Eng Int Syst (2007) 1: 29–35


**Engineering**
*Intelligent Systems*

# A multiple scenario security constrained reactive power planning tool using EPSO

**Hrvoje Keko[1], Student Member IEEE, Álvaro Jaramillo Duque[1] and Vladimiro Miranda[1,2], Fellow IEEE**

[1]INESC Porto–Institute of Engineering in Systems and Computers at Porto, Portugal,
(e-mail: hkeko@inescporto.pt, aduque@inescporto.pt)
[2]FEUP – Faculty of Engineering of the University of Porto
(e-mail: vmiranda @inescporto.pt).

Evolutionary Particle Swarm Optimization (EPSO) is a robust optimization algorithm belonging to evolutionary methods. EPSO borrows the movement rules from Particle Swarm Optimization (PSO) and uses it as a recombination operator that evolves under selection. This paper presents a reactive power planning approach taking advantage of EPSO robustness, in a model that considers simultaneously multiple contingencies and multiple load levels. Results for selected problems are summarized including a trade-off analysis of results.

Keywords: Reactive power planning, evolutionary algorithms, particle swarm optimization.

## 1. INTRODUCTION

Reactive power planning (RPP) belongs to the most complex problems of power system optimization. It can be defined as determining the amount and location of reactive power compensation devices to be purchased within a planning period. While trying to keep investment costs as low as possible over the planning period, the voltage profile should be adequate and also energy losses in the network should be minimized. These criteria are opposed so the optimal solution becomes a trade-off between these criteria. While the reactive power control problem gives the optimal set of control variables [1], reactive power planning results also include the most appropriate locations to install the reactive power compensation devices and their types and sizes.

Since the load levels in an electric power system vary significantly, the control variables need also to be determined within the reactive power planning process. This way the voltage profile can be kept within adequate limits and voltage collapse avoided. Besides the amount of reactive power compensated by newly installed devices, the control variables for RPP include setting transformer taps and voltage on voltage-regulated buses. These control variables need to be included in the reactive power planning problem formulation.

Reactive power planning is a nonlinear optimization problem with many uncertainties. Conventional calculusbased optimization algorithms have been applied to this problem; however, many conventional optimization techniques are prone to delivering local minima solutions. Having this in mind, a number of techniques relying on artificial intelligence have been applied to this problem. In particular, evolutionary algorithms offered new tools for its optimization [2], [3]. In general, the need for tuning the algorithm parameters by users with expertise in power system is preferably avoided, so in real life applications adaptive methods capable of self-tuning are sought.

The Evolutionary Particle Swarm Optimization algorithm (EPSO) [4] is an algorithm with heritage from both evolutionary algorithms and particle swarm algorithms. It can either be viewed either as PSO with evolving weights or as an evolutionary algorithm with a movement rule borrowed from PSO. EPSO has already proven to be efficient, accurate and robust, therefore applicable to power system problems [5–8]. This paper shows how we were able to build an application that proposes decisions on the installation of capacitor banks, FACTS devices or other means of reactive compensation, taking in account not only a basic system scenario but also selected contingency scenarios associated with given probabilities. This multiple scenario approach is in general hard in computational terms but we were successful by taking advantage of EPSO robustness of convergence.

## 2. HERITAGE AND OUTLINE OF EPSO ALGORITHM

Despite observed advantages over classic techniques, evolutionary algorithms also fail sometimes in fine-tuning of final solutions. This is the reason for development of hybrid methods – methods that take advantage of the excellent search capabilities of evolutionary algorithm and the advantages of classic methods in exploring the search space in the proximity of a solution found by an evolutionary algorithm [9].

Besides "classic" evolutionary methods, researchers have been further inspired in nature's mechanisms. One example is Particle Swarm Optimization (PSO), inspired in the collective movement of flocks of birds, schools of fish or swarms of bees. PSO was presented in 1995 by James Kennedy and Russell Eberhart in [10]. It was discovered through simulation of a simplified social model. The originality of this algorithm relies on interchanging of information about the location of currently found best position, while particle movement is modeled with simple movement equation.

EPSO – Evolutionary Particle Swarm Optimization algorithms, is a method with heritage from both evolutionary algorithms and particle swarm optimization. It borrows the movement rules from PSO methods and uses them as a recombination operator that evolves under pressure of selection. In an EPSO algorithm, given a particle (a point in search space) in generation $k$, particle in the following generation $k+1$ is reproduced from that particle using the following movement rules.

$$\mathbf{x}_i^{(k+1)} = \mathbf{x}_i^{(k)} + \mathbf{v}_i^{(k+1)} \tag{1}$$

$$\mathbf{v}_i^{(k+1)} = w_{i1}^{*}\mathbf{v}_i^{(k)} + w_{i2}^{*}(\mathbf{b}_i - \mathbf{x}_i^{(k)}) + w_{i3}^{*}\mathbf{P}(\mathbf{b}_g^{*} - \mathbf{x}_i^{(x)}) \tag{2}$$

where the values are

$\mathbf{b}_i$ – best point found by i-th particle itself, in its past life, up to the current generation

$\mathbf{b}_g$ – best point found by the swarm of particles in their past life

$\mathbf{x}_i^{(k)}$ – location of particle $i$ in vector space, generation k

$\mathbf{v}_i^{(k)} = \mathbf{x}_i^{(k)} - \mathbf{x}_i^{(k-1)}$ – velocity of particle $i$ at generation k

$w_{i1}$ – weight conditioning the *inertia* term (the particle tends to maintain previous movement)

$w_{i2}$ – weight conditioning the *memory* term (the particle is attracted to its previous best position)

$w_{i3}$ – weight conditioning the *cooperation* or *information exchange* term (the particle is attracted to the overall best-sofar found by whole swarm).

**P** - communication factor – a diagonal matrix containing value 1 with probability p and value 0 with probability (1-p); the value of *communication probability* p controls the passing of information within the swarm and ensures that the interchange of information between particles is stochastic, i.e. the information (about the best position found) is not always propagated.

In each algorithm step, each particle is replicated a certain number of times. Afterwards, each replica of the particle has its strategic parameters (weights) mutated. All replicas and an original particle generate offspring particles through recombination, according to the particle movement rule described above. The evaluation (calculation of fitness) of each offspring is followed by a selection procedure that ensures the best offspring particles form a new generation.

By mutation and selection the particles *learn* the values of their strategic parameters. In comparison with other adaptive evolutionary methods, EPSO is specific in its adaptive recombination operator while usually the adaptive operator in other methods is the mutation operator. In comparison with classic PSO, EPSO doesn't use an explicit random factor in the movement equation: instead it relies on evolving weights. In real life applications, perhaps the most significant positive characteristic of EPSO is the robustness of convergence. This paper presents the application of EPSO in a flexible planning tool suitable for usage in real applications. The planning application presented in this paper is capable of handling real planning problems by taking into account physical network constraints over arbitrary number of time spans.

## 3. REACTIVE POWER PLANNING PROBLEM DEFINITION

Reactive power planning is a non-smooth, nondifferentiable minimization problem for an objective function that includes operating costs with the purpose of reducing real power losses and improving the voltage profile as well as the allocation cost of additional reactive power sources.

Similar to evolutionary algorithms, EPSO algorithm relies on a fitness function concept. A fitness function $f : \mathbf{x} \to \mathbb{R}$ gives a real-valued assessment of a vector $\mathbf{x}$ (a position in the search space). The fitness function is solely responsible for modeling the particular problem being optimized, and for the RPP problem it is built from three different functions: capacitor investment cost, cost of energy losses and penalization function for violating voltage limits. It also includes penalties associated to the violation of constraints.

There are two factors that may distinguish network scenarios: load levels and structural changes. Load levels are associated with given load profiles and a single structural scenario may encompass several load levels or load scenarios. Each load level is associated with some probability extracted from an annual load curve, usually considering its duration.

Structural scenarios are distinct from one another in the fact that they correspond to a diversity of contingencies (doesn't matter of which order), each contingency associated with a given probability or unavailability. Usually one considers a base case with a high probability value and then contingency scenarios. Therefore, each scenario to consider in the RPP model will correspond to a combination of load and structural scenarios and its probability will be the product of partial scenario probabilities.

The problem is then formulated as:

$$\text{Minimize } OBJ = \sum_{s=1\ldots N_s} \Pr_s \cdot (CC + EC + VL)_s$$

$$\text{subject to } \forall s \begin{cases} f(X,V) = 0 \\ Q_{ci}^{\min} \le Q_{ci} \le Q_{ci}^{\max},\ i \in N_c \\ Q_{gi}^{\min} \le Q_{gi} \le Q_{gi}^{\max},\ i \in N_g \\ T_i^{\min} \le T_i \le T_i^{\max},\ i \in N_T \\ V_i^{\min} \le V_i \le V_i^{\max},\ i \in N_B \end{cases} \tag{3}$$

where

$s$ = index of network scenarios
$\Pr_s$ = probability of scenario s
$CC$ = capacitor investment function
$EC$ = energy cost function
$VL$ = voltage limit penalty function and
$f(X,V) = 0$ - equality constraints of power flow problem
$Q_{ci}$ = reactive power source installation at node $i$
$N_c$ = set of candidate nodes for reactive power installation
$Q_{gi}$ = reactive power generation at node $i$
$N_g$ = set of generator nodes
$T_i$ = transformer tap setting — the transformer winding connected to bus $i$
$N_g$ = set of nodes with transformer windings with tap changers
$V_i$ = voltage magnitude at bus $i$
$NB$ = set of buses

The conditions stated above must hold for each observed load level. Investment costs for each reactive power injection device consist of installation cost and purchase cost:

$$CC = sf_c \left[ \sum_{k \in N_C} C_k = \sum_{k \in N_A} (C_{i,k} + C_{p,k}) \right] \tag{4}$$

where $sf_C$ represents a scaling factor for the investment function, $N_C$ represents candidate buses, $N_A$ the set of nodes where devices may be installed - subset of the set of candidate nodes. $C_{i,k}$ and $C_{p,k}$ represent the cost of installation and purchase of devices for the $k$-th candidate bus.

The algorithm is capable of handling installation and purchase costs of a variety of devices such as capacitor banks or FACTS (for instance, static VAR compensators) and taking in account variations related to specific installation conditions. Existing devices just get zero cost and may be reinforced or ignored at planner's discretion.

Optimal device sizes are varying with load levels and these can vary significantly, so for instance more expensive switched capacitor banks can be used to vary the amount of reactive power produced. The algorithm handles purchase costs for each bank by breaking it down in two components:

$$C_{p,i} = NFC_i \cdot CF_i + NSC_i \cdot CS_i \tag{5}$$

where, for node $i$, $NFC_i$ represents quantity of fixed capacitor banks, $CF_i$ per-unit cost of fixed capacitors, $NSC_i$ quantity of fixed capacitor banks, and finally $CS_i$ per-unit cost of switched capacitors.

The second part of the fitness function takes into account the total power losses in the system, used as the main factor for the energy loss function. For each load level and structural scenario observed, system losses are calculated from power flow results and load level duration. The formulation is

$$EC = sf_E \left[ \sum_{j=1}^{N_L} PL_j \cdot T_j \cdot MC_j \right] \tag{6}$$

where $sf_E$ represents scale factor for energy cost function, $N_L$ number of load levels, $PL_j$ power loss at load level $j$, $T_j$ planning time duration of load level $j$, and $MC_j$ marginal cost of energy for load level $j$. The costs are specified for each load level since energy prices usually have high volatility and significantly vary between peak and off-peak periods.

Total voltage level penalty is a linear combination of factors proportional with linear and squared voltage deviation.

$$\mathrm{VL} = sf_V \left[ \sum_{j=1}^{N_L} \left( \sum_{i \in N_B} SF_{i,j} + sf_M \sum_{i \in N_B} LF_{i,j} \right) \right] \tag{7}$$

where $sf_V$ is a scale factor for the voltage limit penalty function, and $sf_M$ represents a scale factor for the linear penalty function.

For each node and load level, voltage penalization is

$$SF_{i,j} = \begin{bmatrix} (V_{i,j} - V_{iN})^2 & \text{if } V_{i,j} > V_{i,\max} \\ 0 & \text{otherwise} \end{bmatrix}$$
$$LF_{i,j} = \begin{bmatrix} (V_{i,j} - V_{iN})^2 & \text{if } V_{i,j} < V_{i,\min} \\ 0 & \text{otherwise} \end{bmatrix} \tag{8}$$

This way undervoltages and overvoltages are kept in control. Penalties act as soft constraints and influence particle movement in the search space like a repulsive physical force. The scaling of penalties modifies the intensity of the push that forces the particles to leave the search space regions where the network has unacceptable voltage conditions. By configuring the scaling factor for voltage penalization, it is also possible to take legislative implications into account — legislation might deliver financial penalties to companies violating the voltage limits. Increasing the penalty scaling factor ensures that the algorithm strongly prefers solutions with acceptable voltage levels, so the particles are even more vigorously kept in the regions with acceptable voltages.

# 4. EPSO SEARCH SPACE FOR RPP

The application has the following input: a complete network configuration (i.e. nodes and branches of network configurations, for the base case and for contingency cases), the subset of nodes defined as candidate nodes for the installation of devices along with installation limits, sets defining ranges for available transformer tap changers, and finally node voltage limits for all nodes with voltage regulation. These variables are encoded in a search vector space as follows: the first $N_c$ variables represent the reactive power injection in a particular node, ranging from $Q_{ci}^{\min}$ to $Q_{ci}^{\max}$, $N_T$ variables represent settings of transformer taps, ranging from $T_i^{\min}$ to $T_i^{\max}$ and finally $N_{VR}$ variables represent $N_{VR}$ settings for voltage-regulated nodes, ranging from minimum to maximum voltages for each node.

$$X = [Q_{c,1}^1, \ldots, Q_{c,N_C}^1, \ldots, T_1^1 \ldots T_{N_T}^1, V_1^1 \ldots V_{N_{VR}}^1 \ldots Q_{c,1}^{N_L} \ldots Q_{c,N_C}^{N_L}, T_1^{N_L} \ldots T_{N_L}^{N_L}, V_1^{N_L} \ldots V_{N_{VR}}^{N_L}] \tag{9}$$

The number of space dimensions is $N_c + N_T + N_{VR}$. These variables are repeated $N_s$ times in order to represent the situation for each load level at each structural configuration scenario. This means that the algorithm searches for optimal solutions for each period "separately": the reactive power device sizes in different load levels are independent and not correlated until the solution evaluation.

An outline of the fitness function that evaluates the mentioned vector follows. In the first place, new reactive injection devices are inserted into network topology. Afterwards, the second phase sets the transformer taps and voltage levels for voltage-regulated nodes. A Newton- Raphson power flow calculation is performed and energy losses and node voltages become available. The process is repeated for each scenario. Finally, new device sizes are determined: if the adequate size is the same for all periods, then a fixed device (such as a capacitor bank) should be purchased. Otherwise, if the size varies over periods, some fixed size device should be purchased for the minimum determined reactive power needed, accompanied with switched or varying injection devices for the remainder.

The EPSO algorithm, described in Section II, has been implemented in INESC Porto on a Java agent-based platform capable of parallel processing. There are also two singleprocessor implementations, in Java and C++. The results presented in this paper used the C++ based EPSO implementation. This implementation takes advantage of Cbased calculation library capable of quick processing of large network topologies and calculation of power flows. The topology processor library is developed in-house in INESC Porto, and is responsible for Newton-Raphson power flow calculations and maintaining the network configurations in memory. This library also takes care of operational constraints of all network elements.

When modeling a sequence of load levels for a given configuration, one must assure that no unnecessary capacity is added and that a maximum of fixed capacity is in fact used (because it is in general cheaper). However, if an algorithm gets stuck in some local optimum it might yield unnecessary reactive power injections and implicate purchasing of switched

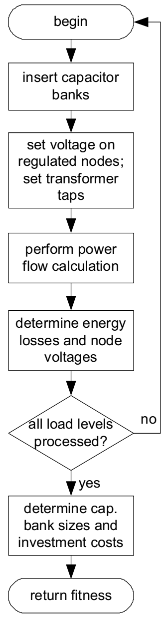


**Figure 1** Outline of reactive power planning fitness function.

devices. Such a case has been observed in [11]. A robustness test for the algorithm is therefore to run it with several scenarios, forcing the multiplication of variables, but leaving the network configurations and load level unchanged. The results should propose only fixed-type reactive injection devices. EPSO has successfully passed this test which confirms the validity of previous assumption. This test along with other tests is presented in the following chapter.

# 5. TEST CASES

In this Section we show results from the application of the tool to the IEEE 24-bus and IEEE 30-bus test systems. We chose the IEEE 24-bus transmission network (known as the RTS – Reliability Test System) since it allows the inclusion of all types of control variables that the algorithm implementation can handle. This possibility is seen as particularly useful in the context of MV networks with distributed generation, where one finds not only a multitude of potential locations for reactive power injection (namely at distributed generation buses) but also the electrical characteristics of meshed networks.

In all tests, we have used swarms with 30 individuals (or particles) and communication probability of 0.2. The stopping criterion adopted was to halt iterations after 2000 generations.

For the purpose of building stressing tests, we have significantly increased the usual load and generation specified for the IEEE 24-bus network initial configuration (Figure 2).

In total, 8 candidate nodes for installation of capacitor banks were defined: 4 nodes on 138 kV and 4 nodes on 230 kV voltage level. The initial configuration did not include any capacitor banks. All the nodes with synchronous generators were defined as PV nodes with voltage regulation ranging from 0.97 to 1.1 p.u. Investment and installation costs were set at 27000 cost units for one fixed unit of 3 MVAr and 45000 per one 3 MVAr unit of switched banks. Finally, voltage deviations higher than 9% were penalized.

We run 2 tests with IEEE 24-bus system:

- Test 1 - two scenarios with equal load levels and equal duration (4380 hours)
- Test 2 – two scenarios with a structural base case and a contingency case (line 16 out) and both with the same load.

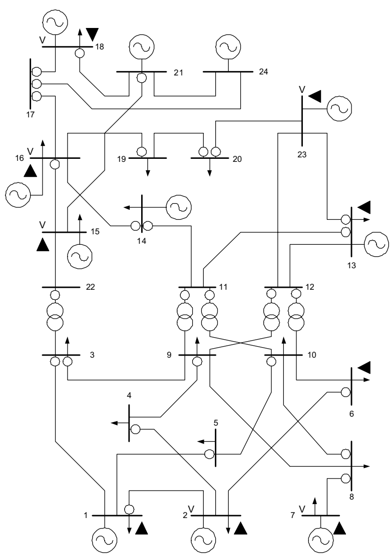

**Figure 2** IEEE 24 bus test network.

**Table 1** IEEE-24 with equal load scenarios – results for test 1.

| | initial | final |
|---|---|---|
| $P_g$ [MW] | 5244.5 | 5231.4 |
| $P_{\text{load}}$ [MW] | 5130.6 | 5130.6 |
| $P_{\text{losses}}$ [MW] | 113.9 | 100.8 |
| $\max\lvert V_{i,j} - V_{iN}\rvert$ | +6.2% | +9.3% |

One should expect from Test 1 a generalized adoption of fixed capacity devices and it did happen. Table 1 presents some results from this exercise. The total installed power of reactive power injection was of $Q_i = 441$ MVAr, all in fixed capacitor banks. We may observe that, given the penalty scaling factors used, the algorithm gave preference to reducing losses at the expense of higher deviations from the nominal voltage values (in general, overvoltages).

Test 2 was used for testing the algorithm's capability of handling various network topologies simultaneously, within a single optimization process. The planner is therefore able to check how the installation of new reactive power injection devices is influenced by security constraints in the form of lists of contingencies (leading to distinct network topologies) and their probabilities of occurrence. With all other settings left unchanged, one transmission line was removed from network topology for the second scenario.

In this case, the algorithm chose switched capacitor banks in order to avoid overvoltages for both base and contingency planning periods while covering the energy losses. The total

**Table 2** IEEE-24 with a contingency scenario – results for test 2

| | initial | | final | |
|---|---|---|---|---|
| scenario | base case | contingency | base case | contingency |
| $P_{\text{losses}}$ MW | 113.9 | 137.5 | 100.8 | 120.6 |
| $\max\lvert V_{i,j} - V_{iN}\rvert$ | −6.2% | −5.99% | 9.3% | 9.8% |

amount of installed banks is consistent with the previous test – 441 MVAr in fixed capacitor banks and additional 62 MVAr in switched ones. Of course, the values are conditioned by the settings of scaling factors and costs. We selected these results to illustrate the trade-off between voltage control and loss reduction.

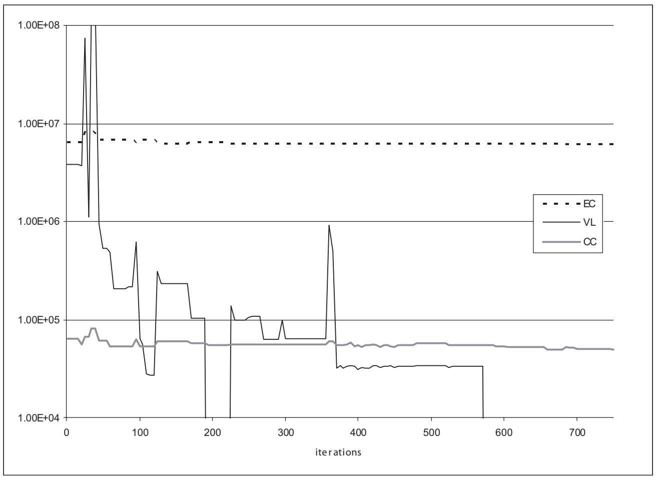


**Figure 3** IEEE 30 bus test network.

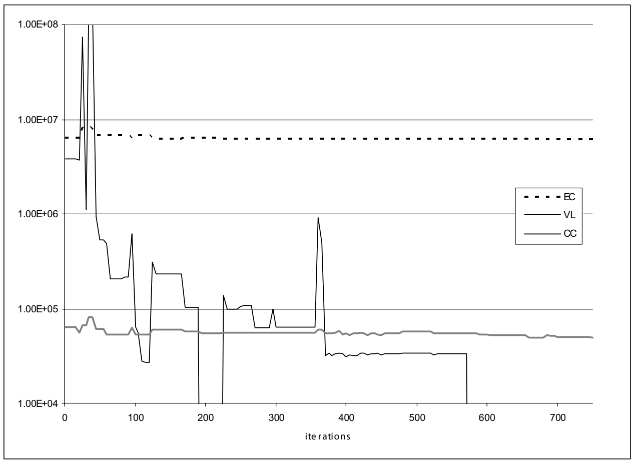


**Figure 4** Algorithm convergence in the IEEE 30 bus test case: voltage deviations higher than 3% are penalized along iterations (axis x).

We run a Test 3 with the IEEE 30-bus network (see Figure 3) where all the PQ nodes were selected as candidate nodes. This test was composed of three runs specifying different levels for the voltage penalty factors, in order to show again the trade-off between voltages and losses.

To illustrate the evolution of the search, we present Figure 4, where the three components of the objective function are present: investment, cost of losses and voltage penalty. The significant curve is the latter, because the scaling factors hide the evolution of the former cost components in the figure. The curves are established for the best current solution at every iteration.

The convergence graph of the voltage penalty component reveals that the algorithm is searching in regions resulting in high voltage costs, and moves with iterations into regions where the solutions are less and less penalized, although in an oscillating manner. Eventually, this cost component becomes erased and the voltages at all nodes fall in the admissible band – see Table 3.

**Table 3** IEEE-30 network - results.

| penalty | initial | $>3\%$ | $>5\%$ | $>8\%$ |
|---|---|---|---|---|
| $P_g$ [MW] | 161.72 | 160.85 | 160.76 | 160.63 |
| $P_{\text{losses}}$ [MW] | 3.25 | 2.35 | 2.26 | 2.13 |
| max $\lvert V_{i,j} - V_{iN}\rvert$,% | $-4.35$ | $+3.00$ | $+4.91$ | $+7.7\%$ |
| $\sum Q_i$ [MVAr] | - | 58.1 | 56.2 | 50.8 |

These results allow one to identify a Pareto surface where decisions depend on accepted trade-offs. For instance Figure 5 shows the conflict detected between voltage regulation and power losses.

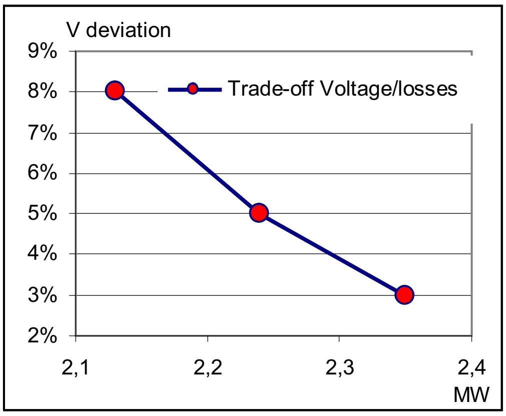


**Figure 5** Admissible voltage band vs. power losses in the IEEE 30-bus system: a Pareto optimum front – both criteria should be minimized but there is no optimal solution.

## 6. CONCLUSIONS

In this paper, we have presented an application to the reactive power planning problem of an EPSO algorithm implementation, coupled with a high-performance network calculation library. The distinguishing features of the model are:

- The ability to take into account multiple scenarios including under contingency, so that the planned network is able to respond to emergency situations.
- The ability to take into account different load levels for each structural scenario, in a global optimization procedure.
- The use of an Evolutionary Particle Swarm algorithm to perform computations.
- The observation of algorithm flexibility and robustness in convergence.
- The possibility of using the model in EMS or DMS environment also for operation – by just setting the investment costs to zero.

An interesting foreseen development is coupling the algorithm with Monte Carlo simulations responsible for generating network states to validate the influence of reactive power injection devices on varying network conditions, especially in networks with distributed generation. This is made possible due to the acceptable running times of the EPSO algorithm. This is not straightforward but the potential to reuse information from particles allows one to consider seriously this development.

## REFERENCES

1. J.G. Vlachogiannis and K.Y. Lee, “Reactive power control based on particle swarm multi-objective optimization”, *Intelligent Systems Application to Power Systems, 2005. Proceedings of the 13th International Conference on*, Vol., Iss., 6–10 Nov. 2005
2. J.T. Ma and L.L. Lai, “Evolutionary programming approach to reactive power planning”, *IEE Proceedings on Generation, Transmission and Distribution*, Vol.143, Iss.4, Jul 1996, Pages:365–370
3. K.Y. Lee and F.F. Yang, “Optimal reactive power planning using evolutionary algorithms: a comparative study for evolutionary programming, evolutionary strategy, genetic algorithm, and linear programming”, *Power Systems, IEEE Transactions on*, vol.13, Iss.1, Feb 1998, Pages:101–108
4. V. Miranda and Naing W. Oo, “New experiments with EPSO – Evolutionary Particle Swarm Optimization,” Proceedings *IEEE Swarm Intelligence Symposium 2006*, Indianapolis, Indiana, USA, May 2006.
5. W. Lou Chin, L.M. Proença, V. Miranda, “Demonstrating an Efficient Capacitor Location and Sizing Method for Distribution Systems - Application to the Macau Network”, *Actas de ELAB’96 - 3° Encontro Luso-Afro-Brasileiro de Planeamento e Exploração de Redes de Energia*, vol.1, Oct. 1996.
6. V. Miranda and N. Fonseca, “EPSO - Evolutionary Particle Swarm Optimization, a New Algorithm with Applications in Power Systems”, *Proceedings of IEEE T&D AsiaPacific 2002 - IEEE/PES Transmission and Distribution Conference and Exhibition 2002: Asia Pacific*, vol.2, October 2002, pp.745–750.
7. V. Miranda and N. Fonseca, “Reactive Power Dispatch with EPSO - Evolutionary Particle Swarm Optimization”, *Proceedings of PMAPS’2002 - International Conference on Probabilistic Methods Applied to Power Systems*, September 2002.
8. V. Miranda and N. Fonseca, “EPSO – Best-Of-Two-Worlds Meta-Heuristic Applied To Power System Problems”, *Proceedings of WCCI’2002 - CEC - World Congress on Computational Intelligence - Conference on Evolutionary Computing*, Proceedings of, May 2002.
9. V. Gopalakrishnan, P. Thirunavukkarasu, R. Prasanna, “Reactive power planning using hybrid evolutionary programming method”, *Power Systems Conference and Exposition, 2004.* IEEE PES, Oct. 2004., pp. 1319–1323
10. J. Kennedy and R. Eberhart, “Particle swarm optimization”, *IEEE International Conference on Neural Networks*, Proceedings on, 1995, vol.4, pp.1942–1948.

11. V. Miranda, N.W. Oo, J.N. Fidalgo, "Experimenting in the Optimal Capacitor Placement and Control Problem with Hybrid Mathematical-Genetic Algorithms", *Proceedings of ISAP'2001 - Intelligent Systems Applications to Power Systems Conference*, vol.1, June 2001.

## BIOGRAPHIES

**Hrvoje Keko** received his graduation degree from Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia in 2003. He is currently a PhD student at the same University, where he also worked as a researcher and teaching assistant. From May 2006 he is a Junior Researcher in Power Systems Unit of INESC Porto, Portugal. His primary research interests are evolutionary algorithms and computational intelligence applied to power systems.

**Álvaro Jaramillo Duque** received a degree in Electrical Engineering from University of Antioquia, Colombia in 2005. He is currently Junior Researcher in INESC-Porto, in Power Systems Unit. His research interests include evolutionary computation and swarm theory applied to solving power system optimization problems.

**Vladimiro Miranda** received his graduation, Ph.D. and Agregado degrees from the Faculty of Engineering of the University of Porto, Portugal (FEUP) in 1977, 1982 and 1991, all in Electrical Engineering. In 1981 he joined FEUP and currently holds the position of Professor Catedrático (Full Professor). He is also currently Director of INESC Porto – the Institute of Engineering in Systems and Computers of Porto, Portugal, a private non-for-profit research institute. He has authored many papers and been responsible for many projects in areas related with the application of Computational Intelligence to Power Systems, namely including Fuzzy Logic models, Artificial Neural Networks and Evolutionary Algorithms.