Military Technical College
Kobry El-Kobbah,
Cairo, Egypt

9th International Conference
on Electrical Engineering
ICEENG 2014

Hybrid Particle Swarm Optimization Based Local Search Technique for Reactive Power Compensation Problem

By

A.A. Mousa

Department of Mathematics and Statistics, Faculty of sciences, Taif University,
Department of Basic Engineering Science, Faculty of Engineering, Menoufia University, Egypt.

M.A. El-Shorbagy *

Department of Basic Engineering Science, Faculty of Engineering, Menoufia University, Egypt.

Abstract:

This paper presents an enhanced particle swarm optimization (PSO) algorithm, which applied to reactive power compensation (RPC) problem. It integrates the merits of both GAs and PSO and it has two characteristic features. Firstly, the algorithm is initialized by a set of random particles which traveling through the search space, during this travel; an evolution of these particles is performed by a PSO coupled with GA to get approximate nondominated solutions. Secondly, to improve the solution quality, dynamic version of pattern search technique is implemented as neighborhood search engine where it intends to explore the less-crowded area in the current archive to possibly obtain more nondominated solutions. Also In order to study the algorithm performance, the effect of change of the most significant parameters of the proposed approach was studied. The proposed approach is carried out on the standard IEEE 30-bus 6-generator test system. The results demonstrate the capabilities of the proposed approach to generate true and well-distributed Pareto optimal nondominated solutions of the multiobjective RPC. Also the results declare that it is quite difficult to find fixed values for these significant parameters, thus we recommend to develop dynamic version of the proposed approach using any monitoring algorithm.

Keywords:

Multiobjective optimization, reactive power compensation, particle swarm Optimization and local search

* E-Mail: mohammed_shorbagy@yahoo.com

1. Introduction:

Reactive Power Compensation (RPC) in power systems is a very important issue in the expansion planning and operation of power systems because it leads to increased transmission capability, reduced losses and improved power factor using shunt capacitors that have been very commonly installed in transmission and distribution networks [1,2]. By applying capacitors adjacent to loads, several advantages are obtained some of them are [3,4,5]:

- improved power factor,
- reduced transmission losses,
- increased transmission capability,
- improved voltage control,
- improved power quality.

Achievement of these items depends mainly on an adequate allocation of shunt capacitor banks. Thus, this problem can be stated as the determination of the locations and the capacities of new sources of reactive power, searching simultaneously to accomplish the following goals:

A good bus tension profile: the quality of service is directly related to the difference between the effective and the nominal bus voltage,

Minimization of transmission losses: active power transmission losses can be directly translated into monetary losses since they are the main component in the difference between the generated power and the consumed power,

Minimization of the amount of reactive compensation installed: although shunt capacitor compensation generally provides the most economical reactive power source for voltage control, heavy use of these devices could lead to the reduction of stability margins and poor voltage regulation [6,7].

Traditionally, this problem is addressed as a single objective optimization problem (SOP) [8,9,10-14]. A Single-objective Optimization Algorithms (SOA) usually provides a unique optimal solution. Typically, the objective function is formulated as a linear combination of several factors such as investment or transmission losses, that are subject to operational constrains such as reliability and voltage profile. These factors that are considered as the optimization objectives usually are contradictory, making very difficult to find the right linear combination. Practically most problems have more than one objective to be optimized, e.g. RPC problem requires the optimization of: investment, power losses, and voltage profile. The objectives are usually contradictory. Accordingly a single objective optimization algorithm will not be preferable to solve the RPC problem. Considering this situation, Multiobjective Optimization Algorithms (MOA) were proposed to optimize independent and simultaneously several objectives [15,16,17,18,19]. Therefore, a MOA usually provides a whole set of optimal tradeoff solutions known as Pareto set. The Pareto set gives the engineer the opportunity to

consider more options before making a final decision.

Local search techniques have long been used to attack many optimization problems [20,21]. The basic idea is to start from an initial solution and to search for successive improvements by examining neighboring solutions. The local search used in this paper is based on a dynamic version of pattern search technique. Pattern search technique is a popular paradigm in Direct Search (DS) methods [22]. DS methods are evolutionary algorithms used to solve constrained optimization problems. DS methods, as opposed to more standard optimization methods, are often called derivative-free as they do not require any information about the gradient or higher derivatives of the objective function to search for an optimal solution. Therefore direct search methods may very well be used to solve non-continuous, nondifferentiable and multimodal (i.e. multiple local optima) optimization problems.

This paper presents a new approach based on hybrid particle swarm optimization PSO. It integrates the merits of two heuristic optimization techniques GA and PSO. In order to improve the solution quality, we implement modified local search algorithm. Also In order to study the algorithm performance, the effect of change of the most significant parameters of the proposed approach was studied. Finally, The standard IEEE 30-bus 6-genrator test system then used to verify the validity of the proposed approach.

This paper is organized as follows. In section 2, MOO is described. Section 3, provides a multi-objective formulation of RPC Problem. In section 4, the proposed algorithm is presented. Results are presented in section 5. Finally, section 7 gives a brief conclusion about this study.

2. Multiobjective optimization:

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

$$\text{Minimize} \quad \left\{ f_1(X), f_2(X), \ldots, f_m(X) \right\}$$
$$\text{subject to} \quad g(X) \leq 0, \tag{1}$$

Where X is vector of decision variables; $f_i(X)$ is the ith objective function; and $g(X)$ is constraint vector. A decision vector X is said to dominate a decision vector Y (also written as $X \succ Y$) iff:

$$f_i(X) \leq f_i(Y) \quad \textit{for all} \quad i \in \{1, 2, \ldots, m\} \tag{2}$$

and

$$f_i(X) < f_i(Y) \quad \textit{for at least one} \quad i \in \{1, 2, ..., m\} \tag{3}$$

All decision vectors that are not dominated by any other decision vector are called nondominated or Pareto-optimal. These are solutions for which no objective can be improved without detracting from at least one other objective.

3. *Multiobjective formulation of RPC problem*:

The following assumptions are considered in the formulation of the problem:
A shunt-capacitor bank cost per MVAr is the same for all busbars of the power system,
Power system is considered only at peak load.
Based on these considerations [22,23], three objective functions $f_i(\bullet)$ (to be minimized) have been identified for the present work: $f_1(\bullet)$ and $f_2(\bullet)$ are related to investment and transmission losses, while $f_3(\bullet)$ are related to quality of service. The objective functions to be considered are:

A. F_1 **Investment in reactive compensation devices:**

$$F_1 = \sum_{i=1}^{n} \alpha_i B_i \quad s.t. \quad \begin{cases} 0 \leq F_1 - \\ 0 \leq B_i \leq _{max} \end{cases} \tag{4}$$

Where for simplicity the cost per MVAr is taken as unity $\alpha = 1$, n is the number of buses in the power system; F_1 is the total required compensation; F_{1MAX} is the maximum amount available for investment; B_i is the compensation at busbar i measured in MVAr and $B_{i\,max}$ is the maximum compensation allowed at a particular bus of the system.

B. F_2 **Active power losses:**

$$F_2 = P_g - P_l \geq 0 \quad s.t. \quad P_{g\,min} \leq P_g \leq P_{g\,max} \tag{5}$$

Where: F_2 is the total transmission active losses of the power system in MW; P_g is the total active power generated in MW and P_l is the total load of the system in MW.

C. F_3 *Average voltage deviation:*

$$F_3 = \frac{\sum_{i=1}^{n}\left|V_i - V_i^*\right|}{n} \quad s.t. \quad V_{i\min} \le V_i \le V_{i\max} \tag{6}$$

Where: F_3 is the per unit (pu) average voltage difference; V is the actual voltage at busbar i (pu) and V_i^* is the desired voltage at busbar i (pu).

In summary, the optimization problem to be solved is the following:

Minimize $F = \begin{bmatrix} F_1 & F_2 & F_3 \end{bmatrix}$

Where

$$F = \begin{bmatrix} \sum_{i=1}^{n} B_i{}_1 & P_g - P_l & \frac{\sum_{i=1}^{n}\left|V_i - V_i^*\right|}{n} \end{bmatrix} \tag{7}$$

Subject to $0 \le F_1 \le F_{1\max}$, $0 \le B_i \le B_{i\max}$, $P_{g\min} \le P_g \le P_{g\max}$, $V_{i\min} \le V_i \le V_{i\max}$ and the load flow equations [24]:

$$\Delta P_p = P_{Gp} - \blacksquare\cos\left(\delta_p - \delta_q - \theta_{pq}\right) \tag{8}$$

$$\Delta Q_p = Q_{Gp} - \blacksquare\sin\left(\delta_p - \delta_q - \theta_{pq}\right) \tag{9}$$

Where, P_{Gp}, Q_{Gp} are the real and reactive power generations at bus p; P_{cp}, Q_{cp} the real and reactive power demands at bus p ; V_p, the voltage magnitude at bus P; V_q, the voltage magnitude at bus q; δ_p , the voltage angle at bus p; δ_q; the voltage angle at bus q; Y_{pq}, the admittance magnitude; θ_{pq}, the admittance angle; N_B, the total number of buses; $p = 1, 2, ..., N_B$ and $q = 1, 2, ..., N_B$.

The load flow equations reflect the physics of the power system as well as the desired voltage set points throughout the system. The physics of the power system are enforced through the power flow equations which require that the net injection of real and reactive power at each bus sum to zero.

To represent the amount of reactive compensation to be allocated at each busbar i , a

decision vector B [25], is used to indicate the size of each reactive bank in the power system, i.e.:

$$\boldsymbol{B} = \left[\boldsymbol{B_1}, \boldsymbol{B_2}, \ldots, \boldsymbol{B_n} \right], \boldsymbol{B_i} \in \boldsymbol{R}, \left| \boldsymbol{B_i} \right| \leq \boldsymbol{B_{i\max}} \tag{10}$$

Thus RPC is a complex combinatorial optimization problem involving multiple nonlinear functions having multiple local minima, which may be ill-defined and nonlinear with discontinuous constraint, which lead to non-convex Pareto-optimal front [25,26].

4. The Proposed approach:

In this section we present a novel optimization algorithm to solve the RPC problem formulated in the previous section. The proposed methodology introduces a hybrid approach combining GAs and PSO to improve the performance of each algorithm. Also, to improve the solution quality we implement LS technique as neighborhood search engine where it intends to explore the less-crowded area in the current archive to possibly obtain more nondominated solutions nearby (i.e. that we search near every solution by LS technique to obtain a new solution best than current one or nondominated with it and therefore the less-crowded area will be discovered automatically). The description diagram of the proposed algorithm is shown in Figure 1, and it is described as follows:
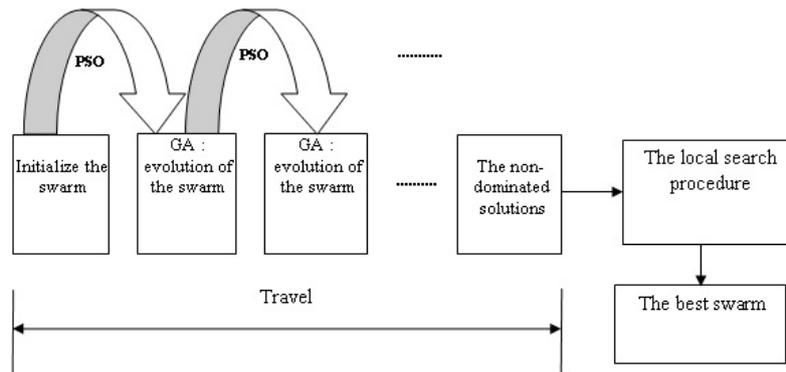
Figure (1): The description diagram of the proposed algorithm

A. PSO Stage:

In this stage, we implement PSO as follows

Step 1: Initialize parameters for PSO, initialize randomly *N* particles with position $X_i^{t=0}$

with velocities $V_i^{t=0}$ where t is the time counter and $i = 1,....,N$.

Step 2: Identify the local set $\left(L_i^{t=0}\right)$ for each particle as $L_i^{t=0} \sqcup \left\{X_i^{t=0} \middle| i = 1,...,N\right\}$. Also, identify the local preferred element $\left(LP_i^{t=0} \subset L_i^{t=0}\right)$ of the i-th particle as $\forall$ particle i $\exists LP_i^{t=0} \sqcup \left\{X_i^{t=0}\right\}$.

Step 3: Collect all local sets $\left\{L_i^{t=0}\right\} \forall i = 1,...,N$ in a pool C such that $C = \bigcup_{i=1}^{N} L_i^{t=0}$.

Step 4: Define a global set $G^{t=0} \sqcup ND(C)$, where we assume that the function $ND(\bullet)$ can get all nondominated solutions.

Step 5: In the objective space, The distance between $X_i^{t=0} \forall i = 1,...,N$ and the members in $G^{t=0}$ are measured using the Euclidean distance, where the distance between any two d-dimensional points $\vec{x}_i$ and $\vec{x}_j$ is given by

$$\mathbf{d}\left(x_i, x_j\right) \quad \|x_i \quad x_j\|_2 \quad \sqrt{\sum_{p=1}^{d}\left(x_{i,p} \quad x_{j,p}\right)^2} \tag{11}$$

The nearest member in G^t to the i-th particle set as the global preferred element GP_i^t .

Step 6: Set the external set $E^{t=0}$ equal to $G^{t=0}$.

In an example, let we have 6 particles initially located as shown in Figure 2.
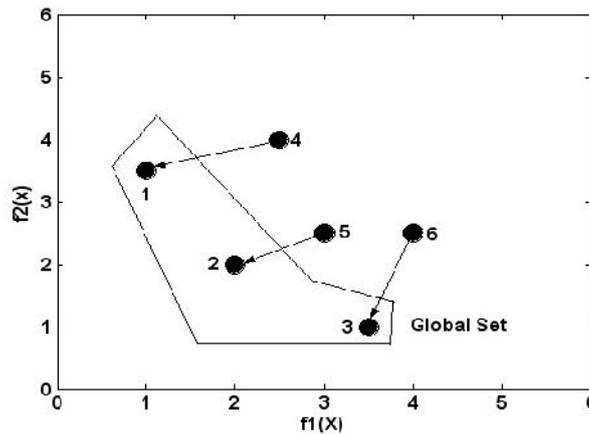
Figure (2): *Location of initially 6 particles*

Define the local set $\forall$ particle i $\exists L_i^{t=0} \sqcup \left\{X_i^{t=0} \middle| i = 1,...,N\right\}$

$$L_1^{t=0} = \{x_1\}, L_2^{t=0} = \{x_2\}, L_3^{t=0} = \{x_3\},$$

$$L_4^{t=0} = \{x_4\}, L_5^{t=0} = \{x_5\}, L_6^{t=0} = \{x_6\}$$

Define the local preferred element $\forall$particle i $\exists LP_i^{t=0} \sqcup \{X_i^{t=0}\}$

$$LP_1^{t=0} = \{x_1\}, LP_2^{t=0} = \{x_2\}, LP_3^{t=0} = \{x_3\},$$

$$LP_4^{t=0} = \{x_4\}, LP_5^{t=0} = \{x_5\}, LP_6^{t=0} = \{x_6\}$$

Construct a pool C such that $C = \bigcup_{i=1}^{N} L_i^{t=0} \Rrightarrow C = \bigcup_{i=1}^{6} L_i^{t=0} = \{x_1, x_2, x_3, x_4, x_5, x_6\}$

Define a global set $G^{t=0} \sqcup ND(C) \Rrightarrow G^{t=0} = ND(C) = \{x_1, x_2, x_3\}$

Identify the global preferred element GP_i^{t} $\forall$ particle i

$$GP_1^{t=0} = \{x_1\}, GP_2^{t=0} = \{x_2\}, GP_3^{t=0} = \{x_3\},$$

$$GP_4^{t=0} = \{x_1\}, GP_5^{t=0} = \{x_2\}, GP_6^{t=0} = \{x_3\}$$

Define External set $E^{t=0} = G^{t=0} \Rrightarrow E^{t=0} = G^{t=0} = \{x_1, x_2, x_3\}$

Step 7: *Update particles:* Update the velocity v_i^t and position x_i^t of each particle to get new velocity v_i^{t+1} and position x_i^{t+1} according to the following equations:

$$v_i^{t+1} = wv_i^t + c_1 r_1 \left(LP_i^t - x_i^t \right) + c_2 r_2 \left(GP_i^t - x_i^t \right) \tag{12}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{13}$$

where $i = 1, 2, \ldots\ldots, N$, and N is the size of the population; w is the inertia weight; c_1 and c_2 are two positive constants, called the cognitive and social parameter respectively; r_1 and r_2 are random numbers uniformly distributed within the range [0,1].

Step 8: *Evolution of particles:* To restrict velocity and control it, we present a modified constriction factor (i.e., dynamic constriction factor) to keep the feasibility of the particles. e.g., Figure 3 shows the movement of the particle i through the search space with and without modified constriction factor. Where the particle i start at position x_i^t with velocity v_i^t in the feasible space, the new position x_i^{t+1} depends on velocity v_i^{t+1}. Then, v_i^{t+1} makes the particle to lose its feasibility, so we introduce a modified factor χ such that:
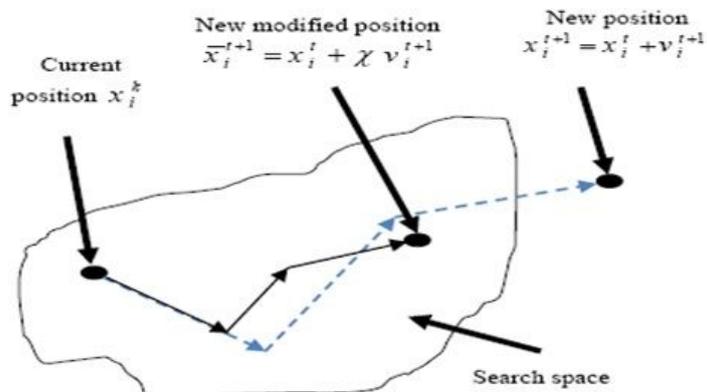
Figure (3): *The movement of the particle i through search space*

$$= \frac{\blacksquare}{|-2 - \tau -|} \tag{14}$$

Where, τ is the age of the infeasible particle (i.e., How long it's still unfeasible) and it is increased with the number of failed trial to keep the feasibility of the particle. The new modified position of the particle is computed as:

$$\bar{x}_i^{t+1} = x_i^t + \chi \ v_i^{t+1} \tag{15}$$

For each particle we check its feasibility, if it is infeasible, we implement χ parameter to control its position and velocity according to algorithm 1.

Algorithm 1: Evolution of particles

$\text{input}\left(x_i^t, v_i^{t+1}, x_i^{t+1}\right)$

while

   x_i^{t+1} is unfeasible $|$ number of trial not satisfied

   generate $\bar{x}_i^{t+1} = x_i^t + \chi \ v_i^{t+1}$

   $x_i^{t+1} = \bar{x}_i^{t+1}$

end

$\text{output}\left(v_i^{t+1}, x_i^{t+1}\right)$

Figure (4): *Algorithm 1: Evolution of particles*

Step 9: Update local set L_i^t to get $L_i^{t+1} \forall i = 1,....N$: The new position of each particle X_i^{t+1} is added to L_i^t to form L_i^{t+1} which is updated according to algorithm 2

Algorithm 2: Update local set L_i^t

Input $\left(L_i^t, X_i^{t+1} \right)$

    If $\exists X \in L_i^t \mid X \succ X_i^{t+1}$ then

        $L_i^{t+1} = L_i^t$

    Else if $\exists X \in L_i^t \wedge X_i^{t+1} \succ X$ then

        $L_i^{t+1} = L_i^t \bigcup \left\{ X_i^{t+1} \right\} / \left\{ X \right\}$

    Else if $\nexists X \in L_i^t \mid X \succ X_i^{t+1}$ then

        $L_i^{t+1} = L_i^t \bigcup \left\{ X_i^{t+1} \right\}$

    End

Output $\left(L_i^{t+1} \right)$

Figure (5): *Algorithm 2: Evolution of particles*

Step 10: Update global set G : $G^{t+1} = ND \left(\bigcup_{i=1}^{N} L_i^{t+1} \right)$ which contain all nondominated solution of $\bigcup_{i=1}^{N} L_i^{t+1}$.

Step 11: Update external set E^t ***:*** Copy the members of G^{t+1} to E^t and apply dominance criteria to remove all dominated solution from E^t (i.e., each member of G^{t+1} has three probabilities as in algorithm 3)

Step 12: Update local preferred element LP_i^{t+1} ***and global preferred element*** GP_i^{t+1} ***for each particle:*** In the objective space, The distance between $X_i^{t+1} \forall i = 1,...,N$ and members in L_i^{t+1} are measured using Euclidean distance. The nearest member in L_i^{t+1} to the *i-th* particle set as LP_i^{t+1}. Also, The distance between $X_i^{t+1} \forall i = 1,...,N$ and the members in G^{t+1} are measured using Euclidean distance. The nearest member in G^{t+1} to the *i-th* particle set as the global preferred GP_i^{t+1}.

Algorithm 3: Update external set E^t

$\text{Input}\left(E^t, X \in G^{t+1}\right)$

If $\exists\ Y \in E^t \mid Y \succ X$ then

   $E^t = E^t$

Else if $\exists\ Y \in E^t \wedge X \succ Y$ then

   $E^t = E^t \bigcup \{X\} / \{Y\}$

Else if $\nexists\ Y \in E^t \mid Y \succ X$ then

   $E^t = E^t \bigcup \{X\}$

End

$\text{Output}\left(E^t\right)$
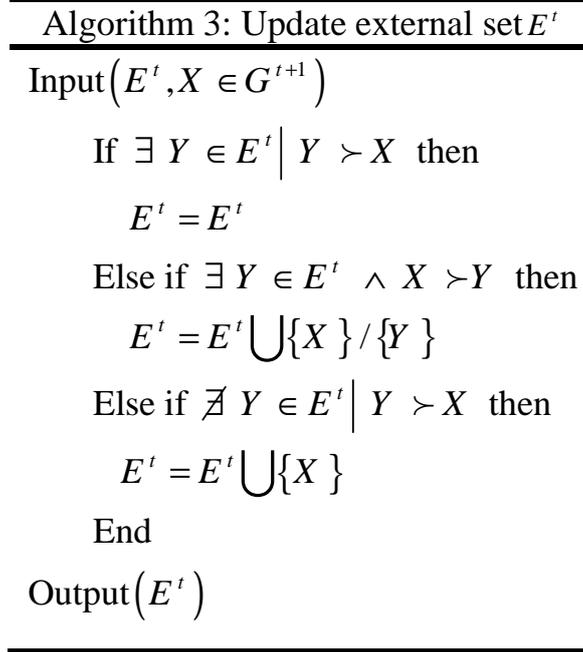
Figure (6): Algorithm 3: Update external set

B. GA Stage:

In this subsection, we describe the procedure of GA.
Step 1: Initialize parameters for GA.
Step 2: Evaluation & Ranking: A way to transform the values of objective functions to the fitness function of each string in the genotype world is to combine the *m* objective functions into a scalar function as follows:

$$f(x) = w_1 f_1(x) + w_2 f_2(x) + \dots + w_m f_m(x) \tag{16}$$

Where $f(x)$ is the fitness function of x and $w_1,\dots,w_m$ are non-negative weights which determined as follows:

$$w_i = random_i \bigg/ \sum_{j=1}^{m} random_i, \forall\ i = 1,\dots,m \tag{17}$$

Where $random_1, random_2,\dots, random_m$, are non-negative random integers. Then rank them on the basis of the fitness values.
Step 3: Selection: Selection is an operator to select two parent strings for generating new strings (*i.e.*, offspring). In the selection, a selection probability $P_s(x_i)$ of each

string x based on the linear scaling is defined by the roulette wheel selection as follows:

$$P_s(x_i) = \frac{f(x_i) \ldots}{\sum_{j=1}^{N}\{f(\ldots (x_\psi)\}} \ldots, 2, \ldots, N \tag{18}$$

Where $f_{min}(x_\psi)$ is the minimum fitness value (*i.e.*, the worst fitness value) in the current population ψ. According to this selection probability, a pair of parent strings are selected from the current population ψ.

Step 4: Crossover: Crossover is an operator to generate new strings (*i.e.*, offspring) from parent strings according to the crossover probability (*Pc*). Various crossover operators have been proposed for GAs [27,28]. In the proposed approach we implement single point crossover.

Step 5: Mutation: Mutation is an operator to change elements in a string which is generated by a crossover operator. Such a mutation operator can be viewed as a transition from a current solution to its neighborhood solution in local search algorithms [29] according to the mutation probability (*Pm*). In the proposed approach, we mutate each variable in a string $x_i \in [a_i, b_i]$ with *Pm* by addition of small random values according to the equations below:

$$x_i' = \begin{cases} x_i \ldots x_i) & \text{if} \quad \delta = 0 \\ x_i \ldots - a_i) & \text{if} \quad \delta = 1 \end{cases} \tag{19}$$

$$\Delta(t, y) = y\left[1 - r^{\ldots /t_{max})^\beta}\right] \tag{20}$$

Where r is a random number $r \in [0,1]$, t_{max} is the maximum number of generations, and β is a positive constant chosen arbitrarily.

Step 6: Elitist strategy (Replacing): Randomly remove a string from the current population and add the best string in the previous population to the current one.

Step 7: Repairing: Repair the infeasible individuals of the population to be feasible. The idea of this technique is to separate any feasible individuals in a population from those that are infeasible by repairing infeasible individuals. This approach co-evolves the population of infeasible individuals until they become feasible, the reader is referred to [30].

C. LS Stage:

The local search phase is implemented as a dynamic version of pattern search technique. Pattern search technique is a popular paradigm in Direct Search (DS) methods. DS methods are evolutionary algorithms used to solve constrained optimization problems. DS methods, as opposed to more standard optimization methods, are often called derivative-free as they do not require any information about the gradient or higher derivatives of the objective function to search for an optimal solution. Therefore direct search methods may very well be used to solve non-continuous, nondifferentiable and multimodal (i.e. multiple local optima) optimization problems. This study examines the usefulness of a dynamic version of pattern search technique to improve the solution quality of MOPs. The search procedure looks for the best solution "near" another solution by repeatedly making small changes to a starting solution until no further improved solutions can be found.

The local search is started by loading the Pareto solutions for a given MOPs. At iteration t, we have an iterate $x_t \in E^t$, where the changes on the values for each dimension $(i = 1, 2, .., n)$ can be implemented as

$$\Delta(t) = R\left(1 - r^{\left(1 - k/k_{\max}\right)}\right) \tag{21}$$

Where r is the random number in the range [0,1]; k is number of trial (*s.t.,* $k = 1,...,k_{\max}$) to obtain preferred solution than x_t; R is the search radius.

Let $e_i; (i = 1, 2, ..., n)$, denote the standard unit basis vectors. We successively look at the points $x_{new} = x_t \pm \Delta(t)e_i; (i = 1,...,n)$, until we find x_{new} for which $f_j(x_{new}) \succ f_j(x_t)$ for at least one objective $f_j, j \in m$. Then we update the Pareto solutions by nondominated ones and the dominated ones are removed. If we find no x_{new} such that $f_j(x_{new}) \succ f_j(x_t)$, we reduce the radius R and repeat the local search again. This situation is represented in Figure 7 for the case in R^2.

This local search scheme is implemented on all nondominated solutions in E^t to get the true Pareto optimal solution and to explore the less-crowded area in the external archive. The pseudo code of the proposed algorithm showing in Figure 8.
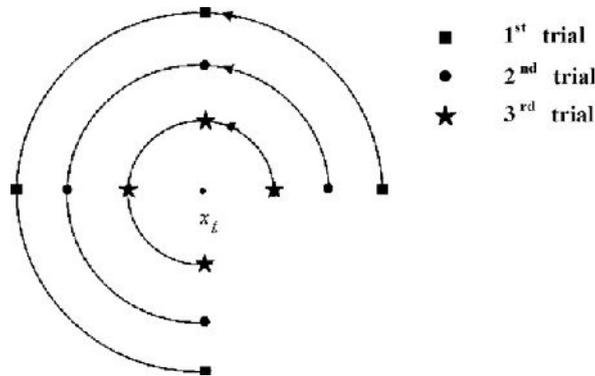
Figure (7): *Mechanism of dynamic pattern search in* R^2

Initialize parameters for PSO and GA.

Identify local set $L_i^{t=0} \triangleq \{X_i^{t=0}\} \forall i = 1,...,N$ and $LP_i^{t=0} \triangleq \{X_i^{t=0}\} \forall i = 1,...,N$.

Identify global set $G^{t=0} = ND\left(\bigcup_{i=1}^{N} L_i^{t=0}\right)$

The nearest member in $G^{t=0}$ to the i-th particle is $GP_i^{t=0}$

Set the external set $E^{t=0} = G^{t=0}$
While travel not completed.
    PSO algorithm.
      While sub-travel not completed.
        Generate v_i^{t+1} and x_i^{t+1} .
        Evolve the unfeasible particles until they can be feasible.
        Update $L_i^{t+1} \forall i = 1,...,N$ and G^{t+1}
        Identify LP_i^{t+1} and GP_i^{t+1}
        Update E^t .
    End while.
    GA algorithm.
    While evolution not completed.
      Evaluation.
      Keep the best.
      GA operators.
      Repair.
    End while.
End travel.
Local search phase by adaptive pattern search
$t = 0$, load the Pareto solutions
while $t \le T$ **do**
    for $j = 1 : n$ **do**
        $Pareto(i , j)_1 = Pareto(i , j) + \Lambda(t)$ {generate new search point}
      If $Pareto(i ,:)_1 \succ Pareto(i ,:)$ **do**
          $Pareto(i ,:) = Pareto(i ,:)_1$; break while {replace old solution}
      end if
    end for , $t = t + 1$
end while
 Output $\rightarrow$ Pareto
End

Figure (8): *Mechanism of dynamic pattern search in* R^2

5. Implementation of the proposed approach:

The described methodology is applied to the standard IEEE 30-bus 6-generator test system to investigate the effectiveness of the proposed approach. The detailed data for this system are given in [31]. Table 1 lists the parameter setting used in the algorithm for all runs.

Table (1): The Proposed Approach Parameter

Cognitive parameter	2.8
Social parameter	1.3
Inertia weight	0.6
Crossover probability	0.95
Mutation probability	0.01
Selection operator	roulette wheel selection
Crossover operator	Single point crossover
Mutation operator	Real-value mutation
PSO iteration	5
GA generation	5

A. Results:

Figure 9 shows well-distributed Pareto optimal nondominated solutions obtained by the proposed algorithm after 200 generations. On the other hand, Table 2 shows 29 Pareto optimal points obtained by proposed algorithm.
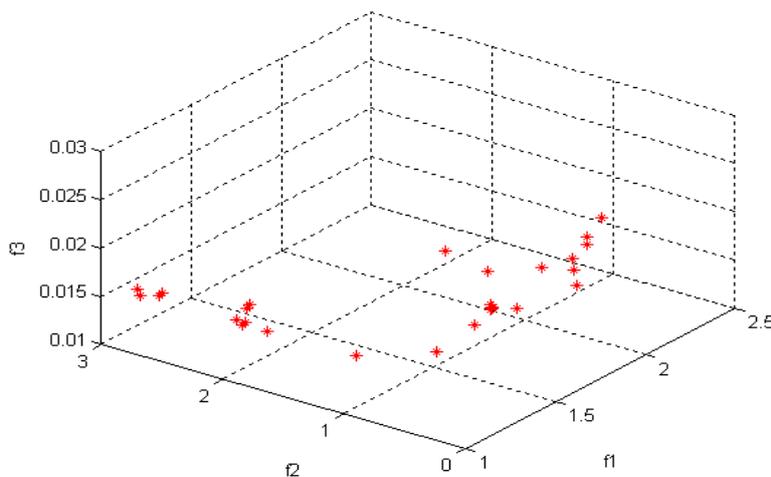
Figure (9): Pareto optimal front of the RPC problem

Out of the Pareto-optimal set Table 3 shows the values of $f_1(\bullet)$, $f_2(\bullet)$ and $f_3(\bullet)$ in the three cases:

Case 1: Corresponding to minimum amount of reactive compensation devices.
Case 2: Corresponding to minimum amount of active power losses.
Case 3: Corresponding to minimum amount of average voltage deviation.

Table (2): *The Pareto optimal points obtained by the proposed algorithm*

$f_1(\bullet)$	$f_2(\bullet)$	$f_3(\bullet)$
1.7667	0.019393	0.026343
1.7318	0.20475	0.021874
1.7461	0.19506	0.01898
1.6824	0.38219	0.0208
1.8048	0.19226	0.023471
1.8048	0.19457	0.022631
1.7376	0.19506	0.020647
1.2727	0.64324	0.015005
1.0942	2.8501	0.015312
1.1257	2.03	0.014349
1.1928	2.6405	0.01572
1.097	1.963	0.015049
1.1101	2.8447	0.014567
1.1	1.9291	0.01706
1.1007	1.0507	0.014921
1.097	2.0325	0.01502
1.1103	1.797	0.014567
1.0928	2.6577	0.015345
1.5668	0.6271	0.016723
1.5643	0.62755	0.016692
1.0993	1.9493	0.01653
1.6523	0.54007	0.016278
1.5694	0.62663	0.016824
1.5252	0.60126	0.021104
1.4752	0.62976	0.015917
1.5586	0.62042	0.017193
1.5652	0.62738	0.016704
2.0039	1.6488	0.014887
2.0028	1.6492	0.01483

Table (3): *Values of $f_1(\bullet)$, $f_2(\bullet)$, and $f_3(\bullet)$ in three cases*

	Cases 1	Cases 2	Cases 3
$f_1(\bullet)$	1.0928	1.7667	1.1257
$f_2(\bullet)$	2.6577	0.019393	2.03
$f_3(\bullet)$	0.015345	0.026343	0.014349

A. Parameter analysis:

In this section, we show the effect of changing the parameter of PSO (inertia weight w, cognitive learning factor c_1 and social learning factor c_2) on the results of the application of MORPC. Previously, Shi and Eberhart [32] introduced constant inertia weight and linear inertia weight varying usually between 0.8 and 0.4 where it in the first iteration 0.8 and decreasing during the process of run to be 0.4 in the last iteration. On the other hand, Kennedy [33] asserted that the sum of the cognitive and social values c_1 and c_2 should approximately equal 4.0. For constriction, Carlisle and Dozier [34] have shown that it is advantageous to adjust the cognitive/social ratio to favor cognitive learning (an individualistic swarm). They report that values of 2.8 and 1.3 respectively for the cognitive and social components yield the best performance for the test set they consider.

In order to analyze the impact of the parameters on our approach, we considered several configurations, and performed a comprehensive number of runs. For the inertia weight (w) parameter we make studies on the constant values of it where, $w = \{0.4, 0.5, 0.6, 0.7, 0.8\}$ and for c_1 and c_2 we varying the social value c_2 between 1.1 and 1.5, with the cognitive value calculated in each case as $c_1 = 4.1 - c_2$. Then we determine the values of $f_1(\bullet)$, $f_2(\bullet)$, and $f_3(\bullet)$ in the three cases corresponding to minimum amount of them and draw the change of the parameter versus the minimum value of each objective function.

- **Effect of Changing Inertia Weight w**

In this subsection we study the effect of changing w on the results obtained by our approach and we study five states from $w = 0.4$ to $w = 0.8$. Table 4, lists the values of $f_1(\bullet)$, $f_2(\bullet)$, and $f_3(\bullet)$ in the three cases for different values of inertia weight w.

Table (4): *Values of $f_1(\bullet)$, $f_2(\bullet)$, and $f_3(\bullet)$ in three cases for different values of inertia weight w*

	Weight w	Cases 1	Cases 2	Cases 3
$f_1(\bullet)$	$w = 0.4$	1.6824	1.8048	1.7461
$f_2(\bullet)$	$w = 0.4$	0.19506	0.19226	0.38219
$f_3(\bullet)$	$w = 0.4$	0.01898	0.023471	0.015049
$f_1(\bullet)$	$w = 0.5$	1.757	2.0028	2.8895
$f_2(\bullet)$	$w = 0.5$	1.6492	0.27701	1.1195
$f_3(\bullet)$	$w = 0.5$	0.01483	0.017264	0.014537
$f_1(\bullet)$	$w = 0.6$	1.5586	1.5652	2.0039
$f_2(\bullet)$	$w = 0.6$	1.6488	0.62042	0.62735
$f_3(\bullet)$	$w = 0.6$	0.016704	0.017193	0.014887
$f_1(\bullet)$	$w = 0.7$	1.4752	1.5252	1.5694
$f_2(\bullet)$	$w = 0.7$	0.62976	0.60126	0.62663
$f_3(\bullet)$	$w = 0.7$	0.016824	0.021104	0.015917
$f_1(\bullet)$	$w = 0.8$	1.0993	1.5643	1.6523
$f_2(\bullet)$	$w = 0.8$	0.62755	0.54007	1.9493
$f_3(\bullet)$	$w = 0.8$	0.016692	0.016278	0.01653

Figure 10 showing the minimum value variation of each objective function with increasing of inertia weight (w) from 0.4 to 0.8. From the Figure we find that the change of the inertia weight w from 0.4 to 0.5 making improvement to the value of the objective function f_3 but the value of the objective function f_1 is worse. After that changing of w from 0.5 to 0.8 making improvement to the value of the objective function f_1 but the value of the objective function f$_3$ is worse. For the objective function f_2 we find that its value get worse when w change from 0.4 to 0.6 and its value improved when w change from 0.6 to 0.8. In general, we can say that the change of the inertia weight parameter (w) improves one of the objective functions but at the same moment making the other objectives are worse.
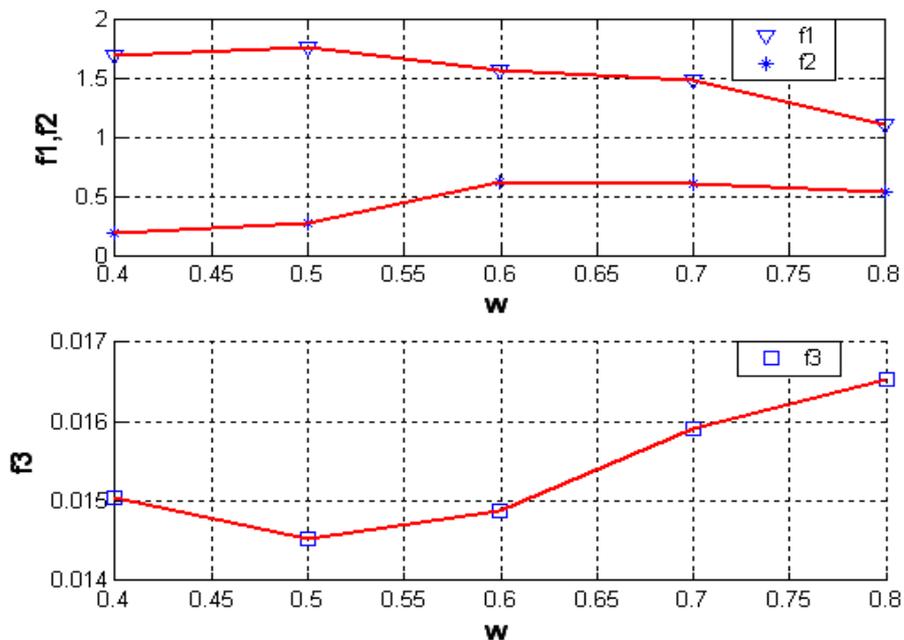
Figure (10): *The minimum value variation of the objective functions (f₁, f₂ and f₃) with increasing of w from 0.4 to 0.8*

- **Effect of Changing c_1 and c_2**

In this subsection we study the effect of changing c_1 and c_2 on the results of the application of MORPC and study five states where the social value c_2 between 1.1 and 1.5, with the cognitive value calculated in each case as $c_1 = 4.1 - c_2$. Table 5, lists the values of $f_1(\bullet)$, $f_2(\bullet)$, and $f_3(\bullet)$ in the three cases for different values of inertia weight c_2.

Figure 11 shows the minimum value variation of each objective function with increasing of social learning factor (c_2) from 1.1 to 1.5. From the Figure we find that the change of the parameter c_2 from 1.1 to 1.3 making the value of the objective function f_1 worse and when it change from 1.3 to 1.5 the value of f_1 is improved. For the objective function f_2, we find that its value oscillating; when the value of c_2 change from 1.1 to 1.2 its value is improved and when change from 1.2 to 1.3 is worse and improved again when c_2 change from 1.3 to 1.5. Also, The value of the objective function f_3 is worse when c_2 change from 1.1 to 1.2 but its values is improved when c_2 change from 1.2 to 1.5.

Table (5): *Values of $f_1(\bullet)$, $f_2(\bullet)$, and $f_3(\bullet)$ in three cases for different values of social value c_2*

	Social value c_2	Cases 1	Cases 2	Cases 3
$f_1(\bullet)$	$c_2 = 1.1$	1.097	1.5668	1.1103
$f_2(\bullet)$	$c_2 = 1.1$	2.0325	0.6271	1.797
$f_3(\bullet)$	$c_2 = 1.1$	0.01502	0.016723	0.014567
$f_1(\bullet)$	$c_2 = 1.2$	1.1648	1.4307	1.3706
$f_2(\bullet)$	$c_2 = 1.2$	0.61493	0.52814	0.83764
$f_3(\bullet)$	$c_2 = 1.2$	0.01775	0.021472	0.017476
$f_1(\bullet)$	$c_2 = 1.3$	1.2338	1.2579	1.2585
$f_2(\bullet)$	$c_2 = 1.3$	0.80845	0.80654	0.81007
$f_3(\bullet)$	$c_2 = 1.3$	0.016667	0.016372	0.016357
$f_1(\bullet)$	$c_2 = 1.4$	1.2127	1.2608	1.2406
$f_2(\bullet)$	$c_2 = 1.4$	0.81061	0.78001	0.8061
$f_3(\bullet)$	$c_2 = 1.4$	0.016438	0.016943	0.016173
$f_1(\bullet)$	$c_2 = 1.5$	1.0996	1.121	1.1064
$f_2(\bullet)$	$c_2 = 1.5$	2.7748	0.73438	2.5392
$f_3(\bullet)$	$c_2 = 1.5$	0.013961	0.027328	0.013683
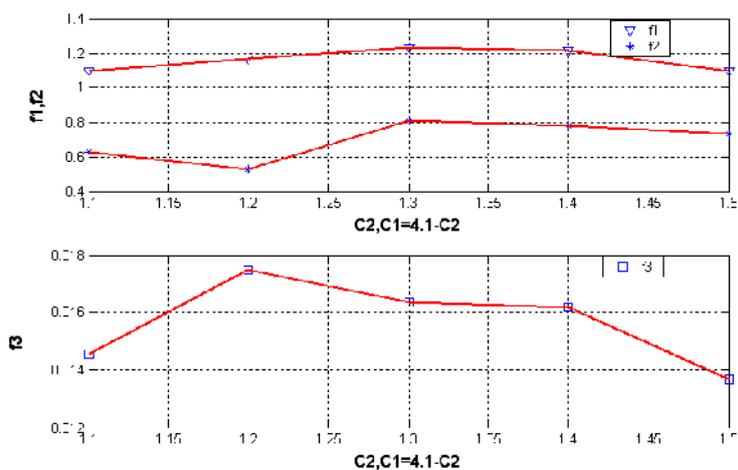
Figure (11): *The minimum value variation of the objective functions (f1, f2 and f3) with increasing of c_2 from 1.1 to 1.5*

Also, we can say that the change of social learning factor (c_2) may improves one of the objective functions values but at the same moment making the other objectives are worse. In general, the change of the parameter may give the DM Other solutions of its problem but these solution may be (by the concept of dominance) not acceptable.

The proposed approach has been used to increase the solution quality by combining the two merits of two heuristic algorithms. However, the goal is not only to increase the solution, but also to generate a representative subset, which maintains the characteristics of the general set and take the solution diversity into consideration. On the other hand, classical techniques aim to give single point at each iteration of problem solving by converting the multiobjective problem to a single objective problem by linear combination of different objectives as a weighted sum. On the contrary, the proposed approach is a heuristics-based multiobjective optimization technique where, it uses a population of solutions in their search, multiple Pareto-optimal solutions can, in principle, be found in one single run. Another advantage is the reality of using the proposed approach to handle complex problems of realistic dimensions has been approved due to procedure simplicity.

6. Conclusions:

The reactive power compensation problem formulated as multiobjective optimization problem with competing amount of reactive compensation devices, active power losses and average voltage deviation is solved in this paper using a combination of GA and PSO. Our approach integrates the merits of both GA and PSO. In order to improve the solution quality, we implement LS technique as neighborhood search engine where it intends to explore the less-crowded area in the current archive to possibly obtain more nondominated solutions. The algorithm have an external archive to keep track of all the feasible solutions found during the optimization and therefore do not have any restrictions on the number of the Pareto-optimal solutions found. The proposed approach is carried out on the standard IEEE 30-bus 6-generator test system. The results demonstrate the capabilities of the proposed approach to generate true and well-distributed Pareto-optimal nondominated solutions of the multiobjective RPC, which is useful in giving a reasonable freedom in choosing compensation devices from the available commercial devices.

By studying the effect of change the parameters w, c_1, and c_2 on the results of the proposed approach, we found that it is quite difficult to find fixed values for the three most significant parameters of our approach (w, c_1, and c_2) because the change of the parameter may improve some of the objective functions and worse at least one other objective.

For further work we intend to develop dynamic approach by coupling the proposed approach with fuzzy logic controller, This controller monitors the variation of the

decision variables during process of the algorithm and modifies the significant parameters to restart the next round of the algorithm. These characteristics will make this approach well suited for finding optimal solutions to the highly multiobjective problems.

References:

[1] P. R. Sujin, T. Ruban Deva Prakash, M. Mary Linda, *Particle Swarm Optimization Based Reactive Power Optimization*, Journal of computing, Vol. 2, No. 1, P. 73-78, 2010.

[2] F. Yoshikazu, *Comparative Studies of Particle Swarm Optimization Techniques for Reactive Power Allocation Planning in Power Systems*, IEEE Transactions on Power and Energy, Vol. 124, No. 5, P. 690-696, 2004.

[3] L. Jikeng, W. Xudong and Z. Weihong, *Reactive Power Optimization Based on Adaptive Immune Algorithm*, Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 Vol. II IMECS 2008, 19-21 March, 2008, Hong Kong.

[4] P. Kundur, *Power System Stability and Control*, New York: Mc Graw-Hill, 1993.

[5] X. Zhang, W. Chen, C. Dai and A. Guo, *Self-adaptive Differential Evolution Algorithm for Reactive Power Optimization*, 2008 Fourth International Conference on Natural Computation October, 18-October, 2008.

[6] S. K. Nandha Kumar and Dr. P. Renuga, *Reactive Power Planning using Real GA Comparison with Evolutionary Programming*, International Journal of Recent Trends in Engineering, Vol. 1, No. 3, May 2009.

[7] D. Thukaram and G. Yesuratnam, *Optimal reactive power dispatch in a large power system with AC - DC and FACTS controllers*, IET Generation, Transmission & Distribution, Vol. 2, No. 1, P. 71-81, 2008.

[8] J. Carlisle, A. El-Keib, D. Boyd and K. Nolan, *A review of capacitor placement techniques on distribution feeders*, in Proc. IEEE 29 Southeastern Symposium on system Theory (SSST'97), 1997.

[9] M. Delfanti, G. Granelli, P. Marannino and M. Montagna, *Optimal capacitor placement using deterministic and genetic algorithms*, IEEE Trans. Power Systems, Vol. 15, No. 3, P. 1041-1046, August 2000.

[10] J. Lin and X. Wang, *Reactive Power Optimization Based on Adaptive Immune Algorithm*, International Journal of Emerging Electric Power Systems, Vol. 10, No. 4, 2009.

[11] Y. Liu, L. Ma and J. Zhang, *Reactive power optimization by GA/SA/TS combined algorithms*, International Journal of Electrical Power & Energy Systems, Vol. 24, No. 9, P. 765-769, November 2002.

[12] J. Lu, L. Zhang, H. Yang and J. Du, *Improved strategy of particle swarm optimisation algorithm for reactive power optimization*, International Journal of Bio-Inspired Computation, Vol. 2, No. 1, P. 27 – 33, 2010.

[13] J. T. Ma and L. L. Lai, *Evolutionary programming approach to reactive power planning*, IEEE proc. Of Generation Transmission and Distribution, Vol. 143, No. 4, July 1996.

[14] K. Mahadevan and P. S. Kannan, *Comprehensive learning particle swarm optimization for reactive power dispatch*, Applied Soft Computing archive, Vol. 10, No. 2, P. 641-652, March 2010.

[15] M. A. Abido, *Multiobjective evolutionary algorithms for electric power dispatch problem*, IEEE Trans. On Evolutionary Computation, Vol. 10, No. 3, June 2006.

[16] C. H. Antunesa, D. F. Pires, C. Barrico, Á. Gomesa and A. G. Martinsa, *A multi-objective evolutionary algorithm for reactive power compensation in distribution networks*, Applied Energy, Vol. 86, No. 7-8, P. 977-984, July-August 2009.

[17] M. Azzam and A. A. Mousa, *Using genetic algorithm and TOPSIS technique for multiobjective reactive power compensation*, Electric Power Systems Research, Article, Vol. 80, No. 6, P. 675-681, June 2010.

[18] A. A. Mousa, M. A. El-Shorbagy and Waiel. F. Abd El-Wahed, *Local search based hybrid particle swarm optimization for multiobjective optimization*, International journal of Swarm and evolutionary computation, Vol. 3, P. 1-14, 2012.

[19] M. S. Osman, M. A. Abo-Sinna and A. A. Mousa, *Epsilon-Dominance based Multiobjective Genetic Algorithm for Economic Emission Load Dispatch Optimization Problem*, Electric Power Systems Research, Vol. 79, P. 1561–1567, 2009.

[20] K. Sindhya, A. Sinha, K. Deb and K. Miettinen, *Local search based evolutionary multiobjective optimization algorithm for constrained and unconstrained problems*, 2009 IEEE Congress on Evolutionary Computation, CEC 2009, P. 2919-2926, 2009.

[21] K. Harada, J. Sakuma, I. Ono and S. Kobayashi, *Local search for multiobjective optimization: Pareto descent method*, In Proceedings of the 18th Symposium on Decentralized Autonomous Systems of The Society of Instrument and Control Engineers, P. 345–350, 2006.

[22] R. Hooke and T. A. Jeeves, *Direct search solution of numerical and statistical problems*, Journal of the ACM, Vol. 8, No. 2, P. 212-229, 1961.

[23] H. Dommel and W. Tinney, *Optimal Power Flow Solutions*, IEEE Trans. Power Apparatus and Systems, Vol. PAS-87, No. 10, P. 1866-1876, Oct. 1968.

[24] J. D. Glover and M. Sarma, *Power System Analysis and Design*, Boston: PWS Publishing Company, 1994.

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

[26] K. Miettinen, *Non-linear multiobjective optimization*, Dordrecht: Kluwer Academic Publisher, 2002.

[27] D. E. Goldberg, *Genetic Algorithms in Search, Optimization & Machine Learning,* Addison-Wesley, Reading, MA, 1989.

[28] M. Tanaka, *GA- based decision support system for multi-criteria optimization*, In: Proceedings of the International Conference on systems, Man and Cybernetics, Vol. 2, P. 1556-1561, 1995.

[29] M. F. Bramlette, *Initialization Mutation and Selection Methods in Genetic Algorithms for Functions Optimization*, Proc ICGA, Vol. 4, P. 100-107, 1991.

[30] M. S. Osman, M. A. Abo-Sinna and A.A. Mousa, *IT-CEMOP: An Iterative Co-evolutionary Algorithm for Multiobjective Optimization Problem with Nonlinear Constraints*, Journal of Applied Mathematics & Computation (AMC), Vol. 183, P. 373-389, 2006.

[31] R. Zimmerman and D. Gan, *MATPOWER: A Matlab power system simulation package*, Available: http://www.pserc.cornell.edu/ matpower/.

[32] Y. Shi and R. C. Eberhart, *A modified particle swarm optimizer*, In Proceedings of the IEEE International Conference on Evolutionary computation, P. 69–73, IEEE Press, Piscataway, USA, 1998.

[33] J. Kennedy, R. C. Eberhart and Y. Shi, *Swarm Intelligence*, Morgan Kaufmann, 2001.

[34] A. Carlisle and G. Dozier, *Adapting particle swarm optimization to dynamic environments*, In International Conference on Artificial Intelligence, Vol. I, P. 429–434, Las Vegas, 2000.