SPOT applied to non-stochastic optimization problems - An experimental study

Thomas Bartz-Beielstein, Martina Friese, Boris Naujoks, Martin Zaefferer

Cologne University of Applied Sciences,
Faculty for Computer and Engineering Science,
51643 Gummersbach, Germany
{firstname.lastname}@fh-koeln.de

Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: Optimization; G.3 [**Probability and Statistics**]: Experimental design

Keywords

Sequential Parameter Optimization

1. INTRODUCTION

Most parameter tuning methods feature a number of parameters themselves. This also holds for the Sequential Parameter Optimization [1] Toolbox (SPOT[1]). It provides default values, which are reasonable for many problems, but these defaults are set to favor robustness over performance.

By default, a Random Forest (RF) [2] model is used for the surrogate optimization. The RF model is built rather fast. It runs robustly (i.e. it does not crash) and can handle non-ordered parameters (i.e. factors) very well. However, the RF model does provide poor optimization performance for a number of problems, due to the inbuilt discontinuities. It would often be more reasonable to use Kriging models [4]. These usually perform well for small and medium sized decision space dimensions. For use with the SPOT package, there are several existing packages that provide Kriging methods that often fit the required problem well (DiceKriging, mlegp, etc.). However, these methods have one thing in common, they are not robust. Especially when several design points (samples in the decision space) are close to each other, those functions often fail. Hence, in SPOT versions greater 1.0, a Kriging model based on the Matlab code by Forrester et.al. [3] was introduced.

2. RESEARCH GOALS AND QUESTIONS

To examine SPOT parameterizations, we invoke SPOT as an optimization method on a number of 2 dimensional optimization problems. These test functions are Ackley (f1), Branin (f2), Goldstein-Price (f3), Griewank (f4), Mexican-Hat (f5), Rastrigin (f6), Rosenbrock (f7), Sphere (f8) and Weierstrass (f9)[2]. Since first results indicate that Forrester modeling performs much better than RF models, we seek for configurations where the RF approach outperforms the Forrester one and want to know why one model is superior to the other one in the corresponding situations. Moreover, we look for suggestions on how to parameterize SPOT in general and the Forrester approach in particular.

In detail, we investigate the influence of the parameter `seq.design.new.size` (i.e. the number of new points evaluated on a target function in each sequential step). Additionally, we vary the size of the region of interest on the test functions around the global optimum (ROI size). For all approaches, the initial solutions are chosen randomly inside these boundaries. Moreover, we compared different optimization methods for solving the internal optimization problem for the maximum likelihood estimation in Forrester. Besides comparing different parameterizations among each other, these are tested against reference algorithms. Therefore, we considered Simulated Annealing (SANN), Broyden-Fletcher-Goldfarb-Shanno (BFGS) and Covariance Matrix Adaptation Evolution Strategy (CMAES) according to their R implementations.

For internal optimization purposes within SPOT, we employed Limited-memory BFGS with Bounds (L-BFGS-B) and CMAES. Note, that L-BFGS-B is different to the reference algorithm BFGS.

3. EXPERIMENTAL SETTINGS

Due to the fact that SPOT and it's default settings were developed for noisy optimization, some of these settings have to be set to non-default values. To this end, we do not use SPOT's OCBA functionality and set all repeats to 1, i.e. `spot.ocba = FALSE, init.design.repeats = 1, seq.design.maxRepeats = 1`.

Moreover, we specified how many design points are evaluated on the surrogate model in each sequential SPOT step, i.e., 2 000. When using the RF model, all 2 000 evaluations were used for a Latin Hypercube Design (LHD). For the Forrester model, only 1 600 were used for LHD, the remaining 400 evaluations were used for optimization on the surrogate featuring the L-BFGS-B approach. The reason for only using LHD sampling on the RF are discontinuities in the model. We allowed 100 function evaluations on the test functions for each algorithm.

[1]SPOT and all other used R packages can be retrieved from the CRAN homepage, i.e. `http://cran.r-project.org`. Detailed information on the algorithms can be found in the package documentations published there.

[2]All functions used are part of the soobench R-package, which also provides further information on these functions

GECCO'12 Companion, July 7–11, 2012, Philadelphia, PA, USA.
ACM 978-1-4503-1178-6/12/07.

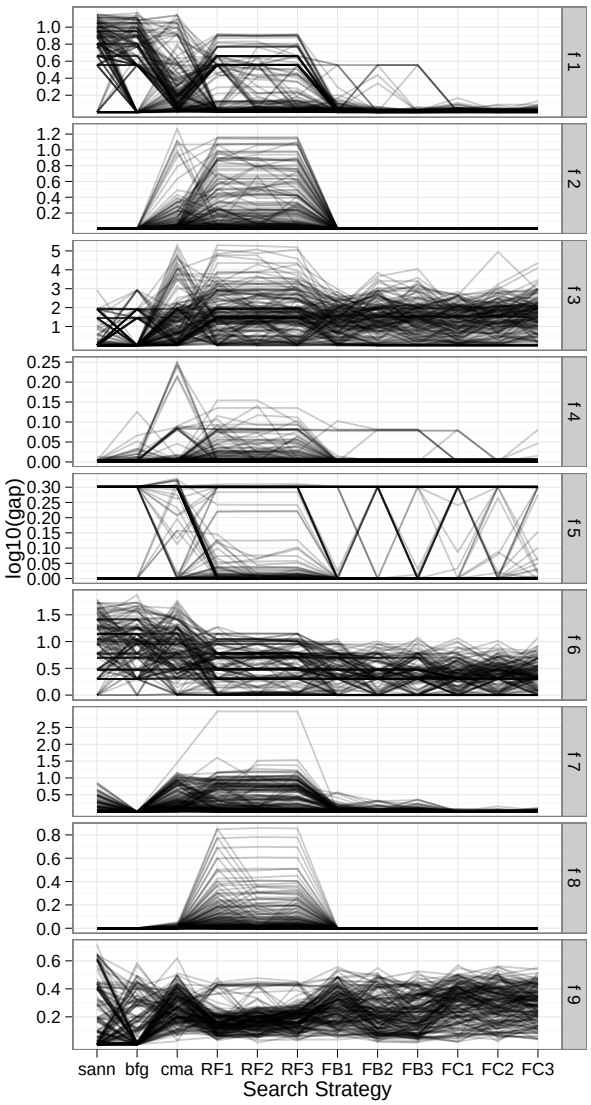

Figure 1: Parallel axes plot of all results found, separated by function and method. Each line shows results of a unique combination of one seed and one ROI size.

4. RESULTS

The parallel axes plot in Fig.1 summarizes the found results. Despite of the parameters that have been distinguished in the plot, i.e. test function, optimization algorithm, points on one line have been received with the same combination of random seed and ROI size We present different plots for the 9 test functions from section 2. The algorithms are laid on the x axis. Next to the 3 classical approaches, we feature the RF approach and two Forrester ones, i.e. employing L-BFGS-B (FB) and CMAES (FC). All these SPOT based approaches are further distinguished according to the values of the `seq.design.new.size` parameter $\{1, 2, 3\}$. The y axis indicates the final gap between the received function value and the global optimum in logarithmic scale.

The experiments showed that BFGS performed best on any problem where the boundaries were close to the global optimum. Choosing the boundaries this way basically reduces the test instance to a local optimization problem. On larger distances, SPOT managed to outperform BFGS occasionally, in particular if the functions are multi modal. On other functions, it receives comparable results. However, when SPOT is able to locate an optimum, it often does not do so with the same precision as BFGS or SANN.

One problem in these results is that the implementation of BFGS sometimes used significantly more evaluations on the target function than specified, even exceeding 200 instead of stopping after 100 evaluations. For a better comparison, this should be taken care of in future experiments.

It has to be noted that SPOT requires a longer run time because building the surrogates is time consuming. Therefore, we suggest to apply SPOT for real world applications if the function evaluations dominate the overall run time.

On any function except for f9 (Weierstrass), Forrester (FB, FC) outperforms RF by a large margin. One reason for RF performing this well on f9 might be the rectangular regions in the Weierstrass function. These are probably rather well modeled by the RF, as RF models also subdivide the modeled space into rectangular areas.

Regarding the choice of the `seq.design.new.size` in SPOT, the experiments did not show results in favor of any of the three in general. We observed that a new design size of 2 or 3 was sometimes better for Forrester. Thus, for future experiments a `seq.design.new.size` of 3 is advisable since this leads to a reduced run time of SPOT, as the surrogate models are built less frequently. Concerning the optimization problem of the maximum likelihood estimation in Forrester, no clear difference between L-BFGS-B (FB) and CMAES (FC) can be observed.

5. SUMMARY AND OUTLOOK

The proposed versions of SPOT are able to outperform classical optimization approaches on some of the functions in our limited test environment. We seek to enlarge the test bed in future experiments, e.g. with the BBOB test set. Further parameters of SPOT have to be examined as well. At present, we suggest to consider SPOT optimization with Forrester models and a new design size of 3 due to the findings presented.

Acknowledgements

The authors kindly acknowledge funding of the German Federal Ministry of Education and Research, Research Grants 170N2309 (FIWA), 17N0311 (MCIOP), and 17002X11 (CIMO).

6. REFERENCES

[1] T. Bartz-Beielstein, K. E. Parsopoulos, and M. N. Vrahatis. Design and analysis of optimization algorithms using computational statistics. *Applied Numerical Analysis and Computational Mathematics (ANACM)*, 1(2):413–433, 2004.

[2] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[3] A. Forrester, A. Sobester, and A. Keane. *Engineering Design via Surrogate Modelling*. Wiley, 2008.

[4] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–435, 1989.