Differential Evolutionary Particle Swarm Optimization (DEEPSO): a successful hybrid

Vladimiro Miranda, *Fellow, IEEE* and Rui Alves, *Student Member IEEE*

***Abstract* — This paper explores, with numerical case studies, the performance of an optimization algorithm that is a variant of EPSO, the Evolutionary Particle Swarm Optimization method. EPSO is already a hybrid approach that may be seen as a PSO with self-adaptive weights or an Evolutionary Programming approach with a self-adaptive recombination operator. The new hybrid DEEPSO retains the self-adaptive properties of EPSO but borrows the concept of rough gradient from Differential Evolution algorithms. The performance of DEEPSO is compared to a well-performing EPSO algorithm in the optimization of problems of the fixed cost type, showing consistently better results in the cases presented.**

***Index Terms* — Evolutionary Particle Swarm Optimization, Differential Evolution, fuzzy clustering, unit commitment, PAR location.**

I. Introduction

THIS paper presents a new approach to build a hybrid between Evolutionary Programming, Particle Swarm Optimization and Differential Evolution. The reason behind the search for hybrid algorithms is that each "pure" method exhibits some characteristics that push the search for the optimum in a globally right direction. However, each method also displays its own difficulties. The hope is that, by suitably blending methods, a more robust and general method may be derived.

The work reported in this paper departed from an algorithm denoted EPSO, for Evolutionary Particle Swarm Optimization. The basic version of this algorithm was presented in 2002 [1] and introduced as a way "to join together the exploratory power of PSO (Particle Swarm Optimization) with the self-adaptation power of Evolutionary Algorithms (EA) and have as a result the *best of two worlds*". The results obtained in competition with classical versions of PSO were indeed promising and this was demonstrated by several authors and in several publications. Early reports as well as more recent works [2]-[16] confirmed the quality and reliability of the algorithm as well as its good performance in a diversity of domains. The EPSO algorithm then received further improvement and the latest version is available from [17], where examples and a source code are made public.

In a parallel path, the Differential Evolution concept (DE), early proposed in [18][19], has motivated many proposals for improvement and variants. A comprehensive survey may be found in [20]. In this survey, the allegations that DE is a fast and general optimization method for any kind of objective function are substantiated, although the authors caution against a hasty conclusion, reminding the reader of the No Free Lunch theorem. In particular, the attempts to generate a synergy of DE with PSO are well documented in this survey. Many of the proposed hybrid models adopt a form of alternate use of DE and PSO iterations or DE and PSO operators [21][22] or even some mixture of operators [23][24]. These references are just examples and not to be taken as exhaustive.

Adaptive versions of DE have been attempted also with many variations [25][26]. The pursuit for successful self-adaptive schemes is justified by the desire to achieve some algorithmic form close to a non-parametric or parameter-free definition. This search is also the motivation behind the inception of the EPSO algorithm.

The advanced version of the EPSO algorithm included the positive effect of a probability of communication among particles, implementing the scheme of the "stochastic star". Its success reinforced the idea that a degree of controlled random variation is beneficial to the search for the optimum. Therefore, the idea that some noise could be added to the EPSO search by embedding a DE operator in the global mechanism of the generation of new particles is worth exploring.

This paper presents a new hybrid DE-EA-PSO, denoted DEEPSO. As in many other cases, there is no deductive demonstration of superiority over other options but illustration by example. A didactic and a complex study case will be presented, in the domain of power systems, to put in evidence the strong points of the new approach. There is no comparison with DE, as the purpose is to show that the modification does improve EPSO. The benchmarking of EPSO with DE has been made by some authors, such as in [27].

II. Basic Models

A. PSO as a recombination process

The PSO – Particle Swarm Optimization [28] does not rely on a selection operator as its driving force: it depends on a *movement rule* that generates new individuals in space from a set of known alternatives, called a swarm (the same as population). Several variants have been proposed but the basic movement rule, producing a new individual **X** for iteration

This work is partially funded by the ERDF from the EU through the Programme COMPETE and by the Portuguese Government through FCT - Foundation for Science and Technology, project ref. LASCA PTDC/EEA-EEL/104278/2008.

V. Miranda (vmiranda@inescporto.pt) and R. Alves (ee08272@fe.up.pt) are with INESC TEC (INESC Technology and Science, coordinated by INESC Porto) and also with FEUP, Faculty of Engineering of the University of Porto, Portugal.

(k+1) is based on

$$\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} + \mathbf{V}^{(k)} \quad (1)$$

where **V** is called the particle velocity and is defined by

$$\mathbf{V}^{(k+1)} = \mathbf{A}\mathbf{V}^{(k)} + \mathbf{B}(\mathbf{b}_i - \mathbf{X}^{(k)}) + \mathbf{C}(\mathbf{b}_G - \mathbf{X}^{(k)}) \quad (2)$$

where $\mathbf{b}_G$ is best point so far found by the swarm and $\mathbf{b}_i$ is the best past ancestor in the direct life line of the particle, with $\{\mathbf{b}_i, i = 1,..., \text{no. particles}\} = \mathbf{P}_b$ forming the set of the historical past best ancestors of each particle. Of course, $\mathbf{b}_G \in \mathbf{P}_b$.

The parameters **A, B, C** are diagonal matrices with weights defined in the beginning of the process. In a classical formulation, the parameter **A** is affected by a decreasing value with time (iterations), while the initial parameters **B** and **C** are successfully multiplied by random numbers [28] sampled from a uniform distribution in [0,1].

From eq. (1) and (2) we conclude that a new particle $\mathbf{X}^{(k+1)}$ is formed as a combination of four other points:

- Its direct ancestor $\mathbf{X}^{(k)}$
- The ancestor $\mathbf{X}^{(k-1)}$ of its ancestor $\mathbf{X}^{(k)}$
- A (possibly) distant past best ancestor $\mathbf{b}_i$
- The current global best of the swarm $\mathbf{b}_G$.

We can give a different aspect to the movement rule in (2):

$$\mathbf{X}^{(k+1)} = (1 + \mathbf{A} - \mathbf{B} - \mathbf{C})\mathbf{X}^{(k)} - \mathbf{A}\mathbf{X}^{(k-1)} + \mathbf{B}\mathbf{b}_i + \mathbf{C}\mathbf{b}_G \quad (3)$$

In this expression, the sum of the parameters multiplying the four contributors to generate the offspring is equal to 1. It is therefore tempting to identify this expression with an intermediary recombination in EA with 4 parents and a special rule to determine who the parents are (they are not randomly selected). This means that we are considering an *enlarged population* including not only the active particles but also the immediate ancestors and the set of the past best ancestors.

B. EPSO as an evolutionary adaptive recombination

The idea behind the EPSO algorithm was to provide adaptive capability to this recombination operator. To achieve this, the parameters in (2) are subject to mutation and selection in order to try to achieve a higher progress rate.

Given a population with a set of particles, the general scheme of EPSO became:

```
REPLICATION - each particle is replicated (cloned) r
  times [usually r = 1]
MUTATION - all r particles have their A,B,C parameters
  mutated
REPRODUCTION - each of the r+1 particles (original and
  clones) generate an offspring through recombination,
  according to the particle movement rule (2) or (3)
EVALUATION - the offspring have their fitness
  evaluated
SELECTION - by stochastic tournament or other
  selection procedure (among siblings), the best child
  from each ancestor survives to form a new generation
  - every individual in the previous generation has one
  descendant.
```

The mutation of any parameter A,B,C (represented by w in the following) is ruled by multiplicative Lognormal random numbers such as in $w_i^* = w_i \left[\log N(0,1)\right]^\tau$ or by additive Gaussian distributed random numbers such as in $w_i^* = w_i + \sigma N(0,1)$. The learning parameter (τ or σ) must be fixed externally. The recombination operator is defined by the set (A,B,C). The scheme results in an adaptive recombination operator.

The EPSO algorithm was further improved in efficiency by the introduction of two additions. In early versions, it was shown that noise affecting the exact location of $\mathbf{b}_G$ was beneficial, so a forth parameter or weight in the form of a diagonal matrix $\mathbf{w}_G$ was introduced, such that

$$\mathbf{b}_G^* = \mathbf{b}_G (1 + \mathbf{w}_G N(0,1)) \quad (4)$$

This weight is also subject to mutations of the kind referred to above, so it also enters in the self-adaptive model.

Finally, in the most recent and efficient version, the Communication Factor **P** was introduced, creating a Stochastic Star communication topology among the swarm.

The recombination (or movement) rule for EPSO becomes

$$\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} + \mathbf{V}^{(k)} \quad (5)$$

$$\mathbf{V}^{(k+1)} = \mathbf{A}\mathbf{V}^{(k)} + \mathbf{B}(\mathbf{b}_i - \mathbf{X}^{(k)}) + \mathbf{P}[\mathbf{C}(\mathbf{b}_G^* - \mathbf{X}^{(k)})] \quad (6)$$

P is a diagonal matrix affecting all dimensions of an individual, containing binary variables of value 1 with probability p and value 0 with probability (1-p); the p value (communication probability) controls the passage of information within the swarm and is 1 in classical formulations (the *star*).

This stochastic scheme conceptually oscillates between the star arrangement and a selfish version called *cognitive model* in [28], where no communication exists. In fact, the stochastic star causes that some components of the global best become "known" by a particle while other components are ignored, so that the production of a new particle is affected in different ways in its distinct dimensions. This favors the uncoupling of the evolution for all the dimensions.

Experiments in a diversity of problems made it quite clear that one could achieve a fine tuning of the convergence of EPSO by adequately setting a value for p, the communication probability [13].

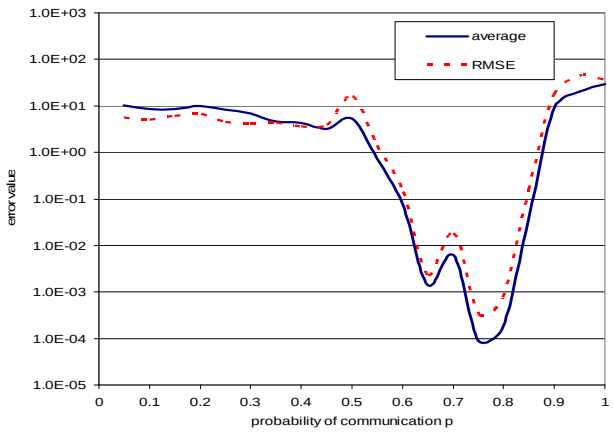

Fig. 1. Rosenbrock function in 30 dimensions after 100000 fitness function evaluations – average and RMSE of achieved error values for 20 runs with EPSO, as a function of the communication probability p – from [13].

In many problems, p = 0.75 seems a very good option but in some problems a much lower value favors the convergence. Figure 1 from [13] illustrates the sharp tuning in error, achieved for p = 0.75 in the Rosenbrock function problem in 30 dimensions (note the logarithmic scale).

C. Differential Evolution

The original idea behind DE, given a population (swarm) of individuals (particles, vectors), is to generate a new solution from an existing individual by adding some fraction of the difference between two other points $\mathbf{X}_{r1}$ and $\mathbf{X}_{r2}$ sampled from the population or swarm. Then, having a new population generated, some further recombination ensures more diversity and a selection procedure produces a new generation. This selection is elitists and one-on-one based, meaning that each parent competes for survival directly with its single offspring and the best is retained.

There are many variations to this scheme. One interesting case is the one that was denoted DE2 in [18], DE/rand-to-best/1 in [19] and DE/target-to-best/1 in [20], where the generation of a new individual may be written as

$$\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} + \mathbf{V}^{(k)} \quad (7)$$

$$\mathbf{V}^{(k+1)} = \mathbf{B}(\mathbf{X}_{r1}^{(k)} - \mathbf{X}_{r2}^{(k)}) + \mathbf{C}(\mathbf{b}_G^{*} - \mathbf{X}^{(k)}) \quad (8)$$

A notation slightly distinct from the usually seen in the DE literature is adopted here to enhance the similarities with (1) and (2), i.e. between DE and PSO in the process of generating new individuals. The canonic version of DE makes $\mathbf{C} = \mathbf{0}$; the canonic DE/target-to-best/1 version makes $\mathbf{B} = \mathbf{C}$.

Neglecting recombination, DE then proceeds with a *parent selection* (choosing the next generation from both the parent and offspring populations) of a special type – each parent competes only with its offspring – while PSO adopts in a way a trivial *survivor selection* (the next generation is chosen among the offspring only).

EPSO has also a special survivor selection procedure where competition is established only among the direct descendants of each particle.

III. The DEEPSO alternative

The DE scheme, in a way, makes a sample of a local macro-gradient of the objective function by picking up two random individuals from the population. The same kind of sampling is produced by the PSO movement equation, but picking up the current position and the particle past best. So, it is natural to ask if the DE scheme would not work also when inserted in the PSO equation.

Also, the DE scheme is usually based on fixed B parameter values in [0.1, 1]. One must refer that in [8] the authors claimed to have a self-adapting process for this parameter; however, its value would only change with a certain probability (0.1), remaining fixed most of the time, so it must be seen as a quite modest effort into self-adaptation. But the EPSO scheme is truly self-adaptive, so it is natural to wonder if the EPSO scheme would not work also when acting over the DE parameter.

This reasoning led to the proposal of the model that will be denoted DEEPSO to clearly express its hybrid character. The DEEPSO algorithm is equal to the EPSO sequence; however, to grasp the flavor of DE, the following general equation should now express the movement rule:

$$\mathbf{V}^{(k+1)} = \mathbf{A}\mathbf{V}^{(k)} + \mathbf{B}(\mathbf{X}_{r1}^{(k)} - \mathbf{X}_{r2}^{(k)}) + \mathbf{P}[\mathbf{C}(\mathbf{b}_G^{*} - \mathbf{X}^{(k)})] \quad (9)$$

where $\mathbf{b}_G^{*}$ is given by (4). In (9), $\mathbf{X}_{r1}^{(k)}$ and $\mathbf{X}_{r2}^{(k)}$ should be any pair of distinct particles, in principle belonging to the set $\mathbf{P}_C$ of the particles in the current generation. But extensive testing led to an improved proposal, which regains back the spirit of PSO and also retains the spirit of DE. First of all, PSO relies on macro-gradients being sensed by a particle. So, these particles should be ordered such that, for minimization,

$$f(\mathbf{X}_{r1}^{(k)}) < f(\mathbf{X}_{r2}^{(k)}) \quad (10)$$

Then, one may enlarge the definition of which set must these particles be sampled from: this may be the set $\mathbf{P}_C$ of particles from the current generation or the set $\mathbf{P}_b$ of historical past best particles. Finally, the DEEPSO model defines that $\mathbf{X}_{r2}^{(k)}$ equal to $\mathbf{X}^{(k)}$ so only $\mathbf{X}_{r1}^{(k)}$ is sampled.

To complete the model, the sampling of $\mathbf{X}_{r1}^{(k)} (= \mathbf{b}_{r1}^{(k)})$ among $\mathbf{P}_\mathbf{b}$ may repeated for each component of $\mathbf{V}$ to be calculated. This means that one is, in fact, calculating $\mathbf{X}_{r1}^{(k)}$ from a uniform recombination of all the particles in $\mathbf{P}_b$. The equations regulating DEEPSO are, therefore,

$$\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} + \mathbf{V}^{(k)} \quad (11)$$

with $\mathbf{V}^{(k)}$ in 4 versions:

1. **DEEPSO Sg** (sampling in the same generation):
$$\mathbf{V}^{(k+1)} = \mathbf{A}\mathbf{V}^{(k)} + \mathbf{B}(\mathbf{X}_{r1}^{(k)} - \mathbf{X}^{(k)}) + \mathbf{P}[\mathbf{C}(\mathbf{b}_G^{*} - \mathbf{X}^{(k)})] \quad (12)$$
with $\{ \mathbf{X}_{r1}^{(k)}, \mathbf{X}^{(k)} \}$ ordered according to (10) and $\mathbf{X}_{r1}^{(k)}$ sampled once from the current generation.
2. **DEEPSO Sg-rnd**: the same but with $\mathbf{X}_{r1}^{(k)}$ re-sampled in the current generation for each component of $\mathbf{V}$.
3. **DEEPSO Pb** (sampling from the past bests):
$$\mathbf{V}^{(k+1)} = \mathbf{A}\mathbf{V}^{(k)} + \mathbf{B}(\mathbf{b}_{r1}^{(k)} - \mathbf{X}^{(k)}) + \mathbf{P}[\mathbf{C}(\mathbf{b}_G^{*} - \mathbf{X}^{(k)})] \quad (13)$$
with $\{ \mathbf{b}_{r1}^{(k)}, \mathbf{X}^{(k)} \}$ ordered according to (10) and $\mathbf{b}_{r1}^{(k)}$ sampled once from $\mathbf{P}_b$.
4. **DEEPSO Pb-rnd**: the same but with $\mathbf{b}_{r1}^{(k)}$ re-sampled among $\mathbf{P}_b$ for each component of $\mathbf{V}$.

In the following sections, some examples will be presented to ilustrate the virtues of the DEEPSO scheme.

IV. DEEPSO vs. EPSO

A. Fuzzy clustering

The first example to show that EPSO is indeed improved concerns an application to fuzzy clustering with the fuzzy c-means algorithm [31]. It is an example of a continuous function where EPSO is expected to behave well.

The fuzzy c-means algorithm minimizes the following

function:

$$J = \sum_{i=1}^{N} \sum_{j=1}^{Cl} u_{ij}^{m} \, \| \mathbf{X}_i - \mathbf{C}_j \|^2 \qquad (14)$$

where $\mathbf{X}_i$ is a member of the set of d-dimensional data, m is any real number greater than 1, u_{ij} is the degree of membership of $\mathbf{X}_i$ in the cluster j, $\mathbf{C}_j$ is the d-dimension center of the cluster, and ||*|| is any norm expressing the similarity between data and the centroids.

Comparative tests have been done in a set of problems. A 2-dimension problem is depicted in Fig. 2 (coordinates in integers may be inspected directly). Fig. 3 makes it evident that the DEEPSO concept seems to bring value to the swarm optimization. It displays the value of the objective function (14), on an average of 20 runs, for three experiments, using a swarm of 8 particles, and with the best tuned parameters for each model:

- EPSO with p = 0.1 (best value)
- DEEPSO Sg-rnd as in (12), i.e. sampling $\mathbf{X}_{r1}^{(k)}$ in the current generation $\mathbf{P}_C$
- DEEPSO Pb-rnd as in (13), sampling in the set of past bests of the particles $\mathbf{P}_b$.

This example is interesting because it convincingly argues for the advantage of sampling within the population instead of using the canonic PSO choice.

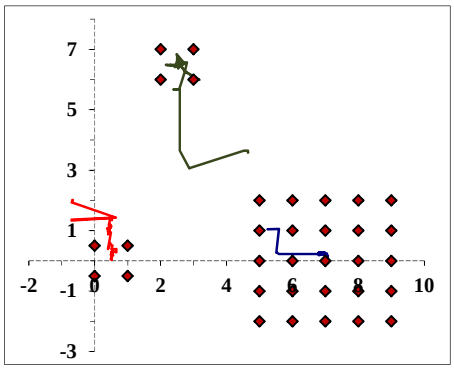

Fig. 2. Three clusters and the trajectory of the centroids during one run of the optimization process with EPSO. Point coordinates are integer numbers.

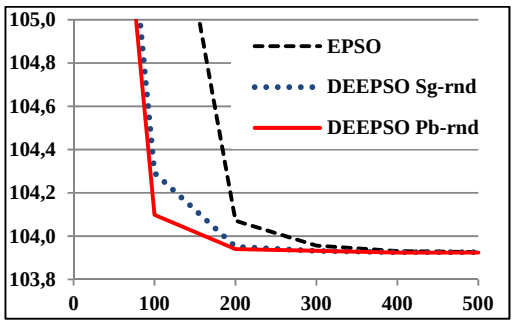

Fig. 3. Value J of the fuzzy c-means function plotted against the number of generations for EPSO and 2 versions of DEEPSO.

B. Unit commitment

The problem of unit commitment in power systems is mathematically defined as a fixed cost problem or a mixed-integer non-linear programming problem: given a set of generators and their generation cost curves, define which generators should be shut down and which should be in service and at which loading level, in order to minimize the overall cost (start up costs plus operation costs).

Because of technical limits, the domain of a generator is not connected – there is a point (0,0) corresponding to generator shut down and then there is a gap until a point (P_{min}, $c(P_{min})$) corresponding to the technical minimum of the machine. This general shape of the cost functions implies that the problem has a non-convex nature – therefore, many local optima may appear.

An illustrative problem of this type was included in [31], where a preliminary suggestion for a DEPSO algorithm was formulated. The data are:

- the number of generators – ngen = 5
- the parameters of the cost function of each generator – this function is assumed to be a cubic polynomial, with 4 parameters a_i, i=1 to 4: $C = a_0 + a_1 P + a_2 P^2 + a_3 P^3$ where C is the generation cost in $/hour and P is the generator output in MW.
- the technical minimum and maximum of each generator P_{min} and P_{max}.
- the load, located at a single bus (transmission system neglected): L = 15 MW (see [32]).

The objective is to minimize the sum of the costs for the five generators, noting that the domain of each variable is not continuous.

The cost curves and technical limits are given as:

Generator	a_0	a_1	a_2	a_3	Pmin	Pmax
g1	1	0,5	0,1	0,03	0 or 1	10
g2	2	0,4	0,2	0	0 or 2	10
g3	4	0,3	0,3	0	0 or 7	10
g4	6	1,5	0,15	0	0 or 2	10
g5	0	4	0	0	0 or 1	10

The optimal solution is

g1	g2	g3	g4	g5	Cost
3.414	4.586	7	0	0	33.9068

Adopting a swarm of 16 particles, in 100 trials with random initialization and 1000 iterations, the number of times the optimal solution was discovered is the following:

EPSO	DEEPSO Sg-rnd	DEEPSO Pb-rnd
46%	71%	81%

The same conditions were kept for all experiments – namely initializing the weights with A = 0.1, B = C = 0.5, w_G = 0.1, σ = 0.1 and p = 0.3. For less iterations or smaller populations, the same difference in performance was observed.

This result is yet another argument in favor of the superior performance of the DEEPSO versions, with advantage to DEEPSO Pbest.

C. PAR/PST location and sizing in power grids

A Phase Angle Regulating (PAR) transformer of a Phase Shifting Transformer (PST) is a special arrangement of power

transformers used to control the flow of active power in meshed three phase power system transmission grids. Because the power through a line is roughly proportional to the sine of the angle between voltages at the sending and receiving ends of a line, the control of such angle may re-route power through alternative paths in the system, preventing overloads and giving better use to the transmission capacity available.

This comes at a high capital cost per device but it may be compensated by avoiding costly line reinforcements or allowing a more flexible operation with higher security and reduced operation costs.

Given a set of load scenarios as well as wind power scenarios, a system operator may be faced with the need to curtail wind generation (at a cost) and replace it by conventional generation (at a cost) or, in more severe cases, to curtail load (at the highest cost of power not supplied). Instead, the suitable location of PAR transformers and their optimal dimensioning (in terms of the maximum angle they may inject in a line, admitting that they are of the variable phase shift type) may serve to reduce or eliminate such curtailment needs.

The capital cost of each PAR may be modeled as being composed of a fixed cost plus a non-linear variable cost which is a function of the maximum angle that the PAR may inject. Some candidate locations in the power network must be specified and, in each location, a tentative device allocation may be defined. This forms a possible solution to the problem, which must be evaluated by solving the power flow equations in all scenarios considered and deciding if and how much power must be curtailed and of what nature: wind generation or load.

Furthermore, each scenario may have a probability of occurrence associated. The problem becomes of the type of stochastic optimization. In the following paragraphs we will describe a model for this problem and its application to a realistic problem built around the IEEE RTS 24 bus system.

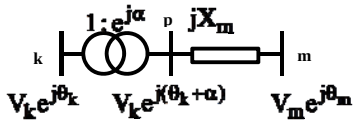

Fig. 4. Equivalent circuit for a PAR

The equivalent circuit for a PAR is in Fig. 4. Its effect is to force to a power flow from node k to node m:

$$P_{pm} = \frac{\theta_p - \theta_m}{X_{km}} = \frac{(\theta_k - \alpha) - \theta_m}{X_{km}} = \frac{\theta_k - \theta_m}{X_{km}} - \frac{\alpha}{X_{km}} \quad (15)$$

So this is equivalent to having a series reactance X_{km} plus a power injection which will be a load in node k and a generation in node m. This allows a network power flow model to be written, as a function of α.

Given a specific set of N candidate locations to install a PAR and considering a generation system composed only of conventional units, the allocation and sizing of PAR is defined by the following

$$\min J_k = \sum_{i=1}^{N} u_i (A + B(\alpha_i^{Max})^2) + \text{Penalties} \quad (16)$$

where u_i is a binary variable representing the installation of a PAR on location i, A and B are cost constants and α_i^{Max} is the maximum angle introduced by the device at location i. The constraints are the usual power flow equations of the DC model, incorporating eq. (15), plus limits on generation and on line flows and limits on the PAR angles:

$$\alpha_i^{min} \le \alpha_i \le \alpha_i^{max} \quad (17)$$

These constraints may be transformed into penalties, in eq. (16), when adopting a meta-heuristic as the solver. Finally, the penalty term will include if necessary the cost for load curtailment, which is usually modeled as fictitious generators by the loads with generation cost equal to the usually high cost of power not supplied.

The objective function is further modified when in the presence of wind power, because there is also the possibility to spill wind (disconnect wind generation) if necessary, to assure the network security described by the constraints. This may be represented as a negative load which is supplied at the cost associated with wind curtailment (compensation to wind power producers).

The wind power resource may be represented by a set of S scenarios stratified according to a Weibull distribution, associating each scenario k with a probability value p_k. This allows a stochastic optimization model to be built where a solution is evaluated in all S scenarios:

$$\min J = \sum_{k=1}^{S} p_k J_k \quad (18)$$

A chromosome for an EPSO algorithm will have a length of N and each component i is a proposal for α_i^{max} at location i.

This model was applied to the IEEE RTS 24-bus test system [33], with 8 possible locations for PAR. This is a realistic power system; data have been adjusted to fit in the problem of optimal PAR location. A comparison among EPSO and DEEPSO variants is presented in Fig. 5 for 100 runs of each algorithm, with a swarm of 30 particles.

The figure counts how many times each algorithm reached the optimum, in 100 runs, with varying number of generations. The DEEPSO Pb-rnd algorithm displays remarkable superiority: at 60 generations it had already reached a 96% efficiency in finding the optimum. In second place, we meet the DEEPSO Pb and the original EPSO algorithms with similar development. The algorithm using the DE trick with particles in the same generation lags definitely behind.

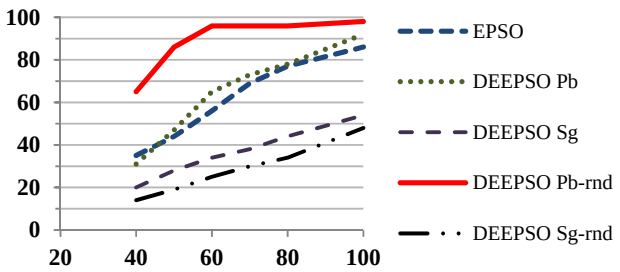

Fig. 5. Number of hits on the optimum (y-axis) vs. number of generations (x-axis) in 100 runs for EPSO and 4 DEEPSO variants.

V. Conclusion

The examples shown, selected among many other tested by the authors, illustrate that a successful hybrid between the evolutionary particle swarm algorithm and the differential evolution algorithm concept, deemed DEEPSO, leads to better performance in the optimization of problems with a fixed-cost mixed-integer objective function. These problems display generally a deceptive landscape which makes it difficult to discover the optimal solution in many cases.

The advantage of having an adaptive recombination scheme associated to the PSO logic had already been demonstrated with EPSO. With the DEEPSO Pb-rnd formulation, one now suggests that the recombination scheme should be enlarged to the set of particle past bests. The soundness of this idea should be further confirmed in a diversity of sets of tests and cases.

References

[1] V. Miranda and N. Fonseca, "EPSO – Best-of-Two-Worlds Meta-Heuristic Applied to Power System Problems", Proceedings of WCCI/CEC – World Conf. on Computational Intelligence, Conf.on Evolutionary Computation, Honolulu (Hawaii), USA, June 2002

[2] V. Miranda and N. Fonseca, "Reactive Power Dispatch with EPSO - Evolutionary Particle Swarm Optimization", Proc. of PMAPS – Int. Conf. Probabilistic M. App. to Power Systems, Naples, Italy, Sep 2002

[3] I. M. S. Dias, N. Fonseca, H. M. Salgado and V. Miranda, "Optical Fibre Bragg Gratings Synthesis by Evolutionary Particle Swarm Optimization", Proceedings of the Portuguese Physics Conference FISICA 2002, Évora, Portugal, Sep 2002.

[4] V. Miranda and N. Fonseca, "EPSO - Evolutionary Particle Swarm Optimization, a New Algorithm with Applications in Power Systems ", Proceedings of IEEE/PES Transmission and Distribution Conference and Exhibition 2002: Asia Pacific, vol.2, pp.745-750, October, 2002

[5] H. Mori and Y. Komatsu, "A Hybrid Method of Optimal Data Mining and Artificial Neural Network for Voltage Stability Assessment", Proc. IEEE St. Petersburg PowerTech, St. Petersburg, Russia, June 2005

[6] V. Miranda, "Evolutionary Algorithms with Particle Swarm Movements", Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems - ISAP 2005, pp. 6-21, Arlington (VA), USA, 6-10 Nov. 2005

[7] N. W. Oo and V. Miranda, "Multi-energy Retail Market Simulation with Intelligent Agents", Proceedings of IEEE St. Petersburg Power Tech, St. Petersburg, Russia, Jun 2005

[8] N. W. Oo and V. Miranda, "Evolving Agents in a Market Simulation Platform – A Test for Distinct Meta-Heuristics", Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems ISAP 2005, pag. 482–487, Arlington (VA), USA, Nov. 2005

[9] Mehdi Eghbal, E.E.El-Araby, Naoto Yorino and Yoshifumi Zoka, "Application of Metaheuristic Methods to Reactive Power Planning: A Comparative Study for GA, PSO and EPSO", Proceedings - IEEE International Conference on Systems, Man and Cybernetics 2007 , pp. 3755-3760, Montreal (Quebec), Canada, Oct 7-10,2007

[10] P. Hilber, V. Miranda, M. Matos and L. Bertling, "Multiobjective Optimization Applied to Maintenance Policy for Electrical Networks", IEEE Trans. on Power Systems, Volume 22, No. 4, November 2007.

[11] "Advances in Evolutionary Computing for System Design", Series Studies in Computational Intelligence, Eds. L. C. Jain, V. Palade, D. Srinivasan, Volume 66, 2007, pp 139-167

[12] "Modern Heuristic Optimization Techniques – Theory and Applications to Power Systems", Eds. Kwang Y. Lee and Mohamed A. el-Sharkawi, IEEE Press series on Power Engineering, John Wiley and Sons Inc., Hoboken, New Jersey, 2008

[13] V. Miranda, H. Keko, Á. J. Duque, "Stochastic star communication topology in Evolutionary Particle Swarms (EPSO)", Intern. Journal of Computational Intelligence Research, vol.4, no.2, pp. 105-116, 2008

[14] V. Miranda, L. Carvalho, M. Rosa, A. M. Leite da Silva, C. Singh, "Improving Power System Reliability Calculation Efficiency With EPSO Variants", IEEE Transactions on Power Systems , vol. 24, no. 4, pp. 1772-1779, 2009

[15] H. Leite, J. Barros, V. Miranda, "The evolutionary algorithm EPSO to coordinate directional overcurrent relay", 10th IET International Conference Developments in Power System Protection (DPSP 2010) Managing the Change, pp. 1–5 (2010)

[16] D. L. Souza, O. N. Teixeira, D. C. Monteiro, R. C. L. Oliveira, "A New Cooperative Evolutionary Multi-Swarm Optimizer Algorithm Based on CUDA Architecture Applied to Engineering Optimization", Proc. CIMA 2012, Montpellier, France, Aug. 2012, book series on Smart Innovation, Systems and Tech., no.23, pp 95-115, Springer Berlin Heidelberg, 2013

[17] EPSO Home Page – http://epso.inescporto.pt

[18] R. Storn, and K. Price, "Differential Evolution - a Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces", Technical Report TR-95-012, ICSI, March 1995

[19] R. Storn, and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces", Journal of Global Optimization no. 11, pp. 341–359, 1997

[20] S. Das and P. N. Suganthan, "Differential Evolution: A Survey of the State-of-the-Art", IEEE Transactions on Evolutionary Computation, vol. 15, no. 1, pp. 4-31,February 2011

[21] T. Hendtlass, "A combined swarm differential evolution algorithm for optimization problems," Lecture Notes in Computer Science, vol.2070, pp.11-18, 2001

[22] W.-J. Zhang and X.-F. Xie, "DEPSO: Hybrid particle swarm with differential evolution operator," in Proc. IEEE Int. Conf. Syst. Man & Cybernetics, pp. 3816–3821, 2003

[23] B. Liu, L. Wang, Y. H. Jin, and D. X. Huang, "Designing neural networks using hybrid particle swarm optimization," Lecture Notes in Computer Science, vol.3469, pp.391-397, 2005

[24] P. W. Moore and G. K. Venayagamoorthy, "Evolving digital circuit using hybrid particle swarm optimization and differential evolution," Int. J. Neural Syst., vol. 16, no. 3, pp. 163–177, 2006

[25] X. Xu, Y. Li, S. Fang, Y. Wu, and F. Wang, "A novel differential evolution scheme combined with particle swarm intelligence," IEEE Congress in Evolutionary Computation. (CEC08), pp.1057-1062, 2008

[26] M. G. H. Omran, A. P. Engelbrecht, and A. Salman, "Bare bones differential evolution," Eur. J. Oper. Res., vol. 196, no. 1, pp. 128–139, Jul. 2009

[27] D. Midence and A. Vargas, "Estudio Comparativo de Algoritmos de Computación Evolutiva en la Optimización de la Confiabilidad en Redes de Distribución de Potencia" (in Spanish), XIII ERIAC – CIGRE, Puerto Iguazú, Argentina, May 2009

[28] J. Kennedy and R. C. Eberhart, "Particle Swarm Optimization", *Proceedings of the 1995 IEEE International Conference on Neural Networks*, pp. 1942-1948, Perth, Australia, 1995

[29] J. Kennedy, "The Particle Swarm: Social Adaptation of Knowledge", *Proceedings of the 1997 International Conference on Evolutionary Computation*, pp. 303-308, IEEE Press, 1997

[30] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems," IEEE Trans. Evolutionary Computing, vol. 10, no. 6, pp. 646–657, Dec. 2006.

[31] J. C. Bezdek, "Pattern Recognition with Fuzzy Objective Function Algoritms", Plenum Press, New York, 1981

[32] V. Miranda and Naing Win Oo, "New experiments with EPSO – Evolutionary Particle Swarm Optimization", Proc. IEEE Swarm Intelligence Symposium 2006, Indianapolis (Indiana), USA, May 2006

[33] IEEE RTS Task Force of APM Subcommittee, "IEEE Reliability Test System", *IEEE Trans. PAS*, Vol 98, No 6, pp 2047-2054, Nov 1979.

Vladimiro Miranda (M'90, SM'04, F'06) received his B.Sc. and Ph.D. degrees from the Faculty of Engineering of the University of Porto, Portugal (FEUP) in 1977 and 1982 in Electrical Engineering. In 1981 he joined FEUP and currently holds the position of Full Professor. He is also a researcher at INESC since 1985 and is currently Director at INESC Porto, the leading institution of INESC TEC – INESC Technology and Science, an advanced research network in Portugal. He is also the President of INESC P&D Brasil, in São Paulo, Brazil. He has authored many papers and been responsible for many projects in areas related with the application of Computational Intelligence to Power Systems. He is the recipient of the 2013 IEEE PES Ramakumar Family Renewable Energy Excellence Award.

Rui Alves (StM'12) is a MSc. student in Power Systems at FEUP and junior researcher at INESC TEC.