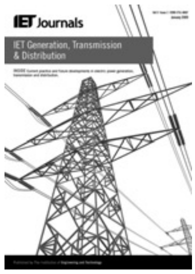
Published in IET Generation, Transmission & Distribution
Received on 8th October 2008
Revised on 7th July 2009
doi: 10.1049/iet-gtd.2009.0184



ISSN 1751-8687

Differential evolution solution to transformer no-load loss reduction problem

P.S. Georgilakis
School of Electrical and Computer Engineering, National Technical University of Athens, GR 15780, Athens, Greece
E-mail: pgeorgil@central.ntua.gr

Abstract: After the completion of core manufacturing and before the assembly of transformer active part, 2N small individual cores and 2N large individual cores are available and have to be optimally combined into N transformers so as to minimise the total no-load loss (NLL) of N transformers. This complex combinatorial optimisation problem is called transformer no-load loss reduction (TNLLR) problem. A new approach combining differential evolution (DE) and multilayer perceptrons (MLPs) to solve TNLLR problem is proposed. MLPs are used to predict NLL of wound core distribution transformers. An improved differential evolution (IDE) method is proposed for the solution of TNLLR problem. The modifications of IDE in comparison to the simple DE method are (i) the scaling factor F is varied randomly within some range, (ii) an auxiliary set is employed to enhance the population diversity, (iii) the newly generated trial vector is compared with the nearest parent and (iv) the simple feasibility rule is used to treat the constraints. Application results show that the performance of the proposed method is better than that of two other methods, that is, conventional grouping process and genetic algorithm. Moreover, the proposed method provides 7.3% reduction in the cost of transformer main materials.

Nomenclature

CGP	conventional grouping process
DE	differential evolution
GA	genetic algorithm
IDE	improved differential evolution
MAPE	mean absolute percentage error
MLP	multilayer perceptron
NLL	no-load loss
TNLLR	transformer no-load loss reduction

1 Introduction

Construction of distribution transformers of high quality at a minimum possible cost is crucial for any transformer manufacturing industry to face market competition. A critical measure of transformer quality is transformer no-load loss (NLL). The less the transformer NLL is, the higher the transformer quality and efficiency become. The transformer designer can reduce transformer NLL by using appropriate magnetic materials (e.g. Hi-B or amorphous instead of typical M3 grade of magnetic material) or reducing core flux density or flux path length [1]. Transformer actual (measured) NLL deviates from the designed NLL because of the variability in production process [2]. Reduction of transformer actual NLL is a very important task for any manufacturing industry, since (i) it helps the manufacturer not to pay NLL penalties and (ii) it reduces the material cost (since smaller NLL design margin is used).

Electric utilities should use more generating capacity to produce additional electrical energy so as to compensate for transformer losses. In addition, transformer NLL appears 24 h per day, everyday, for a continuously energised transformer. Thus, it is in general preferable to design a transformer at minimum NLL [3].

After the completion of core manufacturing and before the assembly of transformer active part, 2N small individual cores and 2N large individual cores are available and have to be optimally combined into N transformers so as to minimise the total NLL of N transformers. This complex combinatorial optimisation problem is called transformer no-load loss reduction (TNLLR) problem.

The current industrial practice to solve the TNLLR problem is to pre-measure and assign a grade (quality category) to each individual core and then combine higher and lower graded individual cores to achieve an 'average' value for the entire transformer [1]. This is referred to as conventional grouping process (CGP).

Differential evolution (DE) is a relatively new evolutionary optimisation algorithm [4, 5]. DE has been successfully applied for the solution of difficult power system problems [6–16], but has not been applied to TNLLR so far. A new approach combining DE and multilayer perceptrons (MLPs) to solve TNLLR problem is proposed in this paper. MLPs are used to predict NLL of wound core distribution transformers. An improved differential evolution (IDE) method is proposed for the solution of TNLLR problem.

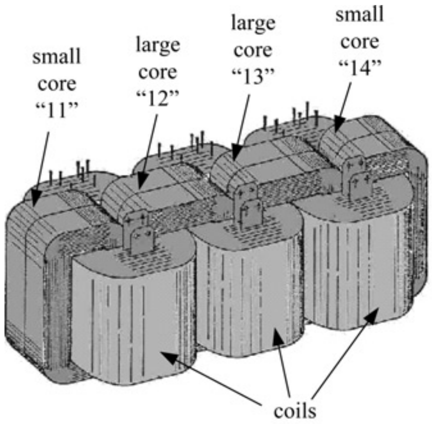
Many studies demonstrated that DE converges fast and is robust, simple in implementation and use, and requires only a few control parameters. In spite of the prominent merits, sometimes DE shows the premature convergence and slowing down of convergence as the region of global optimum is approached. In this paper, to remedy these defects, some modifications are made to the simple DE. An auxiliary set is employed to increase the diversity of population and prevent the premature convergence. In the simple DE, the trial vector, or offspring, is compared with the target vector with the same running index, whereas in this paper, the trial vector is compared with the nearest parent in the sense of Euclidean distance. Moreover, the comparison scheme is changed according to the convergence characteristics. The scaling factor F, which is constant in the original DE, is varied randomly within some specified range. The above modifications form an IDE algorithm that is applied for the solution of TNLLR problem. The proposed IDE algorithm is extensively tested on a transformer manufacturing industry and the results of the proposed IDE are compared with the results of two other methods.

2 Formulation of TNLLR problem

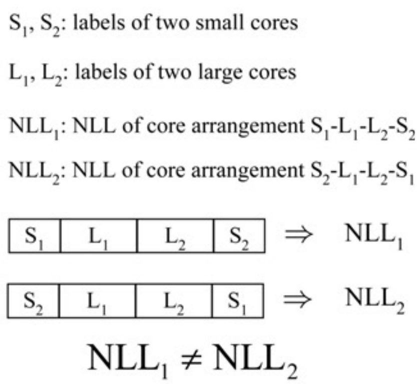
The wound core shell-type distribution transformer is composed of two small individual cores and two large individual cores as shown in Fig. 1*a*. We denote with '11' the left small individual core, with '12' the left large individual core, and with '13' and '14' the other two individual cores, so the arrangement of individual cores from left to right is '11'-'12'-'13'-'14', as Fig. 1*a* shows. Industrial experiments have shown that if the position of one core within the transformer changes, then the transformer NLL also changes [1], for example, the transformer with core arrangement $S_1 - L_1 - L_2 - S_2$ has different transformer NLL in comparison with the transformer with core arrangement $S_2 - L_1 - L_2 - S_1$ (Fig. 1*b*). The small cores S_1 and S_2 theoretically have the same technical characteristics (e.g. individual core NLL); however, in practice their characteristics are different because of the variability in production process [2], so the above two mentioned core arrangements have different transformer NLL because of the non-homogeneous electromagnetic field of the individual cores [1].

Transformers are usually produced in production batches so as to minimise labour cost [1]. In this paper, the production batch is defined as the production of N theoretically identical transformers with exactly the same technical characteristics that have been computed by one and the same transformer design.

N three-phase transformers are constructed from $2N$ small individual cores and $2N$ large individual cores. Let us denote



a



b

Figure 1 *Wound core distribution transformer*

a Assembled active part
b Impact of core position on transformer NLL [1]

as $V_s(V_l)$ the set of all $2N$ small (large) cores. A transformer is represented by a vector $\boldsymbol{t}_i$, the elements of which correspond to the four individual cores that assemble the transformer

$$\boldsymbol{t}_i = \begin{bmatrix} s_i^l & l_i^l & l_i^r & s_i^r \end{bmatrix}^T \tag{1}$$

Variables s_i^l, $s_i^r \in V_s$ represent the left and right small cores of transformer $\boldsymbol{t}_i$, while l_i^l, $l_i^r \in V_l$ the left and right large cores, respectively. Since only one core (small or large) can be assigned to one transformer and one position (left or right), the following restrictions are held

$$s_i^l \neq s_i^r, \quad l_i^l \neq l_i^r \tag{2}$$

$$s_k^{\{l,r\}} \neq s_i^{\{l,r\}}, \quad l_k^{\{l,r\}} \neq l_i^{\{l,r\}} \quad \text{with } k \neq i \tag{3}$$

where $s_i^{\{l,r\}}(l_i^{\{l,r\}})$ indicates the small (large) core in the left or right position for transformer $\boldsymbol{t}_i$.

Let us denote as $\boldsymbol{c}$ a vector containing one possible combination of N three-phase transformers $\boldsymbol{t}_i$, $i = 1, 2, \ldots, N$, that can be constructed by $2N$ small individual cores and $2N$ large individual cores

$$\boldsymbol{c} = \begin{bmatrix} \boldsymbol{t}_1^T & \boldsymbol{t}_2^T & \ldots & \boldsymbol{t}_N^T \end{bmatrix}^T \tag{4}$$

where T indicates the transpose of a vector.

Vector $\boldsymbol{c}$ is of $4N \times 1$ dimensions since each transformer $\boldsymbol{t}_i$ is represented by a 4×1 vector as (1) indicates. A specific arrangement (combination) of all small and large cores, for constructing the N three-phase transformers, corresponds to a given value of vector $\boldsymbol{c}$. Therefore any reordering of the elements of vector $\boldsymbol{c}$ results in different arrangement of individual cores, that is, different three-phase transformers. Fig. 2 presents an example of vector $\boldsymbol{c}$ in case that six small and six large cores are available. In particular, the serial numbers from 1 to 6 correspond to small cores, whereas the numbers from 7 to 12 to large cores. A randomly selected arrangement of these cores is also presented in Fig. 2 for constructing three different transformers. For example, the first transformer consists of the small cores with labels 5 and 1 and the large cores with labels 10 and 12. This is represented by the vector $\begin{bmatrix} 5 & 10 & 12 & 1 \end{bmatrix}^T$ in accordance with (1). Then, vector $\boldsymbol{c}$ is constructed by concatenating the vectors of the three transformers. The core arrangement for the other two transformers is generated accordingly and depicted in Fig. 2.

It is clear that the estimation of N transformers with optimal quality (minimum total NLL) is equivalent to the estimation of vector $\boldsymbol{c}$, which minimises the following

$$\boldsymbol{c}_{opt} = \arg \min_{\boldsymbol{c}} \left\{ \sum_{i=1}^{N} \text{NLL}_{\boldsymbol{t}_i} \right\} \tag{5}$$

where $\text{NLL}_{\boldsymbol{t}_i}$ is the estimated NLL of transformer $\boldsymbol{t}_i$ and $\boldsymbol{c}_{opt}$ is a vector that contains the optimal arrangement of all available small and large cores so that the estimated total NLL over all N transformers is minimised.

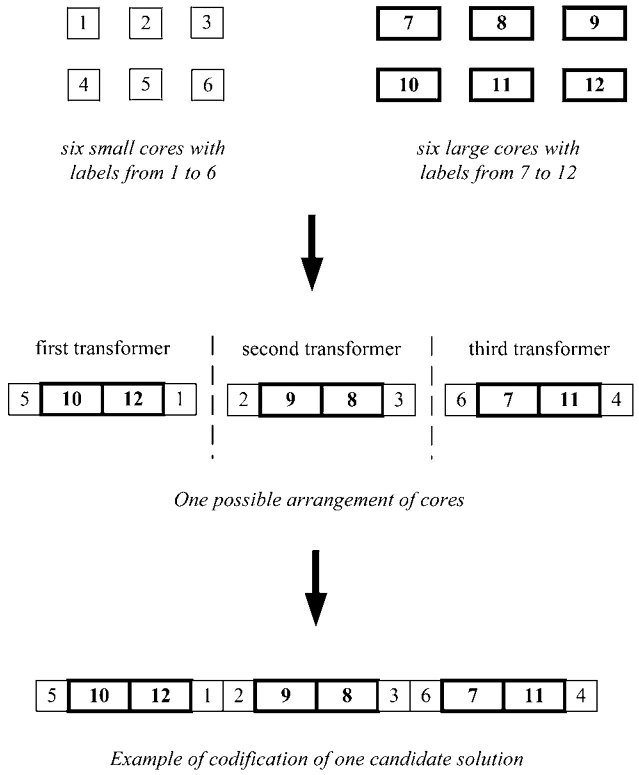


Figure 2 *Example of codification of one candidate solution*

The estimated NLL, $\text{NLL}_{\boldsymbol{t}_i}$, of each transformer $\boldsymbol{t}_i$ is computed as follows

$$\text{NLL}_{\boldsymbol{t}_i} = w_{\boldsymbol{t}_i} \cdot \text{SNLL}_{\boldsymbol{t}_i}, \quad \forall \boldsymbol{t}_i, \; i = 1, 2, \ldots, N \tag{6}$$

where $w_{\boldsymbol{t}_i}$ is the actual (measured) weight of the four individual cores of transformer $\boldsymbol{t}_i$ and $\text{SNLL}_{\boldsymbol{t}_i}$ is the specific NLL (W/kg) of transformer $\boldsymbol{t}_i$ that is estimated by the MLP architecture of Section 3.1.

The estimated NLL of each transformer $\boldsymbol{t}_i$ must be smaller than a maximum NLL, NLL_{max}

$$\text{NLL}_{\boldsymbol{t}_i} < \text{NLL}_{max}, \quad \forall \boldsymbol{t}_i, \; i = 1, 2, \ldots, N \tag{7}$$

It should be noted that the transformer manufacturer pays NLL penalties for each transformer $\boldsymbol{t}_i$ that violates (7).

In brief, the TNLLR problem is mathematically formulated as follows: minimise the objective function (5) subject to the constraints (2), (3) and (7).

As observed from (5), the estimation of the optimal core arrangement results in a combinatorial optimisation problem. For a typical number of small/large cores, direct minimisation of (5) is practically infeasible since the computational complexity for an exhaustive search is very

doi: 10.1049/iet-gtd.2009.0184

high. For example, for a typical production batch of 50 transformers (i.e. when 100 small and 100 large individual cores are available), 5.35×10^{22} combinations of core arrangements should be considered [1]. Moreover, quality control of individual cores can only check if the individual cores are of acceptable quality or not, so quality control is impossible to solve TNLLR problem. Because of the above reasons, this paper introduces IDE-MLP technique for the solution of TNLLR problem.

3 Overview of proposed methodology

3.1 NLL prediction with MLPs

The transformer NLL is affected by magnetic material parameters (e.g. thickness, type and hardness of magnetic material), design parameters (e.g. rated magnetic induction, dimensions of small and large individual cores) and production parameters (e.g. mechanical stresses during the formation of core, parameters of the annealing process, actual weight of individual cores, actual NLL of individual cores) [1, 17–24]. Unfortunately, there is no analytical expression for NLL estimation that takes into account the above qualitative and quantitative parameters. Moreover, transformer NLL prediction is highly non-linear [1, 23–25].

Artificial neural networks, because of their highly non-linear capabilities and universal approximation properties, have been proven very effective for NLL prediction [1]. After enough experimentation, it was found that MLPs with eight neurons in the input layer, one hidden layer and one neuron in the output layer (i.e. transformer-specific NLL) effectively solve the NLL prediction problem [1, 17]. Moreover, it was found that the sigmoid activation function provided the best results [1].

Table 1 shows the eight input parameters for NLL prediction [1]. The three out of eight input parameters and more specifically the parameters I_6, I_7 and I_8 reflect the importance of core arrangement on NLL [1]. In Table 1, the attribute I_5 represents the ratio of actual over designed total NLL of the four individual cores. The attribute I_4 represents the ratio of actual over designed total weight of the four individual cores. The attribute I_2 represents the average specific NLL of the magnetic material of the four individual cores, where

Table 1 Input parameters (attributes) for NLL prediction

Attribute	Description
I_1	Rated magnetic induction
I_2	$\dfrac{s_{15\,000}^{\text{material, '11'}} + s_{15\,000}^{\text{material, '12'}} + s_{15\,000}^{\text{material, '13'}} + s_{15\,000}^{\text{material, '14'}}}{4}$
I_3	$\dfrac{s_{17\,000}^{\text{material, '11'}} + s_{17\,000}^{\text{material, '12'}} + s_{17\,000}^{\text{material, '13'}} + s_{17\,000}^{\text{material, '14'}}}{4}$
I_4	$\dfrac{w_{\text{actual}}^{\text{core, '11'}} + w_{\text{actual}}^{\text{core, '12'}} + w_{\text{actual}}^{\text{core, '13'}} + w_{\text{actual}}^{\text{core, '14'}}}{w_{\text{designed}}^{\text{core, '11'}} + w_{\text{designed}}^{\text{core, '12'}} + w_{\text{designed}}^{\text{core, '13'}} + w_{\text{designed}}^{\text{core, '14'}}}$
I_5	$\dfrac{s_{\text{actual}}^{\text{core, '11'}} \cdot w_{\text{actual}}^{\text{core, '11'}} + s_{\text{actual}}^{\text{core, '12'}} \cdot w_{\text{actual}}^{\text{core, '12'}} + s_{\text{actual}}^{\text{core, '13'}} \cdot w_{\text{actual}}^{\text{core, '13'}} + s_{\text{actual}}^{\text{core, '14'}} \cdot w_{\text{actual}}^{\text{core, '14'}}}{s_{\text{designed}}^{\text{core, '11'}} \cdot w_{\text{designed}}^{\text{core, '11'}} + s_{\text{designed}}^{\text{core, '12'}} \cdot w_{\text{designed}}^{\text{core, '12'}} + s_{\text{designed}}^{\text{core, '13'}} \cdot w_{\text{designed}}^{\text{core, '13'}} + s_{\text{designed}}^{\text{core, '14'}} \cdot w_{\text{designed}}^{\text{core, '14'}}}$
I_6	$\dfrac{s_{\text{actual}}^{\text{core, '11'}} + s_{\text{actual}}^{\text{core, '12'}}}{s_{\text{designed}}^{\text{core, '11'}} + s_{\text{designed}}^{\text{core, '12'}}}$
I_7	$\dfrac{s_{\text{actual}}^{\text{core, '12'}} + s_{\text{actual}}^{\text{core, '13'}}}{s_{\text{designed}}^{\text{core, '12'}} + s_{\text{designed}}^{\text{core, '13'}}}$
I_8	$\dfrac{s_{\text{actual}}^{\text{core, '13'}} + s_{\text{actual}}^{\text{core, '14'}}}{s_{\text{designed}}^{\text{core, '13'}} + s_{\text{designed}}^{\text{core, '14'}}}$

$s_{15\,000}^{\text{material, `11'}}$ denotes the specific NLL (W/kg) at 15 000 Gauss of the magnetic material of the individual core that is placed at position '11' shown in Fig. 1*a*. Table 1 uses some more variables. In particular, the parameter $s_{17\,000}^{\text{material, `11'}}$ denotes the specific NLL at 17 000 Gauss of the magnetic material of the individual core that is placed at position '11'. The variable $s_{\text{actual}}^{\text{core, `11'}}$ represents the actual specific NLL of the individual core at place '11', that is, the ratio of actual NLL of the individual core at place '11' over its actual weight. It should be noted that $s_{\text{actual}}^{\text{core, `11'}}$ refers to the manufactured individual core, whereas $s_{15\,000}^{\text{material, `11'}}$ refers to the magnetic material of the individual core. Moreover, for an individual core operated at 15 000 Gauss, $s_{\text{actual}}^{\text{core, `11'}} > s_{15\,000}^{\text{material, `11'}}$, because of the effect of the manufacturing process on the NLL of the individual core [1]. The variable $s_{\text{designed}}^{\text{core, `11'}}$ represents the designed specific NLL of the individual core at place '11', which is calculated from the NLL curve of the individual core [1]. The parameter $w_{\text{actual}}^{\text{core, `11'}}$ represents the actual weight of the individual core at place '11', whereas the variable $w_{\text{designed}}^{\text{core, `11'}}$ represents the designed weight of the individual core at place '11'.

Table 1 shows that the following data have to be recorded for each individual core:

1. The specific NLL of the magnetic material of the individual core at 15 000 and at 17 000 Gauss.

2. The actual weight of the individual core.

3. The actual NLL of the individual core.

It should be noted that for the same production batch, all individual cores have the same rated magnetic induction and the same designed specific NLL. Moreover, all small individual cores have the same designed weight. Similarly, all large individual cores have the same designed weight.

It was found that the best results are obtained if each of the MLPs corresponds to a different environment (i.e. to a certain supplier, grade and thickness of magnetic material) [1, 17]. The MLP performance is evaluated by the mean absolute percentage error (MAPE), which is defined as follows

$$\text{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\frac{|S_i - \hat{S}_i|}{S_i} \times 100\% \qquad (8)$$

where S_i is the actual (measured) specific NLL of transformer i, $\hat{S}_i$ is the specific NLL of transformer i that is predicted by the MLP and N is the number of samples in the considered set (training set or test set).

3.2 NLL reduction with IDE-MLP

Table 2 presents a summary of the main steps of IDE-MLP solution to TNLLR problem.

Table 2 Summary of the main steps of IDE-MLP solution to TNLLR problem

Step 1	Based on customer requirements and several techno-economical criteria, design the transformers of a specific production batch [1]. From the transformer design, the environment type (i.e. supplier, thickness and grade of magnetic material) is defined
Step 2	Based on transformer design, construct the small and large individual cores and measure all necessary parameters (i.e. the actual NLL and weight) so that the eight attributes of Table 1 for a specific core arrangement can be calculated
Step 3	Solve the TNLLR problem of Section 2 using IDE (Section 4)
Step 4	Assemble the transformers using the optimal core arrangement $\boldsymbol{c}_{\text{opt}}$ that has been computed by the IDE of Step 3
Step 5	Measure the actual NLL for all constructed transformers of the production batch. Then, compare them with the predicted NLL, which are provided by the MLP
Step 6	In case of large deviation between the measured and the predicted NLL, retrain the MLP [1]. Then, store the new estimated weights in the MLP database to be used for the following production batches. Otherwise, retain the same MLP weights

4 IDE methodology for TNLLR

4.1 Overview

DE is a genetic algorithm (GA)-like optimisation algorithm but it differs from GA with respect to the mechanisms of reproduction and selection.

The proposed IDE methodology for the solution of TNLLR problem is composed of the following steps:

Step 1: Generation counter G is set to zero. Next, initialisation takes place (Section 4.2).

Step 2: Evaluation of candidate solutions (Section 4.9).

Step 3: Generation counter G is increased by one.

Step 4: Mutation with randomly varied scaling factor F (Section 4.3).

doi: 10.1049/iet-gtd.2009.0184

Step 5: Crossover (Section 4.4).

Step 6: Evaluation of candidate solutions (Section 4.9).

Step 7: Selection (Section 4.5) with the use of the auxiliary set concept (Section 4.6).

Step 8: If the termination criterion (maximum number of generations) is not met, then go to Step 3, else IDE terminates.

4.2 Initialisation

An initial population of NP candidate solutions is randomly generated and is used as the parent population of the first iteration or generation. More specifically, the population is initialised by randomly generated individuals within the boundary constraints

$$x_{j,i}^{0} = \text{rand}_{j,i}[0,\ 1](x_j^{(\text{U})} - x_j^{(\text{L})}) + x_j^{(\text{L})} \tag{9}$$

where $i = 1, 2, \ldots, \text{NP}, j = 1, 2, \ldots, D$, D is the variable dimension, $x_j^{(\text{L})}$ and $x_j^{(\text{U})}$ are the lower and upper boundary of the j component, respectively, and $\text{rand}_{j,i}[0,\ 1]$ denotes a uniformly distributed random value in the range [0, 1]. In the initial population, the solution of CGP is also included.

4.3 Mutation

For each target vector, or parent vector $\boldsymbol{x}_i^G$, a mutant vector is generated according to

$$\boldsymbol{v}_i^{G+1} = \boldsymbol{x}_{n1}^G + F(\boldsymbol{x}_{n2}^G - \boldsymbol{x}_{n3}^G) \tag{10}$$

where random indexes $n1$, $n2$ and $n3$ are integers, mutually different, and also chosen to be different from the running index i. In the initial DE scheme [4], the parameter F is a real and constant factor during the entire optimisation process, whose range is $F \in (0,\ 2]$. However, no optimal choice of F has been proposed in the bibliography of DE. All the studies used an empirically derived value, and in most cases, F varies from 0.4 to 1. This means F is strongly problem-dependent and the user should choose F carefully after some trial and error tests. In this paper, F is varied randomly within some specified range, as follows

$$F = a + b \cdot \text{rand}_i[0,\ 1] \tag{11}$$

where a and b are positive and real-valued constants, the sum of a and b is less than 1, and $\text{rand}_i[0,\ 1]$ denotes a uniformly distributed random value in the range [0, 1].

Consequently, F is different for each generation, and the computation of F by (11) is effective when the optimal value of F is difficult to be determined for complicated problems like TNLLR.

4.4 Crossover

The trial vector $\boldsymbol{u}_i^{G+1}$ is generated using the parent and mutated vectors as follows

$$\boldsymbol{u}_{j,i}^{G+1} = \begin{cases} \boldsymbol{v}_{j,i}^{G+1} & \text{if } \text{rand}_{j,i}[0,\ 1) \le \text{CR} \text{ or } j = k \\ \boldsymbol{x}_{j,i}^{G} & \text{otherwise} \end{cases} \tag{12}$$

where $k \in \{1, 2, \ldots, D\}$ is the randomly selected index chosen once for each i, and CR is the parameter that is a real-valued crossover factor in the range [0, 1] and controls the probability that a trial vector component comes from the randomly chosen, mutated vector $\boldsymbol{v}_{j,i}^{G+1}$, instead of the current vector $\boldsymbol{x}_{j,i}^{G}$. If CR is 1, then the trial vector $\boldsymbol{u}_i^{G+1}$ is the replica of the mutated vector $\boldsymbol{v}_i^{G+1}$.

4.5 Selection

To select the population for the next generation, the trial vector $\boldsymbol{u}_i^{G+1}$ and the target vector $\boldsymbol{x}_i^{G}$ are compared, and the individual of the next generation $\boldsymbol{x}_i^{G+1}$ is obtained according to the following rule for minimisation problems

$$\boldsymbol{x}_i^{G+1} = \begin{cases} \boldsymbol{u}_i^{G+1} & \text{if } f(\boldsymbol{u}_i^{G+1}) \le f(\boldsymbol{x}_i^{G}) \\ \boldsymbol{x}_i^{G} & \text{otherwise} \end{cases} \tag{13}$$

In the original DE, the trial vector or offspring $\boldsymbol{u}_i^{G+1}$ is compared with the target vector $\boldsymbol{x}_i^{G}$ whose index is the same as running index i using (13). More specifically, the offspring replaces the parent if it is fitter. Otherwise, the parent survives and is passed on to the next generation (iteration of the algorithm). This means that the original DE uses a greedy selection scheme where the offspring only replaces the parent if it has a better fitness score. According to the selection scheme of the original DE, a trial vector is compared with only one individual, not all the individuals in the current population. Owing to the greedy selection scheme, all the individuals of the next generation are as good as or better than their counterparts in the current generation.

In the improved DE, the only modification to the original DE is regarding the individual (parent) being replaced. In the original DE, the offspring only replaces the parent if it has a better fitness score, whereas in the improved DE the offspring replaces the most similar individual of the population (if it is fitter) [26]. Since DE uses a real-valued encoding, the similarity measure used was Euclidean distance between two candidate solutions [26]. By this scheme, as the optimisation proceeds, the individuals are scattered and gathered around the local optimal points. However, in this paper, only global optimisation is considered, and if there is no improvement of the optimal value during a predefined number of generations, then the comparison scheme is changed to that of the original DE. Therefore in the initial period of optimisation, the DE algorithm explores to find not only global but also local optima, and in the later stage, it searches only for the global optima with greedy selection scheme.

4.6 Auxiliary set

In the original DE, one population set is used: the main set P_m. In the improved DE, instead of one population set P_m, two population sets are used: the main set P_m and the auxiliary set P_a. The reason for this is to make use of potential trial points that are normally rejected in the original DE. It has been shown in [27] that the introduction of P_a increases the exploration of the search in the case of very practical large-scale global optimisation problems. Initially, two sets each containing NP points are generated in the following way: create Ω using (9) and next iteratively sample two points from Ω, the best point x_i going to P_m and the other x'_i to P_a. At each generation, if the trial point y_i, corresponding to the target x_i, does not satisfy the greedy criterion $f(y_i) < f(x_i)$, then the point y_i is not abandoned altogether, rather it competes with its corresponding target in the set P_a. If $f(y_i) < f(x'_i)$, then y_i replaces x'_i in the auxiliary set P_a. Some good points from P_a can then be used to replace some bad points in P_m periodically.

4.7 Treatment of constraints

Most optimisation problems in the real world have constraints to be satisfied. One common approach to deal with constraints is to penalise constraint violations using an appropriate penalty function [28]. In this approach, considerable effort is required to tune the penalty coefficients. In this paper, three selection criteria are used to handle the constraints of the TNLLR problem:

1. If two solutions are in the feasible region, then the one with the better fitness value is selected.

2. If one solution is feasible and the other is infeasible, then the feasible one is selected.

3. If both solutions are infeasible, then the one with the lowest amount of constraint violation is selected.

4.8 Handling of integer variables

DE in its initial form is a continuous variables optimisation algorithm, and was extended to mixed variables problems [5, 29]. During the evolution process, the integer variable is treated as a real variable, and in evaluating the objective function, the real value is transformed to the nearest integer value as follows

$$f = f(Y) : Y = y_j \tag{14}$$

where

$$y_j = \begin{cases} x_j & \text{if } x_j \text{ is integer} \\ \text{INT}(x_j) & \text{if } x_j \text{ is continuous} \end{cases} \tag{15}$$

where $INT(x_j)$ function gives the nearest integer to x_j, and the solution vector is $\boldsymbol{x} = [x_1, x_2, \ldots, x_D]$.

4.9 Evaluation

For each candidate solution $\boldsymbol{c}$, the specific NLL, $\text{SNLL}_{\boldsymbol{t}_i}$, of transformer $\boldsymbol{t}_i$ is computed by the MLP method. Next, the estimated NLL, $\text{NLL}_{\boldsymbol{t}_i}$, of transformer $\boldsymbol{t}_i$ is computed by (6), since $w_{\boldsymbol{t}_i}$ (the actual weight of the four individual cores of transformer $\boldsymbol{t}_i$) is known (it has been measured and recorded into a database). Finally, the value of the objective function is $\sum_{i=1}^{N} \text{NLL}_{\boldsymbol{t}_i}$, as (5) shows.

5 Results and discussion

This section presents NLL prediction and reduction results. In case of NLL prediction, the MLP method is compared with the NLL curve method. In case of NLL reduction, the proposed IDE-MLP is compared with the CGP and the GA-MLP method. All these methods were implemented using MATLAB 6.1 on a computer with Pentium 1.5 GHz processor. MATLAB neural network toolbox 4.0 was used for training the MLPs for TNLLR prediction.

5.1 NLL prediction

The MLP-based NLL prediction technique has been extensively tested on a transformer industry and three different environments have been examined [1, 17]. The first environment corresponds to a magnetic steel of grade M3 (according to USA AISI 1983), thickness of 0.23 mm, while the supplier of the magnetic material is Supplier A. The second corresponds to grade M4, 0.27 mm, Supplier B. The third environment has grade Hi-B, 0.23 mm thickness, also from Supplier A.

In case of the first environment, a set of 1680 actual industrial measurements (training set) has been used to train the MLP and a set of 560 independent industrial measurements (test set) has been utilised to evaluate the prediction accuracy of the MLP. The trained MLP presents a MAPE of 0.95% on the test set in comparison with 2.85% MAPE that is obtained by the NLL curve (current practice). This performance has been also verified for the other two environments where the MLP method improves the NLL prediction accuracy by more than 65% in comparison with the NLL curve method, as Table 3 shows.

5.2 NLL reduction

5.2.1 Test cases: Ten test cases have been considered that correspond to commercial transformers with rated power of 50, 100, 160, 250, 400, 630, 800, 1000, 1250 and 1600 kVA, respectively. Each test case corresponds to a production batch of 50 transformers, that is, for each test case it is requested to optimally group 100 small and 100 large cores so as to produce $N = 50$ transformers. Because of the complexity of each test case and the huge number of 5.35×10^{22} combinations of core arrangements of the

doi: 10.1049/iet-gtd.2009.0184

Table 3 Comparison of NLL curve and MLP method in the prediction of transformer NLL

Environment	MAPE on test set, %		MAPE reduction of MLP against NLL curve, %
	NLL curve	MLP	
1	2.85	0.95	66.7
2	3.76	1.18	68.6
3	3.08	1.07	65.3

whole solution space, currently there is no tool available to find the global optimum solution with 100% certainty. A Greek transformer manufacturer provided all the necessary data for these test cases.

5.2.2 Parameter values for IDE: The population size and the maximum number of generations are set to 30 and 200, respectively. The best parameter values for IDE were selected after 100 trials of IDE method with varied values of IDE parameters. The average total NLL of the final solutions for different values of IDE parameters are shown in Table 4. The best settings are $a = 0.4$, $b = 0.4$ and $CR = 0.9$, since they provide the minimum total NLL, as Table 4 shows. These settings were also confirmed for the other nine test cases of Section 5.2.1.

5.2.3 Comparison of TNLLR methods: The proposed IDE-MLP is compared with the CGP as well as the GA-MLP method. The IDE-MLP and the GA-MLP are evolutionary optimisation techniques. In each evolutionary optimisation technique: (i) the population size is set to 30, (ii) the maximum number of generations is set to 200 and (iii) the integer codification of (4) is used for the candidate solutions. In case of IDE-MLP, the parameter values of IDE are $a = 0.4$, $b = 0.4$ and $CR = 0.9$, as computed in Section 5.2.2. In case of GA-MLP, the parameter values of GA are crossover probability 0.3, mutation probability 0.06 and uniform crossover [1, 17]. The statistic results of IDE-MLP and GA-MLP over 100 trials are shown in Table 5 together

Table 4 Impact of IDE parameters on the computed final solution for the case of 100 kVA transformers

IDE parameters			Average total NLL, W
a	*b*	CR	
0.2	0.3	0.8	10 931
0.3	0.3	0.9	10 855
0.3	0.4	0.8	10 693
0.4	0.4	0.9	10 688
0.4	0.5	0.8	10 796

Table 5 Optimisation results for grouping 100 small and 100 large cores of the same transformer design with 100 kVA rated power

Parameter	Method		
	CGP	GA-MLP	IDE-MLP
minimum total NLL, W	11 221	10 737	10 685[a]
average total NLL, W		10 846	10 688
maximum total NLL, W		10 963	10 708
minimum error, %	5.02	0.49	0.00
average error, %		1.51	0.03
maximum error, %		2.60	0.22
success rate, %	0	0	83
average number of generations		168	91
average execution time per trial, min	0.23	2.45	2.08

[a]10 685 W is considered as the optimum solution and error values and success rates are computed according to this optimal solution

with the results of a single execution of CGP deterministic technique. Table 5 shows that only the proposed IDE-MLP technique converges to the optimal solution, for example, 10 685 W minimum total NLL for the 50 transformers. The success rate of IDE-MLP is 83%, that is, 83 times out of 100 trial runs the same optimal answer is obtained. As can be seen from Table 5, the execution time of all methods is very low, so all methods are applicable in an actual industrial environment; however, the IDE-MLP method is finally the best, since it is the only technique that finds the optimum solution. It should be noted that the superiority of IDE-MLP method has been also verified on the other nine test cases of Section 5.2.1.

5.2.4 Exploitation of results: It was found in Section 5.2.3 that the best technique for the solution of TNLLR problem is the proposed IDE-MLP method. That is why several core production batches of various power and voltage ratings have been grouped using IDE-MLP method and the MAPE of these batches is 1.14%, as Table 6 shows. This MAPE value is coming from the accuracy of the MLP (Section 5.1) that is used during the IDE-MLP-based grouping process. This is compared with 5.22% MAPE for the CGP, as Table 6 shows. It is common practice that the transformer designer usually uses a NLL design margin, for example, 15% higher [1] than the respective MAPE value, so NLL design margins of 1.31% and 6.00% have been used when the core grouping process is based on IDE-MLP and CGP, respectively, as can be seen from Table 6.

doi: 10.1049/iet-gtd.2009.0184

Table 6 Accuracy of optimisation methods and NLL design margin

Method	MAPE, %	NLL design margin, %
CGP	5.22	6.00
IDE–MLP	1.14	1.31

The significant reduction of NLL design margin that is due to the effectiveness of IDE-MLP in the solution of TNLLR problem yields significant reduction in the cost of transformer main materials. As an example, the design of the same transformer (same specification) with 160 kVA rated power and 315 W NLL is implemented twice. The first design, denoted as Design 1, uses 6.00% NLL design margin (the core grouping is based on CGP), so the designed NLL is $315 \times (1 - 0.06)$, that is, 296 W. The second design, denoted as Design 2, uses 1.31% NLL design margin (the core grouping is based on IDE-MLP), so the designed NLL is $315 \times (1 - 0.0131)$, that is, 311 W. With the help of appropriate software that is based on a parallel mixed integer programming-finite element method [1], the above two transformer designs are optimised, that is, their main materials cost is minimised and the results are shown in Table 7. Using the cost data of Table 7, Fig. 3 is created that compares the cost of materials of the two different designs of Table 7. Fig. 3 shows that a 100 – 92.7, that is, 7.3% cost saving on the four main materials is obtained because of the reduced NLL design margin that is a reality thanks to the use of IDE-MLP grouping method. It can be also seen from Fig. 3 that the cost saving of magnetic material is 11.8%, whereas the cost saving of winding material (copper) is 2.4%.

Table 7 Comparison of the cost of materials of two different designs for the same 160 kVA transformer specification

Description	Design 1 (CGP)	Design 2 (IDE-MLP)
rated power, kVA	160	160
specified NLL, W	315	315
NLL design margin, %	6.00	1.31
designed NLL, W	296	311
magnetic material cost, $	1825.18	1610.66
winding material cost, $	1626.32	1587.03
insulating material cost, $	194.01	172.40
oil cost, $	298.45	286.62
main materials cost, $	3943.96	3656.71

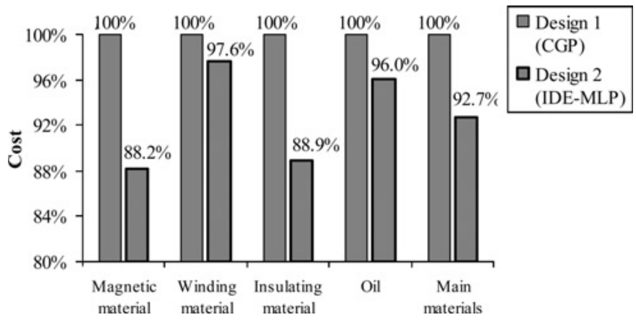


Figure 3 *Comparison of the cost of materials of the two different designs of Table 7 for the same 160 kVA transformer specification*

6 Conclusion

A novel approach combining IDE and MLPs to solve the TNLLR problem is proposed in this paper. More specifically, MLPs are used to predict NLL of wound core-type transformers prior to their assembly. The proposed IDE-MLP performs better than GA-MLP and CGP. The application of IDE-MLP method to the solution of TNLLR problem reduces the NLL design margin to 1.31%, which in turn provides 7.3% cost saving on the four main materials of transformer. Thanks to its effectiveness, the IDE-MLP method protects the manufacturer from paying NLL penalties.

7 Acknowledgments

This paper is part of the 03ED045 research project that is co-financed by E.U.-European Social Fund (75%) and the Greek Ministry of Development-GSRT (25%). Schneider Electric AE is gratefully acknowledged for providing all the necessary data to conclude the research of this paper.

8 References

[1] GEORGILAKIS P.S.: 'Spotlight on modern transformer design' (Springer, London, UK, 2009)

[2] TENYENHUIS E.G., GIRGIS R.S.: 'Measured variability of performance parameters of power and distribution transformers'. Proc. IEEE PES Transmission and Distribution Conf. and Exposition, 2006, pp. 523–528

[3] HEATHCOTE M.J.: 'The J&P transformer book' (Elsevier, 2007, 13th edn.)

[4] STORN R., PRICE K.: 'Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces', *J. Global Optim.*, 1997, **11**, pp. 341–359

[5] PRICE K.V., STORN R., LAMPINEN J.A.: 'Differential evolution: a practical approach to global optimization' (Springer, Berlin Heidelberg, 2005)

doi: 10.1049/iet-gtd.2009.0184

[6] SU C.T., LEE C.S.: 'Network reconfiguration of distribution systems using improved mixed-integer hybrid differential evolution', *IEEE Trans. Power Deliv.*, 2003, **18**, (3), pp. 1022–1027

[7] CHIOU J.P., CHANG C.F., SU C.T.: 'Ant direction hybrid differential evolution for solving large capacitor placement problems', *IEEE Trans. Power Syst.*, 2004, **19**, (4), pp. 1794–1800

[8] CHIOU J.P., CHANG C.F., SU C.T.: 'Variable scaling hybrid differential evolution for solving network reconfiguration of distribution systems', *IEEE Trans. Power Syst.*, 2005, **20**, (2), pp. 668–674

[9] LAKSHMINARASIMMAN L., SUBRAMANIAN S.: 'Short-term scheduling of hydrothermal power system with cascaded reservoirs by using modified differential evolution', *IEE Proc. Gener. Transm. Distrib.*, 2006, **153**, (6), pp. 693–700

[10] LIANG C.H., CHUNG C.Y., WONG K.P., DUAN X.Z., TSE C.T.: 'Study of differential evolution for optimal reactive power flow', *IET Gener. Transm. Distrib.*, 2007, **1**, (2), pp. 253–260

[11] WANG S.-K., CHIOU J.-P., LIU C.-W.: 'Non-smooth/non-convex economic dispatch by a novel hybrid differential evolution algorithm', *IET Gener. Transm. Distrib.*, 2007, **1**, (5), pp. 793–803

[12] YANG G.Y., DONG Z.Y., WONG K.P.: 'A modified differential evolution algorithm with fitness sharing for power system planning', *IEEE Trans. Power Syst.*, 2008, **23**, (2), pp. 514–522

[13] CAI H.R., CHUNG C.Y., WONG K.P.: 'Application of differential evolution algorithm for transient stability constrained optimal power flow', *IEEE Trans. Power Syst.*, 2008, **23**, (2), pp. 719–728

[14] WANG Z., CHUNG C.Y., WONG K.P., TSE C.T.: 'Robust power system stabilizer design under multi-operating conditions using differential evolution', *IET Gener. Transm. Distrib.*, 2008, **2**, (5), pp. 690–700

[15] VARADARAJAN M., SWARUP K.S.: 'Solving multi-objective optimal power flow using differential evolution', *IET Gener. Transm. Distrib.*, 2008, **2**, (5), pp. 720–730

[16] CHIOU J.-P.: 'A variable scaling hybrid differential evolution for solving large-scale power dispatch problems', *IET Gener. Transm. Distrib.*, 2009, **3**, (2), pp. 154–163

[17] GEORGILAKIS P.S., DOULAMIS N.D., DOULAMIS A.D., HATZIARGYRIOU N.D., KOLLIAS S.D.: 'A novel iron loss reduction technique for distribution transformers based on a combined genetic algorithm – neural network approach', *IEEE Trans. Syst. Man Cybern. C: Appl. Rev.*, 2001, **31**, (1), pp. 16–34

[18] GODEC Z.: 'Influence of slitting on core losses and magnetization curve of grain-oriented electrical steels', *IEEE Trans. Magn.*, 1977, **13**, (4), pp. 1053–1057

[19] VALKOVIC Z.: 'Influence of transformer core design on power losses', *IEEE Trans. Magn.*, 1982, **18**, (2), pp. 801–804

[20] MOSES A.J.: 'Factors affecting localized flux and iron loss distribution in laminated cores', *J. Magn. Magn. Mater.*, 1984, **41**, (1–3), pp. 409–414

[21] LING P.C.Y., MOSES A.J., MCQUADE F., GRIMMOND W., FOX D.: 'Investigation of magnetic degradation of wound cores due to adhesive bonding', *J. Magn. Magn. Mater.*, 1992, **112**, (1–3), pp. 77–80

[22] ILO A., WEISER B., BOOTH T., PFUTZNER H.: 'Influence of geometric parameters on the magnetic properties of model transformer cores', *J. Magn. Magn. Mater.*, 1996, **160**, pp. 38–40

[23] GIRGIS R.S., TENYENHUIS E.G., GRAMM K., WRETHAG J.E.: 'Experimental investigations on effect of core production attributes on transformer core loss performance', *IEEE Trans. Power Deliv.*, 1998, **13**, (2), pp. 526–531

[24] TENYENHUIS E.G., GIRGIS R.S., MECHLER G.F.: 'Other factors contributing to the core loss performance of power and distribution transformers', *IEEE Trans. Power Deliv.*, 2001, **16**, (4), pp. 648–653

[25] OLIVARES J.C., YILU L., CANEDO J.M., ESCARELA-PEREZ R., DRIESEN J., MORENO P.: 'Reducing losses in distribution transformers', *IEEE Trans. Power Deliv.*, 2003, **18**, (3), pp. 821–826

[26] THOMSEN R: 'Multimodal optimization using crowding-based differential evolution'. Proc. Evolutionary Computation Conf., 2004, pp. 1382–1389

[27] ALI M.M., TÖRN A.: 'Population set-based global optimization algorithms: some modifications and numerical studies', *Comput. Oper. Res.*, 2004, **31**, pp. 1703–1725

[28] RUNARSSON T.P., YAO X.: 'Stochastic ranking for constrained evolutionary optimization', *IEEE Trans. Evol. Comput.*, 2000, **4**, (3), pp. 284–294

[29] LAMPINEN J., ZELINKA I.: 'Mixed integer-discrete-continuous optimization by differential evolution, part I: the optimization method'. Proc. Int. Mendel Conf. Soft Computing, 1999, pp. 77–81