Decomposition-based Modern Metaheuristic Algorithms for Multi-Objective Optimal Power Flow – A comparative study

Miguel A. Medina, Swagatam Das, Carlos A. Coello Coello, and Juan M. Ramírez

***Abstract*—This article presents multi-objective variants of two popular metaheuristics, namely, the artificial bee colony algorithm (ABC) and the teaching learning based optimization algorithm (TLBO). Both of them are used to solve an optimal power flow problem. The proposed multi-objective variants are based on a decomposition approach, where the multi-objective optimization problem is decomposed into a number of scalar optimization sub-problems which are simultaneously optimized. The proposed algorithms are tested on the IEEE 30-bus system with different objectives. In addition, an algorithm based on fuzzy set theory is used to select the best committed solution. The proposed approaches are compared with others metaheuristic algorithms available in the specialized literature. Results indicate that the proposed approaches are highly competitive and also able to generate a well-distributed set of non-dominated solutions for the optimal power flow problem.**

***Index Terms*—Artificial bee colony, Decomposition approach, Multi-objective optimal power flow, Teaching-learning algorithm**

I. Introduction

The optimal power flow (OPF) problem has a significant importance in the power system's operation, planning, economic scheduling, and security. It is a non-linear constrained optimization problem, where the solution attains the control variables optimal adjustment, while at the same time satisfying equality and inequality constraints related to the equipments' rating, in order to optimize a certain objective function.

In general, the optimal power flow problem may include several objective functions, possibly in conflict each other. Such kind of optimization problem has a set of possible solutions (named Pareto optimal set), which represent the best commitment among the objectives [1]. Two major solution approaches may be identified:

(1) The first approach is based on conventional methods. Such as Gradient-based Methods, Non-Linear Programming (NLP), Quadratic Programming (QP), Linear Programming (LP) and Interior Point Methods [2-4], the Weighting Method [5], and the ε-Constraint Method [6].

(2) The second approach is based on the use of metaheuristic algorithms such as the Differential Evolution (DE) [7], the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [8,9], Particle Swarm Optimization (PSO) [10], Harmony search algorithm [11], and the Hybrid Evolutionary Programming Technique [12].

Conventional methods are based on an estimation of the global minimum. However, due to difficulties of differentiability, non-linearity, and non-convexity, these methods may not guarantee to reach the global optimum [13]. Moreover, these methods exhibit some limitations, depending upon the type of problem, e.g., when the objective function is not available in algebraic form. Thus, metaheuristics (from which evolutionary algorithms is a particular subclass) have become a popular choice for solving complex optimization problems, due to their flexibility, generality, and ease of use. Additionally, most metaheuristics require little or no specific domain knowledge.

Modern multi-objective evolutionary algorithms (MOEAs) aim at generating a number of Pareto-optimal solutions as diverse as possible. Indeed, MOEAs need a density estimator that distributes solutions along the Pareto front (e.g., crowding distance, fitness sharing, niching). However, there is evidence that these methods cannot always provide good results, especially when dealing with complex multi-objective problems (MOP) [14, 15].

Recently, a novel MOEA framework called the multi-objective evolutionary algorithm based on decomposition (MOEA/D) [14], has been proposed. MOEA/D decomposes a MOP into several single-objective optimization sub-problems with neighborhood relationship. In this way, a set of optimal solutions is achieved by minimizing each sub-problem instead of using the traditional Pareto ranking methods. This has given rise to a new generation of MOEAs. Nevertheless, the performance of MOEA/D in power system applications has not been fully investigated.

This paper proposes a modified artificial bee colony algorithm and a teaching-learning algorithm in the MOEA/D framework. The proposed approaches are used to solve an

Carlos A. Coello Coello acknowledges support from CONACyT project no. 103570. Juan M. Ramírez acknowledges support from CONACyT projects no. 167933 and 188167.

Miguel A. Medina is a Ph.D. student at CINVESTAV-Unidad Guadalajara, MEXICO, (e-mail: mmedina@gdl.cinvestav.mx). Carlos A. Coello Coello is with the Department of Computer Science at CINVESTAV-Zacatenco, MEXICO (e-mail: ccoello@cs.cinvestav.mx). Juan M. Ramírez is with CINVESTAV-Unidad Guadalajara, MEXICO (e-mail: jramirez@gdl.cinvestav.mx).

Swagatam Das is with the Electronics and Communication Sciences Unit of Indian Statistical Institute, Kolkata 700108, India. (e-mail: swagatam.das@isical.ac.in).

optimal power flow problem, with competing objectives.

In order to minimize the total fuel cost, the active power losses and a voltage stability index [16], the proposed algorithms estimate the following optimal values: (*i*) the generators' voltage magnitudes; (*ii*) generators' active power outputs, (*iii*) transformers' tap settings; (*iv*) the compensating value for shunt elements (reactors/capacitors). In addition, an algorithm based on fuzzy set theory is used to select the best committed solution.

The effectiveness of the proposed approaches is demonstrated and compared with respect to a MOEA based on decomposition, which is representative of the state-of-the-art in the area: MOEA/D-DRA [17]. Results are also compared with respect to the NSGA-II [9], which remains as the most popular Pareto-based MOEA. The methods are applied on an IEEE 30-bus test system. Additionally, results reported in the open research [7, 11] are also included for a comparative study.

The rest of the paper is organized as follows. Section II presents some basic background. In section III, the general framework of the proposed approaches is summarized. Section IV presents the problem formulation and the method based on fuzzy theory for choosing the best committed solution. Simulation results and a comparative study are presented in section V. Finally, our conclusions are provided in Section VI.

II. Preliminaries

A. Multi-objective optimization

A multi-objective optimization problem (MOP) is formulated as follows:

$$\begin{aligned} &\min \quad F(x) = \{f_1(x), ..., f_m(x)\} \\ &\text{subject to} \quad x \in \Omega \end{aligned} \tag{1}$$

where x is the vector of decision variables, and Ω is the feasible region within the decision space. $F: \Omega \rightarrow \Re^m$ is defined as the m objective functions mapping.

In multi-objective optimization, the goal is to find the best possible trade off among the objectives since, frequently, one objective can be improved only at the expense of worsening another. To describe the concept of optimality for problem (1) the following definitions are provided.

Definition 1: Let $x, y \in \Omega$, such that $x \neq y$, we say that x dominates y (denoted by $x \prec y$) if and only if, $f_i(x) \leq f_i(y)$ for all i = 1, ..., m.

Definition 2: Let $x^* \in \Omega$, we say that x^* is a Pareto optimal solution, if there is no other solution $y \in \Omega$ such that $y \prec x^*$.

Definition 3: The Pareto Optimal Set ($\overrightarrow{PS}$) is defined by $\overrightarrow{PS} = \{x \in \Omega | x \text{ is Pareto Optimal Solution}\}$, while its image $\overrightarrow{PF} = \{F(x) | x \in \overrightarrow{PS}\}$ is called the Pareto Optimal Front.

B. Decomposition of a multi-objective optimization problem

There are several approaches for transforming a MOP into a number of scalar optimization problems, which have been described in detail in [18]. Usually, these methods use a weighting vector to define a scalar function and, under certain assumptions (e.g., the minimum is unique, the weighting coefficients are positive, etc.), a Pareto optimal solution is achieved by minimizing such function. In this paper, the weighted *Tchebycheff* approach is used to decompose the MOP. In this approach, the scalar optimization problem is stated as [18]:

$$\begin{aligned} &\text{Minimize} \quad g\left(x | w, z^*\right) = \max_{i \in \{1,..,m\}} \left\{ w_i \left| f_i(x) - z_i^* \right| \right\} \\ &\text{Subject to} \quad x \in \Omega \end{aligned} \tag{2}$$

where $w = (w_1,..,w_m)$ is a weighting vector and $w_i \geq 0$ for all i = 1,..., m. $\Sigma w_i = 1$ and vector $z^* = (z^*_1, \ldots, z^*_m)$ represents the reference point, i. e., $z_i^* = \min\{f_i(x) | x \in \Omega\}$, i = 1, …, m, where m is the number of objective functions.

For each Pareto-optimal solution x^* there exists a weighting vector w such that x^* is the optimal solution of (2), and each optimal solution is a Pareto-optimal solution for (1). Therefore, it is possible to obtain different Pareto optimal solutions using different weighting vectors w.

C. Modified Artificial Bee Colony

The first framework of the Artificial Bee Colony (ABC) was introduced by Karaboga in 2005 as a new swarm intelligent technique inspired by the foraging behavior of a honey bee swarm [19]. In ABC, a colony of artificial bees consists of three groups of bees: employed bees, onlooker bees, and scout bees. In the algorithm, the position of a food source represents a possible solution to the optimization problem, and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. Each food source is exploited by only one employed bee. In other words, the number of employed bees is equal to the number of food sources existing around the hive (number of solutions in the population). The employed bee whose food source has been abandoned becomes a scout.

Akay and Karaboga [20] proposed some modifications to the standard ABC algorithm in order to improve the convergence rate. The pseudo-code of the modified ABC algorithm proposed by Akay and Karaboga can be summarized in the following way [20]:

```
initialization
Evaluation
cycle = 1:
repeat
   Employed Bees Phase
   Calculate probability for Onlookers
   Onlooker Bees Phase
   Scout Bee Phase
   Memorized the best solution achieved so far
cycle = cycle + 1
Until cycle = Maximum Cycle Number
```

D. Teaching-Learning based optimization algorithm

The original teaching learning based optimization (TLBO) algorithm was proposed by Rao *et al*. [21] to obtain global solutions for continuous non-linear functions. In TLBO, the design variables are analogous to different subjects offered to learners. The learners' grade is analogous to the '*fitness*' as in any other evolutionary algorithm, and the teacher is considered as the best solution reached so far. Hence, the TLBO's performance is based on two main phases: the teacher phase, which involves learning from the teacher, and the learner phase, which involves learning through the interaction among learners. The pseudo-code of the TLBO algorithm may be summarized in the sequel.

1: Initialization
2: Evaluation
3: iteration = 1
4: **Repeat**
5: Teacher Phase
6: Keep the best solutions
7: Learner Phase
8: Keep the best solutions
9: iteration = iteration + 1
10: **Until** iteration = Maximum number of iterations

III. Multi-Objective Algorithms based on Decomposition

A. Multi-objective Artificial Bee Colony

The proposed Multi-Objective Artificial Bee Colony Algorithm based on Decomposition (MOABC/D) utilizes the *Tchebycheff* approach to decompose the MOP into N scalar optimization sub-problems by choosing N weighting vectors: $w^i = \{w_1^i,...,w_m^i\}$, where i = 1,…, N and m = number of objective functions. For two objective functions, i.e., $m = 2$, $w^i = (w_1^i, w_2^i)$ can be set as:

$$w_1^i = (j-1)/(N-1), \quad w_2^i = 1 - w_1^i \tag{3}$$

This method for generating weighting vectors works well for the formulation in this paper. However other methods may be used, as well.

The i-th sub-problem is associated with weighting vector w^i and its scalar function is denoted as $g(x \mid w^i)$. MOABC/D solves these sub-problems simultaneously by evolving a population of solutions that mimics the intelligent behavior of a honey bee swarm in a similar way as that described in the modified ABC [20]. Because $g(x \mid w)$ is a continuous function of w, two sub-problems are likely to have similar solutions if their weighting vectors are close from each other [14]. Therefore, any information about the weighting vectors close to w^i should be helpful for optimizing $g(x \mid w^i)$. Based on this observation, the neighborhood $B(i)$ of sub-problem i contains the T_n sub-problems with the closest weighting vectors with respect to w^i. The Euclidean distance is used to measure the closeness between any two weight vectors and it is assumed that $i \in B(i)$, that is, the i-th sub-problem is its own neighbor. The neighborhood of each sub-problem represents an artificial colony and the group of bees: employed, onlooker, and scout bees are responsible to solve each sub-problem by using the information from its neighboring sub-problems. The neighborhood size T_n should be much smaller than the population size N [14]. The main steps of the proposed MOABC/D are summarized in the sequel.

1) Initial population: firstly, the algorithm generates a randomly distributed initial population (food sources position) within the range of the parameters' boundaries [$x_j^{\max}$, $x_j^{\min}$],

$$x_{i,j} = x_j^{\min} + rand(0,1) \cdot (x_j^{\max} - x_j^{\min}) \tag{4}$$

where i = 1,..., N, j = 1,.., D. N is the number of food sources (potential solutions) and D is the number of optimization parameters. In addition, counters which store the number of trials of solutions are reset to 0 in this phase.

2) Selection of the artificial colony: for the i-th sub-problem, the artificial colony is selected between the neighborhood $B(i)$ and the population N according to,

$$C_{i\,\mathrm{th}} = \begin{cases} B(i) & if\ rand < \delta \\ \{1,...N\} & otherwise \end{cases} \tag{5}$$

where *rand* is a random number within [0,1] and δ the probability to select the neighborhood $B(i)$ as the colony.

3) Employed phase: in this phase, the artificial colony of the i-th sub-problem may be expressed as:

$$C_{i\,\mathrm{th}} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,D} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,D} \\ \vdots & \vdots & \ddots & \vdots \\ x_{\Omega_T,1} & x_{\Omega_T,2} & \cdots & x_{\Omega_T,D} \end{bmatrix} \tag{6}$$

where the subscript Ω_T is the size of the artificial colony, and D is the number of design variables. An employed bee produces a modification in the food source position (x_i) by finding a new food source (new solution), and then evaluates the new food source's quality (fitness value). The new food source ($x_{new,i}$) is generated by [20]:

$$x_{new\,i,j} = \begin{cases} x_{i,j} + \phi_{i,j}(x_{i,j} - x_{k,j}) & if\ R_{i,j} < MR \\ x_{i,j} & otherwise \end{cases} \tag{7}$$

By means of this modification, for each parameter $x_{i,j}$, a uniformly distributed random number ($R_{i,j}$) within the interval [0,1] is generated, then parameter $x_{i,j}$ is modified by (7) where MR is the modification rate, j is a random integer within the range [1,D] and $k \in \{1,...,\Omega_T\}$ is randomly chosen index that has to be different from i. $\phi_{i,j}$ is a random number in the interval [-1,1]. If a parameter value of the new food source found by this operation exceeds its predetermined boundaries, the parameter is set to an acceptable value.

After generating a new solution ($x_{new,i}$) within the boundaries, a fitness value for a minimization problem is assigned to the new solution given by:

$$fitness_i = \begin{cases} 1/(1+f_i) & if\ f_i \geq 0 \\ 1+abs(f_i) & if\ f_i \leq 0 \end{cases} \tag{8}$$

where f_i is the cost value of the new solution. A greedy selection is applied between x_i and $x_{new,i}$; therefore, the better one is selected depending on its fitness values. If the new solution $x_{new,i}$ is equal or better than the old one x_i in terms of quality, the employed bee memorizes the new position and forgets the old one. Otherwise the previous position is kept in memory. If solution x_i cannot be improved, the number of *trials* is incremented by 1; otherwise, the counter is reset to 0.

After all employed bees complete their searching procedures, an onlooker bee evaluates the nectar information gathered from all employed bees and assigns a probability to each food source. This probability depends on the fitness values of the solutions and is calculated by the following expression,

$$p_i = \frac{fitness_i}{\sum_{i=1}^{N} fitness_i} \tag{9}$$

4) Onlooker phase: In this phase, a random number (r_j) within the interval [0, 1] is generated for each source. If the probability value p_j associated with that source is greater than r_j then the onlooker bee produces a modification on the position of this food source by,

$$x_{newi,d} = \begin{cases} x_{i,d} + \phi_{i,d}(x_{j,d} - x_{k,d}) & if \;\; R_{i,d} < MR \\ x_{i,d} & otherwise \end{cases} \tag{10}$$

where index i corresponds to the current index of i-th sub-problem, x_j is the food source, which probability value (p_j) is greater than r_j; the index d is a random integer within the interval [1, D]. ($\phi_{i,d}$) is a random number between [-1, 1], ($R_{i,d}$) is an uniformly distributed random number within [0,1] and the *MR* is the modification rate.

Similarly to the employed phase, after the new source is evaluated, greedy selection is applied. If solution x_i cannot be improved, the number of trials is incremented by 1; otherwise, the counter is reset to 0.

5) Scout phase: In a cycle, after all employed bees and onlooker bees complete their searching, the algorithm verifies if there is any exhausted source to be abandoned. In order to decide if a source is to be abandoned, the counters which have been updated during searching are used. If the value of the counter is greater than the control parameter, known as the "*limit*", then the source associated with this counter is abandoned. The food source abandoned by its bee is replaced with a new food source discovered by the scout. This is simulated by randomly produce a new position by using (4) and replacing it with the abandoned one. If more than one counter exceeds the "*limit*" value, one of the maximum ones might be chosen.

6) Updating strategy: Set $n = 0$ and then do [15]:

(a) $if \quad n = s_r \quad or \quad C$ is empty

break. Otherwise *random* pick an index j from C

(b) $if \;\; g(x_{new} | w^j) \le g(x_j | w^j)$, Then set $x_j = x_{new}$ (11)

and $n = n + 1$

(c) remove j form C and go to (a)

where (s_r) is the maximum number of solutions replaced by the new solution.

Summarizing, the proposed MOABC/D algorithm can be described in pseudo-code format in the following way:

Step1) Initialization
- Generate a well-distributed set of N weighting vectors by (3)
- Find the neighborhood of each sub-problems $B(i)$
- Set *trial* (i) = 0, for i = 1,.., N
- Generate the initial population according to (4) and evaluate its fitness.
- Initialize the reference point z^*

Step 2)
For i = 1 to N
- Determine the colony (C_i) according to (5)
- **Employed phase**: Create a new solution ($x_{new,i}$) by (7)
- Update the reference point z^*
- Update solutions by (11)

End For i

Step 3)
For i = 1 to N
- Determine the colony (C_i) according to (5)
- **Onlooker Phase**: Create a new solution ($x_{new,i}$) by (10)
- Update the reference point z^*
- Update solutions by (11)

End For i

Step 4) Scout Phase

Step 5) Stop Criterion: **If** the stop condition is satisfied, **then** stop MOABC/D. Otherwise, go to **Step 2)**.

B. Multi-objective Teaching Learning algorithm

The proposed Multi-Objective Teaching-Learning Algorithm based on Decomposition (MOTLA/D) utilizes the *Tchebycheff* approach to decompose the MOP into N scalar optimization sub-problems, in an analogous way to the procedure described in the previous section. MOTLA/D solves these sub-problems simultaneously; the neighborhood relationships among these sub-problems are defined by computing the minimum Euclidean distances between the weighting vectors. In this case, the neighborhood of each sub-problem represents a group of learners or a class, responsible to solve such sub-problem.

The main steps of the proposed MOTLA/D may be summarized as follows:

1) Initial learners: At the first step, the algorithm generates a randomly distributed initial population (learners) within the range of the parameters' boundaries according to (4).

2) Selection of the Class: The size of the class is selected between $B(i)$ and the population size N by (5).

3) Teacher phase: In this phase, the class of the i-th sub-problem may be expressed as,

$$C_{i\,th} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,D} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,D} \\ \vdots & \vdots & \ddots & \vdots \\ x_{\Omega_T,1} & x_{\Omega_T,2} & \cdots & x_{\Omega_T,D} \end{bmatrix} \tag{12}$$

where Ω_T is the size of the class, and D is the number of design variables. Within the teacher phase, the mean of the class (M_{class}) for each design variable is calculated column-wise:

$$M_{class} = [mean_1, mean_2, ..., mean_D] \tag{13}$$

The teacher (M_{new}) for the i-th sub-problem represents the best learner of the class $C_{i,th}$. Thus, the teacher is determined by,

$$M_{new} = \{x_i \,|\min_{x_i \in \Omega_T} g(x_i \,|w^i, z^*)\} \tag{14}$$

The solutions are updated according to the difference between the mean of the class (M_{class}) and the new mean (M_{new}) by

$$x_{new,i} = x_i + r_i(M_{new} - T_F M_{class}) \tag{15}$$

where index i corresponds to the current index of i-th sub-problem, r_i is a random number within the interval [0, 1]. T_F is the teaching factor, which value can be either 1 or 2; this is decided randomly with equal probability as T_F = round [1 + rand (0, 1)]. The new solution (x_{new}) is accepted if it gives a better function value.

4) Learner phase: In this phase, for the i-th sub-problem, two learners x_j, and x_k are selected randomly such that $i \neq j \neq k$. A new solution (x_{new}) is generated as follows,

$$\begin{aligned} &if \quad f(x_j) < f(x_k) \\ &\quad x_{new} = x_i + r_i(x_j - x_k) \\ &else \\ &\quad x_{new} = x_i + r_i(x_k - x_j) \\ &end \end{aligned} \tag{16}$$

Additionally, a polynomial mutation operator is applied to maintain solutions' diversity. The new solution (x_{new}) is accepted if it gives a better function value. If a parameter value produced in the teacher o learner phase exceeds its predetermined boundaries, the parameter is set to an acceptable value.

5) Updating strategy: For updating the solutions to the i-th sub-problem, we adopt (11).

The proposed MOTLA/D may be summarized in the sequel.

Step1) Initialization
- Generate a well-distributed set of N weighting vectors by (3)
- Find the neighborhood of each sub-problem $B(i)$
- Set *trial* (i) = 0, for i = 1,.., N
- Generate the initial population according to (4) and evaluate its fitness.
- Initialize the reference point z^*

Step 2)

For i = 1 to N
- Determine the class (C_i) according to (5)
- **Teacher phase**: Create a new solution ($x_{new,i}$) by (15)
- Update the reference point z^*
- Update solutions by (11)

End For i

Step 3)

For i = 1 to N
- Determine the class (C_i) according to (5)
- **Learner Phase**: Create a new solution ($x_{new,i}$) by (16)
- Update the reference point z^*
- Update solutions by (11)

End For i

Step 4) Stop Criterion: **If** the stop condition is satisfied, **then** stop MOTLA/D. Otherwise, go to **Step 2)**.

C. Modified Phases of MOABC/D and MOTLA/D

The onlooker phase and the learner phase of the ABC [19, 20] and TLBO [21], respectively, create a new solution from the random selection of two parents. This strategy may increase the probability that algorithms remain trapped in local minima. Therefore, to prevent premature convergence and to avoid getting trapped in local minima, a new strategy had to be implemented. In these phases (onlooker and learner), in order to create a new solution for the i-th sub-problem, three parents (x_i, x_j, and x_k) are selected such that $x_i \neq x_j \neq x_k$. Additionally, in the learner phase, a polynomial mutation operator is applied to maintain the solutions' diversity.

IV. Problem Statement

In this paper, an optimal power flow problem is formulated as a multi-objective optimization problem where three objective functions are taken into account for minimization, while satisfying a number of equality and inequality constraints. The problem is formulated in the sequel.

A. Objective functions

A.1 Fuel cost minimization

The objective is to minimize the generation total fuel cost F_{cost}. The generators' fuel cost curves are modelled by quadratic functions and the total fuel cost F_{cost} in ($/h) may be expressed by,

$$F_{\text{cost}} = \sum_{i=1}^{N_g} a_i + b_i P_{gi} + c_i P_{gi}^2 \tag{17}$$

where N_g is the number of generators; a_i, b_i, and c_i are the cost coefficients of the i-th generator, and P_{gi} is the corresponding active power output.

A.2 Active power losses minimization

The objective is to minimize the active power losses (P_{loss}) through the transmission lines, which are calculated by,

$$P_{loss} = \sum_{k=1}^{nl} g_k [V_i^2 + V_j^2 - 2V_i V_j \cos(\theta_i - \theta_j)] \tag{18}$$

where nl is the number of transmission lines, g_k is the conductance of the k-th transmission line connecting the i-th and j-th bus; V_i, V_j, θ_i, and θ_j are the voltages magnitudes and phase angles of i-th and j-th bus, respectively.

A.3. Voltage stability enhancement

A conventional way for the voltage stability assessment is the use of indexes, which estimate the proximity to voltage instability and determine those buses exhibiting weak stability. Nowadays there is a variety of indexes that help to assess the steady state voltage stability [22].

In this research, voltage stability enhancement is achieved through minimizing the voltage stability index L_{index} [23], which is able to evaluate the steady state voltage stability margin of each bus. The L_{index} value lies between 0 (no load) and 1 (voltage collapse). This value implicitly includes the load effect. The bus with the highest L_{index} value will be the most vulnerable, and therefore, this method helps to identify weak areas that require reactive power critical support. The L_{index} is calculated in the following way [23]:

The network equations in terms of the bus admittance matrix can be written as,

$$I_{bus} = Y_{bus} V_{bus} \tag{19}$$

The buses are broken down into two categories: (*i*) the set of load buses (α_L); and (*ii*) the set of generator buses (α_G). Thus, equation (16) becomes,

$$\begin{bmatrix} I^L \\ I^G \end{bmatrix} = \begin{bmatrix} Y_1 & Y_2 \\ Y_3 & Y_4 \end{bmatrix} \cdot \begin{bmatrix} V^L \\ V^G \end{bmatrix} \tag{20}$$

It is assumed that the transmission system is linear and allows a representation in terms of a hybrid matrix H:

$$\begin{bmatrix} V^L \\ I^G \end{bmatrix} = H \cdot \begin{bmatrix} I^L \\ V^G \end{bmatrix} = \begin{bmatrix} Z^{LL} & F^{LG} \\ K^{GL} & Y^{GG} \end{bmatrix} \cdot \begin{bmatrix} I^L \\ V^G \end{bmatrix} \tag{21}$$

where V^L and I^L are voltage and current vectors for load buses; V^G and I^G are voltage and current vectors for generator buses; Z^{LL}, F^{LG}, K^{GL}, and Y^{GG} are sub-matrices of the hybrid matrix H.

Matrix H is generated from the admittance matrix (Y_{bus}) by a partial inversion, where the load buses voltage's vector is exchanged for the current's vector. This representation may then be utilized to define a voltage stability index in the load bus, namely L_j which is defined by [23],

$$L_j = \left| 1 - \frac{\sum_{i \in \alpha_G} F_{ji} V_i}{V_j} \right| \tag{22}$$

For stable conditions, $0 \le L_j \le 1$, must not be violated for any j. Hence, a global indicator L_{index} describing the whole system's stability is defined by [23],

$$L_{index} = \max_{j \in \alpha_L} (L_j) \tag{23}$$

The L_{index} in (23) is associated with the worst bus in the sense of voltage stability. The L_{index} minimization implies to take such bus toward a less stressed condition.

B. Constraints

1) Equality constraints:

The equality constraints are the balance of the active and reactive power described by the set of power flow equations. They may be expressed as follows,

$$P_{gi} - P_{di} - \sum_{j=1}^{N_b} |V_i||V_j||Y_{ij}| \cos(\delta_i - \delta_j - \theta_{ij}) \tag{24}$$

$$Q_{gi} - Q_{di} - \sum_{j=1}^{N_b} |V_i||V_j||Y_{ij}| \sin(\delta_i - \delta_j - \theta_{ij}) \tag{25}$$

where, N_b is the number of buses, P_{gi} is the i-th active power generation, Q_{gi} is the i-th reactive power generation, P_{di} is the i-th active power load, Q_{di} is the i-th reactive power load, and $|Y_{ij}|$ is the ij-th element of the bus admittance matrix. These equality constraints are handled by running the power flow program.

2) Inequality constraints

These constraints represent the system operating limits as follows,

A) Generators: these constraints are associated to the generator voltages (V_g), active power output (P_g), and reactive power output (Q_g),

$$V_{gi}^{\min} \le V_{gi} \le V_{gi}^{\max}, \quad i = 1, ..., N_g \tag{26}$$

$$P_{gi}^{\min} \le P_{gi} \le P_{gi}^{\max}, \quad i = 1, ..., N_g \tag{27}$$

$$Q_{gi}^{\min} \le Q_{gi} \le Q_{gi}^{\max}, \quad i = 1, ..., N_g \tag{28}$$

where N_g is the number of generators.

B) Transformers: Transformers tap settings are restricted by their minimum and maximum limits as follows:

$$T_i^{\min} \le T_i \le T_i^{\max}, \quad i = 1, ..., N_t \tag{29}$$

where N_t is the number of transformers.

C) Shunt VAR: Reactive power injections at buses are restricted by their minimum and maximum limits as follows:

$$Q_{ci}^{\min} \le Q_{ci} \le Q_{ci}^{\max}, \quad i = 1, ..., N_c \tag{30}$$

where N_c is the number of shunt VAR sources.

D) Load bus voltage: each load bus is restricted by its limits as follows:

$$V_i^{\min} \le V_i \le V_i^{\max}, \quad i = 1, ..., N_{PQ} \tag{31}$$

where N_{PQ} is the number of load buses.

C. Decision variables

The decision variables include the generator voltages (V_g), generator active power outputs (P_g) except at the slack bus P_{g1}, transformers tap settings (T), and shunt VAR compensations. Hence, the vector of control variables (u) is expressed as,

$$u = [V_{g1}, ..., V_{g_{Ng}}, P_{g2}, ..., P_{g_{Ng}}, T_1, ..., T_{Nt}, Q_{c1}, ..., Q_{c_{Nc}}] \tag{31}$$

It is worth noting that the decision variables are self-constrained by the optimization algorithm.

D. Best committed solution

Having obtained the Pareto optimal solution, choosing a best committed solution is important in decision making process. In this research, a technique based on fuzzy set theory is applied to find the best committed solution. In this technique, the i-th objective function f_i of the Pareto optimal solution k is represented by a membership function μ_i^k defined as [24],

$$\mu_i^k = \begin{cases} 1 & f_i \le f_i^{\min} \\ \dfrac{f_i^{\max} - f_i}{f_i^{\max} - f_i^{\min}} & f_i^{\min} < f_i < f_i^{\max} \\ 0 & f_i \ge f_i^{\max} \end{cases} \tag{32}$$

where $f_i^{\max}$ and $f_i^{\min}$ are the maximum and minimum value of i-th objective function, respectively.

For each non-dominated solution k, the normalized member-ship function μ^k is calculated as,

$$\mu^k = \frac{\sum_{i=1}^{N_{obj}} \mu_i^k}{\sum_{k=1}^{M} \sum_{i=1}^{N_{obj}} \mu_i^k} \tag{33}$$

where N_{obj} is the number of objective functions; M is the total number of Pareto optimal solutions, and μ^k is the member-ship value of the non-dominated solution k.

The best compromise solution is the one achieving the maximum value for μ^k.

E. MOABC/D and MOTLA/D for solving the OPF problem

The initial population for both algorithms consists of the vector of decision variables (31) and is generated by (4): $x_{i,j} = [V_{g1},..,P_{g2},..,T_1,..,Q_{c1},...]$, i = 1,…,N and j = 1,…,D. where N represents the population size and D is the number of design variables. This generates N individuals for solving the MOP. During the initialization, the decision variables are randomly generated within the allowable intervals. The minimum and maximum values for each decision variable are exhibited in the sequel.

The dependent variables are handled by adding them as the quadratic penalty terms to the objective functions to form a generalized objective functions:

$$\begin{aligned} f_1 &= F_{\cos t} + K_L \sum_{N_{PQ}} \Delta V_{L_i}^2 + K_G \sum_{N_g} \Delta Q_{G_i}^2 \\ f_2 &= P_{loss} + K_L \sum_{N_{PQ}} \Delta V_{L_i}^2 + K_G \sum_{N_g} \Delta Q_{G_i}^2 \\ f_3 &= L_{index} + K_L \sum_{N_{PQ}} \Delta V_{L_i}^2 + K_G \sum_{N_g} \Delta Q_{G_i}^2 \end{aligned} \quad (34)$$

where K_L and K_G are defined as penalty factors. ΔV_L and ΔQ_G are defined by,

$$\Delta V_{L_i} = \begin{cases} V_{L_i}^{\min} - V_{L_i} & \text{if } (V_{L_i} < V_{L_i}^{\min}) \\ 0 & \text{if } (V_{L_i}^{\min} \le V_{L_i} \le V_{L_i}^{\max}), i = 1,..., N_{PQ} \\ V_{L_i} - V_{L_i}^{\max} & \text{if } (V_{L_i} > V_{L_i}^{\max}) \end{cases}$$

$$\Delta V_{G_i} = \begin{cases} Q_{G_i}^{\min} - Q_{G_i} & \text{if } (Q_{G_i} < Q_{G_i}^{\min}) \\ 0 & \text{if } (Q_{G_i}^{\min} \le Q_{G_i} \le Q_{G_i}^{\max}), i = 1,..., N_g \\ Q_{G_i} - Q_{G_i}^{\max} & \text{if } (Q_{G_i} > Q_{G_i}^{\max}) \end{cases} \quad (35)$$

The flowchart for MOABC/D and MOTLA/D is summarized in Figure 1. An important difference between the original scheme of decomposition in MOEA/D [14] and the proposed one, is the number of stages. Whilst MOEA/D uses one stage to apply the DE strategy [15], in the proposed scheme the main phases of each algorithm are in different stages, see Fig. 1. The applied phase depends on the used algorithm. This scheme allows that each phase explores the solution space individually, and also helps to compose hybrid algorithms easily by including the desired strategy within the stages. However, this topic is not addressed in this paper.

V. Simulation Results and comparison

In order to assess the effectiveness of the proposed algorithms, the MOABC/D and MOTLA/D have been compared with respect to the MOEA/D-DRA [17] and NSGA-II [9]. The algorithms have been tested in the IEEE 30 bus system. This system consists of six generators, four transformers with off-nominal tap ratio, and nine reactive power injection sources. The complete system data are given in [2]. The optimization problem has 24 parameters.

The parameters used in each algorithm are summarized in Table 1, where N_{pop} represents the population size; S_r, is the number of solutions which are replaced in the neighborhood; T_n defines the neighborhood size; Cr is the crossover rate. For MOABC/D the Cr is equal to its MR parameter. F is the scaling factor used in MOEA/D-DRA [17]. η_m, is the mutation index. For the algorithms using the mutation operator, the mutation rate (P_m=1/n) is applied, where n is the number of decision variables of the problem; δ is the probability of selecting solutions from the neighborhood. For MOEA/D-DRA, π_s and Δ_r, represent the percentage selection and decay rate for the utility, respectively; *limit* is the algorithm

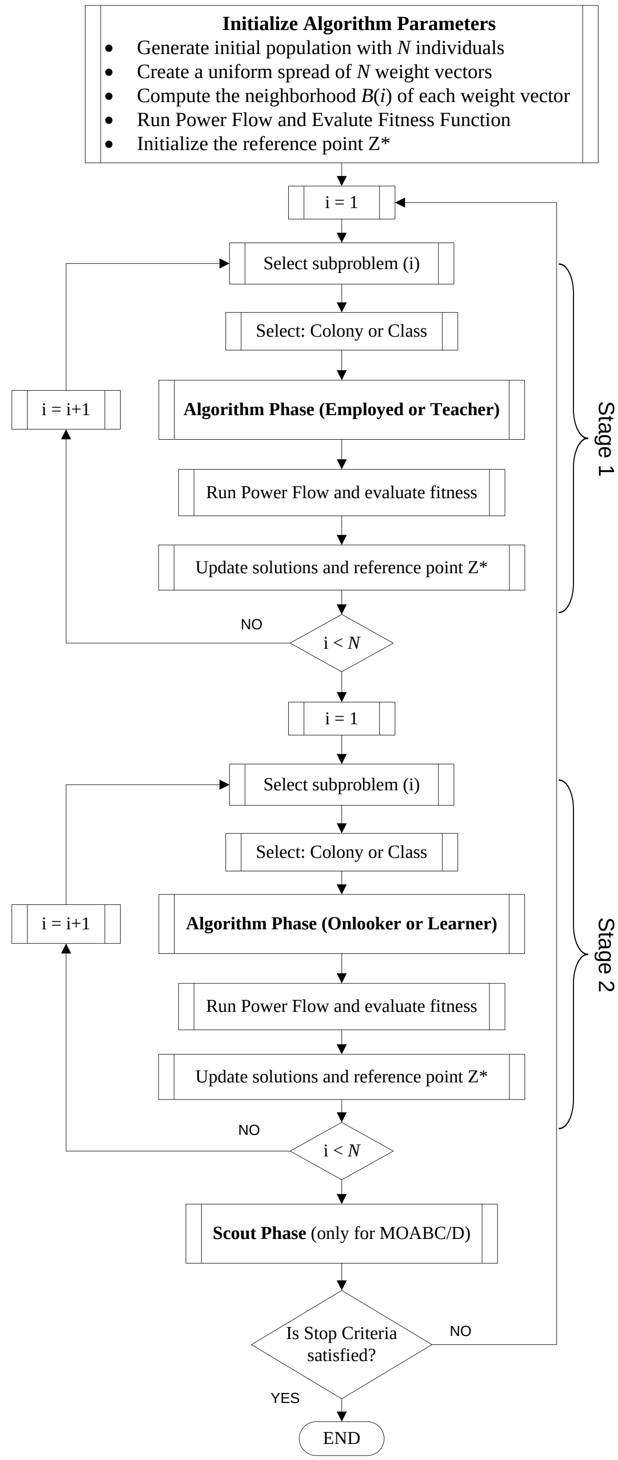

Figure. 1. Flowchart of MOABC/D and MOTLA/D.

TABLE 2. Best solutions for case study 1.

Variable	Limits Min	Limits Max	MOABC/D Min Cost	MOABC/D Min L_{index}	MOABC/D Best Comp	MOTLA/D Min Cost	MOTLA/D Min L_{index}	MOTLA/D Best Comp	MOEAD/DRA Min Cost	MOEAD/DRA Min L_{index}	MOEAD/DRA Best Comp	NSGA-II Min Cost	NSGA-II Min L_{index}	NSGA-II Best Comp
V_1(p.u)	0.95	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.0999	1.1
V_2(p.u)	0.95	1.1	1.087	1.089	1.089	1.087	1.093	1.090	1.087	1.094	1.090	1.088	1.1	1.0924
V_5(p.u)	0.95	1.1	1.061	1.067	1.064	1.062	1.080	1.068	1.061	1.072	1.066	1.063	1.1	1.0725
V_8(p.u)	0.95	1.1	1.070	1.080	1.073	1.0703	1.090	1.077	1.069	1.088	1.076	1.072	1.1	1.086
V_{11}(p.u)	0.95	1.1	1.1	1.099	1.099	1.1	1.099	1.099	1.1	1.1	1.1	1.094	1.0987	1.0946
V_{13}(p.u)	0.95	1.1	1.1	1.099	1.1	1.1	1.1	1.1	1.1	1.097	1.099	1.092	1.1	1.0939
P_2(MW)	20	80	48.686	48.779	48.745	48.662	49.537	48.673	48.809	48.561	48.723	48.742	44.495	47.866
P_5(MW)	15	50	21.301	21.378	21.308	21.322	21.444	21.331	21.332	21.175	21.221	21.451	23.355	21.564
P_8(MW)	10	35	20.956	20.918	21.069	20.963	21.085	21.039	20.866	21.254	20.984	20.815	25.411	21.817
P_{11}(MW)	10	30	11.873	11.935	11.916	11.818	12.085	12.076	11.897	12.868	12.170	11.927	14.727	13.112
P_{13}(MW)	12	40	12.000	12.001	12.001	12	12.028	12.005	12.000	12.011	12.003	12.001	14.319	12.022
$T_{6\text{-}9}$	0.9	1.1	1.017	1.023	1.021	1.025	0.983	1.014	1.015	0.995	1.000	0.993	0.990	0.974
$T_{6\text{-}10}$	0.9	1.1	0.905	0.900	0.9	0.900	0.934	0.908	0.907	0.922	0.917	0.947	0.923	0.949
$T_{4\text{-}12}$	0.9	1.1	0.968	0.976	0.975	0.969	0.982	0.977	0.968	0.969	0.975	0.986	0.978	0.973
$T_{28\text{-}27}$	0.9	1.1	0.956	0.955	0.955	0.958	0.960	0.958	0.955	0.959	0.956	0.970	0.965	0.970
Qc_{10}(Mvar)	0	5	4.845	4.576	4.871	4.738	0.027	4.505	4.980	4.687	4.833	2.842	4.274	2.291
Qc_{12}(Mvar)	0	5	4.985	4.758	4.989	4.994	4.836	4.942	4.998	3.803	4.670	4.837	2.804	4.974
Qc_{15}(Mvar)	0	5	4.993	4.115	4.785	4.775	4.918	4.990	4.999	4.413	4.840	4.307	2.554	4.535
Qc_{17}(Mvar)	0	5	4.996	4.342	4.965	4.999	4.215	4.664	4.991	3.608	4.634	4.359	3.276	4.544
Qc_{20}(Mvar)	0	5	3.721	3.661	3.871	3.988	4.936	4.000	3.690	1.321	3.052	3.237	3.498	2.807
Qc_{21}(Mvar)	0	5	4.992	4.848	4.999	4.993	4.986	4.994	4.999	4.994	4.995	5	0.160	5
Qc_{23}(Mvar)	0	5	2.272	2.422	2.260	2.42	1.547	1.739	2.445	0.801	2.018	4.023	0.496	4.593
Qc_{24}(Mvar)	0	5	4.999	4.903	4.992	4.999	4.278	4.946	4.999	4.914	4.977	3.994	4.488	3.648
Qc_{29}(Mvar)	0	5	1.640	0.002	0.855	1.832	0.102	0.886	1.446	0.29	0.852	2.459	0.003	3.002
OBJECTIVE FUNCTIONS' VALUES														
Fuel Cost ($/h)			**799.043**	799.186	**799.064**	**799.046**	799.58	**799.116**	**799.044**	799.487	**799.111**	**799.135**	801.941	**799.446**
L_{index}			0.1292	**0.1286**	**0.1288**	0.1292	**0.1283**	**0.1287**	0.1292	**0.1284**	**0.1287**	0.1315	**0.1279**	**0.1287**

TABLE 1. Algorithms' control parameters

Parameter	MOABC/D	MOTLA/D	MOEA/D-DRA	NSGA-II
N_{pop}	100	100	100	100
T_n	30	30	30	-
S_r	3	3	3	-
δ	0.9	0.9	0.9	0.9
Cr	0.5	-	1	0.9
limit	15	-	-	-
η_m	-	20	20	20
F	-	-	0.5	-
π_s	-	-	5	-
Δ_r	-	-	0.95	-

parameter of the MOABC/D. It is worth mentioning that the stop condition of each algorithm is the number of function evaluations, (30,000 function evaluations).

Two different cases for the minimization problems are considered in the sequel.

A. Case study 1: Fuel cost and L_{index}

In this case, two competing objective functions, i.e., *fuel cost* and the voltage stability indicator L_{index}, are considered. The multi-objective optimization problem is solved by the proposed methods, and also through MOEA/D-DRA and NSGA-II.

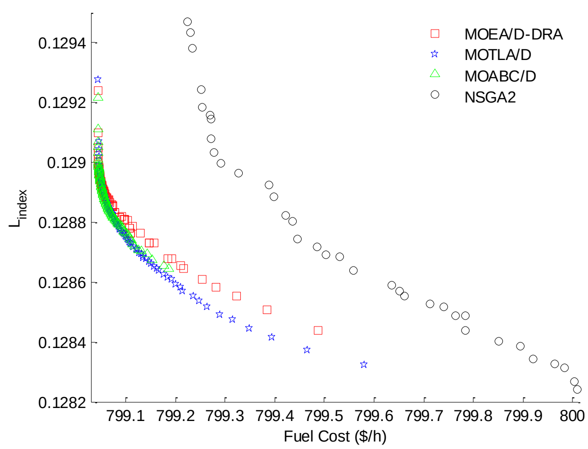

Figure. 2. Pareto optimal solution for case study 1.

The Pareto optimal set of the solutions is shown in Fig. 2. From the Pareto optimal solution, it is clear that the proposed MOABC/D and MOTLA/D are able to give well-distributed solutions and better convergence than MOEA/D-DRA and NSGA-II. Also, it is clear that methods based on decomposition, i.e., MOABC/D, MOTLA/D, and MOEA/D-DRA have better convergence with respect to the method based on Pareto ranking, i.e., NSGA-II.

TABLE 3. Best solutions for case study 2.

	Limits		MOABC/D			MOTLA/D			MOEAD/DRA			NSGA-II		
Variable	Min	Max	Min Cost	Min Loss	Best Comp	Min Cost	Min Loss	Best Comp	Min Cost	Min Loss	Best Comp	Best Cost	Best Loss	Best Comp
V_1(p.u)	0.95	1.1	1.099	1.099	1.099	1.1	1.0965	1.1	1.099	1.099	1.1	1.1	1.1	1.1
V_2(p.u)	0.95	1.1	1.087	1.093	1.090	1.089	1.089	1.089	1.088	1.094	1.092	1.087	1.096	1.098
V_5(p.u)	0.95	1.1	1.058	1.070	1.067	1.062	1.072	1.066	1.056	1.079	1.069	1.057	1.082	1.079
V_8(p.u)	0.95	1.1	1.071	1.083	1.077	1.072	1.083	1.077	1.070	1.084	1.079	1.067	1.089	1.088
V_{11}(p.u)	0.95	1.1	1.098	1.079	1.098	1.097	1.074	1.092	1.049	1.1	1.093	1.098	1.097	1.099
V_{13}(p.u)	0.95	1.1	1.099	1.1	1.099	1.097	1.099	1.098	1.027	1.099	1.099	1.095	1.1	1.086
P_2(MW)	20	80	48.570	58.726	50.942	48.020	54.790	51.557	47.240	62.510	52.004	49.121	52.609	49.508
P_5(MW)	15	50	21.211	46.597	30.967	21.146	48.980	29.85	21.267	48.161	30.995	21.397	50	34.442
P_8(MW)	10	35	20.935	34.972	34.997	20.5405	34.9902	34.9878	26.770	34.893	34.994	20.695	35	35
P_{11}(MW)	10	30	12.227	29.982	26.911	12.502	29.9949	27.7995	13.270	29.998	26.579	11.789	30	27.359
P_{13}(MW)	12	40	12.123	38.410	21.173	12.003	35.447	20.323	12.498	36.723	20.998	12.013	39.882	23.657
$T_{6\text{-}9}$	0.9	1.1	1.011	1.041	1.016	1.0123	0.9827	0.9851	1.018	1.019	1.026	1.070	0.995	1.008
$T_{6\text{-}10}$	0.9	1.1	0.918	0.911	0.9111	0.905	0.924	0.947	1.030	0.908	0.900	0.901	0.954	0.929
$T_{4\text{-}12}$	0.9	1.1	0.978	0.967	0.978	0.986	0.964	0.970	0.960	0.974	0.978	1.019	1.008	1.016
$T_{28\text{-}27}$	0.9	1.1	0.959	0.958	0.966	0.968	0.962	0.960	1.011	0.963	0.968	0.974	0.962	0.953
Qc_{10}(Mvar)	0	5	3.236	4.998	3.806	0.382	4.945	4.461	3.791	1.114	1.292	4.240	0.0004	0.154
Qc_{12}(Mvar)	0	5	3.901	3.071	4.757	3.618	4.889	3.662	2.432	0.661	2.889	4.247	0.995	1.671
Qc_{15}(Mvar)	0	5	4.276	3.797	4.420	4.155	4.960	4.168	3.603	4.712	3.302	2.028	2.798	2.872
Qc_{17}(Mvar)	0	5	4.155	2.745	3.746	4.531	4.791	4.972	3.565	3.393	4.937	1.925	1.605	1.327
Qc_{20}(Mvar)	0	5	2.976	2.633	3.995	3.743	1.858	4.084	1.916	3.351	2.104	3.210	1.807	1.053
Qc_{21}(Mvar)	0	5	4.673	4.557	4.762	3.104	1.896	4.73	2.343	4.122	3.585	1.549	4.823	3.977
Qc_{23}(Mvar)	0	5	0.024	1.220	1.507	3.302	3.300	1.874	2.767	2.391	3.681	4.143	2.590	2.897
Qc_{24}(Mvar)	0	5	3.916	4.999	4.998	4.287	4.390	4.961	2.669	4.557	4.873	3.599	3.752	3.485
Qc_{29}(Mvar)	0	5	1.250	0.154	1.851	0.177	1.737	0.837	2.917	0.968	1.591	0.436	1.987	1.571
OBJECTIVE FUNCTIONS' VALUES														
Fuel Cost ($/h)			**799.179**	912.854	**827.636**	**799.202**	912.241	**826.446**	**800.756**	920.298	**827.717**	**799.319**	924.509	**837.416**
Active power losses (MW)			8.6446	**3.3714**	**5.2451**	8.6933	**3.3869**	**5.3074**	8.4171	**3.2468**	**5.2556**	8.7103	**3.2745**	**5.0397**

Table 2 summarizes the optimal values of the decision variables corresponding to the best *fuel cost* and L_{index}, (corresponding to the extreme points of the corresponding Pareto front), as well as the best compromise solution evaluated by the fuzzy membership approach described in (32)-(33). In this table, the best values are displayed in **boldface**.

It is worth mentioning that the *fuel cost* and L_{index} of the base case (before optimization) are 901.975$/h and 0.1766, respectively. Therefore, according to Table 2, the *fuel cost* reduction corresponding to the extreme point obtained by MOABC/D, MOTLA/D, MOEA/D-DRA and NSGA-II is 11.41%, 11.41%, 11.41%, and 11.40%, respectively. The improvement in L_{index} attained by MOABC/D, MOTLA/D, MOEA/D-DRA and NSGA-II is 27.18%, 27.34%, 27.29%, and 27.57%, respectively. Although these results are not significantly different, it is evident from Fig. 2, that our proposed algorithms have better convergence than the other MOEAs with respect to which it was compared.

B. Case study 2. Fuel cost and active power losses

In this case, the minimization of the *fuel cost* and *active power losses* are approached. These two competing objective functions are optimized simultaneously with the proposed methods, also with MOEA/D-DRA and the NSGA-II algorithm.

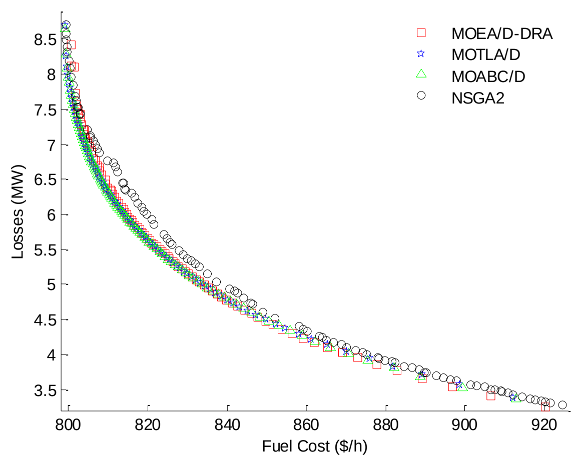

Figure. 3. Pareto optimal solution for case study 2.

The Pareto optimal set of the solutions is shown in Fig. 3. It is noticed from this figure that the proposed algorithms have similar performance than that of the MOEA/D-DRA and better convergence than the NSGA-II. Also, it is worth mentioning that similarly to case study 1, it is clear that the methods based on decomposition exhibit better convergence than the algorithm based on Pareto ranking, NSGA-II.

The optimal values of decision variables of the best *fuel cost* and *active power losses*, and the best compromise solution obtained by the algorithms are given in Table 3. The best values are displayed in **boldface.**

The base case has a fuel cost and active power losses of 901.975$/h and 5.831 MW, respectively. Therefore, comparing with the base case, the *fuel cost* reduction obtained by MOABC/D, MOTLA/D, MOEA/D-DRA and NSGA-II is 11.40%, 11.39%, 11.22%, and 11.38%, respectively. The *active power losses* reduction obtained by MOABC/D, MOTLA/D, MOEA/D-DRA and NSGA-II is 42.18%, 41.92%, 44.32%, and 43.84%, respectively. Although these results are slightly different from each other, it can be observed from Fig. 3, that our proposed algorithms exhibit better convergence.

C. *Comparison with other optimization strategies*

In order to show the effectiveness and performance of the proposed methods, the results of the proposed algorithms, MOABC/D and MOTLA/D, are compared with those obtained by the Differential Evolution (DE) [7], and Multi-objective harmony search (MOHS) [11] algorithms, which are based on Pareto ranking. Table 4 and 5 summarize the best compromise solution for the case study 1 and 2, respectively.

From Table 4, notice that the proposed methods based on decomposition achieve better results with respect to DE and MOHS in the *fuel cost* objective function. Regarding the L_{index} objective function, the proposed methods reach slightly higher value than DE and MOHS.

According to Table 5, the proposed methods achieve better results in both objective functions with respect to DE and MOHS.

TABLE 4. Comparison of the best compromise solution for case study 1.

Objective Functions	MOABC/D	MOTLA/D	DE[7]	MOHS[11]
Fuel cost ($/h)	**799.064**	799.116	800.59	799.9401
L_{index}	0.1288	0.1287	0.1249	**0.1075**

TABLE 5. Comparison of the best compromise solution for case study 2.

Objective Functions	MOABC/D	MOTLA/D	DE[7]	MOHS[11]
Fuel cost ($/h)	827.636	**826.446**	828.59	832.6709
Active power losses (MW)	**5.2451**	5.3074	5.69	5.3143

D. *Performance measure*

In order to assess the algorithms' performance, the coverage of two sets measure is adopted. This performance measure was proposed by Zitzler et al. [25]. This performance measure compares two sets of non-dominated solutions (A, B) and calculates the percentage of individuals in one set dominated by the individuals on the other set. It is defined as:

$$C(A,B) = \frac{\left|\{b \in B \middle| \exists\ a \in A : a \preceq B\}\right|}{|B|} \tag{36}$$

The value $C(A,B)$ = 1 means that all points in B are dominated by or equal to all points in A. $C(A,B)=0$ represents the condition when none of the solutions in B are covered by the set A. Note that both $C(A,B)$ and $C(B,A)$ have to be considered, since $C(A,B)$ is not necessarily equal to $1-C(B,A)$. When $C(A,B)$ = 1 and $C(B,A)=0$ then, we say that the solutions in A completely dominate the solutions in B (i.e., this is the best possible performance for A).

For each case study, twenty independent runs were made, and the results of the performance measure (36) are summarized in Tables 6 and 7. These tables present the average and standard deviation (in brackets) for each case study. The best results are displayed in **boldface**. In addition, the average CPU time is also shown.

TABLE 6. Results of coverage of two sets measure for case study 1.

Algorithm		C(A,B)	C(B,A)
A	B	average (std)	average (std)
MOABC/D	MOEA/D-DRA	**0.563** (0.2518)	0.251 (0.1844)
CPU time: 108.8 s	NSGA-II	**0.915** (0.1804)	0 (0)
MOTLA/D	MOEA/D-DRA	**0.502** (0.2074)	0.31 (0.1934)
CPU time: 107.3 s	NSGA-II	**0.955** (0.1012)	0 (0)

TABLE 7. Results of coverage of two sets measure for case study 2.

Algorithm		C(A,B)	C(B,A)
A	B	average (std)	average (std)
MOABC/D	MOEA/D-DRA	**0.408** (0.1726)	0.203 (0.1928)
CPU time: 62.4 s	NSGA-II	**0.868** (0.0998)	0 (0)
MOTLA/D	MOEA/D-DRA	**0.30** (0.1773)	0.26 (0.1860)
CPU time: 60.4 s	NSGA-II	**0.858** (0.1465)	0 (0)

Note from Tables 6 and 7 that the proposed approaches outperformed MOEA/D-DRA and NSGA-II in all cases regarding the Coverage of two sets performance measure. This indicates that the proposed approaches produce more solutions that dominate (according to Pareto optimality) the solutions produced by MOEA/D-DRA and NSGA-II. Table 6 indicates that in case study 1, MOABC/D and MOTLA/D generate solutions that dominate 56% and 50% of the solutions generated by MOEA/D-DRA, respectively. Likewise, MOABC/D and MOTLA/D produce solutions that dominate 91% and 95% of the solutions generated by NSGA-II, respectively. Table 7 shows that in case study 2, MOABC/D and MOTLA/D produce solutions that dominate 40% and 30% of those solutions generated by MOEA/D-DRA, respectively. Further, MOABC/D and MOTLA/D produce solutions that dominate 86% and 85% of the solutions generated by NSGA-II, respectively.

Moreover, in order to verify the effectiveness of the proposed approaches with respect to a deterministic algorithm, the same objective functions are individually minimized by the sequential quadratic programming (SQP) method from the MATLAB Optimization Toolbox. Table 8 summarizes the corresponding statistical analysis.

TABLE 8. Statistical analysis for SQD method.

Algorithm	SQD		
objective function	Cost ($/h)	L_{index}	Loss (MW)
Best	799.0436	0.1279	2.8778
Worst	831.6104	0.1648	7.7541
Average	808.2715	0.1445	5.1938
Standard Deviation	9.9928	0.0131	1.9788
average CPU time (s)	2.54	2.56	3.99

As seen in Table 8, SQD attains the best L_{index} and *active power losses* objective functions, and the average CPU time of SQD is less than that of the proposed algorithms. However, even though MOABC/D and MOTLA/D consume more execution time, their standard deviations and average values are better and quite satisfactory in comparison to those obtained by the SQP method.

Based on the above results and comparisons, it can be concluded that MOTLA/D and MOTLA/D are reliable for the multi-objective optimal power flow problems approached in this paper. This is evidenced by the fact that the Pareto optimal set obtained by them have exhibited satisfactory diversity and convergence for each of the case studies considered. Also, the proposed algorithms produced highly competitive results with respect to those obtained by the other approaches with respect to which they were compared.

VI. Conclusions

This paper presented two multi-objective optimization methods based on decomposition for solving a multi-objective optimal power flow problem. The first method is based on intelligent behavior of honey bees and the second one is based on the teaching-learning strategy.

In order to validate the effectiveness and performance of the proposed methods, MOABC/D and MOTLA/D were applied to the IEEE 30 bus test system and compare with respect to MOEA/D-DRA and the NSGA-II in two different multi-objective optimal power flow problems. Additionally, results reported in the open research were considered for a comparative study.

The results indicated that the proposed methods outperformed and are highly competitive with respect to the algorithms in comparison to the cases that were analyzed. Moreover, the proposed algorithms based on decomposition have better convergence that the traditional multi-objective evolutionary technique based on Pareto ranking, NSGA-II. Thus, it may be concluded that the proposed algorithms may be a quite promising and reliable choice for power systems applications.

References

[1] W. Stadler. “Multicriteria Optimization in Engineering and the Sciences”. New York: Plenum Press, 1st Edition, 1988.

[2] K.Y. Lee, Y.M. Park, and J. L. Ortiz, “A United *Approach to Optimal Real and Reactive Power Dispatch*”. IEEE Trans. On Power Apparatus and Systems, vol. PAS-104, no. 5, pp.1147-1153, May, 1985.

[3] J. A. Momoh, M. E. El-Hawary and R. Adapa. “A review of selected optimal power flow literature to 1993. I. Nonlinear and quadratic programming approaches”. IEEE Trans. On Power System, vol. 14, no. 1, pp: 96-104, 1999.

[4] J. A. Momoh, M. E. El-Hawary and R. Adapa. “A review of selected optimal power flow literature to 1993. II. Newton, linear programming and interior point methods”. IEEE Trans. Power System, vol. 14, no. 1, pp: 105-111, 1999.

[5] C. C. Kuo, P. H. Chen, C. L. Hsu, and Y. T. Chao, “The reactive power and voltage control of distribution systems using the normalized weighting method”, in Transmission and Distribution Conference and Exhibition: Asia and Pacific, pp. 1-6, August, 2005.

[6] Y. T. Hsiao, H. D. Chiang, C. C. Liu, and Y. L. Chen, “A Computer Package for Optimal Multiobjective VAR Planning in Large Scale Power Systems”, IEEE Trans. Power Systems, vol. 9, no.2, pp. 668-676, May 1994.

[7] M. A. Abido and N. A. Al-Ali. “Multi-objective Optimal Power Flow Using Differential Evolution”, Arabian Journal for Science and Engineering, vol. 37, no.4, pp. 991-1005, June 2012.

[8] S. Jeyadevi, S. Baskar, C.K. Babulal and M. Willjuice Iruthayarajan. “Solving multiobjective optimal reactive power dispatch using modified NSGA-II”. International Journal of Electrical Power and Energy Systems, vol. 33, no. 2, pp. 219-228, February 2011.

[9] K. Deb, S. Agrawal, A. Pratab, and T. Merayivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182-197, April 2002.

[10] M. A. Abido, “Multiobjective Particle Swarm Optimization for Optimal Power Flow Problem”, Handbook of Swarm Intelligence, vol. 8, pp. 241-268, Springer: Berlin, 2011.

[11] S. Sivasubramani, k. S. Swarup, “Multi-objective harmony search algorithm for optimal power flow problem”, International Journal of Electrical Power and Energy systems, vol. 33, pp. 745-752, 2011.

[12] K. O. Alawode, A. M Jubril and O. A. Komolafe, “Multiobjective Optimal Power Flow Using Hybrid Evolutionary Algorithm” International Journal of Electrical and Electronics Engineering, vol. 4, no. 7, pp.506-511, 2010.

[13] V. Yamille, K. V. Ganesh, M. Salman, H. Jean-Carlos and G. H. Ronald, “Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems”, IEEE Trans. On Evolutionary Computation, vol. 12, no. 2, April 2008.

[14] Q. Zhang, and H. Li, “MOEA/D: A multiobjective evolutionary algorithm based on decomposition,” IEEE Transaction on Evolutionary Computation, vol. 11, no. 6, pp. 712-731, December 2007.

[15] H. Li, and Q. Zhang, “Multiobjective optimization problems with complicate Pareto sets, MOEA/D and NSGA-II,” IEEE Transactions on Evolutionary Computation, vol. 13, no.2, pp. 284-302, April 2009.

[16] P. Kessel, and H. Glavitsch, "Estimating the Voltage Stability of a Power System," IEEE Transactions on Power Delivery, vol.1, no.3, pp.346-354, July 1986.

[17] Q. Zhang, W. Liu, and H. Li, “The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances,” IEEE congress on Evolutionary Computation, Trondheim Norway, May 2009.

[18] K. Miettinen. “Nonlinear Multiobjective Optimization”. Kluwer Academic Publishers, Boston. Massachuisetts, 1999.

[19] D. Karaboga, “An idea based on a honey bee swarm for numerical optimization,” Erciyes University, Engineering Faculty, Computer Engineering Department, Technical-Report TR06, October 2005.

[20] B. Akay, and D. Karaboga, “A modified artificial bee colony for Real-parameters optimization,” Information Sciences, vol. 192, no. 1, pp. 120-142, June 2012.

[21] R. V. Rao, V. J. Savsani and D. P. Vakharia, “Teaching-learning based optimization: A novel method for constrained mechanical design optimization problems”, Computer-Aided Design, vol. 43, no. 3, pp. 303-315, Mar. 2011.

[22] M.V. Suganyadevia, and C.K. Babulal, "Estimating of loadability margin of a power system by comparing Voltage Stability Indices", in International Conference on Control, Automation, Communication and Energy Conservation, 2009. INCACEC 2009, pp.1-4, June 4-6, 2009.

[23] P. Kessel, and H. Glavitsch, "Estimating the Voltage Stability of a Power System", IEEE Transactions on Power Delivery, vol.1, no.3, pp.346-354, July 1986.

[24] M. Sakawa, H. Yano, and T. Yumine, “An interactive fuzzy satisficing method for multiobjective linear programming problems and its application”, IEEE Trans. Syst. Man, Cybern, vol. SMC-17, no.4, pp.654-661, 1987.

[25] E. Zitzler, K. Deb and L. Thiele. “Comparison of Multiobjective evolutionary algorithm: empirical results”. Evolutionary Computation, vol. 8, no. 2, pp. 173-195, 2000.