Two-layered Surrogate Modeling for Tuning Optimization Metaheuristics

Günter Rudolph[1], Mike Preuss[2], and Jan Quadflieg[1]

Technische Universität Dortmund, Lehrstuhl für Algorithm Engineering
44221 Dortmund, Germany
{guenter.rudolph,mike.preuss,jan.quadflieg}@uni-dortmund.de

1 Introduction

The problem of detecting suitable parameters for metaheuristic optimization algorithms is well known long since. As these non-deterministic methods, e.g. *evolution strategies* (ES) [1], are highly adaptible to a specific application, detecting good parameter settings is vital for their success. Performance differences of orders of magnitude (in time and/or quality) are often achieved by means of automated tuning methods. In the last years, several of these have been suggested, and many incorporate surrogate models for the algorithm parameter space. Although tuning methods are reliable tools to build specific optimization algorithms by modifying the parameters of canonic ones, their use is somewhat restricted to relatively cheap objective functions (in terms of computational cost). As a huge number of algorithm runs is necessary, they are simply not applicable when the evaluation time of the objective function increases above a certain level. If the objective function is too expensive for applying tuning methods directly, one can resort to simpler approaches:

1. One may try to find a suitable parameter setting for a canonical metaheuristic by means of a very simple randomized or space-filling design, or
2. one may give up on the parameter optimization approach and optimize the problem directly with a heuristic or metaheuristic that incorporates surrogate models itself.

The latter methodology is employed e.g. by EGO [2], but also by surrogate model enhanced metaheuristics like the *model-assisted evolution strategy* (MAES) [3]. Whereas tuning methods utilize many runs of standard algorithms which are performed on the original optimization problem and do not model it, MAES and similar algorithms model the optimization problem 'on-the-fly' but do not adapt the algorithm itself. There is no reason to believe that these two approaches could not go well together and thus make tuning applicable to more expensive optimization problems than before. This is where our new method comes into play.

2 Two-layered model-supported optimization

The basic idea of our approach is to start an algorithm tuning process with a minimal space-filling design and to use the obtained objective function samples to build a first-layer surrogate (kriging) model on which the algorithm tuning is then continued. As soon as the tuning process converges, validation runs are performed to test the tuned algorithm on the real objective function. Thereby, more samples become available for updating the first-layer model. The tuning process may then be continued on the updated model and the resulting algorithm configuration validated again. This loop shall be terminated if either a predefined budget of objective function evaluations is used up or no better algorithm configurations are obtained from the tuning any more.

We make the assumptions that the algorithm tuning problem is easier to solve than the optimization problem itself and that an surrogate model represents the real problem good enough to allow for tuning the optimization algorithm to perform well on the original problem. The first assumption is supported by the smaller dimensionality of the tuning problem (usually around 5, many real-world optimization problems have between 10 and 30 variables), and prior knowledge about the mechanisms of the optimization algorithms. The second assumption has to be tested experimentally. It is clear that although we head for greatly reduced tuning times, our approach is suitable only if objective function evaluations are on the order or minutes at most; otherwise, tuning is not possible any more and few runs of any optimization algorithm have to suffice.

For the implementation of our method, we build on the *sequential parameter optimization* (SPO) [4] as tuning method. It employs kriging as means to setup a second-layer surrogate model in the algorithm parameter space and has been successfully applied on several metaheuristic algorithms recently. The canonical metaheuristic optimization algorithms utilized here are the self-adaptive evolution strategy [1] and the *covariance matrix adaptation evolution strategy* (CMA-ES) in the variant suggested in [5]. Together with the kriging model on the original problem, we have two surrogate models, and either one is used to feedback into the other. We thereby strive for maximum exploitation of the available objective function evaluations to obtain a suitable optimization algorithm which is then also able to cope with similar problem instances.

3 A Real-World Test Problem

As a suitable test problem, we employ the optimization of a relatively new ship propulsion system (a linearjet, 1, right) which possesses 20 design variables. It consists of a tube with a rotor and a stator, and several lengths, angles and thicknesses can be variated. Our objective function is a very basic fuild dynamic simulation of a linearjet that takes about 3 minutes to compute, and the task is to reduce cavitation at a predefined efficiency. A MAES [3] has been applied to a simpler form (less variables) of the problem with limited success [6]. This may be due to a very rugged search space with many plateaus, cliffs and bumps. As indicated by a random sample around a good search point (1, left), the quality to distance correlation looks fairly unstructured.

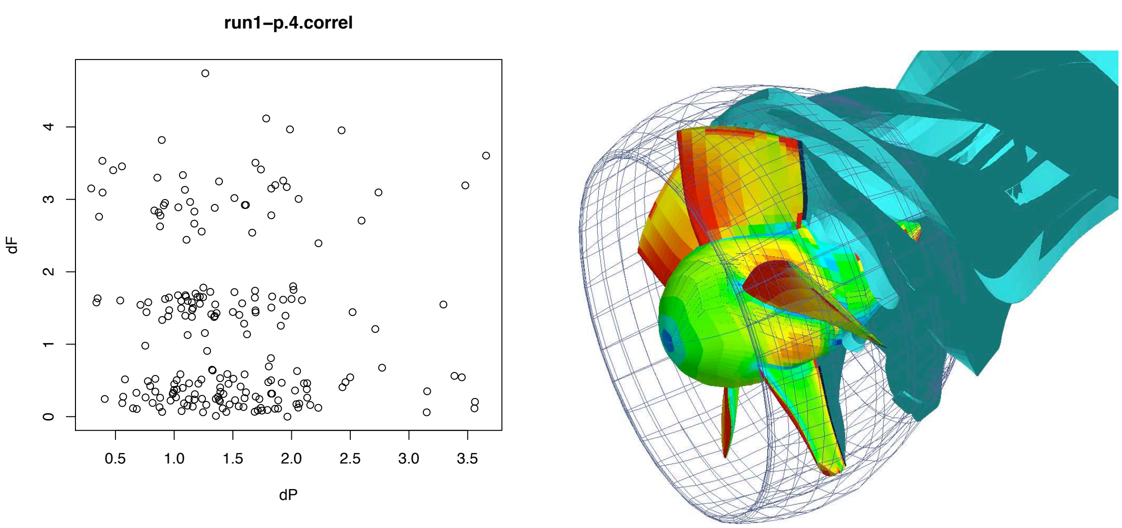

Fig. 1. Left: Quality to search space distance correlation around a good solution, Δquality over Δsearch space distance; Right: Vizualization of the running linearjet propulsion system simulation.

References

1. Beyer, H.G., Schwefel, H.P.: Evolution strategies: A comprehensive introduction. Natural Computing **1**(1) (2002) 3–52
2. Jones, D., Schonlau, M., Welch, W.: Efficient global optimization of expensive black-box functions. Journal of Global Optimization **13** (1998) 455–492
3. Emmerich, M., Giannakoglou, K., Naujoks, B.: Single- and multi-objective evolutionary optimization assisted by gaussian random field metamodels. IEEE Transactions on Evolutionary Computation **10**(4) (2006) 421–439
4. Bartz-Beielstein, T.: Experimental Research in Evolutionary Computation – The New Experimentalism. Natural Computing Series. Springer, Berlin (2006)
5. Auger, A., Hansen, N.: A Restart CMA Evolution Strategy With Increasing Population Size. In McKay, B., et al., eds.: Proc. 2005 Congress on Evolutionary Computation (CEC'05), Piscataway NJ, IEEE Press (2005) 1769–1776
6. Naujoks, B., Steden, M., Müller, S.B., Hundemer, J.: Evolutionary optimization of ship propulsion systems. In: Proc. 2007 Congress on Evolutionary Computation (CEC'07), Singapore, Piscataway NJ, IEEE Press (2007)