Safety Isolating Transformer Design using HyDE-DF algorithm

Joao Soares, Fernando Lezama, Zita Vale
GECAD
Polytechnic of Porto, Portugal
jan,flz,zav@isep.ipp.pt

Stephane Brisset, Bruno François
L2EP
Ecole Centrale de Lille, France
stephane.brisset,bruno.francois@centralelille.fr

***Abstract*—This paper presents an application of Evolutionary Computation (EC) to the benchmark of the safety isolating transformer problem. The benchmark adopts multidisciplinary optimization strategies, namely the multidisciplinary feasible (MDF) and the individual discipline feasible (IDF) formulations. The benchmark meets the requirements of engineers and scientists working with machine design problem, such as in the first part of the design process that is the choice of structure and materials. The EC methods employed in this paper are based on Evolutionary Algorithms (EAs), namely two variants of Differential Evolution (DE), two variants of Hybrid Adaptive DE (HyDE) and the Vortex Search (VS). The results showed in this paper suggest that EA methods are competitive with the classical optimization method, the sequential quadratic programming (SQP). Among the developed EAs, HyDE-DF is able to obtain better values than SQP on a significant battery of trials.**

I. Introduction

Optimal design of the safety isolation transformer is a complex problem, which requires accounting for different physical phenomena. The transformer can be represented by an analytical model [1]. Usually, analytical models can be solved within fast number evaluations and with acceptable level of accuracy. The study in [2] addresses the optimal design of the isolation transformer adopting the use of single-level multidisciplinary optimization strategies, namely the multidisciplinary feasible (MDF), individual discipline feasible (IDF) and all-at-once (AAO). While single-level methods (MDF, IDF and AAO) have been widely studied and have made full proof of their capabilities in the past. However, most of the studies available in the literature have addressed the optimal design of transformers using classical optimization approaches [2]. Evolutionary algorithms (EA) are an alternative to classical optimization and have showed their potential to a high number of applications in power systems [3]. EAs have gained some attention due to its effectiveness in providing acceptable solutions to complex problems when many times classical optimization cannot deal. EAs present numerous advantages that contribute to their success in the energy domain, including their simplicity of implementation and can handle nonlinear, non-differentiable and non-separable functions without much of convergence compromise [3]. This paper is based on the benchmark presented earlier in [4] of the multidisciplinary optimal design of a single-phase low-voltage safety isolation transformer. We apply advanced state-of-the-art EAs methods to the design problem of safety transformer and compare the results with those available on the L2EP benchmark[1]. One of the EAs we adopt and develop in the work is called Hybrid Adaptive Differential Evolution or HyDE[2]. HyDE has been developed by the authors of ISEP/GECAD and has showed excellent performance in the past in a decent number of problems and benchmark functions [5], [6], [7]

This paper is organized as follows: after this Introduction, Section II describes the optimization problem of the safety transformer design; Section III presents the formulation of two benchmark problems (MDF and IDF); Section IV presents the adopted HyDE algorithm in the paper (including other EAs that have been included for comparison); Section V presents the results and discussion of the paper while Section VI discloses the conclusions of the work.

II. Safety Transformer Design problem

Many analytical test functions are available in the literature to compare optimization algorithms. They exhibit some interesting features such as explicit equations, fast to compute, obvious minimum, and scaled decision variables. As algorithms' performances depend on the optimization problem and the model, the benchmark used in this paper aims to compare them for design (pre-sizing) problems in electrical engineering and more precisely in electromagnetic devices. This benchmark exhibits other interesting features such as multiphysics, implicit equations, highly constrained, badly scaled design variables, and multiple minima. Fully detailed materials of this benchmark are available online[1]. The physical phenomena within the transformer are thermal, electric and magnetic, all expressed in equations. This model consists in two sub-models. The first one contains electric and magnetic equations and requires the knowledge of copper temperature and voltage drop. Its assumptions are uniform distribution of magnetic induction in the iron core and no voltage drop due to the magnetizing current. The magnetic field in coils is in

[1]http://optimisation.l2ep.ec-lille.fr/benchmarks/
[2]HyDE is available in https://fernandolezama.github.io/CodesImple

This research has received funding from FEDER funds through the Operational Programme for Competitiveness and Internationalization (COMPETE 2020), under Project POCI-01-0145-FEDER-028983;by National Funds through the FCT Portuguese Foundation for Science and Technology, under Projects PTDC/EEI-EEE/28983/2017 (CENERGETIC), and UIDB/00760/2020; Joao Soares is supported by FCT CEECIND/02814/2017 grant.

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

the direction of the coil axis. The thermal sub-model requires the knowledge of iron and copper losses. Its main assumption is uniform temperature in coils and iron. To address the multidisciplinary coupling, two formulations are used.

A. Multidisciplinary feasible

The multidisciplinary feasible (MDF) formulation ensures the consistency of the model and solves the non-linear implicit system by using a fixed-point loop. The electromagnetic sub-model computes the iron and copper losses, and updates the voltage drop. The losses are inputs of the thermal sub-model that computes the temperatures. At the beginning of the next iteration, the electromagnetic sub-model updates the copper conductivity according to its temperature, and the voltage drop. Consequently, both sub-models run several times for each model evaluation.

B. Individual discipline feasible

In the individual discipline feasible (IDF) formulation, the model is not consistent. From the initial values of voltage drop and copper temperature, the electromagnetic sub-model computes the losses and updates the voltage drop. The thermal sub-model computes the temperatures from the losses. After one run of both sub-models, the copper temperature and voltage drop are a priori not equal to their initial values. The computing time of the model for IDF is consequently smaller than for MDF.

III. BENCHMARK PROBLEMS FORMULATION

According to the multidisciplinary formulation used, two optimization problems are given.

A. Multidisciplinary feasible (MDF)

The single-objective optimization problem of a safety-isolating transformer contains seven design variables. There are three parameters a, b, c for the shape of the lamination, one for the frame d, two for the section of conductors S_1, S_2, and one for the number of primary turns n_1. There are eight non-linear inequality constraints in the MDF optimization problem. The copper and iron temperatures T_{cond}, T_{iron} should be less than 120°C and 100°C, respectively. The efficiency should be higher than 80%. The relative magnetizing current I_{10}/I_1 and drop voltage $\Delta V_2/V_{20}$ should both be less than 10%. Finally, the filling factors of coils f_1 and f_2 should both be lower than 0.5. The objective is to minimize the total mass M_{tot} of iron and copper materials. As a mechanism prevents from an infinite fixed-point loop, a constraint on the residue is added to guarantee the convergence. Fig. 1 shows the structure of the safety isolating transformer considered in this problem.

To summarize, the optimization problem is minimization of M_{tot} (total mass in kg):

$$\text{minimize} \quad M_{tot} \tag{1}$$

s.t.

$$T_{cond} \leq 120^{\circ}C; \quad T_{\text{iron}} \leq 100^{\circ}C; \quad \frac{I_{10}}{I_1} \leq 0.1; \quad \frac{\Delta V_2}{V_{20}} \leq 0.1$$

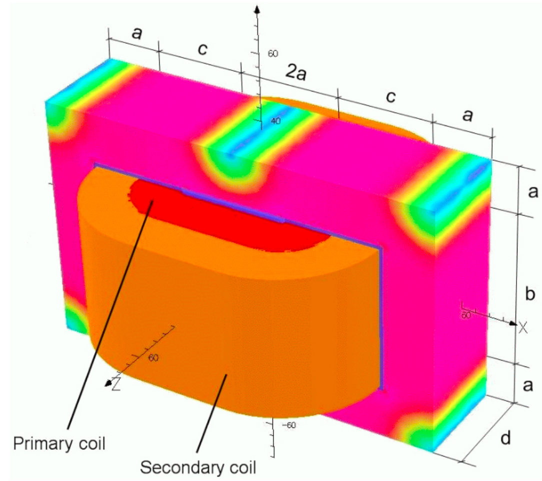


Fig. 1: Structure of a safety isolating transformer [4].

$$f_1 \leq 1; \quad f_2 \leq 1; \quad \eta \geq 0.8; \quad \text{residue} < 10^{-6}$$

$$3mm \leq a \leq 30mm; \quad 14mm \leq b \leq 95mm$$

$$6mm \leq c \leq 40mm; \quad 10mm \leq d \leq 80mm$$

$$200 \leq n_1 \leq 1200; \quad 0.15mm^2 \leq S_1 \leq 19mm^2$$

$$0.15mm^2 \leq S_2 \leq 19mm^2$$

B. Individual discipline feasible (IDF)

To ensure the model consistency, two additional equality constraints are used with two additional variables that link the physics: T_{cond_IDF} and ΔV_{2_IDF} are respectively the initial values for the copper temperature and the voltage drop. The additional equality constraints are the difference between the initial values and the updated ones.

The optimization problem becomes:

$$\text{minimize} \quad M_{tot} \tag{2}$$

s.t.

$$T_{cond} \leq 120^{\circ}C; \quad T_{\text{iron}} \leq 100^{\circ}C; \quad \frac{I_{10}}{I_1} \leq 0.1; \quad \frac{\Delta V_2}{V_{20}} \leq 0.1$$

$$f_1 \leq 1; \quad f_2 \leq 1; \quad \eta \geq 0.8; \quad \text{residue} < 10^{-6}$$

$$3mm \leq a \leq 30mm; \quad 14mm \leq b \leq 95mm$$

$$6mm \leq c \leq 40mm; \quad 10mm \leq d \leq 80mm$$

$$200 \leq n_1 \leq 1200; \quad 0.15mm^2 \leq S_1 \leq 19mm^2$$

$$0.15mm^2 \leq S_2 \leq 19mm^2$$

$$40^{\circ}C \leq T_{cond\_IDF} \leq 400^{\circ}C$$

$$0.1V \leq \Delta V_{2\_IDF} \leq 24V$$

The size of the optimization problem is larger with IDF formulation which may lead to a higher number of model evaluations. Moreover, some algorithms could be in difficulty with equality constraints.

IV. HyDE-DE and other Evolutionary Algorithms

As we can see, the optimal design of a safety isolation transformer is a complex problem that cannot be easily solved, even with the use of deterministic techniques. It is in this situations when the use of alternatives methods, such as approximate algorithms, becomes reasonable to find solutions in a more efficient way. Thus, we evoke the use of evolutionary algorithms (EA) for the proposed optimization problem. EAs are a family of optimization algorithms inspired by the evolution process seen in nature [3]. Once a problem is mathematically defined, including the objective function and a way to represent a solution to the problem, different EAs can be explored to find optimal and near-optimal solutions to a given problem. In this paper, we applied a recently proposed self adaptive version of the well-known differential evolution (DE) called Hybrid-Adaptive DE with decay function, or HyDE-DF [8]. HyDE-DF achieved the third place (out of 36 algorithms) in the 100-digit challenge at CEC/GECCO 2019 [7].

In addition, we compare its performance againts other EAs, namely DE/rand/1 and DE/target-to-best/1 [9], HyDE [5] (a previous version of HyDE-DF), and the vortex search (VS) [10]. We provide an explanation to these EAs in the following subsections.

A. Hybrid-Adaptive DE with Decay Function

HyDE-DF is inspired in the evolutionary mechanism of the original DE. For instance, HyDE-DF uses a population (Pop) of individuals (solutions to the problem) $\vec{x}_{j,i,G} = [x_{1,i,G}, ..., x_{D,i,G}]$, where G is the generation number, and $i = [1, ..., NP]$ is the number of individuals in the population, to optimize a D-dimensional function. In an initialization phase, NP solutions are generated randomly within the lower and upper ranges $[x_{lb,j}, x_{ub,j}]$ (i.e., the bounds defined in the design variables of the isolated transformer in Eqs. 1 and 2). HyDE-DF follows the general iterative process of EAs by creating new solutions applying a mutation and recombination operator, and performing elitist selection (solutions with better performance in the objective function survive into the next generation).

The first difference between DE and HyDE-DF is in the mutation operator. HyDE-DF uses a new mutation operator known as "DE/target-to-$perturbed_{best}$/1" (similar to that in HyDE) in combination with a decay function as follows:

$$\vec{m}_{i,G} = \vec{x}_{i,G} + \delta_G \cdot [F_i^1(\epsilon \cdot \vec{x}_{best} - \vec{x}_{i,G})] + F_i^2(\vec{x}_{r1,G} - \vec{x}_{r2,G}) \tag{3}$$

where F_i^1 and F_i^2, are scale factors in the range $[0, 1]$ independent for each individual i, and $\epsilon = \mathcal{N}(F_i^3, 1)$ is a random perturbation factor taken from a normal distribution with mean F_i^3 and standard deviation 1. F_i^1, F_i^2 and F_i^3 are updated each iteration following the same rule of a well-known adaptive version of DE called jDE algorithm (see Sect. III.B of [11]). The new defined operator modifies the DE/target-to-best/1 strategy with a perturbation of the best individual (inspired by the evolutionary PSO [12]). This modification tries to take advantage of the strong convergence properties towards the best solution of these two strategies, but might suffer from premature convergence in some cases. To alleviate the premature convergence effect, the δ_G factor is used to gradually decrease the influence of the term $F_i^1(\epsilon \cdot \vec{x}_{best} - \vec{x}_{i,G})$ responsible for the fast convergence towards the best individual in the population.

δ_G is a function that decreases its value from $1 \rightarrow 0$, gradually mitigating the influence towards x_{best}, and taking advantage of the inherent DE exploitation capabilities in later stages of the evolutionary process. The decay factor at each generation G is calculated as:

$$\delta_G = e^{1-1/a^2}; \qquad \text{with} \qquad a = (GEN - G)/GEN \tag{4}$$

where a is a value that linearly decreases from $1 \rightarrow 0$. Such a decrease value of a is proportional to the number of generations GEN. Figure 2 is used to illustrate the impact of the decay factor in the evolutionary process. It can be noticed that the decay factor reaches a value near to 0 at the 60% of the evolutionary process. In this way, the operator has strong exploration properties towards the x_{best} in the first part of the evolutionary process, while switching to a more local exploitation phase at the end of it.

After the mutation operator is applied, the recombination and selection process follow the same rules as the original DE. For instance, the recombination operator is applied between the mutant and the current target vector as:

$$\vec{t}_{j,i,G} = \begin{cases} \vec{m}_{j,i,G} & \text{if } (rand_{i,j}[0,1] < \text{Cr}) \vee (j = \text{Rnd}) \\ \vec{x}_{j,i,G} & \text{otherwise} \end{cases} \tag{5}$$

where Cr is the recombination parameter that is updated with the same jDE rule [13], and Rnd is a random integer in the range $[1, D]$ to guarantee that at least one element is taken from the mutant individual $\vec{m}_{i,G}$.

New individuals are evaluated in a given fitness function to measure the performance of an individual (i.e., the objective functions described in Eqs. 1 and 2). After that, the elitist selection process is performed and good solutions are preserved while solutions with lower fitness are deleted from the population.

As can be seen, the second difference between DE and HyDE-DF resides in the self-adaptation of parameters involved. HyDE-DF uses the same mechanism as jDE algorithm [13] to self-control the parameters F_i^1, F_i^2, F_i^3 and Cr_i and avoid the tuning of them for each problem. The only difference, regarding jDE, is that the main operator of HyDE-DF employs three F_i parameters instead of just one. Therefore, each individual in the population is extended with parameter values $F_i^1 = F_i^2 = F_i^3 = 0.5$ and $Cr_i = 0.9$.

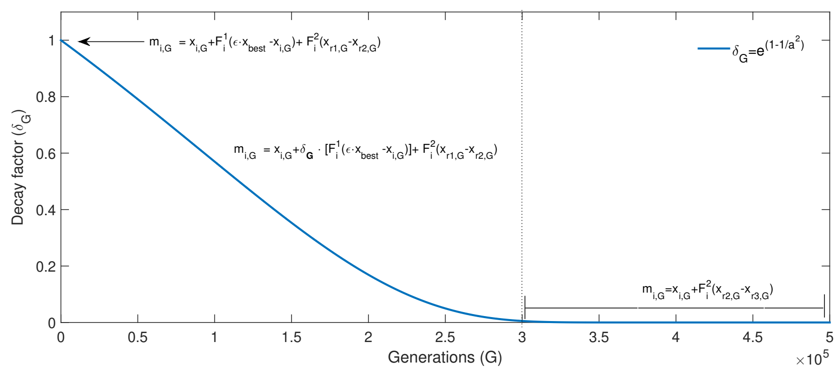


Fig. 2: HyDE-DF decay factor is used to gradually switch between the original HyDE operator to the DE/rand/1 operator.

B. Other EAs used for comparison

In this subsection, we briefly describe the algorithms used for comparison purposes. The reader is directed to the following references for further details: A detailed explanation of DE can be found in [14]; HyDE is described in [5]; VS is introduced in [10].

- DE/rand/1 is the most basic version of the main of operator of DE, yet has been successfully applied to a wide range of domains and problems. The main operator of DE/rand/1 is defined as:

$$\vec{m}_{i,G} = \vec{x}_{r1,G} + F(\vec{x}_{r2,G} - \vec{x}_{r3,G}) \tag{6}$$

  where $\vec{x}_{r1,G}, \vec{x}_{r2,G}, \vec{x}_{r3,G} \in$ Pop are three random individuals from the Pop, mutually different from each other. F is a mutation parameters usually set in the range $[0, 1]$. After that, the recombination and selection operators follows the same mechanisms as HyDE-DF.
- DE/current-to-best/1 strategy uses information of the best individual in the population to modify the convergence capabilities of the algorithm. Its main operator is defined as:

$$\vec{m}_{i,G} = \vec{x}_{i,G}+F(\vec{x}_{best,G}-\vec{x}_{i,G})+F(\vec{x}_{r1,G}-\vec{x}_{r2,G}) \tag{7}$$

  where $\vec{x}_{i,G}$ is the current target vector, and $\vec{x}_{best,G}$ is the solution with best fitness in the population. DE/target-to-best/1, in its original form, favors exploitation only since all the vectors are attracted toward the same best position found by the entire population,thereby converging faster toward the same point.
- Hybrid-adaptive DE (HyDE) is a new self-adaptive version of DE proposed in [11]. HyDE uses a new mutation operator known as "DE/target-to-$perturbed_best/1$":

$$\vec{m}_{i,G} = \vec{x}_{i,G} + F_i^1(\epsilon \cdot \vec{x}_{best} - \vec{x}_{i,G}) + F_i^2(\vec{x}_{r1,G} - \vec{x}_{r2,G}) \tag{8}$$

  where F_i^1 and F_i^2, are scale factors in the range $[0, 1]$ independent for each individual i, and $\epsilon = \mathcal{N}(F_i^3, 1)$ is a random perturbation factor taken from a normal distribution with mean F_i^3 and standard deviation 1. F_i^1, F_i^2 and F_i^3 are updated each iteration following the same rule of a well-known adaptive version of DE called jDE algorithm (see Sect. III.B of [11]). The new defined operator modifies the DE/target-to-best/1 strategy with a perturbation of the best individual (inspired by the evolutionary PSO [12]). This modification tries to take advantage of the strong convergence properties towards the best solution of these two strategies, but might suffer from premature convergence in some cases.
- Vortex search (VS) is classified as a single-solution based metaheuristic, although its framework is very similar to that of EAs as well. In each iteration, an N given number of neighbor solutions are generated using a multivariate Gaussian distribution around the initial solution. Those N solutions are evaluated in the fitness function, and the single-solution is updated with the best solution found. The radius of search is gradually reduced during the iterative process, favoring exploitation capabilities in the final iterations. This process is iterative repeated until a stop criterion is achieved [10].

C. Encoding of solutions and fitness function

One positive feature of EAs that share similar frameworks (i.e., initialization, mutation, recombination, and selection), is that they can be applied easily once a valid encoding of solutions and a fitness function is defined. The reason we selected the algorithms presented in Sect. IV, is because all of them share similar iterative frameworks, so the comparison can be done almost straightforward.

For the two defined problems (i.e., the MDF formulation from Eq. (1) and the IDF formulation from Eq. (2)), the encoding of solutions is represented by vectors including the design variables.

Particularly, a solution for the IDF formulation is defined as a vector of dimension $D = 7$ as follows:

$$\vec{x} = [a, b, c, d, n_1, S_1, S_2] \tag{9}$$

where each element represents the value of a design variable. On the other hand, the IDF formulation requires two extra variables, thus, a solution is defined as a vector of dimension $D = 9$ as follows:

$$\vec{x} = [a, b, c, d, n_1, S_1, S_2, T_{cond\_IDF}, \Delta V_{2\_IDF}] \quad (10)$$

Vectors with the above structures, having values withing the allowed bounds of desing variables, can be evaluated in objective functions described in Eqs. (1) and (2). The resulting value is called the fitness of a solution, and reflect the performance that a solution has. The less the value, the better the individual. Therefore, the fitness functions used in this study correspond to Eqs. (1) and (2).

V. Results and Discussion

In this section we present the results using deterministic and the developed EAs applied to the safety transformer problem.

A. Deterministic

The sequential quadratic programming (SQP) method from MATLAB Optimization Toolbox is used. All options are set to the default values. As the gradient is required, it is computed by using a forward finite-difference approximation, resulting in additional model evaluations. This method starts from an initial solution given by the user. If no good starting point is known, it may be drawn with the random uniform law.

Thus, the optimization process becomes stochastic and statistics for the objective values are given in Tables I and II with the mean computing time and the convergence rate (conv). This last is defined as the number of solutions with optimal objective value below the lowest one plus 2e-5 divided by the number of trials.

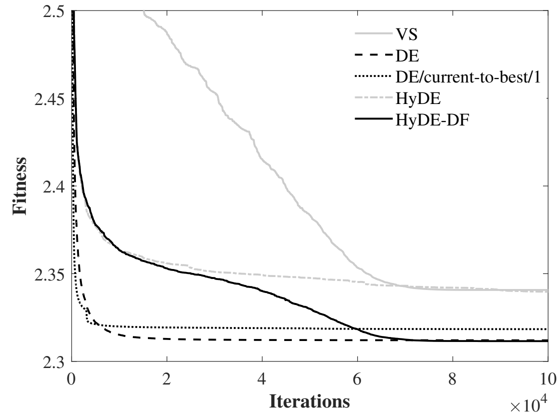
We use SQP with two variations (SQP and an improved SQP_{imp}). Both have multi-start but to increase the convergence rate of SQP, two techniques are used in SQP_{imp}): All design variables are scaled, and several initial solutions

TABLE I: Performance comparison of algorithms using MDF formulation.

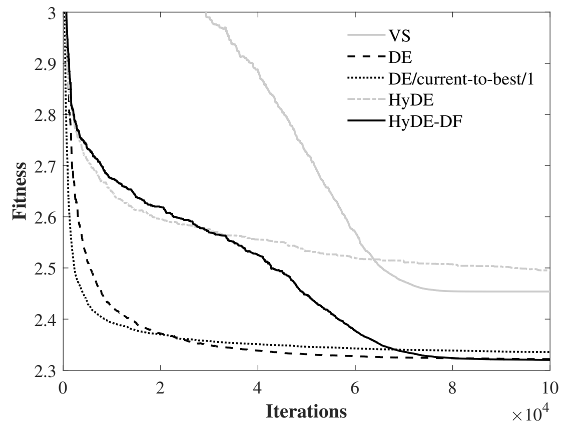
	min	max	mean	std	time	Conv
VS	2.312060	2.407100	2.340715	0.02	180.9	0.01
DE	2.311153	2.316645	2.312002	0.00	173.7	0.06
DE_best	2.311153	2.386627	2.318276	0.01	172.6	0.02
HyDE	2.315386	2.417907	2.339663	0.02	178.5	0.01
HyDE-DF	2.311151	2.321686	2.311534	0.00	178.3	0.34
SQP	2.311220	9.72453	2.998278	1.54	0.1	0.13
SQP_{imp}	2.311153	2.986595	2.331611	0.1	0.1	0.86

TABLE II: Performance comparison of algorithms using IDF formulation.

	min	max	mean	std	time	Conv
VS	2.315072	2.940729	2.453896	0.10	176.3	0.01
DE	2.311410	2.409213	2.321811	0.01	166.7	0.01
DE_best	2.311246	2.408570	2.335629	0.02	167.7	0.01
HyDE	2.366087	2.727329	2.495261	0.08	170.1	0.01
HyDE_DF	2.311187	2.405398	2.320237	0.01	176.7	0.01
SQP	2.311193	7.082574	2.750605	0.93	0.2	0.11
SQP_{imp}	2.311153	9.506476	2.437748	0.79	0.2	0.88



[a]



[b]

Fig. 3: Average convergence of the tested EA. [a] MDF formulation. [b] IDF formulation.

(multi-start) are randomly sampled. The results are given for 100 starting points with uniform sampling over the design space. In Table I, it can be seen that MDF formulation SQP reports a solution of 2.311220 kg and 2.311153 kg using SQP_{imp}. In IDF formulation (cf. Table II), SQP reported value is 2.311193 kg while SQP_{imp} is 2.311153 (the same as MDF formulation). Convergence rate is improved as expected when design variables are scaled (SQP_{imp}) in both MDF and IDF formulation, i.e. 86% and 88% against 13% and 11%, respectively. The average number of evaluations is 187.

B. Stochastic EAs

1) Algorithm settings: The parameters for each algorithm where chosen according to other studies. For DE, the mutation factor and recombination constant (F and Cr) were set to the recommended values 0.5 and 0.9 respectively [15]. HyDE and HyDE-DF [11] are a self-adaptive parameter versions but initial values for F^i and Cr where set to 0.5. VS algorithm does not have any parameter to configure [10]. The size of population (NP) chosen is 10 and iterations (GEN) is 10e4.

2) Performance comparison and results: The experiments have been run 100 times to produce the statistical results presented here. The performance of the EAs can be seen

TABLE III: Wilcoxon comparison of HyDE-DF (best EA method) against all others.

Function	HyDE-DF vs. VS				HyDE-DF vs. DE/rand/1				HyDE-DF vs. DE/current-to-best/1			
	p-value	T+	T-	Winner	p-value	T+	T-	Winner	p-value	T+	T-	Winner
MDF	4.73E-30	5048	2	'+'	8.92E-08	4.02E+03	1031	'+'	5.44E-22	4897	1.53E+02	'+'
IDF	3.21E-27	5020	30	'+'	0.258477	2855	2195	'='	1.94E-09	4181	8.69E+02	'+'
'+/=/-'				'2/0/0'				'1/1/0'				'2/0/0'

Function	HyDE-DF vs. HyDE				HyDE-DF vs. SQP				HyDE-DF vs. SQP_{imp}			
	p-value	T+	T-	Winner	p-value	T+	T-	Winner	p-value	T+	T-	Winner
MDF	1.58E-30	5050	0	'+'	5.24E-23	4928	122	'+'	8.47E-05	1403	3647	'-'
IDF	1.58E-30	5050	0	'+'	3.98E-18	4747	303	'+'	7.63E-07	1134	3916	'-'
'+/=/-'				'2/0/0'				'2/0/0'				'0/0/2'

in Table I and Table II for MDF and IDF, respectively. In MDF formulation, HyDE-DF achieves a better result than both variants of SQP, with a reported value of 2.311151 kg. Overall, DE variants are able to compete with SQP in MDF formulation. Standard HyDE and VS is not as good as the other tested EAs in this case with the worse reported min. and mean values in both MDF and IDF (but not worse than the mean values of SQP). The execution time under the proposed algorithm settings varies between 172 and 181 seconds (MDF) and 167 and 177 seconds (IDF). Standard DE versions are lighter and thus faster in both cases. In IDF formulation, the HyDE-DF can do better in mean values than SQP but not better in the min. value than SQP_{imp}. HyDE-DF obtains the highest convergence rate[3] of 34% in MDF formulation whereas the convergence rate in IDF formulation is only 1% in all the EAs, which means that a better solution may still exist. Compared with other EAs, HyDE-DF is significantly better than VS, DE variants and the standard HyDE, which is proved by the Wilcoxon test in Table III.

The convergence plot in each iteration of HyDE-DF and other tested EAs is shown in Fig. 3. HyDE-DF has similar convergence characteristics in both problems, stabilizing after 8,000 iterations. DE versions converge faster than HyDE variants and VS. However, HyDE-DF is able to overcome the the limitations of exploitation seen in early versions of DE (get stuck in a local optima). Nevertheless, DE performs quite well here when compared to more recent EAs like standard HyDE and VS. Indeed, VS has a similar convergence characteristic of HyDE-DF by also implementing a decay function, which in turns enables them to transit between exploration and exploitation (the belly curve), however VS is stuck after a while.

Since the results of HyDE-DF using NP=10 and GEN=10e4 provided inferior performance in IDF regarding the min. value when compared with SQP_{imp}, cf. Table II, we increased NP to 50 and GEN to 5e5. In a similar fashion as the former results, we present the statistics of the 100 runs in Table IV for this setting. In this case the results of both DE and HyDE-DF compared with best results available from SQP method are displayed. HyDE-DF is better than DE and SQP in all measures (cf. Wilcoxon validity test in Table V). DE performance by reference is only better or statistically equivalent to SQP_{imp} in MDF but not IDF formulation (cf. Table V) The convergence rate in HyDE-DF increases to 86% and 100% in MDF and IDF formulation but the execution time increases from a few minutes to around one hour in this setting.

[3]The convergence rate is a measure of how many solutions have been found, within the defined tolerance that are equivalent in 100 runs.

Tables VI and VII present the variables of the best solutions found for MDF and IDF formulation using the increased NP size and generations ($NP = 50$ and $GEN = 5e5$) using the respective physical units as well as the values for the constraints of the design problem.

TABLE IV: Performance comparison of best EAs increasing population size ($NP = 50$) and generations ($GEN = 5e5$) against SQP_{imp}.

	MDF formulation					
	min	max	mean	std	time	Conv
DE	2.311151	2.311151	2.311151	4E-15	4525	1
HyDE-DF	2.311151	2.311151	2.311151	2E-15	4577	1
SQP_{imp}	2.311153	2.986595	2.331611	1E-01	-	0.86
	IDF formulation					
	min	max	mean	std	time	Conv
DE	2.311154	2.311535	2.311249	8E-05	4313	0.20
HyDE-DF	2.311150	2.311150	2.311150	2E-13	4350	1
SQP_{imp}	2.311153	9.506476	2.437748	8E-01	-	0.88

VI. CONCLUSIONS

In this paper an application of Evolutionary Computation (EC) to the benchmark of the safety isolating transformer problem has been presented. The presented benchmark problem adopts multidisciplinary feasible (MDF) and the individual discipline feasible (IDF) optimization model. We apply several Evolutionary Algorithms (EAs) to the benchmark problem under MDF and IDF formulations. A comparison between EAs and SQP algorithm is extensively made. The results are evaluated using an adequate scientific approach adopting Wilcoxon test to validate the statistical meaning of the average results of the trials.

The results indicate that EA methods are competitive with the SQP method (both SQP and SQP_{imp}). Among the

TABLE V: Wilcoxon comparison of best EAs increasing population and generations against SQP_{imp}

Function	HyDE-DF vs. DE/rand/1				HyDE-DF vs. SQP_imp				SQP_imp vs. DE/rand/1			
	p-value	**T+**	**T-**	**Winner**	**p-value**	**T+**	**T-**	**Winner**	**p-value**	**T+**	**T-**	**Winner**
MDF	2.38E-07	5021	29	'+'	4.04E-28	4278	0	'+'	4.04E-28	0	4278	'-'
IDF	1.58E-30	5050	0	'+'	5.05E-29	4560	0	'+'	8.32E-11	3916	644	'+'
'+/=/-'				'2/0/0'				'2/0/0'				'1/0/1'

TABLE VI: Best solutions found (MDF formulation) increasing population size ($NP = 50$) and generations ($GEN = 5e5$)

	Parameter Desing							
	a (mm)	**b (mm)**	**c (mm)**	**d (mm)**	**n_1 (-)**	**S_1 (mm2)**	**S_2 (mm2)**	**Obj. Mass (Kg)**
HyDE_DF	0.012917	0.050122	0.016611	0.043258	640.770836	3.25E-07	2.91E-06	**2.311151**
SQP_{imp}	0.012917	0.050122	0.016611	0.043258	640.771478	3.25E-07	2.91E-06	**2.311153**

	Constraints						
	T_con	**Tiron**	**DeltaV/DeltaV20**	**I10/I1**	**f1**	**f2**	**n**
HyDE_DF	108.8182	100	0.069082	0.1	1	1	0.800001
SQP_{imp}	108.8182	100	0.069083	0.1	1	1	0.895537

TABLE VII: Best solutions found (IDF formulation) with increasing population size ($NP = 50$) and generations ($GEN = 5e5$)

	Parameter Design									
	a (mm)	**b (mm)**	**c (mm)**	**d (mm)**	**n_1 (-)**	**S_1 (mm2)**	**S_2 (mm2)**	**T_cond_IDF C°**	**Delta_V_2 V**	**Obj. Mass (Kg)**
HyDE_DF	0.01292	0.05012	0.01661	0.04326	640.77092	3.25E-07	2.91E-06	108.8181	1.6580	**2.311150**
SQP_{imp}	0.01292	0.05012	0.01661	0.04326	640.77136	3.25E-07	2.91E-06	108.8182	1.6580	**2.311153**

	Constraints						
	T_con	**Tiron**	**DeltaV/DeltaV20**	**I10/I1**	**f1**	**f2**	**n**
HyDE_DF	108.8182	100	0.069082	0.100001	1.000001	1.000001	0.8954091
SQP_{imp}	108.8182	100	0.069083	0.1	1	1	0.895537

developed EAs, HyDE-DF is able to obtain better values than SQP and other tested EAs on a significant battery of trials, regarding convergence and total mass (kg) objective function.

References

[1] T. Tran, S. Brisset, and P. Brochet, "Combinatorial and multi-level optimizations of a safety isolating transformer," *International Journal of Applied Electromagnetics and Mechanics*, vol. 26, no. 3-4, pp. 201–208, 2007.

[2] R. B. Ayed and S. Brisset, "Multidisciplinary optimization formulations benefits on space mapping techniques," *COMPEL-The international journal for computation and mathematics in electrical and electronic engineering*, vol. 31, no. 3, pp. 945–957, 2012.

[3] J. Soares, T. Pinto, F. Lezama, and H. Morais, "Survey on complex optimization and simulation for the new power systems paradigm," *Complexity*, vol. 2018, 2018.

[4] T. Tran, S. Brisset, and P. Brochet, "A benchmark for multi-objective, multi-level and combinatorial optimizations of a safety isolating transformer," in *COMPUMAG*, 2007, pp. 167–168.

[5] F. Lezama, J. Soares, R. Faia, T. Pinto, and Z. Vale, "A new hybrid-adaptive differential evolution for a smart grid application under uncertainty," in *2018 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2018, pp. 1–8.

[6] F. Lezama, J. Soares, R. Faia, and Z. Vale, "Hybrid-adaptive differential evolution with decay function (hyde-df) applied to the 100-digit challenge competition on single objective numerical optimization," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2019, pp. 7–8.

[7] K. Price, N. H. Awad, M. Z. Ali, and P. Suganthan, "The 2019 100-digit challenge on real-parameter, single objective optimization: Analysis of results," Tech. Rep., 2019.

[8] F. Lezama, J. a. Soares, R. Faia, and Z. Vale, "Hybrid-adaptive differential evolution with decay function (hyde-df) applied to the 100-digit challenge competition on single objective numerical optimization," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, ser. GECCO '19. New York, NY, USA: ACM, 2019, pp. 7–8. [Online]. Available: http://doi.acm.org/10.1145/3319619.3326747

[9] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE transactions on evolutionary computation*, vol. 15, no. 1, pp. 4–31, 2011.

[10] B. Dogan and T. Olmez, "A new metaheuristic for numerical function optimization: Vortex search algorithm," *Information Sciences*, vol. 293, pp. 125 – 145, 2015.

[11] F. Lezama, J. Soares, R. Faia, T. Pinto, and Z. Vale, "A new hybrid-adaptive differential evolution for a smart grid application under uncertainty," in *IEEE Congress on Evolutionary Computation (CEC)*, July 2018, pp. 1–8.

[12] V. Miranda and N. Fonseca, "EPSO-evolutionary particle swarm optimization, a new algorithm with applications in power systems," in *IEEE/PES Transmission and Distribution Conference and Exhibition*, vol. 2, Oct 2002, pp. 745–750.

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

[14] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, "Differential evolution using a neighborhood-based mutation operator," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 526–553, 2009.

[15] K. Price, R. M. Storn, and J. A. Lampinen, *Differential evolution: a practical approach to global optimization*. Springer Science & Business Media, 2006.