Chapter 18

A multi-objective teaching-learning algorithm for power losses reduction in power systems

Juan M. Ramirez, Miguel A. Medina, Carlos A. Coello Coello

Abstract The reactive power dispatch is one of the most complex problems of power systems and may include the simultaneous optimization of several objective functions, possibly in conflict among them. Hence, the optimal reactive power dispatch problem becomes a multi-objective optimization problem. Such kind of optimization problem has a set of possible solutions, which represent the best commitment among the objective functions.

Novel methods based on meta-heuristics have become a popular choice for solving complex real world multi-objective optimization problems due to their flexibility, generality and ease of use. The advantages of evolutionary algorithms in terms of the modeling capability and excellent global search characteristics have encouraged their application to the reactive power dispatch problem in power systems.

The Teaching Learning-Based Optimization (TLBO) is a population-based optimization algorithm suitable for solving complex problems. TLBO imitates the interaction between a teacher and her/his students. The global solution search process of this approach consists of two phases: the Teacher- and the Learner-Phase. This chapter proposes a multi-objective teaching learning algorithm based on decomposition (MOTLA/D) for solving a reactive power handling problem.

In order to assess the effectiveness of the proposed approaches to solve the multi objective reactive power dispatch problem, the algorithms are tested in three power systems of different complexity: IEEE 14 bus, 30 bus and 118 bus system. Several studies have been carried out among the algorithms involving fuel cost minimization, power losses reduction and voltage stability enhancement as objective functions. Furthermore, the proposed method is applied to a simplified Mexican power grid.

Juan M. Ramirez

Centro de Investigacion y de Estudios Avanzados del Instituto Politecnico Nacional
Av. del Bosque 1145, Col. el Bajio, Zapopan Jal., 45019, Mexico.
Email: jramirez@gdl.cinvestav.mx

Miguel A. Medina

Centro Nacional de Control de Energia. Area de Control Peninsular
Calle 10 #312a Gonzalo Guerrero
Email: mmedina@gdl.cinvestav.mx

Carlos A. Coello Coello

Centro de Investigacion y de Estudios Avanzados del Instituto Politecnico Nacional
Department of Computer Science Mexico
Av. Instituto Politécnico Nacional 2508, Col. San Pedro Zacatenco, Delegación Gustavo A. Madero, México D.F., 07360.
Email: ccoello@cs.cinvestav.mx

Keywords: Multi-objective optimization, optimal power flow, power losses, reactive power management, teaching-learning algorithm.

Nomenclature

ABC	Artificial Bee Colony Algorithm
DE	Differential Evolution
GA	Genetic Algorithm
HS	Harmony Search
MOEA/D	Multi-Objective Evolutionary Algorithm based on Decomposition
MOP	Multi Objective Problem
MOTLA/D	Multi-Objective Teaching Learning Algorithm based on Decomposition
NSGA-II	Non-dominated Sorting Genetic Algorithm II
OPF	Optimal Power Flow
ORPD	Optimal Reactive Power Dispatch
PSO	Particle Swarm Optimization
TLBO	Teaching Learning-Based Optimization
Ω	feasible region

18.1 Introduction

Over the past half-century, optimal power flow (OPF) has become one of the most important and widely studied problems in power system operation, analysis and planning. The OPF was proposed in the early 1960s based on the economic dispatch problem. In general, the OPF problem is a large-scale, highly constrained, non-linear, non-convex optimization problem which may contain both continuous and discrete control variables. The main purpose of the optimal power flow is to optimize a selected objective function, such as cost, planning, or reliability by controlling power flow within an electrical network without violating network power flow constraints or system and equipment operating limits. Many different classes of the OPF problem have been developed to address specific instances of the problem, using varying assumptions and selecting different objective functions to be optimized, different sets of control variables and different system constraints. The resulting optimization problems have different names depending on the particular objective function being addressed and the constraints under consideration. The optimal reactive power dispatch (ORPD) is a special case of the optimal power flow problem and is one of the major issues in terms of secure and economic operation. Generally, the control variables of the ORPD consist of voltage magnitudes of generator buses, output of static reactive power compensators, transformer tap-settings, and shunt capacitors/reactors. The aim of the reactive power dispatch problem is to minimize real power transmission losses of the network, while maintaining the system voltage profile in an acceptable range by regulating generator bus voltages, transformer tap-settings and reactive power generation of reactive power sources. Furthermore, physical and operation constraints must be maintained within the allowable limits. Since output shunt capacitors/reactors and tap-settings of transformers are discrete variables, but some other variables are continuous, the reactive power dispatch problem can be modelled as a large-scale, mixed-integer, non-differentiable and non-linear problem. Additionally, the difficulty of solving the ORPD problem significantly increases with the increment of network's size and complexity.

Thus, optimal reactive power (ORP) plays a significant role in the secure operation of power systems. One of the main tasks of a power system operator is to manage the system in such a way that its operation is safe and reliable. Its main aim is to determine the optimal operating capacity and the physical distribution of the compensation devices such as voltage rating of generators, reactive power injection of shunt capacitors/reactors, and tap ratios of the tap setting transformers, in order to ensure a satisfactory voltage profile, while minimizing the transmission losses. Active power line losses are small while reactive power line losses are large. Reducing the reactive power losses enables more active power to be transferred over a single line. Due to the continuous growth in the demand for electricity with unmatched generation and transmission capacity expansion, voltage stability has emerged as a challenge to power system planning and operation. Therefore, a voltage stability index should also be considered as an objective of the ORP problem.

It is important that each system and control area handles capacitive and inductive reactive resources at proper levels to maintain the voltages within established high and low limits. Reactive generation scheduling, transmission and switching, and load shedding, if necessary, should be implemented to maintain these levels. Likewise, each control area should provide its reactive power requirements, including appropriate reserves to protect the voltage levels for contingency conditions.

The optimal reactive power problem is a nonlinear, non-convex, over-determined system, a large-scale optimization problem with both continuous and discrete variables; additionally, its high dimensionality represents a major difficulty. This problem is quite important for power system security. In this chapter, the basic objective is to estimate proper adjustments on the control variables, such as generator bus voltages and tap setting transformers that help to maintain an acceptable voltage profile and minimize the reactive power losses; one voltage stability metric (L_{index}) is also used. Thus, an optimal formulation that contributes to attain these purposes becomes appropriate. In general, it may include several objective functions, possibly in conflict among them.

Such kind of optimization problem has a set of possible solutions (named Pareto optimal set), which represents the best commitment (feasible) among the objectives [1]. Several optimization techniques have been proposed to solve such optimal reactive power problems. From them, two major approaches may be identified [2]:

(1) The first approach is based on the use of evolutionary algorithms such as Differential Evolution (DE) [3], Non-dominated Sorting Genetic Algorithm II (NSGA-II) [4], Particle Swarm Optimization (PSO) [5], an Improved Hybrid Evolutionary Programming Technique [6], and Artificial Bee Colony Algorithm (ABC) [7].

(2)The second approach is based on conventional methods. They include Gradient-based Methods, Non-Linear Programming (NLP), Quadratic Programming (QP), Linear Programming (LP) and Interior Point Methods [8-12], the Weighting Method [13], and the ε-Constraint Method [14].

These conventional methods are based on an estimation of the global minimum. However, due to difficulties of differentiability, non-linearity, and non-convexity, these methods do not guarantee reaching the global optimum [15]. Thus, these methods present limitations when dealing with certain types of problems. For instance, they cannot be used when the objective function is not available in an algebraic form. This has motivated the development of alternative methods, such as meta-heuristics. Over the years, meta-heuristics (from which evolutionary algorithms is a particular subclass) have become a popular choice for solving complex optimization problems, due to their flexibility, generality (they are less sensitive to the actual shape or continuity of the Pareto front than conventional methods) and ease of use. Additionally, most meta-heuristics require little or no specific domain knowledge.

Multi-objective optimization is a design methodology that optimize a set of objective functions systematically and simultaneously. Such kind of optimization problem has a set of possible solutions (named Pareto Optimal Set), which represent the best commitment (or trade-off) among the objectives. In the open research, several methods have been suggested and successfully applied for solving multi-objective power systems optimization problems. These approaches are generally divided into two categories: classical mathematical optimization algorithms and Pareto-based optimization algorithms.

The classical methods transform the multi objective optimization problem into a single objective optimization problem usually by either aggregating the objective functions into a single weighted function, or simply by optimizing one objective function and treating the remaining as constraints. Therefore, the resulting single optimization problem can be solved by deterministic or non-deterministic (heuristic) algorithms. However, these classical single-objective approaches have several limitations to solve multi-objective optimization problems: 1) it requires a priori knowledge about the relative importance of the objective functions; 2) the aggregated function leads to only one solution, and therefore, it requires multiple simulation runs to obtain the Pareto-optimal Set; 3) the trade-offs among objective functions cannot be easily evaluated, and 4) the solution may not be attainable unless the Optimal Pareto Set is convex. Thus, classical optimization methods are not suitable to solve multi-objective optimization problems. Furthermore, due to the fact that real life problems involve several objectives and system engineers may desire to know all possible optimization solutions of all objectives simultaneously, Pareto-based approaches have been proposed as an alternative to address the shortcomings of classical single- objective methods. These algorithms handle the objective functions simultaneously as competing objectives.

In this chapter, a multi-objective teaching learning algorithm based on decomposition (MOTLA/D) is proposed for solving the reactive power handling problem. In order to minimize the reactive power losses and a voltage stability index [16], the proposed algorithm estimates the following optimal values: (*i*) generator bus voltages; (*ii*) tap setting transformers. The effectiveness of the proposed approach is demonstrated and compared with respect to the multi-objective evolutionary algorithm based on decomposition (MOEA/D) [17], which is representative of the state-of-the-art on the subject; both methods are applied on three test systems: 9-, 26-, and 118-buses.

The rest of the chapter is organized as follows. Section 18.2 exposes a brief review of previous work. Section 18.3 presents the problem approached in this chapter. Section 18.4 describes the formulation. Section 18.5 summarizes the decomposition strategy. In Sections 18.6, 18.7 and 18.8, the general framework of the proposed approach is summarized. Results of a comparative study are presented in Section 18.9. Finally, conclusions are provided.

18.2 A brief review of previous work

Several review articles about evolutionary algorithms in power systems applications have been published since the 1990s: [18-21]. This section reviews the technical literature related to applications of multi-objective evolutionary algorithms to voltage stability enhancement, reactive power dispatch and economic dispatch optimization problems in power systems, which are the topics dealt with in this chapter. The review in this section has been drawn from refereed journal articles from the following journals:

- IEEE Transactions on Power Systems.
- International Journal of Electrical Power & Energy Systems.
- Electric Power Systems Research.
- Energy Conversion and Management.
- Engineering Applications of Artificial Intelligence.

These journals have published a large quantity of heuristic algorithms in the context of power system applications. Books and conference proceedings have not been included.

Novel modern heuristics algorithms have been reported in Power systems applications. Chaohua Dai, et al. [22] proposed a seeker optimization algorithm (SOA) for the optimal reactive power dispatch considering three objective functions: active power losses, voltage deviation, and voltage stability index. SOA is a novel population-based heuristic search algorithm, which attempts to simulate the act of human searching for real-parameter optimization. The algorithm's performance is studied on the IEEE 57 and 118-bus power systems.

Liao, H.L., et al. [23] presented a new method called Multi-objective Optimization by Reinforcement Learning (MORL) to solve the optimal power system dispatch and voltage stability problem by achieving two objectives: reduction of fuel cost and enhancement of voltage stability. Reinforcement Learning stems from the idea of trial and-error learning, and focuses on goal-directed learning from interactions between learner and environment.

Panigrahi B.K., et al. [24] extend the bacteria foraging meta-heuristic into the domain of multi-objective optimization to solve the multi-objective environmental/economic dispatch (EED) problem. The idea of bacteria foraging principle is based on the fact that natural selection tends to eliminate animals with poor foraging strategies through methods for locating, handling, and ingesting food, and to favor the propagation of genes of those animals that have successful foraging strategies. In this work the multi-objective bacteria foraging (MOBF) optimization technique was compared respect to other multi-objective evolutionary algorithms on the standard IEEE 30-bus test system.

Sivasubramani S. and Swarup K.S. [25] model the optimal power flow problem as a non-linear constrained multi-objective optimization problem where power losses, fuel cost and voltage stability objectives are considered into the formulation. The problem is solved by a multi-objective harmony search (MOHS) algorithm. The Harmony search algorithm has been recently developed in an analogy with improvisation process where musicians always try to polish their pitches to obtain a better harmony.

Niknam T., et al. [26] present a multi-objective teaching-learning-based optimization algorithm (θ-MTLBO) to solve the dynamic economic/emission dispatch problem considering ramp-rate limits and valve-point effects. The Teaching-Learning Based Optimization algorithm is a new efficient optimization algorithm which has been inspired by learning mechanism in a class. The proposed algorithm is applied on three test systems and compared with respect to several heuristic optimization techniques and with a popular optimization software named general algebraic modeling system (GAMS), which uses linear programming to optimize the problems. In [30] a multi-objective optimization algorithm based on modified teaching-learning-based optimization (MTLBO) algorithm is proposed in order to solve the optimal location of automatic voltage regulators (AVRs) in distribution systems in presence of distributed generators (DGs).

Khorsandi A., et al. [27] propose a fuzzy based modified artificial bee colony (MABC) algorithm to solve the optimal power flow problem. The problem was formulated as a multi-objective mixed-integer nonlinear problem where the simultaneous minimization of the total fuel cost of thermal units, total emission, total real power losses, and voltage deviation were considered. The artificial bee colony algorithm is a relatively new optimization technique which simulates the intelligent foraging behavior of a bee swarm. The proposed approach is applied on the IEEE 30-bus and IEEE 118-bus test systems.

More new modern meta-heuristic algorithms for multi-objective optimization can be found in the literature. This dissertation proposes the multi-objective variant for two popular meta-heuristics of current interest, namely, the artificial bee colony algorithm (ABC), and the teaching-learning-based optimization algorithm (TLBO). These algorithms are the two with the highest efficiencies in their class. The effectiveness of these algorithm have been tested in several benchmark functions and they have been compared with those of state-of-the-art algorithms such as: Differential Evolution (DE), Particle Swarm Optimization (PSO) and their variants [28] and [29]. Moreover, the performance and effectiveness of these algorithms to solve large scale optimization problems and real world optimization problems, including problems in power systems, have been demonstrated [26]-[27], [30]-[32].

There are multi-objective variants of the TLBO and ABC in the literature such as those presented in [26]-[27], [30], [33]. These variants are Pareto-based approach. Thus, these methods require some other techniques for ranking and distribution of the solutions (e.g., crowding distance, fitness sharing, niching). However, as above mentioned, these methods cannot always provide good results, especially when dealing with complex multi-objective optimization problems. Therefore, in this dissertation the proposed multi-objective variants of TLBO and ABC algorithm are based on the MOEA/D framework. This framework decomposes a multi-objective problem into several single-objective optimization sub-problems with neighborhood relations. In this way, a set of approximate solutions to the Pareto front is achieved by minimizing each sub-problem instead of using Pareto ranking.

The performance of the multi-objective evolutionary algorithms based on decomposition in power system problems, has not been fully investigated. A publication related with the application of MOEA/D in reactive power dispatch problem is presented on Innovative Smart Grid Technologies Conference Europe, 2010 *IEEE PES* [23]. In this work, Liao, H. L et al, compared a novel multi-objective optimization by reinforcement learning with the first version of the MOEA/D to solve the economic power dispatch with voltage stability enhancement.

This chapter proposes a multi-objective Teaching-Learning algorithm for managing reactive power. The proposition is applied on several power test systems with different dimensionality.

18.3 Multi-Objective Power Flow Formulation

Over the past half-century, the optimal power flow (OPF) has become one of the most important and widely studied problems in power system operation, analysis and planning. In general, the OPF problem is a large-scale, highly constrained, non-linear, non-convex optimization problem which may contain both continuous and discrete control variables. The main purpose of the OPF is to optimize a selected objective function, cost for instance, with planning or reliability purposes by controlling power flow within an electrical network without violating network constraints or system and equipment operating limits [34]-[35].

Many different OPF classes have been developed to address specific instances of the problem, among them the reactive power dispatch and economic dispatch problems have always been the major issues in terms of secure and economic power systems operation. These problems can be considered into the analysis of multi objective optimization. The objective functions in these problems have many variants, which include transmission network losses, transmission capacity, and investment of compensation devices. In addition, some technical indices such as voltage stability indices to prevent security margins, may be used as objectives. This research takes the voltage stability into account within the short-term operation planning context, where an optimal preventive action has to be found to enhance voltage stability including operational limits.

In this chapter the minimization of the total power transmission losses has been selected as an objective function in the optimal reactive power dispatch and economic dispatch problems. 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.

18.3.1 Problem Objectives

The multi-objective optimal power flow addressed in this chapter uses the following objective functions: (*i*) minimization of the total fuel cost; (*ii*) minimization of the total power transmission losses; and (*iii*) voltage stability enhancement through the minimization of the L-index.

18.3.1.1 Fuel cost minimization

Figure 18.1 depicts a conventional cost curve for the *i*-th generator with active power output P_{gi}.

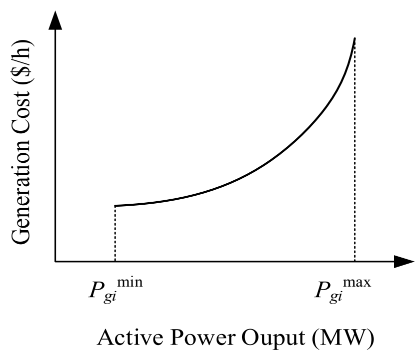

Fig. 18.1. Operating cost curve for the *i*-th generator.

The objective is to minimize the total fuel cost F_{cost}. Conventionally, the fuel cost for *i*-th generator with active power output P_{gi} is modelled as a quadratic function of the active power generation, and the total fuel cost F_{cost} in ($/h) may be expressed as,

$$F_{\text{cost}} = \sum_{i=1}^{N_g} a_i + b_i P_{gi} + c_i P_{gi}^2 \tag{18.1}$$

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.

18.3.1.2 Active power losses minimization

Transmission losses constitute economic loss providing no benefits. Therefore, the objective is to minimize the active power losses (P_{loss}) through the transmission lines. If we express these losses in terms of bus voltages and associated angles, then the losses may be calculated by,

$$P_{loss} = \sum_{k=1}^{nl} g_k [V_i^2 + V_j^2 - 2V_i V_j \cos(\theta_i - \theta_j)] \tag{18.2}$$

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.

18.3.1.3 Voltage stability enhancement

A conventional way looking for voltage stability assessment is the use of indices, which estimate the proximity to voltage instability and determine those buses exhibiting weak stability. Nowadays, there are a variety of indexes that help to assess the steady state voltage stability [36-38].

In this research, voltage stability enhancement is achieved through minimizing the voltage stability index L_{index} [39], 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 [39].

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

$$I_{bus} = Y_{bus} V_{bus} \tag{18.3}$$

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 (18.3) becomes,

$$\begin{bmatrix} I^L \\ I^G \end{bmatrix} = \begin{bmatrix} Y_1 & Y_2 \\ Y_3 & Y_4 \end{bmatrix} \cdot \begin{bmatrix} V^L \\ V^G \end{bmatrix} \tag{18.4}$$

It is assumed that the transmission system is linear and allows a representation in terms of a hybrid matrix H. Therefore rearranging the above equation,

$$\begin{bmatrix} V^L \\ I^G \end{bmatrix} = H \cdot \begin{bmatrix} I^L \\ V^G \end{bmatrix} = \begin{bmatrix} Z^{LL} & F^{LG} \\ K^{GL} & Y^{GG} \end{bmatrix} \cdot \begin{bmatrix} I^L \\ V^G \end{bmatrix} \tag{18.5}$$

where V^L and I^L are load voltage and current vectors; V^G and I^G are generator voltage and current vectors; 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 vector is exchanged for the current vector. This representation may then be utilized to define a voltage stability index for load buses, namely L_j which is defined by,

$$L_j = \left| 1 - \frac{\sum_{i \in \alpha_G} F_{ji} V_i}{V_j} \right| \tag{18.6}$$

For stable conditions, $0 \leq L_j \leq 1$, must not be violated for any load bus j. A global indicator L_{index} describing the whole system's stability is defined by,

$$L_{index} = \max_{j \in \alpha_L} (L_j) \tag{18.7}$$

L_{index} in Eq. (18.7) 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.

18.3.2 Problem Constraints

The minimization of the above functions must satisfy a number of equality and inequality constraints. The OPF equality constraints reflect the physics of the power system by imposing Kirchhoff's Laws to the electrical network, and the inequality constraints of the OPF reflect the limits on physical devices, as well as the limits created to ensure system security. These constraints are described in following.

18.3.2.1 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,

$$P_{gi} - P_{di} - \sum_{j=1}^{N_b} |V_i||V_j||Y_{ij}|\cos(\delta_i - \delta_j - \theta_{ij}) \tag{18.8}$$

$$Q_{gi} - Q_{di} - \sum_{j=1}^{N_b} |V_i||V_j||Y_{ij}|\sin(\delta_i - \delta_j - \theta_{ij}) \tag{18.9}$$

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.

18.3.2.2 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) by lower and upper limits as follows,

$$V_{gi}^{\min} \le V_{gi} \le V_{gi}^{\max}, \quad i = 1, \ldots, N_g \tag{18.10}$$

$$P_{gi}^{\min} \le P_{gi} \le P_{gi}^{\max}, \quad i = 1, \ldots, N_g \tag{18.11}$$

$$Q_{gi}^{\min} \le Q_{gi} \le Q_{gi}^{\max}, \quad i = 1, \ldots, N_g \tag{18.12}$$

where N_g is the number of generators.

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

$$T_i^{\min} \le T_i \le T_i^{\max}, \quad i = 1, \ldots, N_t \tag{18.13}$$

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:

$$Q_{ci}^{\min} \le Q_{ci} \le Q_{ci}^{\max}, \quad i = 1, \ldots, N_c \tag{18.14}$$

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

$$V_i^{\min} \le V_i \le V_i^{\max}, \quad i = 1, \ldots, N_{PQ} \tag{18.15}$$

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

18.4 Multi-objective optimal power flow formulation

Considering the above-mentioned objective functions, equality, *g(x,u)*, and inequality constraints, *h(x,u)*, the multi-objective optimal power flow is defined as the optimization of the following study case.

Minimization of fuel cost and power losses

$$\text{minimize } [F_{cost}, P_{loss}]$$

Subject to:

$$g(x,u) = 0$$

$$h(x,u) \le 0$$

where g is the equality constraints representing the non-linear power flow equations (18.8)-(18.9); h represents the inequality constraints through the system operating limits (18.10)-(18.15); and x is the vector of dependent variables including:

1. Active power P_{g1} generated by the slack bus
2. Load bus voltage V_L
3. Generator reactive power output Q_G

Hence x can be expressed as:

$$x = [P_{g1}, V_{L1}, ..., V_{LN_{PQ}}, Q_{G1}, ..., Q_{GN_g}] \quad (18.16)$$

where N_{PQ}, N_g, are the number of load buses and the number of generators, respectively.

Likewise, u is the vector of control or independent variables including:

1. Generator active power outputs P_g except the slack bus P_{g1}.

2. Generator bus voltage V_g.

3. Transformer tap setting T.

4. Shunt VAR compensation Q_C.

Hence, the vector of control variables (u) is expressed as,

$$u = [P_{g2}, ..., P_{gN_g}, V_{g1}, ..., V_{gN_g}, T_1, ..., T_{Nt}, Q_{c1}, ..., Q_{cN_c}] \quad (18.17)$$

where N_t and N_c denote the number of regulating transformers and number of shunt compensations, respectively. It is worth noting that the decision variables are self-constrained by the optimization algorithm.

18.4.1 Handling of equality and inequality constraints

The equality constraints of power balance equations shown in (18.8)-(18.9) are handled by the Newton-Raphson based power flow calculations; therefore, there is no need to integrate them into the objective function.

Due to the optimal power flow results may violate the inequality constraints, it is common to use penalization to handle the inequality violation. The violations are multiplied by its corresponding penalty coefficient and added to the objective function in order to constitute a fitness function. It is simple of implementing and easy of understanding. However, it introduces new parameters into the algorithm and the value of such coefficients significantly influences the solution reached by algorithm. There is not a specific rule to choose the suitable penalty coefficient values and the trial-error strategy is applied in most situations. The electricity quality offered by power system is vastly important and a large variation could result in a huge economic loss. For this consideration, a non-parameter strategy is adopted to handle the inequality constraints.

In this chapter Deb's heuristic constrained handling technique [40] is used in the proposed MOTLA/D and MOABC/D methods. This technique uses a tournament selection operator in which two solutions are selected and each other compared. Therefore, this approach does not require the use of the penalty function method.

The following heuristic rules are implemented for the selection:

i. If one solution is feasible and the other infeasible, then the feasible solution is preferred.

ii. If both are feasible solutions, then the solution with better objective function value is preferred.

iii. If both solutions are infeasible, then the solution with least constraint violation is preferred.

iv. If both solutions are infeasible and have the same number of constraint's violations, then the solution with a smaller objective function value is preferred.

These rules are implemented at the end of the main phases of the proposed algorithms, i.e., at the end of the teacher and the learner phases for MOTLA/D, and at the end of the employed and onlooker phases for the MOABC/D. Instead of accepting the new solution x_{new}, if it gives better function value at the end of these phases, Deb's constraint handling rules are used to select x_{new} based on the above heuristic rules.

18.5 Metaheuristic Algorithms based on Decomposition

"*A metaheuristic is formally defined as an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space, learning strategies are used to structure information in order to find efficiently near-optimal solutions* [41]". Some well-known algorithms falling within this category are: Genetic algorithms (GAs) [42], Differential Evolution (DE) [43], Particle Swarm Optimization (PSO) [44, 45], Artificial Bee Colony (ABC) [46, 47], and the Harmony Search (HS) [48].

Problems with multiple objectives arise in a natural manner in many disciplines and their solutions have been a challenge to researchers for a long time [49]. These problems are known as multi-objective optimization, multi-criteria optimization, or vector optimization problem and can be defined (in words) as the problem of finding [50]: "a vector of decision variables which satisfies constraints and optimizes a vector function whose elements represent the objective functions. These functions form a mathematical description of performance criteria which are usually in conflict with each other. Hence, the term "optimize" means finding such a solution which would give the values of all the objective functions acceptable to the decision maker."

The mathematical definition of a multi-objective problem (MOP) is important in providing a basis of understanding between the interdisciplinary nature of every possible solution technique (deterministic, non-deterministic), i.e., search algorithms [49]. Without loss of generality, a multi-objective problem may be stated as:

Finding a vector $\vec{x}^* = [x_1^*, x_2^*, ..., x_n^*]^T$ which optimizes the vector function:

$$\text{Minimize} \quad \boldsymbol{F}(\vec{x}) = \left[f_1(\vec{x}), f_2(\vec{x}), ..., f_k(\vec{x})\right]^T \tag{18.18}$$

Subject to the constraints:

$$g_i(\vec{x}) \le 0, \quad i = 1, ..., m \tag{18.19}$$

$$h_j(\vec{x}) = 0, \quad j = 1, ..., p \tag{18.20}$$

where a solution $\vec{x} \in \Re^n$ is a vector of n decision variables: $\vec{x} = (x_1, x_2, ..., x_n)^T$. Notice that T indicates *transposition*. The inequality and equality constraints in (18.19) and (18.20), respectively, define the feasible region Ω and any solution $\vec{x} \in \Omega$ defines a feasible solution. One of the notable differences between single-objective and multi-objective optimization is that in the later one the objective functions constitute a multi-dimensional space, in addition to the usual decision variable space ($\Re^n$). This additional *K*-dimensional space is called *objective space*, $\Re^k$. Therefore, for each solution $\vec{x}$ in the decision variable space, there exists a point $\vec{z}$ in the objective space, $\vec{z} \in \Re^k$. The vector function $\boldsymbol{F}(\vec{x})$ is a function which maps the set of solutions from the decision variable space to the objective space: $\boldsymbol{F} : \vec{x} \in \Re^n \rightarrow \Re^k$, Fig. 18.2. The *k*-elements of $\boldsymbol{F}(\vec{x})$ represents the values of the objective functions.

Vector $\vec{x}^*$ denotes the optimal solution. However, since the objective functions of a multi-objective optimization problem usually are in conflict with each other, and cannot be optimized simultaneously. Therefore, there no exists a unique optimal solution instead it has to be found a satisfactory trade-off between the conflicting objectives. This leads to a set of optimal solutions known as *Pareto optimal* solutions or *non-dominated* solutions. In multi-objective optimization, the goal is to find the best possible compromise (or trade-off) among the objectives since, frequently, one objective can be improved only at the expense of worsening another. Therefore, it is necessary to establish some criteria to define what would be considered as an “optimal” solution.

In order to describe the concept of “optimality” for a multi-objective problem, which was generalized by Vilfredo Pareto [51] on the concept “*Pareto optimum*”, the following definitions are provided [49]:

Definition 1 (Pareto Dominance): A solution $\vec{x} = (x_1, x_2, ..., x_n)$ is said to dominate another solution $\vec{y} = (y_1, y_2, ..., y_n)$ (denoted by $\vec{x} \preceq \vec{y}$) if and only if $\vec{x}$ is partially less than $\vec{y}$ on the objective space, i.e., $\forall i \in \{1,...,k\}$, $f_i(\vec{x}) \leq f_i(\vec{y}) \ \wedge \ \exists i \in \{1,...,k\} : f_i(\vec{x}) < f_i(\vec{y})$.

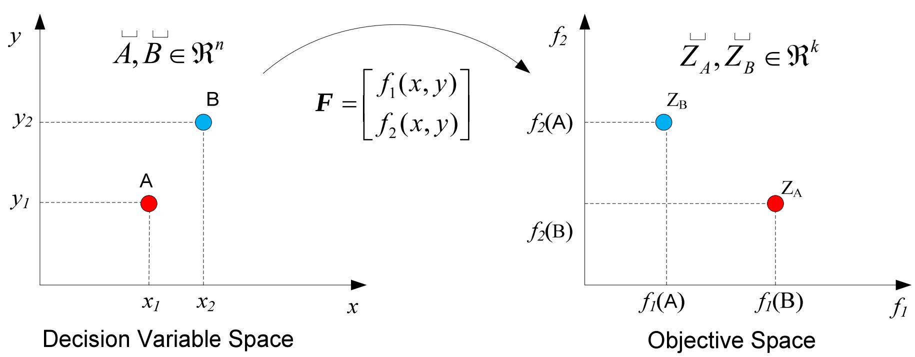

Fig. 18.2 Example of the multidimensional space in a multi-objective problem

Definition 2 (Pareto Optimality): A solution $\vec{x} \in \Omega$ is said to be a *Pareto optimal* solution $\vec{x}^*$, if and only if there is no other solution $\vec{y} \in \Omega$ such that $\vec{y} \preceq \vec{x}$.

Definition 3 (Pareto Optimal Set): For a given MOP, the Pareto Optimal Set ($\overline{PS}$) is defined by $\overline{PS} = \{\vec{x} \in \Omega \,|\, \vec{x}$ is Pareto Optimal Solution$\}$.

Definition 4 (Pareto Front): For a given MOP and a Pareto Optimal Set $\overline{PS}$, the Pareto Front $\overline{PF}$ is defined as the image of $\overline{PS}$ in the objective space, i.e., $\overline{PF} = \{F(\vec{x}) \,|\, \vec{x} \in \overline{PS}\}$.

Based on the concept of “*Pareto Dominance*”, the comparison between two decision vectors $\vec{x}$ and $\vec{y}$ can lead to the following three possibilities:

1) ($\vec{x}$ **strictly dominates** $\vec{y}$): $\vec{x} \prec \vec{y}$ *iff* $f_i(\vec{x}) < f_i(\vec{y})$ for $i = 1,..,k$

2) ($\vec{x}$ **weakly dominates** $\vec{y}$): $\vec{x} \preceq \vec{y}$ *iff* $f_i(\vec{x}) \le f_i(\vec{y})$ for $i = 1,..,k$

3) ($\vec{x}$ is **incomparable** to $\vec{y}$): $\vec{x} \sim \vec{y}$ *iff* $f_i(\vec{x}) \not\le f_i(\vec{y}) \wedge f_i(\vec{y}) \not\le f_i(\vec{x})$ for $i = 1,..,k$

A graphical illustration of the previous concepts about *Pareto Dominance*, is provided in Fig. 18.3.

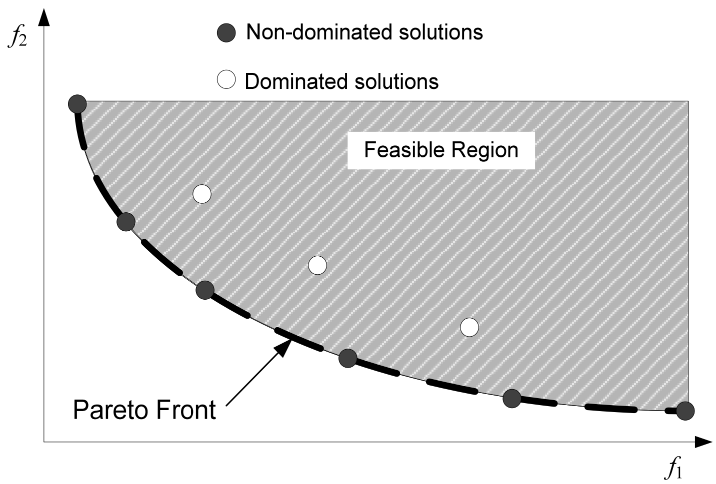

Fig. 18.3 Illustrative example of the concepts about Pareto Dominance

When the Pareto optimal solutions (or non-dominated solutions) are plotted in the objective space are collectively known as the *Pareto front*. The Pareto Front dominates all other possible solutions, and in many cases, is located on the boundary of the objective space (i.e., feasible solution space). In Fig. 18.4 a **bold** dash line is used to mark this boundary for a bi-objective problem which is known as the *Pareto front*.

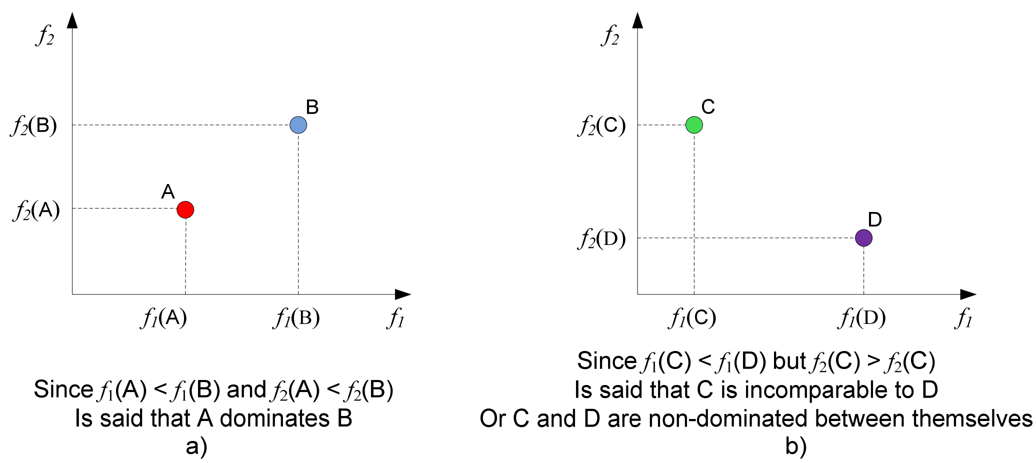

Fig. 18.4 Graphical illustration of the *Pareto Front* of a bi-objective minimization problem

In general, it is not easy to find an analytical expression of the line or surface that contains the Pareto optimal solutions, and in most cases it turns out to be impossible. The normal procedure to generate the Pareto Front is to compute many solutions in the feasible region Ω and their corresponding $f(\Omega)$. When there is a sufficient number of these, it is then possible to determine the non-dominated points and to generate a Pareto front [49].

18.5.1 Decomposition of a multi-objective problem

It is well-known that a Pareto-optimal solution to a multi-objective optimization problem, under certain conditions, could be an optimal solution of a single-objective optimization problem in which the objective is an aggregation function of all the individual objectives. Therefore, the approximation of the Pareto front can be decomposed into a number of single objective optimization sub-problems. This is the basic idea behind many traditional mathematical programing methods for approximating the Pareto front. Several methods for constructing aggregation functions can be found in the literature (see for example [52]). These methods use a weighted vector to define a scalar function. In this way, 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. Among these methods, probably the two most widely used are the *Weighted Sum* and the *Tchebycheff* approaches.

- ***Weighted Sum Method***

The classical aggregating approach to solve a multi-objective optimization problem is the *weighted sum method*. This method consists of assigning a weight w_i to each objective function so that the multi-objective problem is converted to a single objective problem with a scalar objective function of the form:

$$\begin{aligned} &\text{Minimize} \quad g(\vec{x}|w) = \sum_{i=1}^{k} w_i f_i(\vec{x}) \\ &\text{subject to} \quad x \in \Omega \end{aligned} \tag{18.21}$$

where w_i is a vector of weights usually set by the decision maker such that, $\sum_{i=1}^{k} w_i = 1$ and $w_i \geq 0$ for all i = 1… k. If all of the weights are positive, then minimizing (18.21) provides a sufficient condition for the Pareto optimality, which means that the minimum of (18.21) is always a Pareto optimal solution of the multi-objective problem [53]. As with most methods involving scalar weights w_i, setting one or more of the weights to zero may result in weakly Pareto optimal points [54]. The relative value of the weights generally reflects the relative importance of the objectives. This is another common characteristic of weighted methods. This method is easy to use, however, there are a few recognized difficulties with the weighted sum method [54-57]:

1) This method does not provide a well-distributed set of solutions along the Pareto front, with a consistent change in weights, even with a consistent Euclidean distance between consecutive solutions.

2) This method is unable to obtain points on non-convex portions of the Pareto Front.

- ***Weighted Tchebycheff Approach***

Under this scheme, the scalar optimization problem can be stated as [58]:

$$\begin{aligned} &\text{Minimize} \quad g(\vec{x}|w, z^*) = \max_{i\in\{1,\ldots,k\}} \left\{ w_i \left| f_i(\vec{x}) - z_i^* \right| \right\} \\ &\text{subject to} \quad x \in \Omega \end{aligned} \tag{18.22}$$

where w_i is a weighting vector such that, $w_i \geq 0$ for all i = 1,...,k, and $\sum_{i=1}^{k} w_i = 1$. The vector $z^* = (z_1^*, ..., z_k^*)$ represent the reference point, i.e., $z_i^* = \min\left\{ f_i(\vec{x}) | x \in \Omega \right\}$, i = 1,...,k.

Whereas the *weighted sum method* discussed previously always yields Pareto optimal solutions, this method may provide the complete Pareto optimal set, through weights variations; it provides a necessary condition for Pareto Optimality [52]. In addition, the solution using the *Tchebycheff* approach is always weakly Pareto optimal, and if the solution is unique, then it is Pareto optimal. This means that for each Pareto optimal solution x^* there exists a weight vector "*w*" such that x^* is an optimal solution of (18.22) and each optimal solution of (18.22) is a Pareto optimal solution of (18.18). Thus, it is able to obtain different Pareto optimal solutions by modifying the weighting vector.

The advantages of the *Tchebycheff* approach are as follows:

1) It provides a clear interpretation of minimizing the largest difference between f_i and the reference point z_i.

2) It can provide the complete Pareto optimal set.

3) It always provides a weakly Pareto optimal solution.

4) It is relatively well suited for generating the complete Pareto optimal set (with weight variation)

The main disadvantage is that it requires the minimization of each objective when using the referent point, which can be computationally expensive.

18.6 Teaching Learning based optimization (TLBO)

Teaching learning based optimization (TLBO) algorithm was proposed by Rao et al. [59] in 2011 as a metaheuristic algorithm inspired on the philosophy of teaching-learning process in a class between the teacher and learners (students). TLBO has emerged as a simple and efficient technique for solving single-objective complex benchmark problems and real world problems including power system applications. It has been well tested on many optimization problems [59, 60, 61-63]. Thus, it is attractive to extend the TLBO in order to solve multi-objective optimization problems [64-66, 67-68].

Metaheuristics, in general, require parameters that affect their performance. For example, DE depends on the mutation strategy adopted, and on its intrinsic control parameters such as its scaling factor and the crossover rate. PSO requires learning factors, the variation of the inertia weight and the maximum value of velocity. ABC requires the number of employed bees, onlooker bees and a value of limit. HS, requires harmony memory consideration rate, pitch adjusting rate and the number of improvisations. In contrast, TLBO does not require any specific parameter to

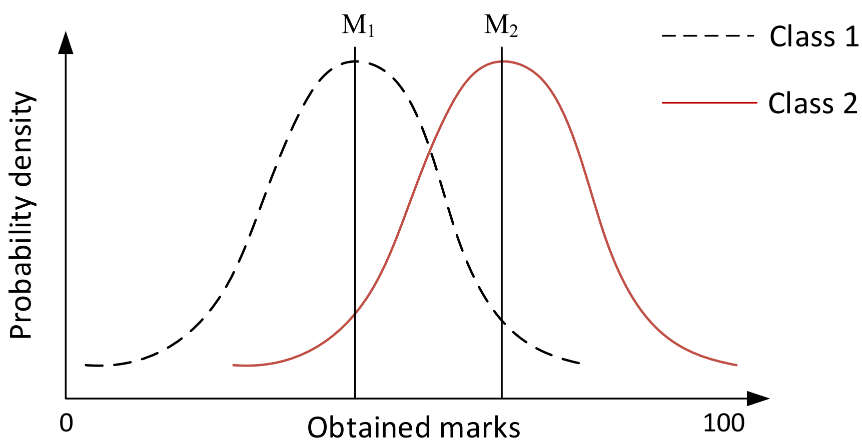

Fig. 18.5. Distribution of marks obtained by learners in two different classes

be tuned, which facilitates its implementation and use [62]. This represents another remarkable characteristic of TLBO.

The main idea of TLBO is explained in the sequel [59]. Figure 18.5 shows the distribution of marks obtained by the learners of two different classes. A normal distribution is assumed for the marks, but in actual practice it can have skewness. The normal distribution is defined as,

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{18.23}$$

where σ is the variance, μ is the mean and x is the independent variable.

It is evident from Fig. 18.5 that the teacher in class 2 is better than the teacher in class 1 since the mean of the grades, M_2, obtained by the learners in class 2 attain better results than M_1. Therefore, it can be stated that a good teacher produces a better average in learners. Students also learn from interaction between themselves, which also helps in their results. However, in practice a teacher can only move the mean of a class toward some extent depending on the capability of the class [59]. Consequently, since a teacher, who is the most experienced person on a subject in the society, influences the learner's behavior to attain good marks or grades, and it is expected that the teacher increases the knowledge level of the whole class depending on his/her skills. Therefore, the teacher will put maximum effort into teaching his/her students, but students will gain knowledge according to the quality of teaching delivered by a teacher and the quality of students attending the class [59]. In addition, the students can also gain knowledge by discussing, communicating, discovering and interacting each other.

Based on the above teaching process a mathematical model of a novel optimization technique called Teaching-Learning-based optimization (TLBO) was developed by Rao et al. [59]. Like any other nature-inspired algorithm, TLBO is a population-based method that use a population of solutions to proceed to the global solution. For TLBO, the population is described as a class of learners. In optimization algorithms, the population consists of different design variables. In TLBO, the different design variables represent the number of subjects offered to learners, and the grade attained by learner is analogous to the 'fitness', as in any other population-based optimization technique. The teacher is considered as the best solution obtained so far. The process of TLBO is based on two main phases. The first phase is the "teacher phase", which involves learning from the teacher, and the second phase is the "learner phase", which involves learning through the interaction among learners. It should be noted that the output solutions of the "teacher phase" are the input solutions for the "learner phase".

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

A brief description of the main phases is provided in the following sub-sections.

18.6.1 Teacher phase

A good teacher is one who brings his (or her) learners up to his (or her) level in terms of knowledge. But in practice this is not possible and a teacher can only move the mean of a class up to some extent depending on the capability of the class. This follows a random process which is interfered by many factors [59].

Let M_i be the mean of the class and $T_{best,i}$ be the teacher of that class (best solution so far) at the i-th iteration. Hence, $T_{best,i}$, will try to move the mean of the class (M_i) toward its own level. Thus, the new mean will be $T_{best,i}$, designated as $M_{new,i}$. The difference between the mean of the class (M_i) and the new mean ($M_{new,i}$) is expressed by [59]:

$$\Delta_i = r_i\left(M_{new,i} - T_F M_i\right) \tag{18.24}$$

where T_F is a teaching factor that weights the current mean value. The value of T_F can be either 1 or 2, which is determined randomly with equal probability as $T_F = \text{round}[1 + rand(0,1)]$, and r_i is a random number within [0, 1]. TLBO uses the current best solution to improve the existing solution, thereby increasing the convergence rate. The difference (18.24) updates the current solution according to the following expression [59]:

$$x_{new,i} = x_{old,i} + \Delta_i \tag{18.25}$$

x_{new} is accepted if it improves the function value.

18.6.2 Learning phase

A learner interacts randomly with other learners through group discussions, presentations, formal communications, etc. Thus, each learner may acquire new knowledge if the others have more knowledge than him/her. The modification of the learners is expressed as [69]:

$$
\begin{aligned}
&\textit{for } i = 1 \text{ to number of learners} \\
&\quad \textit{Randomly} \text{ select two learners } x_i \text{ and } x_j \text{ such that } x_i \neq x_j \\
&\quad \textit{if } f(x_i) < f(x_j) \\
&\qquad x_{new,i} = x_i + r_i(x_i - x_j) \\
&\quad \textit{else} \\
&\qquad x_{new,i} = x_i + r_i(x_j - x_i) \\
&\quad \textit{end if} \\
&\textit{end for}
\end{aligned}
\tag{18.26}
$$

x_{new} is accepted if it accomplishes a better objective function value. The new solutions update the initial learners and the teaching-learning process continues until the stopping criterion is achieved.

18.7 Multi-objective Teaching Learning Algorithm based on decomposition (MOTLA/D)

This is a proposed metaheuristic algorithm [70]: the Multi-objective Teaching Learning Algorithm based on decomposition (MOTLA/D). This approach is an extension of the Teaching learning based optimization algorithm previously described, and its main aim is to deal with multi-objective optimization problems employing a decomposition framework. This framework decomposes a multi-objective problem into several single-objective optimization sub-problems. In this way, a set of approximate solutions to the *Pareto front* is achieved by minimizing each sub-problem through neighborhood relationships, instead of using a Pareto ranking method. This characteristic makes MOTLA/D different from the multi-objective variants of the published TLBO algorithm since the proposed approach does not need a density estimator that distributes the solutions along the *Pareto front*.

In this chapter MOTLA/D uses the *Tchebycheff* approach to decompose a multi-objective optimization problem into single-objective sub-problems by choosing N uniformly distributed weighting vectors. The goal of the i-th sub-problem is minimize

$$g(x \mid w^i, z^*) = \max_{j \in \{1,..,m\}} \{ w_j^i \left| f_j(x) - z_j^* \right| \} \tag{18.27}$$

where $w^i = \{w_1^i, ..., w_m^i\}$ is the weighting vector with $i = 1, 2, ..., N$, and $w_j^i \geq 0$ for all objective functions $j = 1, 2, ..., m$. $\sum_j w_j^i = 1$ and vector $z^* = (z_1^*, ..., z_m^*)$ represents the reference point, i.e., $z_j^* = \min\left\{ f_j(x) \mid x \in \Omega \right\}$. If N is large enough and these weights are uniformly distributed, then under a mild condition, the N optimal solutions to these sub-problems will be a good approximation to the *Pareto front*. In case of two objective functions i.e., $m = 2$ the weighting vector, $w^i = (w_1^i, w_2^i)$ may be set as,

$$w_1^i = j/N \quad , \quad j = 1,2,...,N$$
$$w_2^i = (N-j)/N \tag{18.28}$$

This method for generating weighting vectors works well for the formulation in this proposition. Nevertheless, 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 $g(x|w^i)$. MOTLA/D solves simultaneously these sub-problems based on the philosophy of teaching-learning process in a class among teacher and learners, in a similar way as that described in the TLBO [59]. Due to $g(x|w)$ is a continuous function of w, two sub-problems are likely to have similar solutions if their weighting vectors are close from each other [71]. Therefore, any information about the weighting vectors close to w^i should be helpful for optimizing $g(x|w^i)$. Based on this observation, the neighborhood of sub-problem i, defined as $B(i)$, contains a predetermined number (T_n) of sub-problems with the closest weighting vectors with respect to w^i. The Euclidean distance is used to measure the closeness between any two weighting vectors, while it is assumed that $i \in B(i)$. That is, the i-th sub-problem is its own neighbor. The neighborhood for each sub-problem represents a group of learners or a class, responsible to solve such sub-problem. The neighborhood size T_n should be much smaller than the population size N [71]. 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 of learners ($x_{i,j}$) within the range of the parameters' boundaries by,

$$x_{i,j} = x_j^{\min} + rand(0,1)\cdot(x_j^{\max} - x_j^{\min}) \tag{18.29}$$

where i = 1,…, N and j = 1,…, D. The population size is N, i.e., the potential solutions, and D is the number of decision variables. $x_j^{\min}$ and $x_j^{\max}$ are the lower and upper limits for the j-th decision variable, respectively.

2) *Selection of the class*

The class for the i-th sub-problem is selected between the neighborhood $B(i)$ and the entire population N according to,

$$C_{i\,\text{th}} = \begin{cases} B(i) & if\ rand < \delta \\ \{1,...N\} & otherwise \end{cases} \tag{18.30}$$

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

3) *Teacher phase*

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

$$C_{i\text{-}th} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,D} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,D} \\ \vdots & \vdots & \ddots & \vdots \\ x_{\Omega_T,1} & x_{\Omega_T,2} & \cdots & x_{\Omega_T,D} \end{bmatrix} \tag{18.31}$$

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 particular design variable is calculated column-wise,

$$M_{class,i} = [mean_1, mean_2, ..., mean_D] \tag{18.32}$$

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

$$M_{new,i} = \{x_i \left| \min_{x_i \in \Omega_T} g(x_i \left| w^i, z^* \right.) \right.\} \tag{18.33}$$

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

$$x_{new} = x_{i,j} + rand_{i,j} \cdot (M_{new,i} - T_F M_{class,i}) \tag{18.34}$$

where index i corresponds to the current i-th sub-problem, $rand_{i,j}$ 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 $T_F = \text{round}[1 + rand(0,1)]$. If one or more decision variables of the new solution exceeds its predetermined boundaries, the j-th variable is set to an acceptable value through,

$$x_{new} = \begin{cases} x_j^{\min}, & if\ x_{new,j} \le x_j^{\min} \\ x_j^{\max}, & if\ x_{new,j} \ge x_j^{\max} \end{cases} \tag{18.35}$$

The new solution (x_{new}) is accepted if it improves the function value and it replaces the old one (x_i).

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,

$$\begin{aligned} & if \quad f(x_j) < f(x_k) \\ & \quad x_{new} = x_i + rand \cdot (x_j - x_k) \\ & else \\ & \quad x_{new} = x_i + rand \cdot (x_k - x_j) \\ & end \end{aligned} \tag{18.36}$$

Additionally, a polynomial mutation operator is applied to maintain solutions' diversity. This operator uses the polynomial distribution,

$$\delta_j = \begin{cases} (2 \cdot r_j)^{\frac{1}{\mu+1}} - 1, & if\ r_j < 0.5 \\ 1 - [2 \cdot (1 - r_j)]^{\frac{1}{\mu-1}}, & if\ r_j \ge 0.5 \end{cases} \tag{18.37}$$

where r_j is a random number in the interval [0, 1], and μ is a mutation distribution index. The mutated element is given by,

$$x_{new} = x_{new,j} + \left[x_j^{\max} - x_j^{\min} \right] \delta_j \tag{18.38}$$

where $x_j^{\max}$ and $x_j^{\min}$ are the upper and lower limits for the j-th decision variable, respectively. For the new solution, if one or more decision variables exceed their predetermined boundaries, the j-th variable is set to an acceptable value, (18.38). The new solution (x_{new}) is accepted if it improves the function value and replaces the old one (x_i).

5) ***Updating the Class***

At the end of each phase (*teacher* and *learner*) the next strategy is applied in order to update the class of the *i*-th sub-problem [72]. Set $n = 0$ and then do the following:

$$\begin{aligned}
&(a) \quad if \quad n = s_r \quad or \quad C \text{ is empty} \\
&\qquad \textbf{break}. \text{ Otherwise } \textit{randomly} \text{ picks an index } i \text{ from } C \\
&(b) \quad if \;\; g(x_{new} \mid w^i) \le g(x_i \mid w^i), \text{ Then set } x_i = x_{new} \\
&\qquad \text{and } n = n + 1 \\
&(c) \quad \text{remove } i \text{ form } C \text{ and go to } (a)
\end{aligned} \tag{18.39}$$

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

Summarizing, the proposed MOTLA/D algorithm may be described as in the next pseudo-code.

Step 1) Initialization

- Generate a well-distributed set of N weighting vectors by (18.28)
- Find the neighborhood of each sub-problem $B(i)$
- Generate the initial population by using eq. (18.29) and evaluate its fitness.
- Initialize the reference point z^*

Step 2)

For $i = 1$ to N

- Determine the class (C_i) according to eq. (18.30)
- ***Teacher phase***: Create a new solution (x_{new}) by eq. (18.34)
- Update the reference point z^*
- Update solutions by eq. (18.39)

End For *i*

Step 3)

For $i = 1$ to N

- Determine the class (C_i) according to eq. (18.30)
- ***Learner Phase***: Create a new solution (x_{new}) by eq. (18.36) and eq. (18.38)
- Update the reference point z^*
- Update solutions by eq. (18.39)

End For *i*

Step 4) Stop Criterion

If the stop condition is satisfied, (such as getting the maximum number of iterations or the maximum number of function evaluations), then stop MOTLA/D. Otherwise, go to **Step 2)**.

18.7.1 Remarks about MOTLA/D

The flow chart of the proposed MOTLA/D algorithm is presented in Figure 18.6. Notice that algorithm MOABC/D (multi-objective Artificial Bee Colony based on decomposition) is included too. Although the proposed approaches are based on the MOEA/D (multi-objective Evolutionary Algorithm based on Decomposition) framework [71], an important difference between the original scheme of decomposition in MOEA/D and the proposed one, is the number of stages. While MOEA/D uses only one stage to apply a search strategy, in the proposed scheme two individual stages are implemented, Fig. 18.6. The main phases of each algorithm are included within each of these stages. The applied phase depends on the used algorithm. This scheme allows that each phase explores individually the searching space, and also helps to compose hybrid algorithms easily by including the desired strategy within the stages.

It is noteworthy, that the *onlooker phase* and the *learner phase* of the proposed MOABC/D and MOLTA/D, respectively, has been modified from the original ABC and TLBO. These two phases in the original ABC and TLBO, 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 was needed. In the proposed onlooker and learner phase, 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.

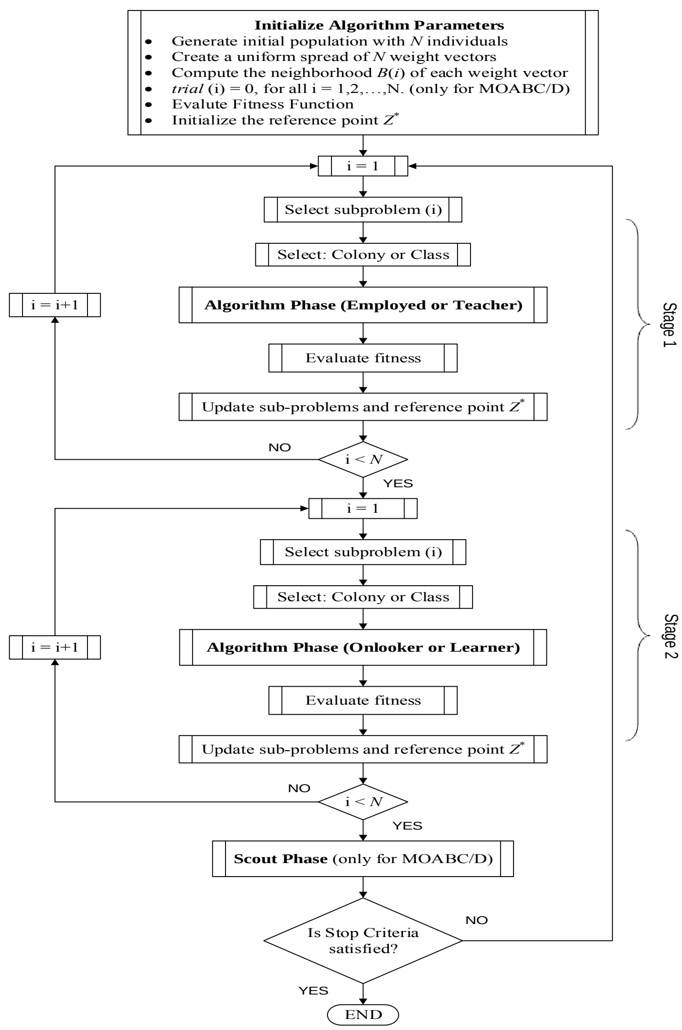

Fig. 18.6. Flowchart of MOABC/D and MOTLA/D

18.8 Performance assessment

The ultimate goal of a multi-objective optimization algorithm is to identify solutions in the Pareto optimal set. However, identifying the entire Pareto optimal set is practically impossible. In addition, for many problems, especially for combinatorial optimization problems, the solution optimality proof is computationally infeasible [73]. Therefore, a practical approach to multi-objective optimization is to investigate a set of solutions (the best-known Pareto set) that represents the Pareto Optimal Set as well as possible. With these concerns in mind, a multi-objective optimization approach should achieve the following three conflicting goals [74]:

1) The estimated *Pareto front* should be as close as possible to the true *Pareto front*. Ideally, it should be a subset of the Pareto optimal set.

2) Solutions in the estimated Pareto set should be uniformly distributed and diverse over the true *Pareto front* in order to provide the decision-maker a true picture of tradeoffs.

3) The estimated *Pareto front* should capture the whole spectrum of the true *Pareto front*. This requires investigating solutions at the extreme ends of the objective function space.

These tree tasks cannot be measured adequately by a performance measure. Therefore, in order to assess the performance of the proposed algorithms the following performance measures are adopted.

18.8.1 Inverted generational distance (IGD)

The concept of the generational distance (GD) was introduced by David Veldhuizen and Gary Lamont as a way of estimating how far the Pareto-optimal solutions obtained by an algorithm are from those in the *Pareto front* of the problem [75]. The inverted generational distance (IGD), as its name suggests, it is an inverted variation of the widely used GD performance metric. There are different ways of computing and averaging the distances in the GD (e.g., [75] and [76]). The version of the IGD used in this chapter inverts the γ version in [76]. This means, that the *Pareto front* is used as the reference point and all its elements are compared with respect to the approximation generated by the algorithms. This measure is described as follows.

Let P^* be a set of points uniformly distributed on the *Pareto front* and A be the approximation obtained by an algorithm. IGD represents the average distance from P^* to A defined as,

$$IGD(A,P^*) = \frac{\sum_{v \in P^*} d(v,A)}{\left|P^*\right|} \tag{18.40}$$

where d (v, A) is the minimum Euclidean distance between v and the points in A. If the points in the set P^* can appropriately represent the Pareto front, IGD may measure both the spread and convergence of set A. The smaller the value of this metric is, the better is the performance of the algorithm. A value of IGD equal to zero implies that all obtained solutions lie on the true *Pareto front* and have the best possible spreads.

18.8.2 Spacing (S_p)

This performance measure was proposed by Schott [77], and it quantifies the spread of solutions (i.e., how uniformly distributed the solutions are) along a Pareto front approximation. This is defined by,

$$S_p = \sqrt{\frac{1}{|n-1|}\sum_{i=1}^{n}\left(\bar{d} - d_i\right)^2} \tag{18.41}$$

where n is the number of non-dominated solutions, $d_i = \min_{i,i \neq j} \sum_{m=1}^{k} \left|f_m^i - f_m^j\right|$, $i, j = 1,2,..,n$, and k denotes the number of objectives, while $\bar{d} = \sum_{i=1}^{n} d_i \big/ |n|$. A value of zero implies that all solutions are uniformly spread (i.e., the best possible performance).

18.8.3 Coverage of two sets $C(A,B)$

This performance measure was proposed by Zitzler et al. [74]. This metric compares two sets of non-dominated solutions (A, B) and evaluates the percentage of individuals in one set dominated by the individuals on the other set. This performance measure is defined by,

$$C(A,B) = \frac{\left|\{b \in B \middle| \exists\ a \in A : a \preceq B\}\right|}{|B|} \tag{18.42}$$

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

18.9 Case studies

This chapter compares the effectiveness and performance of the proposed algorithm with respect to that of the MOEA/D. Both MOTLA/D and MOEA/D have been applied to three test systems. In the first case study, we consider the nine-bus test system; this system consists of 9 transmission lines and 3 generating units. The system model and data can be found in [78]. The second case study is related to the IEEE-26 bus test system, which has 26 buses, 46 branches, 6 generators, 7 transformers, and 9 shunt capacitors. The detailed data of this problem can be found in [79]. Finally, in the third case study, the IEEE 118-bus test system is used. The system has 54 generator buses, 64 load buses and 186 transmission lines with 9 tap setting transformers. The complete system data are taken from [80]. For each case study, 20 independent runs are performed. The number of sub-problems considered by each algorithm are 100 for cases 1 and 2, and 200 for case 3. It is worth mentioning that the stopping criterion of each algorithm is the number of generations N_{gen}, (120, 180, and 200 generations for cases 1, 2 and 3, respectively). For all test instances, the control parameter settings utilized by the MOTLA/D and MOEA/D are summarized in the following. The neighborhood size (T_{size}), is 30. The distribution index (μ), used in the polynomial mutation, is 20. The parameter of scale factor (F_s) associated with MOEA/D, which represents the amount of perturbation added to the main parent, is 0.5. The Crossover rate (P_{cr}) associated with MOEA/D, which determines the quantity of elements to be exchanged by the crossover operator, is 1. Finally, a mutation rate P_m=1/n is taken into account, where n is the number of decision variables. This parameter indicates the probability that each decision variable has of being changed.

18.9.1 Performance measures

There are two goals in multi-objective optimization: (*a*) to achieve convergence to the Pareto-optimal set; and (*b*) to obtain a well-distributed set of solutions along the Pareto front. These two tasks cannot be measured adequately by one performance measure each. Therefore, in order to assess the algorithms' performance two performance measures are adopted: (*i*) spacing (18.41), and (*ii*) coverage of two sets (18.42).

18.9.2 Experimental results and comparison

The advantage of evolutionary algorithms is that they have minimum requirements regarding the problem formulation; objectives can be easily added, removed, or modified. Likewise, in this application, they are well-suited to tackle highly complex problems such as those existing in power systems.

All algorithms compared were implemented in MATLAB 7.3 and run on a PC with a Pentium core duo processor operating @ 2 GHz with 2 GB RAM. Three test systems were used: the IEEE 9-bus, IEEE 26-bus, and the IEEE 118-bus systems, operating under their corresponding base case. For each test power system and each algorithm, 20 runs were executed. The following results correspond to the best solution attained by each algorithm, with respect to the coverage of two set performance measure.

18.9.2.1 Case study 1: 9-buses test system

The decision variables are related to the generator voltage V_{gi}, and range in the interval [1.0, 1.05] pu. Table 1 summarizes the best solution for minimum reactive losses calculated through MOTLA/D and MOEA/D. Notice that for the optimized case, a reduction of the reactive losses and voltage stability index is attained. Both algorithms reduce the losses in 38.54%, which represents an important proportion of the losses with respect to the base case. Regarding the voltage stability index, L_{index}, this has been decreased in 10.99% relative to the base case. In summary, the objective functions become:

$f_{Rloss\ (Base\ Case)}$ = 0.0755 p.u $f_{Lindex\ (Base\ Case)}$ = 0.1673

$f_{Rloss\ (MOTLA/D)}$ = 0.0464 p.u $f_{Lindex\ (MOTLA/D)}$ = 0.1489

$f_{Rloss\ (MOEA/D)}$ = 0.0464 p.u. $f_{Lindex\ (MOEA/D)}$ = 0.1489

Table 1. Case study 1: best solutions calculated by both MOTLA/D and MOEA/D

Decision variables	Base case	f_{Rloss}		f_{Lindex}	
		MOTLA/D	MOEA/D	MOTLA/D	MOEA/D
V_{g1} (p.u)	1.04	1.05	1.05	1.05	1.05
V_{g2} (p.u)	1.02533	1.0376	1.0377	1.05	1.05
V_{g3} (p.u)	1.02536	1.0328	1.0331	1.0499	1.0499

18.9.2.2 Case study 2: IEEE 26-buses test system

The decision variables are related to the generator voltage V_{gi}, and range in the interval [1.0, 1.05] pu. Likewise, another decision variable is the transformer tap setting T_i, which ranges in the interval [0.95, 1.05].

The best solution for minimum reactive losses (R_{loss}) and voltage stability index (L_{index}) is summarized in Table 2. The minimum R_{loss} and L_{index} for the base case is *0.6302* p.u. and *0.1241*, respectively. As can be noticed, MOTLA/D estimates R_{loss} = 0.1487 p.u and L_{index} = 0.1023, while MOEA/D attains R_{loss} = 0.2105 p.u and L_{index} = 0.0995. This means that MOTLA/D reaches 76.4% reduction in losses and 17.56% reduction in L_{index} with respect to the base case. Meanwhile, MOEA/D reaches 66.59% reduction in losses and 19.82% reduction in L_{index} with respect to the base case. It is assumed that the tap positions vary among 32 positions (16 up, and 16 down), and the closest is selected in Table 2. In summary, the objective functions become:

$f_{Rloss\ (Base\ Case)}$ = 0.6302 p.u $f_{Lindex\ (Base\ Case)}$ = 0.1241

$f_{Rloss\ (MOTLA/D)}$ = 0.1487 p.u $f_{Lindex\ (MOTLA/D)}$ = 0.1023

$f_{Rloss\ (MOEA/D)}$ = 0.2105 p.u $f_{Lindex\ (MOEA/D)}$ = 0.0995

Table 2. Case study 2: best solutions calculated by both MOTLA/D and MOEA/D

Decision variables	Base case	f_{Rloss}		f_{Lindex}	
		MOTLA/D	MOEA/D	MOTLA/D	MOEA/D
V_{g1} (p.u)	1.025	1.0265	1.0466	1.05	1.0498
V_{g2} (p.u)	1.02	1.0109	1.0112	1.0429	1.0365
V_{g3} (p.u)	1.03	1.021	1.0137	1.0498	1.0420
V_{g4} (p.u)	1.045	1.0498	1.0487	1.05	1.0499

V_{g5} (p.u)	1.045	1.0248	1.0445	1.05	1.0379
V_{g26} (p.u)	1.015	1.0459	1.0216	1.0499	1.0500
T_3	0.96 (-13)	1.0135 (4)	0.9705 (-9)	0.95 (-16)	0.9850(-5)
T_6	0.96 (-13)	0.95 (-16)	1.0493 (13)	0.95 (-16)	1.0129(1)
T_8	1.017 (5)	1.0016 (1)	0.9840 (-6)	0.95 (-16)	0.9579(-14)
T_9	1.05 (16)	0.9637 (-12)	0.9703 (-10)	0.95 (-16)	0.9500(-16)
T_{10}	1.05 (16)	0.9735 (-8)	0.9598 (-13)	0.95 (-16)	0.9507(-16)
T_{15}	0.95 (-16)	0.964 (-12)	0.9507 (-16)	0.95 (-16)	0.9515(-16)
T_{18}	0.95 (-16)	0.9768 (-7)	0.9730 (-9)	0.95 (-16)	0.9500(-16)

18.9.2.3 Case study 3: 118-buses tests system

The decision variables are related to the generator voltage V_{gi}, and range in the interval [0.98, 1.05] pu. Likewise, another decision variable is the transformer tap setting T_i, which ranges in the interval [0.95, 1.05].

Table 3 summarizes the optimal values for the two objective functions (R_{loss}) and (L_{index}) estimated by both algorithms. The minimum R_{loss} and L_{index} for the base case became 7.8223 p.u and 0.0693, respectively. MOTLA/D reduced reactive losses from 7.8223 p.u to 6.9097 p.u (a reduction of approximately 11.66%) and improved the L_{index} from 0.0693 to 0.0630 (a reduction of approximately 9.1%). Meanwhile, MOEA/D reduced reactive losses from 7.8223 p.u to 6.9116 p.u (a reduction of approximately 11.64%) and improved the L_{index} from 0.0693 to 0.0630 (a reduction of approximately 9.1%). It is assumed that taps vary among 32 positions (16 up, and 16 down), and the closest is selected in Table 3. In summary, the objective functions become:

$f_{Rloss\ (Base\ Case)} = 7.8223$ p.u $\quad f_{Lindex\ (Base\ Case)} = 0.0693$

$f_{Rloss\ (MOTLA/D)} = 6.9097$ p.u $\quad f_{Lindex\ (MOTLA/D)} = 0.0630$

$f_{Rloss\ (MOEA/D)} = 6.9116$ p.u $\quad f_{Lindex\ (MOEA/D)} = 0.0630$

Table 3. Case study 3: best solutions calculated by both MOTLA/D and MOEA/D

Decision variables	Base case	f_{Rloss}		f_{Lindex}	
		MOTLA/D	MOEAD	MOTLA/D	MOEAD
V_{g1} (p.u)	0.955	1.0356	1.0334	1.0285	1.0273
V_{g4} (p.u)	0.998	1.05	1.05	1.0497	1.05
V_{g6} (p.u)	0.99	1.0472	1.0451	1.0459	1.0401
V_{g10} (p.u)	1.05	1.0499	1.05	1.0491	1.05
V_{g19} (p.u)	0.963	1.0435	1.037	1.0497	1.0397
V_{g24} (p.u)	0.992	1.0497	1.0436	1.0498	1.047
V_{g27} (p.u)	0.968	1.037	1.0371	1.049	1.0327

V_{g31} (p.u)	0.967	1.033	1.031	1.0362	1.035
V_{g36} (p.u)	0.98	1.0495	1.0493	1.0493	1.0495
V_{g40} (p.u)	0.97	1.0398	1.0351	1.0415	1.0461
V_{g42} (p.u)	0.985	1.0499	1.0435	1.0499	1.0498
V_{g54} (p.u)	0.955	1.0187	1.0122	1.0203	1.0118
T_8	0.985 (-5)	0.9935 (-2)	0.9903 (-3)	0.9871 (-4)	0.9911 (-3)
T_{32}	0.96 (-13)	1.000 (1)	1.0006 (1)	1.0013 (1)	0.9995 (-1)
T_{36}	0.96 (-13)	0.9948 (-2)	0.9965 (-5)	1.0006 (1)	1.0032 (2)
T_{51}	0.955 (-14)	0.983 (-6)	0.9857 (-5)	0.9725 (-9)	0.9814 (-6)
T_{93}	0.96 (-13)	1.0259 (9)	1.0177 (6)	1.05 (16)	1.0223 (8)
T_{95}	0.985 (-5)	1.0179 (6)	1.013 (5)	1.0377 (12)	1.0136 (5)

18.9.2.4 Comparison of MOTLA/D and MOEA/D

For each case study, MOTLA/D and MOEA/D are evaluated using the two performance measures (18.41) and (18.42). The results are summarized in Tables 4 and 5. Each of these Tables present the average and the standard deviation (in brackets) of each performance measure for each case study. The best results are displayed in **boldface**.

Notice in Table 4 that the proposed approach (MOTLA/D) outperformed MOEA/D in all cases regarding the *Coverage of two sets (C)*. This indicates that the proposed approach produces more solutions that dominate (according to Pareto optimality) the solutions produced by MOEA/D. The difference among the non-dominated solutions produced by MOTLA/D and MOEA/D is more noticeable in cases 2 and 3. According to Table 4, in the case study 2, MOTLA/D produced solutions which dominate to 55% of the solutions generated by MOEA/D. In contrast, MOEA/D produced solutions that dominate only to 30% of the solutions generated by MOTLA/D. In the case study 3, the solutions obtained by MOTLA/D dominate about 41% of the solutions generated by MOEA/D; in contrast, MOEA/D produced solutions that only dominate 25% of the solutions generated by MOTLA/D.

Regarding Spacing (S), MOEA/D attains relatively better results for cases 1 and 2. However, since coverage (which relates to convergence) has precedence over spread, we can conclude that our proposed MOTLA/D outperformed MOEA/D in the analyzed cases of study.

The Pareto's fronts obtained by MOTLA/D and MOEA/D for all cases are depicted in Fig. 18.7. These curves represent the best case, according to the performance measures defined in (18.41)-(18.42). Notice that both algorithms perform similarly for case study 1. The difference between the approximations obtained by MOTLA/D and MOEA/D is more noticeable in cases 2 and 3. It is noteworthy that MOTLA/D is able to achieve more distributed solutions in the case study 3. A distribution of non-dominated solutions as uniform as possible along the Pareto front, ensures that there are not big gaps in the Pareto front and, therefore, all the different types of trade-off solutions are generated. This is relevant, because if big gaps occur, it may happen that the trade-off solution in which we are interested on is not produced (i.e., the solution of concern may be located in the missing portion of the Pareto front).

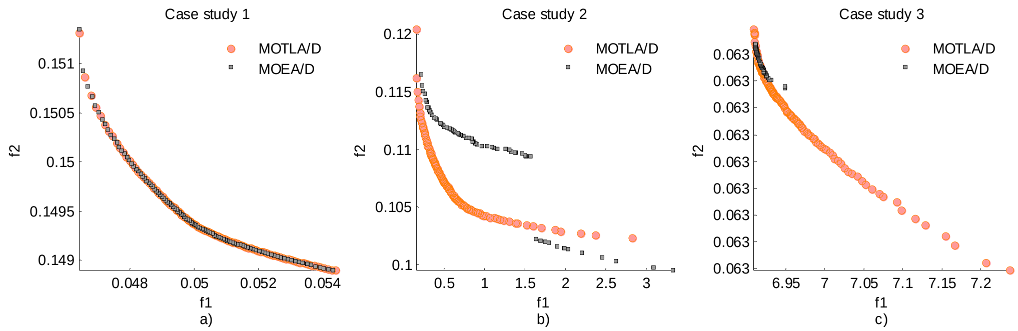

Fig. 18.7. From left to right: Pareto fronts (f1 – f2) for both the MOTLA/D and MOEA/D (best result): (a) Case study 1; (b) Case study 2; (c) Case study 3.

Table 4. Results of *Coverage of two set(C)* performance measure

TEST	C(MOTLA/D,MOEA/D)	C(MOEA/D,MOTLA/D)
	Average (*Std. Dev.*)	Average (*Std. Dev.*)
Case study 1	**0.014** (*0.009*)	0.011 (*0.011*)
Case study 2	**0.545** (*0.415*)	0.293 (*0.335*)
Case study 3	**0.411** (*0.354*)	0.252 (*0.224*)

Table 5. Results of *Spacing (S)* performance measure

TEST	MOTLA/D	MOEA/D
	Average (*Std. Dev.*)	Average (*Std. Dev.*)
Case study 1	0.0228 (*0.000*)	**0.0197** (*0.002*)
Case study 2	0.0256 (*0.002*)	**0.0184** (*0.003*)
Case study 3	**0.0254** (*0.013*)	0.0338 (*0.014*)

Conclusions

This chapter proposed a multi-objective teaching learning algorithm based on decomposition (MOTLA/D) for solving a reactive power system problem. The effectiveness and performance of MOTLA/D were compared with respect to those of MOEA/D, which represents a state-of-the-art algorithm, in three cases of study: 9-, 26-, and 118-buses test systems. The results indicate that the proposed algorithm was able to obtain better solutions than MOEA/D in all the analyzed cases. Thus, it may be concluded that the proposed algorithm is a reliable choice for power systems applications. In this chapter, an improvement of both reactive losses and voltage stability were attained. Likewise, some other additional objectives could be taken into account as well.

References

[1] W. Stadler. Multicriteria Optimization in Engineering and the Sciences. New York: Plenum Press, 1st Edition, 1988.
[2] M. Belazzoug, and M. Boudour, "FACTS placement multiobjective optimization for reactive power system compensation," in Systems Signals and Devices (SSD), 2010 7th International Multi-Conference, pp. 1-6, June 27-30, 2010.
[3] A.A. Abou El Ela, M.A. Abido, S.R. Spea, "Optimal power flow using differential evolution algorithm". Electrical Engineering, vol. 91, no. 2, pp. 69-78, 2009.
[4] 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 & Energy Systems, vol. 33, no. 2, pp. 219-228, February 2011.
[5] M. A. Abido, "Optimal power flow using particle swarm optimization". International Journal of Electrical Power and Energy systems, vol. 24, no. 7, pp. 563-571, 2002.
[6] W. Yan, S. Lu, and D. C. Yu, "A Novel Optimal Reactive Power Dispatch Method Based on an Improved Hybrid Evolutionary Programming Technique," IEEE Transactions on Power Systems , vol.19, no.2, pp. 913- 918, May 2004.
[7] C. Sumpavakup, I. Srikun and S. Chusabapiputt, "A solution to the Optimal Power Flow Using Artificial Bee Colony Algorithm," International Conference on Power System Technology, pp. 1-5, October 2010.
[8] H.W. Dommel, and W.F. Tinny, "Optimal Power Flow Solutions," IEEE Trans. on Power Apparatus and Systems, PAS- 87, pp 1866-1876, 1968.
[9] 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, PAS-104, pp.1147-1153, 1985.
[10] S. Granville, "Optimal reactive power dispatch through interior point methods," IEEE Trans. on Power Systems, vol.9, no.1, pp.136-146, Feb. 1994.
[11] G.R.M. da Costa, "Optimal reactive dispatch through primal- Dual method," IEEE Trans. on Power Systems, vol.12, no.2, pp. 669-674, May 1997.
[12] M. O. Mansour, and T. M. Abdel-Rahman, "Non-linear VAR Optimization using Decomposition and Coordination," IEEE Trans. Power Apparat. Systems, PAS-103, no. 2, pp. 246-255, 1984.
[13] C. C. Kuo, P. H. Chen, C. L. Hsu, y 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.
[14] 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.
[15] S. Ramesh, S. Kannan, and S. Baskar, "Multi-objective evolutionary algorithm based Reactive Power Dispatch with thyristor controlled series compensator," in Power Electronics, Drives and Energy Systems (PEDES) & 2010 Power India, 2010 Joint International Conference on, pp.1-7, Dec. 20-23, 2010.
[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] H. Li and Q. Zhang, "Multiobjective Optimization Problems with Complicated Pareto Sets, MOEA/D and NSGA-II", IEEE Trans on Evolutionary Computation, vol. 13, no. 2, April 2009.
[18] S. Madan and K. E. Bollinger, «Applications of artificial intelligence in power systems», *Electric Power Systems Research*, vol. 41, No. 2, pp. 117-131, may 1997.
[19] V. Miranda, D. Srinivasan, and L. Proença, «Evolutionary computation in power systems», *International Journal of Electrical Power & Energy Systems*, vol. 20, No. 2, pp. 89-98, 1998.

[20] M. R. AlRashidi and M. E. El-Hawary, «Applications of computational intelligence techniques for solving the revived optimal power flow problem», *Electric Power Systems Research*, vol. 79, No. 4, pp. 694-702, 2009.
[21] X. Xia and A. M. Elaiw, «Optimal dynamic economic dispatch of generation: A review», *Electric Power Systems Research*, vol. 80, No. 8, pp. 975-986, 2010.
[22] C. Dai, W. Chen, Y. Zhu, and X. Zhang, «Reactive power dispatch considering voltage stability with seeker optimization algorithm», *Electric Power Systems Research*, vol. 79, No. 10, pp. 1462-1471, 2009.
[23] H. L. Liao, Q. H. Wu, and L. Jiang, «Multi-objective optimization by reinforcement learning for power system dispatch and voltage stability», on *Innovative Smart Grid Technologies Conference Europe (ISGT Europe), 2010 IEEE PES*, pp. 1-8.
[24] B. K. Panigrahi, V. R. Pandi, R. Sharma, Swagatam Das, and Sanjoy Das, «Multiobjective bacteria foraging algorithm for electrical load dispatch problem», *Energy Conversion and Management*, vol. 52, No. 2, pp. 1334-1342, 2011.
[25] S. Sivasubramani and K. S. Swarup, «Multi-objective harmony search algorithm for optimal power flow problem», *International Journal of Electrical Power & Energy Systems*, vol. 33, No. 3, pp. 745-752, 2011.
[26] T. Niknam, F. Golestaneh, and M. S. Sadeghi, «θ-Multiobjective Teaching- Learning-Based Optimization for Dynamic Economic Emission Dispatch», *IEEE Systems Journal*, vol. 6, No. 2, pp. 341-352, 2012.
[27] A. Khorsandi, S. H. Hosseinian, and A. Ghazanfari, «Modified artificial bee colony algorithm based on fuzzy multi-objective technique for optimal power flow problem», *Electric Power Systems Research*, vol. 95, pp. 206-213, 2013.
[28] D. Karaboga and B. Basturk, «Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems», on *Proceedings of the 12th international Fuzzy Systems Association world congress on Foundations of Fuzzy Logic and Soft Computing*, Cancún, México, 2007, vol. 4529, pp. 789-798.
[29] 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, 2011.
[30] T. Niknam, R. Azizipanah-Abarghooee, and M. Rasoul Narimani, «A new multiobjective optimization approach based on TLBO for location of automatic voltage regulators in distribution systems», *Engineering Applications of Artificial Intelligence*, vol. 25, No. 8, pp. 1577-1588, 2012.
[31] B. Akay and D. Karaboga, «Artificial bee colony algorithm for large-scale problems and engineering design optimization», *Journal of Intelligent Manufacturing*, vol. 23, No. 4, pp. 1001-1014, 2012.
[32] R. V. Rao, V. J. Savsani, and D. P. Vakharia, «Teaching–Learning-Based Optimization: An optimization method for continuous non-linear large scale problems», *Information Sciences*, vol. 183, No. 1, pp. 1-15, 2012.
[33] M. R. Nayak, C. K. Nayak, and P. K. Rout, «Application of Multi-Objective Teaching Learning based Optimization Algorithm to Optimal Power Flow Problem», *Procedia Technology*, vol. 6, pp. 255-264, 2012.
[34] S. Frank, I. Steponavice, and S. Rebennack, «Optimal power flow: a bibliographic survey I», *Energy Systems*, vol. 3, No. 3, pp. 221-258, 2012.
[35] S. Frank, I. Steponavice, and S. Rebennack, «Optimal power flow: a bibliographic survey II», *Energy Systems*, vol. 3, No. 3, pp. 259-289, 2012.
[36] C. W. Taylor, *Power System Voltage Stability*. McGraw-Hill, 1993.
[37] T. van Cutsem and C. Vournas, *Voltage Stability of Electric Power Systems*. Springer, 2007
[38] C. A. Cañizares, Editor. «Voltage stability assessment: concepts, practices and tools», *lEEE-PES Power Systems Stability Subcommittee Special Publication, Product Number SP101PSS (ISBN 0780378695)*, 2003.
[39] 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, 1986.
[40] K. Deb, «An efficient constraint handling method for genetic algorithms», *Computer Methods in Applied Mechanics and Engineering*, vol. 186, No. 2-4, pp. 311-338, 2000.
[41] I. H. Osman and G. Laporte, «Metaheuristics: A bibliography», *Ann Oper Res*, vol. 63, n.o 5, pp. 511-623, 1996.
[42] D. E. Goldberg, Genetic algorithms in search, optimization, and machine learning, 1st ed. Boston: Addison-Wesley, 1989.
[43] R. Storn and K. Price, «Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces», Journal of Global Optimization, vol. 11, n.o 4, pp. 341-359, 1997.
[44] J. Kennedy and R. Eberhart, «Particle swarm optimization», on Neural Networks, 1995. Proceedings. IEEE International Conference, 1995, vol. 4, pp. 1942-1948 vol.4.
[45] J. F. Kennedy, R. C. Eberhart, and Y. Shi, Swarm intelligence. San Francisco: Morgan Kaufmann Publishers, 2001.

[46] D. Karaboga, «An idea based on Honey Bee Swarm for Numerical Optimization», Erciyes University, Computer Engineering Department, Technical report - TR06, 2005.
[47] D. Karaboga and B. Basturk, «A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm», Journal of Global Optimization, vol. 39, n.o 3, pp. 459-471, 2007.
[48] Z. W. Geem, J. H. Kim, and G. V. Loganathan, «A New Heuristic Optimization Algorithm: Harmony Search», SIMULATION, vol. 76, n.o 2, pp. 60-68, 2001.
[49] C. C. Coello, G. B. Lamont, and D. A. van Veldhuizen, Evolutionary Algorithms for Solving Multi-Objective Problems, 2nd ed. Springer, 2007.
[50] A. Osyczka, «Multicriteria optimization for engineering design», on Design Optimization, John Gero, Ed. Academic Press, 1985, pp. 193-227.
[51] V. Pareto, «Cours d'économie politique», vol. I and II. Lausanne: F. Rouge, 1896-1897.
[52] K. Miettinen, Nonlinear multiobjective optimization. Boston: Kluwer Academic Publishers, 1999.
[53] L. Zadeh, «Optimality and non-scalar-valued performance criteria», IEEE Transactions on Automatic Control, vol. 8, n.o 1, pp. 59-60, 1963.
[54] R. T. Marler and J. S. Arora, «The weighted sum method for multi-objective optimization: new insights», Structural and Multidisciplinary Optimization, vol. 41, n.o 6, pp. 853-862, 2010.
[55] J. Koski, «Defectiveness of weighting method in multicriterion optimization of structures», Communications in Applied Numerical Methods, vol. 1, n.o 6, pp. 333–337, 1985.
[56] W. Stadler and J. Dauer, «Multicriteria Optimization in Engineering: A Tutorial and Survey», on Structural Optimization: Status and Promise, Manohar P. Kamat, Ed. Washington DC: American Institute of Aeronautics and Astronautics, 1992, pp. 209-249.
[57] I. Das and J. E. Dennis, «A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems», Structural Optimization, vol. 14, n.o 1, pp. 63-69, 1997.
[58] V. Joseph Bowman Jr, «On the Relationship of the Tchebycheff Norm and the Efficient Frontier of Multiple-Criteria Objectives», on Multiple Criteria Decision Making, H. Thiriez and S. Zionts, Eds. Springer Berlin Heidelberg, 1976, pp. 76-86.
[59] 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, n.o 3, pp. 303-315, 2011.
[60] R. V. Rao, V. J. Savsani, and D. P. Vakharia, «Teaching–Learning-Based Optimization: An optimization method for continuous non-linear large scale problems», *Information Sciences*, vol. 183, n.o 1, pp. 1-15, 2012.
[61] S. Chandra Satapathy, A. Naik, and K. Parvathi, «High dimensional real parameter optimization with teaching learning based optimization», *International Journal of Industrial Engineering Computations*, vol. 3, n.o 5, pp. 807-816, 2012.
[62] R. V. Rao and V. Patel, «An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems», *international journal of industrial engineering computations*, vol. 3, n.o 4, pp. 535-560, 2012.
[63] V. Toğan, «Design of planar steel frames using Teaching–Learning Based Optimization», *Engineering Structures*, vol. 34, pp. 225-232, 2012.
[64] T. Niknam, R. Azizipanah-Abarghooee, and M. Rasoul Narimani, «A new multi objective optimization approach based on TLBO for location of automatic voltage regulators in distribution systems», *Engineering Applications of Artificial Intelligence*, vol. 25, n.o 8, pp. 1577-1588, 2012.
[65] T. Niknam, F. Golestaneh, and M. S. Sadeghi, «θ-Multiobjective Teaching-Learning-Based Optimization for Dynamic Economic Emission Dispatch», *IEEE Systems Journal*, vol. 6, n.o 2, pp. 341-352, 2012.
[66] M. R. Nayak, C. K. Nayak, and P. K. Rout, «Application of Multi-Objective Teaching Learning based Optimization Algorithm to Optimal Power Flow Problem», *Procedia Technology*, vol. 6, pp. 255-264, 2012.
[67] R. V. Rao and V. Patel, «Multi-objective optimization of heat exchangers using a modified teaching-learning-based optimization algorithm», *Applied Mathematical Modelling*, vol. 37, n.o 3, pp. 1147-1162, 2013.
[68] R. Venkata Rao and V. Patel, «Multi-objective optimization of two stage thermoelectric cooler using a modified teaching–learning-based optimization algorithm», *Engineering Applications of Artificial Intelligence*, vol. 26, n.o 1, pp. 430-445, 2013.
[69] A. Mohamed, G. B. Jasmon, and S. Yusoff, «A static voltage collapse indicator using line stability factors», *Journal of industrial technology*, vol. 7, n.o 1, pp. 73-85, 1989.
[70] Miguel Alfonso de Jesús Medina López, *Meta-heuristic algorithms based on decomposition for multi-objective optimal power flow problems*, *Dissertation*, Cinvestav-Unidad Guadalajara, Julio 2014.
[71] Q. Zhang and H. Li, «MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition», *IEEE Transactions on Evolutionary Computation*, vol. 11, n.o 6, pp. 712-731, 2007.

[72] H. Li and Q. Zhang, «Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II», *IEEE Transactions on Evolutionary Computation*, vol. 13, n.o 2, pp. 284-302, 2009.
[73] A. Konak, D. W. Coit, and A. E. Smith, «Multi-objective optimization using genetic algorithms: A tutorial», *Reliability Engineering & System Safety*, vol. 91, n.o 9, pp. 992-1007, 2006.
[74] E. Zitzler, K. Deb, and L. Thiele, «Comparison of Multiobjective Evolutionary Algorithms: Empirical Results», *Evolutionary Computation*, vol. 8, n.o 2, pp. 173–195, 2000.
[75] D. A. V. Veldhuizen and G. B. Lamont, «Evolutionary Computation and Convergence to a Pareto Front», on *Late Breaking Papers at the Genetic Programming 1998 Conference*, Stanford University, California, 1998, pp. 221–228.
[76] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, «A fast and elitist multiobjective genetic algorithm: NSGA-II», *IEEE Transactions on Evolutionary Computation*, vol. 6, n.o 2, pp. 182-197, 2002.
[77] J. R. Schott, «Fault tolerant design using single and multicriteria genetic algorithm optimization», Thesis, Massachusetts Institute of Technology, 1995.
[78] P. W. Sauer, M. A. Pai. Power System Dynamics and Stability. Prentice Hall. 1997.
[79] S. Hadi. Power System Analysis. New York: McGraw-Hill, 2004.
[80] Power System Test Case Archive, 2006. Available at: http://www.ee.washington.edu/research/pstca/pf118/pg_tca118bus.htm