A Fuzzy Niching Evolution Strategy for Multiobjective Optimization

P. Alotto*, M. Jaindl†, A. Köstinger†, C. Magele†, and W. Renhart†

*Dipartimento di Ingegneria Elettrica, Università degli Studi di Padova – Italy
†Institut für Grundlagen und Theorie der Elektrotechnik, TU Graz – Austria

***Abstract*—Evolution strategies are known to be powerful stochastic metaheuristics for the solution of complex single-objective engineering problems. In this paper a fuzzy weighting of objectives combined with niching techniques is used within the framework of a classical evolution strategy in order to make the optimizer suitable for the solution of multiobjective problems. The proposed approach is described and its performance is compared with a Pareto-enabled Differential Evolution technique on an eddy current shielding benchmark problem.**

***Index Terms*—Evolution strategy, Differential evolution, Multiobjective optimization, Fuzzy logic, Niching**

I. Introduction

Evolution Strategies (ES) rely on a number of simplified features of biological evolution like reproduction, mutation, competition and selection. Classical Evolution Strategies (ES) are able to find a good local if not the global solution of single-objective optimization problems. If more objectives have to be targeted simultaneously it has been shown that it is advantageous to merge all objectives into a single value using fuzzy membership functions and appropriate inference rules inside a classical ES [1]. However, introducing a niching technique it is possible to detect, store and postprocess additional local solutions within a single optimization run [2]. This results, in general, in several solutions which are very close to a limited portion of the Pareto front depending on the weights used in the respective inference rule. Different weighted sums, however, cover different parts of the Pareto front, which has already been shown using a sequential approach [3]. To evaluate a larger part of the Pareto front within a single optimization run it is therefore suggested to process more than one objective function with variable weights in parallel threads which are readily provided by modern multicore CPU architectures. In order to further reduce the number of function calls, configurations from one thread can become candidate configurations in another thread. Furthermore, all used configurations and the corresponding solutions are stored in an archive. If one of the archived configurations is within a certain range of a new candidate configuration, its objective function values are re-used, which further improves the efficiency of the proposed method.

II. Parallel $(\mu/\rho, \lambda)$ Evolution Strategy

In the proposed implementation, in contrast to more classical versions of ES, several populations following different objective functions are evolving in parallel. Only a certain percentage of individuals for the next generation of a single thread is set up in the classical way (recombination of parental configurations and mutation), while the rest is taken from all other threads and directly inserted into the population without any computational cost. Additionally, all newly generated configurations are compared against all configurations produced so far. If a newly created one is located very close to such an archived configuration, it will be replaced by the old values thus saving computational effort. Finally, this parallel version of ES yields one optimal solution for each thread and its corresponding objective function.

III. Fuzzy weighting of objectives

After all objectives have been evaluated for all members of the population, the current value of each objective of a single configuration is normalized to values between 0 and 1 (providing a so-called level of satisfaction) by means of appropriate types of nonlinear fuzzy membership functions [1]. Then a weighted sum of all levels of satisfaction is computed in order to assess the quality of the configuration and to make implicit or explicit selection (two main operators of ES) possible. Additionally, using different weights one can put more or less emphasis on one or the other objective. The weighting is in this case facilitated by the implicit scaling provided by the fuzzy membership functions.

IV. Cluster sensitive recombination

To find more than one local solution within the individual threads each population is clustered into niches. The following recombination is performed with higher probability within a niche than beyond niche boundaries. As soon as an isolated niche is identified, its best solution is stored. The aproximate number κ of niches can be automatically estimated from several cluster metrics [3]. This enables the $[\kappa(\mu/\rho, \lambda)]$ ES to adjusts its population size dynamically and, in general, to decrease it gradually. In a post processing step all these solutions are evaluated and clustered again. The result is a more or less large number of local solutions which are in general located very close to (or even on) the Pareto optimal front. Furthermore, numerical experiments have demonstrated that this kind of recombination results in the $[\kappa(\mu/\rho, \lambda)]$ ES having a more global convergence behaviour than the $(\mu/\rho, \lambda)$ ES [3].

V. Multiobjective differential evolution

Differential Evolution (DE) [4] is an evolutionary population-based optimization metaheuristic which is characterized by a particular method for the generation of new candidate solutions and by the use of a very greedy selection scheme. Furthermore, the algorithms lends itself very well to the implementation of self-adaptive features so that the resulting methods are essentially parameter-free. Multiobjective version of DE can be very easily constructed with slight modifications to the underlying single objective algorithm, following for example the ideas proposed in [5]. In such approaches at each generation the original population is subject to the classical DE mutation and crossover operators and the $2n$ individuals of the original population and the one resulting from mutation and crossover are subject to non-dominated, least-crowded sorting according to the NSGA-II [6] philosophy and the best n individuals are promoted to the next generation. With such minor modifications DE becomes capable of finding solutions on the Pareto front while preserving its extremely greedy character. DE is used in this paper to provide an independent method for the computation of the Pareto front.

VI. Numerical results

The chosen benchmark problem refers to the optimization of a simplified magnetic shunting configuration of a power transformer. The geometry of the problem is sketched in Fig. 1 together with the material parameters used in the model. The problem is characterized by six box-constrained geometric dimensions, namely the total height h of the shunts in the range [20,100] mm and the widths w of the individual layers in the range [50,280] mm. The problem consists in minimizing the eddy current losses in the tank while at the same minimizing the area (volume) of the magnetic shunts.

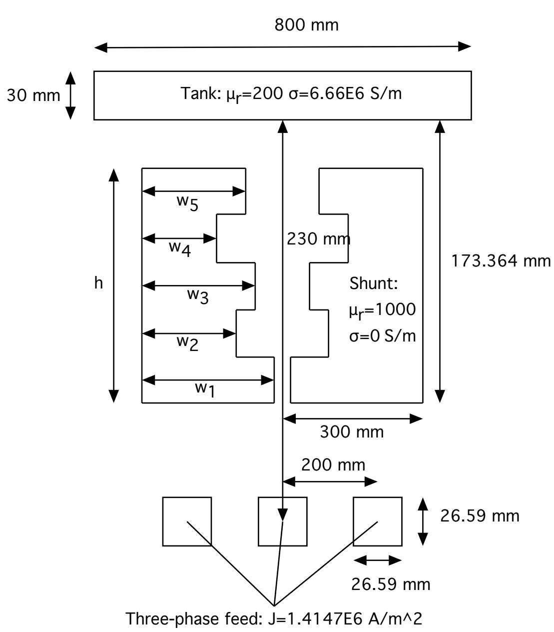

Fig. 1. Geometry and degrees of freedom of the benchmark problem.

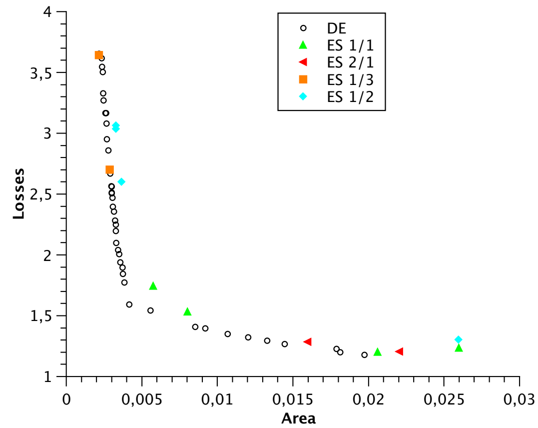

Fig. 2. Comparison of DE and various Fuzzy-ES wight combinations

Optimization results, shown in Fig. 2, demostrate that the proposed approach is indeed capable of finding groups of solutions which well approximate the true Pareto front, which is computed with DE.

In the extended version of the paper a more complicated 3-objective version of the same problem will be presented. This extended version of the benchmark will include a further copper shielding layer.

VII. Conclusions

This paper explores the possibility of modifying a standard Evolution Strategy algorithm in order to efficiently solve multiobjective optimization problems without resorting to the Pareto approach. The algorithm features a fuzzy weighting of objectives and niching strategies and makes use of the multi-threading features of modern microprocessor architectures. In the extended version of the paper all algorithmic details will be presented in detail and a more complicated version of the benchmark will be solved.

References

[1] P. Alotto, A. Bertoni, G. Molinari, M. Nervi, B. Brandstätter, Ch. Magele, K.R. Richter, C. Ragusa, M. Repetto, "A Combined Approach for the Stochastic Optimisation of Multiminima Problems Using Adaptive Fuzzy Sets and Radial Basis Functions," IEEE Trans. on Magnetics, vol. 34, no. 5, pp. 2837 - 2840, 1998.

[2] O. Aichholzer, F. Aurenhammer, B. Brandstätter, T. Ebner, H. Krasser, Ch. Magele, M. Mühlmann, and W. Renhart, "Evolution Strategy and Hierarchical Clustering," IEEE Trans. on Magnetics, vol. 38, no. 2, pp. 1041-1044, 2002.

[3] C. Magele, A. Köstinger, M. Jaindl, W. Renhart, B. Cranganu-Cretu and J. Smajic, "Niching Evolution Strategies for Simultaneously Finding Global and Pareto Optimal Solutions," IEEE Trans. on Magnetics, vol. 46, no. 8, pp. 2743-2746, 2010.

[4] R. Storn and K. Price, "Differential Evolution A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," Journal of Global Optimization, vol. 11, Dec. 1997, pp. 341-359.

[5] T. Robic and B. Filipic, "DEMO: Differential Evolution for multiobjective optimization", In Proceedings of the 3rd International Conference on Evolutionary MultiCriterion Optimization (EMO 2005), 2005, pp. 520–533.

[6] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," IEEE Trans. on Evolutionary Computation, vol.6, no.2, Apr. 2002, pp.182-197.