Title
The Statistical Filter Approach to Constrained Optimization

Journal
Technometrics, 62(3)

ISSN
0040-1706

Authors
Pourmohamad, Tony
Lee, Herbert KH

Publication Date
2020-07-02

DOI
10.1080/00401706.2019.1638304

Peer reviewed

The Statistical Filter Approach to Constrained Optimization

Tony Pourmohamad*
Genentech, Inc.
and
Herbert K.H. Lee
Department of Statistics,
University of California, Santa Cruz

June 13, 2019

Abstract

Expensive black box systems arise in many engineering applications but can be difficult to optimize because their output functions may be complex, multi-modal, and difficult to understand. The task becomes even more challenging when the optimization is subject to multiple constraints and no derivative information is available. In this paper, we combine response surface modeling and filter methods in order to solve problems of this nature. In employing a filter algorithm for solving constrained optimization problems, we establish a novel probabilistic metric for guiding the filter. Overall, this hybridization of statistical modeling and nonlinear programming efficiently utilizes both global and local search in order to quickly converge to a global solution to the constrained optimization problem. To demonstrate the effectiveness of the proposed methods, we perform numerical tests on a synthetic test problem, a problem from the literature, and a real-world hydrology computer experiment optimization problem. Supplementary materials for this article are available online.

Keywords: Black box function, multivariate Gaussian process, computer simulator, computer experiment

*The authors gratefully acknowledge Stefan Wild for first introducing them to filter methods and for the many insightful conversations thereafter, as well as Samantha Crane for her proof of concept of the hybrid approach.

1 Introduction

The difficult problem of black box constrained gradient-free optimization problems has driven research on an increasing number of model-based methods (Sasensa et al. (2002), Lindberg and Lee (2015), Gramacy et al. (2016)). Primarily, model-based methods have utilized Gaussian processes as a means of building surrogate models for computer simulators, with the surrogate used to guide optimization and heuristics used for handling constraints (Santner et al. (2003), Letham et al. (2018)). A typical challenge here is that the computer models are often expensive to evaluate and so efficient statistical models are needed to serve as a fast approximation to the computer model. Furthermore, constrained optimization is usually difficult because at least one of the constraints operates in opposition to the objective function, i.e., they are negatively correlated. It is due to these problems that we develop a novel method for constrained optimization capable of reliably modeling computer models quickly, with few function evaluations, and that is also able to efficiently incorporate information about and predict negatively correlated outputs through joint modeling techniques.

In this paper, we introduce a new methodology that combines statistical modeling with a numerical optimization approach known as the filter method. Filter methods take a Pareto frontier approach to optimizing the objective function and meeting all of the constraints. The difference from a multi-objective optimization via exploration of a Pareto front is that all of the constraints must be met for the solution to be valid, and so the filter method treats the constraints differently from the objective function and uses several approaches to improve the efficiency of this process. As a result, filter methods are more effective at global optimization than a typical more local numerical optimization routine. However, filter methods rely upon a good way of searching the input space, and it is that component where the statistical model can make a major contribution, leading to an improved hybrid algorithm that provides for efficient global optimization that outperforms our comparator methods. Filter methods are described in more detail in Section 2.

In this paper, we address problems of the form

$$\min_{\mathbf{x}}\{f(\mathbf{x}) : \mathbf{c}(\mathbf{x}) \leq 0,\, \mathbf{x} \in \mathcal{X}\} \tag{1}$$

where $\mathcal{X} \subset \mathbb{R}^d$ is a known, bounded region such that $f : \mathcal{X} \rightarrow \mathbb{R}$ denotes a scalar-valued objective function and $\mathbf{c} : \mathcal{X} \rightarrow \mathbb{R}^m$ denotes a vector of m constraint functions. We assume that a solution to (1) exists, however, in our problems of interest (1) is an expensive black box optimization problem where both the objective, f, and constraint, $\mathbf{c}$, may require black box simulation (i.e., running a computer model revealing little about the functional form of the objective and constraints). Furthermore, we focus on the *derivative-free* situation where no information about the derivatives of the objective and constraint functions is available (Conn et al., 2009). Herein we focus on the case of deterministic computer models where running the computer model for the same input always yields the same output. A motivating real-world example, the pump-and-treat hydrology problem (Matott et al., 2011) involves a groundwater contamination scenario based on the Lockwood Solvent Groundwater Plume Site located near Billings, Montana. Due to industrial practices in the area, two plumes containing chlorinated contaminants have developed near the Yellowstone river. The two plumes are slowly migrating towards the Yellowstone river and of primary concern is keeping the chlorinated contaminants from leaking into the Yellowstone river. A pump-and-treat remediation is proposed, in which six wells in total (two wells are placed in one plume and four wells in the other plume) are placed to pump out contaminated water, purify it, and then return the treated water, at six locations. A computer simulator was built to model this physical process where the inputs to the simulator are the pumping rates for the six pump-and-treat wells, $x_1, ..., x_6 \in \mathcal{X}$, and the output of the simulator is the cost of running the pump-and-treat wells (f) and whether or not the plumes have been contained ($\mathbf{c}$). Thus, the objective of the pump-and-treat hydrology problem is to minimize the cost of running the pump-and-treat wells while containing both plumes.

This paper further advances the literature on the use of statistical surrogate modeling for constrained optimization. Some relevant earlier work also uses Pareto frontiers, although in less sophisticated ways than filter methods do. Wilson et al. (2001) proposed exploring the Pareto frontier using surrogate approximations in order to solve a biobjective optimization problem, and following the same suit, Parr et al. (2012) took it a step further by incorporating constrained expected improvement into building the Pareto frontier. Following the earlier works of Jones et al. (1998), Svenson and Santner (2012) extended the idea of expected improvement to the multiobjective optimization case by exploring Gaussian process surrogate modeling and Pareto frontiers as well. Filter methods improve upon these earlier works by providing an optimization approach that is provably convergent (Fletcher et al. (1998), Fletcher et al. (2002), Audet and Dennis (2006), Ribiero et al. (2008)) and that converges in practice with fewer functional evaluations.

We note that Pareto front approaches are just one way to address the problem, and a variety of other approaches exist. Sasensa et al. (2002) used surrogate models based on Gaussian processes and handled the constraints by transforming the problem into an unconstrained problem by use of a penalty function. Similarly, Gramacy et al. (2016) used a penalty function approach based on augmented Lagrangians to handle the constraints and searched the objective space using Gaussian process methods with expected improvement techniques. Lee et al. (2011) and Lindberg and Lee (2015) solved black box optimization problems using constrained expected improvement; the former made use of calculations of the probability of satisfying the constraint and the latter used asymmetric entropy as a guiding measure of satisfying the constraints.

Toy problem. To illustrate the kind of constrained optimization problems we seek to solve, we present the following toy test problem that was introduced in Gramacy et al. (2016), and is a problem of the form (1). The problem consists of a linear objective, $f(\mathbf{x}) = x_1 + x_2$, with domain $\mathcal{X} \in [0,1]^2$, and two nonlinear constraints given by

$$\begin{aligned} c_1(\mathbf{x}) &= \frac{3}{2} - x_1 - 2x_2 - \frac{1}{2}\sin\left(2\pi(x_1^2 - 2x_2)\right) \\ c_2(\mathbf{x}) &= x_1^2 + x_2^2 - \frac{3}{2}. \end{aligned}$$

Figure 1 shows the feasible region for the toy problem and the three local optima, with $\mathbf{x}_B$ being the global solution to the problem. And although a toy problem, Gramacy et al. (2016) chose this problem since it shared many characteristics with the real-world hydrology computer experiment presented in Section 5.3, in particular, both problems have a linear objective function and two constraint functions which may be highly nonlinear and nonconvex.

$$\mathbf{x}_A \approx (0.7197, 0.1411)$$
$$f(\mathbf{x}_A) \approx 0.8609$$
$$\mathbf{x}_B \approx (0.1954, 0.4044)$$
$$f(\mathbf{x}_B) \approx 0.5998$$
$$\mathbf{x}_C = (0, 0.75)$$
$$f(\mathbf{x}_C) = 0.75$$

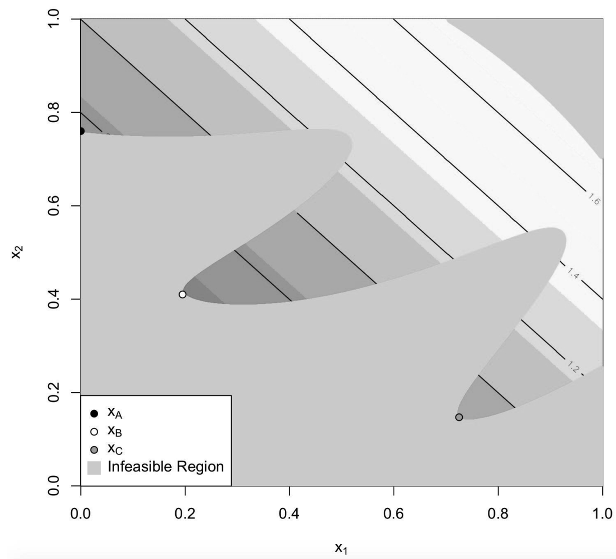

Figure 1: The toy problem introduced in Gramacy et al. (2016). $\mathbf{x}_B$ is the global solution to the problem, while $\mathbf{x}_A$ and $\mathbf{x}_C$ are local minimizers.

The remainder of this paper is organized as follows. We first introduce the mathematical work horse, filter methods, used for solving constrained optimization problems, and then give a brief introduction to surrogate modeling and our choice of statistical model to use. After that, we introduce our new algorithm, the statistical filter, for solving problems of the form (1) and evaluate the performance of the new method on a synthetic test problem, a welded beam problem from the literature, and the real-world hydrology computer experiment. Lastly, we finish with some potential extensions to the new methodology and end with concluding remarks.

2 Filter Methods

Filter methods were introduced by Fletcher and Leyffer (2002) as a means of solving nonlinear programming problems without the use of a penalty function. For example, a linear combination of the objective function and a constraint violation function, say $h(\mathbf{x}) = ||\max\{\mathbf{0}, \mathbf{c}(\mathbf{x})\}||$ for some norm, can define the penalty function

$$p(\mathbf{x}; \pi) = f(\mathbf{x}) + \pi h(\mathbf{x}), \tag{2}$$

where $\pi > 0$ is the penalty parameter. The new penalized objective function (2) transforms the constrained problem in (1) into an unconstrained optimization problem. Penalty functions suffer from the drawback of having to specify a suitable penalty parameter that balances the often-competing aims of minimizing f and h. Instead of combining the objective and constraint violation into a single function (2), filter methods take a biobjective optimization approach and try to minimize both f and h simultaneously. However, priority is placed on minimizing h since a feasible solution only exists when $h(\mathbf{x}) = 0$. Borrowing concepts from multiobjective optimization, filter methods solve the constrained optimization problem in (1) by locating the set of all nondominated inputs $\mathbf{x} \in \mathcal{X}$. A point $\mathbf{x}_i \in \mathcal{X}$ dominates a point $\mathbf{x}_j \in \mathcal{X}$ if and only if $f(\mathbf{x}_i) \leq f(\mathbf{x}_j)$ and $h(\mathbf{x}_i) \leq h(\mathbf{x}_j)$ with $(h(\mathbf{x}_i), f(\mathbf{x}_i)) \neq (h(\mathbf{x}_j), f(\mathbf{x}_j))$. Geometrically, $\mathbf{x}_i$ dominates $\mathbf{x}_j$ if the point $(h(\mathbf{x}_i), f(\mathbf{x}_i))$ is below and to the left of $(h(\mathbf{x}_j), f(\mathbf{x}_j))$ (Figure 2). Fletcher and Leyffer (2002) defined a filter (similar to the Pareto front in the multiobjective literature), denoted by $\mathcal{F}$, by first determining a set of points $\mathbf{x}_i \in \mathcal{X}$, for $i = 1, \ldots, n$, such that no point in the set dominates another point in the set, and then finding the corresponding pairs $(h(\mathbf{x}_i), f(\mathbf{x}_i))$, and defining $\mathcal{F}$ as the set of all pairs $(h(\mathbf{x}_i), f(\mathbf{x}_i))$. Figure 2, shows a filter defined by the four indicated points (as well as some additional aspects to be described shortly). A particular filter may not necessarily provide a good solution to the optimization problem, so the goal of the filter method is to successively improve the filter until it converges to the optimal solution. Given the definition of the filter $\mathcal{F}$, we summarize the generic filter method as follows:

Initialize the filter $\mathcal{F}$;
while *not terminated* **do**
    Solve a subproblem to obtain a candidate point $\mathbf{x}_*$;
    Evaluate $f(\mathbf{x}_*)$ and $c(\mathbf{x}_*)$;
    if $(h(\mathbf{x}_*), f(\mathbf{x}_*))$ *is acceptable to* $\mathcal{F}$ **then**
        Add $(h(\mathbf{x}_*), f(\mathbf{x}_*))$ to $\mathcal{F}$;
        Remove any entries in $\mathcal{F}$ dominated by $(h(\mathbf{x}_*), f(\mathbf{x}_*))$;
    end
    Check for termination;
end

Algorithm 1: Generic filter method

In its basic form, $(h(\mathbf{x}_*), f(\mathbf{x}_*))$ is acceptable to $\mathcal{F}$ if $f(\mathbf{x}_*) \leq f(\mathbf{x}_j)$ and $h(\mathbf{x}_*) \leq h(\mathbf{x}_j)$ with $(h(\mathbf{x}_*), f(\mathbf{x}_*)) \neq (h(\mathbf{x}_j), f(\mathbf{x}_j))$ for some $(h(\mathbf{x}_j), f(\mathbf{x}_j)) \in \mathcal{F}$. Typically termination of the algorithm requires that some tolerance, with respect to the solution, has been achieved or that all budgetary resources, for example time or money, have been exhausted. As simple as Algorithm 1 may look, extra care must be taken to avoid convergence to infeasible points ($h(\mathbf{x}) > 0$) or to feasible points that are not stationary points of (1) (Fletcher et al., 2006). One proposed way to avoid these pitfalls is an envelope, which is added around the current filter to avoid convergence to infeasible points. A candidate point x_* generates an $(h(\mathbf{x}_*), f(\mathbf{x}_*))$ that is acceptable to the filter if

$$h(\mathbf{x}_*) \leq \beta h(\mathbf{x}_i) \text{ or } f(\mathbf{x}_*) \leq f(\mathbf{x}_i) - \gamma h(\mathbf{x}_*) \qquad \forall (h(\mathbf{x}_i), f(\mathbf{x}_i)) \in \mathcal{F}, \tag{3}$$

where $\beta, \gamma \in (0, 1)$ are constants. The envelope in (3) has stronger convergence properties (Chin and Fletcher, 2003) due to its sloping nature although an axis aligned envelope could be used if γ is allowed to equal zero. However, the true strength of the filter method is that provable convergence can be almost guaranteed near a feasible nonstationary point, when the reduction of the objective function is "large" (Ribiero et al., 2008). Additionally, an upper bound U may be placed on the constraint function in order to ensure a practical limit to the largest allowable constraint violation to the filter. An illustration of a filter with sloping envelope is given in Figure 2. Note that the dark grey shaded area in Figure 2, defined by U, is an upper bound on the acceptable constraint violation.

The success of Algortihm 1 depends on understanding the three different spaces we are working in. The most important space we denote the "design space", which is the domain of the inputs, $\mathbf{x}$, of the computer experiment (Figure 3). Being able to understand, and obtain a representative sample of, the design space is imperative for a good robust optimization algorithm. The second space, denoted the constraint space, is built from the outputs of the computer simulator, namely the constraint function, $\mathbf{c}(\mathbf{x})$, and the objective function, $f(\mathbf{x})$ (Figure 3). The constraint space is always at minimum two dimensions, i.e., $p \geq 2$, but can have dimension as high as the number of constraints plus one, i.e., $p \geq m + 1$, and can thus be extremely high dimensional. Lastly, the third space, denoted the filter space,

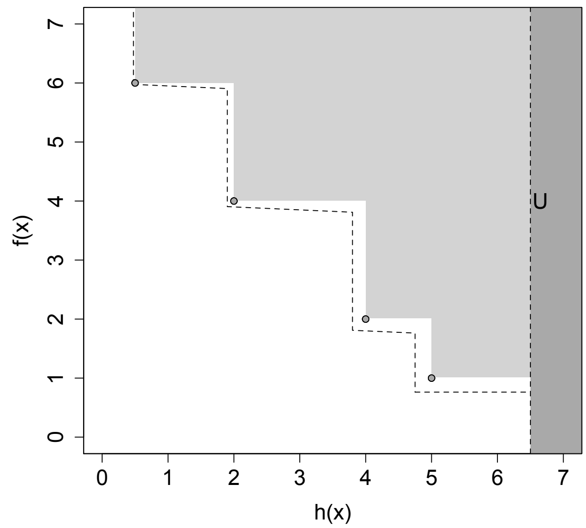

Figure 2: A typical filter, this one generated by the four marked points. All pairs $(h(\mathbf{x}), f(\mathbf{x}))$ that are below and to the left of the envelope (dashed line), and to the left of U, are acceptable to the filter. In this example, $\beta = 0.95$ and $\gamma = 0.05$.

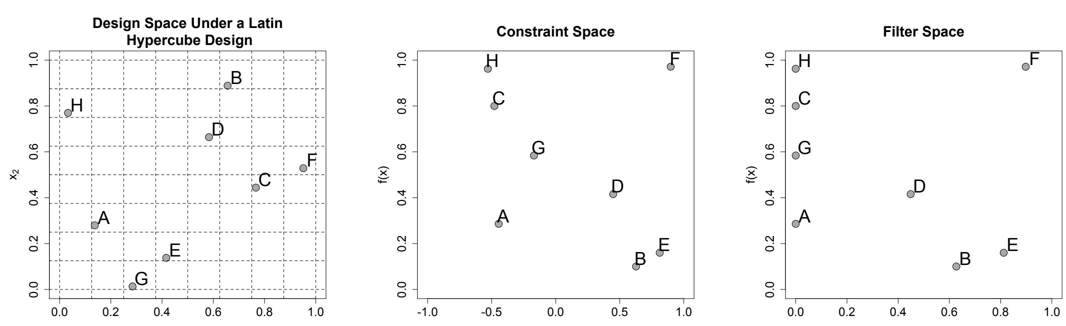

Figure 3: An example of a 2-D design space (left) where eight points were sampled using a Latin hypercube design. Also, an example of the constraint space $(c(\mathbf{x}),\ f(\mathbf{x}))$ (center) and the mapping to the filter space $(h(\mathbf{x}),\ f(\mathbf{x}))$ (right). Points in the constraint space such that $c(\mathbf{x}) \leq 0$ get mapped to $h(\mathbf{x}) = 0$ in the filter space.

is built from the objective function, $f(\mathbf{x})$, and the measure of feasibility

$$h(\mathbf{x}) = ||\max\{\mathbf{0}, \mathbf{c}(\mathbf{x})\}||_1 = \sum_{i=1}^{p-1} \max\{0, c_i(\mathbf{x})\} \tag{4}$$

The filter space is always two dimensional since the measure of feasibility, $h(\mathbf{x})$, is an aggregate measure of the constraint functions (Figure 3). Important to note that points will be distinct in the design space, but may not be so in the other two spaces, i.e., $\mathbf{x}_1 \neq \mathbf{x}_2$ with $c(\mathbf{x}_1) = c(\mathbf{x}_2)$ and/or $f(\mathbf{x}_1) = f(\mathbf{x}_2)$.

Lastly, it is important to note that Algorithm 1 is a general outline of the filter method with no explicit explanation of how to obtain a candidate $\mathbf{x}_*$. Evaluating $(f(\mathbf{x}_*),\ h(\mathbf{x}_*))$ and updating the filter are relatively simple tasks, however, the main difficulty of Algorithm 1 is in specifying and solving the subproblem. Many different subproblems have been proposed within the filter method framework, however, we prefer to think of the subproblem as a challenge in statistical modeling. In particular, our preferred approach relies on surrogate modeling via the particle learning multivariate Gaussian process (PLMGP) model of Pourmohamad and Lee (2016), as explained in the next section.

3 Fast Surrogate Modeling of Computer Experiments

Traditionally, the canonical choice for modeling computer experiments has been the Gaussian process (GP) (Sacks et al., 1989; Santner et al., 2003). Gaussian processes are distributions over functions such that the joint distribution at any finite set of points is a multivariate Gaussian distribution. Gaussian processes make for convenient surrogate models because they are conceptually straightforward (a form of nonparametric regression). Gaussian processes have a number of desirable properties such as being flexible, being able to closely approximate most functions, and often being much cheaper/faster to evaluate than the actual computer model. More importantly, using Gaussian processes for surrogate modeling allows for uncertainty quantification of computer outputs at untried (or unobserved) inputs, and also provides a statistical framework for efficient design and analysis of computer experiments (Santner et al., 2003; Kleijnen, 2015). It is for these reasons that all of the work done in this paper will be built upon the foundations of Gaussian process modeling.

We thus seek an efficient statistical model to serve as a fast approximation to the true objective function and constraints, which in our applications are computer simulation experiments. Typically in computer simulation experiments, constrained optimization is challenging because the outputs of the simulators arise from highly nonlinear functions. This lack of linearity is what makes nonparametric methods so desirable. However, fully Bayesian inference for GP models can be slow due to the need for repeated matrix inverse calculations in Markov chain Monte Carlo algorithms. Bayesian inference is needed for good uncertainty estimates, which feed into criteria in our search algorithms. Computations can get even more complicated with multivariate outputs, as we have with our objective function and one or more constraints. Furthermore, the constraints tend to be negatively correlated with the objective function, and in that part of the space most improving the objective function tends not to meet the constraints; this tradeoff between the objective function and the constraints is what makes constrained optimization problems difficult. Because we are repeatedly refitting the model after adding additional simulation runs, a sequential fitting algorithm such as particle learning makes a lot of sense, as each update is generally only a small change from the previous round with the addition of one new observation. Thus we use the particle learning multivariate Gaussian process (PLMGP) model of Pourmohamad and Lee (2016) since the particle learning approach provides efficient sequential inference and it is capable of handling $p \geq 2$ correlated outputs.

4 The Statistical Filter

We combine the filter methods of Section 2 with Gaussian process surrogate modeling in order to solve constrained optimization problems of the form (1). More formally, consider an expensive black box computer model with a d-dimensional input $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^d$ that produces a p-dimensional vector of outputs $\mathbf{y} \in \mathbb{R}^p$ where the output $\mathbf{y}$ contains the values of the objective function f and constraint function $\mathbf{c}$ at input $\mathbf{x}$. We follow the traditional statistical modeling techniques from the computer modeling literature (Sacks et al., 1989; Santner et al., 2003; Conti and O'Hagan, 2010) and build surrogate models based on joint multivariate Gaussian processes for the objective and constraint functions. Due to the expense of evaluating our black box model, we use the particle learning methods developed in Pourmohamad and Lee (2016) to sequentially update our joint multivariate Gaussian processes. As a general guideline, we follow the rule of thumb put forth by Loeppky et al. (2009), and require that the number of initial runs (or samples) of the computer model to be about ten times the input dimension, i.e., $n = 10d$, in order to achieve reasonable surrogate model fits. Likewise, we sample inputs for our initial $n = 10d$ runs of the computer model using a Latin Hypercube Design (LHD) (McKay et al., 1979; Santner et al., 2003; Fang et al., 2005). In our experience, we found LHD to be an adequate space-filling design for sampling our initial inputs and possible future candidate inputs under our PLMGP framework.

In order to combine the filter methodology with the surrogate modeling, we must recall that each space serves its own unique purpose. The design space is responsible for the sampling of inputs and dictates where the optimization algorithm is able to explore. Bad representative sampling of the inputs in the design space can lead to poor performance of the optimization algorithm, as the surrogate model may be less accurate. The constraint space is used solely for the fitting of the surrogate models, but is critical for obtaining good fits, as each constraint can be modeled separately and on a continuous space. The filter involves combining all of the constraints into a single non-negative function that is difficult to model with a statistical surrogate model. The objective function, $f(\mathbf{x})$, and the constraint functions, $c(\mathbf{x})$, are modeled in the constraint space and mapped deterministically into the filter space based on the fitted surrogate models. Once in the filter space, the actual filter methods of Section 2 are applied and new candidate points are searched for until a minimum has been declared. Updating our generic filter Algorithm 1, we have the following new algorithm:

Sample initial inputs from a LHD;

Initialize the filter $\mathcal{F}$;

while *not terminated* **do**

  Fit surrogate models for $f(\mathbf{x})$ and $\mathbf{c}(\mathbf{x})$ using the joint PLMGP model;

  Map the surrogate model in the constraint space to the filter space;

  Solve a subproblem to obtain a candidate point $\mathbf{x}_*$;

  Evaluate $f(\mathbf{x}_*)$ and $c(\mathbf{x}_*)$;

  if $(h(\mathbf{x}_*), f(\mathbf{x}_*))$ *is acceptable to* $\mathcal{F}$ **then**

    Add $(h(\mathbf{x}_*), f(\mathbf{x}_*))$ to $\mathcal{F}$;

    Remove any entries in $\mathcal{F}$ dominated by $(h(\mathbf{x}_*), f(\mathbf{x}_*))$;

  end

  Check for termination;

end

Algorithm 2: Statistical filter method

Although we have an updated statistical filter method, we still need to solve a subproblem in order to obtain a new better candidate point $\mathbf{x}_*$. In the following section we provide our solution for the subproblem of selecting candidate points and in subsequent sections test its performance on test problems and a real-world hydrology computer experiment.

4.1 Probability Acceptable to the Filter

The subproblem, dubbed "probability acceptable to the filter" (PAF), selects a new candidate point, $\mathbf{x}_*$, by maximizing the probability that the candidate point falls beyond the current filter (i.e., the candidate point dominates a point in the current filter). Stated more formally, we wish to find an $\mathbf{x}_*$ such that

$$\mathbf{x}_* = \max_{\mathbf{x} \in \mathcal{X}} \Pr\{(h(\mathbf{x}), f(\mathbf{x})) \text{ is acceptable to the filter } \mathcal{F}\}. \tag{5}$$

At each iteration of the statistical filter algorithm we choose a Latin hypercube sample from the input space $\mathcal{X}$ to take as potential solution to the PAF subproblem. Having fit a joint surrogate model to $f(\mathbf{x})$ and $\mathbf{c}(\mathbf{x})$, using the joint PLMGP model of Pourmohamad and Lee (2016), the joint predictive distribution of $(f(\mathbf{x}), \mathbf{c}(\mathbf{x}))$ is a multivariate $\mathcal{T}$ process. Given this fact, we can obtain a prediction for $(f(\mathbf{x}), \mathbf{c}(\mathbf{x}))$ based on the mean of the multivariate

$\mathcal{T}$ process (Gupta and Nagar, 2000), and quantify the uncertainty around that prediction with a probability ellipse based on the covariance matrix of the multivariate $\mathcal{T}$ process. For example, Figure 4 shows a typical filter, with envelope, and two candidate points A and B and their respective 95% probability contours based on their multivariate $\mathcal{T}$ process. Thus, finding a candidate point $\mathbf{x}_*$ satisfying (5) is equivalent to finding a candidate point with the largest area of its contour falling outside (to the left and below) of the filter's envelope. Although calculating this area of the contour is a well posed problem, solving for the area analytically is not a trivial task, and so we use Monte Carlo integration to estimate the probability of being acceptable to the filter instead. Of note, we also considered another subproblem to solve, based on a similar hypervolume metric as described in Keane (2006), Emmerich et al. (2011), and Binois and Picheny (2016), where new candidate points would be chosen conditional on maximizing the expected area (MEA) between the the current filter and a hypothesized new filter arising from potential new candidate points. Although constrained optimization performance was good under this subproblem, we found the PAF subproblem to generally work as well as the MEA subproblem (Pourmohamad, 2016).

4.2 Joint Modeling of the Objective and Constraints

Constrained optimization is typically difficult because at least one of the constraint violations operates in opposition to the objective function, i.e., they are negatively correlated. We model the objective and constraint functions jointly because we want to be able to capture this negative dependency assumption, however, we could also forgo this assumption and model the objective and constraint functions independently using independent particle learning Gaussian process (PLGP) models (Gramacy and Polson, 2011) instead. There is an increase in the time and computational burden of joint modeling as compared to independent modeling of the objective and constraint functions, however, as seen in Pourmohamad and Lee (2016), there are significant gains in predictive accuracy and coverage by use of joint modeling when the objective and constraint functions are correlated. Given that the solution of the subproblem in Sections 4.1 relies heavily on the accuracy of the prediction of new candidate points, it is imperative that we do a good job in modeling the objective and constraint functions. Furthermore, modeling the objective and constraint functions independently implicitly assumes that the objective function and constraint functions are not correlated. Clearly the independence assumption can be violated in real-world applications and typically, a rational assumption would be to assume that the objective and constraint function are negatively correlated. Moreover, the benefits of using the PLMGP model in the statistical filter rather than the independent PLGP model are twofold. First, the PLMGP model in the statistical filter can improve our PAF calculations immensely when the objective function and constraint functions are correlated. Under the independence assumption the contours in our probability calculations will be always axis aligned which could lead to an under (or over) statement in our probability calculation. Allowing the objective and constraint functions to be modeled jointly allows for the contours to be angled and non-axis aligned (Figure 4) resulting in more accurate probability calculations when correlations are present.

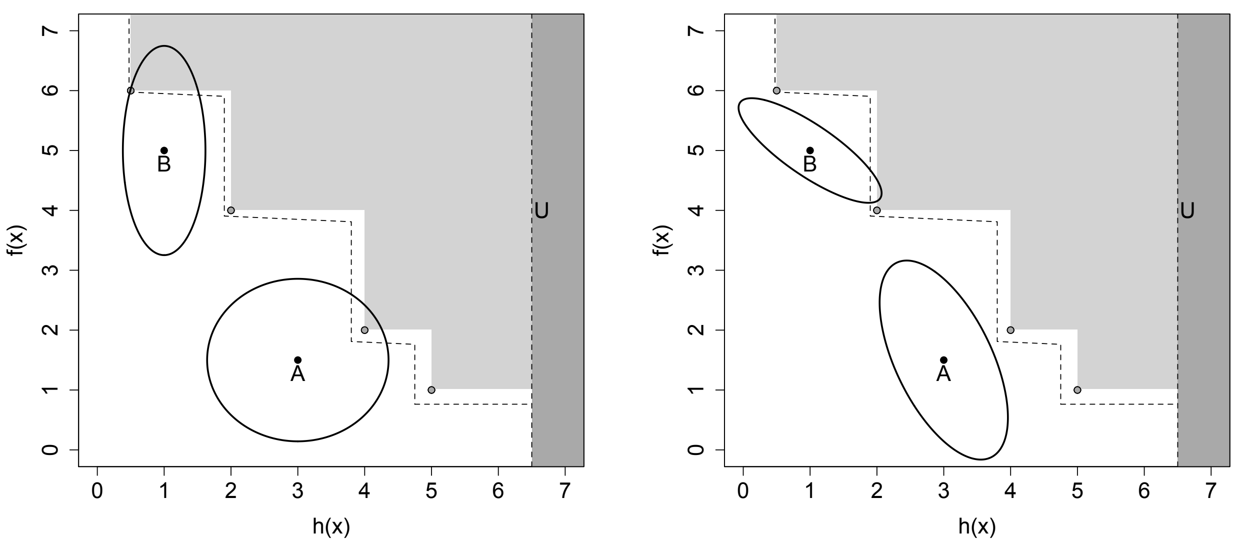

Figure 4: The probability acceptable to the filter subproblem. Selecting between candidate points A and B is determined by finding the area of their respective contours to the left of the filter envelope. Modeling the objective and constraint functions jointly (right) can lead to probability contours that are non-axis aligned as compared to independently (left) which are always axis aligned.

Second, when the objective and constraint function are correlated, the shared information in modeling the functions jointly can lead to better model fits, which ultimately leads to far fewer function evaluations of the computer model (see supplementary materials for an example). Being able to obtain good surrogate models with few function evaluations is especially important when modeling expensive black box functions where the time to evaluate function calls is a limiting factor. Thus, we implement the PLMGP model for solving the filter subproblem.

5 Illustrative Examples

We demonstrate the effectiveness of our statistical filter method on a synthetic test problem, a welded beam problem from the literature, as well as the real world pump-and-treat hydrology problem of Section 5.3. As a benchmark, we choose to compare the results of our method to the results of the methods used in Hedar (2004) and Gramacy et al. (2016). The approach used in Hedar (2004) was very different from the statistical approach we take and so we focus less on replicating the exact conditions of their experiment, and concentrate more on arriving at the same solution but in fewer objective function evaluations. However, in order to make a fair comparison of our methodology with Gramacy et al. (2016) we try to follow and mimic their initial conditions as closely as possible. For the toy problem, we initialize the PLMGP model with 10 random input-output pairs from $\mathcal{X}$ before starting our optimization algorithm. Likewise, we use our surrogate model to predict outputs based on a random set of 1000 candidate $\mathbf{x}$ locations in $\mathcal{X}$. Finer details for the initialization steps of the real-world hydrology computer experiment were not present in Gramacy et al. (2016) and so we chose to initialize the PLMGP model with 60 random input-output pairs from $\mathcal{X}$. Of note, our statistical filter method was designed to treat both the objective and constraint functions as outputs from a black box computer models, however, in Gramacy et al. (2016) they treat the objective function as being known. Therefore, to further facilitate a fair comparison of methods, we chose to also treat the objective functions as known and to only treat the constraint functions as black box outputs. This decision does not interfere with our statistical filter methodology as we may still treat the objective function as a black box output during the modeling and prediction stages, but then we shall deterministically predict the value of the objective function given that we now know the true form of it but still use our PLMGP model to predict the values of the constraint functions. In the welded beam problem (Hedar (2004)), we treat both the objective and constraint functions as unknown functions, i.e., arising from a black box computer model.

5.1 Results for the Toy Problem

Table 1 summarizes the results of a Monte Carlo experiment for the toy problem described in Section 1. Following the conditions of Gramacy et al. (2016), we conduct 100 repetitions of the Monte Carlo experiment and, on-average, arrive at the approximate solution to the constrained optimization problem after 100 input-output updates to the statistical filter algorithm.

n	25	50	100
95%	0.769	0.616	0.604
Average	0.710	0.606	0.600
5%	0.606	0.600	0.599

Table 1: After 100 updates: The progress in minimization for 100 Monte Carlo repetitions with random initial conditions using the statistical filter algorithm for the toy problem. The table shows the average best feasible minimum over the 100 runs, as well as 5^{th} and 95^{th} percentiles for the best feasible minima found.

Benchmarking our results to the analysis done in Gramacy et al. (2016) we compare the two plots in Figure 5. The left plot shows the results of the analysis done by Gramacy et al. (2016) while the right plot shows the results of the statistical filter method superimposed over the left plot for comparison. As shown by both the mean and the 95% credible intervals, the statistical filter method reliably finds the global minimum faster than any of the other methods in Gramacy et al. (2016).

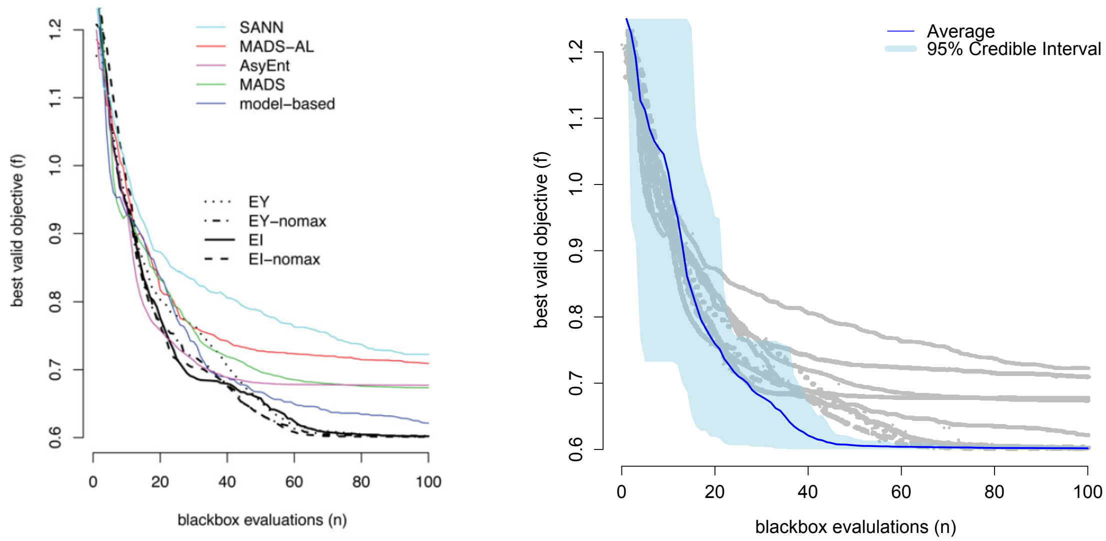

Figure 5: Results from running the statistical filter method (right side) against the comparative methods (left side) in Gramacy et al. (2016). The results are from running 100 Monte Carlo repetitions with random starting inputs. The plots show the best valid value found of the objective function over the 100 black box iterations.

5.2 Welded Beam Problem

The welded beam problem (Coello and Montes (2002); Hedar (2004)) has four inputs x_1, x_2, x_3, and x_4, and six constraints $c_1(\mathbf{x}), \ldots, c_6(\mathbf{x})$. The objective function, f, is the cost associated to construct a welded beam subject to constraints on sheer stress ($c_1(\mathbf{x})$, t), bending stress in the beam ($c_2(\mathbf{x})$, s), buckling load on the bar ($c_6(\mathbf{x})$, P_c), end deflection of the beam ($c_5(\mathbf{x})$, d), and side constraints ($c_3(\mathbf{x})$, $c_4(\mathbf{x})$). Here, Here, x_1 corresponds to the thickness of the welds, x_2 is the length of the welds, x_3 is the height of the beam, and x_4 is the width of the beam. We would like to minimize the objective, f, while simultaneously satisfying the constraints, $c_1, \ldots, c_6$, i.e.,

$$
\begin{aligned}
\min_{\mathbf{x}} \quad & f(\mathbf{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2) \\
\text{s.t.} & \\
P &= 6000,\ L = 14,\ E = 30 \times 10^6, G = 12 \times 10^6 \\
t_{\max} &= 13600,\ s_{\max} = 30000,\ x_{\max} = 10,\ d_{\max} = 0.25 \\
M &= P(L + x_2/2),\ R = \sqrt{0.25(x_2^2 + (x_1 + x_3)^2)} \\
J &= \sqrt{2}x_1x_2(x_2^2/12 + 0.25(x_1 + x_3)^2) \\
P_c &= \frac{4.013E}{6L^2}x_3x_4^3\left(1 - 0.25x_3\frac{\sqrt{E/G}}{L}\right) \\
t_1 &= P/(\sqrt{2}x_1x_2),\ t_2 = MR/J \\
t &= \sqrt{t_1^2 + t_1t_2x_2/R + t_2^2} \\
s &= 6PL/(x_4x_3^2) \\
d &= 4PL^3/(Ex_4x_3^3) \\
c_1(\mathbf{x}) &= (t - t_{\max})/t_{\max} \leq 0 \\
c_2(\mathbf{x}) &= (s - s_{\max})/s_{\max} \leq 0 \\
c_3(\mathbf{x}) &= (x_1 - x_4)/x_{\max} \leq 0 \\
c_4(\mathbf{x}) &= (0.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0)/5.0 \leq 0 \\
c_5(\mathbf{x}) &= (d - d_{\max})/d_{\max} \leq 0 \\
c_6(\mathbf{x}) &= (P - P_c)/P \leq 0 \\
0.125 &\leq x_1 \leq 10,\ 0.1 \leq x_i \leq 10 \text{ for } i = 2, 3, 4
\end{aligned}
$$

The optimal solution to the welded beam problem, as reported by Hedar (2004), is $f(\mathbf{x}) = 1.7250022$, which occurs at $\mathbf{x} = (0.2056, 3.4726, 9.0366, 0.2057)$.

One of the toughest challenges of the welded beam problem is that the set of possible feasible points is very small as compared to the entire input space. Taking a Latin hypercube sample of size 1,000,000 yields only 972 feasible points, or rather, only 0.0972% of the design points in a latin hypercube design (LHD) will be feasible for the problem. Having fewer than 1% of the initial set of inputs be feasible means that our PLMGP model needs to do a good job at prediction and uncertainty quantification of the objective and constraint functions so that the filter method can direct the search towards areas where the chance of encountering feasible points is high.

To solve the welded beam problem, we start with an initial sample of 40 inputs from a LHD over the input space and sequentially sample 960 more inputs. Using a LHD to predict new outputs is an inefficient space filling design in the context of the welded beam problem, and so, because our LHD provides us with so few feasible inputs, we decided to resample our 40 initial inputs over and over again until we obtained at least one feasible input. Once we obtain our initial sample, we follow the strategy of Taddy et al. (2009) and select the new candidate set of inputs to predict at from a LHD of size 500 times the input dimension augmented by an additional 10% of the candidate locations taken from a smaller LHD bounded to within 5% of the domain range of the current best feasible point. Using the approach of Taddy et al. (2009) better ensures that our search should continue to predict at some feasible inputs once we have found at least one feasible input. Note that these candidates are evaluated by the statistical surrogate model, and only the single most promising candidate is actually evaluated by the real model. Under these conditions, we ran our statistical filter algorithm and progressed the search for new inputs. Solving the PAF subproblem led to an optimal input configuration of $\mathbf{x} = (0.2057357, 3.4715605, 9.0349330, 0.2058335)$, which yields a feasible value of 1.725504. The solution we found was better than that of the best solution found in Coello and Montes (2002) and was comparable to the optimal solution reported in Hedar (2004) of 1.7250022. However, the average number of objective function evaluations for Coello and Montes (2002) and Hedar (2004) were 80,000 and 58,238, respectively, while our statistical filter algorithm only required 1000 function evaluations, and thus, was far superior in terms of number of evaluations it took to converge to an optimal solution of the problem. This is an important distinction to make since the number of function evaluations plays a critical role when the true computer model is expensive to evaluate and so a method that requires too many function evaluations to adequately solve the constrained optimization problem becomes infeasible. Of course if the computer model is not an expensive one (in this paper we assume it is) then methods that require a higher number of function evaluations may be preferable to our method if the actual computation time of those methods is minimal.

5.3 Pump-and-treat Hydrology Problem

Our motivating example, the goal of the pump-and-treat hydrology problem is to minimize the cost of running the pump-and-treat wells while containing both plumes. We reformulate this problem in the framework of a constrained optimization problem as

$$\min_{\mathbf{x}}\{f(\mathbf{x}) = \sum_{j=1}^{6} x_j : \mathbf{c}(\mathbf{x}) \leq 0,\ \mathbf{x} \in [0, 20 \cdot 10^4]^6\}, \tag{6}$$

where the objective f we wish to minimize is (known) linear and describes the cost required to operate the wells. The two plumes are contained when the constraint, $\mathbf{c}$, is met, and the input to the computer simulator are the six pumping rates $x_1, \ldots, x_6$ (which can be set between 0 and 20,000). The time it takes to run the computer simulator is nontrivial and so it is not feasible to run the computer simulator at every possible combination of inputs and find the one that optimizes the problem (6).

Table 2 summarizes the results of a Monte Carlo experiment for the real-world hydrology problem. Following Gramacy et al. (2016), we conduct 30 repetitions of a Monte Carlo experiment and, on-average, obtain a best value of 23,738 after 500 input-output updates to the statistical filter algorithm. This best average value is noticeably better than the best values found in Gramacy et al. (2016) and in the literature.

Comparing the two plots in Figure 6 we also see that the statistical filter method reliably decreases the objective function faster than any of the other methods. While a few of the comparator methods catch up around 100 to 200 iterations, the statistical filter method continues to improve after that point, finding a better overall solution by 500 iterations.

n	100	200	500
95%	34763	30220	24742
Average	28974	25604	23738
5%	27647	24464	23236

Table 2: After 500 updates: The progress in minimization for 30 Monte Carlo repetitions with random initial conditions using the statistical filter algorithm for the pump-and-treat hydrology problem. The table shows the average best feasible minimum over the 30 runs, as well as 5^{th} and 95^{th} percentiles for the best feasible minima found.

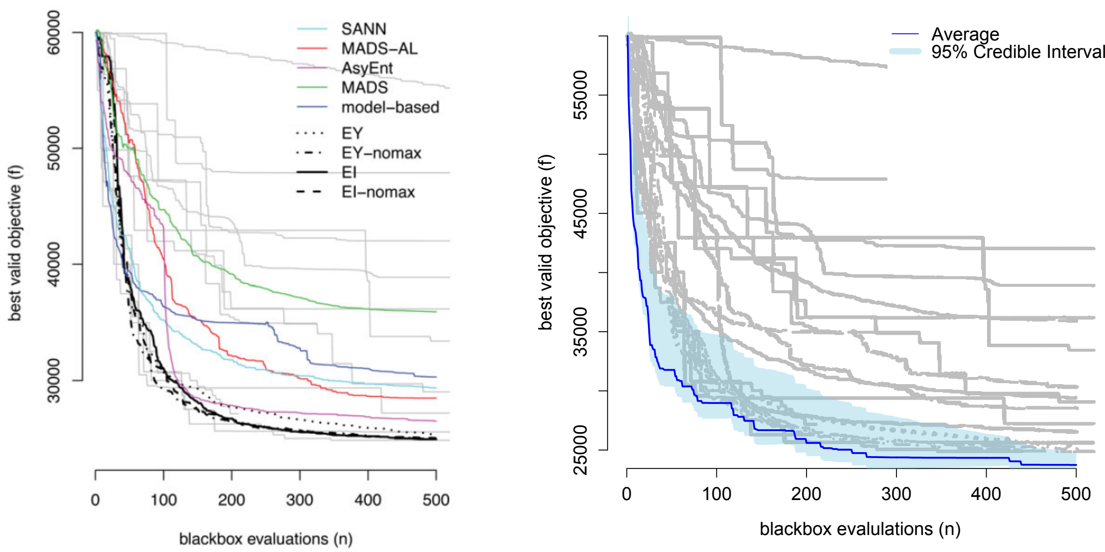

Figure 6: Results from running the statistical filter method (right side) against the comparative methods (left side) in Gramacy et al. (2016). The results are from running 30 Monte Carlo repetitions with random starting inputs. The plots show the best valid value found of the objective function over the 500 black box iterations.

6 Potential Extensions

We briefly mention two possible directions for future work. First, we hypothesize that we may be able to gain faster convergence to a global solution by incorporating a time dependent upper bound U_k on the constraint function that shrinks (or slides) over time. Currently within the statistical filter algorithms we use an upper bound U on the constraint functions that is set arbitrarily large or by some real-word physical property of the problem. In applications, setting the upper bound U this way works well, however, we think that by allowing U to shrink over time towards $h(\mathbf{x}) = 0$ we may be able to force the filter to find feasible solutions more rapidly.

Second, we note that there is a substantial amount of information lost in compressing the constraint values $\mathbf{c}(\mathbf{x})$ into a single nonnegative scalar $h(\mathbf{x})$. Under our statistical filter framework, we follow the accepted convention that we build our surrogate models in the constraint space and that we build the filter $\mathcal{F}$ in the two dimensional filter space. However, other than the fact that we saw little literature in the optimization community on filter methods that built the filter $\mathcal{F}$ in the higher dimensional constraint space, we see no apparent reason why this should not be explored. While it is beyond the scope of this paper, it would be interesting to examine a multidimensional statistical filter. The closest works in the literature are Gould et al. (2005a) and Gould et al. (2005b), who investigate the use of a multidimensional filter for solving systems of linear equations of the form $\mathbf{c}(\mathbf{x}) = \mathbf{0}$, and Shen et al. (2009), who built a three dimensional filter for solving constrained optimization problems where one constraint is an equality constraint and the other is an inequality constraint.

7 Discussion

Many complex phenomena in the world cannot be studied physically and so there is an ever growing need for computer models to study these complicated systems. As science evolves and questions become more complex and expensive, the need for efficient computer models, and surrogate models, becomes even more abundant. An important topic, constrained optimization of computer experiments is a very challenging problem. Computer models may be difficult to optimize because their output functions may be complex, multi-modal, and difficult to understand. The novelty of the work presented in this paper is in proposing a new methodology for solving constrained optimization problems by combining statistical modeling and nonlinear programming. The marriage of stochastic process models with filter methods is a powerful combination for understanding and optimizing black box computer models.

In order to solve constrained optimization problems for expensive black box computer experiments we embedded the joint stochastic process model of Pourmohamad and Lee (2016) within a filter algorithm that outperformed the established comparators. By building a joint surrogate model for the objective and constraint functions, we were able to establish a novel statistical metric for guiding the filter efficiently. We demonstrated the success of our new statistical filter algorithm on a suite of computer model optimization problems. Accompanying code for the independent case is given as it is relatively quick to run (in comparison to using joint modeling) and is easily extendable to other types of models.

8 Supplementary Materials

The Statistical Filter using Independent Gaussian Processes: The toy test problem is solved using independent Gaussian processes, rather than the PLMGP model, in order to provide a comparison of the impact in speed (as measured by function evaluations) and quality of the solution.

References

Audet, C. and Dennis, Jr., J. E. (2006). Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on Optimization*, 17:188–217.

Binois, M. and Picheny, V. (2016). Gpareto: Gaussian processes for pareto front estimation and optimization. *To appear: Journal of Statistical Software.*

Chin, C. M. and Fletcher, R. (2003). On the global convergence of an slp-filter algorithm that takes eqp steps. *Mathematical Programming*, 96:161–177.

Coello, C. A. C. and Montes, E. M. (2002). Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Advanced Engineering Informatics*, 16:2002.

Conn, A. R., Scheinberg, K., and Vicente, L. N. (2009). *Introduction to Derivative-Free Optimization*. SIAM, Philadelphia.

Conti, S. and O'Hagan, A. (2010). Bayesian emulation of complex multi-output and dynamic computer models. *Journal of Statistical Planning and Inference*, 140:640–651.

Emmerich, M. T. M., Deutz, A. H., and Klinkenberg, J. W. (2011). Hypervolume-based expected improvement: Monotonicity properties and exact computation. *IEEE Congress on Evolutionary Computation (CEC)*, pages 2147–2154.

Fang, K., Li, R., and Sudjianto, A. (2005). *Design and Modeling for Computer Experiments*. Chapman & Hall/CRC.

Fletcher, R. and Leyffer, S. (2002). Nonlinear programming without a penalty function. *Mathematical Programming*, 91:239–270.

Fletcher, R., Leyffer, S., and Toint, P. (1998). On the global convergence of a slp-filter algorithm. Technical report, Numerical Analysis Report NA/183, University of Dundee, UK.

Fletcher, R., Leyffer, S., and Toint, P. (2002). On the global convergence of a filter-sqp algorithm. *SIAM Journal on Optimization*, 13:44–59.

Fletcher, R., Leyffer, S., and Toint, P. (2006). A brief history of filter methods. Technical report, ANL/MCS-P1372-0906, Argonne National Laboratory, Mathematics and Computer Science Division.

Gould, N. I. M., Leyffer, S., and Toint, P. L. (2005a). A multidimensional filter algorithm for nonlinear equations and nonlinear least squares. *SIAM Journal on Optimization*, 21:17–38.

Gould, N. I. M., Sainvitu, C., and Toint, P. L. (2005b). A filter trust region method for unconstrained optimization. *SIAM Journal on Optimization*, 16:341–357.

Gramacy, R. B., Gray, G. A., Digabel, S. L., Lee, H. K. H., Ranjan, P., Wells, G., and Wild, S. M. (2016). Modeling an augmented lagrangian for blackbox constrained optimization. *Technometrics*, 58(1):1–11.

Gramacy, R. B. and Polson, N. G. (2011). Particle learning of Gaussian process models for sequential design and optimization. *Journal of Computational and Graphical Statistics*, 20:102–118.

Gupta, A. K. and Nagar, D. K. (2000). *Matrix Variate Distributions*. Chapman & Hall/CRC Press, Boca Raton, Florida.

Hedar, A. (2004). *Studies on Metaheuristics for Continuous Global Optimization Problems*. PhD thesis, Kyoto University, Japan.

Jones, D., Schonlau, M., and Welch, W. J. (1998). Efficient global optimization of expensive black box functions. *Journal of Global Optimization*, 13:455–492.

Keane, A. J. (2006). Statistical improvement criteria for use in multiobjective design optimization. *AIAA*, 44:879–891.

Kleijnen, J. P. (2015). *Design and Analysis of Simulation Experiments*. Springer International Publishing.

Lee, H. K. H., Gramacy, R. B., Linkletter, C., and Gray, G. A. (2011). Optimization subject to hidden constraints via statistical emulation. *Pacific Journal of Optimization*, 7:467–478.

Letham, B., Karrer, B., Ottoni, G., and Bakshy, E. (2018). Constrained bayesian optimization with noisy experiments. *Bayesian Anal.*

Lindberg, D. and Lee, H. K. H. (2015). Optimization under constraints by applying an asymmetric entropy measure. *Journal of Computational and Graphical Statistics*, 24:379–393.

Loeppky, J. L., Sacks, J., and Welch, W. J. (2009). Choosing the sample size of a computer experiment: A practical guide. *Technometrics*, 4:366–376.

Matott, L. S., Leung, K., and Sim, J. (2011). Application of matlab and python optimizers to two case studies involving groundwater flow and contaminant transport modeling. *Computers and Geosciences*, 37(1894–1899).

McKay, M. D., Conover, W. J., and Beckman, R. J. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21:239–245.

Parr, J. M., J., K. A., Forrester, A. I. J., and Holden, C. M. E. (2012). Infill sampling criteria for surrogate-based optimization with constraint handling. *Engineering Optimization*, 44:1147–1166.

Pourmohamad, T. (2016). *Combining Multivariate Stochastic Process Models with Filter Methods for Constrained Optimization*. PhD thesis, University of California, Santa Cruz.

Pourmohamad, T. and Lee, H. K. H. (2016). Multivariate stochastic process models for correlated responses of mixed type. *Bayesian Anal.*, 11(3):797–820.

Ribiero, A. A., Karas, E. W., and Gonzaga, C. C. (2008). Global convergence of filter methods for nonlinear programming. *SIAM Journal on Optimization*, 19:1231–1249.

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

Santner, T. J., Williams, B. J., and Notz, W. I. (2003). *The Design and Analysis of Computer Experiments*. Springer-Verlag, New York, NY.

Sasensa, M. J., Papalambros, P., and Goovaerts, P. (2002). Exploration of metamodeling sampling criteria for constrained global optimization. *Engineering Optimization*, 4:263–278.

Shen, C., Xue, W., and Pu, D. (2009). Global convergence of a tri-dimensional filter sqp algorithm based on the line search method. *Applied Numerical Mathematics*, 59:235–250.

Svenson, J. D. and Santner, T. J. (2012). Multiobjective optimization of expensive black-box functions via expected maximin improvement. Technical report, Department of Statistics, Ohio State University.

Taddy, M., Lee, H. K. H., Gray, G. A., and Griffin, J. D. (2009). Bayesian guided pattern search for robust local optimization. *Technometrics*, 51:389–401.

Wilson, B., Capoelleri, D., Simpson, T. W., and Frecker, M. (2001). Efficient pareto frontier exploration using surrogate approximations. *Optimization and Engineering*, 2:31–50.