An adaptive evolution control based on confident regions for surrogate-assisted optimization

Guillaume Briffoteaux, Nouredine Melab, Mohand Mezmaz, Daniel Tuyttens

An adaptive evolution control based on confident regions for surrogate-assisted optimization

Guillaume Briffoteaux*†, Nouredine Melab†, Mohand Mezmaz*, Daniel Tuyttens*
*Mathematics and Operational Research Department (MARO)
University of Mons, Belgium
Email: {Guillaume.Briffoteaux,Mohand.Mezmaz,Daniel.Tuyttens}@umons.ac.be
†Inria Lille - Nord Europe, CNRS/CRIStAL
University of Lille, France
Email: Nouredine.Melab@univ-lille1.fr

***Abstract*—In simulation-based optimization the objective function is often computationally expensive for many optimization problems. Surrogate-assisted optimization is therefore a major approach to efficiently solve them. One of the major issues of this approach is how to integrate the approximate models (surrogates or metamodels) in the optimization process. The challenge is to find the best trade-off between the quality (in terms of precision) of the provided solutions and the efficiency (in terms of execution time) of the resolution. In this paper, we investigate the evolution control that alternates between the simulator and the surrogate within the optimization process. We propose an adaptive evolution control mechanism based on the distance-based concept of confident regions. The approach has been integrated into an ANN-assisted NSGA-2 and experimented using the ZDT4 multi-modal benchmark function. The reported results show that the proposed approach outperforms two other existing ones.**

***Index Terms*—Surrogate-modeling, multi-objective optimization, evolution control, direct fitness replacement, machine learning.**

I. Introduction

In simulation-based optimization, the evaluation of the fitness function relies on complex simulations often characterized by a high computational cost [1]. For many engineering problems, the dimensionality is high, the cost function is non-convex and no gradient information is available. For these problems, Evolutionary Algorithms (EAs) have proven to be efficient optimization techniques [2]. The idea behind these algorithms is to simulate the natural evolution of species. An initial population of solutions, generated by sampling the search space, is iteratively evolved by selecting parent individuals, mating them and replacing them by the generated offsprings if they improve the considered objectives. The process usually leads to a high number of fitness assessments. Improving the design of a manufactured product, for instance, could require to evaluate a great number of different models implying a great number of simulator calls and making the procedure unrealizable given certain computational time budgets. It is worth noticing that the computational time budget can be expressed in terms of number of simulator calls since other operations are negligible in a lot of engineering cases such as studies involving Computational Fluid Dynamic (CFD) [3].

A way to circumvent such computational constraints is to resort to Meta-Models (MMs). Meta-models, also called surrogate-models, aim at emulating and simplifying an actual model making its evaluation cheaper but at the cost of a coarser accuracy. The trade-off between the accuracy of the MM and the computational costs entailed to build it has to be controlled to respond to the final purpose. The term *model management* is usually used to refer to this task.

Indeed, when addressing meta-modeling, several possibilities can be examined [4] as the number of MMs, the way they collaborate with each other, their fidelity, their type, the region of the decision space they act upon and the objective they emulate but also and more importantly the way they cooperate with the real fitness function and the EA. In this sense, a taxonomy has been proposed by A.Díaz-Manríquez *et al.* [5] to categorize the studies of this field, with an emphasis on fitness replacement. Since the surrogate-model is constructed based on the real fitness function and aims at substituting this last, a strategy has to be devised to determine whether calling the real fitness function or the MM at a given step of the evolution, and when building or updating the surrogate. Two fitness replacement modes can be envisioned: Direct Fitness Replacement (DFR) and Indirect Fitness Replacement (IFR). In the former, it is assumed that a surrogate-evaluated solution carries the same degree of accuracy that a real-evaluated one and the EA is allowed to embed it into the population at the replacement stage of the evolution. In the latter, the EA appeals to the approximated model so as to give a lift to exploitation, letting it only proposing new offsprings then re-evaluated with the original fitness function if they are selected. The drawback of the first approach is to converge toward a false optimum while the second approach might converge prematurely.

In DFR, the MM is used alternately with the simulator according to an alternation criterion, or evolution control. For low dimensional problems, the EA may exclusively call the surrogate-model previously built thanks to simulations

performed in the past. This concept is called No Evolution Control (NEC) since it is decided *a priori* that only the MM serves the evolution. A particular attention has to be paid to the accuracy of the approximated model. For problems of higher dimension, it could be better to alternate between the fitness functions during the evolution. A control parameter sets the percentage of original and surrogate evaluations performed at a generational or individual level. This strategy is referred to as Fixed Evolution Control (FEC). The advantage is the opportunity of updating the surrogate through the search and thus refining its precision on new regions of interest. Nevertheless, the disadvantage is the adding of a user-defined parameter for the control. To avoid this drawback, the Adaptive Evolution Control (AEC) suggests to test the accuracy of the MM as an alternation criterion. The issue that arises is how to evaluate the surrogate precision.

In this paper, an heuristic is proposed to define an alternation criterion for AEC between the MM and the original fitness function based on the known sub-regions of the decision space. In Section 2, a review of some studies illustrating the different model management approaches is presented. Section 3 is dedicated to introduce the Non-dominated Sorting Genetic Algorithm (NSGA-II) for Multi-objective Optimization (MO) and Artificial Neural Network (ANN) for surrogate modeling. In Section 4, the confidence regions AEC is proposed along with a detailed presentation of two evolution controls from past studies. Preliminary experiments and results are reported in Section 5, and conclusions and some directions about further works are made in Section 6.

II. Related works

A cooperation strategy between elitist NSGA-II and an ANN is proposed in [6] and validated in a cantilever plate optimal shape problem according to two objectives. The percentage of real fitness evaluations and surrogate fitness evaluations is fixed *a priori*. For a predefined number of Q generations, the n first are produced based on the real fitness function and the remaining ones based on the surrogate fitness function. The model management considered is DFR with a FEC.

The drawbacks of the previous work is that MM precision is not taken into account to determine the values of the parameters Q and n. From this observation, A.Gaspar-Cunha and A.Vieira offer to take advantage of the ANN approximation error to switch from one fitness function to the other [7]. The evolution control thus becomes adaptive. The real optimization problem considered in this study consists in maximizing the mass and the mixing degree of a screw produced by a single-screw polymer extruder.

In [8], a combination of different surrogate-models is presented, in an IFR flavor, to evaluate candidate solutions chosen randomly in the decision space. The best candidate solution is then re-evaluated with the real fitness function and compared to the best solution found so far. This single-objective non-evolutionary optimization technique is validated solving a 12-dimensional ground water bioremediation problem where the objective is to decontaminate ground water at minimum cost. The fitness function evaluation requires a system of nonlinear partial differential equations reflecting both water flow and chemical reactions to be treated and takes up to 4 minutes.

In [1], a MO strategy based on an iteratively enhanced Kriging meta-model coupled to a fully parallel 3D CFD solver is proposed to improve the design of gas turbine engines. Kriging, also named Gaussian Processes, is a regression method appreciated for the uncertainty measure it provides on the predictions. Predictions and uncertainties are combined into a new function on which local search is performed to propose new candidate solutions. The model management applied here is IFR since candidate solutions are all evaluated with the solver. The total number of CFD evaluations is one of the stopping criteria of the whole search.

C.Poloni *et al.* propose an approach based on the collaboration between a two-objective Genetic Algorithm (GA), a Conjugate Gradient (CG) algorithm, an ANN as surrogate-model and a 3D Navier-Stokes CFD solver to enhance the hydro-dynamism of a sailing yacht [9]. The GA is first run to provide both an approximation of the Pareto front and an initial database to construct the ANN. The second step of the search is carried by CG, initialized thanks to the Pareto approximation, in cooperation with the ANN which is updated after each step. The model management applied here is IFR since, at each CG step, the solver is called to evaluate the proposed solution.

Scheduling problems are also tackled by MM-assisted optimization as shown in [10] where ANN approximations and uncertainties are used to optimize a factory production planning maximizing devices utilization and minimizing tardiness. In a parallel EA, offsprings are evaluated and ranked thanks to the meta-model, the ones marked with the better rank are re-evaluated with the original cost function before to be inserted into the population. Given that the population is solely constituted of solutions evaluated with the real fitness function, the model management is qualified as IFR.

To improve the aerodynamism of an airfoil, an ANN surrogate-model assists a GA based on a 3D flow simulator in [3]. The first generations are obtained with the simulator and evaluations are kept to built the ANN. Then, GA relies on the ANN and only the best solutions obtained at the end of a generation are re-evaluated with the simulator. The model management is then a mix between DFR and IFR.

M.Mlakar *et al.* combine Gaussian Processes with a multi-objective differential evolutionary algorithm [11]. The alternation between the surrogate and the real fitness function happens at the replacement step of the EA and depends on the uncertainty upon the prediction of the offspring objective values. If the original version of the MM is good enough, the model management is a DFR, otherwise it can be defined as

IFR or as a mix of both fitness replacements. Optimization of the quality of cast steel, is chosen to validate the strategy. Given a model of the steel casting process, the goal is to determine the best input values of the simulator to obtain the desired outputs. It is indicated than one simulation lasts approximately 4 minutes on a single computing core.

Inspired by [6] and [7], an heuristic-based adaptive evolution control is presented in this paper and compared to the previous studies. NSGA-II and ANN are chosen as MO algorithm and surrogate model respectively.

III. Background on NSGA-II and ANN

A. Comparison in multi-objective optimization

In multi-objective optimization the comparison between two solutions is performed according several criteria. A set of dedicated comparison operators are available [12] in order to assist the decision maker in his/her final choice.

Given a, b two decision vectors, from the decision space, and X a set of decision vectors containing a, it is stated that:

- a *dominates* b iff a is at least better than b regarding one criterion and as good as b regarding the remaining criteria.
- a and b are *incomparable* iff no solution dominates the other and there exists at least one criterion whose value differs from a to b.
- a is a *non-dominated solution* of X iff there is no solution in X that dominates a.

These definitions still stand for objective vectors picked up within the objective space.

Given a population of solutions, the subset composed of non-dominated individuals, such that one non-dominated individual is incomparable with all other non-dominated individuals, is called the *Non-Dominated Front* (NDF). To confront NDFs with each other, the accuracy of the solutions just as their spread and distribution over the objective space can be analyzed [13]. The best NDF for a given problem is called the *Pareto front* of this problem.

B. Non-dominated Sorting Genetic Algorithm (NSGA-II)

The idea behind population-based evolutionary algorithms for optimization is to simulate the natural evolution of species. Algorithm 1 gives the general skeleton of EAs emphasizing the main operations performed over the population: initialization, selection, reproduction (mating) and replacement. The actual choice of operators and their parameters depends on the desire trade-off between diversification, understood as the exploration of the entire search space, and intensification, the exploitation of solution characteristics encountered in a promising region of the search space.

The particularity of NSGA-II is the use of the crowded-comparison operator both at the selection and replacement step [14]. When comparing two solutions, both the accuracy, represented by the rank of the NDF the solutions belong to, and the distribution, shown by the crowded distance are taken into account. To the NDF of the whole population is assigned rank 1 (the best), then after removing solutions of rank 1, the NDF of the remaining population is ranked 2. Repeating the procedure allows the individuals to be classified by group depending on their accuracy. To distinguish solutions of the same rank, the crowded distance indicating the average distance from one solutions to its closest neighbors along each objective is computed. A higher value of the crowded distance exhibits a solution with a wider empty neighborhood, which is attractive to enhance the distribution of the population.

Algorithm 1 Basic sketch of population-based evolutionary algorithms.

```
Initialize a population P
while the stopping criterion is not met do
    Select a set of parents from P
    Mate the parents to father a set of offsprings
    Evaluate the offsprings
    if the replacement criterion is met then
        Replace the parents by the offsprings to form the new
        population
    end if
end while
return Non-dominated front from the last population
```

In NSGA-II, crossover and mutation are the reproduction operators employed to generate offsprings from parents. Crossover combines good features extracted from each parent to produce a child thus increasing intensification while mutation disrupts the child by modifying the features inherited from its parents thus reinforcing diversity. A lot of versions of these operators have been proposed in order to reach different shade of exploitation and exploration under the constraint of producing valid candidate solutions [2].

C. Artificial Neural Networks (ANN)

Artificial Neural Networks are often selected as surrogate models thanks to their approximation universality [15] and ease of update. An ANN architecture is composed of units organized in 1-D fully connected layers: input layer, hidden layers and output layer. One unit of a given layer is linked with all units from the previous and next layers by a weighted connection. Each unit is also equipped with an activation function f supposed to be the same for all hidden units in this application. Training a network requires a set of sample data called *training set* and composed of several inputs with corresponding outputs. Inputs flow through the layers such that $x^{(i)}$, the input of layer l_i, is given by:

$$x^{(i)} = W^{(i-1)}.f(x^{(i-1)}) + b^{(i-1)} \tag{1}$$

with $W^{(i-1)}$ and $b^{(i-1)}$ respectively the matrix of weights and the vector of biases from the $l_{i-1} \rightarrow l_i$ connection. The output layer gives finally an estimation of the expected

output. A gradient descent (GD) algorithm is used iteratively to improve the estimation accuracy by updating weights and biases thanks to backpropagation [16].

The way data is fed into the network during the training phase is of important relevance for the training quality. Two training procedures have been proposed. The first, called *batch training*, consists in gathering training data in batches and update the ANN's weights once for the whole batch. The second, called *iterative learning*, considers one sample at a time. On the one hand, all the batch samples contribute together to modify the ANN weights and biases whereas on the other hand, only one sample leads to the update.

Hyperparameters (such as the number of neurones, number of layers and learning rate) and methods (such as GD algorithm, sample normalization and early stopping) specializing the ANN are difficult to be tuned because of the number of possible configurations and the problem dependence.

IV. A new confidence-regions Adaptive Evolution Control

The general purpose of the adaptive evolution control proposed hereafter is to reduce the number of real fitness assessments while keeping a good convergence toward the Pareto front as well as preserving a good spread of solutions. Strategies proposed in [6] and [7] are reproduced with NSGA-II as EA and ANN as MM for comparison purposes.

In the DFR-FEC strategy, called B_Q_n in [6] and reproduced in [7], cycles (shown in Figure 1), each producing $(p+q)$ generations, are run until the real evaluations budget is exhausted. The first p populations are fathered using the real fitness function only, then the surrogate is constructed or updated and the last q generations are created relying on the surrogate only. The determination of the parameters p and q such as the training mode are let to the user. Except for the first p generations of the first cycle, the population can be composed of individuals evaluated with the surrogate function, consequently a low value of p with regard to q provokes a low accuracy on the surrogate predictions and may drive the search in a wrong direction. Conversely, a high value of p, particularly at the beginning of the EA, leads to restricting the search to a narrow location and low diversity.

The model management suggested in [7], named $e_{NN}-EC$ in this paper, offers to automatically adapt the number q of MM-based generations depending on the approximation error commited by the ANN. For a given cycle i, p generations are first built with the real fitness function, the surrogate is then built or updated and the very next generation is obtained with the surrogate. During this latter a few number M of solutions are assessed with both fitness functions to compute the approximation error e_{NN}. If $e_{NN} > e_0$, with e_0 a given threshold, the surrogate is not trusted any more and the cycle ends with $q_i = 1$, otherwise the same procedure is repeated. In order to ensure the end of a cycle, a limit q_{max} is imposed on the number of surrogate-based generations. This trust criterion alleviates the probable damages sustained by an inaccurate surrogate with the drawback of defining the parameters p, q_{max}, e_0 and M. Moreover, the selection of individuals to be assessed with the real fitness function to compute e_{NN} is carried out randomly so that these evaluations might not benefit the next surrogate update.

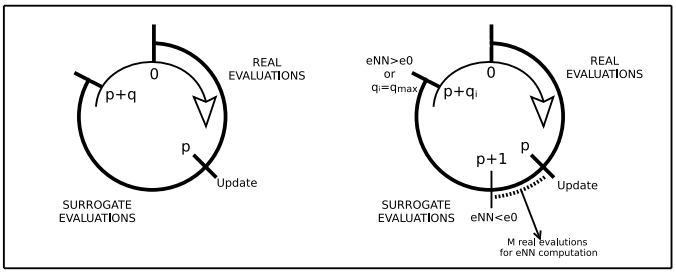

Fig. 1. Cycle of generations. $B\_Q\_n - EC$ on the left and $e_{NN} - EC$ on the right.

In covariance-based MM such as Kriging [11], the expected error on predictions is low at points used to construct the surrogate and increases when moving away from these known points. Even if ANN is not an interpolation method, a similar behavior could be expected assuming an acceptable training error. Based on this observation and on the two methods depicted previously, it is proposed to alternate between the fitness functions at an individual level depending on the distance between the candidate individual and the known solutions. In other terms, given S^* the set of solutions evaluated with the original cost function that have been used to construct the surrogate and s a candidate solution to be evaluated. If there exists $s* \in S^*$ such that $d(s, s^*) < \epsilon$ then the candidate solution resides within the confidence region where the surrogate is trusted and s is thus assessed with the surrogate, otherwise, the simulator is called. The union of the hyperspheres $\mathcal{H}$ with center s^* and radius ϵ in the decision space is defined as the confidence region CR:

$$CR = \begin{cases} \emptyset \text{ if } S^* = \emptyset \\ \cup_{s^* \in S^*} \mathcal{H}(s^*, \epsilon) \text{ otherwise} \end{cases}$$

The fact that the alternation criterion is tested at an individual-level and not at a population-level reduces greatly the probable damages sustained by an inaccurate surrogate that is to converge to a false Pareto front. Another advantage of this method is the saving in real evaluations provided by removing the need to compute the approximation error e_{NN}.

In [7], training is realized in batch mode with a training set composed of 50% of all available real evaluations. The underlying hypothesis is to build a strong surrogate on promising regions as the search moves forward. Besides, in order to grant the same importance to each training sample, training targets (outputs) are preprocessed by $\log(x+1)$ and

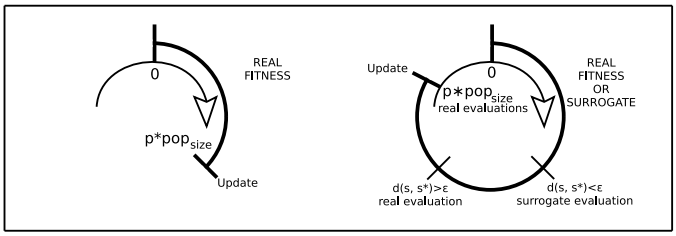

Fig. 2. Cycle of evaluations in $CR-EC$. The first cycle, on the left, is realized with real fitness function only. Once the surrogate is built, cycles are realized with both real fitness and surrogate function depending on the confidence region criterion.

normalized linearly in the range $[0, 1]$ thanks to minimum and maximum values taken from the same training set. Training features (inputs) are normalized linearly in the range $[0, 1]$. These assumptions are respected in this study to maintain comparison fairness. The number of real evaluations between two surrogate updates has to be analyzed too. In NSGA-II, completing a generation implies pop_{size} real evaluations, with pop_{size} being the size of the population. So in B_Q_n-EC, $p*pop_{size}$ real evaluations are performed between two updates. To that amount has to be added the small number of real evalutions needed to compute e_{NN} in $e_{NN}-EC$. In the confidence region AEC ($CR-EC$), it is decided to trigger the update step after $p*pop_{size}$ real evaluations.

V. Preliminary experiments

A preliminary experiment is set up so as to reproduce design engineering problem conditions. It is assumed that the real evaluation budget is small in comparison to the number of real evaluations needed, theoretically, to converge to the Pareto front. Benchmark problem ZDT4 [17] is chosen for its multi-modality aspect, a feature recognized to cause difficulty in MO. ZDT4 real fitness function, presented in Equation (2), is a continuous bi-objective function with 10 decision variables and admitting 21 local Pareto-optimal fronts. It has been shown that 30000 real evaluations are needed to converge to the Pareto front with NSGA-II [6], so the budget is fixed to 5000 here to respect the previous assumption.

$$\begin{aligned} F_1(x) &= x_1 \\ F_2(x) &= g(x)\left(1-\sqrt{\frac{x_1}{g(x)}}\right) \\ g(x) &= 91+\sum_{i=2}^{10}(x_i^2-10\cos(4\pi x_i)) \\ x_1 &\in [0,1] \text{ and } x_i \in [-5,5] \forall i \in \{2,\ldots,10\} \end{aligned} \tag{2}$$

The Hypervolume indicator H is the performance index used to compare NDFs in this study. Given a reference point O', H is the area generated by the set of solutions composing the NDF and O'. The higher the value of H, the better is the NDF as suggested in Figure 3 where H is displayed for a set of 3 solutions $\{s_1, s_2, s_3\}$. In this application, O' is set to $(1, 406)$ by roughly computing an upper bound for both objective functions. The interest of this index is to reflect the accuracy, the distribution and the spread of the solutions at once.

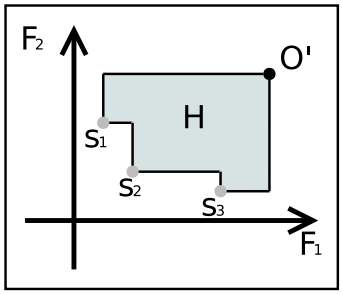

Fig. 3. Hypervolume indicator.

Optimization with EAs is a stochastic process because of the choice of the initial population and the frequency of applying reproduction operators. To get the trend of efficiency of the different evolution controls, several searches are launched, each with a fixed initial population for all the strategies. For each search, the initial population is constituted randomly over the decision space and, at the end, a point is given to the strategy that produces the best H value. The EA without MM and the three evolution control strategies (B_Q_n-EC, $e_{NN}-EC$ and $CR-EC$) are finally ranked.

The Pagmo [18] implementation of NSGA-II is parameterized as follows:

- Population size: 100
- Crossover probability: 0.9
- Distribution index for crossover: 10
- Mutation probability: 0.1
- Distribution index for mutation: 50

For B_Q_n-EC, p is set to 3 and q to 7 in accordance with the guidelines of [6]. To ensure comparison fairness in terms of number of samples in the training sets and to follow guidelines of [7], $p = 3$ and $q_{max} = 10$, $e_0 = 0.03$ and $M = 5$ in $e_{NN}-EC$. In $CR-EC$, ϵ is fixed to 1% of the maximum distance between two decision vectors. Specifications of the Scikit-Learn [19] implementation of ANN are as follows:

- Number of hidden layer: 1
- Number of hidden neurones: 10
- Optimizer: Stochastic Gradient Descent with Nesterov's momentum
- Activation function: sigmoid
- Batch size: training set size
- Learning rate: 0.2 (constant)
- Momentum factor: 0.2
- Weights and biases initialization: Glorot & al. method [20]
- Loss function: mean squared loss

Quality of training is conveyed by the coefficient of determi-

TABLE I
RESULTS OF THE FIRST TOURNAMENT WITH NSGA-II ALONE (ON THE LEFT) AND WITHOUT NSGA-II ALONE (ON THE RIGHT).

Strategy	Grade
NSGA-II	46/50
NSGA-II/B_Q_n-EC	0/50
NSGA-II/eNN-EC	1/50
NSGA-II/CR-EC	3/50

Strategy	Grade
NSGA-II/B_Q_n-EC	1/50
NSGA-II/eNN-EC	22/50
NSGA-II/CR-EC	27/50

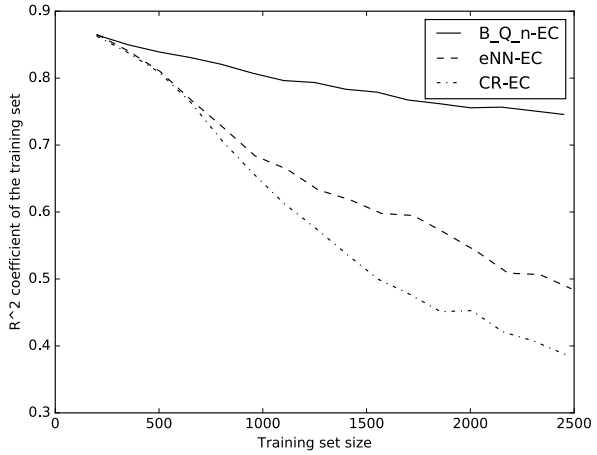

Fig. 4. Average training quality for the first tournament.

nation R^2 [21] computed over the training set:

$$R^2(y, y_{pred}) = 1 - \frac{\sum(y - y_{pred})^2}{\sum(y - E(y))^2} \quad (3)$$

The first tournament between the different methods is composed of 50 searches. ANN training is stopped when the coefficient of determination doesn't improve by at least 10^{-4} during 4 iterations. Results, shown on the left of Table I, indicate the superiority of NSGA-II without MM in this tournament. The explanation resides in the poor training quality of the ANN as shown in Figure 4. This same figure suggests that training is more complex in the case of $CR-EC$ than in other cases. Even if $CR-EC$ admits the worst training quality, it is the best method out of the three evolution controls as shown on the right of Table I. From Figure 5, it is worth noticing that $e_{NN}-EC$ limits the use of the inaccurate surrogate, with $q_i = 1$ on average, as expected.

To ensure fairness regarding the training quality and to keep a low training time, it is suggested to stop training as soon as R^2 reaches 0.1 for the second tournament. The results, presented in Table II, demonstrate the superiority of $e_{NN}-EC$ for this tournament. Since the training quality is maintained to a low level, the limitation of the inaccurate surrogate implied by $e_{NN}-EC$ gets the drop on the heuristic proposed in $CR-EC$ as it can be remarkably seen from cycle 14 in Figure 6.

Since the computational costs to attain a high R^2 score are prohibitive with ANN trained sequentially (more than an hour

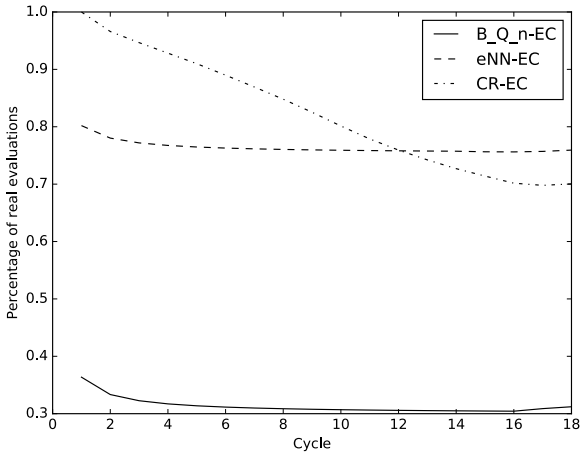

Fig. 5. Mean percentage of real evaluations along the cycles for the first tournament.

TABLE II
RESULTS OF THE SECOND TOURNAMENT WITH NSGA-II ALONE (ON THE LEFT) AND WITHOUT NSGA-II ALONE (ON THE RIGHT).

Strategy	Grade
NSGA-II	13/50
NSGA-II/B_Q_n-EC	7/50
NSGA-II/eNN-EC	19/50
NSGA-II/CR-EC	11/50

Strategy	Grade
NSGA-II/B_Q_n-EC	11/50
NSGA-II/eNN-EC	21/50
NSGA-II/CR-EC	18/50

for $CR-EC$ with 2500 training samples), Random Forests are considered as MM [22]. A forest of 10 regression trees each based on a sub-sampling created with replacement is used, given a training time of approximately 0.1 second. The quality of the MM is high on the training set in all the cases as shown in Figure 7. $eNN-EC$ trusts more the surrogate in this tournament as shown in Figure 8. Table III gives $CR-EC$

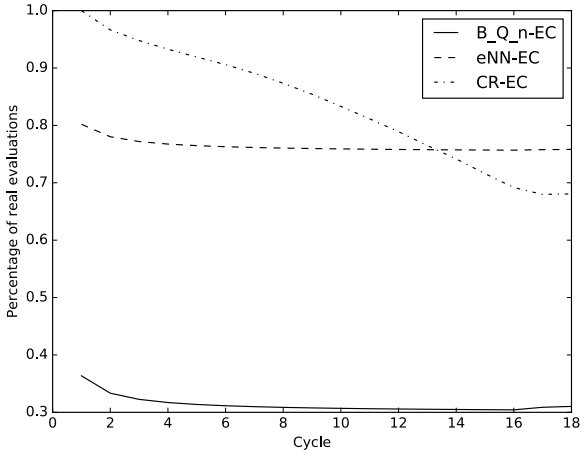

Fig. 6. Percentage of real evaluations along the cycles for the second tournament.

TABLE III
RESULTS OF THE THIRD TOURNAMENT WITH NSGA-II ALONE (ON THE LEFT) AND WITHOUT NSGA-II ALONE (ON THE RIGHT).

Strategy	Grade
NSGA-II	14/50
NSGA-II/B_Q_n-EC	14/50
NSGA-II/eNN-EC	6/50
NSGA-II/CR-EC	16/50

Strategy	Grade
NSGA-II/B_Q_n-EC	19/50
NSGA-II/eNN-EC	7/50
NSGA-II/CR-EC	24/50

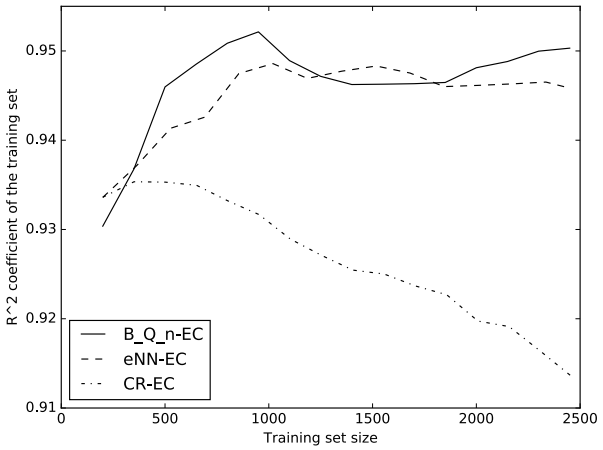

Fig. 7. Average training quality for the third tournament.

as the best method for the third tournament.

The curves of mean percentage of real evaluations over the cycles are the same in all three tournaments for $CR-EC$. This observation indicates a convergence toward regions containing the best known solutions.

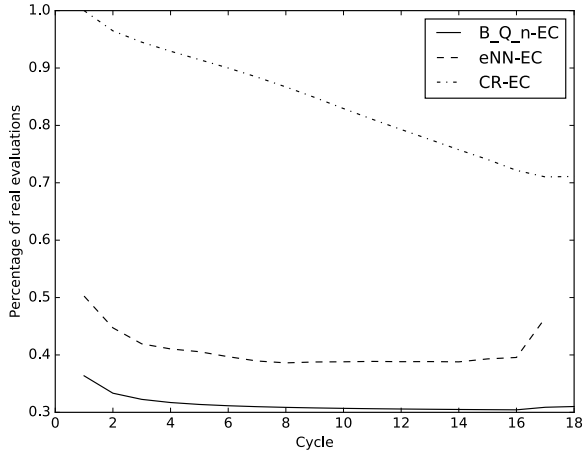

Fig. 8. Mean percentage of real evaluations along the cycles for the third tournament.

VI. CONCLUSION

The new confidence regions adaptive evolution control ($CR-EC$) proposed in this paper relies on a distance-based heuristic evolution control at an individual level during the evolution. If a candidate solution resides nearby a solution already evaluated and that has been valued to develop the surrogate then this latter is called, otherwise a complete simulation has to be launch. The preliminary experiments led in this study are set up in a way that imitates the processing of a complex engineering problem with a small real evaluations budget.

$CR-EC$ appears to be a promising surrogate-assisted MO method according to the preliminary results. However, further experiments have to be carried out, in particular, to analyze the dependence between the training quality and the value of the hyperspheres radius. It is also planed to use GPU-based parallel computing to improve training quality with ANN in a reasonable time. An engineering problem as well as other benchmark problems have to be considered and other EAs and MMs will be investigated to comparison.

REFERENCES

[1] Florent Duchaine, Thierry Morel, and Gicquel L.Y.M. Computational-fluid-dynamics-based kriging optimization tool for aeronautical combustion chambers. *AIAA Journal*, 47:631–645, 03 2009.

[2] E.G. Talbi. *Metaheuristics: From Design to Implementation*. Wiley Series on Parallel and Distributed Computing. Wiley, 2009.

[3] R. Duvigneau and M. Visonneau. Hybrid genetic algorithms and artificial neural networks for complex design optimization in cfd. *International Journal for Numerical Methods in Fluids*, 44(11):1257–1278, 2004. doi: http://dx.doi.org/10.1002/fld.688.

[4] Yaochu Jin. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2):61 – 70, 2011. doi: https://doi.org/10.1016/j.swevo.2011.05.001.

[5] Jose Hugo Barron-Zambrano Alan Díaz-Manríquez, Gregorio Toscano and Edgar Tello-Leal. A review of surrogate assisted multiobjective evolutionary algorithms. *Computational Intelligence and Neuroscience*, 2016:14, 2016. doi: https://doi.org/10.1155/2016/9420460.

[6] Kalyanmoy Deb and Pawan Nain. An Evolutionary Multi-objective Adaptive Meta-modeling Procedure Using Artificial Neural Networks. In Shengxiang Yang, Yew-Soon Ong, and Yaochu Jin, editors, *Evolutionary Computation in Dynamic and Uncertain Environments*, volume 51, chapter 13, pages 297–322. Springer, Berlin, Heidelberg, 2007. doi: http://dx.doi.org/10.1007/978-3-540-49774-5_13.

[7] Antonio Gaspar-Cunha and Armando Vieira. A multi-objective evolutionary algorithm using neural networks to approximate fitness evaluations. *International Journal of Computers, Systems and Signals*, 6:18–36, 01 2005.

[8] Juliane Müller and Christine A. Shoemaker. Influence of ensemble surrogate models and sampling strategy on the solution quality of algorithms for computationally expensive black-box global optimization problems. *Journal of Global Optimization*, 60(2):123–144, Oct 2014. doi: https://doi.org/10.1007/s10898-014-0184-0.

[9] Carlo Poloni, Andrea Giurgevich, Luka Onesti, and Valentino Pediroda. Hybridization of a multi-objective genetic algorithm, a neural network and a classical optimizer for a complex design problem in fluid dynamics. *Computer Methods in Applied Mechanics and Engineering*, 186(2):403 – 420, 2000. doi: https://doi.org/10.1016/S0045-7825(99)00394-1.

[10] A. Syberfeldt, H. Grimm, A. Ng, and R. I. John. A parallel surrogate-assisted multi-objective evolutionary algorithm for computationally expensive optimization problems. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 3177–3184, June 2008. doi: https://doi.org/10.1109/CEC.2008.4631228.

[11] Miha Mlakar, Dejan Petelin, Tea Tušar, and Bogdan Filipič. Gp-demo: Differential evolution for multiobjective optimization based on gaussian process models. *European Journal of Operational Research*, 243(2):347 – 361, 2015. doi: https://doi.org/10.1016/j.ejor.2014.04.011.

[12] S. Huband, P. Hingston, L. Barone, and L. While. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506, Oct 2006. doi: https://doi.org/10.1109/TEVC.2005.861417.

[13] T. Okabe, Y. Jin, and B. Sendhoff. A critical survey of performance indices for multi-objective optimisation. In *Evolutionary Computation, 2003. CEC '03. The 2003 Congress on*, volume 2, pages 878–885 Vol.2, Dec 2003. doi: https://doi.org/10.1109/CEC.2003.1299759.

[14] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, Apr 2002. doi: https://doi.org/10.1109/4235.996017.

[15] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359 – 366, 1989. doi: https://doi.org/10.1016/0893-6080(89)90020-8.

[16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[17] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evol. Comput.*, 8(2):173–195, June 2000. doi: http://dx.doi.org/10.1162/106365600568202.

[18] Francesco Biscani, Dario Izzo, and Marcus Märtens. esa/pagmo2: pagmo 2.6, November 2017. doi: https://doi.org/10.5281/zenodo.1054110.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[20] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

[21] A. Colin Cameron and Frank A.G. Windmeijer. An r-squared measure of goodness of fit for some common nonlinear regression models. *Journal of Econometrics*, 77(2):329 – 342, 1997. doi: https://doi.org/10.1016/S0304-4076(96)01818-0.

[22] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, October 2001. doi: https://doi.org/10.1023/A:1010933404324.