International Journal of Mechatronics, Electrical and Computer Technology

Dec. 2013, Special Number, pp 449-469, **ISSN**: 2305-0543

Available online at: http://www.aeuso.org

Attractive and Repulsive Particle Swarm Optimization and Random Virus Algorithm for Solving Reactive Power Optimization Problem

K. Lenin, B. Ravindranath Reddy and M. Surya Kalavathi

Electrical and Electronics Engineering, Jawaharlal Nehru Technological University, Kukatpally, Hyderabad, India

*Corresponding Author's E-mail: *gklenin@gmail.com*

Abstract

Reactive Power Optimization is a complex combinatorial optimization problem involving non-linear function having multiple local minima, non-linear and discontinuous constrains. This paper presents Attractive and repulsive Particle Swarm Optimization (ARPSO) and Random Virus Algorithm (RVA) in trying to overcome the Problem of premature convergence. RVA and ARPSO is applied to Reactive Power Optimization problem and is evaluated on standard IEEE 30Bus System. The results show that RVA prevents premature convergence to high degree but still keeps a rapid convergence. It gives best solution when compared to Attractive and repulsive Particle Swarm Optimization (ARPSO) and Particle Swarm Optimization (PSO).

Keywords*:* Attractive and repulsive, particle Swarm, random virus algorithm, Reactive Power Optimization.

1. Introduction

The reactive power optimization problem has a significant influence on secure and economic operation of power systems. The reactive power generation, although itself having no production cost, does however affect the overall generation cost by the way of the transmission loss. A procedure, which allocates the reactive power generation so as to minimize the transmission loss, will consequently result on the lowest production cost for

International Journal of Mechatronics, Electrical and Computer Technology

Dec. 2013, Special Number, pp 449-469, **ISSN**: 2305-0543

Available online at: http://www.aeuso.org

which the operation constraints are satisfied. The operation constraints may include reactive power optimization problem. The conventional gradient-based optimization algorithm has been widely used to solve this problem for decades. Obviously, this problem is in nature a global optimization problem, which may have several local minima, and the conventional optimization methods easily lead to local optimum. On the other hand, in the conventional optimization algorithms, many mathematical assumptions, such as analytic and differential properties of the objective functions and unique minima existing in problem domains, have to be given to simplify the problem.

Otherwise it is very difficult to calculate the gradient variables in the conventional methods. Further, in practical power system operation, the data acquired by the SCADA (Supervisory Control and Data Acquisition) system are contaminated by noise. Such data may cause difficulties in computation of gradients. Consequently, the optimization could not be carried out in many occasions. In the last decade, many new stochastic search methods have been developed for the global optimization problems such as simulated annealing, genetic algorithms and evolutionary programming.

A major problem with evolutionary algorithms (EAs) in multi-modal optimization is premature convergence (PC), which results in great performance loss and sub-optimal solutions. As far as GAs is concerned, the main reason for premature convergence is a too high selection pressure or a too high gene flow between population individuals. With PSOs the fast information flow between particles seems to be the reason for clustering of particles.

Diversity declines rapidly, leaving the PSO algorithm with great difficulties of escaping local optima. Consequently, the clustering leads to low diversity with fitness stagnation as an overall result. Recently R. Ursem has suggested a model called the Diversity-Guided Evolutionary Algorithm (DGEA) [1]. He redefines the traditional mutation operator, the Gaussian mutation, to be a directed mutation instead. The important issue is that this directed

International Journal of Mechatronics, Electrical and Computer Technology

Dec. 2013, Special Number, pp 449-469, **ISSN**: 2305-0543

Available online at: http://www.aeuso.org

mutation, in general, increases the diversity, whereas normal Gaussian mutation is not likely to do this, because it simply adds random noise from some distribution with a mean of zero, normally $N(0; \sigma^2)$. Consequently, the DGEA applies diversity-decreasing operators (selection, recombination) and diversity-increasing operators (mutation) to alternate between two modes based upon a distance-to-average-point measure. The performance of the DGEA clearly shows its potential in multi-modal optimization.

As [1] rightfully pinpoints, the diversity measure is traditionally used to *analyze the* evolutionary algorithms rather than *guide* them. We are great believers of *adaptive controlling*; that measuring and using different properties of the swarm/population while running, adds significant potential to the algorithm. We have therefore adopted the idea from Ursem with the decreasing and increasing diversity operators used to control the population into the basic PSO model. We find, it is a natural modification of the PSO, and the idea behind it is surprisingly simple. The modified model uses a diversity measure to have the algorithm alternate between exploring and exploiting behavior. We introduce two phases *attraction* and *repulsion*.

By measuring the diversity we let the swarm alternate between these phases. As long as the diversity is above a certain threshold $d_{low}$the particles attract each other. When the diversity declines below $d_{low}$the particles simply switch and start to repel each other until the threshold $d_{high}$is met. With this simple scheme we obtain our modified model, which we have chosen to call the ARPSO model – the attractive and repulsive PSO. Random Virus Algorithm (RVA) [6] a new, evolutionary type algorithm is proposed to the multidisciplinary optimization. The algorithm is based on the simulation of spreading of biological viruses, which results in a boom-like propagation in the number of entities and quick improvement of the objective function in subsequent iterations. A virus is always a very simple construction, contains only the most important information necessary for life and reproduction. This simplicity gives a very high flexibility to a virus in changing and mutation, therefore they can

International Journal of Mechatronics, Electrical and Computer Technology

Dec. 2013, Special Number, pp 449-469, **ISSN**: 2305-0543

Available online at: http://www.aeuso.org

accommodate to several conditions very easily. Therefore the efficiency of a virus is very high in point of view of behaviour, construction, and life reproduction and changing.

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.

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

Subject to

i) The control vector constraints

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

ii) The dependent vector constraints

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

and

iii) The power flow constraint

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

where

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

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

International Journal of Mechatronics, Electrical and Computer Technology

Dec. 2013, Special Number, pp 449-469, **ISSN**: 2305-0543

Available online at: http://www.aeuso.org

NB	-	Number of buses in the system.
δ	-	Vector of bus phase angles
P^i	-	Real Power injection into the i^{th} bus
V_G	-	Vector of Generator Voltage Magnitudes
T	-	Vector of Tap settings of on load Transformer Tap changer.
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. Basic PSO Model

The basic PSO model consists of a swarm of particles moving in an n-dimensional, real valued search space of possible problem solutions. For the search space, in general, a certain quality measure, the fitness, is defined making it possible for particles to compare different problem solutions. Every particle has a position vector x and a velocity vector v. Moreover, each particle contains a small memory storing its own best position seen so far p and a global best position g obtained through communication with its fellow neighbor particles. This information flow is obtained by defining a neighborhood topology on the swarm telling particles about immediate neighbors.

The intuition behind the PSO model is that by letting information about good solutions spread out through the swarm, the particles will tend to move to good areas in the search space. At each time step t the velocity is updated and the particle is moved to a new position. This new position is simply calculated as the sum of the previous position and the new velocity:

$$\vec{x}(t+1) = \vec{x}(t) + \vec{\upsilon}(t+1) \tag{7}$$

International Journal of Mechatronics, Electrical and Computer Technology

Dec. 2013, Special Number, pp 449-469, **ISSN**: 2305-0543

Available online at: http://www.aeuso.org

The update of the velocity from the previous velocity to the new velocity is, as implemented in this paper, determined by:

$$\vec{\upsilon}(t+1) = \omega.\vec{\upsilon}(t) + \varphi_1(\vec{p}(t)) + \varphi_2(\vec{g}(t) \quad \vec{x}(t)), \tag{8}$$

Where

ϕ_1 and ϕ_2 are real numbers chosen uniformly and at random in a given interval, usually [0, 2]. These values determine the significance of $\vec{p}(t)$ *and* $\vec{g}(t)$ respectively. The parameter w is the inertia weight and controls the magnitude of the old velocity $\vec{\upsilon}(t)$ in the calculation of the new velocity $\vec{\upsilon}(t+1)$.[2]

3.1 The Modified Model – ARPSO

We define the attraction phase merely as the basic PSO algorithm. The particles will then attract each other, since in general they attract each other in the basic PSO algorithm because of the information flow of good solutions between particles. We define the second phase repulsion, by "inverting" the velocity-update formula of the particles:

$$\vec{\upsilon}(t+1) = \omega.\vec{\upsilon}(t) \quad \varphi_1(\vec{p}(t) \quad \vec{x}(t)) \quad \varphi_2(\vec{g}(t) \quad \vec{x}(t)). \tag{9}$$

In the repulsion phase the individual particle is no longer attracted to, but instead repelled by the best known particle position vector g(t) and its own previous best position vector p(t). In the attraction phase the swarm is contracting, and consequently the diversity decreases. When the diversity drops below a lower bound, d_{low}, we switch to the repulsion phase, in which the swarm expands due to the above inverted update-velocity formula (9). Finally, when a diversity of d_{high} is reached, we switch back to the attraction phase. The result of this is an algorithm that alternates between phases of exploiting and exploring – attraction and repulsion – low diversity and high diversity. The pseudo-code for the ARPSO algorithm is shown in Fig. 1 and 2.

International Journal of Mechatronics, Electrical and Computer Technology

Dec. 2013, Special Number, pp 449-469, **ISSN**: 2305-0543

Available online at: http://www.aeuso.org


```
Program PSO
        Init ();
        While not done do
        SetDirection ();            //new!
        UpdateVelocity ();
        NewPosition ();
        AssginFitness ();
        CalculateDiversity ();      // new!
```

Fig. 1: The ARPSO algorithm.

```
Function setDirection
        If (dir > 0 && diversity < dLow) dir = -1;
        If (dir > 0 && diversity < dHigh) dir = 1;
```

Fig. 2: setDirecton

The first of the two new functions, **setDirection** determines which phase the algorithm is currently in, simply by setting a sign-variable, dir, either to 1 or -1 depending on the diversity. In the second function, **calculateDiversity**, the diversity of the swarm (in the pseudo-code stored in the variable "diversity"), is set according to the diversity-measure:

$$diversity(S) = \frac{1}{|S|.|L|} \cdot \sum_{i=1}^{|S|} \sqrt{\sum_{j=1}^{N} p_{ij} \quad \bar{p}_j)^2}, \qquad (10)$$

where S is the Swarm, [S] is the swarmsize, [L] is the length of longest the diagonal in the search space, N is the dimensionality of the problem, p_{ij} is the j'th value of the I'th particle and

International Journal of Mechatronics, Electrical and Computer Technology

Dec. 2013, Special Number, pp 449-469, **ISSN**: 2305-0543

Available online at: http://www.aeuso.org

pj is the j'th value of the average point p. Note that this diversity measure is independent of swarm size, the dimensionality of the problem as well as the search range in each dimension.

Finally, the velocity-update formula, eqn. (9) is changed by multiplying the sign-variable direction to the two last terms in it. This decides directly whether the particles attract or repel each other:

$$\vec{v}(t+1) = \omega.\vec{v}(t) + dir(\varphi_1(\vec{p}(t) \quad \vec{x}(t)) + \varphi_2(\vec{g}(t) \quad \vec{x}(t))). \qquad (11)$$

4. Algorithm for RPO using ARPSO

The proposed RPO algorithm using the ARPSO can be expressed as follows:

Step 1. Initial searching points and velocities of agents are generated.

Step 2. Ploss to the searching points for each agent is calculated using the load flow calculation. If the constraints are violated, the penalty is added to the loss (evaluation value of agent).

The fitness function of each particle is calculated as:

$$f_n = P^n{}_{L} + \alpha \sum_{j=1}^{NG} Q_{G,j}^{lim,n} + \beta \sum_{j=1}^{NL} V_{L,j}^{lim,n} \quad ; n = 1,2..,N_n \qquad \ldots (12)$$

α, β = penalty factors

$P^n{}_L$ = total real power losses of the n^{-th} particle

$$Q_{G,j}^{lim,n} = \begin{cases} Q_{G,min} - Q_{G,j}^n & \text{if } Q_{G,j}^n < Q_{G,min} \\ Q_{G,j}^n - Q_{G,max} & \text{if } Q_{G,j}^n > Q_{G,max} \end{cases} \qquad \ldots (13)$$

International Journal of Mechatronics, Electrical and Computer Technology
Dec. 2013, Special Number, pp 449-469, **ISSN**: 2305-0543
Available online at: http://www.aeuso.org

and

$$V_{L,j}^{\lim,n} = \begin{cases} \left|V_{L,j}^{n}\right| - V_{L,\max}, & \text{if } \left|V_{L,j}^{n}\right| > V_{L,\max} \\ 0 & \text{otherwise} \end{cases} \quad \ldots \quad (14)$$

Step 3. Pbest is set to each initial searching point. The initial best evaluated value (loss with penalty) among pbests is set to gbest.

Step 4. New velocities are calculated using eqn. (7).

Step 5. Update the velocity from previous velocity to the new velocity using eqn. (8).

Step 6. To new function applied.

i. ***setdirection***

ii. calculateDiversity to control swarm.

Step 7. Ploss to the new searching points and the evaluation values are calculated.

Step 8. If the evaluation value of each agent is better than the previous pbest, the value is set to pbest. If the best pbest is better than gbest, the value is set to gbest. All of gbests are stored as candidates for the final control strategy.

Step 9. If the iteration number reaches the maximum iteration number, then stop. Otherwise, go to Step 4. If the voltage and power flow constraints are violated, the absolute violated value from the maximum and minimum boundaries is largely weighted and added to the objective function (1) as a penalty term. The maximum iteration number should be determined by pre-simulation. As mentioned below, PSO requires less than 100 iterations even for large-scale problems.

International Journal of Mechatronics, Electrical and Computer Technology

Dec. 2013, Special Number, pp 449-469, **ISSN**: 2305-0543

Available online at: http://www.aeuso.org

5. Random Virus Algorithm

A new evolutionary type algorithm is proposed to the multidisciplinary optimization. The algorithm is based on the simulation of spreading of biological viruses, which results in a boom-like propagation in the number of entities and quick improvement of the objective function in subsequent iterations. It is possible to control the number of entities if in each iteration step 20- 30% of the entities (e.g. the ones having the worst objective function value or those having child's and grandchild's all with worst objective function values than themselves) are eliminated.

5.1. The concept of the Random Virus Algorithm (RVA)

Let us investigate a very efficient biological construction: a virus. The efficiency of this biological system is in his very fast reproduction capacity. If the circumstances and conditions (temperature, light, oxygen, and food) are good, viruses can reproduce themselves in a very high speed and they will cover almost all the possible places in the area of investigation. If the life conditions are not good, or the conditions show a non-uniform distribution, higher number of viruses can be found in an area of better conditions than in some areas giving poor conditions of life for viruses.

Therefore these biological structures are very efficient in finding the best conditions of life, the highest number of virus entities will be found in the area having the best life conditions. Another very important thing is that a virus is always a very simple construction, contains only the most important information necessary for life and reproduction. This simplicity gives a very high flexibility to a virus in changing and mutation, therefore they can accommodate to several conditions very easily.

Therefore the efficiency of a virus is very high in point of view of behaviour, construction, and life reproduction and changing. This efficiency is applied during the development of

International Journal of Mechatronics, Electrical and Computer Technology

Dec. 2013, Special Number, pp 449-469, **ISSN**: 2305-0543

Available online at: http://www.aeuso.org

computer viruses, which show several similarities to real biological viruses (simple structures, very fast reproduction, easy changing). Mainly these characteristics give that a computer virus is also a very efficient system. Applying this multi-form efficiency in development of an optimization algorithm could result in high efficiency in the optimum searching process, too. The algorithm presented here destroys 30% of points having the worst objective function value, after the third generation step.

This gives high efficiency in the starting phase of the searching process and later the total number of entities will be limited. This is also good in point of view of small number of objective function evaluations and constraint checking to reach the final optimum. On the basis of the above considerations, it is possible to build the algorithm. First step is to find the starting points fulfilling all the explicit and implicit constraints. It is possible to generate point coordinates inside of explicit constraints limits and check the generated points against the implicit constraints.

In order to keep the simplicity of the virus algorithm, the number of the starting points is proposed to be set at a very low value. The coordinates of the starting points can be denoted by x_i , i = 1,2,…,n where n is the number of design variables. In this case the points can be denoted as P_j , j= 1,2,…,m , where m is the number of the starting points. The implicit constraints of the design variables:

$$l_i \leq x_i \leq h_i \qquad i = 1,2,\dots,n \tag{15}$$

l_i are the lower limits, h_i are upper limits of the constrains. The implicit constraints can be written in the following form:

$$u_k \leq f_k(x_i) \leq v_k \quad , \quad \text{i=1,2,…,n} \quad , \quad \text{k=1,2,…,p} \tag{16}$$

International Journal of Mechatronics, Electrical and Computer Technology

Dec. 2013, Special Number, pp 449-469, **ISSN**: 2305-0543

Available online at: http://www.aeuso.org

Where p is the number of implicit constraints. The starting points are vectors in the design space:

$$P_j = \{x_i\}_j \tag{17}$$

They can be found in the feasible region of explicit constraints by using random numbers and after they are checked against the implicit constraints. If a point is unfeasible, a new one should be generated. The goal of the optimization process is to find the extreme value (maximum or minimum) of the objective function:

$$\Omega = F(x_i) \tag{18}$$

F is an arbitrary nonlinear function of the design variables. Once the starting points generated, the reproduction procedure is starting:

$$y_i = x_i + R_i q(h_i - l_i) \tag{19}$$

Here y_i are the coordinates of the new point generated, R_i are random numbers between the values of 0 and 1 and α is the number of new entities generated in the reproduction procedure, q is the spreading parameter. Proposed value of spreading parameter is between 0.5 and 0.8 in case of the first three generations and between 0.2 and 0.4 afterwards. The reproduction step is executed for each starting point. The next generation (we can call it generation β, after γ and so on) can be created using the reproduction formula of equation (5) for each point of the previous generation. In order to prevent the overwhelmingnumber of points controlled at the same time, it is necessary to select the points having the best objective function value and destroy the points having the worst objective function value. This procedure can be continued until a given number of generations are reached or the procedure can be ended if the maximum difference in objective function values regarding the last generation created will be under a given small value. In some cases, where the evaluation of the objective function needs a large amount of calculations, (for example solution of a large

International Journal of Mechatronics, Electrical and Computer Technology

Dec. 2013, Special Number, pp 449-469, **ISSN**: 2305-0543

Available online at: http://www.aeuso.org

finite element model) it is possible to insert a security upper limit for the total number of calculations of the objective function or a given constraint checking, in order to prevent too much amounts of calculations. It is recommended to consider all these parameters before ending the optimization procedure.

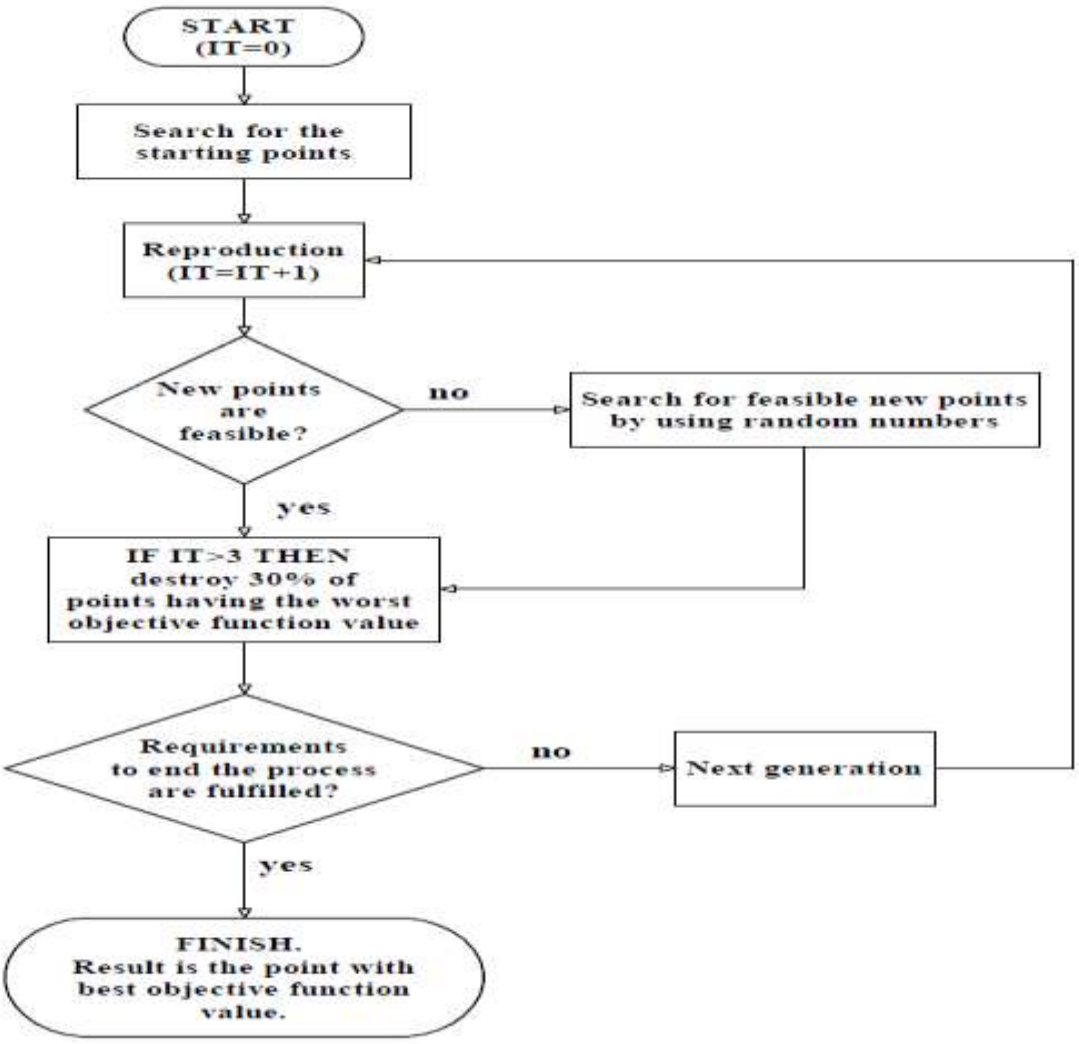

Fig3. The flowchart of the RVA algorithm.

International Journal of Mechatronics, Electrical and Computer Technology

Dec. 2013, Special Number, pp 449-469, **ISSN**: 2305-0543

Available online at: http://www.aeuso.org

6. Simulation Results

NB = 30, ***NL*** = 41, ***NG*** = 6, ***NTR*** = 4 Population size = 50

d_{low} = 0.01, d_{high} = 0.1

Table 1. Optimal Control values

	ARPSO	**PSO**	**RVA**
VG1	1.05	1.06	1.05
VG2	1.03	1.04	1.04
VG3	1.01	1.01	1.02
VG4	1.01	1.02	1.00
VG5	1.07	1.09	1.08
VG6	1.08	1.08	1.08
T_1	0.99	0.98	0.97
T_2	0.95	0.95	0.94
T_3	1.00	1.00	1.00
T_4	0.94	0.93	0.93

Table 2. Parameter sensitivity analysis of IEEE 30 (100 trails)

Method	**Compared item**	**IEEE 30 bus**	**Time (sec)**	**Iterations**
ARPSO	Min. loss	9.4769	8.227	178
	Avg. loss value	9.4792		
PSO	Min. loss	9.4911	12.425	225
	Avg. loss value	9.5001		
RVA	Min. loss	9.4755	8.011	152
	Avg. loss value	9.4714		

International Journal of Mechatronics, Electrical and Computer Technology

Dec. 2013, Special Number, pp 449-469, **ISSN**: 2305-0543

Available online at: http://www.aeuso.org

Table 3. Parameter sensitivity analysis of IEEE 30 (100 trails)

W_{max} / W_{min}		C_i 1.0	1.5	2.0	2.5	3.0
0.9	Avg.	9.4799	9.4827	9.4827	9.4827	9.4827
0.4	Min	9.4769	9.4818	9.4817	9.4817	9.4817
2.0	Avg.	9.4827	9.4826	9.4826	9.4823	9.4825
0.9	Min	9.4818	9.4818	9.4818	9.4817	9.4817
2.0	Avg.	9.4827	9.4827	9.4827	9.4827	9.4827
0.4	Min	9.4819	9.4819	9.4819	9.4819	9.4819

W = Weight function for velocity of agent

C_i = Weight co-efficient for each term

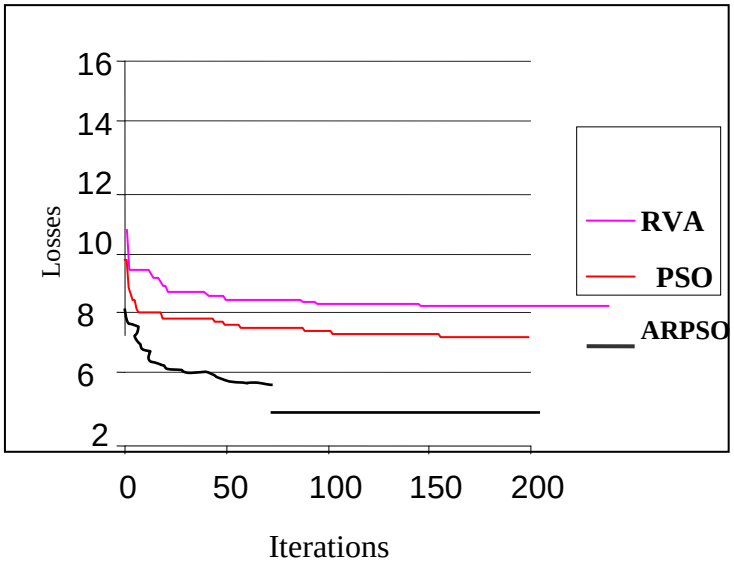

Fig. 4. Comparative study of Convergence characteristics of IEEE 30 Bus system with ARPSO, PSO and RVA.

International Journal of Mechatronics, Electrical and Computer Technology

Dec. 2013, Special Number, pp 449-469, **ISSN**: 2305-0543

Available online at: http://www.aeuso.org

7. Nomenclature

NB = total no. of buses, ***NL*** = total no. of load buses, ***NG*** = total no. of generator buses

TR = total no. of transformers, ***VG*** = generator voltage $\boldsymbol{V_g}$ is a vector of generator bus voltages.

$\boldsymbol{Q_s}$is a vector of switchable VAR sources and

T is a vector of tap settings of on-load tap changing (OLTC) of transformers.

Q_g is a vector of reactive power generations of the generator buses and V_L is a vector of load bus voltages.

Conclusion

In this paper ARPSO and RVA algorithm has been developed for determination of global optimum solution for reactive power optimization problem. The performance of the proposed algorithm demonstrated through its evaluation on IEEE 30 bus power system shows that RVA is able to undertake global search with a fast converges rate and a future of robust computation. From the simulation study it has been found that RVA converges to the global optimum than PSO and ARPSO.

References

[1] R.K.Ursem, 'Diversity guided evolutionary algorithms', in submission for the problem solving from nature conference, (PPSN 7)

[2] Y.Fukuyama, 'A Particle Swarm Optimization for Reactive Power and voltage control in Electric Power System', IEEE Trans. on Power Systems, pp. 87 – 93, 2001.

[3] Jakob S. Vesterstrom and Jacques Riget, ' A Diversity-Guided Particle Swarm Optimizer the ARPSO', EVALife Technical report no. 2002.

[4] Y. Fukuyama, et al., "Practical Distribution StateEstimation Using Hybrid Particle Swarm Optimization", Proc. of IEEE Power Engineering Society Winter Meeting, Columbus, 2001.

International Journal of Mechatronics, Electrical and Computer Technology

Dec. 2013, Special Number, pp 449-469, **ISSN**: 2305-0543

Available online at: http://www.aeuso.org

[5] R. C. Eberhart and Y. Shi, "Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization", Proc. of CEC 2000.

[6] Youxin Luo , "Optimization for PID control parameters on hydraulic servo control system based on random virus algorithm" advance computer control (ICACC) 2 international conference vol 3 pp 432- 435 march 2010 .

[6] K.Y Lee ,Y.M Park , and J.L Oritz, "Fuel –cost optimization for both real and reactive power dispatches" , IEE Proc; 131C,(3), pp.85-93.

[7] M.K. Mangoli, and K.Y. Lee, "Optimal real and reactive power control using linear programming" , Electr.Power Syst.Res, Vol.26, pp.1-10,1993.

[8] S.R.Paranjothi ,and K.Anburaja, "Optimal power flow using refined genetic algorithm", Electr.Power Compon.Syst , Vol. 30, 1055-1063,2002.

[9] D. Devaraj, and B. Yeganarayana, "Genetic algorithm based optimal power flow for security enhancement", IEE proc-Generation.Transmission and. Distribution; 152, 6 November 2005.

[10] Luo, F.F., Chen, G.L., Guo, W.Z.: An improved 'fish-search' algorithm for information retrieval. In: Proceedings of IEEE International Conference on Natural Language, Processing and Knowledge Engineering (NLP-KE 2005), Wuhan, China, pp. 523–528 (2005)

[11] Carmelo J.A.Bastos Filho , Fernando B. de Lima Neto , Anthony J.C.C.Lins , Antonio I.S. Nascimento , Marilia P.Lima , Fish school search nature – inspired algorithms for optimization studies in computational intelligence vol 193 , 2009 , pp 261-277

[12] C.A. Canizares , A.C.Z.de Souza and V.H. Quintana , " Comparison of performance indices for detection of proximity to voltage collapse ,'' vol. 11. no.3 , pp.1441-1450, Aug 1996

[13] B.Gao ,G.K Morison P.Kundur 'voltage stability evaluation using modal analysis ' Transactions on Power Systems ,Vol 7, No .4 ,November 1992.

[14] D. Dasgupta, Artificial Immune Systems and Their Applications, Springer, Berlin, 1999

[15] Cayzer S and Aickelin U (2002), A Recommender System based on the Immune Network, in Proceedings CEC2002, pp 807-813, Honolulu, USA.

[16] Cayzer S and Aickelin U (2002b), On the Effects of Idiotypic Interactions for Recommendation Communities in AIS, Proceedings 1st International Conference on AIS, pp 154-160, Canterbury, UK.

International Journal of Mechatronics, Electrical and Computer Technology

Dec. 2013, Special Number, pp 449-469, **ISSN**: 2305-0543

Available online at: http://www.aeuso.org

Appendix

Table 4. Line data – 30-bus system

Branch No.	From	To	R (p.u)	X (p.u)	Y/2 (p.u)	Line phase angle limit (deg.)
1	2	1	0.0192	0.0575	0.0264	5.0
2	1	3	0.0452	0.1852	0.0204	17.0
3	2	4	0.0570	0.1737	0.0184	7.0
4	3	4	0.0132	0.0379	0.0042	3.5
5	2	5	0.0472	0.1983	0.0209	15.0
6	2	6	0.0581	0.1763	0.0187	7.0
7	4	6	0.0119	0.0414	0.0045	2.5
8	5	7	0.0460	0.1160	0.0102	5.5
9	6	7	0.0267	0.0820	0.0085	6.0
10	6	8	0.0120	0.0420	0.0045	2.0
13	9	11	0.0000	0.2080	0.0000	4.0
14	9	10	0.0000	0.1100	0.0000	4.0
16	12	13	0.0000	0.1400	0.0000	4.0
17	12	14	0.1231	0.2559	0.0000	5.0
18	12	15	0.0662	0.1304	0.0000	3.0
19	12	16	0.0945	0.1987	0.0000	4.0

International Journal of Mechatronics, Electrical and Computer Technology

Dec. 2013, Special Number, pp 449-469, **ISSN**: 2305-0543

Available online at: http://www.aeuso.org

20	14	15	0.2210	0.1997	0.0000	2.5
21	16	17	0.0824	0.1932	0.0000	2.0
22	15	18	0.1070	0.2185	0.0000	2.5
23	18	19	0.0639	01292	0.0000	1.5
24	19	20	0.0340	0.0680	0.0000	3.0
25	10	20	0.0360	0.2090	0.0000	4.0
26	10	17	0.0324	0.0845	0.0000	2.0
27	10	21	0.0348	0.0749	0.0000	2.0
28	10	22	0.0727	0.1499	0.0000	2.0
29	21	22	0.0116	0.0236	0.0000	1.5
30	15	23	0.1000	0.2020	0.0000	3.0
31	22	24	0.1150	0.1790	0.0000	3.5
32	23	24	0.1320	0.2700	0.0000	3.0
33	24	25	0.1885	0.3292	0.0000	2.0
34	25	26	0.2544	0.3800	0.0000	2.0
35	25	27	0.1093	0.2087	0.0000	2.5
37	27	29	0.2198	0.4153	0.0000	3.5
38	27	30	0.3202	0.6027	0.0000	5.0
39	29	30	0.2399	0.4533	0.0000	4.5
40	8	28	0.0636	0.2000	0.0214	4.0
41	6	28	0.0169	0.0599	0.0065	3.0

International Journal of Mechatronics, Electrical and Computer Technology

Dec. 2013, Special Number, pp 449-469, **ISSN**: 2305-0543

Available online at: http://www.aeuso.org

Table 5. Transformer data – 30-bus system

Branch No.	From	To	R (p.u)	X (p.u)	Tap	Tap max	Tap min	Tap step
11	6	9	0.0000	0.2080	1.0155	1.1000	0.9000	0.0250
12	6	10	0.0000	0.5560	0.9629	1.1000	0.9000	0.0250
15	4	12	0.0000	0.2560	1.0129	1.1000	0.9000	0.0250
36	28	27	0.0000	0.3960	0.9581	1.1000	0.9000	0.0250

Authors

K. Lenin has received his B.E., Degree, electrical and electronics engineering in 1999 from university of madras, Chennai, India and M.E., Degree in power systems in 2000 from Annamalai University, TamilNadu, India. At present pursuing Ph.D., degree at JNTU, Hyderabad,India.

Bhumanapally. RavindhranathReddy, Born on 3rd September,1969. Got his B.Tech in Electrical & Electronics Engineering from the J.N.T.U. College of Engg., Anantapur in the year 1991. Completed his M.Tech in Energy Systems in IPGSR of J.N.T.University Hyderabad in the year 1997. Obtained his doctoral degree from JNTUA,Anantapur University in the field of Electrical Power Systems. Published 12 Research Papers and presently guiding 6 Ph.D. Scholars. He was specialized in Power Systems, High Voltage Engineering and Control Systems. His research interests include Simulation studies on Transients of different power system equipment.

International Journal of Mechatronics, Electrical and Computer Technology

Dec. 2013, Special Number, pp 449-469, **ISSN**: 2305-0543

Available online at: http://www.aeuso.org

M. Surya Kalavathi has received her B.Tech. Electrical and Electronics Engineering from SVU, Andhra Pradesh, India and M.Tech, power system operation and control from SVU, Andhra Pradesh, India. she received her Phd. Degree from JNTU, hyderabad and Post doc. From CMU – USA. Currently she is Professor and Head of the electrical and electronics engineering department in JNTU, Hyderabad, India and she has Published 16 Research Papers and presently guiding 5 Ph.D. Scholars. She has specialised in Power Systems, High Voltage Engineering and Control Systems. Her research interests include Simulation studies on Transients of different power system equipment. She has 18 years of experience. She has invited for various lectures in institutes.