A Two-Level Genetic Algorithm for Electromagnetic Optimization

Guillaume Crevecoeur[1], Peter Sergeant[1,2], Luc Dupré[1], and Rik Van de Walle[3]

[1]Department of Electrical Energy, Systems and Automation, Ghent University, Ghent, Belgium
[2]Department of Electrotechnology, Faculty of Applied Engineering Sciences, Ghent University College, Ghent, Belgium
[3]Department of Electronics and Information Systems, Ghent University, Ghent, Belgium

Optimizing complex engineering problems may demand large computational efforts because of the use of numerical models. Global optimization can be established through the use of evolutionary algorithms, but may demand a prohibitive amount of computational time. In order to reduce the computational time, we incorporate in the global optimization procedures a physics-based fast coarse model. This paper presents a two-level genetic algorithm (2LGA) for electromagnetic optimization. This algorithm employs the global convergence properties of the genetic algorithm, where acceleration of the optimization results from the fast computations of the coarse model (low level) and where accuracy is guaranteed by using a limited number of fine model (high level) evaluations. Using the coarse model, we iteratively build surrogate models (intermediate levels) where metamodels produce surrogate models which approximate the fine model. The proposed algorithm comprises internal parameters which are self-tunable. We applied the 2LGA to the optimization of an algebraic test function, to the optimization of a die press model (TEAM Workshop Problem 25) and to the optimization of an octangular double-layered electromagnetic shield. The results show that the 2LGA is converging to the optimal solutions as the traditional genetic algorithm and that the acceleration is dependent on the accuracy of the low level. An acceleration factor of more than two can be achieved.

Index Terms—**Computationally efficient, genetic algorithm, inverse problem, optimization, shielding, two-level optimization, TEAM Workshop Problem 25.**

I. Introduction

SEARCHING the global optimum of a certain electromagnetic system, is the main goal when optimizing electromagnetic systems. Stochastic optimization algorithms: genetic algorithm [1], simulated annealing [2], particle swarm optimization [3], etc., have been extensively applied to optimization problems in electrical engineering [4]. These algorithms however need a reasonably large computational time in finding an optimal or a near-optimal solution. Parallelization of the stochastic algorithms is a remedy to the computational overhead, but is mostly not sufficient. Efficient algorithms need to be built which demand only a few model evaluations in order to arrive at an optimal or near-optimal solution.

Efficient local optimization algorithms were constructed for accelerating the electromagnetic optimization of high fidelity (fine) models (*high level*). These algorithms are based on the construction of a surrogate model within the optimization procedure. A first type of surrogate models are models which interpolate preliminary computer simulations of the fine model. Such a model is "a model of the model", also called metamodel, where optimization can be carried out within this model. Several frameworks have been proposed and developed [5], [6], where such a metamodel is used. A second type of surrogate-based optimization is the use of a physics-based coarse model (*low level*), next to the computationally demanding fine model, in the optimization procedures. Space mapping techniques [7] and manifold mapping techniques [8] were applied for designing electromagnetic devices. These techniques use a surrogate model (*intermediate level*) which is iteratively refined. A main drawback of these (local) two-level algorithms is their lack of robustness to a given problem and their global convergence properties. Hybridization, i.e., extending the algorithm with more reliable optimization techniques [9], introduction of a trust-region framework [10] has to be implemented in many cases.

Within global optimization frameworks, the recently presented Efficient Global Optimization (EGO) algorithm [11], uses surrogate models in the form of a metamodel. We present a two-level genetic algorithm (2LGA) which uses surrogate models based on a physical model in the optimization procedure and which employs the principles of a conventional genetic algorithm in order to find a global optimum in a fast way.

Manuscript received November 06, 2008; revised December 14, 2009; accepted February 15, 2010. First published March 08, 2010; current version published June 23, 2010. Corresponding author: G. Crevecoeur (e-mail: Guillaume.Crevecoeur@ugent.be).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMAG.2010.2044186

II. Surrogate Models

A. Metamodels

From a computer model $\mathbf{f}$, which is physics-based (Maxwell's equations), with input parameters $\mathbf{x} \in X_f \subset \mathrm{R}^n$ and response $\mathbf{y} \in \mathrm{R}^m$

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) \tag{1}$$

an approximation model $\mathbf{m}$ can be built [12]. X_f is the feasible region of the n-dimensional parameter vector $\mathbf{x}$. The model

$$\mathbf{s} = \mathbf{m}(\mathbf{x}) \tag{2}$$

is much more efficient to run than the time-demanding computer model and yields the possibility to get insight into the functional relationship between input parameters and responses. Further, this metamodel (2) can be used within a fast optimization framework. The responses $\mathbf{s}$ and $\mathbf{y}$ need to be similar, so that the metamodel is correctly used within such a framework. We emphasize that these models are not physics-based but that they make an as accurate as possible interpolation of a physical model. The following basic steps are involved when constructing metamodels [13].

- Selection of sample points $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ in the design space (design of experiments).
- Evaluation at each sample point in the (expensive) model $(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n)$.
- Choosing a metamodel to represent the model.
- Fitting of the response data to a surrogate model.

The first and third step are the most crucial steps. The design of experiments (DoEs) has a large impact on the number of evaluations in the expensive model (computational time) and the accuracy of the metamodel. Further, the choice of the metamodel is important for the accuracy and is problem-dependent. A proper choice of the type of metamodel, depending on the application, has to be taken. In the following, we explain shortly DoEs and widely used metamodels in electromagnetic problems.

1) Design of Experiments: Were originally developed for investigating the impact of certain parameters on a certain physical system [14]. These techniques were then applied for the analysis of computer simulations. An experimental design represents a sequence of computer experiments which need to be performed. A popular approach is the use of the Latin hypercube sampling method [15] where a collection of parameters are statistically generated. In [13], a survey of possible and more advanced designs are explained.

2) Response Surface Models: Are regression models where the m responses $s_{\mathrm{RSM},l}(\mathbf{x}), l = 1, \ldots, m$ are dependent on fitting parameters $\boldsymbol{\alpha}_l$. A full quadratic response surface model for a single response output has the following form:

$$s_{\mathrm{RSM},l}(\mathbf{x}, \boldsymbol{\alpha}_l) = \alpha_{0,l} + \boldsymbol{\alpha}_{1,l}^T \mathbf{x} + \mathbf{x}^T \boldsymbol{\alpha}_{2,l} \mathbf{x} \tag{3}$$

with $\alpha_{0,l}$ single-valued, $\boldsymbol{\alpha}_{1,l}$ an n-dimensional vector and $\boldsymbol{\alpha}_{2,l}$ an n-dimensional matrix. The above-mentioned response surface models are used in the response surface methodology optimization framework. In [16], the RSM is used for electromagnetic optimization.

3) Kriging Models: Were originally developed and used within the geostatistical community [17]. Kriging exploits the spatial correlation of data in order to build interpolations. Several approaches exist: the geostatistical approach which was initially developed uses maximum likelihood estimates (MLE) for the construction of the Kriging models. The surrogate model for the lth $(l = 1, \ldots, m)$ output can be constructed as follows:

$$s_{K,l}(\mathbf{x}) = \mathcal{V}_l(\mathbf{x}) + \mathcal{Z}_l(\mathbf{x}) \tag{4}$$

with regression model $\mathcal{V}_l$, similar to (3), and random function $\mathcal{Z}_l$. The random process $\mathcal{Z}_l$ is assumed to have mean zero and the following covariance:

$$\mathrm{Cov}[\mathcal{Z}_l(\mathbf{x}_i), \mathcal{Z}_l(\mathbf{x}_j)] = \sigma^2 \mathbf{R}(\mathcal{R}(\mathbf{x}_i, \mathbf{x}_j, \boldsymbol{\theta})) \tag{5}$$

with $\mathbf{R}$ the correlation matrix and $\mathcal{R}(\mathbf{x}_i, \mathbf{x}_j, \theta)$ the correlation function with parameter vector $\boldsymbol{\theta}$ (which has the same dimension as $\mathbf{x}$). In general, the Gaussian correlation function is adopted [18]. The regression model $\mathcal{V}_l$ makes it possible to follow the general tendency. Deviations from $\mathcal{V}_l$ are understood as local and the introduction of $\mathcal{Z}$ makes it possible to follow these fluctuations. The popular Design and Analysis of Computer Experiments (DACE), proposed in [18], is a framework for dealing with kriging approximations to expensive computer models. In [19], Kriging is used for electromagnetic device optimization.

4) Artificial Neural Networks (ANNs): Can also be used to approximate the expensive computer model $\mathbf{f}$. The neuromodel needs to be trained in such a way that it approximates the responses $\mathbf{y}$. The neural network, represented by the input-output relationship $\mathcal{N}(\mathbf{x}, \mathbf{w})$ contains internal free parameters $\mathbf{w}$ (weighting factors, bias, etc.) which are determined after training the network. For a certain number of training samples Q with parameters $\mathbf{x}_i$ $(\mathrm{i} = 1, \ldots, \mathrm{Q})$, the following cost needs to be minimized [20]:

$$\mathbf{w}^* = \arg\min \left\| [\mathbf{e}_1^T \ldots \mathbf{e}_Q^T]^T \right\| \tag{6}$$

with error vector

$$\mathbf{e}_i = \mathbf{f}(\mathbf{x}_i) - \mathcal{N}(\mathbf{x}_i, \mathbf{w}). \tag{7}$$

The number of internal free parameters (complexity of the ANN) has to be chosen sufficiently small to avoid poor generalization performance, and large enough in order to achieve a small learning error. Multilayer feed-forward (MLFF) perceptrons are most commonly used for the implementation of neuromodels [21]. The neural network can then be used for optimization purposes by using

$$\mathbf{s}_{\mathrm{ANN}}(\mathbf{x}) = \mathcal{N}(\mathbf{x}, \mathbf{w}^*) \tag{8}$$

instead of $\mathbf{f}$; see, e.g., [22].

B. Physics-Based Local Surrogate Models

The user can provide next to an expensive model $\mathbf{f}$, a much cheaper to evaluate model $\mathbf{c}$. Mostly, this model is an approximation and can not approximate the expensive model well. This model is based on the same physical laws or on an approximation of the laws employed in $\mathbf{f}$. For electromagnetic problems, approximations can be performed in geometry (e.g., 2-D geometry instead of 3-D), material characteristics (e.g., complex permeability instead of full hysteresis model), source distributions. For example, the behavior of an electrical machine where $\mathbf{f}$ is calculated using numerical methods can be approximated by a magnetic circuit; see, e.g., [23]. A coarse model can also be a numerical method with coarse discretizations instead of a fine one.

These coarse models can be used as basis for surrogates so that these surrogates approximate (1). Several forms exist, we explain briefly the basic forms. The space mapping surrogate [24] aligns the parameter spaces of the two models

$$\mathbf{s}(\mathbf{x}) = \mathbf{c}(\mathbf{p}(\mathbf{x})) \tag{9}$$

by using a parameter mapping function $\mathbf{p}$, which is linear with $\mathbf{x}$.

If the response functions are incorrect in the responses, a simple response correction $\mathbf{r}$ can be applied on the responses of $\mathbf{c}$, so that following surrogate model is obtained:

$$\mathbf{s}(\mathbf{x}) = \mathbf{r}(\mathbf{c}(\mathbf{x})). \tag{10}$$

In the manifold mapping concept, $\mathbf{r}$ is linear with $\mathbf{c}(\mathbf{x})$, so that following (local) surrogate model is obtained in the vicinity of a certain point $\tilde{\mathbf{x}}$

$$\mathbf{s}(\mathbf{x}) = \mathbf{r}(\tilde{\mathbf{x}}) + \mathbf{J}_r(\mathbf{c}(\tilde{\mathbf{x}}))[(\mathbf{c}(\mathbf{x})) - (\mathbf{c}(\tilde{\mathbf{x}}))] \tag{11}$$

with Jacobian $\mathbf{J}_r$ of $\mathbf{r}$. It is possible to make the surrogate interpolate $\mathbf{f}$ by assuming $\mathbf{r}(\tilde{\mathbf{x}}) = \mathbf{f}(\tilde{\mathbf{x}})$ and [8]

$$\mathbf{J}_r(\mathbf{c}(\tilde{\mathbf{x}})) = \mathbf{J}_{\mathbf{f}}(\tilde{\mathbf{x}})\mathbf{J}_{\mathbf{c}}^{\dagger}(\tilde{\mathbf{x}}). \tag{12}$$

We remark that the calculation of the Jacobian $\mathbf{J}_{\mathbf{f}}$ can be very time demanding.

C. Physics-Based Global Surrogate Models

The proposed 2LGA in the next section uses surrogate models which are based on the coarse model and which need to be globally valid. The starting point for building the surrogate model is a certain given set of P points

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_P] \tag{13}$$

distributed in the design variable space. These points can be evaluated in the coarse and fine model with set of responses:

$$\mathbf{F} = [\mathbf{f}(\mathbf{x}_1), \ldots, \mathbf{f}(\mathbf{x}_P)] \tag{14}$$

$$\mathbf{C} = [\mathbf{c}(\mathbf{x}_1), \ldots, \mathbf{c}(\mathbf{x}_P)] \tag{15}$$

and the following error between the two models:

$$\mathbf{E} = \mathbf{F} - \mathbf{C}. \tag{16}$$

The most simple form as surrogate model is the following:

$$\mathbf{s}_1(\mathbf{x}) = \mathbf{A} + \mathbf{B}\,\mathbf{c}(\mathbf{x}) \tag{17}$$

where the constants $\mathbf{A}$ and $\mathbf{B}$ can be fitted based on (14)–(15). Using this surrogate model only is not accurate enough because of local fluctuations of errors between coarse and fine model. Therefore, we propose to use the following surrogate model:

$$\mathbf{s}(\mathbf{x}) = \mathbf{s}_1(\mathbf{x}) + \mathbf{e}(\mathbf{x}) \tag{18}$$

where the error function $\mathbf{e}(\mathbf{x})$ has to be determined using metamodels, discussed in Section II-A. The procedure is as follows: (a) from $\mathbf{F}$ and $\mathbf{C}$ determine the constants $\mathbf{A}$ and $\mathbf{B}$ in (17) by using a least-squares routine, (b) determine $\mathbf{e}(\mathbf{x})$ by using a metamodel, e.g., training of an ANN or using a Kriging metamodel, where the input (13) and the output

$$[\mathbf{f}(\mathbf{x}_1) - \mathbf{A} - \mathbf{B}\mathbf{c}(\mathbf{x}_1), \mathbf{f}(\mathbf{x}_2) - \mathbf{A} - \mathbf{B}\mathbf{c}(\mathbf{x}_2), \mathbf{f}(\mathbf{x}_P) - \mathbf{A} - \mathbf{B}\mathbf{c}(\mathbf{x}_P)] \tag{19}$$

are interpolated. We mark that the accuracy of surrogate model (18) is dependent on the distribution of the set $\mathbf{X}$ (design of experiments).

We are aware of the fact that deviations locally occur between $\mathbf{f}$ and $\mathbf{c}$, which are important and which have to be accounted for. Therefore, the use of Kriging surrogate model (4) is a good option in order to model the local deviations. In this way, a surrogate model is constructed which tries to approximate the fine model in an as global as possible way, based on the coarse model: *global errors* between the fine and coarse model are modelled by $\mathbf{A}$ and $\mathbf{B}$, *local variations* between coarse and fine model are modelled by an error model $\mathbf{e}(\mathbf{x})$.

III. Two-Level Genetic Algorithm

A. Introduction

Two-level optimization algorithms implement two models in the optimization procedure: a computationally demanding fine model (high level) and a fast, less accurate coarse model (low level). The latter enables acceleration of the optimization procedure. This coarse model can be a metamodel or a physics-based coarse model. In each iteration of a two-level optimization algorithm, a surrogate model is built based on the coarse model and based on a limited number of evaluations in the fine model. In this way, the fine model is exploited strategically in the algorithm while the coarse model is used more for exploration in the design variable space.

Local two-level optimization algorithms were introduced which make the combination of the coarse and fine model possible. First, the space mapping algorithm was introduced [24]. This algorithm has proven its usefulness in many electromagnetic applications: [7], [25]. This algorithm basically uses surrogate model (9), which needs to be locally valid. The manifold mapping algorithm [26] uses surrogate model (10). On the basis of these main ideas, many extensions have been developed, e.g., [27]–[29].

A main drawback of these algorithms is that their convergence highly depends on the accuracy of the coarse model relatively to the fine model. Implementing a coarse and fine model into these algorithms does not always lead to an accurate solution. Indeed, in [30] a quality assessment of these coarse models for space mapping optimization is given, that makes it possible to determine a priori, i.e., before implementing the space mapping approach, if the quality of the coarse model is sufficient enough for space mapping optimization. The HASM, see [9], however deals with this problem by allowing the local two-level algorithm to use a direct optimization technique when the two-level algorithm is not converging. Moreover, these algorithms converge to a local optimum. Therefore, we introduce a global two-level algorithm with self-tunable internal parameters. The global properties of the algorithm are obtained by using the basic ideas of the genetic algorithm and acceleration of the algorithm is obtained by using a proper surrogate model. By automatically setting parameters, the algorithm provides the designer a robust, globally convergent optimization tool that accepts two models without requiring extra tuning efforts.

Several possibilities exist for the implementation of a genetic algorithm. The basic idea is to alter the optimization of the cost $\mathcal{Y}$ of the fine model

$$\mathbf{x}_f^* = \arg\min_{\mathbf{x}\in\Omega} \mathcal{Y}(\mathbf{f}(\mathbf{x})) \tag{20}$$

to the optimization of the cost of the surrogate model

$$\mathbf{x}_s^* = \arg\min_{\mathbf{x}\in\Omega} \mathcal{Y}(\mathbf{s}(\mathbf{x})). \tag{21}$$

The cost function $\mathcal{Y}$ is defined by the user and can be a least-squares, a weighted sum of the several outputs of the computer model, a minimax cost function, etc. The following types of surrogate models can be used:

1) a metamodel-based surrogate model;

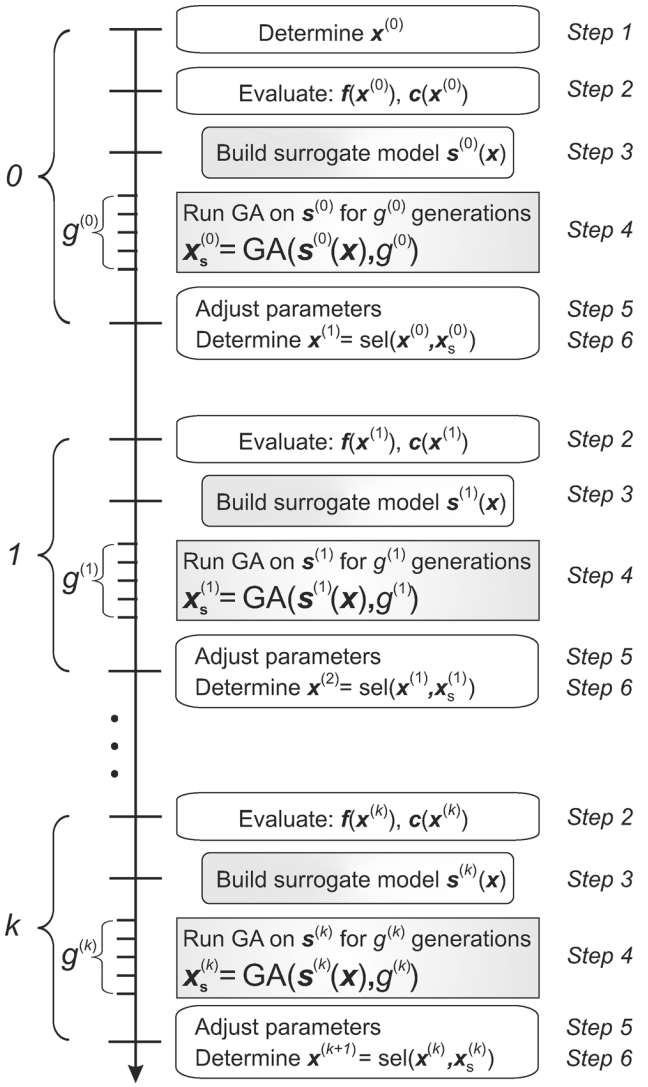

Fig. 1. Flowchart of the 2-level genetic algorithm.

2) a physics-based surrogate model;

and to use the following basic strategies during the optimization:

A) A) no additional evaluations in the fine model;

B) B) refinement of surrogate model and generation of several surrogate models.

A large number of metamodel-based optimization procedures have been developed where the surrogates are often managed in a model management framework [5], [6]. The surrogate model is constructed within a certain region and the surrogate models are sequentially generated (strategy 1B). These frameworks, together with strategies (1A) have the disadvantage to focus more on the convergence of the surrogate model itself rather than the original problem [5]. The EGO-algorithm also follows strategy (1B). We propose an algorithm which uses a surrogate-based optimization using strategy (1+2 B) where a physics-based global surrogate model (18) is used as basis.

B. Two-Level Genetic Algorithm

We explain the 2LGA, based on the flowchart of Fig. 1. The main parameters (number of individuals, number of subpopulations, termination criteria, etc.) of the traditional 1LGA do not need to be altered for the 2LGA. Further, other features of genetic algorithms, such as the non dominated sorting [31], selection/recombination/mutation strategies, study of Pareto optimal fronts, can be implemented in the 2LGA.

Step 1: The 2LGA starts by generating the initial population $\mathbf{x}^{(0)}(k = 0)$. We employ the Latin hypercube sampling method for generating randomly distributed individuals as design of the computer experiments [13]. The number of subpopulations N and the number of individuals per subpopulation n (total number of individuals: nN), are chosen the same as the traditional genetic algorithm. According to [1], nN is strictly dependent on the optimization problem, and has to be determined heuristically.

Step 2: The individuals $\mathbf{x}^{(k)}$ are evaluated in the coarse model and the fine model: $\mathbf{c}(\mathbf{x}^{(k)})$, $\mathbf{f}(\mathbf{x}^{(k)})$, respectively.

Step 3: The global surrogate model (18) is then built starting from these evaluations by determining $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{e}(\mathbf{x})$.

Step 4: The algorithm starts then optimizing the surrogate model for a number of generations $g^{(k)}$ in the traditional (one-level) genetic algorithm

$$\mathbf{x}_s^{(k)} = \mathrm{GA}\left[\mathcal{Y}(\mathbf{s}^{(k)}(\mathbf{x})), g^{(k)}\right]. \tag{22}$$

Parameter $g^{(k)}$ is an auto-tuned parameter, see the next step. The initial number of generations $g^{(0)}$ is chosen so that optimal acceleration of the algorithm is obtained and is function of the computational time of the fine model T_f and of the coarse model T_c. See Section III-C for the time equation of the 2LGA.

Step 5: The accuracy of the previously used surrogate model $\mathbf{s}^{(k-1)}, k > 0$, depends on the fidelity of the coarse model and the accuracy of the metamodel. We determine the accuracy of $\mathbf{s}^{(k-1)}$ by the following error:

$$\rho^{(k)} = \frac{\mathcal{Y}\left(\mathbf{f}\left(\mathbf{x}^{(k-1)}\right)\right) - \mathcal{Y}\left(\mathbf{f}\left(\mathbf{x}^{(k)}\right)\right)}{\mathcal{Y}\left(\mathbf{s}^{(k-1)}\left(\mathbf{x}^{(k-1)}\right)\right) - \mathcal{Y}\left(\mathbf{s}^{(k-1)}\left(\mathbf{x}^{(k)}\right)\right)}. \tag{23}$$

On the basis of $\rho^{(k)}$, we determine the number of generations $g^{(k+1)}$ in the next partial run of the genetic algorithm. We employ a trust region strategy for $g^{(k+1)}$ dependent on $\rho^{(k)}$. If $\rho^{(k)} > (3/4)$ (good correspondence between $\mathbf{s}$ and $\mathbf{f}$), the number of generations can be extended

$$g^{(k+1)} = \min\left(2g^{(k)}, g_M\right) \tag{24}$$

so that more acceleration is obtained. g_M is the maximally allowed number of generations. Further, if $(1/4) < \rho^{(k)} < (3/4)$, the number of generations remains the same as in the previous iteration

$$g^{(k+1)} = g^{(k)}. \tag{25}$$

Finally, if $\rho^{(k)} < (1/4)$, the number of generations has to be reduced

$$g^{(k+1)} = \max\left(\frac{1}{2}g^{(k)}, g_m\right) \tag{26}$$

so that more evaluations can be performed in the fine model and relatively less in the surrogate model. In

this way, the surrogate model can be refined in a better way. g_m is the minimally allowed number of generations.

Step 6: The last step consists of producing a new generation based on the best individuals in the fine model and the best in the surrogate model

$$\mathbf{x}^{(k+1)} = \mathrm{sel}\left[\mathbf{x}^{(k)}, \mathbf{x}_s^{(k)}\right]. \tag{27}$$

Here, the selection is different from the one in step 4, because a new generation has to be composed from two existing populations: on the one hand a population $\mathbf{x}_s^{(k)}$ obtained by partial optimization of the surrogate model in step 4, and on the other hand the population $\mathbf{x}^{(k)}$ of fine individuals calculated in step 2. Only 50% of the individuals can be kept because the new generation needs to have the same number of individuals as the previous ones. Therefore, in step 6, half of the population of surrogate individuals and half of the population of fine individuals are selected by stochastic universal sampling [32]. Then, both populations are merged, and subjected to cross-over and mutation. Notice that selection is carried out *before* the merging of the two populations. If the selection would be done *after* the merging, convergence difficulties may occur in case of a bad surrogate model: if the surrogate model erroneously produces much lower cost values than the fine model, almost no fine individuals are selected. This means that the algorithm optimizes then mainly the bad surrogate model, while it should optimize the fine model.

The algorithm iterates steps 2–6 until the termination criteria are met.

The partial run of the genetic algorithm (step 4) has the following basic procedures (traditional genetic algorithm): selection, recombination, mutation and migration.

Selection is carried out by stochastic universal sampling. The fittest individuals have more chance to be selected than the less fit ones, but bad individuals have a selection chance that is always greater than zero. Several other selection techniques exist that find a compromise between selecting the fittest individuals and preserving diversity [32].

Concerning the *recombination* or cross-over, [1] suggests to set the cross-over probability $P_{\mathrm{c},k}$ at generation k equal to

$$P_{\mathrm{c},k} = 0.3 + 0.4\frac{k-1}{N_\mathrm{g}-1} \tag{28}$$

with N_g the maximum number of generations.

The *mutation* is set differently for the different subpopulations. Some subpopulations have a large mutation range, which is useful especially in the beginning of the optimization in order to explore the parameter space. Other subpopulations have a small mutation range in order to approximate the optimum accurately, which becomes more important at the end of the optimization. An alternative is to set a variable mutation probability $P_{\mathrm{m},k}$ like in [1]

$$P_{\mathrm{m},k} = 0.05 - 0.04\frac{k-1}{N_\mathrm{g}-1} \tag{29}$$

in generation k.

The population in the 2LGA consists of several subpopulations that live isolated for a certain number, e.g., 20, of generations. Then, 10% of the individuals migrate to other subpopulations. *Migration* improves the diversity in the subpopulations.

C. Discussion

The two-level GA has the following properties.

- If the surrogate model is a completely wrong model (inaccurate coarse model and/or failing surrogate mapping), the 2LGA converges like a conventional one-level GA on the fine model.
- If the surrogate model represents the fine model perfectly (coarse model in combination with metamodel are perfect), the number of fine model evaluations reduces drastically due to the increasing number of generations in the surrogate model.
- The 2LGA is faster than other local two-level algorithms (space mapping, manifold mapping, etc.) if optimization of the coarse (surrogate) model needs to be carried out using a genetic algorithm. Further, the algorithm is a means for fast global optimization.
- Each internal parameter of the 2LGA is self-tunable.

The time equation of the 2LGA can be written as a function of the time needed for computing one forward calculation in the coarse and fine model: T_c, T_f, respectively. The total time for the solution using the 2LGA is given by

$$T_{\mathrm{2LGA}} = \sum_{k=0}^{K}\left[g^{(k)}nNT_c + nNT_f\right] \tag{30}$$

with $K+1$ being the total number of iterations in the 2LGA. Time can be reduced using parallel computations where each individual is evaluated in parallel ($nN = 1$ for every k).

The total time for the traditional 1LGA can be written as

$$T_{\mathrm{1LGA}} = (M_{\mathrm{1L}} + 1)nNT_f \tag{31}$$

with M_{1L} the total number of generations in the 1LGA.

Acceleration A_1 of the 2LGA with respect to the 1LGA is determined as follows:

$$\frac{1}{A_1} = \frac{T_{\mathrm{2LGA}}}{T_{\mathrm{1LGA}}}. \tag{32}$$

We determine the initial number of generations $g^{(0)}$ by assuming a certain acceleration compared to the 1LGA and by assuming that the number of generations remains constant ($g^{(k)} = g^{(0)}$) and the number of total generations $M_{\mathrm{1L}} + 1$ in the 1LGA, is the same as $(K+1)g^{(0)}$. The acceleration is then given by

$$\frac{1}{A_1} = \frac{T_c}{T_f} + \frac{1}{g^{(0)}}. \tag{33}$$

If we assume an acceleration the same as (T_f/T_c), $g^{(0)}$ needs to be taken as infinity and optimization is performed in the coarse model only with no evaluations in the fine model. Therefore, we assume heuristically the following acceleration:

$$A_1 = \frac{1}{2}\frac{T_f}{T_c} \tag{34}$$

which gives, using (33), $g^{(0)}$ (integer) closest to

$$\frac{T_f}{T_c}. \tag{35}$$

It is possible that the algorithm accelerates faster or slower than (34), depending on the accuracy of the surrogate model $(\rho^{(k)})$ relatively to the fine model.

The third property of the 2LGA, given above, can be deduced in the following way. The time equation of local two-level algorithms, in particular the Aggressive Space Mapping algorithm, is given by

$$T_{\mathrm{SM}} = S[(M_{1\mathrm{L}}+1)nNT_c] + ST_f \tag{36}$$

where in each step of the space mapping algorithm, the optimization the coarse model is carried out using a genetic algorithm [see (31)] and one evaluation is carried out in the fine model. S is the total number of iterations in the space mapping algorithm. For simplicity, we assume that the number of evaluations that need to be carried out for optimization in the coarse model, is the same as when optimizing the fine model using the genetic algorithm with time (31). We define the acceleration A_2 of the 2LGA with respect to the space mapping algorithm similar to (32)

$$\frac{1}{A_2} = \frac{T_{2\mathrm{LGA}}}{T_{\mathrm{SM}}}. \tag{37}$$

By making the same assumptions as for calculating A_1, i.e., $g^{(k)} = g^{(0)}$ and $M_{1\mathrm{L}}+1 = (K+1)g^{(0)}$, we have the following:

$$\frac{1}{A_2} = \frac{Kg^{(0)}nNT_c + KnNT_f}{S(K+1)g^{(0)}nNT_c + ST_f}. \tag{38}$$

Since we want to find the necessary condition so that $A_2 > 1$, we have

$$Kg^{(0)}nNT_c + KnNT_f < S(K+1)g^{(0)}nNT_c + ST_f \tag{39}$$

and by grouping the terms associated to T_c and T_f, we have

$$\frac{T_f}{T_c} < nNg^{(0)}\frac{(S(K+1)-K)}{KnN-S}. \tag{40}$$

If we assume that the 2LGA needs α more iterations than the space mapping algorithm: $K = \alpha S$, then we can write (40) as follows:

$$\frac{T_f}{T_c} < \frac{(\alpha^2-\alpha+1)nNg^{(0)}}{(\alpha nN-1)}. \tag{41}$$

If $\alpha = 1$ and parallel computations are carried out $(nN = 1)$, then $(T_f/T_c) < +\infty$ is always satisfied for having $A_2 > 1$. For the following realistic values $\alpha = 2$, $nN = 20$, (T_f/T_c) needs to be smaller than $1.5g^{(0)}$. If we assume for example that one minimization in the coarse model needs $M_{1\mathrm{L}} = 1000$ coarse model evaluations, and $S = 10$, then $g^{(0)} = (M_{1\mathrm{L}}+1)/(\alpha S+1) = 47.7$ or (T_f/T_c) needs to be smaller than 71.5 in order to obtain an acceleration $A_2 > 1$. As a conclusion, space mapping algorithms will be faster than the 2LGA if the computational time of the coarse model is much faster than the fine model.

IV. Optimization Problems

We applied the 2LGA to several optimization problems. Here, we explain these optimization problems. First, we applied the algorithm to an algebraic test function. In a next stage, we utilized the 2LGA for the solution of two low-frequent electromagnetic optimization problems: the optimization of a die-press and the optimal design of an octangular-shaped shield. The forward models in these two applications require the use of numerical methods (finite element method).

A. Rosenbrock Test Function

The Rosenbrock test function is an algebraic test function, which is widely used for testing new optimization algorithms. The minimum of the Rosenbrock test function is located in a narrow curved valley. Rosenbrock's Function was originally defined in two dimensions, but it can be augmented to R dimensions as well [33]

$$f_R(\mathbf{x}) = \sum_{i=1}^{R} f_{R,i} = \sum_{i=1}^{R} 100\left(\mathbf{x}_{i+1} - \mathbf{x}_i^2\right)^2 + \left(1-\mathbf{x}_i^2\right). \tag{42}$$

In this paper, $R = 2$ is implemented. The coarse model is built by implementing a linear transformation in the coarse model parameters

$$\mathbf{x}_c = \mathbf{U}\mathbf{x} + \mathbf{V} \tag{43}$$

where a global surrogate model (18) can not be constructed in a straight forward way. $\mathbf{U}$ is a 2-D matrix and $\mathbf{V}$ is a 2-D vector.

B. Optimization of a Die-Press

The die press application, depicted in Fig. 2(a), is a benchmark optimization problem (TEAM Workshop Problem 25) [34]. Four geometrical parameters $\mathbf{x} = [R_1, L_2, L_3, L_4]$, as depicted in Fig. 2(b), have to be optimized in order to produce a radial field between the inner and outer mold. The cost function is therefore defined as follows:

$$\mathcal{Y}(\mathbf{x}) = \sum_{i=1}^{n}\left[(B_{x,i}(\mathbf{x}) - B^*_{x,i})^2 + (B_{y,i}(\mathbf{x}) - B^*_{y,i})^2\right] \tag{44}$$

with $B_{x,i}(\mathbf{x})$ and $B_{y,i}(\mathbf{x})$ the calculated $x(y)$-component of the induction vector for certain geometrical parameters $\mathbf{x}$. $[B^*_{x,i}, B^*_{y,i}]$ are the objective induction values in the cavity. The induction values are uniformly distributed in the cavity with typically $n = 10$.

Two models are made of the die press application for calculating the induction vectors with $B_{x,i}(\mathbf{x})$ and $B_{y,i}(\mathbf{x})$. Both the coarse and the fine model are finite element models (FEM). The geometry, boundary conditions for the vector potential A and flux lines of the fine model are depicted in Fig. 2(a). The yoke

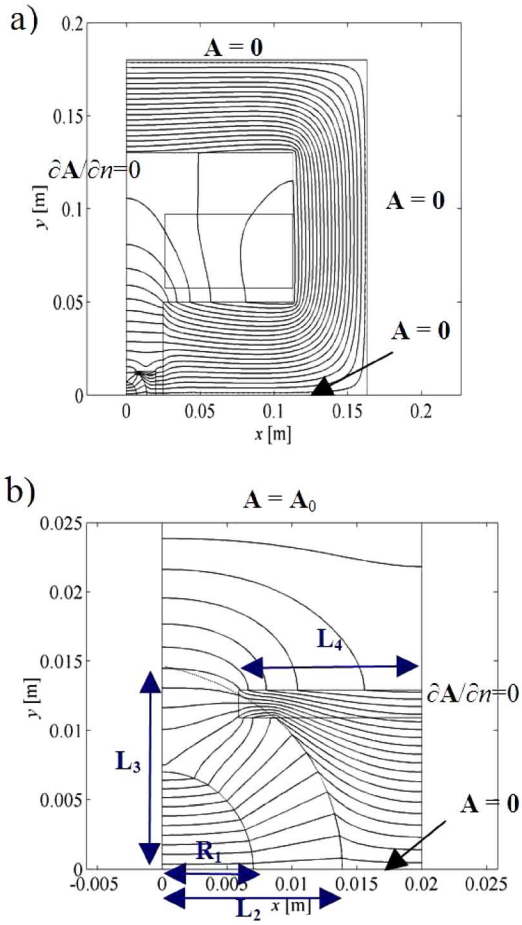

Fig. 2. Geometry, boundary conditions, and flux lines of the fine and coarse model.

consists of nonlinear magnetic material, see [34] for its characteristics. One evaluation takes approximately 3.75 s (somewhat depending on the geometry) on a 2.4 GHz computer. The coarse model does not describe the whole geometry. In this way, the mesh size or the number of unknowns is reduced and also the computational time. The coarse model geometry is shown in Fig. 2(b): it contains only the lower left part of Fig. 2(a), resulting in flux lines which are not completely the same as in the fine model. For example, the upper and lower boundary conditions $A = A_0$ and $A = 0$ force a given amount of flux through the vertical boundaries, which is correct for one default parameter vector, here $\mathbf{x} = [9e-3, 18e-3, 15e-3, 10e-3]$, only. However, for other values of the input parameter vector x, the enforced flux is an approximation of the flux in the fine model. The value of A_0 is proportional to the excitation current density where the proportionality constant is determined by comparing the flux values of coarse and fine model at the default parameter vector. When choosing the height of the domain of the coarse model larger, the more accurate the coarse model becomes. This, because the stray flux in the region $y \geq 0.015$ is modelled more accurately. The execution time is typically 1.05 s, about 36% of the CPU time for the fine model.

C. Optimization of an Octangular-Shaped Shield

An octangular double-layered passive shield is able to reduce an external uniform alternating field in the air region, enclosed

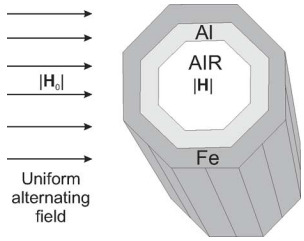

Fig. 3. Double-layered octangular shield placed in an external uniform alternating field.

by the shield. The shield, as illustrated in Fig. 3, consists of two layers of different materials—steel and aluminium—which can be more efficient than a single layered one with the same geometry [35]. Furthermore, the octangular shape has an advantage compared to a cylindrical shape: it can be built by folding a flat metal plate at the corners of the octagon, so that the faces of the octangular shield are not deformed. Indeed, in a cylindrical shape, the deformation due to bending causes mechanical stresses which deteriorate the magnetic properties. The shield is parametrized by the fraction $x_1 (0 < x_1 < 1)$ of Fe (steel) in the shield and the total thickness x_2 (m) of the shield. For practical reasons, we take 1 mm $< x_2 <$ 20 mm. The shield needs to be designed in such a way that a high field reduction is obtained in the interior of the shield, and that the electromagnetic loss and the thickness of the shield are minimized. The cost function of the given two-layered shielding configuration is given by

$$\mathcal{Y}(\mathbf{x}) = s + w_1(P_{\text{ec}} + P_{\text{hy}}) + w_2 x_2 \tag{45}$$

with $\mathbf{x} = [x_1, x_2]$ for a certain height (here, 20 cm) of the shield. The shielding factor s is found by evaluating the field in the middle of the air region inside the shield and dividing it by the uniform imposed field. The second cost term comprises the eddy current loss P_{ec} and hysteresis loss P_{hy} in the shield. The last cost term deals with the shield thickness. The chosen weighting factors are $w_1 = w_2 = 30$. A priori, a certain number of trial and error evaluations of the model were carried out. In this way, we obtained a rough idea concerning the impact of the several cost terms. On the basis of this rough idea, the user chooses the weighting factors. An alternative is to study Pareto fronts where the solution can be chosen that deserves the preference. The shielding factor s, and loss term $l = P_{\text{ec}} + P_{\text{hy}}$ can be modelled by the following two forward models.

The shield is best modelled using a 3-D numerical technique since the shield has a finite height and no axisymmetry. The model discretizes the equation

$$\nabla \times \left(\frac{1}{\mu(\mathbf{A})} \nabla \times \mathbf{A} \right) + j\omega\sigma\mathbf{A} = \mathbf{0} \tag{46}$$

where σ is the conductivity and μ the permeability. The permeability is an analytical complex function of the magnetic field, fitted from the experimental hysteresis loops as in [35], see Fig. 4. A time-harmonic, quasi-static FEM is used as numerical model for this fine model. This 3-D FEM uses first order elements. As a result of nonlinearity, the FEM requires iterative solving. The 3-D FEM has typically 220 000 degrees of freedom and the cost terms are evaluated as follows. The eddy current

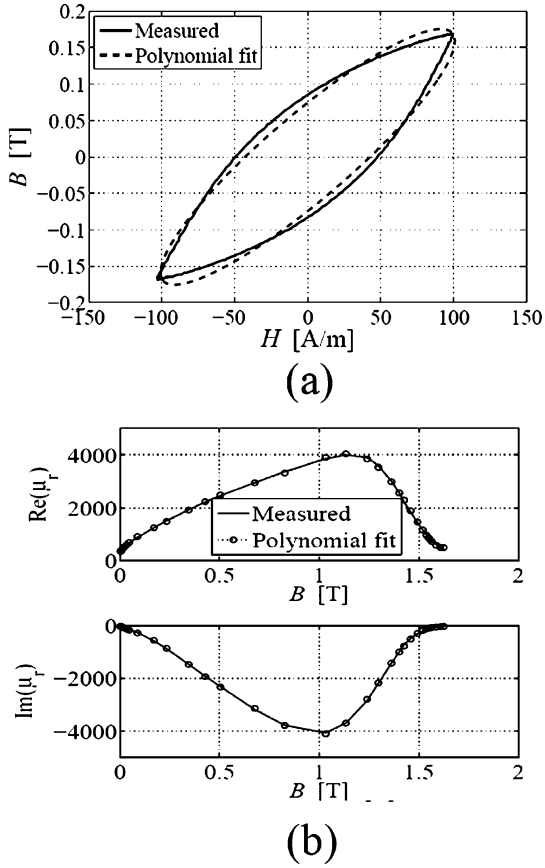

Fig. 4. (a) Fitted hysteresis loop using complex permeability to measured hysteresis loops. (b) Polynomial fit of the complex permeability to the measured hysteresis.

loss $P_{\rm ec}$ and hysteresis loss $P_{\rm hy}$ are calculated by integrals over the shield volume V

$$P_{\rm ec} = \mathrm{Re}\left[\int_V \underline{\mathbf{J}} \cdot \underline{\mathbf{E}}^* \mathrm{d}v\right] \tag{47}$$

$$P_{\rm hy} = \mathrm{Re}\left[\int_V j\omega \underline{\mathbf{B}} \cdot \underline{\mathbf{H}}^* \mathrm{d}v\right]. \tag{48}$$

Because the model uses a frequency approach, the elements of the vector $\underline{\mathbf{J}}$ are phasors. $\underline{\mathbf{E}}^*$ is the complex conjugate of $\underline{\mathbf{E}}$. The time for computing one forward evaluation in the fine model is T_f =13.2 s.

We considered a 2-D formulation as low level where we assume an infinite height of the octangular shield. The vector potential has in this case only a z-component. Because of symmetry, the model describes only a quarter of the geometry and solves (46). The 2-D FEM has typically 3200 degrees of freedom, depending on the shield. This 2-D FEM is much faster, but not so accurate. The time for computing one forward evaluation in the coarse model is: $T_c = 2.17$ s.

V. Optimization Results and Discussion

We compared the 2LGA with the traditional (1L)GA. The efficiency of the optimization algorithms are determined by investigating the number of fine and coarse model evaluations needed to come within a certain relative distance of the true global minima.

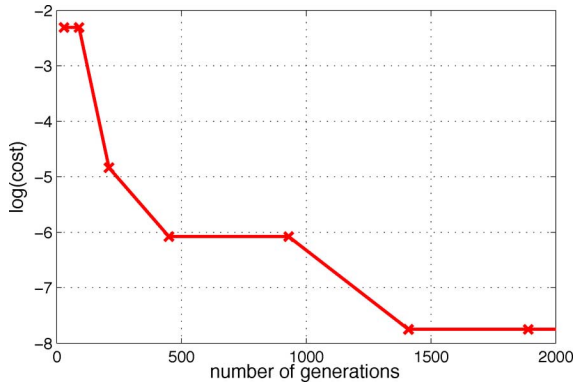

Fig. 5. 2LGA applied on the optimization of the Rosenbrock's test function.

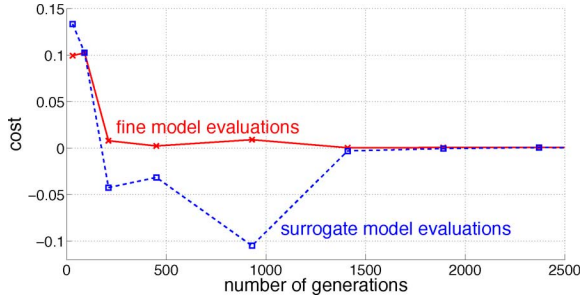

Fig. 6. Surrogate model approximates better the fine model for increasing number of generations.

A. Optimization of the Rosenbrock's Test Function

We defined in first instance the low level with (43) by $\mathbf{U} = \mathbf{I}_2$ with $\mathbf{I}_2$ the identity matrix, and vector $\mathbf{V}$ by

$$x_{c,1} = x_1 + \alpha \tag{49}$$

$$x_{c,2} = x_2 + \alpha \tag{50}$$

with $\alpha = 0.01$. The optimal solution of the coarse model is given by $\mathbf{x}_c^* = [1-\alpha,\ 1-\alpha]$. We implemented a 2LGA with $N = 2$ subpopulations with each $n = 10$ individuals. The termination criteria of the GA are: maximum number of generations: $K + 1 = 2500$ generations and minimal objective value $1.0 \cdot 10^{-8}$. We have heuristically taken the initial number of generations as $g^{(0)} = 30$ and the maximum number of possible generations as $g_M = 16 \times g^{(0)} = 480$. We used the Kriging metamodel as metamodel for determining $\mathbf{e}(\mathbf{x})$. Constants $\mathbf{A}$ and $\mathbf{B}$ of (17) are determined by a least-squares fit of the data. The convergence history of the fine model evaluations is given in Fig. 5, which depicts the best objective values. Fig. 6 shows how the surrogate model $\mathbf{s}^{(k-1)}(\mathbf{x}^{(k)})$ is approximating the fine model $\mathbf{f}(\mathbf{x}^{(k)})$ in a better way for increasing number of generations. This is because more data sets (14)–(15) are available so that the error $\mathbf{e}(\mathbf{x})$ is constructed more accurately.

As shown in Fig. 5, only $9 \times nN = 180$ evaluations are needed in the fine model, while $2500 \times nN = 50\,000$ evaluations are needed in the surrogate model (coarse model combined with metamodel).

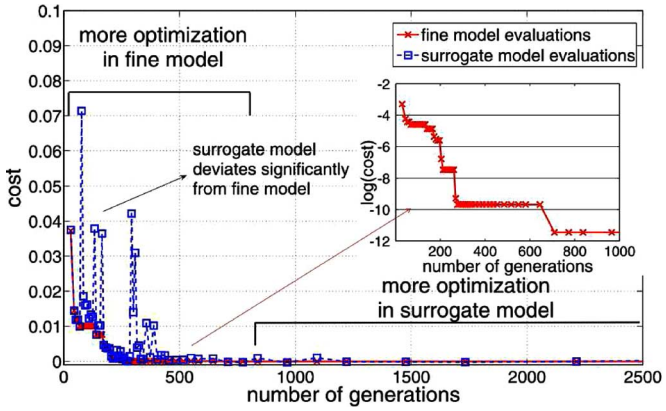

Fig. 7. Convergence history of 2LGA. First 1000 generations are more optimizing the fine model and building the surrogate model, while for the next generations, the surrogate model is more used in the GA.

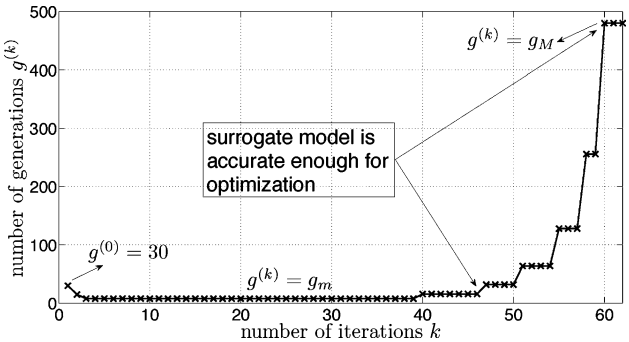

Fig. 8. Number of generations in each iteration. Each iteration corresponds with nN evaluations in the fine model.

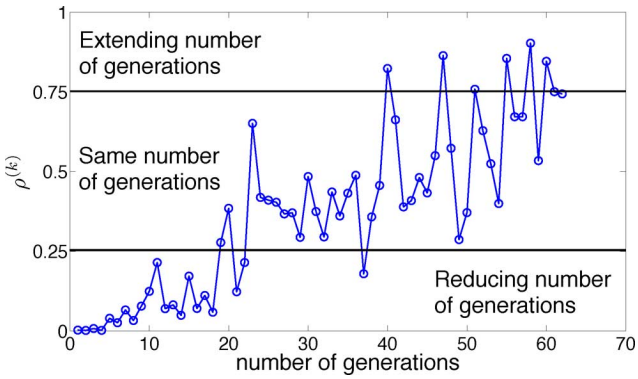

Fig. 9. Evolution of $\rho^{(k)}$ in 2LGA for optimization of Rosenbrock's test function. $\rho^{(k)}$ influences the number of generations $g^{(k)}$ when optimizing the surrogate model $\mathbf{s}^{(k)}$.

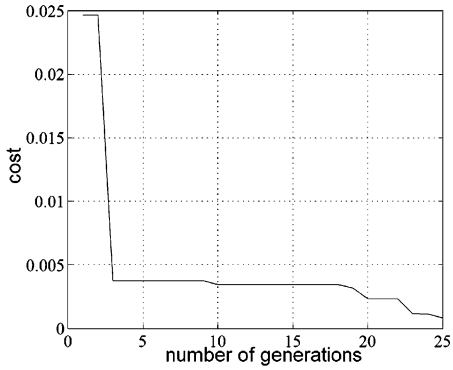

Fig. 10. Traditional genetic algorithm applied on the optimization of the "coarse" die press model.

More complex transformations can be implemented than (49)–(50), so that the low level is less accurate for modelling the high level. This implies that more evaluations are needed in the fine model so that an accurate surrogate model is built. For example, if we have the following transformation:

$$\mathbf{U} = \begin{bmatrix} 0.99 & 0.001 \\ 0.001 & 0.99 \end{bmatrix} \tag{51}$$

then the fidelity of the coarse model relatively to the fine model is very low. During the first iterations of the algorithm, the fine model is preferably optimized (number of generations in surrogate model becomes g_m). Once the surrogate model has gained enough data sets, the surrogate model is preferably used during the optimization. This is illustrated by Fig. 7. The number of generations $g^{(k)}$ in each iteration k is given in Fig. 8. For iterations 1 till 40, $\rho^{(k)}$ remains a small value $\rho^{(k)} < 3/4$, while for the next iterations $\rho^{(k)}$ becomes larger. The surrogate model $\mathbf{s}^{(k)}$ becomes accurate enough. See Fig. 9 for the evolution of the $\rho^{(k)}$ values in the 2LGA.

B. Optimization of a Die Press Model

In order to make a valuable comparison between the 2LGA and the traditional GA, we choose two subpopulations of 10 individuals each, as well as the same methods and settings for selection (stochastic universal sampling), recombination (cross-over), mutation and migration. Termination criteria are: maximum time 80 min, maximally 100 generations and minimal objective value 0.001. We used the Kriging metamodel with Gaussian correlation function as metamodel for determining $\mathbf{e}(\mathbf{x})$. Constants $\mathbf{A}$ and $\mathbf{B}$ of (17) are determined by a least-squares fit of the data. Optimization of the coarse model by a conventional 1LGA required typically 8.6 min CPU and 25 generations to obtain a final objective value of 0.000840 and the best individual is shown in Table I. The convergence history of the best cost values of the coarse model are given in Fig. 10. Optimization of the fine model by a conventional 1LGA resulted in a best objective value of 0.000756 and the best individual is shown in Table I. Convergence is shown in Fig. 11 and the total computational time needed approximately $25 \times nN \times T_c = 31$ minutes. We first implemented the 2LGA with a fixed number of generations $g^{(k)} = g^{(0)} = 5$ for the partial optimization runs of the surrogate models, i.e., step 5 in Fig. 1 is not implemented. The obtained objective value was 0.000654. When implementing the 2LGA with variable number of generations, where step 5 is used, more acceleration is obtained for the optimization of the die press. The fine model cost evaluations are shown in Fig. 13. Only 13.8 min were needed [see time (30)], yielding an acceleration of $A_1 = 2.24$, which is better than 1.60 from (34).

C. Optimization of an Electromagnetic Shield

For the optimization of the electromagnetic shield, we parallelized the calculation of the individuals. We used $N = 2$ subpopulations with each $n = 8$ individuals each. The distributed computing setup consisted of 8 nodes. As metamodel, we used the Kriging metamodel, while constants $\mathbf{A}$ and $\mathbf{B}$ of (17) are

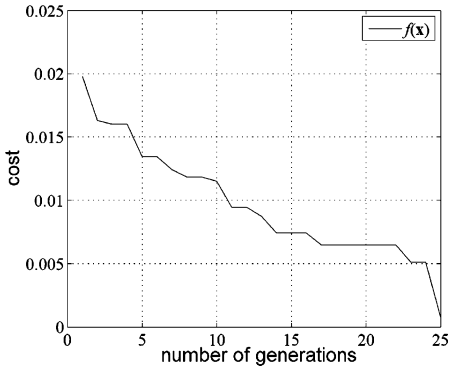

Fig. 11. Traditional genetic algorithm applied on the optimization of the "fine" die press model.

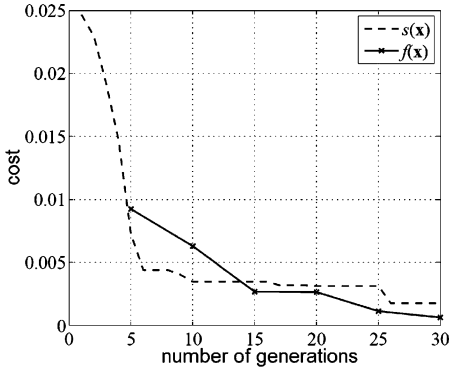

Fig. 12. Two-level genetic algorithm applied on the optimization of the die press model with fixed number of generations.

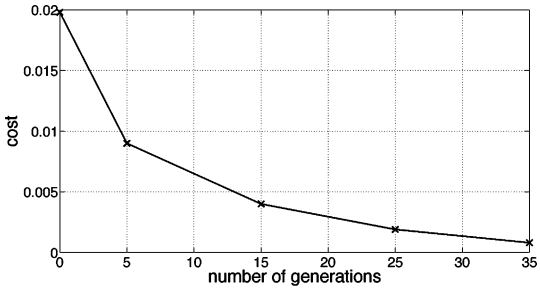

Fig. 13. Two-level genetic algorithm applied on the optimization of the die press model with variable number of generations.

TABLE I
OPTIMAL VALUES OF THE TRADITIONAL AND 2LGA FOR THE DIE PRESS APPLICATION

algorithm	R_1^*	L_2^*	L_3^*	L_4^*	time (min)
1LGA-coarse	0.00662	0.0134	0.0140	0.0140	8.6
1LGA-fine	0.00680	0.0136	0.0143	0.0135	31
2LGA-fixed	0.00682	0.01359	0.0143	0.0135	16.3
2LGA-variable	0.00681	0.0136	0.0143	0.0135	13.8

determined by a least-squares fit. The best optimal parameters values of the fine model are $x_1^* = 5.81\%$Fe and total thickness of the shield $x_2^* = 13.8$ mm with best objective value 1.0562. Convergence history of the fine model evaluations are shown in Fig. 14.

The total time needed for optimizing the electromagnetic shield took approximately 15.3 minutes. In the time (30), nN is taken 2 (the individuals in each subpopulation are computed in parallel). As a comparison, the 1LGA applied on the fine model resulted in an optimal value close ($\mathbf{x}^* = [5.5\%, 12.2\ \mathrm{mm}]$) to the optimal value of the 2LGA, using approximately 26 min (60 generations). The obtained acceleration here is $A_1 = 1.69$, which is lower than 3.04 from (34). The physical reason why the coarse model is not as accurate as wanted is because the finite height of the shield is not modelled by the coarse model.

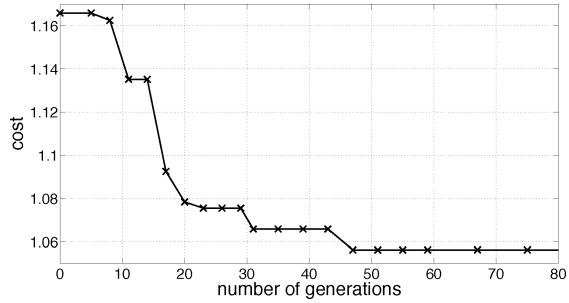

Fig. 14. Two-level genetic algorithm applied on the optimization of the octangular double-layered electromagnetic shield.

VI. CONCLUSION

We proposed a genetic algorithm where in the traditional genetic algorithm optimization of a high level model, a low level is included. Intermediate levels, in the form of surrogate models, are based on the low level and iteratively refined during the optimization. This surrogate model employs metamodels for making the interpolation possible between the high and low level. The proposed 2LGA consists of internal parameters, which are self-tunable. Further, acceleration of the algorithm is obtained if the surrogate model is accurate enough. If the surrogate model is a bad model, optimization is carried out in the fine model. In other words, the 2LGA is a robust algorithm where the accuracy of the obtained solution is not dependent on the fidelity of the coarse model. We applied the 2LGA on an algebraic test function and two electromagnetic optimization problems. We compared the 2LGA with the traditional GA and showed that it is an accurate, robust, and faster optimization procedure.

ACKNOWLEDGMENT

This work was supported under Grants BOF 011/156/04, FWO G.0082.06, and GOA 07/GOA/006, and under IUAP-project P6/21.

REFERENCES

[1] S. Alfonzetti, E. Dilettoso, and N. Salerno, "A proposal for a universal parameter configuration for genetic algorithm optimization of electromagnetic devices," *IEEE Trans. Magn.*, vol. 37, no. 5, pp. 3208–3211, Sep. 2001.
[2] S. Alfonzetti, E. Dilettoso, and N. Salerno, "Simulated annealing with restarts for the optimization of electromagnetic devices," *IEEE Trans. Magn.*, vol. 42, no. 4, pp. 1115–1118, Apr. 2006.
[3] M. Clerc, *Particle Swarm Optimization*. New York: ISTE, 2006.
[4] P. G. Alotto *et al.*, "Stochastic algorithms in electromagnetic optimization," *IEEE Trans. Magn.*, vol. 34, no. 5, pp. 3674–3684, Sep. 1998.
[5] N. Alexandrov, J. E. Dennis, Jr., R. M. Lewis, and V. Torczon, "A trust region framework for managing the use of approximation models in optimization," *Struct. Optim.*, vol. 15, no. 1, pp. 16–23, 1998.
[6] A. J. Booker, J. E. Dennis, Jr., P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset, "A rigorous framework for optimization of expensive functions by surrogates," *Struct. Optim.*, vol. 17, pp. 1–13, 1999.

[7] J. Bandler, Q. Cheng, S. Dakroury, A. Mohamed, M. Bakr, K. Madsen, and J. Søndergaard, "Space mapping: State of the art," *IEEE Trans. Microw. Theory Tech.*, vol. 52, pp. 337–361, 2004.
[8] D. Echeverría, D. Lahaye, L. Encica, E. Lomonova, P. Hemker, and A. Vandenput, "Manifold-Mapping optimization applied to linear actuator design," *IEEE Trans. Magn.*, vol. 42, no. 4, pp. 1183–1186, Apr. 2006.
[9] M. H. Bakr, J. W. Bandler, N. Georgieva, and K. Madsen, "A hybrid aggressive space-mapping algorithm for EM optimization," *IEEE Trans. Microw. Theory Tech.*, vol. 47, pp. 2440–2449, 1999.
[10] M. H. Bakr, J. W. Bandler, R. M. Biernacki, S. H. Chen, and K. Madsen, "A trust region aggressive space mapping algorithm for EM optimization," *IEEE Trans. Microw. Theory Tech.*, vol. 46, no. 12, pp. 2412–2425, Dec. 1998.
[11] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *J. Global Optim.*, vol. 13, pp. 445–492, 1998.
[12] J. P. C. Kleijnen, *Statistical Tools for Simulation Practitioners*. New York: Marcel Dekker, 1987.
[13] T. W. Simpson, J. D. Peplinski, P. N. Koch, and J. K. Allen, "Metamodels for computer-based engineering design: Survey and recommendations," *Eng. Comput.*, vol. 17, pp. 129–150, 2001.
[14] G. Box and K. Wilson, "On the experimental attainment of optimum conditions," *J. R. Statist. Soc.*, vol. B13, pp. 1–45, 1951.
[15] M. D. McKay, W. J. Conover, and R. J. Beckman, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 21, pp. 239–245, 1979.
[16] D. Dyck, D. A. Lowther, Z. Malik, R. Spence, and J. Nelder, "Response surface models of electromagnetic devices and their application to design," *IEEE Trans. Magn.*, vol. 34, no. 3, pp. 1821–1824, May 1999.
[17] G. Host, "Kriging by local polynomials," *Comput. Statist. Data Anal.*, vol. 29, pp. 259–312, 1999.
[18] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, "Design and analysis of computer experiments," *Statist. Sci.*, vol. 4, no. 4, pp. 409–435, 1989.
[19] L. Lebensztajn, C. A. R. Maretto, M. C. Costa, and J.-L. Coulomb, "Kriging: A useful tool for electromagnetic devices optimization," *IEEE Trans. Magn.*, vol. 40, no. 2, pp. 1196–1199, Mar. 2004.
[20] J. E. Rayas-Sánchez, "EM-based optimization of microwave circuits using artificial neural networks: The state-of-the-art," *IEEE Trans. Microw. Theory Tech.*, vol. 52, no. 1, pp. 420–435, Jan. 2004.
[21] P. Burrascano and M. Mongiardo, "A review of artificial neural networks applications in microwave CAD," *Int. J. RF Microw. Computer-Aided Eng.*, vol. 9, pp. 158–174, May 1999.
[22] O. A. Mohammed, D. C. Park, F. G. Uler, and Ziqiang, "Design optimization of electromagnetic devices using artificial neural networks," *IEEE Trans. Magn.*, vol. 28, no. 5, pp. 2805–2807, Sep. 1992.
[23] G. Crevecoeur, L. Dupré, and R. Van de Walle, "Space mapping optimization of the magnetic circuit of electrical machines including local material degradation," *IEEE Trans. Magn.*, vol. 43, no. 6, pp. 2609–2611, Jun. 2007.
[24] J. W. Bandler, R. M. Biernacki, S. H. Chen, P. A. Grobelny, and R. H. Hemmers, "Space mapping technique for electromagnetic optimization," *IEEE Trans. Microw. Theory Tech.*, vol. 42, no. 12, pp. 2536–2544, Dec. 1994.
[25] L. Encica, J. J. H. Paulides, and E. A. Lomonova, "Space-mapping optimization in electromechanics: An overview of algorithms and applications," in *Proc. Xth Int. Workshop on Optimization and Inverse Problems in Electromagnetism*, Ilmenau, Germany, Sep. 2008, pp. 198–199.
[26] D. Echeverría and P. Hemker, "Space mapping and defect correction," *Comput. Meth. Appl. Math.*, vol. 5, no. 1, pp. 107–136, 2005.
[27] J. Bandler, D. Hailu, K. Madsen, and F. Pedersen, "A space-mapping interpolating surrogate algorithm for highly optimized EM-based design of microwave devices," *IEEE Trans. Microw. Theory Tech.*, vol. 52, no. 11, pp. 2593–2600, Nov. 2004.
[28] M. Dorica and D. Giannacopoulos, "Response surface space mapping for electromagnetic optimization," *IEEE Trans. Magn.*, vol. 42, no. 4, pp. 1123–1126, Apr. 2006.
[29] G. Crevecoeur, P. Sergeant, L. Dupré, and R. Van de Walle, "Two-level response and parameter mapping optimization for magnetic shielding," *IEEE Trans. Magn.*, vol. 44, no. 2, pp. 301–308, Feb. 2008.
[30] S. Koziel, J. W. Bandler, and K. Madsen, "Quality assessment of coarse models and surrogates for space mapping optimization," *Optim. Eng.*, vol. 9, pp. 375–391, 2008.
[31] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
[32] J. E. Baker, "Reducing bias and inefficiency in the selection algorithm," in *Proc. 2nd Int. Conf. Genetic Algorithms and Their Application*, Hillsdale, NJ, Oct. 1987, vol. 1, pp. 14–21.
[33] R. Fletcher, *Practical Methods of Optimization*, 2nd ed. New York: Wiley, 1987.
[34] N. Takahashi, M. Natsumeda, M. Otoshi, and K. Muramatsu, "Examination of optimal design method using die press model," *Int. J. Comput. Math. Elect. Electron. Eng.*, vol. 17, no. 5, pp. 595–601, 1998.
[35] P. Sergeant, M. Zucca, L. Dupré, and P. E. Roccato, "Magnetic shielding of a cylindrical shield in nonlinear hysteretic material," *IEEE Trans. Magn.*, vol. 42, no. 10, pp. 3189–3191, Oct. 2006.

Guillaume Crevecoeur was born in 1981. He received the physical engineering degree from Ghent University, Ghent, Belgium, in 2004. In 2009, he received the degree of Doctor in engineering sciences from the same university.

He joined the Department of Electrical Energy, Systems and Automation, Ghent University, in 2004 as a doctoral student of the Special Research Fund (B.O.F.). Since 2009, he has been a Postdoctoral Researcher for the Fund of Scientific Research Flanders (FWO), Ghent University. His main research interests are numerical methods in electromagnetics, bio-electromagnetic computations, and the solution of inverse problems in electromagnetics and bio-electromagnetics for magnetic material characterization, geometrical optimization, image reconstruction, and source localization.

Peter Sergeant was born in 1978. In 2001, he graduated in electrical and mechanical engineering from Ghent University, Belgium. In 2006, he received the degree of Doctor in engineering sciences from the same university.

He joined the Department of Electrical Energy, Systems, and Automation, Ghent University in 2001 as Research Assistant. In 2006, he became a Postdoctoral Researcher for the Fund of Scientific Research Flanders (FWO), Ghent University, and, since 2008, he has been a Researcher with University College of Ghent. His main research interests are numerical methods in combination with optimization techniques to design nonlinear electromagnetic systems, in particular actuators and magnetic shields.

Luc R. Dupré was born in 1966. He graduated in electrical and mechanical engineering in 1989 and received the degree of Doctor in applied sciences in 1995, both from the Ghent University, Belgium.

He joined the Department of Electrical Power Engineering, Ghent University, in 1989 as a Research Assistant. In 1996, he became a Postdoctoral Researcher for the Fund of Scientific Research-Flanders (FWO). Since, 2002 he has been a Professor with the Engineering Faculty, Ghent University. His research interests mainly concern numerical methods for electromagnetics, especially in electrical machines, modeling, and characterization of magnetic materials.

Rik Van de Walle received the M.Sc. and Ph.D. degrees in engineering from Ghent University, Ghent, Belgium, in 1994 and 1998, respectively. After a visiting scholarship at the University of Arizona, Tucson, he returned to Ghent University. The topic of his Ph.D. and his work at the University of Arizona was medical imaging in general and, more specifically, the reconstruction of magnetic resonance images (MRI).

In 2001, he became a Professor with the Department of Electronics and Information Systems, Ghent University, and founded the Multimedia Lab. He has been involved in the organization of and/or review of papers for several international conferences and journals. His current research interests include multimedia content delivery, presentation, and archiving, coding and description of multimedia data, content adaptation, interactive (mobile) multimedia applications, and interactive digital TV. The Multimedia Lab is one of the partners of the Interdisciplinary Institute for Broadband Technology (IBBT), which was founded by the Flemish Government in 2004.