Design optimization of distribution transformers based on Differential Evolution Algorithms

Rasim Salkoski[1], Ivan Chorbev[2]

[1] University for Information Science and Technology, Building at ARM, 6000 Ohrid, R. of Macedonia
rasim.salkoski@uist.edu.mk
[2] Faculty of computer science and engineering, University of Ss Cyril and Methodius, Rugjer Boshkovich 16, P.O. Box 393, 1000 Skopje,R. of Macedonia
ivan.chorbev@finki.ukim.mk

Abstract. Genetic algorithms and their variants have been extensively used for solving combinatorial optimization problems. One area of great importance that can benefit from the effectiveness of such algorithms is electric energy distribution. Transformers deserve extensive treatment in the field of research and production, due to the fact that the electric energy undergoes several transformations on its way from generators to the consumers. In that regard, special interest is dedicated to the minimization of production and exploitation costs of a transformer. In the paper the combinatorial optimization algorithm based on Differential Evolution is described and applied to the problem of minimizing the cost of the active part of wound core distribution transformers. Constraints imposed both by international specifications and customer needs are taken into account. The Objective Function that is optimized is a minimization dependent on multiple input variables. Constraints are normalized and modeled as inequalities.

Keywords: Combinatorial optimization, Transformer design optimization methodology, Differential Evolution algorithm, Optimization methods, distribution transformer, Wound core type transformer.

1 Introduction

With the very rapid development of computers, transformer designers are freed from the cumbersome routine calculations. Within a matter of minutes or even seconds, computers can generate a number of different transformer designs (by changing current density, flux density, core dimensions, type of magnetic material and so on) and eventually come up with an optimum design. Because of the software design approach and the ease of making multiple iterations of the same design layout, it is easy to optimize the transformer to use a minimal set of expensive materials. There are several packages which are covering the branch of transformer optimization like: Non-linear optimization program (**TOPT)** , Transformer Tap Optimization software analysis module (**ETAP**), Transformer Design Optimization (**TDO**) software package for transformer design optimization and economic evaluation analysis and etc. The difficulty in resolving the optimum balance between the transformer cost and its performance is becoming even more complicated nowadays, as the main transformer`s materials (copper or aluminum for transformer windings and steel for magnetic circuit) are stock exchange commodities and their prices vary daily.

Techniques that include mathematical models containing analytical formulas, based on design constants and approximations for the calculation of the transformer parameters are often the base of the design process used by transformer manufacturers. Genetic algorithms and their variants have been extensively used for solving combinatorial optimization problems. One area of great importance that can benefit from the effectiveness of such algorithms is electric energy distribution. The work in this paper introduces the use of an evolutionary algorithm, named Differential Evolution (DE) in conjunction with the penalty function approach to minimize the transformer active part cost while meeting international standards and customer needs. A simple additive penalty function approach is used in order to convert the constrained problem into an unconstrained problem. Due to this conversion, the solution falling outside the feasible region is penalized and the solving process is guided to fall into the feasible solution space after a few generations. The method of penalty function approach is very sensitive when the penalty parameters are large. Penalty functions tend to be very sensitive near the boundary of the feasible domain and that result in a local optimal solution or an infeasible solution. It is always necessary to have careful selection of the penalty parameters for the proper convergence to a feasible optimal solution.

Moreover, the proposed method finds the global optimum transformer design by minimizing the active part cost while simultaneously satisfying all the constraints imposed by international standards and transformer user needs, instead of focusing on the optimization of only one parameter of transformer performance (e.g., no-load losses or short-circuit impedance). Using the proposed technique, a user-friendly DE computer program is developed that combines transformer design with analysis and optimization tools, useful for design optimization. The method is applied to the design of distribution transformers of several ratings and loss categories and the results are compared with a heuristic transformer design optimization methodology, resulting in significant cost savings.

2 Related work

In this paper the Penalty Function method is implemented to handle the constraint using the Differential Evolution (DE) algorithm. Other authors have proposed different approaches to solve constrained optimization with DE-based algorithms.

B.V.Babu and M. Mathew Leenus Jehan in [9] have applied Differential Evolution with a Penalty Function Method and Weighing Factor Method for finding a Pareto optimum set for the different problems. DE is found to be robust and faster in optimization. DE managed to give the exact optimum value within less generations compared to a simple Genetic Algorithm.

Mezura-Montes and Coello Coello in [12] present a Differential-Evolution based approach to solve constrained optimization problems. Three selection criteria based on feasibility are used to deal with the constraints of the problem and also a diversity mechanism is added to maintain infeasible solutions located in promising areas of the search space. The conventional DE algorithm highly depends on the chosen trial vector generation strategy and associated parameter values used. DE researchers have suggested many empirical guides for choosing trial vector generation.

Storn and Price [7] suggested that a reasonable value for NP should be between 5D and 10D, and a good initial choice of F was 0.5. The effective range of F values was suggested between 0.4 and 1. The first reasonable attempt of choosing CR value can be 0.1. However, because the large CR value can speed up convergence, the value of 0.9 for CR may also be a good initial choice if the problem is near unimodal or fast convergence is desired. Moreover, if the population converges prematurely, either F or NP can be increased.

Recently, Rönkkönen in [15] suggested using F values between [0.4,0.95] with 0.9 being a good initial choice. The CR values should lie in [0,0.2] when the function is separable while in [0.9,1] when the function's parameters are dependent. However, when solving a real engineering problem, the characteristics of the problem are usually unknown. Hence, it is difficult to choose the appropriate CR value in advance.

Zaharie proposed a parameter adaptation for DE (ADE) based on the idea of controlling the population diversity, and created a multipopulation approach [16]. Following the same ideas, Zaharie and Petcu designed an adaptive Pareto DE algorithm for multiobjective optimization and analyzed its parallel version [17].

The researchers have developed some techniques to avoid manual tuning of the control parameters. For example, Das et al. [18] linearly reduced the scaling factor F with increasing generation count from a maximum to a minimum value, or randomly varied F in the range (0.5,1). They also have employed a uniform distribution between 0.5 and 1.5 (with a mean value of 1) to obtain a new hybrid DE variant [19].

3 The Differential Evolution (DE) algorithm

Differential Evolution (DE) algorithm is a population-based stochastic method for global optimization developed by Rainer Storn and Kenneth Price [6],[7] for optimization problems over continuous domains. The original version of DE with constituents can be defined as follows ([8], [14]):

1) The population

$$P_{\mathbf{x},g} = \left(\mathbf{x}_{i,g}\right), \quad i = 0,1,.....,NP, \; g = 0,1,.....,g_{max}, \\ \mathbf{x}_{i,g} = \left(x_{j,i,g}\right), \; j = 0,1,.....,D-1. \tag{1}$$

where *NP* is the number of population vectors, g defines the generation counter, and *D* the number of parameters.

2) The initialization of the population through

$$x_{j,i,0} = rand_j\left[0,1\right)\cdot\left(b_{j,U} - b_{j,L}\right) + b_{j,L} \; . \tag{2}$$

The *D*-dimensional initialization vectors, $\mathbf{b}_L$ and $\mathbf{b}_U$ indicate the lower and upper bounds of the parameter vectors $\mathbf{x}_{i,j}$. The random number generator, $rand_j[0,1)$, returns a uniformly distributed random number from within the range [0,1), i.e., $0 \leq rand_j[0,1) < 1$. Indication that a new random value is generated for each parameter is denoted by the subscript j.

3) The perturbation of a base vector $\boldsymbol{y}_{i,g}$ by using a difference vector mutation

$$\boldsymbol{v}_{i,g} = \boldsymbol{y}_{i,g} + F \cdot \left(\boldsymbol{x}_{r1,g} - \boldsymbol{x}_{r2,g} \right). \tag{3}$$

to generate mutation vector $v_{i,g}$. The difference vector indices, r_1 and r_2, are randomly selected once per base vector. Setting $y_{i,g} = x_{r0,g}$ defines what is often called classic DE where the base vector is also a randomly chosen population vector. The random indexes r_0, r_1, and r_2 should be mutually exclusive.

4) Diversity enhancement

The classic variant of diversity enhancement is crossover which mixes parameters of the mutation vector **v**i,g and the so-called ***target vector x***i,g in order to generate the ***trial vector u***i,g. The most common form of crossover is uniform and is defined as

$$\boldsymbol{u}_{i,g} = u_{j,i,g} = \begin{cases} v_{j,i,g} & if \left(rand_j \left[0,1 \right) \leq CR \right) \\ x_{j,i,g} & otherwise \end{cases} \tag{4}$$

In order to prevent the case $\boldsymbol{u}_{i,g} = \boldsymbol{x}_{i,g}$ at least one component is taken from the mutation vector **v**i,g, a detail that is not expressed in Eq. (4).

5) Selection

DE uses simple one-to-one survivor selection where the trial vector $u_{i,g}$ competes against the target vector $x_{i,g}$. The vector with the lowest objective function value survives into the next generation $g + 1$.

$$\boldsymbol{x}_{i,g+1} = \begin{cases} \boldsymbol{u}_{i,g} & if\ f\left(\boldsymbol{u}_{i,g}\right) \leq f\left(\boldsymbol{x}_{i,g}\right) \\ \boldsymbol{x}_{i,g} & otherwise. \end{cases} \tag{5}$$

Along with the DE algorithm came a notation (5) to classify the various DE-variants. The notation is defined by DE/*x/y/z* where *x* denotes the base vector, *y* denotes the number of difference vectors used, and *z* representing the crossover method. For example, DE/rand/1/bin is the shorthand notation for Eq. (1) through Eq. (5) with $y_{i,g} = x_{r0,g}$. DE/best/1/bin is the same except for $y_{i,g} = x_{best,g}$. In this case $x_{best,g}$ represents the vector with the lowest objective function value evaluated so far. With today's extensions of DE the shorthand notation DE/*x/y/z* is not sufficient any more, but a more appropriate notation has not been defined yet.

Price and Storn [6] gave the working principle of DE with single strategy [7]. They suggested ten different strategies for DE. Different strategies can be adopted in the DE algorithm depending upon the type of problem to which DE is applied. The strategies can vary based on the vector to be perturbed, number of difference vectors considered for perturbation, and finally the type of crossover used. The following are the ten different working strategies: 1. DE/best/1/exp, 2. DE/rand/1/exp, 3. DE/rand-to-best/1/exp, 4. DE/best/2/exp, 5. DE/rand/2/exp, 6. DE/best/1/bin, 7. DE/rand/1/bin, 8. DE/rand-to-best/1/bin, 9. DE/best/2/bin, 10. DE/rand/2/bin.

As it is explained the general convention used above is DE/*x/y/z*. DE stands for Differential Evolution, *x* represents a string denoting the vector to be perturbed, *y* is the number of difference vectors considered for perturbation of *x*, and *z* stands for the type of crossover being used (exp: exponential; bin: binomial). Hence the perturbation can be either in the best vector of the previous generation or in any randomly chosen vector. Similarly for perturbation either single or two vector differences can be used. For perturbation with a single vector difference, out of the three distinct randomly chosen vectors, the weighted vector differential of any two vectors is added to the third one. In exponential crossover, the crossover is performed on the D variables in one loop until it is within the CR bound. The first time a randomly picked number between 0 and 1 goes beyond the CR value, no crossover is performed and the remaining D variables are left intact. In binomial crossover, the crossover is performed on each of the D variables whenever a randomly picked number between 0 and 1 is within the CR value. So for high values of CR, the exponential and binomial crossover methods yield similar results [1].

A strategy that works out to be the best for a given problem may not work well when applied to a different problem. Also, the strategy and the key parameters to be adopted for a problem are to be determined by trial and error. However, strategy-7 (DE/rand/1/bin) appears to be the most successful and the most widely used strategy. In all, three factors control evolution under DE, the population size NP, the weight applied to the random differential F and the crossover constant CR. More details regarding DE are available in [7], [8], [9] and [14].

4 Mathematical modeling of power objects and optimization

A mathematical description of a global constrained minimization problem requires us to apply an appropriate model which has limited number of parameters (design variables). Any kind of optimization problem can be formalized to find the appropriate set of design variables in the multidimensional parameter space, which can optimize the main objective function. In the mathematical notation the optimization problem can generally be represented as a pair (S, f) , where $S \subseteq R^n$ is a bounded set on R^n and $f : S \rightarrow R$ is an n-dimensional real-valued function. The problem is to find a point $\boldsymbol{x}_{min} \in S$ such that $f(\boldsymbol{x}_{min})$ is a global minimum on S . More specifically, it is required to find an $\boldsymbol{x}_{min} \in S$ such that

$$\forall \boldsymbol{x} \in S : f(\boldsymbol{x}_{min}) \leq f(\boldsymbol{x}) \quad (6)$$

$$g_i(\boldsymbol{x}) \leq 0,\ i = 1, 2, \bullet\bullet\bullet, q \quad (7)$$

$$h_j(\boldsymbol{x}) = 0,\ j = q + 1, \bullet\bullet\bullet, m \quad (8)$$

where $\boldsymbol{x}$ is the vector of unknown quantities $\boldsymbol{x} = [x_1, x_2, \bullet\bullet\bullet, x_n]^T$, $g_i(\boldsymbol{x})$ and $h_j(\boldsymbol{x})$ are the restriction constraints, which can be represented mathematically as equations and/or inequations, m and q are integer numbers [13]. Generally, for each variable x_i it satisfies a constrained boundary

$$l_i \leq x_i \leq u_i,\ i = 1, 2, \bullet\bullet\bullet, n \quad (9).$$

In order to find the global optimum design of a distribution transformer, DE in conjunction with the penalty function approach technique is used. The goal of the proposed optimization method is to find a set of integer variables linked to a set of continuous variables that minimize the objective function (active part cost) and meet the restrictions imposed on the transformer design. Under these definitions, a DE algorithm in conjunction with the penalty function approach is focused on the minimization of the cost of the transformer's active part:

$$\min_{x} \sum_{j=1}^{3} c_j \cdot f_j(\boldsymbol{x}) \tag{10}$$

where c_1 is the primary winding unit cost (€/kg), f_1 is the primary winding weight (kg), c_2 is the secondary winding unit cost (€/kg), f_2 is the secondary winding weight (kg), c_3 is the magnetic material unit cost (€/kg), f_3 is the magnetic material weight (kg), and $\boldsymbol{x}$ is the vector of the five design variables, namely the width of secondary winding (*a*), the diameter of core leg (*D*), the core window height (*b*), the current density of secondary winding (*g*) and the magnetic flux density (*B*).

The minimization of the cost of the transformer is subject to the constraints:

$S - S_N \leq 0$; $P_{CU} - P_{CUN} \leq 0$; $P_{FE} - P_{FEN} \leq 0$; $U_K - U_{KN} \leq 0$

where: S is designed transformer rating (kVA), S_N is transformer nominal rating (kVA), P_{FE} is designed no-load losses (W), P_{CU} is designed load losses (W), U_K is designed short-circuit impedance (%), P_{FEN} is guaranteed no-load losses (W), P_{CUN} is guaranteed load losses (W) and U_{KN} is guaranteed short-circuit impedance (%).

It should be noted that functions f_1, f_2, f_3, appearing in the objective function (10) are composite functions of the design variables $\boldsymbol{x}$, e.g., $f_1= f_1(g_1(h_1(\boldsymbol{x})))$ the transformer design optimization problem is a hard problem in terms of both modeling and solving.

The single objective Differential Evolution optimization algorithm with penalty function approach has been applied. The program has two input files, "Limits.txt" and "ParameterLimits.txt" and generates two output files, "ReportDE.html" and "Convergence.txt". Accordingly, the objective function for the model is:

$$f(x_2, x_3, x_5) = \left(3.9655 \cdot 10^4 \cdot x_5 + 2.40546 \cdot 10^5 \cdot x_3 + 2.987 \cdot 10^3\right) \cdot x_2^2 + 1.8924 \cdot x_2^3$$
$$+\left(6.96522 \cdot 10^5 \cdot x_2 + 1.42442 \cdot 10^6 \cdot x_3 + 1.3478 \cdot 10^4\right) \cdot x_3 \cdot x_5 \tag{11}$$

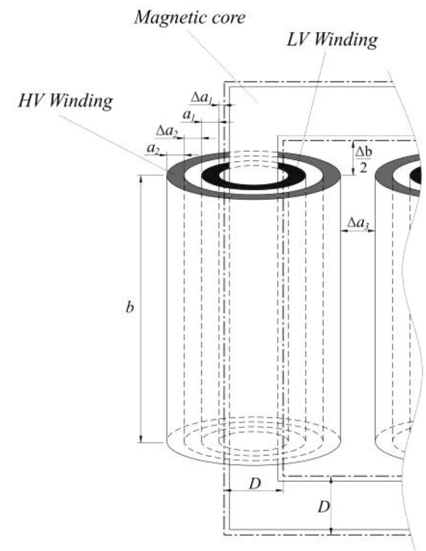

Fig.1 Active part of distribution transformer – main dimensions

The inequality constraints should be modified to the less or equal format, g(x) ≤ 0. If the problem is an unconstrained optimization problem, the user need not enter anything in the space specified for the constraints coding. The constraints of the analyzed mathematical model are entered as follows: Constraint 12 match to transformer nominal rating, Constraint 13 match to guaranteed load losses, Constraint 14 match to guaranteed no-load losses and Constraint 15 guaranteed short-circuit impedance. Constants in front of decision variables have been taken from the Fig.1 and reference [11].

$$317.82 \cdot x_1 \cdot x_2^2 \cdot x_3 \cdot x_4 \cdot x_5 \cdot 10^6 - 50 \cdot 10^3 \le 0 \quad (12)$$

$$\left(3.638 \cdot 10^{-7} \cdot x_2 + 8.113 \cdot 10^{-7} \cdot x_3 + 7.51 \cdot 10^{-9}\right) \cdot x_3 \cdot x_4^2 \cdot x_5 - 1050 \le 0 \quad (13)$$

$$\left(-0.4237 \cdot x_1^2 + 1.2712 \cdot x_1 - 0.0241\right) \cdot \left(\left(3.9655 \cdot 10^4 \cdot x_5 + 2.405 \cdot 10^5 \cdot x_3 + 2.987 \cdot 10^3\right) \cdot x_2^2 + 1.892 \cdot x_2^3\right) \cdot 0.4 - 190 \le 0 \quad (14)$$

$$\left(0.008 \cdot x_2 + 0.0186 \cdot x_2 \cdot x_3 + 0.032 \cdot x_3 + 1.7744 \cdot x_3^2 + 1.6 \cdot 10^{-4}\right) \cdot 317.82 \cdot 0.0186 \cdot x_3 \cdot x_4 / x_1 \cdot x_2^2 - 4.1 \le 0 \quad (15)$$

These values are multiplied by a penalty co-efficient (entered by the user of the program), which is then added to the objective function to continue the process of optimization. This process is often termed as penalty function approach.

4.1 Experimental results

After inserting the objective function and constraints, the user needs to prepare the two input files ("Limits.txt" and "ParameterLimits.txt"). In the "Limits.txt" input file, the lower and upper bound for each decision variable separated by a tab, is entered. The number of decision variables for the analyzed mathematical model is five.

Lower and Upper bound of decision variables in the "Limits.txt" input file for the analyzed mathematical model is as follows: 1.6 up to 1.8 for the magnetic flux density (*B*) in Tesla, .1 up to .125 for the diameter of core leg (*D*) in m, .015 up to .020 for the width of secondary winding (*a*) in m, 2.4 up to 3.0 for the current density of secondary winding (*g*) in A/mm^2 and .210 up to .230 for the core window height (*b*) in m.

The "ParameterLimits.txt" input file requires the number of decision variables (in the first row), maximum number of generations (in the second row), minimum and maximum number of population (*NP)*, crossover constant (*CR*), weighting factor (*F*) along with their step length for sensitivity analysis in the third, fourth and fifth rows respectively.

The input file for the analyzed mathematical model is as follows: Number of decision variables is 5, Maximum number of generations is 30, Minimum, maximum and step length for *NP* 20,20,10, Minimum, maximum and step length for *CR* 0.8, 0.9, 0.1 and Minimum, maximum and step length for *F* 0.5, 0.6, 0.1 .

This completes the preparation of inputs files. To generate the optimal set of solutions for the optimization problem, the DE program is compiled and executed. The program can be compiled and execution using any standard C compiler for the Windows environment.

When the program is run for different combinations of *NP, CR* and *F*, the optimal set of parameters is determined based on two factors i.e., minimum objective function value and lower CPU time requirement. In any given situation, if minimum objective function values are the same for any given combination(s), the next criteria that is chosen for selecting optimal combination is lower CPU time requirement. In this program, these two factors are considered for choosing optimal set of parameters. The output figures in Table 1 are given for the analyzed mathematical model generated after the successful completion of the program's execution [1]. The optimal value of objective function and decision variables for the optimization problem is also recorded.

Table 1. Output figures with time to complete strategies on each of them

Strat. No.	Strategy	NP	CR	F	Optimal Value	Constraint Violation	NFE	Time Taken(ms)
1	DE/rand/1/bin	20	0.80	0.50	4.976027E+002	0.0000E+000	835	30
2	DE/best/1/bin	20	0.80	0.50	4.973189E+002	0.0000E+000	1429	40
3	DE/best/2/bin	20	0.80	0.50	4.973945E+002	0.0000E+000	681	11
4	DE/rand/2/bin	20	0.80	0.60	4.976027E+002	0.0000E+000	791	20
5	DE/rand-to-best/1/bin	20	0.80	0.50	4.973220E+002	0.0000E+000	1869	40
6	DE/rand/1/exp	20	0.80	0.50	4.975862E+002	0.0000E+000	989	20
7	DE/best/1/exp	20	0.80	0.60	4.973355E+002	0.0000E+000	747	20
8	DE/best/2/exp	20	0.90	0.50	4.975857E+002	0.0000E+000	769	10
9	DE/rand/2/exp	20	0.90	0.50	4.976027E+002	0.0000E+000	703	20
10	DE/rand-to-best/1/exp	20	0.90	0.60	4.973240E+002	0.0000E+000	1319	30

Best Strategy is Strategy DE/rand/2/bin, Minimum constraint violation (CV) : 0.0000E+000, Minimum objective value with min CV: 4.976027E+002 and Minimum time taken : 20

Table 2. Output table results of the analyzed mathematical model

Parameter	Value
X_1	1.630365
X_2	0.100062
X_3	0.015006
X_4	2.968991
X_5	0.210373

The parameters X_1, X_2, X_3, X_4, X_5 match respectively to the magnetic flux density (*B*), the diameter of core leg (*D*), the width of secondary winding (*a*), the current density of secondary winding (*g*) and the core window height (*b*).

Table 3. Comparative results of the analyzed mathematical model with produced object by the specified optimized method

	The mag. flux density (T)	LV wind. Curr. density (A/mm^2)	Diam. of Core Leg (mm)	Width of LV wind. (mm)	The Core window (mm)	The cost of the transf. active part
DE Algorithm with penalty function approach	1.630	2.969	100	15	210	497
Lagrange multipliers with Newton Raphson approach [11]	1.642	2.388	117	20	221	703

5 Conclusion

This paper presents an efficient implementation of a Single Objective Optimization Program using the Differential Evolution algorithm with a penalty function approach, applied to a power object. Our penalty function approach integrates established techniques in existing EA's in a single unique algorithm. Our approach was tested on a three phase distribution transformer and the results indicate that this approach can be used to solve a range of SOOs with linear/nonlinear equality/inequality constraints, as well as continuous/discontinuous search spaces. Moreover, this approach is easy to implement and its computational cost is relatively low.

The use of the DE computer program is applied to the analyzed mathematical model. In the first methodology (Table 3) the single objective DE optimization showed that single optimum could be obtained fast even when constraints in the penalty function method are complex and compared with the second methodology in the same table, the cost materials for the active part of the reviewed object are lower.

References

1 Arunachalam, Vasan,: Optimization Using Differential Evolution. Water Resources Research Report. Book 22. Department of Civil and Environmental Engineering, The University of Western Ontario, Publication Date 7-2008.

2. A. Zamuda, J. Brest, B. Bošković, V. Žumer.: Differential Evolution with Self-adaptation and Local Search for Constrained Multiobjective Optimization. IEEE Congress on Evolutionary Computation (CEC), pp. 195-202 (2009)

3. J. Brest, A. Zamuda, B. Bošković, V. Žumer.: Dynamic Optimization using Self-Adaptive Differential Evolution. IEEE Congress on Evolutionary Computation (CEC) 2009, pp. 415-422. Trondheim, Norveška (2009).
4. Pang-Kai Liu, Feng-Sheng Wang.: Hybrid differential evolution including geometric mean mutation for optimization of biochemical systems, Journal of the Taiwan Institute of Chemical Engineers 41 (2010) 65–72, Department of Chemical Engineering, National Chung Cheng University, Chia-yi 62102, Taiwan (2010).
5. Differential Evolution Homepage, http://www.icsi.berkeley.edu/~storn/code.html
6. Onwubolu, G. C., and Babu, B. V.: New Optimization Techniques in Engineering,Springer-Verlag, Germany (2004).
7. Price V. Kenneth., Storn M. Rainer.: Differential evolution - A simple evolution strategy for fast optimization. Dr. Dobb's Journal, 22, 18-24 and 78. (1997).
8. Price, V. Kenneth., Storn, M. Rainer., and Lampinen, A. Jouni.: Differential evolution: A practical approach to global optimization. Springer-Verlag Berlin,Heidelberg (2005).
9. B.V.Babu, M. Mathew Leenus Jehan .: Differential Evolution for Multi-ObjectiveOptimization. Chemical Enginering Department B.I.T.S. Pilani, India (2005).
10. Elefterios I. Amoiralis, Pavlos S. Georgilakis, Marina A. Tsili .: Design optimization of distribution transformers based on mixed integer programming methodology. Technical University of Athens, Greece (2008).
11. Rasim Salkoski.: Selection of an optimal variant of 3-phase transformers with round and rectangular section of the magnetic core from aspect of minimum production costs. Master Thesis, Electrotechnical University in Skopje (2000).
12. Mezura-Montes.: E. Laboratorio NI Avanzada, Rébsamen 80, Centro, Xalapa, Veracruz 91090, Mexico, Velazquez-Reyes, J., Coello Coello, C.A.: Modified Differential Evolution for Constrained Optimization , pp 25 – 32, Conference Publications, Evolutionary Computation, CEC 2006 (2006).
13. Wenyin Gong, Changmin Chen, Zhihua Cai.: Simple Diversity Rules and Improved Differential Evolution for Constrained Global Optimization School of Computer Science China University of Geosciences, Wuhan 430074, P. R. China (2006).
14. Uday K. Chakraborty (Ed.).: Advances in Differential Evolution, Mathematics & Computer Science Department, University of Missouri, St.Louis, USA, Springer-Verlag Berlin Heidelberg (2008).
15. J. Rönkkönen, S., Kukkonen, and K. V. Price.: Real-parameter optimization with differential evolution. Proc. IEEE Congr. Evolut.Comput., Sep. 2005, pp. 506–513, Edinburgh, Scotland (2005).
16. D. Zaharie.: Control of population diversity and adaptation in differential evolution algorithms. Proc. Mendel 9th Int. Conf. Soft Comput., R. Matousek and P. Osmera, Eds., Brno, Czech Republic, pp. 41–46, Brno, Czech Republic (2003)
17. U. K. Chakraborty, S. Das, A. Konar.: Differential evolution with local neighborhood. Proc. Congr. Evolut. Comput., pp. 2042-2049,Vancouver, BC,Canada (2006).