Optimal Reactive Power Flow Using an Extended Evolution Strategy

JOSÉ R. GOMES OSVALDO R. SAAVEDRA
Departamento de Engenharia de Eletricidade
Universidade Federal do Maranhão
São Luís - MA - 65085-580
BRASIL
{jgomes, osvaldo}@dee.ufma.br - www.dee.ufma.br

Abstract: In this paper we present a new extended evolutionary algorithm for solving the optimal reactive power dispatch. The approach proposed has been exhaustively analyzed and compared with a state-of-art method. Good and reliable performance have been achieved and validation tests using the standard IEEE57 system are reported.

Key-words: Evolutionary computation, reactive dispatch, optimization, artificial intelligence, power systems

1 Introduction

The algorithms based on the principles of the natural evolution have been applied successfully to a set of problems of numerical optimization. With good degree of parallelism and stochastic characteristics, they are adequate to solve complicated problems of optimization, such as those found in reactive optimization, distribution systems planning, expansion of transmission systems etc. [1]-[12].

The evolution-based algorithm are referred with the common term "Evolutionary Computation". There are three main approaches where the majority of current implementations are classified:

- Genetic Algorithms (GA's);
- Evolution Strategies (ES's);
- Evolutionary Programming (EP).

Each of these main stream algorithms have clearly demonstrated their capability to yield good approximate solutions even in case of complicated multimodal, discontinuous, non-differentiable, and even noisy or moving response surfaces of optimization problems [13] [20]. In these approaches, a population of individuals is initialized and then evolves into a search space, throughout a stochastic process of selection, mutation, and in some cases, recombination. However, these methods differ in terms of representation, operators and selection process.

Lai and Ma [3] have presented a modified evolutionary programming to solve the reactive power dispatch, obtaining good results. Other authors [10] [12] have applied the same algorithm for other power system problems, reporting results using IEEE14 and IEEE30 systems. A simplified evolution strategy has been used in [12] and compared with genetic algorithms and Lai and Ma algorithm. More recently, a proposal quite similar to [3] has been presented [14]. In spite of these efforts, evolutionary techniques have not yet been explored completely for power system applications.

In this work, an evolution strategy-based approach has been proposed and compared with a state-of-art algorithm. The self-adaptation of parameters is controlled by dynamic limits and no recombination is performed. Due the probabilistic nature of the evolutionary algorithms, a comparative statistical analysis has been performed. The approach has been tested using the IEEE57 system, achieving feasible solution with losses reduction with probability 1.

This article is organized as follow. First, a brief review of problem formulation is presented. Secondly, evolutionary programming and evolution strategies are reviewed. Next, the extended algorithm are presented. Finally, validation tests with comparative analysis are performed and relevant conclusions are presented.

2 Optimal Reactive Power Dispatch

The goal of optimal reactive power dispatch is to minimize real power losses and improve voltage profile by setting generator bus voltages, VAR compensators and transformer taps. These problem can be written in a form penalized as follow:

$$Min \quad f = f_\ell + f_p \tag{1}$$

s. t.:

$$P_i^d - P_i(V,\theta) = 0, \qquad i \in N_{B-1} \tag{2}$$

$$Q_i^d - Q_i(V,\theta) = 0, \qquad i \in N_{PQ} \tag{3}$$

with

$$f_p = \sum_{i \in N_{PV}} \rho_{qi}(Q_{g_i} - Q_{g_i}^l)^2 + \sum_{i \in N_{PQ}} \rho_{vi}(V_i - V_i^l)^2 \tag{4}$$

where: f_ℓ: represents the system losses; N_B, N_{B-1}: represent the system nodes set, and the system nodes set excluding the slack bus, respectively. N_{PQ}, N_{PV}: represent the PQ-buses and PV-buses set, respectively. (2)-(3) represent the load flow equations. The generator bus voltages and the transformer tap-settings are control variables. ρ_{qi} and ρ_{vi} are penalty factors for reactive power violations and voltage violations, respectively. $Q_{g_i}^l$ and V_i^l represent the violated limits. P_i^d - Q_i^d represent the active and reactive power demand at node i, respectively. Penalty parameters are chosen empirically in accord with experience and the particular application.

3 Evolutionary Programming

The original Evolutionary Programming (*EP*) was introduced by L. Fogel in 1962 [15] and extended by Burgin, Atmar e Fogel, recently [16]. The goal of evolutionary programming is to achieve intelligent behavior through simulated evolution. While the original evolutionary programming was proposed to operate on finite machines and the corresponding discrete representations, most of present variants are utilized for continuous parameter optimization problems.

More recently, the technique has been extended and applied to diverse real-valued continuous optimization problems. Rather than use finite state machines, representations are chosen based on the problem at hand and mutation is the main operator used in generating new trials. The last version, called *meta-EP* incorporates parameter self-adaptation per individual quite similar to ES's.

Evolutionary programming was generalized by Fogel [17] to handle numerical optimization problems. In this approach, each component of candidate solution is viewed as a behavioral trait rather than a gene. The resulting change in each behavioral trait will follow a Gaussian distribution with zero mean and some standard deviation. The standard *EP* algorithm has the following structure:

$t := 0$

1. An initial population $P(t)$ with μ individuals is selected at random from a feasible region:
   $P(t) := \{\ x_1(t), ..., x_\mu(t)\ \} \in I^\mu$, where $I = R^n$
2. Calculate the fitness values $\Phi(x_i)$ from objective function values (f_i) by scaling them to positive values (function δ) and possibly by imposing some random alteration κ_i:
   $P(0) : \{\Phi(x_1(t)), ..., \Phi(x_\mu(t))\}$
   where:
   $$\Phi(x_k(i)) = \delta\ (f(x_k(i)), \kappa_i)$$

while *termination criterion not fulfilled* **do**

3. mutate: $x_k'(t) := m\{x_k(t)\ \forall k \in \{1, ..., \mu\}\}$;
4. evaluate $P'(t) := \{x_1'(t), ..., x_\mu'(t)\}$:
   $$\{\Phi(x_1'(t)), .., \Phi(x_\mu'(t))\}$$
5. select: $P(t+1) := S_q(P(t) \cup P'(t))$;
   $t := t + 1$

end do

On step 3, an offspring vector x_i' is created from each parent x_i by adding a gaussian perturbation with mean zero and a standard deviation to each component of vector x_i, as follow:

$$x_i' = x_i + \sigma_i N_i(0,1), \forall i \in 1, ..., n \tag{5}$$

where:

$$\sigma_i = \sqrt{\beta_i \Phi(s) + z_i} \tag{6}$$

n: number of control variables; β_i is a mutation scale parameter, with $0 < \beta < 1$; σ_i is the standard deviation for each individual's mutation; z_i represents an offset; $N(0,1)$ represents a gaussian random variable with mean zero and variance one.

In step 5, the population is formed, temporarily, by parents and offspring. The selection mechanism S_q reduces the set of parents and offspring individuals to a set of μ parents by performing a tournament as follow:

Each individual x_i, $i = 1, ..., 2\mu$ (combined population) must compete with other individuals to get their chance to be transcribed for the next generation (optionally, a subgroup of k individuals can be preserved for the next generation, not participating in the competition). A value w_i is associated in accordance with the competition, thus:

$$w_i = \sum_{t=1}^{q} w_t^* \qquad (7)$$

where q is the number of competitions ; w_t^* is either 0 (loss) or 1 (win) as individual x_i competes with a randomly selected individual x_r selected of combined population. Thus, w_t^* is defined as:

$$w_t^* = \begin{cases} 1 & if\ U_l < f_r/(f_r + f_i) \\ 0 & otherwise \end{cases}$$

where f_r is the *fitness* of randomly selected individual x_r and f_i is the fitness of x_i.

The value U_l is determined from a uniformly distributed set $U(0,1)$.

The individuals $i = 1, ..., 2\mu$ are ranked in descending order of de rank values w_i and the μ individuals having the highest ranks w_i are selected to form the next population.

Lai and Ma EP Algorithm (LM) [10]

An extension of canonical *EP* algorithm widely used in the power system literature has been proposed by Lai and Ma [10] They have introduced the following modification for mutation expression (5):

$$x_i' = x_i + N_i(0, \beta_k \frac{f_i}{f_{max}} \Delta x_j), \forall i \in 1, ..., n \qquad (8)$$

with $0 < \beta_k \leq 1$ and $\Delta x_j = x_{jmax} - x_{jmin}$

Additionally:

$$\beta_k = \begin{cases} \beta_{init} & if\ \ k = 0 \\ \beta_{k-1} \ - \beta_{step} & if\ \ f_{min}(k) \geq f_{min}(k-1) \\ \beta_{k-1} & if\ \ f_{min}(k) < f_{min}(k-1) \\ \beta_{final} & if\ \ \beta_{k-1} - \beta_{step} < \beta_{final} \end{cases}$$

where β_{init} is near to 1, $\beta_{final} \approx 5.10^{-3}$ and $\beta_{step} \in [10^{-3}\ ,\ 10^{-2}]$.

Thus, the mutation scale is modified during the process and in this way, prevents that the search process stops in a local minimum. Soon, the search process starts with high scale values, these will decrease during the process. The speed of scale decreasing of an individual depends on its fitness, in such a way that the lesser it is, the faster the scale diminishes. Another added modification states that the variable can not exceed their limits, the last been given by the limit value. Good results have been reported by the authors using the IEEE30 bus system.

4 Evolution Strategies (*ES*)

Evolution Strategies (ES's) were developed in 1960 by Rechemberg and Schwefel in Germany and extended by other authors, such as Rudolph and Herdy. The first evolution strategies focused on a single-parent offspring search [18]. In this model, termed (1+1)-ES, a single offspring is created for a single parent and both are placed in competition for survival with selection discarding the poorer solution. Rechemberg proposed in 1973 the use of multiple parent but only a single offspring (μ+1)-ES. More recently, two approaches have been explored, denoted by ($\mu + \lambda$)-ES and (μ, λ)-ES [19]. In the former, μ parents generate λ offspring and all solutions compete for survival with the best μ individuals being selected as parents of the next generation. In the latter, only λ offspring compete for survival and the μ parents are completely replaced in each generation. Then, the life of individual is limited to a single generation.

The process of ES is described in [19]. The following pseudocode algorithm summarizes the components of the (μ +λ)-ES evolutionary algorithm, where each individual is characterized by a pair $a = (x, \sigma_i)$:

$t := 0$

1. initialize $P(t) := \{\ a_1(t), ..., a_\mu(t)\ \} \in I^\mu$
   where $I = R^{n+n}$
   and $a_k = (x_i, \sigma_i) \forall i \in \{1, .., n\}$
2. evaluate $P(t) : \{\Phi(a_1(t)), ..., \Phi(a_\mu(t))\}$
   where $\Phi(a_k(t)) = f(x_k(t))$;

while termination criterion not fulfilled **do**

3. recombine: $a'_k(t) := r(P(t))\ \forall k \in \{1, ..., \lambda\}$;

4. mutate: $a''_k(t) := m\{a'_k(t)\ \forall k \in \{1, ..., \lambda\}$;

5. evaluate $P'(t) := \{a''_1(t), ..., a''_\lambda(t)\}$;

   $\{\Phi(a''_1(t)), .., \Phi(a''_\lambda(t))\}$

   where $\Phi(a_k(t)) = f(a_k(t))$;

6. select: $P(t+1) := S_d(P(t) \cup P'(t))$;

   $t := t + 1$

end do

Search points in ES's are n-dimensional vectors $x \in R^n$, and the fitness value of an individual is identical to its objective function value, i.e $\Phi(a) = f(x)$ where x is the object variable component of a and each individual include up to n different variances σ_i ($i \in \{1, ..., n\}$).

Different recombination mechanisms are used in ES's either in their usual form, producing one new individual from two randomly selected parent individuals, or in their global form, allowing the taking of components for one new individual from potentially all individuals available in the parent population. Furthermore, recombination is performed on strategy parameters as well as on the object variables, and the recombination operator may be different for object variables and standard deviations.

The mutation operator $m : I \to I$ (where $I = R^{n+n}$) yields a mutated individual $m(\vec{a}) = (\vec{x}', \vec{\sigma}')$, by first mutating the standard deviations and them mutating the object variables as follow:

$$\sigma'_i = \sigma_i exp(\tau' N(0,1) + \tau N_i(0,1)) \tag{9}$$

$$x'_i = x_i + \sigma'_i N(0,1) \tag{10}$$

The global factor $\tau' N(0,1)$ allows for any overall change of the mutability, whereas $\tau N_i(0,1)$ allows for individual changes of σ_i. The parameters τ and τ' are suggested by Schewfel [21] as $\tau = (\sqrt[4]{4n})^{-1}$ and $\tau' = (\sqrt{2n})^{-1}$.

In constrast with *EP*, selection in *ES's* (S_d) is completely deterministic, selecting the μ best individuals from the union of parents and offspring (($\mu + \lambda$)-selection). The selection is elitist and therefore guarantees a monotonic improving performance.

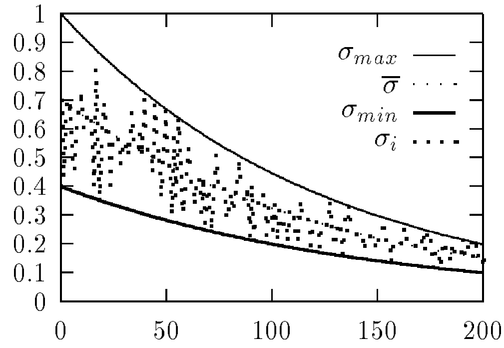

Figure 1: Dynamic upper bound and lower bound functions

5 Extended *ES* Algorithm

Due the nature of evolutionary algorithms, best solutions are expected by increasing generation number and population size. However, in practical application, solutions in reasonable time period are required and in many cases local solutions fulfills these practical requirements. In this section we present a new extension of *ES* algorithm, so-called Bounded Evolution Strategy Approach - *BES*.

Dynamic limits

In this paper, standard $(\mu + \lambda)$-ES algorithm has been modified. The main modification is addressed to limit σ mutations by introducing dynamic upper and lower bounds. Moreover, modified ES algorithm is performed without recombination.

Dynamic limits allow σ mutations fall into an upper and lower limit, both dynamically decreasing exponentially, as follow:

$$\sigma(t)_{max} = \sigma^o_{max} exp(-t/T_1) \tag{11}$$

$$\sigma(t)_{min} = \sigma^o_{min} exp(-t/T_2) \tag{12}$$

where σ^o_{max} and σ^o_{min} are initial values for each function and t denotes the generation; T_1 and T_2 are time constants calculated from final values desired for σ^f_{max} and σ^f_{min}, respectively. If any dynamic limit is violated, then $\sigma(t)$ will be given the average of current values of functions above.

Eq. (11-12) allow "large" mutations in the initial generations and 'small" mutations at the end. In other words, in the first iterations diversity is emphasized while the last generations are dominated by a refined search process (small mutations). Figure (1) illustrates a example of dynamic limits.

In addition to dynamic limits, two modification have been introduced: first, *BES* algorithm is performed without considering the recombination operator. Secondly, creation of an offspring is performed taking in account the feasible range of the variable, similar to *LM* proposal, as follow:

$$x'_i = x_i + \sigma'_i(x_i^{max} - x_i^{min})N(0,1) \qquad (13)$$

where x_i^{max} - x_i^{min} are the limits of control variable x_i. If x_i exceeds its limit, x_i will be given the limit value.

These proposed algorithm has been implemented in this work and compared with the *LM* algorithm. In the next section, practical aspects related with implementation are presented.

Param	LM	BES_{100}	BES_{200}
μ	60	30	30
λ	60	60	60
T_{max}	200	100	200
β_o	0.10	-	-
β_f	5.10^{-4}	-	-
β_{step}	0.10^1	-	-
Selection	S_q	S_d	S_d
q	60	-	-

Table 1: Parameters of implemented algorithms.

6 Implementation Details

The implemented algorithm basically follows the sequence presented in section 4. The fitness function $\Phi(s)$ corresponds to penalized objective function f, given by eq. (1). Control vector x is formed by generator bus voltages and transformer tap-settings. The objective function penalty factors utilized have been $\rho_{qi} = 10^4$ and $\rho_{vi} = 10^5$.

Initial population of μ candidates solutions satisfying eq. (2-3) is generated at random.

Parameters used in practical implementation

Parameters of dynamic limits (11 - 12) have been assumed as σ^o_{min}=10^{-2}, σ^o_{max}=1; σ^f_{min}=10^{-4} and σ^f_{max}=10^{-2}. Other specific parameters of implemented algorithms are presented in table (1).

[1] If in two consecutive generations fitness do not diminishes, then β is reduced by β_f. It is a small modification introduced in *LM* algorithm that improves the performance

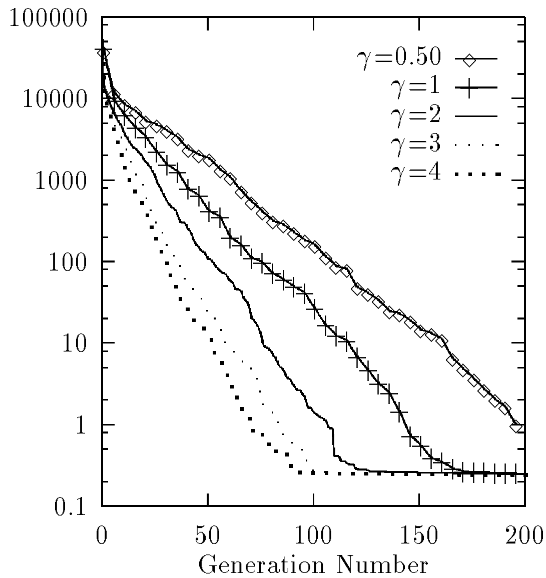

Figure 2: Effect of γ on behavior of *BES* algorithm.

In *EP* algorithms, parameter q defines the number of competitions to which each individual of combined population is submitted. A small value for q leads to random behavior of W_i. On the other hand, a very high q compared with population tends to deterministic behavior [13]. In this work, it has been assumed $q = \mu$.

Reactive Limits							
Bus	1	2	3	6	8	9	12
Q_g^{min}	-1.40	-0.17	-0.10	-0.08	-1.40	-0.03	-0.5
Q_g^{max}	2.00	0.50	0.60	0.25	2.00	0.09	1.55

Table 2: Reactive power generation limits.

Taps and Voltage Limits					
PV Buses		PQ Buses		Taps	
V_g^{min}	V_g^{max}	V^{min}	V^{max}	a^{min}	a^{max}
0.9	1.1	0.95	1.05	0.90	1.10

Table 3: Tap-setting and voltage limits.

7 Test Results

The *EP* approach proposed by Lai and Ma [10], and the *BES* algorithm have been implemented. Tests have been performed using the IEEE57 standard system. The network consist of 7 generator-buses,

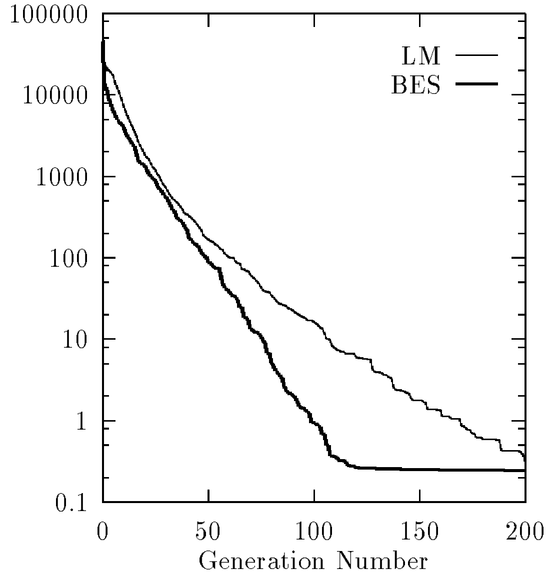

Figure 3: Comparative performance of *LM* and *BES* algorithms.

50 load-buses and 80 branches, of which 17 branches are under load-tap-setting transformer branches. All power and voltage quantities are per-unit values.

Tables (2)- (3) show the main characteristics of these system. Voltage limits have been considered $0.9 - 1.10$ pu for PV-buses and $0.95 - 1.05$ for PQ-buses. Taps limits have been assumed $0.90 - 1.10$. Total losses in case-base have been 0.2793 pu.

Due to probabilistic characteristic of evolutionary algorithms, results reported here correspond to average from 20 trials. From practical point of view, we are interested in reliable software tools that supply good solutions every time. Thus, to evaluate the quality of the proposals, dispersion measures are fundamental.

The modified algorithm *BES* are based on the $(\mu + \lambda) - ES$, without recombination. Two cases 100 and 200 generations, respectively, have been simulated. These approach is sensible to choose of ratio $\gamma = \lambda/\mu$, i.e., the offspring population size related to parent population. In order to state a suitable ratio for it, a behavior study has been performed varying λ and keeping the parent population in $\mu = 30$. Result of this study is shown in fig (2). For ratios γ lesser than the unity, the process evolves slowly. As soon as γ increases the evolution speed also increases. However, for high γ values, no significant improvement is observed. Thus, when choosing γ, must be achieved the best evolution speed with reasonable offspring population. In the practical implementation, it has been chosen $\gamma = 2$.

Figure (3) shows the comparative performance of *LM* and *BES* algorithms. Tests have been performed for 200 generations. Each point corresponds to the average (over 20 trials) of the best fitness on the current generation.

The algorithm proposed outperform the algorithm of Lai and Ma.

In the following, in order to validate the proposed approaches of robustness point of view, a statistical analysis is presented.

Tables (4) and (5) show the best solutions of the control variables obtained by *LM*, and *BES* methods, respectively. Case-base (CB) values are included. The best solution has been obtained with *BES* method, achieving losses reduction of 13.48%.

Generator Bus Voltages			
Bus	CB	LM	BES_{200}
1	1.0400	1.0745	1.0725
2	1.0100	1.0618	1.0596
3	0.9850	1.0510	1.0484
6	0.9800	1.0515	1.0424
8	1.0050	1.0692	1.0662
9	0.9800	1.0335	1.0344
12	1.0150	1.0401	1.0440
f/f_ℓ^r %	100	88.93	86.52

Table 4: Generator bus voltages.

Table (6) presents comparative analysis of implemented algorithms relative to losses minimization. Tests have been performed over 20 trials discarding 2 worst cases. Columns 2-4 show minimum, maximum, average and standard deviation of losses, respectively. Column 5 gives average losses relative to case-base losses. This table is important because shows the robustness and reliability of proposals. We have in mind an algorithm that supplies optimal and feasible solutions with minimal dispersion. The best result has been obtained with *BES* for 200 generations. However, notice that good results have been already obtained for 100 generations.

Table (7) shows the effect of γ on the performance of *BES* over 200 generations. Clearly, as soon as γ increases, best quality solution are obtained in terms of objective function and dispersion measures.

Up till now, it has been analyzed results of algo-

Tranformer Tap-settings			
Branch	CB	LM	BES_{200}
(4,18)	0.9700	1.0190	1.0443
(4,18)	0.9780	1.0190	1.0443
(20,21)	1.0430	0.9248	1.0076
(24,25)	1.000	0.9252	1.0097
(24,25)	1.000	0.9252	1.0097
(24,26)	1.0430	1.0416	1.0097
(7,29)	0.9670	1.0136	1.0435
(32,34)	0.9750	0.9890	0.9803
(11,41)	0.9550	0.9003	1.0221
(15,45)	0.9550	1.0378	1.0381
(14,46)	0.900	0.9751	1.0220
(10,51)	0.9300	0.9707	1.0296
(13,49)	0.8950	0.9272	1.0248
(11,43)	0.9580	1.0430	1.0221
(40,56)	0.9580	1.0884	0.9835
(39,57)	0.9800	0.9737	0.9899
(9,55)	0.9400	1.0467	1.0344

Table 5: Transformer tap-settings.

Global Performance					
Method	f_ℓ^{min}	f_ℓ^{max}	$\overline{f_\ell}$	$\sigma_\ell\%$	$\overline{f_\ell^r}\%$
LM	0.2484	0.2922	0.2641	4.89	94.56
BES_{100}	0.2438	0.2630	0.2541	2.30	90.99
BES_{200}	0.2417	0.2486	0.2443	0.82	87.47

Table 6: Statistic performace of *LM* and *BES* algorithms.

μ=30, T_{max}=200					
γ	f_ℓ^{min}	f_ℓ^{max}	$\overline{f_\ell}$	$\sigma_\ell\%$	$\overline{f_\ell^r}\%$
0.5	.2554	2.5655	.6962	98.96	249
1	.2454	.2652	.2548	2.81	91.23
2	.2434	.2572	.2481	1.79	88.83
3	.2406	.2474	.2438	.74	87.29
4	.2415	.2469	.2436	.55	87.22
5	.2410	.2496	.2433	.87	87.11
6	.2399	.2475	.2426	.71	86.86
7	.2403	.2443	.2421	.44	86.68

Table 7: Statistical performance of the *BES* algorithm as function of γ.

	LM			
Decil	f_ℓ/f %	f/f_ℓ^r %	V_v %	N_v
MIN	100	88.93	-	-
D_1	100	89.79	-	-
D_2	100	90.86	-	-
D_3	100	91.51	-	-
D_4	100	91.90	-	-
D_5	100	93.91	-	-
D_6	100	96.34	-	-
D_7	100	98.74	-	-
D_8	100	101.2	-	-
D_9	99.52	105.4	0.05	2
MAX	90.13	112.7	0.05	2

Table 8: Deciles distribution for *LM* algorithm.

rithms in terms of optimality and dispersion measures. In the following, algorithm performances are analyzed in terms of feasibility.

Tables (8)- (9) show test results distributed in deciles. f_ℓ/f represents a measure of feasibility of solutions. A value equal to 100% means that feasibility is maximum and no violations are registered. f/f_ℓ^r represents the ratio between objective function and case-base losses. It indicates, when feasibility is 100%, the reduction of losses with relation to case-base. In case of unfeasibility, this column has not meaning. $V_v\%$ and N_v represent the maximum voltage violation observed and the number of buses with violations, respectively. Notice that no reactive violations have been registered in all the tests.

Table (8) shows the deciles distribution for *LM* algorithm. Over 20 trials, 10% of cases have presented violations. On the another hand, *BES* algorithm show the best performance, as shows table (9). No violation is observed and in all cases losses reduction has been achieved. Furthermore, solutions obtained over 100 generations already fulfills practical requirements, i.e., reasonable losses reduction and complete feasibility are achieved.

Effect of γ on the population homogeneity

An Homogeneity index at generation k can be stated as $H_k = \overline{ff^k}/ff^k_{max}$; where $ff_i = 1/f_i$, $\overline{ff^k} = 1/\mu \sum_{i=1}^{\mu} ff_i$ and $ff_{max} = \{ff_i \setminus ff_i \geq ff_j \ \forall \ ff_j, \ j = 1, ..., \mu\}$. H_k values are on the range $1/\mu - 1$. Small values of H_k indicates high individuals diversity, while values close to one are associated

	BES_{100}		BES_{200}	
Decil	f_ℓ/f %	f/f_ℓ^r %	f_ℓ/f %	f/f_ℓ^r %
MIN	100	87.31	100	86.52
D_1	100	88.42	100	86.69
D_2	100	89.05	100	86.86
D_3	100	89.83	100	87.02
D_4	100	90.11	100	87.23
D_5	100	91.16	100	87.47
D_6	100	92.85	100	87.67
D_7	100	93.33	100	87.88
D_8	100	93.44	100	88.67
D_9	100	94.64	100	89.03
MAX	100	99.71	100	90.45

Table 9: Deciles distribution for *BES* algorithm.

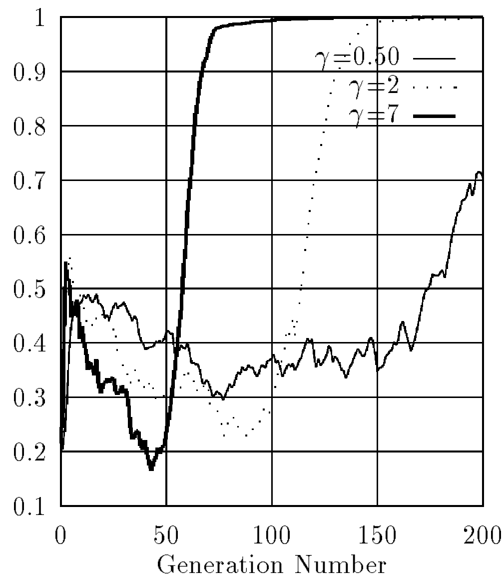

Figure 4: Effect of γ on the homogeneity.

with high homogeneity.

Figure (4) shows the effect of the ratio γ on the population homogeneity considering 200 generations. Clearly, the effect of dynamic limits (11 -12) is embedded here. Small γ values keep intermediate level of diversity; however, speed evolution is slow. High γ values lead quickly to homogeneity, but higher peaks of diversity are achieved in the beginning of process.

In general, high γ values improve the solution quality, obtaining good solutions in a reduced generation number. However, very high values are not interesting, because additional improvement in the solution does not follow the increasing of γ.

8 Conclusions

In this paper we presented an extended algorithm based on evolution strategies. In this proposal mutations in standard deviations have been controlled using dynamic limits. A comparative study between this approach and state-of-art evolutionary algorithm have been performed. In order to validate it, due to the probabilistic nature of algorithms, a statistical analysis has been presented. The proposal outperform the state-of-art algorithm. The inclusion of dynamic limits for standard deviation has shown to be fundamental in the performance.

The Comparative study has shown that *BES* algorithm be performs better than the *LM* approach. In 100% of the tests, feasible solutions with losses reduction have been achieved. Exhaustive tests were performed and reported using the standard IEEE57 system in this work.

References

[1] Iba, K., "Reactive Power Optimization by Genetic Algorithm", IEEE Trans. on Power Systems, Vol. 9 No 2, May 1994, pp. 685-692.

[2] Walters, D.C and Sheble, G.B. , "Genetic Algorithm Solution of Economic Dispach with Valve Point Loading", IEEE Trans. on Power Systems, Vol. 8 No 3, Aug 1993, pp. 1325-1332.

[3] Ma, J.T., Lai, L.L., "Optimal Reactive Power Dispatch using Evolutionary Programming", Proceedings of IEEE/KTH Stockholm Power Tech. Conf., Sweden, July, 1995, pp. 662-667.

[4] Fukuyama, Y. , Chiang, H. , "Parallel Genetic Algorithm for Generation Expansion Planning", IEEE Transaction on Power Systems, Vol 11, No 2 , May 1996, pp. 955-961.

[5] Rudnick, H., Palma, R. Cura, E., Silva C., "Economically Adapted Transmission Systems in Open Access Schemes - Application of Genetic Algorithms", IEEE Trans. on Power systems, Vol. 11, No 3, Aug 1996, pp. 1427-1440.

[6] Kazarlis, S.A., Bakirtzis, A. G., Petridis, V., "A Genetic Algorithm Solution to the Unit Commitment Problem", IEEE Transactions on

Power Systems, Vol 11, No 1, Feb, 1996, pp. 83-92.

[7] Lee, Y., Bai X., Park, Y., "Optimization Method for Reactive Power Planning using a Modified Simple Genetic algorithm", IEEE Transaction on Power Systems, Vol 10, No 4, Nov 1995, pp. 1843-1850.

[8] Yeh, E.C., Venkata, S.S., Sumic Z., "Improved Distribution System Planning using Computational Evolution", IEEE Transaction on Power Systems, Vol 11, No 2, May 1996.

[9] Nara, Koichi, et al.,"Implementation of Genetic Algorithm for Distribution Systems Loss Minimun Re-Configuration", IEEE Trans. on power Systems, Vol. 7 No 3, Aug 1992, pp. 1044-1051.

[10] Lai, L.L., Ma, J. T., "Application of Evolutionary Programming to Reactive Power Planning - Comparison with Nonlinear Programming Approach", IEEE Trans. on Power systems, Vol. 12, No 1, Feb 1997, pp. 198-206.

[11] Wong, K. P., Wong Y. W., "Combined Genetic algorithm/ Simulated Annealing/Fuzzy Set Approach to Short-Term Generation Scheduling with Take-or-Pay Fuel Contract", IEEE Trans. on Power Systems, Vol 11, No 1, Feb 1996, pp. 128-136.

[12] Lee, K. Y. , Yang, F.F. , "Optimal Reactive Planning using Evolutionary Algorithms: A Comparative Study for Evolutionary Programming, Evolutionary Estrategy, Genetic Algorithm and Linear Programming", IEEE Trans. on Power Systems, Vol 13, No 1, Feb 1998, pp. 101-108.

[13] Bäck, T., Schwefel, H.P., "An Overview of Evolutionary Algorithms for Parameter Optimization", Evolutionary Computation, 1993, pp. 1 - 27.

[14] Park, Y., Won J., Park J., Kim D., "Generation Expansion Planning Based on an Advanced Evolutionary Programming", IEEE Trans. on Power Systems, Vol 14, No1, Feb 1999, pp. 299-305.

[15] Fogel, L. J. , "Autonomous Automata", Industrial Research, Vol 4, 1962, pp. 14-19.

[16] Fogel, D. B., "Evolutionary Programming and Optimization", Conference Record of The XXVI ASILOMAR conference on Signal, Systems and Computers, IEEE Comput. Soc. Press, Pacific Grove, CA, USA, 26-28/out. 1992, pp. 409-414.

[17] Fogel, D. B., "Evolutionary Optimization", Conference Record of The XXVI ASILOMAR Conference on Signal, Systems and Computers, IEEE Comput. Soc. Press, Pacific Grove, CA, USA, 26 - 28 oct. 1992, pp. 409-414.

[18] Rechenberg, I., "Cybernetic Solution Path of and Experimental Problem", Royal Aicraft Establishment, Library Translation No 1122, aug, 1965.

[19] Bäck, T., Hammel, U. , Schwefel, H. P. "Evolutionary Computation: An Overview', in Proc. 3rd IEEE Conf. on Evolutionary Computation. Piscataway, NJ: IEEE Press, 1996, pp. 20 - 29.

[20] Bäck, T., Hammel, U. , Schwefel, H. P., "Evolutionary Computation: Comments on the History and Current State", IEEE Trans. on Evolutionary Computation, Vol 1, No 1, Apr, 1997, pp. 3 - 17.

[21] Bäck T., Rudolph, G. and Schwefel, H.P. "Evolutionary Programming and Evolution Strategies: Similarities and Differences", Proc. of European Conf. on Alife, Granada, Spain, 1995 .