Universiteit Leiden
The Netherlands

Self-adjusting surrogate-assisted optimization techniques for expensive constrained black box problems

Bagheri, S.

Citation

Bagheri, S. (2020, April 8). *Self-adjusting surrogate-assisted optimization techniques for expensive constrained black box problems*. Retrieved from https://hdl.handle.net/1887/87271

Version:	Publisher's Version
License:	Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden
Downloaded from:	https://hdl.handle.net/1887/87271

Note: To cite this publication please use the final published version (if applicable).

Cover Page

Universiteit Leiden

Leiden University Repository

The handle http://hdl.handle.net/1887/87271 holds various files of this Leiden University dissertation.

Author: Bagheri, S.
Title: Self-adjusting surrogate-assisted optimization techniques for expensive constrained black box problems
Issue Date: 2020-04-08

Chapter 1

Introduction

1.1 Motivation

Nowadays, optimization problems emerge in nearly every possible field of science, industry or business. Minimizing cost or time, maximizing profit or efficiency of any procedure belongs to the daily challenges we may face regardless of our field of profession. It is very likely that real-world optimization problems are restricted to various sorts of limitations imposed by many different sources, making only a subset of solutions feasible. This being said, in real-world applications it is very common to encounter constrained optimization problems (COPs) dealing with optimization of an objective function subject to a single or multiple constraint functions. Classic gradient-based constrained or unconstrained optimization algorithms including Newton's methods, Lagrange multiplier, etc. can be used for finding optimal solutions of any real-world problem that can be formulated with mathematical functions. However, formulating an optimization problem in terms of simple mathematical functions is not always possible or it might require an oversimplification of the problem.

Significant growth of the computational power in the last decades made it possible for engineers and scientists to develop sophisticated simulation software in order to model complex physical phenomena and therefore introduce new tasks including new classes of optimization problems. In many engineering fields finding an optimal design for large complex systems which are highly parametrized became popular only after development of detailed accurate simulation software replacing real experiments. Many industrial design tasks can be formulated as optimization problems where their objective and/or constraint functions are black box, meaning that they can only be evaluated through conducting a simulation run. Treating such optimization problems which do not have an explicit algebraic formulation is not possible with the classical optimization methods. Modern optimization heuristics, many of them being inspired by natural processes, are often derivative-free techniques suitable to tackle black box optimization problems.

As simulation software tools evolve rapidly and become more detailed and accurate, they become significantly more time-consuming to run. For this reason, the optimization problems for which objective and/or constraint functions can only be evaluated through time-expensive simulation runs are considered as challenging expensive tasks. The conventional modern optimization techniques like evolutionary algorithms are not always a practical tool to handle expensive black box real-world problems, despite their contribution to non-expensive black box problems. This is because modern optimizers often require several ten thousands of function evaluations and their required number of function evaluations usually increases exponentially as the size of the parameter space increases.

In the fast changing automotive industry, finding optimal stable vehicle designs minimizing production costs, fuels consumption, pollutant production, mass or maximizing speed, power, efficiency, etc. is an important task. Such tasks can be seen as black box constrained optimization problems subject to constraints coming from conducting crash tests. Crash test simulation is an example for a time-expensive simulation software which revolutionized the automotive industry by replacing the real experiments (in this case the physical prototype) with the virtual ones [165]. Spethmann et al. in [165] give a comprehensive overview about the evolution of crash simulators since the development of the explicit finite element method (FEM) for crash events in 1960, to the first time that supercomputers, though in high costs, made the FEM-based simulators a practical tool in 1970, up to the present day. One of the very first crash simulations going back to 1983 had only 60 elements and needed 33 hours of CPU time. Only three years later the crash simulation developed for Volkswagen Polo consisted 5661 finite elements but the simulation took only 4 hours [165]. In 1990 Opel Astra introduced a crash model having 70 000 elements that took 2 days to complete a simulation run. In 2003 Opel Astra's sophisticated crash model had more than 1 million elements and its execution took about 2.5 days to 6.3 weeks. Up till now the enhancement of the simulation software as well as the computational power of supercomputers did not slow down, which yield highly detailed simulations running in a time scope of couple of hours to weeks depending on the model, application, number of supercomputers in service and many other factors [28, 165, 90]. Therefore, the development of efficient optimization techniques which can find optimal or near-optimal solutions with very limited number of function evaluations is crucial when we are dealing with real-world optimization problems associated with expensive simulations.

Expensive black box optimization problems are not only limited to the problems involved with crash test simulations. Any sort of constrained/unconstrained optimization problems for which constraint and/or objective functions are outputs

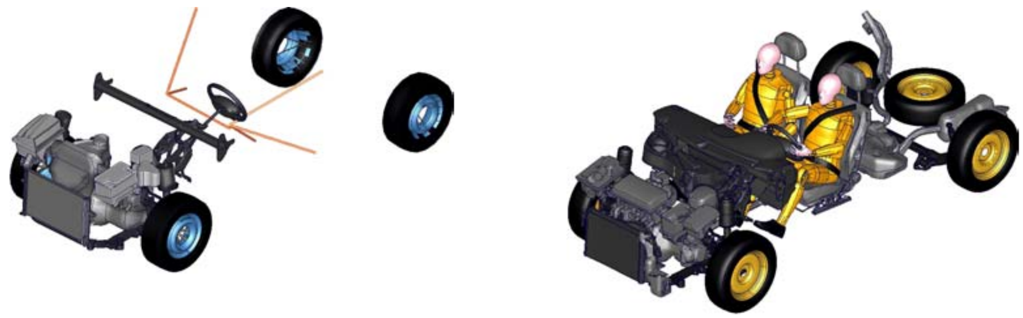

Figure 1.1: Comparing the crash model contents for 1998 and 2003 Opel Astra [165]. The images are taken from [165].

of physical complex procedures modeled with partial differential equations (PDEs), aka PDE constrained optimization problem, are expensive problems due to the time-consuming simulation runs required for their evaluation [139]. A few instances among numerous other examples for real-world black box constrained/unconstrained optimization problems are listed as follows: reducing heat loss by optimizing combustion chamber shape in automotive industry [3], airfoil design optimization in aerospace engineering, aiming at maximizing lift or minimizing the aerodynamic drag subject to multiple constraints on drag force, pressure drag, etc. [14], water turbine shape optimization [175, 64], gas transmission pipeline optimization [34], submersible oil drilling pump design optimization [71], optimization of the cooling system of the motor of an electrical panel [123], shape optimization of medical devices [2], pressure vessel shape optimization [93] and many more.

Although the real-world optimization problems coming from each corner of the industry have different characteristics, there are several important common properties which make them challenging to tackle. Optimization problems from industry are usually *subject to multiple constraints* due to many restrictions in practice caused by resource limitation, geometrical constraints, stability, safety, budget, etc. Real-world optimization problems are *black box*, meaning that their objective and/or constraint functions are too complex to be modeled with an explicit algebraic formula. Therefore, they can only be evaluated by running *time-expensive* simulation runs. One more common challenge for the real-world optimization problems is their *high-dimensionality* in parameter space. Summarizing the main common properties of the real-world optimization problems which makes them challenging, we can say that they are often black box, constrained, expensive and high-dimensional.

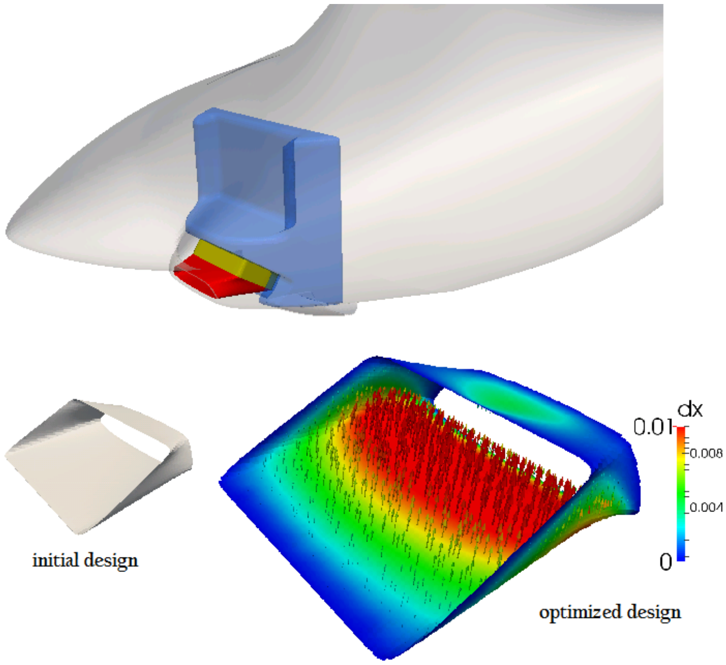

Figure 1.2: Optimizing the geometrical design of the cooling system of the electrical panel on an airplane: (top) the cooling system is shown in color placed on an airplane, only the red part (the diffuser) can be modified. (bottom-left) the initial diffuser design. (bottom-right) the optimized diffuser geometry, colors represent the amount of accumulative change at each point suggested by the optimized model; at the lower parts of the diffuser wall the changes are larger than the other areas [123]. The images are taken from [123].

In the recent years, there were many attempts to handle real-world optimization problems in an efficient manner. To do so, often researchers break down the problems to simpler versions and try to tackle each challenge in an isolated fashion. The development of the modern derivative-free heuristics helped solving the black box optimization problems regardless of their dimensional spaces, but they need usually a large number of function evaluations. Despite the contributions made by these approaches for black box unconstrained inexpensive problems, one of their weak points is that they have several hyperparameters which have to be tuned apriori. The demanding challenge with the expensive function evaluation is often addressed by usage of surrogate-assisted algorithms. Surrogate-assisted optimizers aim at saving expensive simulation runs by replacing the real functions with cheap and fast mathematical models. Also, several different constraint handling methods were developed which can be coupled with the unconstrained optimizers. **Chapter 2** provides an overview about the related work. In **Chapter 2** we give a brief survey about the existing optimization methods suitable for addressing black box unconstrained and constrained optimization problems. Very limited studies are devoted to address all the demanding challenges of the real-world optimization problems simultaneously. For example, a surrogate-assisted efficient constrained optimization framework with no or small need of hyperparameter tuning was missing before this work. As the title suggests, in this dissertation we work on the development of efficient *self-adjusting surrogate-assisted* optimization techniques for *expensive constrained black box problems*.

In this work, we introduce two surrogate-assisted optimization techniques SACOBRA and SOCU. SACOBRA standing for *self-adjusting constrained optimization by radial basis function interpolation* is an efficient technique using RBF interpolations as surrogates for objective and constrained functions which is also able to automatically control some of its important hyperyparameters without any prior information about the problems. Although the algorithm was initially only developed for constrained optimization problems (COPs) with inequality constraints (**Chapter 3**), the framework was later extended to handle COPs with equality and inequality constraints in (**Chapter 4**). Cubic radial basis function interpolation has shown strong performance as surrogate in the SACOBRA optimization framework. However, no theoretical or practical evidence suggested that cubic RBFs are the best choice but our preliminary results have shown the opposite, meaning that different types of radial basis functions delivered contrary performances on modeling different functions. Therefore, we have developed an online model selection procedure for SACOBRA to automatically choose the best type of radial basis function during the optimization procedure in **Chapter 8**.

SOCU standing for *surrogate-assisted optimization encompassing constraints and uncertainties* is the second approach introduced in this dissertation which utilizes Kriging aka Gaussian Processes, a probabilistic modeling technique, as surrogates. SOCU, described in **Chapter 5** and **Chapter 6**, can be considered as an extension to the efficient global optimization algorithm [90] (EGO) for handling constrained optimization problems. To our best knowledge it is the first time that an EGO-based[1] constrained optimizer is evaluated on the challenging G-problem-COPs.

In order to evaluate the proposed optimization techniques, two real-world constrained optimization problems and a set of well-studied toy problems known as G-problems suite [107] are used as benchmarks. MOPTA08 [92] is a large scale

[1]EGO: Efficient Global Optimization technique is a surrogate-assisted solver using Kriging probabilistic modeling, aka Bayesian optimization

mass optimization problem in the auto industry. This problem was presented at the MOPTA 2008 conference as a competition challenge. The problem which was introduced by Don Jones as a technical fellow at General Motors, has a 124-dimensional space with 68 nonlinear black box constraints. The parameters come from part material variables or shape variables and the constraints are the output of the crash simulation which in the ideal case can be computed 60 times a day in practice. It is desired to find a design with 10% to 20% reduced mass within one month which means the maximum number of function evaluations is limited to $60 \times 30 = 1800$. The second real-world COP used in this dissertation is an airfoil design problem aiming at minimizing the aerodynamic drag force subject to multiple equality and inequality constraints, as it is described in **Chapter 5**. Moreover, we use a set of 24 COPs, the so-called G-problem set [107] to asses the developed algorithms on problems with various difficulties in terms of size of the parameter space, number of equality and inequality constraints, type of the objective and constraint functions, etc.

RBF interpolation used in SACOBRA and **Kriging** used in SOCU are both common choices of modeling techniques for surrogate-assisted optimization frameworks. Although both methods come from very different origins, they have undeniable similarities. Some similarities and differences between RBF and Kriging are mentioned in **Chapter 7**. Kriging unlike the RBF interpolation, provides an uncertainty measure, indicating how uncertain the model at each point is. This property makes Kriging a popular choice for many optimizers, especially efficient unconstrained solvers which employ probabilistic concepts. Despite the close ties between RBF interpolation and Kriging, RBFs lack the mentioned property by their nature. This motivated us to investigate whether it is possible to determine an uncertainty measure for any arbitrary RBF kernel by means of analogy between the two modeling techniques.

In order to deliver high quality fits for the objective and constraint functions from widely different classes, there are several self-adjusting elements in SACOBRA that strengthen the modeling phase, like *plog*-transformation and constraint rescaling introduced in **Chapter 3** or the online model selection functionality introduced in **Chapter 8**. However, our attempt to apply SACOBRA on the noiseless BBOB unconstrained optimization benchmark [58] depicted weak performance of SACOBRA in solving problems having objective functions with high-conditioning. This weak performance occurred mainly due to the RBF interpolations struggling to provide any useful model in case of having high conditioning objective functions. Such functions have a high ratio of steepest slope in one direction to flattest slope in another direction. In **Chapter 9** we try to address the challenge that SACOBRA faces in solving unconstrained optimization problems with high conditioning objective functions. We develop an online whitening approach for SACOBRA trying to transform the functions with high-conditioning in a way to become easier to model by means of RBF interpolations. In **Chapter 9** we show that SACOBRA with the online whitening approach (OW) is able to find solutions with significantly better optimization errors. Although a great number of function evaluations are imposed by the proposed OW mechanism, most of them are parallelizable. SACOBRA+OW performs better than SACOBRA without the OW mechanisms in solving BBOB with high-conditioning, though further development of the SACOBRA+OW to improve its efficiency and to compete with the state-of-the-art in solving the BBOB problems is left as a future research direction.

1.2 Summary of Research Questions

For the benefit of the reader, we summarize the set of all research questions which will be tackled in the following chapters.

Chapter 3 describes a self-adjusting constrained optimizer called SACOBRA which uses radial basis function interpolation (RBF) as surrogate. Throughout this chapter we try to answer the following research questions:

Q3.1 Do numerical instabilities occur in RBF surrogates and is it possible to avoid them?

Q3.2 Is it possible with SACOBRA to start with the same initial parameters on all G-problems and to solve them by self-adjusting the parameters on-line?

Q3.3 Is it possible with SACOBRA to solve all G-problems in a given, small number of function evaluations (e. g., 1000) ?

In **Chapter 4**, SACOBRA optimization frameworks is extended to address COPs with equality constraints. The main research questions which will be addressed in **Chapter 4** are listed as follows:

Q4.1 How can SACOBRA be extended to handle COPs with equality constraints efficiently and solve the common dilemma of margin-based equality handling methods?

Q4.2 Is a gradually shrinking feasibility margin an important ingredient for SACOBRA to produce high-quality results on COPs with equality constraints?

SOCU is the second surrogate assisted constrained optimizer developed in this work which performs based on the probabilistic modeling technique Kriging. The SOCU algorithm is described in **Chapter 5**. To our best knowledge it is the first time that an EGO-based[2] constrained optimizer is evaluated on the challenging G-problem-COPs. In **Chapter 5** we try to answer the following research questions:

Q5.1 Is it possible to modify existing EGO-based optimization algorithms to handle challenging COPs with multiple *active* constraints?

[2]EGO: Efficient Global Optimization technique is a surrogate-assisted solver using Kriging probabilistic modeling, aka Bayesian optimization

Q5.2 Is it possible to balance the exploration of feasible and infeasible infill points in a proper way?

In **Chapter 6**, SOCU framework is extended to address COPs with equality constraints.

Chapter 7 compares radial basis function interpolation (RBF) and Gaussian Process modeling (GP) techniques with each other and tackles the following research question:

Q7.1 Can we determine an estimation of the model uncertainty for any arbitrary kernel, e. g., cubic RBF, augmented cubic RBF, etc?

The online model selection extension in SACOBRA framework is described in **Chapter 8**. The research questions listed below are answered in **Chapter 8**:

Q8.1 Are there COPs which significantly benefit from using different types of RBF functions for objective and constraint functions?

Q8.2 Is it possible to advise an online algorithm which automatically selects the right RBF type for each function? That is, does such an algorithm boost up the overall performance of a COP solver?

In **Chapter 9**, we deal with optimization problems with high conditioning fitness functions. After discussing the difficulties of modeling such functions, we try to develop an approach for transforming high conditioning functions to answer the research question below:

Q9.1 Can we advise an approach to transform high conditioning functions to low conditioning in an online manner?

Finally in **Chapter 10**, the main results of this dissertation will be summarized and the strengths and limitations of the developed optimizers will be discussed. Furthermore, possible future directions of research will be mentioned.