A CONTRIBUTION TO THE STUDY ABOUT DIFFERENTIAL EVOLUTION

UMA CONTRIBUIÇÃO AO ESTUDO SOBRE EVOLUÇÃO DIFERENCIAL

Giovana Trindade Silva de Oliveira* and Sezimária F. Pereira Saramago**

Universidade Federal de Uberlândia – Campus Santa Mônica
*Faculdade de Engenharia Mecânica
**Faculdade de Matemática
Av. João Naves de Ávila, 2121
CEP: 38408-100 Uberlândia, MG Brasil
**E-mail: saramago@ufu.br

ABSTRACT

The advance of the computational resources has encouraged the utilization of optimization techniques in the solution of complex engineering problems. Thus, it is very attractive to consider the possibility of joining the feature of natural optimization methods to one algorithm which allows to work with small populations and to reduce computational time greatly. The Differential Evolution (DE) is a simple evolutionary algorithm which has these advantages. The most distinct feature of DE is to perturb individuals of a population by weighted difference among random population individuals. In this work, the theoretical formulation, the basic algorithm and an application are presented. The simplicity and efficiency of the algorithm in terms of easy implementation are demonstrated by an engineering problem. The results are compared with those obtained by using Genetic Algorithms.
Keywords: Optimization, Evolutionary algorithm, Differential evolution, Multi-objective optimization.

RESUMO

O avanço dos recursos computacionais tem incentivado a utilização de técnicas de otimização na solução de problemas complexos de engenharia. Assim, torna-se muito atrativa a possibilidade de unir as características dos métodos naturais de otimização a um algoritmo que permita trabalhar com uma população pequena e uma grande redução do tempo computacional. A Evolução Diferencial (ED) é um algoritmo evolutivo simples que possui estas vantagens. A característica mais importante da ED é perturbar os indivíduos de uma população por meio de uma diferença ponderada entre os indivíduos da população escolhidos aleatoriamente. Neste trabalho, são apresentados a formulação teórica do método, um algoritmo básico e uma aplicação. A simplicidade e eficiência do algoritmo em termos de sua fácil implementação são demonstrados através de um problema de engenharia.Os resultados são comparados com os obtidos usando Algoritmos Genéticos.
Palavras-chave: Otimização, Algoritmos evolutivos, Evolução Diferencial, Otimização multi-objetivo.

1 – INTRODUCTION

During the last decades natural optimization methods, or stochastic algorithms, also known as adaptive random search methods have become an important tool in business, science, and engineering. It is possible to solve problems with hundreds or thousands of variables and many local optimal points. Most of the traditional optimization techniques based on gradient methods generate the possibility of getting trapped at local optimum depending upon the degree of non-linearity, non-differentiability and initial guess of the objective function [1]. Hence, these traditional optimization techniques do not ensure global optimum and have also limited applications. In this case, methods based on the principle of evolution, i.e. survival of the fitness individuals, as Genetic Algorithms [2], [3] and Evolution Strategies [4] have been very useful. These methods are referred to as Evolutionary Algorithms or Evolutionary Computation methods.

Differential Evolution (DE) is an algorithm developed by Storn and Price [5] in 1995 that belongs to the class of evolutionary algorithms. The main idea is to generate new individuals by adding the weighted difference between two population individuals to a third individual. Among the DE's advantages are its simple structure, easiness of use, speed and robustness, and greater probability of finding a function's true global optimum. DE has been successfully applied to solving several complex problems such as system design [6], solution of the linear systems [7], robot manipulator design [8], and has been identified as a potential source for accurate and fast optimization.

Differential Evolution algorithm is a heuristic approach for minimizing nonlinear, non-differentiable, and multimodal cost functions. It is similar to Genetic Algorithms and it utilizes a population with *Np* *n*-dimensional real parameter vectors, i.e. its overall structure resembles that of most other population based searches. In addition, to cope with computation intensive cost functions, DE uses a vector population where the stochastic perturbation of the population vectors can be done independently, resulting in a fast optimization.

Another very important feature of DE is its easy use, i.e. few control variables to steer the minimization. These variables should be robust and easy to choose as well, since the minimization method is self-organizing so that very little input is required from the user. DE borrows the idea from Nelder and Mead [9] of employing information from within the vector population to alter the search space. DE's self-organizing scheme takes the difference vector of two randomly chosen population vectors to perturb an existing vector. This crucial idea is in contrast to the method used by traditional Evolution Strategies in which predetermined probability distribution functions define vector perturbations.

In this study, Differential Evolution is used to solve the environmental economic dispatch problem of electric power generation. This problem has been deserving the several researchers' attention [10, 11, 12]. In this application, the aim is to select the generating unit outputs so as to meet the load demand at minimum operating cost and minimum pollution by atmospheric emission while satisfying all unit and system constraints. The optimum are compared with those obtained using Genetic Algorithms and the results obtained by Coelho and Mariani [13].

2 – GENETIC ALGORITHMS (GA)

The genetic algorithms were introduced by Michalewicz [14] in 1995 and they can be understood as a process of directed random search. The main characteristics of this technique are: GA operates on a population of points and, differently from the conventional methods, they do not invest all the search effort on only one point; they operate in a space of coded solutions, and not in the search space directly; they do not require derivation, unimodality or any other function knowledge to operate - they only need the objective function value for each individual of the population; they use probabilistic transitions and not deterministic ones. This way, similarly to the evolution process in the search for the most adapted individuals throughout successive generations, the optimization procedure improves the solutions until the optimal one is found.

A simple genetic algorithm performs basically three operations: selection, crossover and mutation. The initial population made up of *Np* individuals is usually generated in a random way or through some heuristic process. As in the natural genetics, there is not evolution without variety, and, for this reason, it is important that the individuals have different adaptation degrees to the environment in which they live. This means that the initial population covers in the best possible way the search space. In the operation *Selection* a temporary population of *Np* individuals is generated considering the proportional probability of each individual with respect to its relative adaptability in the population. The individuals presenting low adaptability will have more chance to disappear. The operation *Crossover* works in the sense of selecting two individuals that will exchange genetic material. It is also a random process that occurs with probability fixed by the user. *Mutation* is, as in nature, an event of rare occurrence. Its purpose is to guarantee that important genetic material is not hopelessly lost [2].

3 – DIFFERENTIAL EVOLUTION (DE)

Let the initial population chosen randomly consisting of *Np* individuals called vectors. This population should cover the entire search space. For a problem with *n* design variables each vector has *n* parameters. Generally, this population is created by a uniform probability distribution. In this way the population follows a natural evolution, but *Np* does not change during the minimization process.

The main idea of differential evolution is to generate new individuals, called mutated vector or donor vector, by adding the weighted difference between two random population individuals to a third individual. This operation is called *mutation*. The new donor individual's parameters are then mixed with the parameters of another individual randomly chosen, denoted target vector or vector to be replaced, to yield the called trial vector. This process is often referred to as *crossover* in the evolutionary strategy community. If the trial vector cost yields a lower value than the target vector cost, then the trial vector replaces the target vector in the following generation. This last operation is called *selection*.

The process is ended when limiting the maximum number of generations or through the stagnation concept, i.e. when after several serial iterations no improvement in the population is observed. This methodology has great potential for the solution of optimization problems.

3.1 Differential evolution operations

The differential evolution operations are based on a natural evolution principle whose aim is to keep the population diversity.

Mutation: For the purpose of obtaining the mutated vector $V^{(q+1)}$, let the vectors X_α, X_β and X_γ mutually different and randomly chosen from the population with *Np* individuals, so that $Np \geq 4$. The random indexes $\alpha, \beta, \gamma \in \{1,\dots, Np\}$ are integer mutually different. In generation *q* one pair of vectors (X_β, X_γ) defines a difference vector ($X_\beta - X_\gamma$). *F* multiplies this difference named weighted difference, and it is used to perturb the third vector X_α or the best vector X_{best}. *F* is a real and constant factor $\in [0,2]$ which controls the amplification of the difference vector. This process that yields the mutated vector $V^{(q+1)}$ can be mathematically written as:

$$V^{(q+1)} = X_\alpha^{(q)} + F(X_\beta^{(q)} - X_\gamma^{(q)}) \qquad (1)$$

Figure 1 shows a two-dimensional function that illustrates the different vectors which ~~to~~ take part in the generation of mutated vector.

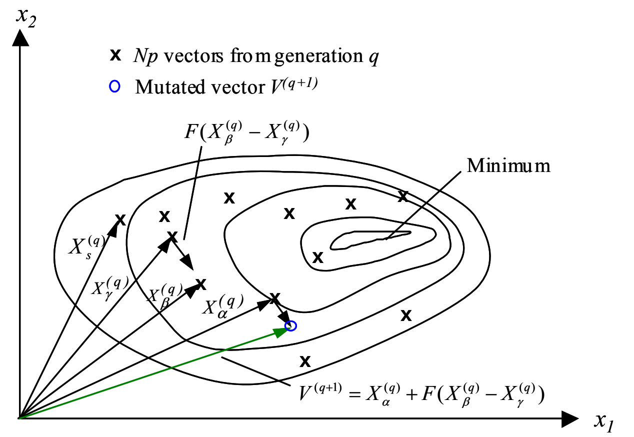

Figure 1 – The process for generating $V^{(q+1)}$ for a two-dimensional function.

Crossover: Consider that for each target vector $X_s^{(q)}$, $s \in \{1,..., Np\}$, different from indexes α, β, γ, a mutated vector $V^{(q+1)}$ was generated. The crossover is introduced in order to increase the diversity of the perturbed individuals. Thus, as represented in Figure (2)-a, the trial vector $U^{(q+1)}$ is formed by:

$$u(i)^{(q+1)} = \begin{cases} v(i)^{(q+1)}, & if \;\; r_i \le Pc. \\ x_s(i)^{(q)}, & if \;\; r_i > Pc, \quad i = 1,..., n. \end{cases} \tag{2}$$

where r_i is ith-evaluation of a uniform random number generator $\in [0, 1]$, $Pc \in [0, 1]$ is the crossover probability and it must be supplied by user. Pc represents the probability of the new trial vector to inherit the variable values from mutated vector. When $Pc = 1$, for example, all trial vector variables will come from mutated vector $V^{(q+1)}$. On the other hand, if $Pc = 0$, all trial vector variables will come from the target vector $X_s^{(q)}$.

This crossover, developed by Storn and Price [5], is called binomial crossover operator, due to independent binomial experiments, which is executed whenever a randomly picked number r is lower than the Pc crossover probability.

Some years later, Storn and Price [15] developed more strategies using exponential crossover operator, in which the crossover is executed on the n variables while the random number r is lower than the Pc crossover probability. The exponential crossover can be observed in Figure 2 –(b). In this case, the first time a randomly picked number is bigger than the Pc the crossover operator is stopped, i.e:

$$\text{while} \;\; r_i \le Pc, \;\; u(i)^{(q+1)} = v(i)^{(q+1)}.$$

$$\text{if } r_i > Pc, \text{ then } u(j)^{(q+1)} = x_s(j)^{(q)}, \; j = (i+1),..., n. \tag{3}$$

An example is illustrated in Figure 2 - (a) and (b) which show the binomial crossover process and exponential crossover process, respectively, both with seven design variables.

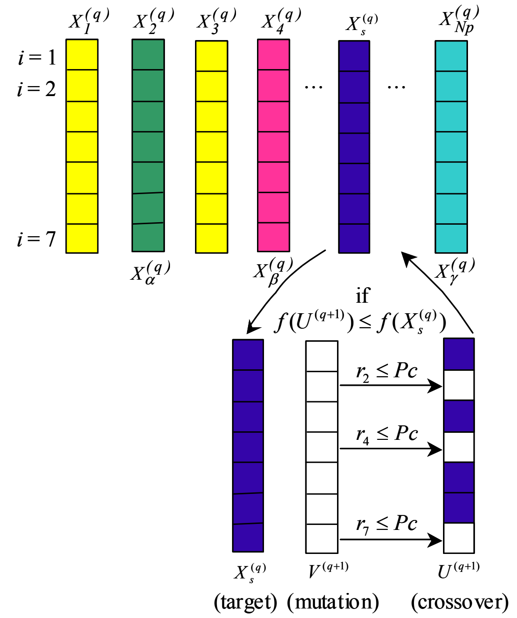

Figure 2 – (a). Illustration of the binomial crossover

After the crossover, if one or more trial vector variables are out of search space then it can be brought in the bound range as following:

$$\text{If } u(i) < x(i)^{\min}, \text{ then } u(i) = x(i)^{\min};$$

$$\text{If } u(i) > x(i)^{\max}, \text{ then } u(i) = x(i)^{\max}, \; i = 1,..., n. \tag{4}$$

where $x(i)^{min}$ and $x(i)^{max}$ are the lower and upper limits, i. e. the side constraints, respectively.

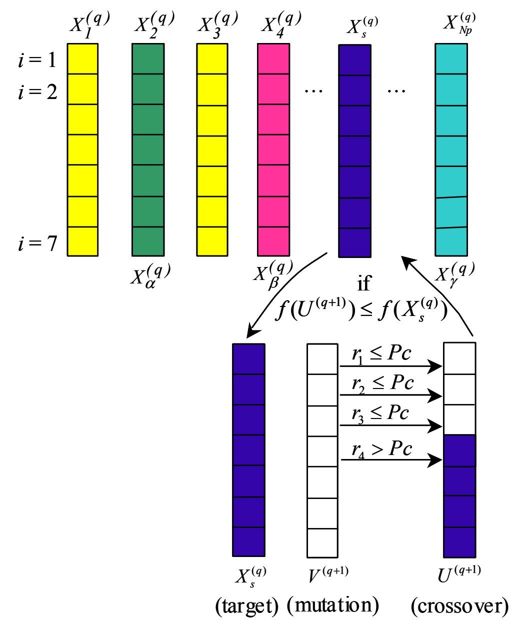

Figure 2 – (b). Illustration of the exponential crossover

Selection: The selection is the process of producing better offspring. Unlike many other evolutionary algorithms, the DE does not use ranking and proportional selection. Instead, the cost of each trial vector $U^{(q+1)}$ is worked out and compared with the cost of target vector $X_s^{(q)}$. If the cost of target vector is lower than that of trial vector, the target is allowed to advance for the next generation *q+1*. Otherwise, the trial vector replaces the target vector in the following generation. In other words this process can be written as:

$$\text{if } f(U^{(q+1)}) \le f(X_s^{(q)}) \text{ then } X_s^{(q+1)} = U^{(q+1)};$$
$$\text{if } f(U^{(q+1)}) > f(X_s^{(q)}) \text{ then } X_s^{(q+1)} = X_s^{(q)}. \qquad (5)$$

Usually, the DE algorithm performance depends mainly on the *Np* population size, search space, and crossover probability.

3.2 Differential evolution strategies

DE different strategies can be obtained altering the way to obtain the mutation operator. In the most used strategies, the mutation can vary according to: the type of individual (X_α) to be modified in the process of formation of mutated vector; the number of difference vectors considered; and the type of crossover to be used.

To classify the different variations, Storn and Price [5] have introduced the following notation: DE/a/b/c, where the symbol *a* specifies the vector to be perturbed which currently can be "rand" (a randomly chosen population vector) or "best" (the vector of lowest cost from the current population), *b* is the number of difference vectors used for perturbation of vector *a*, and *c* denotes the crossover type (bin: binomial; exp: exponential).

Using this notation and supposing binomial crossover, the Eq. (1) can be classified as DE/rand/1/bin.

If the population vectors number *Np* is high enough, the population diversity can be improved using two difference vectors to perturb an existing vector, i.e. five distinct vectors are chosen randomly from the current population, the weighted difference uses two pairs of difference vectors and is used to perturb the fifth vector, X_α (or the best vector X_{best}), of current population *q*. This process can be written as:

$$V^{(q+1)} = X_\alpha^{(q)} + F\,(X_\rho^{(q)} - X_\beta^{(q)}) + F\,(X_\gamma^{(q)} - X_\delta^{(q)}) \qquad (6)$$

or

$$V^{(q+1)} = X_{best}^{(q)} + F\,(X_\rho^{(q)} - X_\beta^{(q)}) + F\,(X_\gamma^{(q)} - X_\delta^{(q)}) \qquad (7)$$

with random integer indexes $\alpha, \beta, \gamma, \rho, \delta \in \{1,\ldots, Np\}$ mutually distints, so that $Np \ge 6$. According to crossover adopted, the Eq. (6) can be written as DE/rand/2/bin or DE/rand/2/exp. In a similar way, the Eq. (7) can be classified as DE/best/2/bin or DE/best/2/exp.

There are two other strategies where the mutated vector has contributions of the best population individual, X_{best}, and some previous generation individual, X_{old}, and the following weighted differences:

$$V^{(q+1)} = X_{old}^{(q)} + F\,(X_{best}^{(q)} - X_{old}^{(q)}) + F\,(X_\gamma^{(q)} - X_\delta^{(q)}) \qquad (8)$$

The Eq. (8) can be represented by strategies DE/rand-to-best/2/bin or DE/rand-to-best/2/exp, it depends on crossover type.

Summarizing, all the strategies can be described according to Table 1.

It is worthwhile to note that a strategy that works well for a given problem may not to work well when applied to a different problem. Also, the strategy to be adopted for a problem is determined by trial and error.

3.3 Differential evolution for constraint multicriterion optimization problems

A multicriterion optimization problem can be formulated as finding a vector of decision variable $X = [X_1, X_2, \ldots, X_n]^T$ which optimizes a vector function whose elements represent the objective functions and satisfy the inequality, equality, and side constraints. The problem can be written as follows:

$$\text{Minimize } f(X) = [f_1(X), f_2(X), \ldots, f_k(X)]\,, \quad X \in \Re^n \qquad (9)$$

$$\text{Subject to } \begin{cases} g_j(X) \le 0, & j = 1,\ldots, J. \\ h_l(X) = 0, & l = 1,\ldots, L. \\ X_i^{\min} \le X_i \le X_i^{\max}, & i = 1,\ldots, n. \end{cases} \qquad (10)$$

Table 1 - Representation of differential evolution strategies

Number	Mutation	Notation
1	$V^{(q+1)} = X_\alpha^{(q)} + F(X_\beta^{(q)} - X_\gamma^{(q)})$	ED/rand/1/bin
2	$V^{(q+1)} = X_{best}^{(q)} + F(X_\beta^{(q)} - X_\gamma^{(q)})$	ED/best/1/bin
3	$V^{(q+1)} = X_\alpha^{(q)} + F(X_\lambda^{(q)} - X_\beta^{(q)} + X_\gamma^{(q)} - X_\delta^{(q)})$	ED/rand/2/bin
4	$V^{(q+1)} = X_{best}^{(q)} + F(X_\alpha^{(q)} - X_\beta^{(q)} + X_\gamma^{(q)} - X_\delta^{(q)})$	ED/best/2/bin
5	$V^{(q+1)} = X_{old}^{(q)} + F(X_{best}^{(q)} - X_{old}^{(q)} + X_\gamma^{(q)} - X_\delta^{(q)})$	ED/rand-to-best/2/bin
6	$V^{(q+1)} = X_\alpha^{(q)} + F(X_\beta^{(q)} - X_\gamma^{(q)})$	ED/rand/1/exp
7	$V^{(q+1)} = X_{best}^{(q)} + F(X_\beta^{(q)} - X_\gamma^{(q)})$	ED/best/1/exp
8	$V^{(q+1)} = X_\alpha^{(q)} + F(X_\lambda^{(q)} - X_\beta^{(q)} + X_\gamma^{(q)} - X_\delta^{(q)})$	ED/rand/2/exp
9	$V^{(q+1)} = X_{best}^{(q)} + F(X_\alpha^{(q)} - X_\beta^{(q)} + X_\gamma^{(q)} - X_\delta^{(q)})$	ED/best/2/exp
10	$V^{(q+1)} = X_{old}^{(q)} + F(X_{best}^{(q)} - X_{old}^{(q)} + X_\gamma^{(q)} - X_\delta^{(q)})$	ED/rand-to-best/2/exp

In this work, the weighting objectives method is used to change the multicriterion optimization problem to a scalar optimization problem [16] by creating one function of the form:

$$f(X)=\sum_{k=1}^{K} w_k \frac{f_k(X)}{f_k^0} \tag{11}$$

where the weighting coefficients w_k are usually assumed as:

$$\sum_{k=1}^{K} w_k = 1 \tag{12}$$

In the Eq. (11), the vector function is normalized by using the ideal solution f_k^0, which is determined by obtaining attainable minima for all the objective functions separately. In others words,

$$\begin{cases} f_k^0 = \min f_k(X), \quad k=1,\dots,K \\ \\ \text{subject to constraints (10)} \end{cases} \tag{13}$$

The Differential Evolution was developed to unconstraint problems. So, in the case of constrained optimization problems, it is necessary to introduce modifications in this method. This work uses the concept of Penalty Function [1]. In this technique, the problems with constrains are transformed in unconstrained problems adding a penalty function *P(X)* to the original objective function to limit constraint violations. This new objective function, called pseudo-objective function, is penalized, according to a factor, every time that meets an active constraint. Let the pseudo-objective function be given in the form:

$$\Phi(X)=f(X)+r_p\,P(X) \tag{14}$$

$$P(X)=\left[\sum_{j=1}^{J}\left\{\max\left[0\,,\,g_j(X)\right]\right\}^2+\sum_{l=1}^{L}\left[h_l(X)\right]^2\right] \tag{15}$$

where *f(X)* is the original objective function given in the Eq. (11), *P(X)* is an imposed penalty function given by Eq. (15), g_j are the inequality constraints, h_l are the equality constraints. The scalar r_p is a multiplier that quantifies the magnitude of the penalty. For the efficiency of the method, a large value of the penalty factor r_p should be used to ensure near satisfaction of all constraints.

4 – NUMERICAL APPLICATION

In this paper, all the strategies describe in Table 1 were used for the solution of two environmental economic dispatch problem (considering power outputs of the 6 and 13 generators). The obtained results were compared. The computational code was implemented in MATLAB®. The optimal results obtained by using differential evolution were compared with the obtained results using genetic algorithms. The program Genetic Algorithms Optimization Toolbox (GAOT) developed by Houck *et al*. [17] has been used to perform the GA, adopting Np = 80 individuals. The parameters used for DE were: number of population vector Np = 15; multiplier of the difference vector F = 0.8; constant probability crossover CR = 0.5; stopping criterion $iter_{max}$ = 200 generations; penalty factor $r_p=1000$, and weighting coefficients $w_1 = w_2 = 0.5$.

Case 1: environmental economic dispatch considering the minimum operating cost and minimum pollution (6 generators)

The aim of environmental economic dispatch of electric power generation is to select the generating unit outputs. Hence to meet the load demand at minimum operating cost and minimum pollution by atmospheric emission while satisfying all unit and system constraints. Thus, the objective is to minimize two competing objective functions, fuel cost and emission, while satisfying equality and side constraints. The vector of real power outputs of the *n* generators is represented by $X = [X_1, X_2, \dots, X_n]^T$, where X_i, $i=1,\dots,n$ are the design variables.

The generator cost curves $F_c(X)$ are represented by quadratic functions. Thus, the total $/h fuel cost can be expressed as:

$$F_c(X)=\sum_{i=1}^{n} a_i + b_i X_i + c_i X_i^2 \tag{16}$$

where *n* is the number of generators, a_i, b_i, and c_i, are the cost coefficients of the *i*th-generator and X_i is the real power output of the *i*th-generator.

The total ton/h emission $F_e(X)$ of atmospheric pollutants such as sulphur oxides SO_x and nitrogen oxides NO_x caused by fossil-fueled thermal units can be expressed as:

$$F_e(X)=\sum_{i=1}^{n} 10^{-2}\left(\alpha_i+\beta_i X_i+\gamma_i X_i^2\right)+\xi_i e^{(\lambda_i X_i)} \tag{17}$$

where α_i, β_i, γ_i, ξ_i and λ_i are emission characteristics coefficients of the *i*th-generator.

For stable operation, real power output of each generator is restricted by lower and upper limits (side constraints) as follows:

$$X_i^{min} \le X_i \le X_i^{max}, \quad i = 1,\dots,n. \tag{18}$$

The total power generation must cover the total demand P_D and the real power loss in transmission lines P_{loss}. Hence,

$$\sum_{i=1}^{n} X_i = P_D + P_{loss} \tag{19}$$

Let $n=6$ generators, the cost coefficients and emission characteristics coefficients given in Tables 2 and 3 according to Abido [18].

Let $P_D + P_{loss}$ = 290 MW and consider that the equality constraint given by Eq. (19) is written as two unequality constraints (Eq.21).

The problem can be mathematically formulated as a non-linear constrained multi-objective optimization problem, according to the Eqs. (9) to (15), as follows:

$$\text{Minimize} \quad \Phi(X) = f(X) + r_p\, P(X) \tag{20}$$

Subject to:

$$\begin{cases} g_1(X) = \sum_{i=1}^{6} X_i - (P_D + P_{loss}) \le 0 \quad (MW) \\ g_2(X) = (P_D + P_{loss}) - \sum_{i=1}^{6} X_i \le 0 \quad (MW) \\ 10 \le X_i \le 120, \quad i = 1,...,6 \quad (MW) \end{cases} \tag{21}$$

where $f(X) = w_1 \dfrac{F_c(X)}{F_c^0} + w_2 \dfrac{F_e(X)}{F_e^0}$ (22)

and $P(X) = \{\max[0\,,\, g_1(X)]\}^2 + \{\max[0\,,\, g_2(X)]\}^2$ (23)

The constraints given by (21) are considered in the objective function by using a penalty function defined in Eq. (23). The ideal solution was calculated using the Eq. (13) and the best results obtained using all the DE-strategies were F_c^0 = 550.1248 \$/h and F_e^0 = 0.1952 ton/h.

Table 2 - Cost coefficients for 6 generators.

	G_1	G_2	G_3	G_4	G_5	G_6
a	10	10	20	10	20	10
b	200	150	180	100	180	150
c	100	120	40	60	40	100

Table 3 - Emission characteristics coefficients for 6 generators.

	G_1	G_2	G_3	G_4	G_5	G_6
α	4.091	2.543	4.258	5.426	4.258	6.131
β	- 5.554	- 6.047	- 5.094	- 3.550	- 5.094	- 5.555
γ	6.490	5.638	4.586	3.380	4.586	5.151
ξ	2×10^{-4}	5×10^{-4}	1×10^{-6}	2×10^{-3}	1×10^{-6}	1×10^{-5}
λ	2.857	3.333	8.000	2.000	8.000	6.667

Table 4 - Cost and emission coefficients (13 generators).

G	P_i^{min}	P_i^{max}	a	b	c	e	f
1	0	680	0.00028	8.10	550	300	0.035
2	0	360	0.00056	8.10	309	200	0.042
3	0	360	0.00056	8.10	307	150	0.042
4	60	180	0.00324	7.74	240	150	0.063
5	60	180	0.00324	7.74	240	150	0.063
6	60	180	0.00324	7.74	240	150	0.063
7	60	180	0.00324	7.74	240	150	0.063
8	60	180	0.00324	7.74	240	150	0.063
9	60	180	0.00324	7.74	240	150	0.063
10	40	120	0.00284	8.60	126	100	0.084
11	40	120	0.00284	8.60	126	100	0.084
12	55	120	0.00284	8.60	126	100	0.084
13	55	120	0.00284	8.60	126	100	0.084

Table 5. Environmental economic dispatch optimization results for Problem 1, when $w_1 = w_2 = 0.5$.

		$f(X)$	F_c (\$/h)	F_e (ton/h)	g_1 (MW)	g_2 (MW)
DE Strategy	1	1.0712	617.0532	0.1992	-0.0823	-0.0177
	2	1.0626	606.1748	0.1997	-0.0994	-0.0006
	3	1.1146	646.4563	0.2058	-0.0187	-0.0813
	4	1.0920	567.2400	0.2250	-0.2999	0.1999
	5	1.0621	609.1248	0.1985	-0.1002	0.0002
	6	1.0760	607.8273	0.2044	-0.0712	-0.0288
	7	1.0625	611.3541	0.1979	-0.1000	-0.00
	8	1.0831	613.0314	0.2053	-0.0737	-0.0263
	9	1.0823	607.0605	0.2071	-0.0963	-0.0037
	10	1.0626	607.2288	0.1994	-0.0995	-0.0005
GAOT		1.1174	643.2788	0.2118	-1.2095	1.1095

Table 6. Optimal power output (*MW*) of each generator for Problem 1, when $w_1 = w_2 = 0.5$.

		x_1	x_2	x_3	x_4	x_5	x_6
DE Strategy	1	25.43	40.89	68.71	52.56	46.45	47.74
	2	26.73	39.05	56.66	62.33	51.12	44.17
	3	37.39	41.97	90.34	45.40	30.24	42.79
	4	14.97	31.29	82.25	65.07	51.84	44.52
	5	31.60	41.22	53.68	60.61	48.71	44.16
	6	22.68	38.47	66.10	70.93	46.12	38.58
	7	32.44	40.10	47.36	57.93	54.50	47.67
	8	28.94	45.71	29.62	71.69	66.42	40.26
	9	10.88	48.95	56.58	64.07	66.44	35.45
	10	30.02	41.80	50.08	63.12	53.23	41.80
GAOT		58.00	44.23	23.98	82.02	47.78	24.78

It is a multicriterion optimization problem, so the best solution depends on designer's interest. In Case 1, the weighting coefficients were assumed equals ($w_1 = w_2 = 0.5$) that represents the same priority for the two competing objective functions: fuel cost and emission of atmospheric pollutants. Thus, the strategies 2, 5, 6, 9, and 10 are a good solution because represent a reasonable compromise solution between the two objective functions (Tables 5 and 6). In this case, the worst result was obtained by using the strategy 3.

Case 2: environmental economic dispatch considering the total fuel cost (13 generators)

The objective of this economic dispatch problem is to minimize the total fuel cost at thermal power plants subjected to the operating constraints of a power system. A cost function is obtained based on the ripple curve for more accurate modeling. This curve contains higher order nonlinearity and discontinuity due to the valve point effect and should be refined by a sine function. Therefore, the objective function can be modified as:

$$min f = F_c(P) + |e_i \sin(f_i(P_i^{mim} - P_i))| \qquad (24)$$

where e_i and f_i are constants of the valve point effect of the generators given in Table 4, and F_c is given in Eq.(16).

In the power balance criterion, an equality constraint must be satisfied, as shown in (19). The generated power should be the same as the total load demand plus total line losses. In this case, is imposed that $P_D + P_{loss} = 1800$ *MW*. The generating power of each generator should lie between maximum and minimum limits represented by (18).

Summarizing, the optimization problem is formulated as minimizing the objective function and constraints, represented in (24), (18), and (19).

The Tables 7 to 9 show the optimal results by using all the Differential Evolution strategies, Genetic Algorithms, and results obtained by Coelho and Mariani [13]. The worst result was obtained by GAs. The DE strategies 2 and 7 found similar results of [13].

In all the strategies the constraints were obeyed while in Genetic Algorithms the constraint g_2 was active.

The population of ED is very small, so the computational cost is lower. It is the main advantage of this methodology.

Table 7 - Best results (50 runs) of 13 generating units with the Valve Point and P_D=1800 *MW*.

		F_c (\$/h)	g_1 (MW)	g_2 (MW)	Time (s)
DE Strategy	1	18269.4740	-11.1351	-8.8649	0.0267
	2	**18153.9451**	-7.6019	-2.3980	0.0271
	3	18272.5078	-8.0942	-1.9057	0.0260
	4	18308.0523	-3.1728	-6.8272	0.0060
	5	18267.7663	-8.7871	-1.2128	0.0250
	6	18263.0732	-9.5152	-0.4848	0.0271
	7	**18125.6291**	-9.8908	-0.1092	0.0167
	8	18326.4323	-9.1687	-0.8313	0.0283
	9	18312.7824	-9.4686	-0.5313	0.0274
	10	18311.7688	-8.3747	-1.6252	0.0166
GAOT		19131.7068	-19133.0	19123.0852	0.0172
DEC(1)-SQP(1) [13]		17938.9521	-	-	0.50

Table 8 - Optimal power output (*MW*) of generators P_1 to P_6, for Problem 2.

		P_1	P_2	P_3	P_4	P_5	P_6
DE Strategy	1	637.2263	281.9363	147.2529	155.79410	60.0000	60.0000
	2	627.1577	360.0000	160.1155	160.1249	60.0000	60.0000
	3	627.8123	360.0000	0	60.0000	107.5155	152.7215
	4	455.7318	294.9726	360.0000	157.5792	60.0000	60.0000
	5	539.9624	296.0688	355.8903	63.9895	60.0000	60.0000
	6	447.0375	298.6173	255.7359	60.0000	156.2771	62.9757
	7	629.0954	0	309.5622	60.0000	60.0000	60.0000
	8	538.7812	252.0971	217.4743	60.0000	60.0000	111.9255
	9	538.8759	227.2284	144.3009	114.3049	118.0862	60.0000
	10	539.3783	0.3502	353.4283	62.6085	110.1435	161.8711
GAOT		586.0267	84.1941	148.6262	94.9364	113.6072	65.4708
DEC(1)-SQP(1) [13]		526.1823	252.1857	257.9200	78.2586	84.4892	89.6198

Table 9 - Optimal power output (*MW*) of generators P_7 to P_{13}, for Problem 2.

		P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}
DE Strategy	1	60.0000	60.0000	76.6329	110.0224	40.0000	55.0000	55.0000
	2	60.0000	60.0000	60.0000	40.0000	40.0000	55.0000	55.0000
	3	60.0000	109.4299	60.0000	40.0000	109.4265	55.0000	55.0000
	4	100.4930	60.0000	60.0000	40.0000	43.0506	55.0000	55.0000
	5	60.0000	60.0000	60.0000	55.6556	40.000	55.000	89.64612
	6	60.0000	164.3779	40.0000	43.0128	40.0000	92.4506	55.0000
	7	165.3751	160.3488	160.7276	40.0000	40.0000	55.0000	55.0000
	8	60.0000	60.0000	160.5766	44.9765	120.000	55.0000	55.0000
	9	156.0000	102.1931	106.1117	40.0000	43.7679	55.0000	89.6623
	10	109.2903	158.2069	60.0000	40.0000	42.9316	102.9889	55.4277
GAOT		156.3092	82.1289	129.3438	109.0419	82.3186	56.0643	95.5535
DEC(1)-SQP(1) [13]		88.0880	101.1571	132.0983	40.0007	40.0000	55.0000	55.0000

5 – CONCLUSIONS

This work presented a differential evolution algorithm theory as well as two applications in engineering well known in literature as Environmental Economic Dispatch. In Case 1 the aim is to select the generating unit outputs so as to meet the load demand at minimum operating cost and minimum pollution by atmospheric emission while satisfying all system constraints. In Case 2, the problem is to minimize the total fuel cost at thermal power plants subjected to the operating constraints of a power system with the valve point effect. All the Differential Evolution Strategies presented results better than Genetic Algorithms did. In Case 2, only the strategies 2 and 7 presented next results of [13]. The best result of ED was obtained by strategy ED/best/1/bin. In all the strategies the constraints were obeyed while in Genetic Algorithms the constraint g_2 was active. This technique can be represent a powerful tool in complex, multimodal optimization problems or when a lot of design variables are considered. It is strongly recommended that the user tests all the strategies and compare the obtained optimal results. This procedure is very simple, once all the strategies are available in the program and the operational cost is low.

REFERENCES

[1] VANDERPLAATS, G. N. "Numerical Optimization Techniques for Engineering Design". 3rd ed. Vanderplaats Research & Development, Inc., Colorado Springs, CO, 1999.

[2] GOLDBERG, D. E. "Genetic Algorithms in Search, Optimization, and Machine Learning". Reading, MA: Addison-Wesley, 1989.

[3] HAUPT, R. L.; HAUPT, S. E. "Practical Genetic Algorithms". Wiley-Interscience Publication, New York, 1998.

[4] SCHWEFEL, H.P. "Evolution and Optimum Seeking". John Wiley & Sons, 1995.

[5] STORN, R.; PRICE, K. "Differential Evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces". Technical Report TR-95-012, International Computer Science Institute, Berkeley, mar.1995.

[6] STORN, R. "System Design by Constraint Adaptation and Differential Evolution". IEEE Transactions on Evolutionary Computation, v. 3, n. 1, p. 22–34, 1999.

[7] CHENG, S. L.; HWANG, C. "Optimal Approximation of Linear Systems by a Differential Evolution Algorithm". IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, v. 31, n. 6, p. 698–707, 2001.

[8] BERGAMASCHI, P.R.; SARAMAGO, S.F.P.; COELHO, L.S. "Comparative Study of SQP and Metaheuristics for Robotic Manipulator Design", submitted to Applied Numerical Mathematics. 2006.

[9] NELDER J. A.; MEAD R. "A simplex method for function minimization". Computer Journal, v.7, p. 308–313, 1965.

[10] CHIOU, J-P."Variable scaling hybrid differential evolution for large-scale economic dispatch problems."Electric Power Systerms Research, v. 77, no. 3-4, p 212-218, 2007.

[11] PEREZ-GUERREIRO, R. E; CEDENO-MALDONADO, J. R. "Differential evolution based economic environmental power dispatch". Power Symposium. Proceedings of the 37th Annual North American. V. 23-25, p. 191-197, 2005.

[12] DONG, Z. Y., LU, M., LU, Z., WONG, K.P. "A Differential Evolution Based Method for Power System Planning". IEEE Congress on Evolutionary Computation. CEC 2006, p. 1-7, 2006.

[13] COELHO, L.d.S.; MARIANI, V.C. "Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect". IEEE Transactions on Power Systems, v. 21, no. 2, p. 989–996, 2006.

[14] MICHALEWICZ, Z. "Genetic Algorithms + Data Structures = Evolution Programs". 3.ed. New York: Springer-Verlag, 1996.

[15] STORN, R.; PRICE, K. "Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces". Journal of Global Optimization, v. 11, p. 341–359, 1997.

[16] OSYCZKA, A. "An Approach to Multicriterion Optimization for Structural Design, Proceedings of International Symposium on Optimum Structural Design", University of Arizona, 1981.

[17] HOUCK, C.R.; JOINEZ, J.A.; KAY, M.G. "A Genetic Algorithms for Function Optimization: a Matlab Implementation". NCSO-IE Technical Reported, 1995.

[18] ABIDO, M. A. "A niched Pareto genetic algorithm for multiobjective environmental/economic dispatch". International Journal of Electrical Power and Energy Systems, V. 25 - (2) , p. 97–105, 2003.