



Science

ACTIVE POWER LOSS REDUCTION BY ASSORTED ALGORITHMS



Dr.K.Lenin *¹

*¹ Professor, Department of EEE, Prasad V.Potluri Siddhartha Institute of Technology, Kanuru, Vijayawada, Andhra Pradesh -520007, India

Abstract

This paper presents assorted algorithms for solving optimal reactive power problem. Symbiosis modeling (SM), which extends the dynamics of the canonical PSO algorithm by adding a significant ingredient that takes into account the symbiotic co evolution between species, Hybridization of Evolutionary algorithm with Conventional Algorithm (HCA) that uses the abilities of evolutionary and conventional algorithm and Genetical Swarm Optimization (GS), which combines Genetic Algorithms (GA) and Particle Swarm Optimization (PSO). All the above said SM, HCA, GS algorithms are used to augment the convergence rate with good Exploration & Exploitation. All the three SM, HCA, GS is applied to Reactive Power optimization problem and has been evaluated in standard IEEE 30 System. The results shows that all the three algorithms perform well in solving the reactive power problem with rapid convergence rate. Of all the three algorithms SM has the slight edge in reducing the real power loss over HCA&GS.

Keywords: Symbiosis modeling Hybridization of Evolutionary algorithm with conventional Algorithm; Genetical Swarm Optimization; Reactive Power Optimization.

Cite This Article: Dr.K.Lenin. (2018). “ACTIVE POWER LOSS REDUCTION BY ASSORTED ALGORITHMS.” *International Journal of Research - Granthaalayah*, 6(5), 263-275. <https://doi.org/10.29121/granthaalayah.v6.i5.2018.1448>.

1. Introduction

The reactive power optimization problem has a significant influence on secure and economic operation of power systems. This paper presents assorted algorithms for solving optimal reactive power problem. Symbiosis modeling (SM), which extends the dynamics of the canonical PSO algorithm by adding a significant ingredient that takes into account the symbiotic co evolution between species, Hybridization of Evolutionary algorithm with Conventional Algorithm (HCA) that uses the abilities of evolutionary and conventional algorithm and Genetical Swarm Optimization (GS), which combines Genetic Algorithms (GA) and Particle Swarm Optimization (PSO). The dynamics of each species in the ecosystem is manipulated by extending the dynamics of the canonical PSO model. We extend the control law of the canonical PSO model by adding a significant ingredient, which takes into account the symbiotic co evolution between species & called as symbiosis modeling (SM). Since mass extinction can be a natural feature of the

ecosystem's dynamics and sometimes is the result of the co evolution of species, species extinction and speciation are also simulated in our model. This model is instantiated as a novel multi-species optimizer; since this proposed algorithm contains two hierarchies and is based on the PSO model. The description of the biological details of the symbiosis theory and mass extinction theory in this paper were taken from [1–6]. The slave swarms execute the canonical PSO algorithm or its variants independently to maintain the diversity of particles, while the particles in the master swarm enhance themselves based on their own knowledge and also the knowledge of the particles in the slave swarms. HCA method is based on a new combination of conventional and evolutionary algorithms for sweeping the search space. In the former works the evolutionary algorithm has been used to scanning the search space in first, then a conventional algorithm has been employed such that the obtained swarm has used as initial guess by it. This method causes the abilities of the two algorithms to be used separately. The new hybrid technique here proposed, called Genetical Swarm Optimization (GS) and consists in a strong co-operation of GA and PSO, since it maintains the integration of the two techniques for the entire run of simulation. In each iteration, in fact, some of the individuals are substituted by new generated ones by means of GA, while the remaining part is the same of the previous generation but moved on the solution space by PSO. Doing so, the problem of premature convergence of the best individuals of the population to a local optimum, one of the most known drawbacks found in tests of hybrid global-local strategies, has been cancelled. All the three SM, HCA, GS is applied to Reactive Power optimization problem and has been evaluated in standard IEEE 30 System. The results shows that all the three algorithms perform well in solving the reactive power problem with rapid convergence rate .Of all the three algorithms SM has the slight edge in reducing the real power loss over HCA&GS.

2. Problem Formulation

The objective of the reactive power optimization problem is to minimize the active power loss in the transmission Network as well as to improve the voltage profile of the system. Adjusting reactive power controllers like Generator bus voltages, reactive Power of VAR sources and transformer taps performs reactive Power scheduling.

$$\min P_L = \sum_{i=1}^{NB} P_i(X, Y, \delta) \quad (1)$$

Subject to

i) The control vector constraints

$$X_{min} \leq X \leq X_{max} \quad (2)$$

ii) The dependent vector constraints

$$Y_{min} \leq Y \leq Y_{max} \quad (3)$$

iii) The power flow constraint

$$F(X, Y, \delta) = 0 \quad (4)$$

Where

$$X = [V_G, T, Q_C] \quad (5)$$

$$Y = [Q_G, V_L, I] \quad (6)$$

Fitness Function

$$F_N = P^n L + \alpha \sum_{j=1}^{NG} Q_{G,j}^{lim,n} + \beta \sum_{j=1}^{NL} V_{L,j}^{lim,n} ; n = 1, 2, \dots, N_n \quad (7)$$

α, β = penalty factors
 $P^n L$ = total real power losses of the n-th particle

$$Q_{G,j}^{lim,n} = \begin{cases} Q_{G,min} - Q_{G,j}^n & \text{if } Q_{G,j}^n < Q_{G,min} \\ Q_{G,j}^n - Q_{G,max} & \text{if } Q_{G,j}^n > Q_{G,max} \end{cases} \quad (8)$$

And

$$V_{L,j}^{lim,n} = \begin{cases} |V_{L,j}^n| - V_{L,max} & \text{if } |V_{L,j}^n| > V_{L,max} \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

Where

- NB - Number of buses in the system.
- δ - Vector of bus phase angles
- P_i - Real Power injection into the ith bus
- V_G - Vector of Generator Voltage Magnitudes
- T - Vector of Tap settings of on load Transformer Tap changer.
- Q_G - Vector of reactive Power generation.
- Q_C - Vector of reactive Power of switchable VAR sources.
- V_L - Vector of load bus Voltage magnitude.
- I - Vector of current in the lines.
- P_L - Vector of current in the lines.

3. Symbiosis Modeling (SM)

Symbiosis, initially defined by Anton de Bary in 1879, is simply the living together of organisms from different species. Here, we denote symbiosis as relationships that are constant and intimate between dissimilar species. Symbiosis includes three classify types: mutualism, commensalism and parasitism. In detail, mutualism describes a relationship in which all organisms involved derive benefits; commensalism result in only one species benefits without apparent benefit or cost to the other members of the association; and parasitism is a relationship in which one organism benefits at the cost to the other members. In a sense, what becomes interesting about symbiosis is not so much that organisms live together, but they cooperate. Over the years, symbiosis is used to refer to the special case of mutualism. Ultimately, what is of interest is not mutualism, but the cooperation that enable dissimilar species to intimately associate with each other over evolutionary

significant durations. Symbiosis is almost ubiquitous in nature. There are practically no plants or animals free of symbionts (organisms in symbiotic relationship) living on or in them. Research shows different types of symbiotic interactions. Some involve internal interactions, like bacteria in human intestines. Some of these interactions have seemed to lead to the evolution of organisms (e.g., the eukaryote cells, from which all plants and animals are descended, have symbiotic origin). Others appear to be purely behavioral, as in a human–honey guide mutualism that discussed above. Currently, enlightened evolutionary theory recognizes symbiosis as an integral process, and a fundamental source of innovation in evolution.

Mass Extinction

Species extinction has played a significant role in the history of life on Earth. Of the estimated one to four billion species have existed on the Earth, while less than 50 million are still alive today and all the others became extinct. There are two primary colleges of thought about the causes of extinction. The traditional view, still held by most paleontologists as well as many in other disciplines, is that extinction is the result of external stresses imposed on the ecosystem by the environment. At the other end of the scale, an increasing number of biologists and ecologists are supporting the idea that extinction has biotic causes – that extinction is a natural part of the dynamics of ecosystems and would take place regardless of any stresses arising from the environment. There is evidence in favour of this viewpoint that extinction can be the result of the interactions and dependencies between species. For example, in symbiotic mutualism or food web interactions context, the extinction of a specific species can cause the extinction of others. There are two best known model of species extinction [10]: the Bak-Sneppen model, which attempts to explain mass extinction as a result of species interactions; the extinction model of Newman, which models extinction as the result of environmental influences on species.

Canonical PSO Model

One of the best developed SI systems is Particle Swarm Optimization (PSO), which is a kind of swarm intelligence inspired by emergent collective intelligence of social model (e.g., bird flocks and fish shoals). The canonical PSO (CPSO) model evolves a single population of interacting particles, moving around in the D dimensional problem space searching for the optimum. Each particle is represented as $\vec{x}_i = (x_{i1}, x_{i2}, \dots, x_{id})$ and records its previous best position represented as $P_i = (P_{i1}, P_{i2}, \dots, P_{id})$, which is also called pbest. The index of the best particle among all the particles in the population is represented by the symbol g, and pg is called gbest. In the canonical PSO model, each particle is accelerated by two elastic forces, i.e., one attracts it to pbest and the other to gbest. The magnitude of the force is randomly chosen at each time step. At each generation, the equation controlling the particles is of the form:

$$a_i = c_1 r_1 (p_i - x_i) + c_2 r_2 (p_g - x_i) \quad (10)$$

Where c_1 and c_2 are two learning rates that control, respectively, the proportion of social transmission and individual learning in the swarm and r_1, r_2 are two random vectors uniformly distributed in $[0, 1]$.

The velocity of particle i, and its position updated every generation using the equations:

$$v_i = \chi(v_i + a_i) \quad x_i = x_i + v_i, \quad (11)$$

Where χ is known as the constriction coefficient [7-15]. However, studies showed that the canonical PSO (or original PSO algorithm) has difficulties in controlling the balance between exploration (global investigation of the search place) and exploitation (the fine search around a local optimum). The CPSO algorithm performs well in the early iterations (i.e., quickly converging towards an optimum in the first period of iterations), while has problems reaching a near optimal solution in some function optimization problems.

Symbiosis Co Evolution Model for Optimization

In this section, we describe our model for the co evolution of symbiotic species and formulate it as an optimization algorithm. We present the outline of our model by making the following assumptions:

- 1) Within species, species members cooperate with each other and rely on the presence of other species members for survival.
- 2) Between species, symbiotic partners from distinct species cooperate with each other and all partners gain an advantage to increase their survival ability.
- 3) Cooperation both within and between species are obligate through the whole life cycles of all species.
- 4) All species feel the same external environmental stress. Some species will become extinct if the stress is severe enough.
- 5) If one or more species went extinct, they will be replaced with equal number of new ones. Thus the number of species remains constant.

These assumptions yield a model that can be instantiated as the optimization algorithm, namely PS²O, which present below:

Here the basic goals is to find the minimum of $f(\vec{x})$, $\vec{x} \in R^d$. We create an ecosystem contains a species set $\Omega = \{S_1, S_2, \dots, S_n\}$, and each species possesses a members set $s_n = \{x_1^n, x_2^n, \dots, x_m^n\}$ mg i.e., totally $n \times m$ individuals co evolve in the ecosystem. The i th member of the k th species is characterized by the $\vec{x}_i^k = (x_{i1}^k, x_{i2}^k, \dots, x_{id}^k)$. Suppose $f(\vec{x}_i^k)$ is the fitness of x_i^k and lower value of the fitness represents the higher ability of survival. Under this presumed external environmental stress, all individuals in our model co evolve to the states of lower and lower fitness by cooperating each other both within species and between species. In each generation t , each individual x_i^k behaves as follow:

(a) Social evolution: this process addresses the cooperation between individuals of the same species. Due to the sociobiological background of the canonical PSO model, x_i^k evolve according to the rules of the canonical PSO algorithm in this process. Within the k th species, one or more members in the neighbourhood of x_i^k contribute their knowledge to x_i^k and x_i^k also share its knowledge with its neighbours. Then x_i^k accelerate towards the personal best position and the best position found by its neighbours:

$$a_i^k = c_1 r_1 (p_i^k - x_i^k) + c_2 r_2 (p_g^k - x_i^k) \quad (12)$$

Where k is the index of the species that the x_i^k belongs to, a_i^k is the acceleration vector of x_i^k , p_i^k is the personal best position found so far by x_i^k , p_g^k is the best position found so far by its neighbours within species k , c_1 are the individual learning rates, c_2 are the social learning rate; and $r_1, r_2 \in R^d$ are two random vectors uniformly distributed in $[0, 1]$.

(b) Symbiotic evolution: this process addresses the cooperation between individuals of distinct species. x_i^k beneficially interacts with and rewards all its symbiotic partners (individuals of dissimilar species), i.e., each symbiotic partner donates its knowledge to aid other partners. Then x_i^k accelerate towards its symbiotic partner of the best fitness:

$$a_i^k = a_i^k + c_3 r_3 (p_s^l - x_i^k) \quad (13)$$

Where l is the index of the species which the best symbiotic partner belongs to, c_3 is the “symbiotic learning rate”, $r_3 \in \mathbb{R}^d$ is a uniform random sequence in the range $[0, 1]$, and p_s^l is the best position found so far by the symbiotic partners of x_i^k .

(c) If all individuals in the ecosystem cannot find a better position after a (here a is a constant) generations, it means that all species suffer a severe external environmental stress. Then we randomly choose half species of the ecosystem go extinct to release this stress for other species to survive. At the same time, random initiate equal number of species in the ecosystem for new experimentations and adaptations. Let e be the set of the index of species that will go extinct, then the velocity v_i^k of each individual and its position are updated according to

$$\text{If } (k \in e) x_i^k = r * (ub - lb) + ub \quad (14)$$

$$\text{else } v_i^k = \chi(v_i^k + a_i^k), x_i^k = x_i^k + v_i^k \quad (15)$$

Where ub and lb are lower and upper boundaries of the search place, and $r \in \mathbb{R}^d$ is a random vectors uniformly distributed in $[0, 1]$. In summary, the equations in the three processes above can be combined as follow:

$$v_i^k = \chi(v_i^k + c_1 r_1 (p_i^k - x_i^k) + c_2 r_2 (p_g^k - x_i^k) + c_3 r_3 (p_s^l - x_i^k)) \quad (16)$$

$$x_i^k = \begin{cases} r * (ub - lb) + ub & \text{if species } k \text{ went extinct} \\ x_i^k + v_i^k & \text{otherwise} \end{cases} \quad (17)$$

The term $c_1 r_1 (p_i^k - x_i^k)$ is associated with cognition since it takes into account the individual’s own experiences. The term $c_2 r_2 (p_g^k - x_i^k)$ represents the social interaction within species k . The new term adding into the velocity update equation, $c_3 r_3 (p_s^l - x_i^k)$, takes into account the symbiotic co evolution (i.e., shared memory) between dissimilar species. With an extra control to the position update equation, the species extinction and speciation are also modelled.

Symbiosis modeling (SM) algorithm for solving reactive power problem

```
Set t: = 0;
INITIALIZE. Randomize positions and velocities of n*m particles in search space. Randomly
divide the whole population into n species each possesses mparticles;
WHILE (the termination conditions are not met)
FOR (each species k) IN PARALLEL
FOR (each particle i of species k)
Update the velocity using Eq. (16)
END FOR
IF (all species cannot find a better position after a generations)
Randomly choose a extinct species set e
IF (k ∈ e)
Update the position using Eq. (14)
END IF
ELSE
Update the position using Eq. (15)
END FOR IN PARALLEL
Set t: = t + 1;
END WHILE
```

4. Hybrid Evolutionary Algorithm with Conventional Algorithm (HCA)

Basic PSO Model

Particle swarm optimization (PSO) is originally attributed to Kennedy and Eberhart [16], based on the social behaviour of collection of animal such as birds flocking and fish schooling. In PSO algorithm each individual of the swarm, be called particle, remembers the best solution found by itself and by the whole swarm along the search trajectory. The particles move along the search space and exchange information with other particle according to the following equation.

$$V_{id} = \omega V_{id} + c_1 r_1 (P_{id} - X_{id}) + c_2 r_2 (P_{gd} - X_{id}) \quad (18)$$

$$X_{id} = X_{id} + V_{id}, \text{ for } d = 1, 2, \dots, N_i = 1, 2, \dots, s \quad (19)$$

Where X_{id} represent the current position of the particle, P_{id} is the best remembered individual particle position, P_{gd} denote the best remembered swarm position. c_1 and c_2 are cognitive and social parameters. r_1 and r_2 are random numbers between 0 and 1 and w is inertia weight which is used to balance the global and local search abilities. A large inertia weight facilitates global search while a small inertia weight facilitates local search. A decreasing function for the dynamic inertia weight can be devised in the following,

$$\omega = (iter_{max} - iter_{cur}) \cdot \left(\frac{\omega_{initial} - \omega_{final}}{iter_{max}} \right) + \omega_{final} \quad (20)$$

Where $w_{initial}$ and w_{final} represent the initial and final inertia weights respectively at the start of a given run, $iter_{max}$ the maximum number of iterations in a offered run, and $iter_{cur}$ the current iteration number at the present time step. However, global search ability at the end of the run may

be inadequate due to the utilization of a linearly decreasing inertia weight. The PSO may fail to find the required optimal in cases when the problem is too complicated. But to some extent, this can be overcome by employing a self-adapting strategy for adjusting the acceleration coefficients. Suganathan [17] applied the optimizing method that make the two factors decrease linearly with the increase of iteration numbers, but the results were not as good as the fixed value 2 of c_1 and c_2 . Ratnaweera [18] improve the convergence of particles to the global optima based on the way that make c_1 decrease and c_2 increase linearly with the increase of iteration numbers. The c_1 and c_2 is given by the following equations;

$$\begin{aligned}c_1 &= c_{1s} + \text{iter}_{cur}(c_{1e} - c_{1s})/\text{iter}_{max} \\c_2 &= c_{2s} + \text{iter}_{cur}(c_{2e} - c_{2s})/\text{iter}_{max}\end{aligned}\tag{21}$$

Steepest Descent Algorithm

The method of steepest descent (SD) is the simplest of the gradient methods. Imagine that there's a function $F(x)$, which can be defined and differentiable within a given boundary, so the direction it decreases the fastest would be the negative gradient of $F(x)$. To find the local minimum of $F(x)$, the method of SD is employed, where it uses a zig-zag like path from an arbitrary point and gradually slide down the gradient, until it converges to the actual point of minimum. Although this optimizing method is less time consuming than the population based search algorithms, it is highly dependent on the initial estimate of solution. This method can be describe by the following formula,

$$u_k = \frac{g(x_k)}{\|g(x_k)\|}\tag{22}$$

$$x_{k+1} = x_k + \alpha_k u_k\tag{23}$$

Where x_{k+1} represent the position of new solution than x_k , $g(x_k)$ is gradient of $F(x)$ in x_k . α_k is a scalar that should decrease in each iteration. Near the optimum point, lower range of α_k make more little movement in x_k and so, a more precise search.

The New Hybrid Evolutionary-Conventional Algorithm (HCA)

This method is based on a new combination of conventional and evolutionary algorithms for sweeping the search space. In the former works the evolutionary algorithm has been used to scanning the search space in first, then a conventional algorithm has been employed such that the obtained swarm has used as initial guess by it. This method cause the abilities of the two algorithms be used separately. So we cannot guarantee a better result because the proper iteration that we should changed the algorithms cannot be determined. On the other hand when we separately use these two algorithms we cannot gain their abilities in the same time. In the proposed method in each iteration after using the evolutionary algorithm, the swarm be employed in the conventional algorithm for a deterministic time. So in each iteration the main abilities such as randomly search and a more precisely search near the optimum point can be satisfied. When the number of iterations increased and the proposed algorithm approach to a global extremum, SD algorithm can be used (HECA) to a more precise search near the global extremum. Because near the global extremum there is no need to a randomly search.

HCA algorithm to solve reactive power optimization problem

Step1. Specify the number of times which steepest descent algorithm should be applied in each iteration of PSO algorithm. This number can be a non-integer number. This means that PSO algorithm applies for a specified number in each iteration of steepest algorithm.

Step2. Determine the PSO parameters, like number of particle, number of iteration, cognitive and social parameter and so on.

Step3. Initialize the positions and velocities of particles randomly in search space.

Step4. Evaluate each particle fitness value. Calculate P_{id} and P_{gd} based on their definitions.

Step5. Based on the number which is specified in step1, move the particles through the search space by PSO and SD algorithms in each iteration. The positions and velocities of all particles are updated according to equations (18), (19), (22) and (23), and then a group of new particles are generated.

Step6. If the maximal iterative of generation are arrived go to step 8 else go to step 7.

Step7. Calculate the new amount of c_1 and c_2 according to equations (20) and (21) and go to step 4.

Step8. Use the obtained swarm as initial guess of SD algorithm and move the particles by SD algorithm for specified times.

5. Genetical Swarm Optimization (GS)

Genetic Algorithms [16], [17] simulate the natural evolution, in terms of survival of the fittest, adopting pseudo-biological operators such as selection, crossover and mutation. Selection is the process by which the most highly rated individuals in the current generation are chosen to be involved as parents in the creation of a new generation. The crossover operator produces two new individuals by recombining the information from two parents. The random mutation of some gene values in an individual is the third GA operator. One of the most recently developed evolutionary techniques is instead represented by the Particle Swarm Optimization [18]-[20], that is based on a model of social interaction between independent agents and uses social knowledge or swarm intelligence in order to find the global maximum or minimum of a function. A brief introduction of PSO algorithm is given in this section. PSO uses social rules to search in the parameter space by controlling the trajectories of a set of independent particles. The position of each particle, representing a particular solution of the problem, is used to compute the value of the fitness function to be optimized. Each particle may change its position, and consequently may explore the solution space, simply varying its associated velocity. The main PSO operator is, in fact, velocity update, that takes into account the best position, in terms of fitness value, reached by all the particles during their paths, G , and the best position that the agent itself has reached during its search, P_i , resulting in a migration of the entire swarm towards the global optimum. At each iteration the particle moves around according to its velocity and position; the cost function to be optimized is evaluated for each particle in order to rank the current location. The velocity of the particle is then stochastically updated, according to the next formula:

$$V_i = \omega V_i + c_1 \phi_1 (P_i - X_i) + c_2 \phi_2 (G - X_i) \quad (24)$$

The term $\omega < 1$ is known as the “inertial weight” and it is a friction factor chosen between 0 and 1 in order to determine to what extent the particle remains along its original course unaffected by the pull of the other two terms; it is very important to prevent oscillations around the optimal value. The other two terms are known as self-knowledge and social knowledge and they are balanced by

the scaling factors $c_{1,2}$ that are constants and by $\phi_{1,2}$ that are random, positive, numbers with a uniform distribution and a value that goes from 0 to 1. Some comparisons of the performances of GA and PSO are present in literature [19-21], underlining the reliability and convergence speed of both methods, but continuing in keeping them separate. In particular PSO have faster convergence rate than GA early in the run, but often they are outperformed by GA for long simulation runs, when the last ones find a better solution. Anyway, the population-based representation of the parameters that characterizes a particular solution is the same for both the algorithms; therefore it is possible to implement a hybrid technique in order to utilize the qualities and uniqueness of the two algorithms. Some attempts have been done in this direction with good results, but with weak integration of the two strategies, because one algorithm is used mainly as the pre-optimizer for the initial population of the other one. The new hybrid technique here proposed, called Genetical Swarm Optimization, and consists in a strong co-operation of GA and PSO, since it maintains the integration of the two techniques for the entire run. In each iteration, in fact, the population is divided into two parts and it is evolved with the two techniques respectively. The two parts are then recombined in the updated population, that is again divided randomly into another two parts in the next iteration for another run of genetic or particle swarm operators.

GS Algorithm for solving the optimal reactive power dispatch problem.

- Step 1. Initialization the population
- Step 2. Randomly population has been selected
- Step 3. Fitness has been calculated for all individuals.
- Step 4. Split the population for GA and PSO
- Step 5. In GA i. Selective reproduction, ii. Crossover, iii. Mutation has been done
- Step 6. In PSO i. Velocity updating, ii. Calculation of new positions, iii. Updating (personal & global) has been done.
- Step 7. Resulting new population in outcome of combining both the PSO & GA.
- Step 8. If best individuals reached means it will stop or go to step 2 .

6. Simulation Results

Validity of the proposed SM, HCA, GS algorithms has been verified by testing in IEEE 30-bus, 41 branch system and it has 6 generator-bus voltage magnitudes, 4 transformer-tap settings, and 2 bus shunt reactive compensators. Slack bus is Bus 1 and 13 PV generator buses are 2, 5, 8, 11 and others are PQ load buses. In Table 1 Control variables limits are listed.

Table 1: Primary Variable Limits (Pu)

List of Variables	Minimum	Maximum	Type
Generator Bus	0.95	1.1	Continuous
Load Bus	0.95	1.05	Continuous
Transformer-Tap	0.9	1.1	Discrete
Shunt Reactive Compensator	-0.11	0.31	Discrete

The power limits of generators buses are listed in Table 2.

Table 2: Generators Power Limits

Bus	Pg	Pgminimum	Pgmaximum	Qgminimum	Qgmaximum
1	96.00	49	200	0	10

2	79.00	18	79	-40	50
5	49.00	14	49	-40	40
8	21.00	11	31	-10	40
11	21.00	11	28	-6	24
13	21.00	11	39	-6	24

Proposed SM, HCA, GS algorithms successfully kept the control variables within limits & it shown in Table 3. Performance of the proposed SM, HCA, and GS algorithms has been shown Table 4. Table 5 list out the overall comparison of real power loss.

Table 3: After optimization values of control variables

List of Control Variables	SM	HCA	GS
V1	1.0427	1.0432	1.0441
V2	1.0465	1.0472	1.0484
V5	1.0121	1.0132	1.0187
V8	1.0282	1.0291	1.0294
V11	1.0662	1.0679	1.0682
V13	1.0464	1.0469	1.0472
T4,12	0.00	0.00	0.00
T6,9	0.01	0.01	0.01
T6,10	0.90	0.90	0.90
T28,27	0.90	0.91	0.91
Q10	0.10	0.10	0.10
Q24	0.10	0.10	0.10
Real power loss	4.2604	4.2678	4.2688
Voltage deviation	0.9082	0.9088	0.9092

Table 4: Performance of SM, HCA & GS approach

	SM	HCA	GSO
Iterations	32	39	41
Time taken (secs)	9.81	9.98	10.01
Real power loss	4.2604	4.2678	4.2688

Table 5: Comparison of results

List of Techniques	Real power loss (MW)
SGA(Wu et al., 1998) [22]	4.98
PSO(Zhao et al., 2005) [23]	4.9262
LP(Mahadevan et al., 2010) [24]	5.988
EP(Mahadevan et al., 2010) [24]	4.963
CGA(Mahadevan et al., 2010) [24]	4.980
AGA(Mahadevan et al., 2010) [24]	4.926
CLPSO(Mahadevan et al., 2010) [24]	4.7208
HSA (Khazali et al., 2011) [25]	4.7624
BB-BC (Sakthivel et al., 2013) [26]	4.690
MCS(Tejaswini sharma et al.,2016) [27]	4.87231

Proposed SM	4.2604
Proposed HCA	4.2678
Proposed GS	4.2688

7. Conclusion

In this paper proposed SM, HCA, GS algorithms has been successfully solved reactive power problem. The performance of the proposed algorithms demonstrated through its evaluation on standard IEEE 30 bus power systems. The results shows that all the three algorithms perform well in solving the reactive power problem with rapid convergence rate .Of all the three algorithms SM has the slight edge in reducing the real power loss over HCA &GS.

References

- [1] S.A. Frank, Models of symbiosis, American Naturalist 150 (1997) 80–99.
- [2] M.D. Jason, S.G. Catherine, A.S. Stephen, J.R. Steven, Symbioticism and complex adaptive systems I: implications of having symbiosis occur in nature,in: Proceedings of the 5th Annual Conference on Evolutionary Programming, Cambridge, 1996, pp. 177–186.
- [3] C.R. Marshall, Mass extinction probed, Nature 392 (1998) 17–20.
- [4] [4] M. Newman, Simple models of evolution and extinction, IEEE Computing in Science and Engineering (2000) 80–86.
- [5] Ben Niu, Yunlong Zhu, XiaoXian He, Xiangping Zeng, Henry Wu, MCPSO: a multi-swarm cooperative particle swarm optimizer, Applied Mathematics and Computation 185 (2) (2007) 1050–1062
- [6] B. Niu, Y.L. Zhu, X.X. He, Multi-population cooperative particle swarm optimization, in: Proceedings ECAL2005, Lecture Notes in Computer Science No.3630, 2005, pp. 874–883.
- [7] J. Kennedy, Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance, in: Proceedings of the Congress on Evolutionary Computation, Piscataway, NJ, 1999, pp. 1931–1938.
- [8] J. Kennedy, R. Mendes, Population structure and particle swarm performance, in: Proceedings of the 2002 Congress on Evolutionary Computation, Piscataway, NJ, 2002, pp. 1671–1675.
- [9] P.N. Suganthan, Particle swarm optimizer with neighborhood operator, in: Proceedings of the Congress on Evolutionary Computation (CEC 1999), Piscataway, NJ, 1999, pp. 1958–1962.
- [10] X.H. Shi, Y.C. Liang, H.P. Lee, C. Lu, L.M. Wang, An improved GA and a novel PSO-GA-based hybrid algorithm, Information Processing Letters 93 (2005)255–261.
- [11] W.J. Zhang, X.F. Xie, DEPSO: Hybrid particle swarm with differential evolution operator, in: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Washington, DC, USA, 2003, pp. 3816–3821.
- [12] B. Ye, C.Z. Zhu, C.X. Guo, Y.J. Cao, Generating extended Fuzzy basis function networks using hybrid algorithm, Lecture Notes in Artificial Intelligence3613 (2005) 79–88.
- [13] S. He, Q.H. Wu, J.Y. Wen, J.R. Saunders, R.C. Paton, A particle swarm optimizer with passive congregation, Biosystems 78 (2004) 35–147.
- [14] B. Niu, Y.L. Zhu, X.X. He, Multi-population cooperative particle swarm optimization, in: Proceedings ECAL2005, Lecture Notes in Computer Science No.3630, 2005, pp. 874–883.
- [15] M. Clerc, J. Kennedy, The particle swarm: explosion stability and convergence in a multidimensional complex space, IEEE Transactions on Evolutionary Computation 6 (1) (2002) 58–73.
- [16] J. Kennedy, R. Eberhart, Particle swarm optimization, In: Proc IEEE intconf neural networks, vol.IV, Perth, Australia;1995. p.1942-48.

- [17] P. N. Suganthan, Particle Swarm Optimizer with Neighborhood Operator, in Proceedings of IEEE International Conference on Evolutionary Computation. Washington D.C. IEEE Press (1999), pp. 1958-1962.
- [18] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, Self-Organizing Hierarchical Particle Swarm Optimizer with Time-Varying Acceleration Coefficients, IEEE Transactions on Evolutionary Computation 8(3) 240-255 (2004).
- [19] D. W. Boeringer, and D. H. Werner, "Particle Swarm Optimization Versus Genetic Algorithms for Phased Array Synthesis," IEEE Transactions on antennas and propagation, vol. 52, n. 3, pp. 771-779, March 2004.
- [20] J. Robinson, S. Sinton, and Y. Rahmat-Samii, "Particle swarm, genetic algorithm, and their hybrids: optimization of a profiled corrugated horn antenna," Proc. of the 2002 IEEE AP-S International Symposium, San Antonio, TX, vol. 1, pp. 314-317, June 2002.
- [21] Chia-Feng Juang, "A Hybrid of Genetic Algorithm and Particle Swarm Optimization for Recurrent Network Design," IEEE Transactions On Systems, Man And Cybernetics-Part B: Cybernetics, vol. 34, n. 2, pp. 997-1006, April 2004.
- [22] Wu.Q.H, Y.J.Cao, and J.Y.Wen, (1998), "Optimal reactive power dispatch using an adaptive genetic algorithm", Int.J.Elect.Power Energy Syst. Vol 20. Pp. 563-569.
- [23] Zhao.B, C.X.Guo, and Y.J.CAO, (2005), "Multiagent-based particle swarm optimization approach for optimal reactive power dispatch", IEEE Trans. Power Syst. Vol. 20, no. 2, pp. 1070-1078.
- [24] Mahadevan.K, Kannan P.S, (2010) "Comprehensive Learning Particle Swarm Optimization for Reactive Power Dispatch", Applied Soft Computing, Vol. 10, No. 2, pp. 641-652.
- [25] Khazali.A.H, M.Kalantar, (2011), "Optimal Reactive Power Dispatch based on Harmony Search Algorithm", Electrical Power and Energy Systems, Vol. 33, No. 3, pp. 684-692.
- [26] Sakthivel.S, M.Gayathri, V.Manimozhi, (2013), "A Nature Inspired Optimization Algorithm for Reactive Power Control in a Power System", International Journal of Recent Technology and Engineering, pp29-33 Vol.2, Issue-1.
- [27] Tejaswini Sharma, Laxmi Srivastava, Shishir Dixit (2016). "Modified Cuckoo Search Algorithm For Optimal Reactive Power Dispatch", Proceedings of 38 th IRF International Conference, pp4-8. 20th March, 2016, Chennai, India, ISBN: 978-93-85973-76-5.

*Corresponding author.

E-mail address: gklenin@ gmail.com