Surrogate-Based MOEA/D for Electric Motor Design With Scarce Function Evaluations

Rodrigo C. P. Silva, Min Li, Tanvir Rahman, David A. Lowther

Department of Electrical and Computer Engineering, McGill University, Montreal, QC, Canada

This paper proposes a surrogate-assisted multiobjective evolutionary algorithm based on decomposition (sMOEA/D) for the design of electric motors. The idea is to improve the surrogate gradually during the optimization. Simulation results show that the proposed method is competitive with state-of-the-art multiobjective optimization algorithms needing only a small number of function evaluations.

Index Terms—**Electrical machine design, interior permanent magnet motor, multiobjective optimization, surrogate models.**

I. INTRODUCTION

MULTIOBJECTIVE evolutionary algorithms (MOEAs) have been widely applied to the design of electric motors. However, when time-consuming finite element (FE) based simulations are used to compute objective function values, MOEAs may sometimes, become impractical. In this context, surrogate models are often used to reduce the optimization execution time (number of FE simulations).

The rationale of surrogate-based optimization is to gradually improve the surrogate by adding infill points in the regions of interest [1]. In single objective optimization, methods can be used to improve the accuracy of a surrogate in a small region [2]. However, in multiobjective optimization (MOO), since the Pareto-optimal solutions may not fall into the same small region, the problem of determining where to improve the surrogate becomes more complex.

Currently, most of the research on surrogate-based MOO has been devoted to the adaptation of single-objective infill criteria. Two main approaches can be identified in the literature: scalarization [2] and expected hyper-volume improvement (EHI) [3]. The first approach independently solves scalarized versions of the MOO problem using a surrogate-based single-objective algorithm. The second extends the idea of expected improvement of [1], and given the prediction uncertainty, it samples the point, which maximizes the EHI, at each iteration. The scalarization approaches suffer from scalability issues as the number of objectives increases. The EHI-based approaches suffer from the fact that they can only sample one point per iteration and the complexity of computing the EHI with a large number of objectives. Thus, these methods tend to require a very high number of iterations to obtain a good representation of the front which can be undesirable in terms of wall clock time [4].

In this paper, we propose a surrogate-based MOEA approach which mitigates the issues of the aforementioned methods. The proposed approach: 1) avoids the individual solution of different (possibly many) scalarized problems and 2) permits the selection of multiple infill points per iteration.

The framework employs the state-of-the-art MOEAs based on decomposition (MOEA/D) [5] and four different criteria for multiple infill point selection in order to balance exploration and exploitation during the optimization. Finally, the proposed algorithm is applied to the design optimization of an interior permanent magnet (IPM) machine, outperforming the regular MOEA/D when they are given the same computational budget.

II. MULTIOBJECTIVE OPTIMIZATION

A standard multiobjective optimization problem (MOOP) can be defined as

$$\begin{aligned} &\min \quad \mathbf{f}(\mathbf{x}) = \{f_1, f_2, \ldots, f_m\} \\ &\text{s.t.} \quad \mathbf{x} \in \mathrm{F} \end{aligned} \tag{1}$$

where, F is the feasible decision space and $\mathbf{f}: F \rightarrow R^m$ consists of m real-valued objective functions.

The most commonly used methods for solving multiobjective problems are evolutionary algorithms with a Pareto-based fitness assignment such as the Nondominated Sorting Genetic Algorithm II (NSGA-II) [6] which is also implemented as the MATLAB standard multiobjective optimization method. In spite of their popularity, the performance of these algorithms deteriorates as the number of objectives increases [7]. In this context, MOEA/D decomposes the MOO into a number of scalar problems, but instead of optimizing each one separately, it uses information from the neighboring problems and optimizes them simultaneously. Due to this peculiar search mechanism, MOEA/D tends to converge faster to the Pareto-front and to obtain more evenly distributed solutions than other evolutionary algorithms [5]. Currently, MOEA/D is the benchmark algorithm for multiobjective optimization.

III. SURROGATE-ASSISTED MOO

A. Surrogate Modeling

Computation-intensive design problems are common in manufacturing industries. Nevertheless, the computation burden caused by expensive analysis and simulation processes hinders the application of the majority of optimization methods [8]. To address such a challenge, surrogate models are

Manuscript received November 20, 2016; revised January 15, 2017; accepted January 24, 2017. Date of publication February 1, 2017; date of current version May 26, 2017. Corresponding author: D. A. Lowther (e-mail: david.lowther@mcgill.ca).

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.2017.2661987

often used. Available surrogates may vary from simple low-order polynomial models to more complex artificial neural networks. Among them, Kriging [1] has been extensively studied and applied in a large variety of problems. Kriging has very mild assumptions about the shape of the function being approximated. In addition to each prediction, $\hat{y}$, Kriging also provides an estimation of the prediction error, $\hat{s}$. This information can be extremely useful and has been incorporated into a number of surrogate-based optimization methods [2]–[4], [9].

B. Surrogate-Assisted MOEAs

The simplest way to couple MOEAs with surrogate models is to fit the model with sample data and run the MOEA directly on the surrogate [10]. The success of this approach, however, is highly dependent on the quality of the surrogate model which is usually difficult to assess [8]. Thus, normally, the surrogate model is gradually updated throughout the optimization procedure to make it more accurate in the regions of high-quality solutions and to avoid spending computational budget in the regions of low-quality solutions.

When Kriging (or other Gaussian processes) is used, an expected improvement with respect to the current estimate of the optimum can be computed using $\hat{y}$ and $\hat{s}$. In this way, the surrogate is continuously updated, where the highest value of the expected improvement is found. Based on this idea, Jones *et al.* [1] proposed the efficient global optimization (EGO) method, which is now one of the state-of-the-art surrogate-based optimization methods. In [2] and [3], EGO is extended to solve MOO problems; however, these methods suffer from some performance drawbacks as mentioned in Section I. Besides, ideally, surrogate-assisted MOEAs should be independent of the choice of surrogate models in order to balance cost and accuracy.

IV. Surrogate-Assisted MOEA/D

In order to overcome the aforementioned issues, in this section we present an sMOEA/D algorithm that: 1) utilizes one of the best MOEAs (see Section II); 2) does not need to solve scalarized problems independently; and 3) allows for the selection of multiple infill points per iteration.

A. sMOEA/D Algorithm

1) Create an archive A by sampling the design space using a Latin-Hypercube design of size n.
2) Fit a Kriging surrogate model with A for each objective function.
3) Obtain a Pareto-front PF by running MOEA/D with the Kriging models.
4) Select k infill points from the PF using a given criterion.
5) Add the infill points to A.
6) Repeat from step two until the preset computational budget is reached.
7) Return the nondominated solutions in A.

In the next section, we propose and discuss four selection criteria for infill points.

B. Selection Criteria

1) Distance in the Search Space: Let $PS = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ be the final set of solutions found by MOEA/D with the surrogate and $A = [\mathbf{a}_1, \ldots, \mathbf{a}_L]$ be the current solution archive. The distance of $\mathbf{x}_i$ to the archive A is defined as

$$\min_{j \in \{1,\ldots,L\}} \|\mathbf{x}_i - \mathbf{a}_j\|_2 \tag{2}$$

If $k = 1$, the point that maximizes the function defined in (2) is selected. If $k > 1$, the distances are re-computed after each addition.

This criterion is based on the idea that the surrogate model is accurate near already sampled points. Hence, the points that are distant from the archive have a higher contribution to the accuracy of the surrogate model in the regions with potential high-quality solutions and the diversity of the Pareto-front.

2) Density in the Objective Space: Let $A_{\mathrm{PF}} = [\mathbf{f}_1, \ldots, \mathbf{f}_L]$ be the current nondominated set of the objective function values of the current A, the distance of $\mathbf{x}_i$ to A_{PF} in the objective space is defined as

$$\min_{j \in \{1,\ldots,L\}} \|\mathbf{f}(\mathbf{x}_i) - \mathbf{f}_j\|_2 \tag{3}$$

Following the same rationale as that for SD, the points which maximize the function defined in (3) are selected. It is important to highlight that $\mathbf{f}(\mathbf{x}_i)$ cannot be dominated by points in A_{PF}. If there are not enough candidates the infill set is filled with random points.

This approach can be seen as more greedy since it does not accept solutions for which the predicted performance is dominated by the current A_{PF}.

3) Random Selection: Here k points are randomly selected from PS. This method simply works as a baseline for the other selection criteria.

4) Estimated Error: Kriging provides the estimated error $\hat{s}$ of the predicted value. Let $\hat{s}(\mathbf{x}_i) = \{\hat{s}_1(\mathbf{x}_i), \ldots, \hat{s}_m(\mathbf{x}_i)$ be the error estimated for each objective and $\hat{s}_{\mathrm{jmax}}$ is the maximum observed error in PS for objective j. The points with highest total estimation error (HTEE) are selected. The HTEE is defined as

$$\mathrm{HTEE}(\mathbf{x}_i) = \sum_{j=1}^{m} \frac{\hat{s}_j(\mathbf{x}_i)}{\hat{s}_{\mathrm{jmax}}}. \tag{4}$$

V. Validation of the Algorithm

The algorithm presented in Section IV has three parameters: 1) the selection criterion for infill points; 2) the number k of infill points selected by iteration; and 3) the size n of the initial Latin-Hypercube sampling. In order to analyze the effect of these parameters and, at the same time, keep the number of experiments manageable, we break the full experiment into two smaller ones.

The goal of Experiment 1 is to analyze the different selection criteria and their interaction with k. The goal of Experiment 2 is to understand the impact of n (initial sample size) on the optimization in conjunction with the best selection criterion obtained in the first experiment. Both experiments are performed on the two-objective, eight-variable versions of the Zitzler, Deb, and Thiele (ZDT) benchmark problems [11].

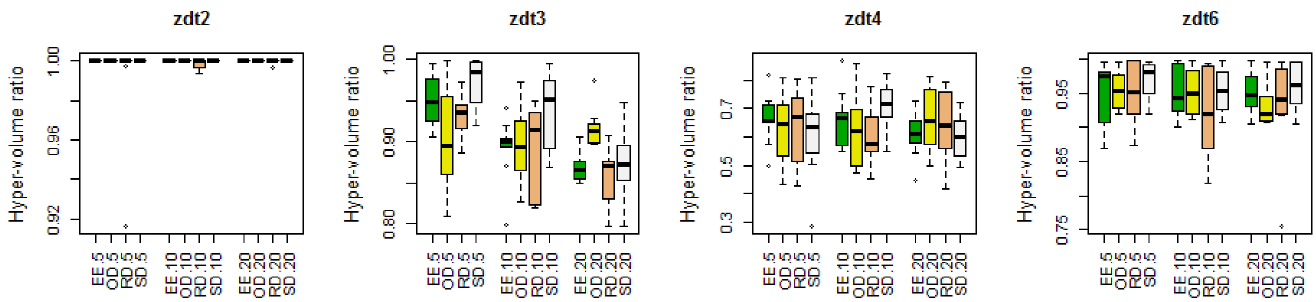

Fig. 1. Box-plots of the relative hyper-volume obtained by different sMOEA/D versions with varying selection criteria (SD) and number of infill points per iteration (k). The closer to 1, the closer the results were from the obtained analytical solution.

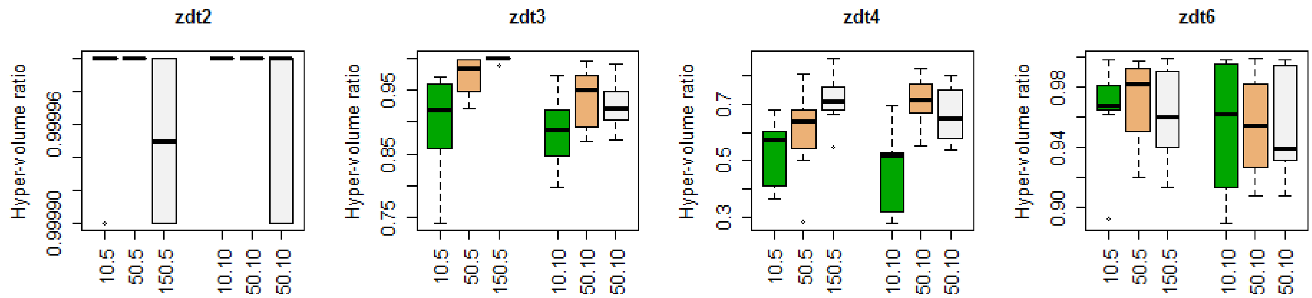

Fig. 2. Box-plots of the relative hyper-volume obtained by different sMOEA/D versions with varying number of infill points per iteration (k) and initial samples sizes (n). The closer to 1, the closer the results were from the obtained analytical solution.

TABLE I
EXPERIMENT 1 SET-UP

PARAMETER	VALUES
n	50
Selection criteria (SC)	EE, OD, SD, RD
k	5, 10, 20
Maximum number of function evaluations	300
MOEA/D Population Size	200
MOEA/D Number of generations	1000

TABLE II
EXPERIMENT 2 SET-UP

PARAMETERS	VALUES
n	10, 50, 150
Selection Criteria (SC)	SD
k	5,10

These test functions are all continuous and present well-defined local minima, which based on [10] and [12], are believed to share the same structure with common objective functions in electrical machine design.

A. Experiment 1

Table I presents the experimental settings of Experiment 1. For the other MOEA/D parameters see [5]. Fig. 1 shows the hyper-volumes (normalized by the analytical solution) obtained over ten independent runs of each version of the algorithm. The x-axis depicts each version of the algorithm in the format SC.k. It can be observed that, on average, SD.5 outperformed the other methods on zdt3 and 6; and SD.10 outperformed the other methods on zdt4. zdt2 is relatively simple, therefore all algorithms perform equally well on it.

It is interesting to notice the performance of the algorithms with 20 infill points, on both zdt3 and zdt4. The performance of the infill methods (except OD.20) deteriorated with 20 samples per iteration, possibly because, given that the number of function evaluations is fixed, fewer iterations can be performed. OD.20, on the other hand, showed a slight improvement compared to OD.5 and OD.10. This is expected from OD as a greedy approach, which might exhibit better performance when the execution time is very limited.

B. Experiment 2

Table II presents the experimental settings of Experiment 2. MOEA/D parameters are the same as in Experiment 1. Fig. 2 shows the hyper-volumes (normalized by the analytical solution) obtained over 10 independent runs. The x-axis depicts each version of the algorithm in the format $n.k$.

It may be observed that the best combination of parameters depends on the specific problem. Nevertheless, given the obtained results, the $n = 150$, $k = 5$ combination seems to be the most reasonable choice. This implies that: first, an adequate number of samples is demanded to build a nontrivial initial surrogate (ten samples were not enough); and second, it is critical that the algorithm has time to perform update iterations (as 150.10 versions do not exploit the high-quality regions thoroughly). For zdt6, the differences among all versions are relatively small, from 92% to 99% of the reference hyper-volume.

To sum up, by carefully choosing the selection criterion and other algorithm parameters, the sMOEA/D performs reasonably well on all the benchmark problems.

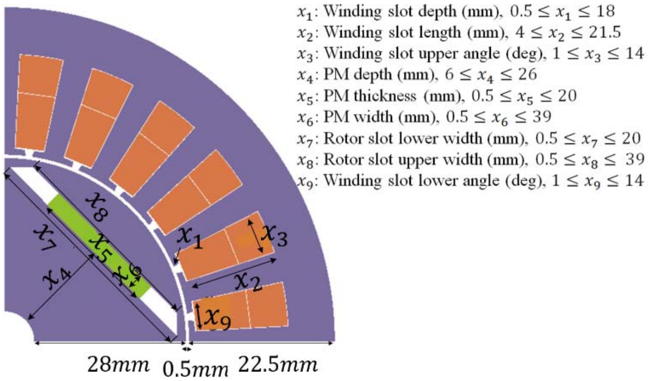

Fig. 3. IPM model and design variables.

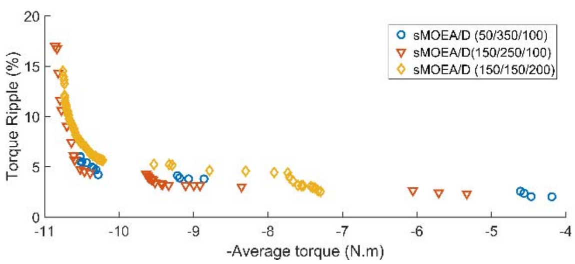

Fig. 4. Nondominated fronts obtained by sMOEA/D for the IPM design problem. In parenthesis, the number of function evaluations for initial sampling, update iterations, and validation of the final population, respectively.

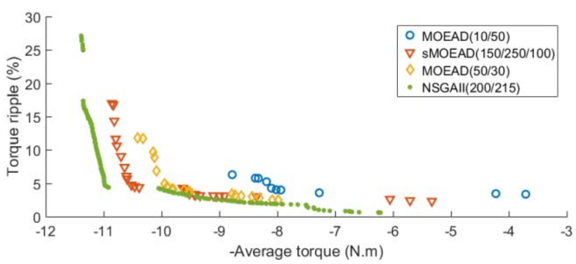

Fig. 5. Nondominated fronts obtained by sMOEA/D and MOEA/D for the IPM design problem.

VI. Machine Design Application

In this section, sMOEA/D is applied to the design optimization of an IPM motor (depicted in Fig. 3) where the average torque is maximized and the torque ripple is minimized. More details on the IPM design problem can be found in [7]. The total computational budget was set to 500 FE simulations.

In Figs. 3 and 4 versions of the sMOEA/D ($SC = SD$, $k = 5$) with varying distributions of function evaluations for initial sampling, update iterations, and validation of the final population, are compared. The results confirm some of the observations made in Section V, and show the importance of the balance between an adequate initial sample and the number of update iterations. Again, as expected from results shown in Section V, the sMOEA/D (150/250/100) was the best performing algorithm in terms convergence and spread of the non-dominated front.

Given that one FE simulation of the presented IPM takes $\sim$13 s and one iteration of the sMOEA/D takes $\sim$150 s, in Fig. 5 we compare the sMOEA/D (150/250/100) with the regular MOEA/D [5] in two situations: 1) the number of function evaluations is the same (MOEA/D (10 individuals/50 generations) and 2) the maximum execution time is the approximately the same (MOEA/D (50 individuals/ 30 generations). In both cases, the sMOEA/D was able to find better quality non-dominated sets which show the superiority of the presented approach. As reference, we also show the results of NSGAII [13] (200 individuals/ 215 generations).

VII. Conclusion

An efficient algorithm, based on surrogate models and MOEA/D for MOOPs with scarce function evaluations has been proposed. Given the same time and computational budgets, it was able to produce a nondominated set which dominated most, if not all, the nondominated solutions found by the regular approaches.

Experiments using analytical test functions showed that selecting infill points that are the farthest from the archive (SD criteria) is a good infill strategy. Nevertheless, if wall clock time is of extreme importance (i.e., a lower number of iterations is allowed), a more greedy strategy, such as the density in the objective space (OD), should be considered.

Compared to the other surrogate-assisted MOEAs mentioned in Section III-B, the proposed framework is independent of the choice of surrogate models. Technically, different types of surrogate models (either more accurate or more efficient) can be fit to different objective functions.

References

[1] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *J. Global Optim.*, vol. 13, no. 4, pp. 455–492, 1998.

[2] G. I. Hawe and J. K. Sykulski, "A scalarizing one-stage algorithm for efficient multi-objective optimization," *IEEE Trans. Magn.*, vol. 44, no. 6, pp. 1094–1097, Jun. 2008.

[3] M. T. M. Emmerich, K. C. Giannakoglou, and B. Naujoks, "Single- and multiobjective evolutionary optimization assisted by Gaussian random field metamodels," *IEEE Trans. Evol. Comput.*, vol. 10, no. 4, pp. 421–439, Aug. 2006.

[4] F. A. C. Viana, R. T. Haftka, and L. T. Watson, "Efficient global optimization algorithm assisted by multiple surrogate techniques," *J. Global Optim.*, vol. 56, no. 2, pp. 669–689, 2013.

[5] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Jun. 2007.

[6] 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.

[7] R. Silva, A. Salimi, M. Li, A. R. R. Freitas, F. G. Guimarães, and D. A. Lowther, "Visualization and analysis of tradeoffs in many-objective optimization: A case study on the interior permanent magnet motor design," *IEEE Trans. Magn.*, vol. 52, no. 3, pp. 1–4, Mar. 2016.

[8] G. G. Wang and S. Shan, "Review of metamodeling techniques in support of engineering design optimization," *J. Mech. Design*, vol. 129, no. 4, pp. 370–380, 2007.

[9] G. I. Hawe and J. K. Sykulski, "A hybrid one-then-two stage algorithm for computationally expensive electromagnetic design optimization," *COMPEL Int. J. Comput. Math. Electr. Electron. Eng.*, vol. 26, no. 2, pp. 236–246, 2007.

[10] M. H. Mohammadi, T. Rahman, R. Silva, M. Li, and D. A. Lowther, "A computationally efficient algorithm for rotor design optimization of synchronous reluctance machines," *IEEE Trans. Magn.*, vol. 52, no. 3, pp. 1–4, Mar. 2016.

[11] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, 2000.

[12] N. Bianchi, S. Bolognani, and F. Luise, "Potentials and limits of high-speed PM motors," *IEEE Trans. Ind. Appl.*, vol. 40, no. 6, pp. 1570–1578, Nov. 2004.

[13] *MATLAB Version 8.5.0.197613 (R2015a)*, The MathWorks Inc., Natick, MA, USA, 2015.