned}
&Min \quad f_1(P_{Gi}) = C_t = \sum_{i=1}^{n} (a_i + b_i P_{Gi} + c_i P_{Gi}^2) \ \$ / hr \\
&Min \quad f_2(P_{Gi}) = E_{NO_x} = \sum_{i=1}^{n} [10^{-2}(\alpha_i + \beta_i P_{Gi} + \gamma_i P_{Gi}^2) + \xi_i \exp(\lambda_i P_{Gi})] \ ton / hr \\
&s.t. \quad \sum_{i=1}^{n} P_{Gi} - P_D - P_{Loss} = 0, \\
&\qquad S_\ell \le S_{\ell\max}, \qquad \ell = 1, ..., n_{Line}, \\
&\qquad P_{Gi\min} \le P_{Gi} \le P_{Gi\max} \qquad i = 1, ..., n \\
&\qquad Q_{Gi\min} \le Q_{Gi} \le Q_{Gi\max} \qquad i = 1, ..., n \\
&\qquad V_{i\min} \le V_i \le V_{i\max} \qquad i = 1, ..., n
\end{aligned}
\tag{8}
$$

5 The Proposed Approach

In this section we present a new multiobjective optimization algorithm, the proposed algorithm operates in two phases: in the first one, multiobjective version of genetic algorithm is used as search engine in order to generate approximate true Pareto front. Then in the second phase, rough set theory is adopted as local search engine in order to improve the spread of the solutions found so far**.** All equality constraints are replaced by inequality constraints with additional parameter tol to define the precision of the system.

$$
\begin{aligned}
&f_i(x_1, x_2, ..., x_n) = 0 \Rightarrow -tol \le (f_i(x_1, x_2, ..., x_n) \le tol \\
&\Rightarrow \left| f_i(x_1, x_2, ..., x_n) \right| \le tol \\
&\Rightarrow \left| f_i(x_1, x_2, ..., x_n) \right| - tol \le 0
\end{aligned}
$$

5.1 Use of Rough Sets in Multiobjective Optimization

For our proposed approach we will try to investigate the Pareto front using a Rough sets grid. To do this, we will use an initial approximate of the Pareto front (provided by any evolutionary algorithm) and will implement a grid in order to get more information about the front that will let to improve this initial approximation [25].

To create this grid, as an input we will have N feasible points divided in two sets: the nondominated points (NS) and the dominated ones (DS). Using these two sets we want to create a grid to describe the set NS in order to intensify the search on it. This is, we want to describe the Pareto front in the decision variable space because then we could easily use this information to generate more efficient points and then improve this initial Pareto approximation. Fig. 2 shows how information in objective function space can be translated into information in decision variable space through the use of a grid. We must note the importance of the DS sets as in a rough sets method, where the information comes from the description of the boundary of the two sets NS, DS. Then the more efficient points provided the better. However, it is also required to provide dominated points, since we need to estimate the boundary between being dominated and being nondominated. Once the information is computed we can simply generate more points in the "efficient side".

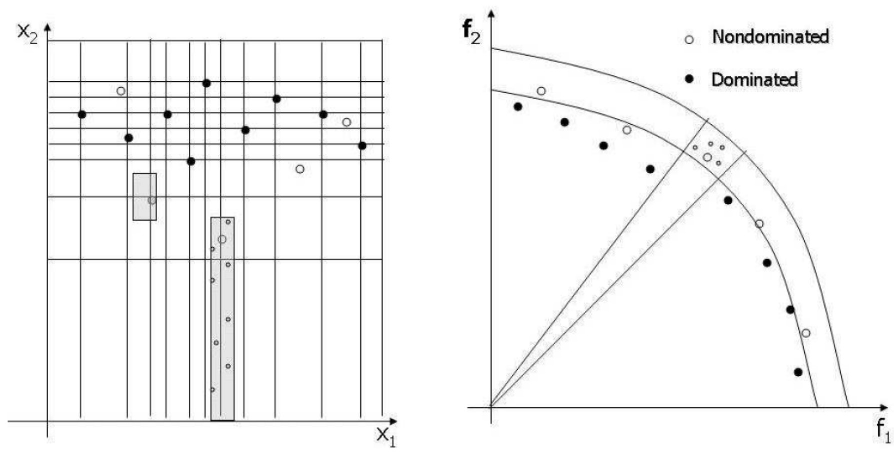

Fig. 2. Decision variable space (left) and objective function space (right) [25]

5.2 Structure of an Iterative Multiobjective Search Algorithm

The purpose of this section is to informally describe the problem we are dealing with. To this end, let us first give a template for a large class of iterative search procedures which are characterized by the generation of a sequence of search points and a finite memory.

Algorithm 1. Iterative search algorithm

1. $t = 0$
2. $A^{(0)} = 0$
3. $while$ terminate $(A^{(t)}, t) = false$ do
4. $t = t + 1$
5. $f^{(t)} = generate(\cdot)$ {generate new search point}
6. $A^{(t)} = update(A^{(t-1)}, f^{(t)})$ {update archive}
7. $end\ \ while$
8. $Output: A^{(t)}$

An abstract description of a generic iterative search algorithm is given in Algorithm 1[22]. The integer t denotes the iteration count, the n-dimensional vector $f^{(t)}$ is the sample generated at iteration t and the set $\boldsymbol{A}^{(t)}$ will be called the archive at iteration t and should contain a representative subset of the samples in the objective space F=[$f_1(x)$, $f_2(x)$,..., $f_m(x)$] generated so far. To simplify the notation, we represent samples by n-dimensional real vectors f where each coordinate represents one of the objective values as shown in Fig. 3.

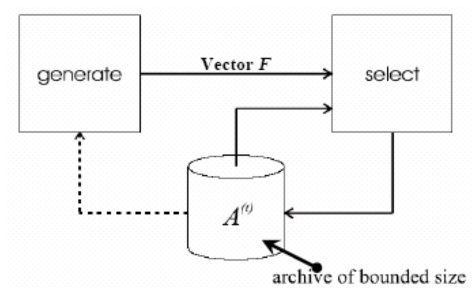

Fig. 3. Block diagram of Archive/selection strategy

The purpose of the function $f^{(t)} = generate(\cdot)$ is to generate a new solutions in each iteration t, possibly using the contents of the old archive set $A^{(t-1)}$. The function $A^{(t)} = update(A^{(t-1)}, f^{(t)})$ gets the new solutions $generate(\cdot)$ and the old archive set $A^{(t-1)}$ and determines the updated one, namely $A^{(t)}$. In general, the purpose of this sample storage is to gather 'useful' information about the underlying search problem during the run. Its use is usually two-fold: On the one hand it is used to store the 'best' solutions found so far, on the other hand the search operator exploits this information to steer the search to promising regions. This procedure could easily be viewed as an evolutionary algorithm when the generate operator is associated with variation (recombination and mutation). However, we would like to point out that all following investigations are equally valid for any kind of iterative process which can be described as Algorithm 1and used for approximating the Pareto set of multiobjective optimization problems.

5.3 Constraint Multiobjective Optimization via Genetic Algorithm

In any interesting multiobjective optimization problem, there exist a number of such solutions which are of interest to designers and practitioners. Since no one solution is better than any other solution in the Pareto-optimal set, it is also a goal in a multiobjective optimization to find as many such Pareto-optimal solutions as possible. Unlike most classical search and optimization problems, GAs works with a population of solutions and thus are likely (and unique) candidates for finding multiple Pareto-optimal solutions simultaneously. There are two tasks that are achieved in a multiobjective GA. (1) Convergence to the Pareto-optimal set, and (2) Maintenance of diversity among solutions of the Pareto-optimal set. Here we present a new optimization system, which is based on concept of co-evolution and repair algorithms. Also it is based on the ε-dominance concept. The use of ε-dominance also makes the algorithms practical by allowing a decision maker to control the resolution of the Pareto set approximation by choosing an appropriate ε value.

5.3.1 Initialization stage

The algorithm uses two separate population, the first population $\boldsymbol{P}^{(t)}$ consists of the individuals which initialized randomly satisfying the search space (The lower and upper bounds), while the second population $\boldsymbol{R}^{(t)}$ consists of reference points which satisfying all constraints (feasible points), However, in order to ensure convergence to the true Pareto-optimal solutions, we concentrated on how elitism could be introduced. So, we propose an "archiving/selection" strategy that guarantees at the same time progress towards the Pareto-optimal set and a covering of the whole range of the non-dominated solutions. The algorithm maintains an externally finite-sized archive $\boldsymbol{A}^{(t)}$ of non-dominated solutions which gets iteratively updated in the presence of new solutions based on the concept of ε-dominance

5.3.2 Repair algorithm

The idea of this technique is to separate any feasible individuals in a population from those that are infeasible by repairing infeasible individuals. This approach co-evolves the population of infeasible individuals until they become feasible. Repair process works as follows. Assume, there is a search point $\omega \notin \boldsymbol{S}$ (where $\boldsymbol{S}$ is the feasible space). In such a case the algorithm selects one of the reference points (Better reference point has better chances to be selected), say $\boldsymbol{r} \in \boldsymbol{S}$ and creates random points $\bar{\boldsymbol{Z}}$ from the segment defined between $\omega, \boldsymbol{r}$, but the segment may be extended equally on both sides determined by a user specified parameter $\mu \in [0,1]$. Thus, a new feasible individual is expressed as:

$$\left.\begin{aligned} z_1 &= \gamma.\omega + (1-\gamma).r \\ z_2 &= (1-\gamma).\omega + \gamma.r \end{aligned}\right\}, \gamma = (1+2\mu)\delta - \mu, \delta \in [0,1] \quad (9)$$

5.3.3 Evolutionary algorithm: phase 1

In the first phase, the proposed algorithm uses two separate population, the first population $\boldsymbol{P}^{(t=0)}$ (where t is the iteration counter) consists of the individuals which initialized randomly satisfying the search space (The lower and upper bounds), while the second population $\boldsymbol{R}^{(0)}$ consists of reference points which satisfying all constraints (feasible points). Also, it stores initially the Pareto-optimal solutions externally in a finite sized archive of non-dominated solutions $\boldsymbol{A}^{(0)}$. We use cluster algorithm to create the next population $\boldsymbol{P}^{t+1}$, if $|\boldsymbol{P}^{(t)}|>|\boldsymbol{A}^{(t)}|$ then new population $\boldsymbol{P}^{t+1}$ consists of all individual from $A^{(t)}$ and the population $\boldsymbol{P}^{(t)}$ are considered for the clustering procedure to complete $P^{(t+1)}$, if $|\boldsymbol{P}^{(t)}|<|\boldsymbol{A}^{(t)}|$ then $|P|$ solutions are picked up at random from $A^{(t)}$ and directly copied to the new population $P^{(t+1)}$.

Since our goal is to find new nondominated solutions, one simple way to combine multiple objective functions into a scalar fitness function [26,27] is the following weighted sum approach

$$f(x) = w_1 f_1(x) + \ldots + w_i f_i(x) + \ldots + w_m f_m(x) = \sum_{j=1}^{m} w_j f_j(x) \quad (10)$$

Where x is a string (i.e., individual), $f(x)$ is a combined fitness function, $f_i(x)$ is the ith objective function. When a pair of strings are selected for a crossover operation, we assign a random number to each weight as follows:

$$w_i = \frac{random_i(.)}{\sum_{j=1}^{m} random_j(.)}, \quad i = 1, 2, .., m \tag{11}$$

Calculate the fitness value of each string using the random weights w_i. Select a pair of strings from the current population according to the following selection probability $\beta(x)$ of a string x in the population $P^{(t)}$

$$\beta(x) = \frac{f(x) - f_{\min}(P^{(t)})}{\sum_{x \in P^{(t)}} \{f(x) - f_{\min}(P^{(t)})\}}, \text{ where } f_{\min}(P^{(t)}) = \min\{f(x) \mid x \in P^{(t)}\} \tag{12}$$

This step is repeated for selecting $|P|/2$ Paris of strings from the current populations. For each selected pair apply crossover operation to generate two new strings, for each strings generated by crossover operation, apply a mutation operator with a prespecified mutation probability. The system also includes the survival of some of good individuals without crossover or mutation. The algorithm maintains a finite-sized archive $\boldsymbol{A^t}$ of non-dominated solutions which gets iteratively updated in the presence of a new solutions based on the concept of ε-dominance, such that new solutions are only accepted in the archive if they are not ε-dominated by any other element in the current archive .The use of ε-dominance also makes the algorithms practical by allowing a decision maker to control the resolution of the Pareto set approximation by choosing an appropriate ε value.

5.3.4 Local search mechanism inspired on rough sets theory: phase 2

Upon termination of phase 1, we start phase2, with initial approximate of the Pareto front (provided by the proposed algorithm in phase1) which noted as NS. Also all dominated solutions are marked as DS. It is worth remarking that NS can simply be a list of solutions.

From the set NS we choose NNS points previously unselected. If we do not have enough unselected points, we choose the rest randomly from the set DS. Next, we choose from the set NS, NDS points previously unselected (and in the same way if we do not have enough unselected points, we complete them in a random fashion) these points will be used to approximate the boundary between the Pareto front and the rest of the feasible set in decision variable space. We store theses points in the set Items and perform rough sets iterations:

1- Range Initialization: for each decision variable i, we compute and sort (from smallest and highest) the different values it takes in the set Items. Then, for each decision variable, we have a set of *rang* values and combining all these sets we have a non-uniform grid in decision variable space.
2- Compute Atoms: we compute "NNS rectangular atoms" centered in the NNS efficient points selected. To build a rectangular atom associated to a nondominated point $x^e \in Items$ we compute the following upper and lower bounds for each decision variable i:

- Lower Bound i : Middle point between x_i^e and the previous value in the set $rang_i$
- Upper Bound i : Middle point between x_i^e and the following value in the set $rang_i$
- If there are no pervious or subsequent values in $rang_i$, we consider the absolute lower or upper bound of variable i . This setting lets the method to explore close to the feasible set boundaries.

3- **Generate Offspring:** inside each atom we randomly generate offspring new points. Each of these points is sent to the set NS to check if it is must be included as a new nondominated point. If any point in NS is dominated by this new point, it is sent to the set DS.

Algorithm 2 shows the operator for ε-approximate Pareto set, the idea is that "new solutions are only accepted in the archive if they are not ε-dominated by any other element of the current archive". If a solution is accepted, all dominated solutions are removed. The pseudo code of the proposed algorithm is declared in **Algorithm 3.**

Algorithm 2: Operator for ε-approximate Pareto set

1. $INPUT : A, x$
2. $if \, \exists x' \in A \text{ such that } x' \succ_{\varepsilon} x \text{ then}$
3. $A' = A$
4. $else$
5. $D = \{x' \in A : x \succ x'\}$
6. $A' = A \cup \{x\} \setminus D$
7. $end\ if$
8. $Output : A'$

Algorithm 3: The proposed algorithm(pseudo code of the proposed algorithm)

t=0
2. Create $P^{(0)}, R^{(0)}$
3. $A^{(0)} = nondominated(P^{(0)})$
3. *while* terminate $(A^{(t)}, t) =$ *false* do
4. $t = t + 1$ $\quad$ *Phase*1
5. $P^{(t)} = generate(A^{(t-1)}, P^{(t-1)})$ $\quad$ {generate new search point}
6. $A^{(t)} = update(A^{(t-1)}, P^{(t)})$ $\quad$ {update archive (algorithm 3)}
7. *end while*
8. *Output* : $A^{(t)}$

8. $A^{(t)} \rightarrow NS$, Dominted Pointes $\rightarrow$ DS
9. Compute "Atom"
10. Comput Upper Bound$_i$, Lower Bound$_i$ $\quad$ *Phase*2
11. *Generate* new offspring.
12. Update NS set

13. *Output* NS

6 Implementation of the Proposed Approach

In order to validate the proposed approach and quantitatively compare its performance with other MOEAs, we present in this section comparison study applied to the standard IEEE 30-bus 6-generator test system with two objective. The single-line diagram of this system is shown in Fig. 4 and the detailed data are given in [7,28]. The values of fuel cost and emission coefficients are given in Table 1. For comparison purposes with the reported results, the system is considered as losses and the security constraint is released. The techniques used in this study were developed and implemented using MATLAB environment. Table 2 gives Reactive power limit and Voltage limits and Table 3 lists the parameter setting used in the algorithm for all runs.

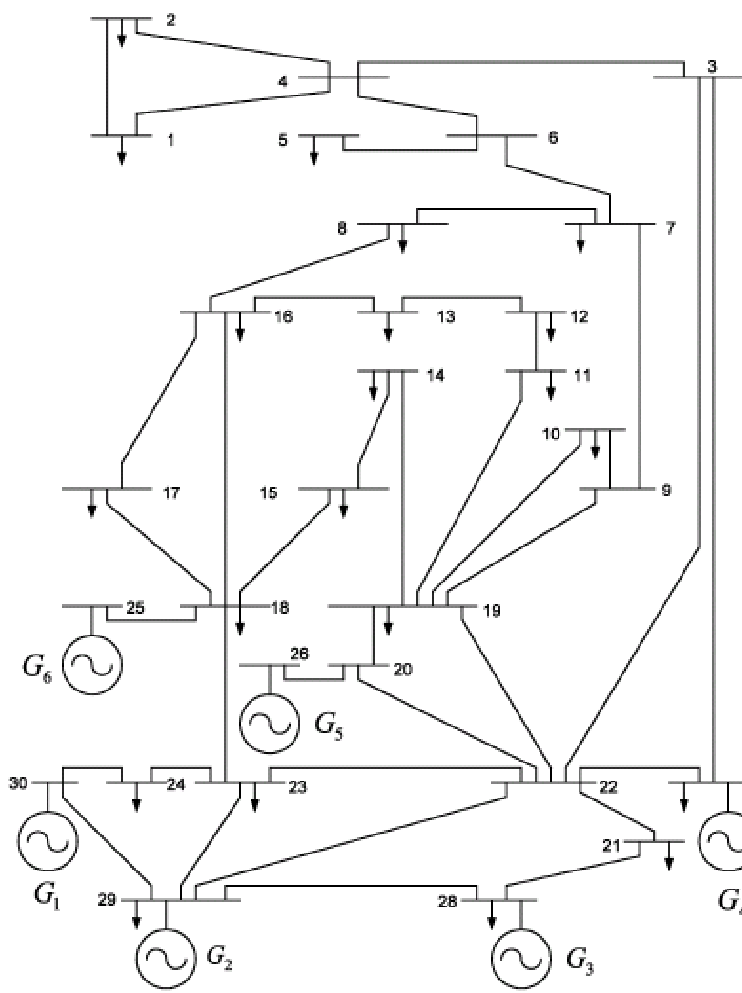

Fig. 4. Single line diagram of IEEE 30-bus 6-generator test system

Table 1. Generator fuel cost and emission coefficients

Cost		**G1**	**G2**	**G3**	**G4**	**G5**	**G6**
	a	10	10	20	10	20	10
	b	200	150	180	100	180	150
	c	100	120	40	60	40	100
Emission	α	4.091	2.543	4.258	5.426	4.258	6.131
	β	-5.554	-6.047	-5.094	-3.550	-5.094	-5.555
	γ	6.490	4.638	4.586	3.380	4.586	5.151
	ζ	2.0E-4	5.0E-4	1.0E-6	2.0E-3	1.0E-6	1.0E-5
	λ	2.857	3.333	8.000	2.000	8.000	6.667

Table 2. Reactive power limit and voltage limits

Bus	**Voltage**		**Reactive power**	
	$V_{i\,\min}$	$V_{i\,\max}$	$Q_{Gi\,\min}$	$Q_{Gi\,\max}$
1	0.9	1.05	-0.2	2.0
2	0.9	1.05	-0.2	2.0
3	0.9	1.05	-0.2	2.0
4	0.9	1.05	-0.2	2.0
5	0.9	1.05	-0.2	2.0
6	0.9	1.05	-0.2	2.0

In Table,Qmin, Qmax, Vmin and Vmax are in (p.u)

Table 3. The parameter setting of proposed algorithm

Parameter	Setting
Population size	200
No. of Generation	200
Crossover probability P_c	0.9
Mutation probability P_m	0.02
Selection operator	Roulette Wheel
Crossover operator	Single point
Mutation operator	Polynomial mutation
Relative tolerance ε	10e-6

7 Results and Discussion

Fig. 5 shows well-distributed Pareto optimal solutions obtained by the proposed algorithm after 200 generations. It is clear from the figure that Pareto-optimal set is well distributed and has satisfactory diversity characteristics compared with Pareto-optimal front of the NPGA [9] and Pareto-optimal front of the SPEA [10].

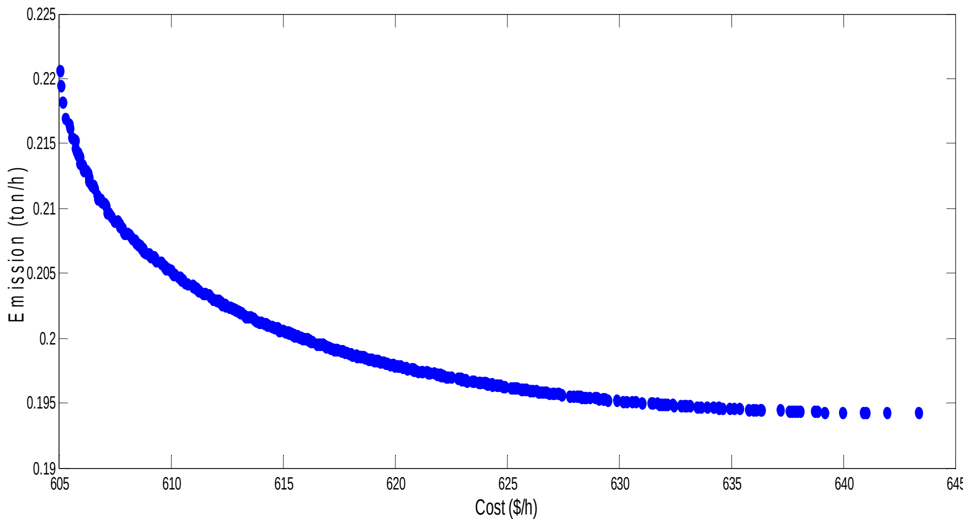

Fig. 5. Pareto-optimal front of the proposed approach

Tables 4 and 5 show the best fuel cost and best NO_X emission obtained by proposed algorithm as compared to Nondominated Sorting Genetic Algorithm (NSGA) [8], Niched Pareto Genetic Algorithm (NPGA) [9], Strength Pareto Evolutionary Algorithm (SPEA) [10] and IT-CEMOP [29]. It can be deduced that the proposed algorithm finds comparable minimum fuel cost and comparable minimum NO_X emission to the four evolutionary algorithms.

Table 4. Best total $/h fuel cost

	NSGA	NPGA	SPEA	IT-CEMOP	Proposed approach
P_{G1}	0.1168	0..1245	0.1086	0.1739	0.928
P_{G2}	0.3165	0.2792	0.3056	0.3578	0.2868
P_{G3}	0.5441	0.6284	0.5818	0.5311	0.5159
P_{G4}	0.9447	1.0264	0.9846	0.9790	0.9896
P_{G5}	0.5498	0.4693	0.5288	0.4429	0.5803
P_{G6}	0.3964	0.39993	0.3584	0.3725	0.3448
Best cost.	608.245	608.147	607.807	606.4533	605.0820
Corresponding Emission	0.21664	0.22364	0.22015	0.2028	0.2206

Convergence of fuel cost and emission objective functions are shown in Figs. 6 and 7. In the view of our results, the algorithm converges to the optimal solution as in Figs. 6 and 7, where the Cost of generating is converges to the optimal value, Also the Emissions of undesired materials is converges to the optimal value.

Table 5. Best total ton/h NO_X emission

	NSGA	NPGA	SPEA	IT-CEMOP	Proposed approach
P_{G1}	0.4113	0.3923	0.4043	0.3885	0.4046
P_{G2}	0.4591	0.4700	0.4525	0.4984	0.4611
P_{G3}	0.5117	0.5565	0.5525	0.5167	0.5289
P_{G4}	0.3724	0.3695	0.4079	0.4502	0.3874
P_{G5}	0.5810	0.5599	0.5468	0.5205	0.5338
P_{G6}	0.5304	0.5163	0.5005	0.5005	0.5014
Best Emission.	0.19432	0.19424	0.19422	0.1882	0.1942
Corresponding cost	647.251	645.984	642.603	642.8976	643.3729

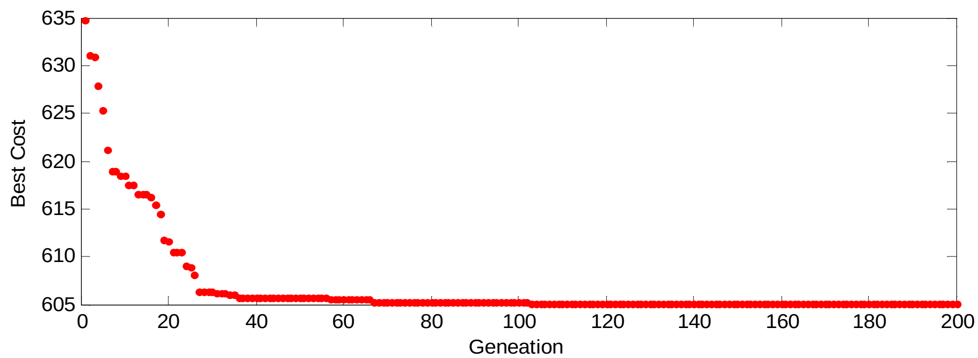

Fig. 6. Convergence of fuel cost objective

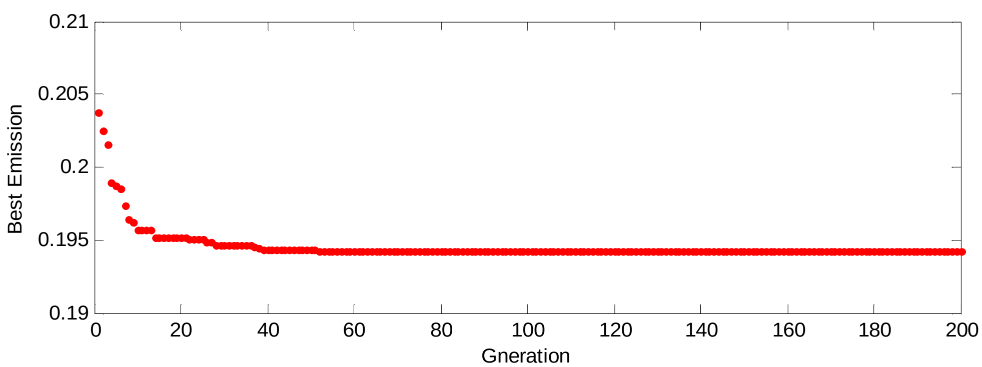

Fig. 7. Convergence emission objective

8 Identifying a Satisfactory Solution

Optimization of the above-formulated objective functions using multiobjective genetic algorithms yields not a single optimal solution, but a set of Pareto optimal solutions, in which one objective cannot be improved without sacrificing other objectives. For practical applications, however, we need to select one solution, which will satisfy the different goals to some extent. Such a solution is called best compromise solution. TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) method [20,21,30] has the ability to identify the best alternative from a finite set of alternatives quickly.

8.1 The Idea of TOPSIS can be expressed in a Series of Steps:

(1) Obtain performance data for n alternatives over Mcriteria x_{ij} (i=1,….,n ,j=1,….,M).

(2) Calculate normalized rating (vector normalization is used) r_{ij}.

(3) Develop a set of importance weights W_j, for each of the criteria. The basis for these weights can be anything, but, usually, is ad hoc reflective of relative importance.

$$V_{ij} = w_j . r_{ij} \tag{13}$$

(4) Identify the ideal alternative (extreme performance on each criterion) S^+.

$$S^+ = \{v_1^+, v_2^+, .., v_j^+, .., v_m^+\} = \left\{ \left(\max \ \mathrm{v_{ij}} \mid j \in J_1 \right), \left(\min \ \mathrm{v_{ij}} \mid j \in J_2 \right), i = 1, ..., n \right\} \tag{14}$$

Where J_1 is a set of benefit attributes and J_2 is a set of cost attributes.

(5) Identify the nadir alternative (reverse extreme performance on each criterion) S^-.

$$S^- = \{v_1^-, v_2^-, .., v_j^-, ..v_m^-\} = \left\{ \left(\min \ \mathrm{v_{ij}} \mid j \in J_1 \right), \left(\max \ \mathrm{v_{ij}} \mid j \in J_2 \right), i = 1, ..., n \right\} \tag{15}$$

(6) Develop a distance measure over each criterion to both ideal (D^+) and nadir (D^-).

$$D_i^+ = \sqrt{\sum_j (v_{ij} - v_j^+)^2}, \qquad D_i^- = \sqrt{\sum_j (v_{ij} - v_j^-)^2} \tag{16}$$

(7) For each alternative, determine a ratio R equal to the distance to the nadir divided by the sum of the distance to the nadir and the distance to the ideal,

$$R = \frac{D^-}{D^- + D^+} \tag{17}$$

(8) Rank alternative according to ratio R (in Step 7) in descending order.
(9) Recommend the alternative with the maximum ratio

Therefore it can be said that TOPSIS method is attractive since limited subjective input (namely the weight values which reflect the degree of satisfactory of each objective) is needed from the DM to get a satisfactory results from the Pareto set quickly. Also, this method can be classified as interactive approach, where the DM specifies input values according his needs.

Here, we need to select one solution (one operating point), which will satisfy the different goals to some extent. Such a solution is called best compromise solution. The identification of a best compromise solution requires taking into account the preferences expressed by the decision-maker DM, which reflect the degree of satisfactory of each objective. We incorporate relative weights of criterion importance as $\{w_1 = 0.2, \mathrm{w}_2 = 0.8\}$, which give relative importance for fuel cost as 0.2 and relative importance for NO_x emission objective as 0.8, the bigger the weighting factor, the more important is the attainment of that objective.

In order to obtain the normalized rating, fuel cost $f_1(\cdot)$, and emission $f_2(\cdot)$ are optimized individually to obtain minimum values of the objectives. The minimum and maximum values of the objectives are given in Table 6 (Minimum values of the objectives are obtained by giving full consideration to one of the objectives and neglecting the others).

Table 6. The minimum and maximum values of the objectives

Objective	**Max**	**Min**
Fuel cost ($)	643.3729	605.0802
Emission (ton)	0.2206	0.1942

For each alternative, determine a ratio *R* equal to the distance to the nadir divided by the sum of the distance to the nadir and the distance to the ideal as in step 7. Alternatives have been ranked by maximizing the ratio R. It is obvious that all set of solutions are ranked corresponding to the relative weights of criterion importance (degree of satisfactory)

To declare the performance of changing the weights $\{w_1, w_2 \mid w_1 + w_2 = 1\}$ on the best compromise solution, we plot different values of weight w_1 versus best compromise solution of $f_1(\cdot)$ (Cost ($/h) and versus best compromise solution of $f_2(\cdot)$ (Emission (ton/h) as in Fig. 8. It is obvious that for each weight (criterion importance), different best compromise solutions had found proportional to the criterion importance (weighting factor).

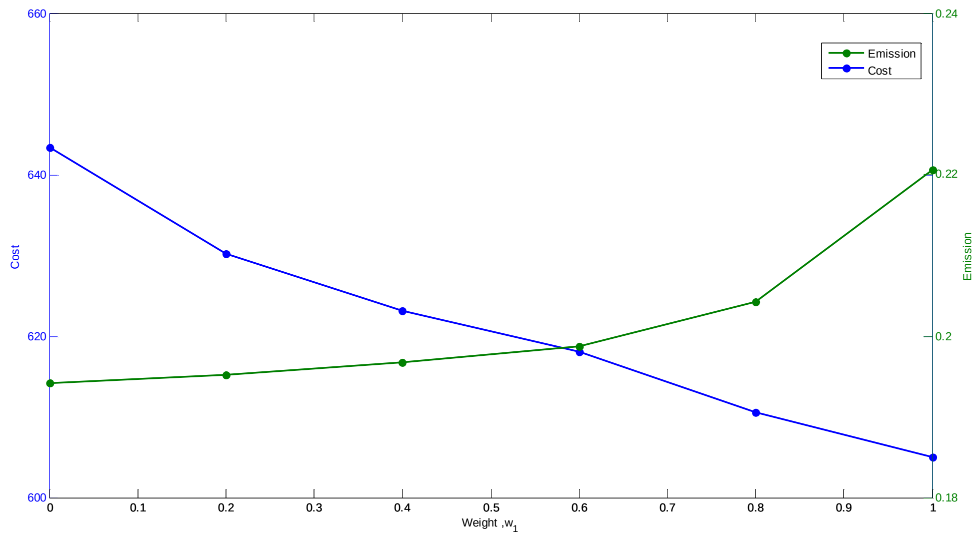

Fig. 8. Weight w_1 versus best compromise cost and emission

9 Conclusion

The proposed approach presented in this paper was applied to Economic Emission Load Dispatch (EELD) problem which formulated as multiobjective optimization problem with competing fuel cost, and emission. The proposed algorithm operates in two Phases: in the first one, multiobjective version of genetic algorithm is used as search engine in order to generate approximate true Pareto front. This algorithm based on concept of co-evolution and repair algorithm. Also it maintains a finite-sized archive of nondominated solutions which gets iteratively updated in the presence of new solutions based on the concept of ε-dominance. Then in the second phase, rough set theory is adopted as local search engine in order to improve the spread of the solutions found so far. Our proposed approach keeps track of all the feasible solutions found during the optimization. The following are the significant contributions of this paper:

1. The results prove superiority of the proposed approach to those reported in the literature.
2. The non-dominated solutions in the obtained Pareto-optimal set are well distributed and have satisfactory diversity characteristics.
3. The proposed approach is efficient for solving nonconvex multiobjective optimization where multiple Pareto-optimal solutions can be found in one simulation run.
4. The proposed technique has been effectively applied to solve the EED considering two objectives simultaneously, with no limitation in handing more than two objectives.
5. TOPSIS method is employed to extract the best compromise solution from the trade-off curve according to the determined weight factor, the bigger the weight factor, the more important is the attainment of that objective.

Competing Interests

Authors have declared that no competing interests exist.

References

[1] Chang CS, Wong KP, Fan B. Security-constrained multiobjective generation dispatch using bicriterion global optimization. IEE Proc.-Gener. Transm. Distrib. 1995;142(4):406-414.

[2] Dhillon JS, Parti SC, Kothari DP. Stochastic economic emission load dispatch. Electric Power System Research. 1993;26:179-186.

[3] Xu JX, Chang CS, Wang XW. Constrained multiobjective global optimization of longitudinal interconnected power system by genetic algorithm. IEE Proc.-Gener. Transm. Distrib. 1996;143(5):435-446.

[4] Zahav JI, Eisenberg L. Economic-environmental power dispatch. IEEE Trans. Syst., Man, Cybern. SMC. 1985;5(5):485- 489.

[5] Brodesky SF, Hahn RW. Assessing the influence of power pools on emission constrained economic dispatch. IEEE Trans. Power System. 1986;1(1):57-62.

[6] Granelli GP, Montagna M, Pasin GL, Marannino P. Emission constrained dynamic dispatch. Electric Power Syst. Res. 1992;24:56-64.

[7] Farag A, Al-Baiyat S, Cheng TC. Economic load dispatch multiobjective optimization procedures using linear programming techniques. IEEE Trans, Power System. 1995;10(2):731-738.

[8] Kermanshahi BS, Wu Y, Yasuda K, Yokoyama R. Environmental marginal cost evaluation by non-inferiority surface. IEEE Transactions on Power Systems. 1990;5(4):1151-1159.

[9] Abido MA. A novel multiobjective evolutionary algorithm for environmental/economic power dispatch. Electric Power Systems Research. 2003;65(1):71-81.

[10] Abido MA. A Niched Pareto genetic algorithm for multiobjective Environmental/Economic Dispatch. Electric Power Systems Research. 2003;25(2):97-105.

[11] Abido MA. Environmental/Economic power dispatch using multiobjective evolutionary algorithms. IEEE Transactions on Power Systems. 2003;18(4):1529-1537.

[12] Mousa AA, El-Shorbagy MA. Enhanced particle swarm optimization based local search for reactive power compensation problem. Applied Mathematics. 2012;3:1276-1284.

[13] Abd Allah A. Galal, Abd Allah A. Mousa, Bekheet N. Al-Matrafi. Hybrid Ant optimization system for multiobjective Optimal Power Flow Problem Under Fuzziness. Journal of Natural Sciences and Mathematics. 2013;6(2):179-199.

[14] Mousa AA, AL-Matrafi BN. Optimization methodology based on neural networks and reference point algorithm applied to fuzzy multiobjective optimization problems. International Journal of Scientific & Engineering Research. 2013;4(11).

[15] AL-MatrafiB N, Mousa AA. Optimization methodology based on Quantum computing applied to Fuzzy practical unit commitment problem. International Journal of Scientific & Engineering Research. 2013;4(11):1138.

[16] Coello C. An Updated Survey of GA-Based Multi-objective Optimization Techniques. ACM Computing Surveys. 2000; 32(2):109-143.

[17] Das DB, Patvardhan C. New Multi-Objective Stochastic Search Technique for Economic Load Dispatch. IEE Proceedings. C, Generation, Transmission, and Distribution. 1998;145(6):747–752.

[18] Deb K. Multi-objective optimization using evolutionary algorithms. NY, USA: Wiley; 2001.

[19] Fonseca CM, Fleming PJ. An overview of evolutionary algorithms in multiobjective optimization. Evolutionary Computation. 1995;3(1):1-16.

[20] Hwang CL, Yoon K. Multiple Attribute Decision Making: Methods and Applications. Springer-Verlag, New York; 1981.

[21] Olson DL. Comparison of Weights in TOPSIS Models. Mathematical and computer Modeling. 2004;40:721-727.

[22] Laumanns M, Thiele L, Deb K, Zitzler E. Archiving with guaranteed convergence and diversity in multi-objective optimization. in: GECCO 2002, Proceedings of the Genetic and Evolutionary Computation Conference, Morgan Kaufmann Publishers, New York, NY, USA. 2002;439-447.

[23] Zahavi J, Eisenberg L. Economic–environmental power dispatch. IEEE Trans. Syst. Man Cybern. 1985;5(5): 485–489.

[24] Osman MS, Abo-Sinna MA, Mousa AA. Epsilon-dominance based multiobjective genetic algorithm for economic emission load dispatch optimization problem. Electric Power Systems Research. 2009;79:1561–1567.

[25] Hernández-Díaz AG, Santana-Quintero LV, CoelloCoello CA, Caballero R, Molina J. Improving multi-objective evolutionary algorithms by using Rough sets. in A. Ligeza, S. Reich, R. Schaefer and C. Cotta (eds.), Knowledge-Driven Computing: Knowledge Engineering and Intelligent Computations, Studies in Computational Intelligence. 2008;102:81-98.

[26] Osman MS, Abo-Sinna MA, Mousa AA. An effective genetic algorithm approach to multiobjective resource allocation problems (MORAPs). Journal of Applied Mathematics & Computation. 2005;163:755-768.

[27] Murata T, Ishibuch H I, Tanaka H. Multiobjective genetic algorithm and its application to flowshop scheduling. Computers and Industrial Engineering. 1996;30(4):957-968.

[28] Yokoyama R, Bae S H, Morita T, Sasaki H. Multiobjective generation dispatch based on probability security criteria. IEEE Transactions on Power Systems. 1988;3(1):317-324.

[29] Osman MS, Abo-Sinna MA, Mousa AA. IT-CEMOP: An iterative co-evolutionary algorithm for multiobjective optimization problem with nonlinear constraints. Journal of Applied Mathematics & Computation (AMC). 2006;183:373-389.

[30] Azzam M, Mousa AA. Using genetic algorithm and topsis technique for multiobjective reactive power compensation. Electric Power Systems Research. 2010;80:675–68.

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
www.sciencedomain.org/review-history.php?iid=615&id=6&aid=5650