EPSO - Evolutionary Particle Swarm Optimization, a New Algorithm with Applications in Power Systems

Vladimiro Miranda   and   Nuno Fonseca

Abstract – **This paper presents a new optimization model – EPSO, Evolutionary Particle Swarm Optimization, inspired in both Evolutionary Algorithms and in Particle Swarm Optimization algorithms. The fundamentals of the method are described, and an application to the problem of Loss minimization and Voltage control is presented, with very good results.**

Keywords: **Evolutionary Algorithms, Particle Swarm Optimization, Voltage/Var control**

I. INTRODUCTION

This paper has the main objective of introducing to the Power System community a new variant in the meta-heuristic set of tools, which we've called EPSO – Evolutionary Particle Swarm Optimization. EPSO is a general-purpose algorithm, whose roots are in Evolutions Strategies and in Particle Swarm Optimization concepts.

Under the name of Evolution Strategies (ES) [1] and [2] and Evolutionary Programming [3], a number of models have been developed that rely on Darwinist selection to promote progress towards the (unknown) optimum. This is a successful technique.

The Particle Swarm Optimization is an optimization algorithm that was introduced in 1995 by Kennedy and Eberhart [4]. We will refer to it as Classic PSO or simply PSO.

The inventors of PSO have since the beginning recognized a proximity between the concepts of PSO and ES. However, no further consequence has been extracted from that idea and, instead, they have concentrated their efforts in explaining their model in terms of "particle movement".

Other attempts have been made to match together evolutionary and particle swarm concepts. It is only fair to give credit to approaches to build bridges between PSO and the world of Evolutionary Computing, such as in [5], or to give an adaptive flavor to a Swarm-type algorithm, such as in [6]. Also in the Power System world some authors have tried such blending with success, and one must give them recognition for that [7][8].

In this paper we present a new variant, EPSO, that is a true hybrid ES/PSO, but with a full Evolutionary interpretation. In fact, we see the method a self-adaptive evolutionary algorithm where we have replaced the operation *recombination* by a new operation called *particle movement*.

This operator *particle movement* seems to be more effective than recombination in generating solutions that approach the optimum. This explains the effectiveness of the model.

Also, in terms of Particle Swarm interpretation, we have given to the model self-adaptive characteristics, so that it no longer depends on external definition of weights or parameters. Because it is self-adaptive, EPSO becomes more robust than Classical PSO and more insensitive to parameter initialization.

As an extra touch, we have added to Classical PSO concept a slight addition in concept, by defining a blurred target instead of a single point, which also improved the quality of the results.

The paper presents the basic concepts of Evolution Strategies and of PSO and proceeds to the description of EPSO. In the last sections we present the application of EPSO to the Voltage/Var control problem in Power Systems and some comments on the comparison of its performance with a simulated annealing model used by some utilities.

The interest of the application lies not only in the demonstration that EPSO can solve such problem, but also that EPSO can deal effectively with a diversity of objective functions, namely of the min-max type. The min-max or max-min functions are more difficult to deal with than linear or quadratic functions, and they are extremely important in several Power System contexts, such as in security problems or in risk analysis problems.

V. Miranda is with INESC Porto – Institute of Engineering in Systems and Computers at Porto, Portugal, and also with FEUP – Faculty of Engineering of the University of Porto (email: vmiranda@inescporto.pt).

N. Fonseca is also with INESC Porto, Portugal (email: nfonseca@power.inescporto.pt)

II. BRIEF REVIEW OF CLASSICAL PSO

In the classical PSO, one must have, at a given iteration, a set of solutions or alternatives called "particles". From one iteration to the following, each particle X_i moves according to a rule that depends on three factors, as follows.

In order to understand this rule, one must also keep record of the best point b_i found by the particle in its past life and the current global best point b_g found by the swarm of particles in their past life.

The movement rule states that

$$X_i^{new} = X_i + V_i^{new}$$

where V_i is called the particle i velocity and is defined by

$$V_i^{new} = Dec(t)w_{i0}V_i + Rnd_1 w_{i1}(b_i - X_i) + Rnd_2 w_{i2}(b_g - X_i)$$

where the first term of the summation represents inertia or habit (the particle keeps moving in the direction it had previously moved), the second represents memory (the particle is attracted to the best point in its trajectory) and the third represents cooperation or information exchange (the particle is attracted to the best point found by all particles).

The parameters w_{ik} are weights fixed in the beginning of the process. Rnd_x are random numbers sampled from a uniform distribution in [0,1]. $Dec(t)$ is a function decreasing with the progress of iterations, reducing progressively the importance of the inertia term [9]. Figure 1 illustrates the concept.
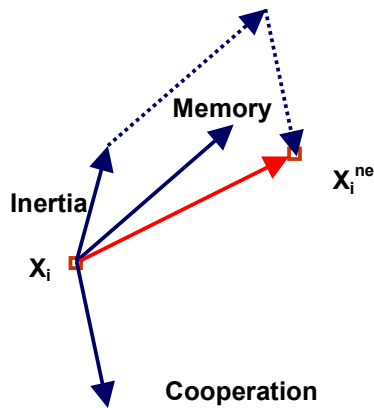

Figure 1 – Illustrating the movement of a particle, influenced by three terms.

III. BRIEF REVIEW OF EVOLUTION STRATEGIES

In the σSA-ES (self-adaptive Evolution Strategies) model, one must have, at a given iteration called generation, a set of solutions or alternatives called individuals. Each individual is characterized by object parameters (the values of the variables describing a solution) and by strategic parameters (mutation rates for each variable, mutation correlation angles and similar).

Although there are many variants, we will describe the following procedure for reasons that will become obvious later on:

- Each individual is duplicated

- The strategic parameters of each individual undergo mutation

- The object parameters of each individual are mutated under a procedure commanded by its strategic parameters (this generates new individuals)

- A number of individuals undergo recombination (this generates new individuals)

- From the set of parents and sons (the original and the new individuals), the best fit are selected to form a new generation

The selection procedure has a number of variants and can be ruled by a stochastic tournament or be purely deterministic, involve elitism, niching, etc. There is an interesting theoretical building providing insight on why ES achieve convergence and how a near optimal progress rate is achieved [10].

IV. CRITICIZING PSO

The most striking point of classical PSO is the fact that it depends of a number of parameters defined externally by a user, and most certainly with values that are problem dependent.

This is certainly true for the definition of the weights w_{ik}, and our experience seems to be in agreement with other authors: a delicate work of tuning the algorithm is often necessary, in practical problems.

Furthermore, the external definition of the decreasing function $Dec(t)$ is also something that can only leave on with a feeling of discomfort. It is intuitive that if the inertia term is eliminated at an early stage of the process, then the algorithm risks to be trapped at some local minimum. Therefore, some more tuning is needed. To avoid this kind of problem, some authors have suggested procedures of "re-seeding" the search by generating new particles at distinct places of the search space.

Last, the random factors Rnd_k, while introducing an useful stochastic flavor, only have a heuristic basis and are not sensitive to the evolution of the process.

Observing PSO as an proto-evolutionary process, we may say that:

- We have in PSO a mechanism to reproduce and generate new individuals from a previous set (the movement rule) – i.e., we see the new iteration not as a movement of particles but as a generation of new alternatives in different positions in space

- But we do not have an explicit selection mechanism in the Darwinist sense; however, the algorithm exhibits a

positive progress rate because the movement rule induces such property implicitly.

V. EPSO –EVOLUTIONARY SELF-ADAPTING PSO

The idea behind EPSO [11] is to grant a PSO scheme with an explicit selection procedure and with self-adapting properties for its parameters. The variables in an EPSO formulation are divided, according to the vocabulary used in the Evolution Strategies community, composed of *object parameters* (the X variables) and *strategic parameters* (the weights w).

At a given iteration, consider a set of solutions or alternatives that we will keep calling particles. A particle is a set of object and strategic parameters [**X, w**]. The general scheme of EPSO is the following:

<u>REPLICATION</u> - each particle is replicated r times

<u>MUTATION</u> - each particle has its weights w mutated

<u>REPRODUCTION</u> - each mutated particle generates an offspring according to the *particle movement* rule

<u>EVALUATION</u> - each offspring has its fitness evaluated

<u>SELECTION</u> - by stochastic tournament the best particles survive to form a new generation

The *particle movement* rule for EPSO is the following: given a particle $\mathbf{X}_i$, a new particle $\mathbf{X}_i^{new}$ results from

$$\mathbf{X}_i^{new} = \mathbf{X}_i + \mathbf{V}_i^{new}$$

$$\mathbf{V}_i^{new} = w_{i0}^* \mathbf{V}_i + w_{i1}^*(\mathbf{b}_i - \mathbf{X}_i) + w_{i2}^*(\mathbf{b}_g^* - \mathbf{X}_i)$$

This formulation is very similar to Classical PSO – the movement rule keeps its terms of inertia, memory and cooperation. However, the weights, taken as object parameters, undego mutation

$$w_{ik}^* = w_{ik} + \tau N(0,1)$$

where $N(0,1)$ is a random variable with Gaussian distribution, 0 mean and variance 1.

The global best $\mathbf{b}_g$ is randomly disturbed to give

$$\mathbf{b}_g^* = \mathbf{b}_g + \tau' N(0,1)$$

The logic behind this modification of the follwing: a) if the current global best is already the global optimum, this is irrelevant; but b) if the optimum hasn't yet been found, it may nevertheless be in the neighborhood and it makes all sense not to aim *exactly* at the current global best – especially when the search is already focused in a certain region, at the latter stages of the process.

The τ,τ' are learning parameters (either fixed or treated also as strategic parameters and therefore subject to mutation).

This scheme benefits from two "pushes" in the right direction: the Darwinistic process of selection and the particle movement rule; therefore, it is natural to expect that it may display advantageous convergence properties when compared to ES or PSO alone. Furthermore, EPSO can also be classified as a self-adaptive algorithm, because it relies on the mutation and selection of strategic parameters, just as any σ-SA Evolution Strategy.

VI. TESTING *EPSO*

We have extensively tested EPSO vs. PSO in solving classical difficult test problems. Some results may be found in [11]. For illustration purposes, we only present a result obtained with the difficult Rosenbrock function:

$$f_2(x) = \sum_{i=1}^{n} (100 \times (x_{i+1} - x_i^2)^2 + (x_i - 1)^2) \quad (6)$$

The test was conducted in a space of n=30 dimensions, in the domain $[0, 30]^n$. Figure 2 illustrates the process of convergence in terms of number of evaluations, to establish a fair comparison in terms of computer effort.

The EPSO trials were done by setting the replication index r equal to 2 (a particle give origin to two descendents, one of them mutated in its strategic parameters, and this offspring is subject to stochastic tournament selection). PSO requires only one evaluation per particle per iteration, but EPS requires r evaluations per particle per iteration.
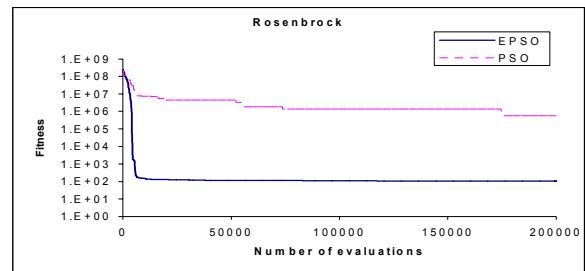

Figure 2 - Typical convergence in the Rosenbrock function (notice that the scale is logarithmic).

The results presented in Figure 2 shows a clear superiority of the EPSO algorithm. In order to have the PSO results optimized one needs to tune by hand the strategic parameters but EPSO was able to provide better results independently of the strategic parameter initialization.

VII. APPLICATION OF *EPSO* TO VOLTAGE/VAR CONTROL

7.1 Loss reduction in distribution systems

We have compared the performance of EPSO with a Simulated Annealing model developed at INESC Porto [12][13]. The test was conducted over the system defined in [14], with 24 nodes/36 branches, including 31 transmission lines, 5 transformers, 11 capacitor banks and 9 synchronous generators.

This Simulated Annealing algorithm is a well tested application developed by INESC Porto. We have used it for

comparison as it is included in a commercial DMS and is currently used by a number of utilities.

In general terms, the problem of Voltage/Var control can be formulated as follows:

Minimize $\quad \varphi(u, x)$ $\quad\quad\quad$ (9)

Subject $\quad\quad H(u, x, p) \le 0$ $\quad\quad$ (10)

Equation (9) is the objective function of this problem and, in general, represents the active losses, calculated from the power flow equations. The constraints of this problem (11) represent the operation limits, namely bands of admissible voltage values at nodes.

In this model, we have dealt with these constraints with a penalty approach, in the tradition of ES.

We may have also other constraints in (10), representing the preference for keeping control margins, i.e., searching for solutions that do not require the set points of controllers to be at their maxima or minima.

Furthermore, we took some of the control variables u as constrained to a number of discrete values, corresponding, for instance, to discrete transformer taps.

The Voltage/Var control problem in distribution systems is usually a problem of minimizing losses and controlling voltage levels, by acting on transformer taps and on capacitor bank taps. It is rare to find synchronous generators directly connected to the network where one could act on their excitation. However, EPSO can deal with these variables as well, with excellent results.

7.2 Results of the loss reduction problem

In order to compare EPSO results with those obtained with the Simulated Annealing algorithm, the stopping criterion has been defined for both approaches as running 270 iterations without objective function improvement.

The test was conducted by manipulating only the discrete controls (transformer and capacitor bank taps). There is a version of Discrete PSO [15]; in EPSO, we decided to deal only with continuous variables and use "probabilistic rounding" obtain an integer solution.

Instead of using simple rounding each variable to the nearest discrete feasible value, we've considered that the probability of rounding to the nearest discrete value increases as the distance decreases and, therefore, there is always a probability that the unfeasible value may be rounded to a more distant value. This scheme proved to be very satisfactory.

In terms of convergence comparison between the both algorithms we can immediately reveal that:

- EPSO finds its best solution in less iterations.

- the initial solution is better for EPSO, because it has a population of particles, while the Simulated Annealing only starts with one initial solution.

- there is an extra computing effort in applying EPSO, when compared to the Simulated Annealing option (measured in the number of load flows run);

- EPSO discovers better solutions than the Simulated Annealing algorithm.
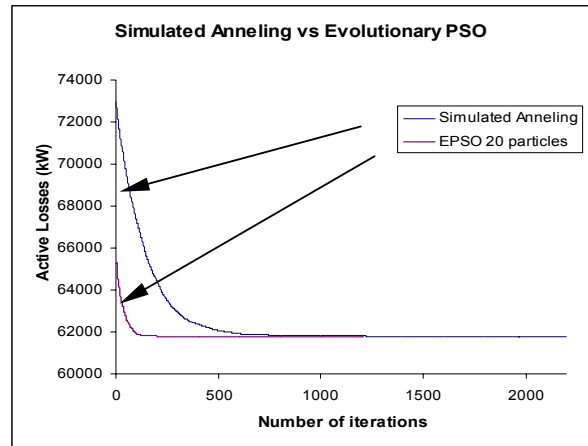

Figure 3 - Comparison between convergence in both EPSO and Simulated Annealing.

A typical convergence pattern observed for both algorithms can be observed in Figure 3.

The quality of results of algorithms with stochastic convergence must be measured by the average value they reach, in a number of runs, because a single run may be dependent of the random initialization of solutions.

But this indicator is not enough – we need to assess also the robustness of the algorithm, which may be evaluated by the variance of the results. A large dispersion of results in a number of runs would not give confidence to users of a given method.

The following table compares the performance of EPSO with 10 and with 20 particles, against the Simulated Annealing approach.

	Average losses (MW)	Std. Deviation (kW)
EPSO 10 particles	61.789	1.966
EPSO 20 particles	61.788	1.485
Simulated Annealing	61.792	9.812

Table 1: Comparison of EPSO with Simulated Annealing

As we can see in Table 1, EPSO reveals superiority in terms of the solution found (evaluated by the average optimum obtained in 1000 runs) and in terms of its robustness (evaluated as the root of the mean square error, or standard deviation, relative to the best solution found).

In fact, EPSO gives consistently a near-optimum result, while the Simulated Annealing model failed many times to

reach a solution as good (and that's why the dispersion of results in this case is much larger than with EPSO).

Therefore, EPSO is a much more reliable algorithm for practical applications.

7.3 Voltage control with feasible solution

For this test we've increase the reactive load in bus 8 of the same IEEE 24-bus system. The voltage at this bus became very low and we run the EPSO algorithm to re-dispatch the reactive power in order to set the voltage back inside the limits.

As it can be seen in Figure 4, the algorithm was able to find a new set point, to both transformers and capacitor banks, which forced the voltage into the acceptable limit (0.9 – 1.1 p.u.).

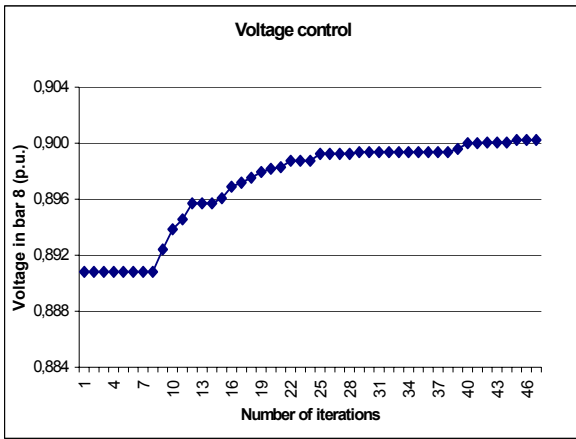The Simulated Annealing algorithm failed to obtain a feasible solution for this case.

Figure 4 - Voltage Control with EPSO – evolution of the controlled voltage along the iterations of an EPSO algorithm.

7.4 Voltage control with unfeasible solution

In this test we've loaded the system to such a degree that in the initial state voltages would fall below the lower bound of the admissible band and no correcting solution would be found respecting all the nodal voltage limits.

If no solution satisfies all constraints, some criterion must be specified in order to select an acceptable alternative. We have tested EPSO/20 particles running under several criteria; having

$$\begin{cases} \Delta V_i = V_{min} - V_i & \text{if} \quad V_i < V_{min} \\ \Delta V_i = V_i - V_{max} & V_i > V_{max} \end{cases}$$

we've run EPSO under the following objective functions:

a) minimizing the sum of voltage deviations outside the admissible voltage band;

$$\min \sum_{i=1}^{n} \Delta V_i \; ; \; n \text{ - no. of PQ buses}$$

b) minimizing the sum of the square of the deviations;

$$\min \sum_{i=1}^{n} (\Delta V_i + 1)^2$$

c) minimizing the maximum deviation relative to the admissible voltage band;

$$\text{Min} \; \{\text{Max}(\Delta V_i), i = 1,...,n\}$$

The two first criteria could seem "natural" mathematical criteria; however, from an operational point of view, the third criterion may be preferable: if one cannot bring all voltages to the admissible band, one would not like to have a "too bad" bus voltage either – and the min-max criterion takes care of that.

Figure 5 illustrates the initial voltage condition and the solutions obtained with the three criteria, under a specification of a lower admissible voltage limit of 0.95 p.u.. EPSO could work as well with all criteria, including the min-max, and produce high quality solutions. Notice that, with the mim-max criterion, voltage at the worst bus (bus 10) has been improved up to a point where another bus (bus 5) would replace it as the one with the worst voltage value.
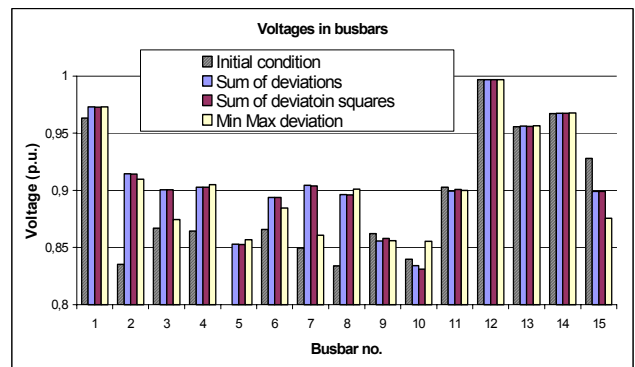

Figure 5 – Bus voltages for the initial condition and as a result of EPSO under three distinct criteria

VIII. CONCLUSIONS

This paper reports two important results:
- A new meta-heuristic variant, with roots both in Evolutionary Computing and in Particle Swarm algorithms – EPSO.
- A new model for loss minimization and voltage control

First of all, there is a new successful optimization tool, available for optimization of complex problems with multiple local optima – EPSO, the Evolutionary Particle Swarm Optimization method. EPSO joins together the characteristics of Evolutionary Algorithms and of Particle Swarm Algorithms.

From an Evolutionary Computing point of view, there is a new operator introduced in the replication phase – the *particle movement* , which generates new (and promising) solutions in the search space. This operator is used instead of recombination and one has observed an improvement in the progress rate towards the optimum of the algorithm, compared with Classical PSO.

From a Particle Swarm point of view, there is a self-adaptive tuning of the algorithm by evolutionary adjustment of the parameters controlling particle movement.

Both points of view are legitimate and justify the improved convergence characteristics of the method.

The second important result is that EPSO proves very successful in solving the Power System optimization problem of voltage control and loss minimization in a power system. In fact, EPSO performed better than a Simulated Annealing model that has been used by utilities, both in the quality of the solution discovered and in the robustness of the result (dispersion around the best result, found in a number of repeated runs).

An important result is that EPSO could find adequate solutions under a min-max criterion. Power System models seldom have used such criterion, because optimization under a L-∞ metric is much more difficult that with a linear or quadratic objective function. However, in many problems such a criterion is an alternative more adjusted to human reasoning: in the area of security (maximizing the minimum distance to a security border), in planning (minimizing the maximum regret of an expansion strategy), in distribution reliability (minimizing the worst local index from a set of consumers).

There are many experiments to be performed under the EPSO paradigm, following the trends already explored with PSO and ES, such as multiple swarms, niching, elitist or non elitist selection and so on. The authors hope that the material in this paper may serve as a source of inspiration for future research work.

REFERENCES

[1] Rechenberg, I. (1973), "Evolutionsstrategie-Optimierung technischer Systeme nach Prinzipen der biologischen Evolution", Frommann-Holzboog, Stuttgart

[2] Schwefel, H.-P. (1995), "Evolution and Optimum Seeking", Ed. Wiley, New York NY

[3] Fogel, D.B. (1992), "*Evolving Artificial Intelligence*", Ph.D. Thesis, University of California, San Diego

[4] Kennedy, J., R.C. Eberhart, "Particle Swarm Optimization", IEEE International Conference on Neural Networks, Pert, Australia, IEEE Service Center, Piscataway, NJ., 1995

[5] Lovberg, M. et al. (2001), "Hybrid Particle Swarm Optimiser with Breeding and Sub-populations", Proceedings of GECCO2001 - Genetic and Evolutionary Computation Conference 2001(pp. 409), July 7-11 2001, San Francisco CA, USA

[6] Clerc, M. (1999), "The Swarm and the Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization", Proceedings of the 1999 Congress of Evolutionary Computation, vol. 3, 1951-1957, IEEE Press

[7] Fukuyama, Y. and Yoshida, H., 'A particle swarm optimization for reactive power and voltage control in electric power systems,' IEEE Proc. of Evolutionary Computation 2001 , Vol.1 , pp. 87 -93, 2001.

[8] Yoshida, H., Fukuyama, Y., Takayama, S. and Nakanishi, Y., 'A particle swarm optimization for reactive power and voltage control in electric power systems considering voltage security assessment,' IEEE Proc. of SMC '99 , Vol. 6, pp.497 -502, 1999.

[9] Shi, Y., Eberhart, R.C., "Parameter Selection in Particle Swarm Optimization", Proceedings of the Seventh Annual Conference on Evolutionary Programming, 1998.

[10] Beyer, H.-G. (1996), "Toward a Theory of Evolution Strategies: Self-Adaptation", in *Evolutionary Computation*, vol. 3, no. 3, pag. 311-347

[11] Miranda, V., Fonseca, N., "New Evolutionary Particle Swarm Algorithm (*EPSO*) Applied to Voltage/Var Control", Proceedings of PSCC'02 – Power System Computation Conference, Sevilla, Spain, June 24-28, 2002.

[12] Pereira, J., Saraiva, J.T., Ponce de Leão, M.T., "Identification of Operation Strategies of Distribution Networks Using a Simulated Annealing Approach", Proceedings of IEEE Budapest Power Tech'99, paper BPT99-357-17, August 1999.

[13] Matos, M., Ponce de Leão, M.T, Saraiva, J.T., Fidalgo, J.N., et al., "Meta-heuristics Applied to Power Systems", Proceedings of MIC´2001 - 4th Metaheuristics International Conference, Porto, Portugal, vol.2, pp.483-488, July, 2001.

[14] Reliability Test System Task Force of the Application of Probability Methods Subcommittee, "IEEE Reliability Test System ", IEEE Trans. On Power Apparatus and Systems, vol. PAS-98, no. 6, Nov./Dec. 1979.

[15] Kennedy, J. and Eberhart, R. C., "A discrete binary version of the particle swarm algorithm", Proc. Conf. on Systems, Man, and Cybernetics, 4104–4109. Piscataway, NJ: IEEE Service Center, 1997.