# Scalable Constrained Bayesian Optimization

**David Eriksson** [1]   **Matthias Poloczek** [1]

## Abstract

The global optimization of a high-dimensional black-box function under black-box constraints is a pervasive task in machine learning, control, and engineering. These problems are difficult since the feasible set is typically non-convex and hard to find, in addition to the curses of dimensionality and the heterogeneity of the underlying functions. In particular, these characteristics dramatically impact the performance of Bayesian optimization methods, that otherwise have become the de-facto standard for sample-efficient optimization in unconstrained settings. Due to the lack of sample-efficient methods, practitioners usually fall back to evolutionary strategies or heuristics. We propose the scalable constrained Bayesian optimization (SCBO) algorithm that addresses the above challenges by data-independent transformations of the functions and follows the recent theme of local Bayesian optimization. A comprehensive experimental evaluation demonstrates that SCBO achieves excellent results and outperforms the state-of-the-art methods.

## 1. Introduction

The global optimization of a black-box objective function under black-box constraints has many applications in machine learning, engineering, and the natural sciences. Examples include fine-tuning the efficiency of a computing platform while preserving the quality of service; maximizing the power conversion efficiency of a solar cell material under stability and reliability requirements; optimizing the control policy of a robot under performance and safety constraints; tuning the performance of an aerospace design averaged over multiple scenarios while ensuring a satisfactory performance on each individual scenario (multi-point optimization). Moreover, industrial optimization tasks often have multiple objectives for which the transformation into

constraints is a popular approach. Here the functions that comprise the objective and the constraints are often given as black-boxes, i.e., upon their evaluation we receive an observation of the respective function, possibly with noise but without derivative information. All of the above examples have in common that their dimensionality, that is, the number of tunable parameters, is large: it is usually up to several dozens, which poses a substantial challenge for current methods in derivative-free optimization.

A high dimensionality makes black-box functions hard to optimize due to the curses of dimensionality (Powell, 2019), even without constraints. Moreover, these functions are often heterogeneous which poses a problem for surrogate-based optimizers. Black-box constraints make the task considerably harder since the set of feasible points is typically non-convex and hard to find, e.g., for control problems.

The main contributions of this work are as follows:

1. We propose the *scalable constrained Bayesian optimization* algorithm (SCBO), the first scalable algorithm for the optimization of high-dimensional expensive functions under expensive constraints. SCBO is also the first algorithm to support large batches for constrained problems with native support for asynchronous observations.

2. A comprehensive evaluation shows that SCBO outperforms previous state-of-the-art methods by far on high-dimensional constrained problems. Moreover, SCBO at least matches and often beats the best performer on low-dimensional instances.

3. Ablation studies investigate the individual contributions of the algorithmic ideas in SCBO to the overall performance.

4. We introduce two new high-dimensional constrained test problems that will be of independent interest given the novelty and impact of large-scale constrained Bayesian optimization.

The code for SCBO will be made available upon publication.

[1] Uber AI, San Francisco, California, USA. Correspondence to: David Eriksson <eriksson@uber.com> and Matthias Poloczek <poloczek@uber.com>.

## 1.1. Related Work

Bayesian optimization (BO) has recently gained enormous popularity for the global optimization of expensive black-box functions, see (Frazier, 2018; Shahriari et al., 2016) for an overview. While the vast majority of work focuses on unconstrained problems, aside from box constraints that describe the search space, a handful of articles consider the presence of black-box constraints. The seminal work of Schonlau et al. (1998) extends the expected improvement criterion (EI) to constraints by multiplying the expected improvement at some point $x$ over the best *feasible* point with the probability that $x$ itself is feasible, leveraging the independence between the objective function and the constraints. Later this cEI algorithm was rediscovered by (Gardner et al., 2014; Gelbart et al., 2014). Letham et al. (2019) extended the approach to noisy observations using quasi Monte Carlo integration and were the first to consider batch acquisition under constraints. Note that the computational cost of their acquisition function grows with the number of observations and thus is not applicable in large-scale settings. Moreover, for noise-free observations, as for the benchmarks that we study, their approach reduces to the original cEI.

Hernández-Lobato et al. (2016) extended predictive entropy search (Hernández-Lobato et al., 2014) to constraints and detailed how to make the sophisticated approximation of the entropy reduction computationally tractable in practice. Their PESC algorithm usually achieves great results and is widely considered the state-of-the-art for constrained BO despite its rather large computational costs. Picheny (2014) considered the volume of the admissible excursion set under the best known feasible point as a measure for the uncertainty over the location of the optimizer. Then he samples a point that yields a maximum approximate reduction in volume.

By lifting constraints into the objective via the Lagrangian relaxation, Gramacy et al. (2016) took a different approach. Note that it results in a series of unconstrained optimization problems that are solved by vanilla BO. SLACK of Picheny et al. (2016) refined this idea by introducing slack variables and showed that this augmented Lagrangian achieves a better performance for *equality* constraints. Very recently, Ariafar et al. (2019) used the ADMM algorithm to solve an augmented Lagrangian relaxation. All these algorithms use the EI criterion.

Traditionally, BO, with or without constraints, has been limited to problems with a small number of decision variables, usually at most 15, and a budget of no more than a couple of hundred samples. Recent work has started exploring scalable BO for budgets with tens of thousands of samples. Hernández-Lobato et al. (2017) extended Thompson sampling (Thompson, 1933) to large batch sizes and a Bayesian

neural network for the surrogate to maintain scalability (see also (Kandasamy et al., 2018)). Wang et al. (2018) proposed the EBO algorithm that partitions the search space to achieve scalability. Eriksson et al. (2019) abandoned a global surrogate and instead maintained several local models that move towards better solutions. Their TURBO algorithm applies a bandit approach to allocate samples efficiently between these local searches. BO has also been investigated for high-dimensional settings with small sampling budgets, albeit without constraints, e.g., see (Binois et al., 2015; 2020; Eriksson et al., 2018; Mutny & Krause, 2018; Nayebi et al., 2019; Oh et al., 2018; Rolland et al., 2018; Wang et al., 2016).

To the best of our knowledge, the present work is the first to explore Bayesian optimization for high-dimensional constrained problems.

The constrained optimization of black-box functions has also been studied in the field of evolutionary strategies and in operations research. CMA-ES is one of the most powerful and versatile evolutionary strategies. It uses a covariance adaptation strategy to learn a second-order model of the objective function. CMA-ES handles constraints by the 'death penalty' that sets the fitness value of infeasible solutions to zero (Kramer, 2010; Arnold & Hansen, 2012). COBYLA (Powell, 1994) and BOBYQA (Powell, 2007) maintain a local trust region and thus perform a local search. In our experience, this strategy scales well to high dimensions, with COBYLA having an edge due to its support for nonlinear constraints. We compare to cEI that we extended to high-dimensional domains, PESC, SLACK, CMA-ES, and COBYLA that are representative for the above lines of work.

**Structure of the article.** The remainder of the article is structured as follows. In the next section we define the problem formally. The SCBO algorithm is presented in Sect. 3 and compared to a representative selection of methods in Sect. 4. Sect. 5 summarizes the conclusions and discusses ideas for future work.

## 2. The Model

The goal is to find an optimizer

$$\operatorname*{argmin}_{x \in \Omega} f(x) \ \text{ s.t. } \ c_1(x) \leq 0, \ldots, c_m(x) \leq 0 \qquad (1)$$

where $f \colon \Omega \to \mathbb{R}$ and $c_\ell \colon \Omega \to \mathbb{R}$ for $1 \leq \ell \leq m$ are black-box functions defined over a compact set $\Omega \subset \mathbb{R}^d$. The term black-box function means that we may query any $x \in \Omega$ to observe the values under the objective function $f$ and all constraints, possibly with noise, but no derivative information. Specifically, we suppose that we observe an i.i.d. $(m+1)$-dimensional vector with the $\ell$-th entry given by $y_0(x) \sim \mathcal{N}(f(x), \lambda_0(x))$ and $y_\ell(x) \sim \mathcal{N}(c_\ell(x), \lambda_\ell(x))$ for $1 \leq \ell \leq m$. Here the $\lambda$'s give the variance of the observa-

tional noise and are supposed to be known. In practice, we estimate the $\lambda$'s along with the hyperparameters of the surrogate model. Note that we may rescale the search space $\Omega$ w.l.o.g. to the unit hypercube $[0,1]^d$. If all functions are observed without noise, then our goal is to find a feasible point with minimum value under the objective function. For noisy functions, we wish to find a point with best expected objective value under all points that are feasible with probability at least $1-\delta$, where $\delta$ is set based on the context, e.g., the degree of risk aversion (cp. Hernández-Lobato et al. (2016)).

## 3. Scalable Constrained Bayesian Optimization (`SCBO`)

We propose the *Scalable Constrained Bayesian Optimization* (`SCBO`) algorithm. `SCBO` follows the paradigm of the generic BO algorithm (Frazier, 2018; Shahriari et al., 2016) and proceeds in rounds. In each round, `SCBO` selects a batch of $q$ points in $\Omega$ that are then evaluated in parallel. Note that `SCBO` is easily extended to asynchronous batch evaluations.

`SCBO` employs the *trust region* approach introduced by Eriksson et al. (2019) that confines samples locally. This addresses common problems of Bayesian optimization in high-dimensional settings, where popular acquisition functions spread out samples due to the inherently large uncertainty and thus fail to zoom in on promising solutions. Moreover, for the popular Matérn kernels, the covariance under the prior is essentially zero for two points if they differ substantially in one coordinate only. The use of trust regions results in more exploitation and often a better fit for the local surrogate. `SCBO` maintains the invariant that the trust region is centered at a point of maximum utility. Thus, the trust region is moved through the domain $\Omega$ as better points are discovered.

The generalization to black-box constraints poses additional fundamental problems that were not considered by Eriksson et al. (2019). For many problems it is hard to even find a feasible solution, since the feasible set is typically non-convex. An investigation in Sect. 4 demonstrates the difficulty of this task. Moreover, the various black-box functions can vary drastically in their characteristics. We will provide examples where some constraints exhibit a huge variability whereas others are smooth. `SCBO` applies tailored transformations that account for the specific roles of the objective and the constraints.

**Extending Thompson sampling to constrained optimization.** `SCBO` extends Thompson sampling (`TS`) to black-box constraints, and is to the best of our knowledge the first to do so. `TS` scales to large batches at low computational cost and is at least as effective as `EI`, as we demonstrate

below. To select a point for the next batch, `SCBO` samples $r$ candidate points in $\Omega$ using a Sobol sequence (see the supplement for details).

Let $x_1, \ldots, x_r$ be the sampled candidate points. Then `SCBO` samples a *realization* $(\hat{f}(x_i), \hat{c}_1(x_i), \ldots, \hat{c}_m(x_i))^T$ for all $x_i$ with $1 \leq i \leq r$ from the respective posterior distributions on the functions $f, c_1, \ldots, c_m$. Let $\hat{F} = \{x_i \mid \hat{c}_\ell(x_i) \leq 0 \text{ for } 1 \leq \ell \leq m\}$ be the set of points whose realizations are feasible. If $\hat{F} \neq \emptyset$ holds, `SCBO` selects an $\operatorname{argmin}_{x \in \hat{F}} \hat{f}(x)$. Otherwise `SCBO` selects a point of *minimum total violation* $\sum_{\ell=1}^m \max\{\hat{c}_\ell(x), 0\}$, breaking ties via the sampled objective value. While we found that this natural selection criterion is able to identify a feasible point quickly for smooth constraints, we observed that it struggles when functions vary significantly in their magnitudes.

**Transformations of Objective and Constraints.** The key observations are that for the objective function we are particularly interested in the location of possible optima, whereas for constraints we are interested in identifying feasible areas, i.e., when the function values become negative. Thus, we apply transformations that emphasize these areas particularly; see also Fig. 1. To the objective function, we apply a *Gaussian copula* (e.g., see Wilson & Ghahramani, 2010). The Gaussian copula first maps all observations under the objective to quantiles using the empirical CDF. Then it maps the quantiles through an inverse Gaussian CDF. Note that this procedure magnifies differences between values that are at the end of the observed range, i.e., minima or maxima. It affects the observed values but not their location. Finally, we apply Gaussian process regression to the mapped observations, as usual. For the constraints we employ the `bilog` transformation: $\operatorname{bilog}(y) = \operatorname{sgn}(y) \ln(1 + |y|)$ for a scalar observation $y$. It magnifies the range around zero to emphasize the change of sign that is decisive for feasibility.

**Maintaining the trust region.** A trust region is initialized as a hypercube with side length $L = L_{\text{init}}$. We count for each trust region the number of *successes* $n_s$ and *failures* $n_f$ since it was resized last. First suppose that all functions are observed without noise. Then a success occurs when `SCBO` observes a better point; by construction, this point must be inside the trust region. A failure happens when no point in the batch is better than the current center of the trust region. The center $C$ of the trust region is chosen as follows. We select the best feasible point for $C$ if any. Otherwise we pick a point with minimum total violations, again breaking ties via the objective. Note that we use (transformed) *observations* from the black-box functions, not *realizations* from the posterior. Thus, the center is moved to a new point whenever a success occurs. The trust region is resized as follows: if $n_s = \tau_s$ then the side length is set to $L = \min\{2L, L_{\max}\}$ and we reset $n_s = 0$. If $n_f = \tau_f$, then we set $L = L/2$ and
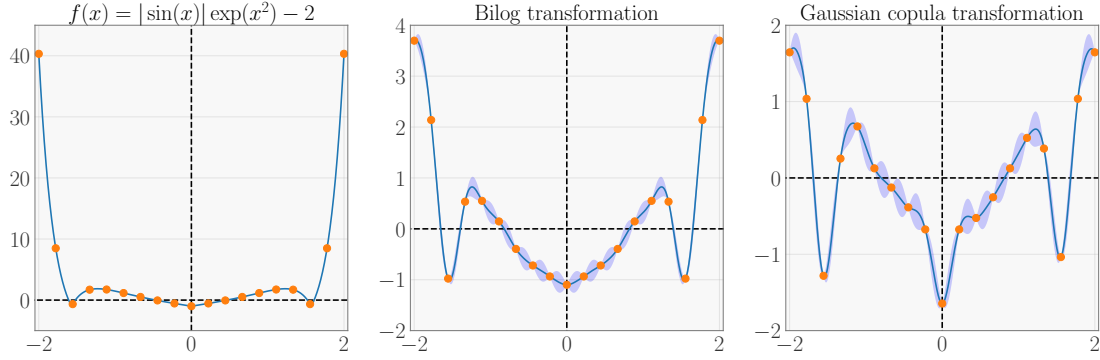
*Figure 1.* **(Left)** The original function where the distance to the origin varies considerably for the observations. If this was a constraint, the feasible region, denoted by the change of the sign, would be hard to detect. If it was the objective function, we would struggle to identify the minima, since the observations in the center differ only slightly and are considerably smaller in absolute value than the observations on the boundary. **(Middle)** The bilog transformation stretches out observations around zero, thereby making it easier to detect feasible areas. Note that a GP has been fitted to the observations given by the orange points in the middle and the right plot. The blue line depicts the posterior mean and the shaded area gives the posterior uncertainty of the GP. **(Right)** The copula transformation magnifies values that are at the ends of the observed spectrum, which facilitates the task of finding optima. Note that these transformations are advantageous over a naive standardization of each function as the latter is insensitive to the areas of interest.

$n_f = 0$. If the side length drops below a set threshold $L_{\min}$, then we initialize a new trust region. For noisy functions we follow the same rules, and use the posterior mean of GP model instead of the observed value. Note that the procedure for maintaining the trust regions follows (Eriksson et al., 2019) and is described here for completeness. We emphasize that SCBO uses the default setting of TURBO for all hyperparameters. In the next section we demonstrate that it achieves excellent performance across all benchmarks.

### 3.1. Summary of the SCBO Algorithm

We summarize the SCBO algorithm.

1. Evaluate an initial set of points and initialize the trust region at a point of maximum utility. Then estimate the hyperparameters of the GP prior.

2. Until the budget for samples is exhausted:

   (a) Generate $r$ candidate points.
   (b) For each of the $q$ points of the next batch we sample a realization
   $\{(\hat{f}(x_i), \hat{c}_1(x_i), \ldots, \hat{c}_m(x_i))^T \mid 1 \le i \le r\}$ from the posterior over each candidate and add a point of maximum utility to the batch.
   (c) Update the $m+1$ posteriors with the $q$ new observations.
   (d) Adapt the trust region by moving the center as described above. Update the counters $n_s$, $n_f$, and size $L$. If $L < L_{\min}$, initialize new trust region.

3. Recommend an optimal feasible point (if any).

For noisy functions, we recommend a point of minimum posterior mean under all points that are feasible with prob. at least $1 - \delta$ (if any). We note in passing that SCBO is *consistent*, that is, it will converge to a global optimum as the number of samples tends to infinity. The proof was deferred to the supplement due to space constraints.

## 4. Experimental Evaluation

We compare SCBO to the state-of-the-art: PESC (Hernández-Lobato et al., 2016) in Spearmint, cEI[1] (Schonlau et al., 1998; Gardner et al., 2014), SLACK (Picheny et al., 2016) in laGP, the implementation of (Jones et al., 2014) for COBYLA (Powell, 1994), CMA-ES (Hansen, 2006) in pycma, and random search (RS). Please see Sect. 1.1 for a discussion of these methods.

**The Benchmarks.** We evaluate the algorithms on a comprehensive selection of benchmark problems. First, we consider four low-dimensional problems in Sect. 4.1: a 3D tension-compression string problem with four constraints, a 4D pressure vessel design with with four constraints, a 4D welded beam design problem with five constraints, and a 7D speed reducer problem with eleven constraints. Next we consider the 10D Ackley problem with two constraints in Sect. 4.2 that is particularly interesting because of its small feasible region. Then we study four large-scale problems: the 30D Keane bump function with two constraints in Sect. 4.3, a 12D robust multi-point optimization problem with a varying number of constraints in Sect. 4.4, a 60D

---

[1]We will make the code for SCBO and cEI available upon publication.

trajectory planning problem with 15 constraints in Sect. 4.5, and a 124-dimensional vehicle design problem with 68 constraints in Sect. 4.6. `PESC` and `SLACK` do not scale to large-scale high-dimensional problems and large batch sizes and are therefore omitted for these problems. Note that all benchmarks have multi-modal objective functions and functions are observed without noise.

To compare feasible and infeasible solutions, we adopt the rationale of Hernández-Lobato et al. (2016) that any feasible solution is preferable over an infeasible one and thus assign a default value to infeasible solutions that is set to the largest found objective value for the respective benchmark. Performance plots show the mean with one standard error. All methods start with an initial set of points given by a Latin hypercube design (LHD). `CMA-ES` and `COBYLA` are initialized from the best point in this design. Recall that `SCBO` applies transformations to the functions. In the supplement we investigate the performances of the baselines under these transformations and show that `SCBO` performs best.

### 4.1. Physics Test Problems

We evaluate the algorithms on a variety of physics problems. We use a budget of 100 evaluations, batch size $q = 1$, and 10 initial points. Fig. 2 summarizes the results for the four test problems. `SCBO` outperforms all baselines on the 3D tension-compression string problem (Hedar & Fukushima, 2006): it found feasible solutions in all runs and consistently obtained excellent solutions. `PESC` and `cEI` are not competitive. Their performance is only slightly better than `RS` search on this problem. For the 4D pressure vessel design problem (Coello & Montes, 2002), `SCBO` obtains the best solutions followed by `cEI`, `PESC`, and `COBYLA`. `SCBO` also performs best for the 4D welded beam design problem (Hedar & Fukushima, 2006), followed by `cEI`. `SCBO` and `PESC` obtain excellent results for the 7D speed reducer design problem (Lemonge et al., 2010).

### 4.2. The 10D Ackley Function

We study the performance on the 10D Ackley function on the domain $[-5, 10]^{10}$ with the constraints $c_1(x) = \sum_{i=1}^{10} x_i \leq 0$ and $c_2(x) = \|x\|_2 - 5 \leq 0$. The Ackley function has a global optimum with value zero at the origin. Note that Ackley is a challenging problem where the probability of randomly selecting a feasible point is only $2.2 \cdot 10^{-5}$. We use a budget of 200 function evaluations, batch size one, and ten initial points. Fig. 3 shows that `COBYLA` initially makes good progress but is eventually outperformed by `SCBO` which achieves the best performance. `PESC` performs well, but is computationally costly: a run with `PESC` took 3 hours, while the other methods ran in minutes.

### 4.3. The 30D Keane Bump Function

The Keane bump function is a common test function for constrained global optimization (Keane, 1994). This function has two constraints. We consider $d = 30$ and large batches of size 50 for `SCBO`, `cEI`, and `CMA-ES`. Each method uses 100 initial points. `COBYLA` does not support batching samples and thus samples sequentially, which is an advantage as it can leverage more data for acquisition. However, Fig. 3 shows that nonetheless `COBYLA` is not competitive. We see that `SCBO` clearly outperforms the other algorithms for this large-scale high-dimensional benchmark. As stated above, we cannot compare to `PESC` and `SLACK` on high-dimensional large-scale benchmarks due to their computational complexity.

### 4.4. Robust Multi-point Optimization

Multi-point optimization is an important task in aerospace engineering (Liem et al., 2014; 2017; Martins, 2018). Here, a design is optimized over a collection of flight conditions. Multi-point optimization produces designs with better practical performance by addressing the issue that tuning a design for a single scenario often leads to designs with poor off-scenario performance (Jameson, 1990; Cliff et al., 2001). In this section we propose a *robust multi-point optimization problem*. The goal is to optimize the performance of the design $x$ averaged over all $m$ scenarios (potentially weighted by importance), subject to individual constraints that assert an acceptable performance for each scenario.

Our problem is derived from the lunar lander problem, where the goal is to find a 12D controller that maximizes the reward averaged over $m$ terrains. We extend this problem by adding $m$ constraints that assert that no individual reward is below 200, which guarantees that the lunar lands successfully. Note that without these constraints, the algorithms often produce control policies that occasionally crash the lander. Adding the constraints produces robust policies which are strictly preferable from the passengers point of view. We evaluate the algorithms with 1000 samples, batch size $q = 50$, and 50 initial points for three experiments that differ in the number of constraints: $m = 10$, $m = 30$, and $m = 50$. We performed ten replications. Tab. 1 summarizes the results.

We see that `SCBO` found a feasible controller for all 10 runs when $m = 10$. `cEI` and `CMA-ES` obtained worse solutions and found feasible points in 8/10 and 6/10 runs respectively. `COBYLA` and `RS` struggled visibly. We note that `SCBO` clearly outperformed the other methods for the more constrained settings with $m = 30$ and $m = 50$. Note that the best reward found by each algorithm clearly decreases when $m$ increases. The problem becomes harder when we add more scenarios, since feasible solutions for $m = 10$ may not satisfy all constraints for $m = 30$ or $m = 50$. Ad-
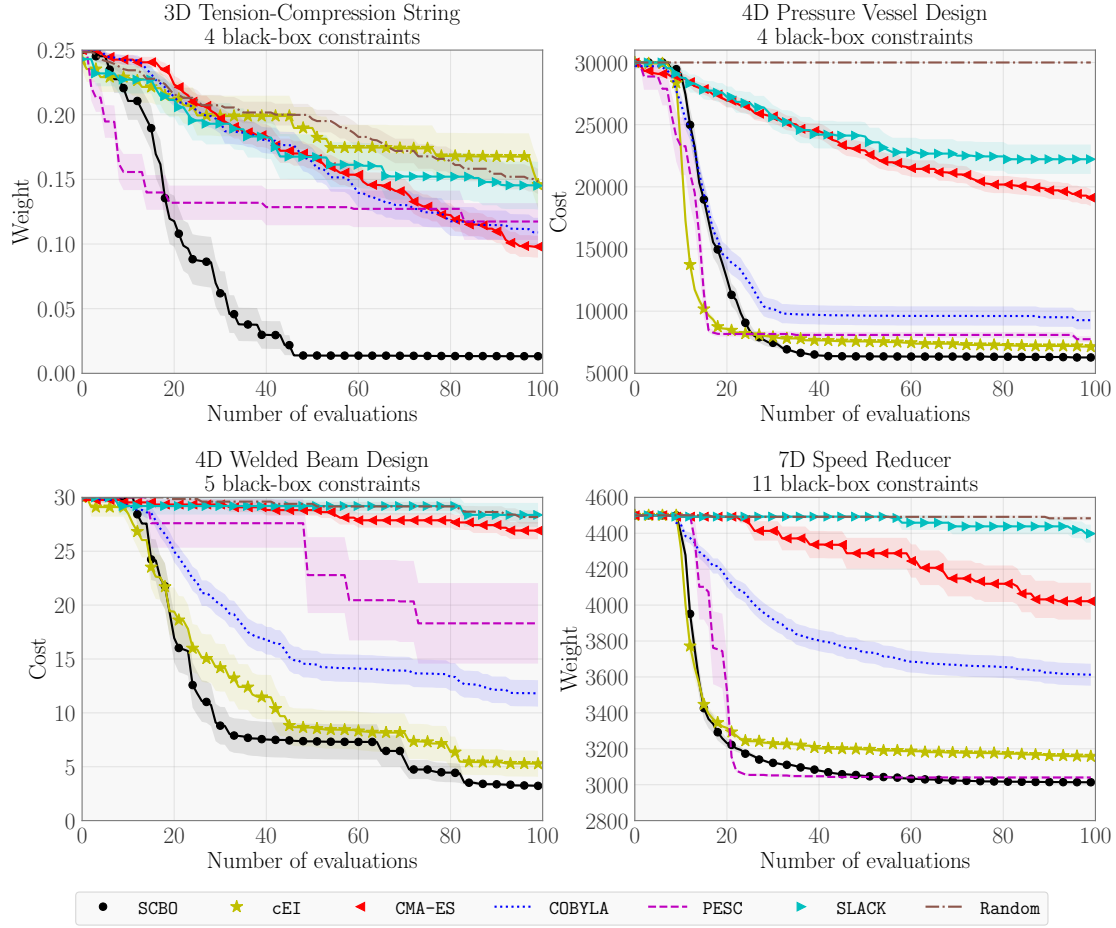
Figure 2. **(Upper left)** SCBO outperforms the other methods on the Tension-compression string problem. **(Upper right)** SCBO finds the best solutions on the pressure vessel design problem, followed by cEI, PESC, and COBYLA. **(Lower left)** SCBO performs best on the welded beam design problem. **(Lower right)** SCBO and PESC perform the best on the speed reducer problem.
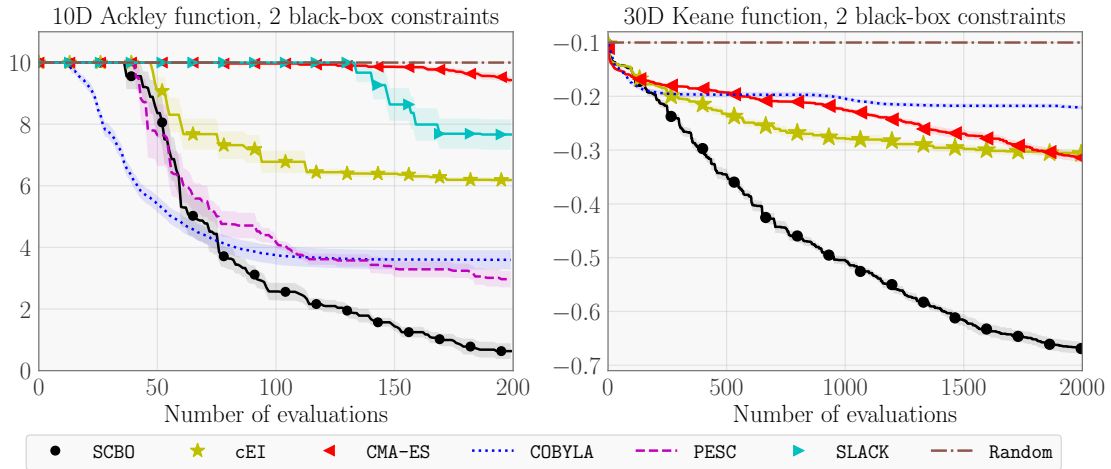


Figure 3. **(Left)** 10D Ackley function with two constraints. SCBO consistently finds solutions close to the global optimum. **(Right)** 30D Keane function with two constraints. SCBO clearly outperforms the other methods from the start.

| $m$ | | | SCBO | cEI | CMA-ES | COBYLA | RS |
|---|---|---|---|---|---|---|---|
| | | Best | **321.3** | 306.3 | 300.7 | 307.8 | NA |
| | | Worst | **302.8** | 278.2 | 290.3 | 269.9 | NA |
| 10 | | Median | **318.0** | 293.0 | 295.7 | 288.9 | NA |
| | | Feasible | **10/10** | 8/10 | 6/10 | 2/10 | 0/10 |
| | | Best | **314.1** | 293.7 | 293.1 | NA | NA |
| | | Worst | **294.8** | 278.2 | 274.2 | NA | NA |
| 30 | | Median | **308.7** | 287.0 | 280.3 | NA | NA |
| | | Feasible | **9/10** | 4/10 | 3/10 | 0/10 | 0/10 |
| | | Best | **309.1** | 284.3 | 284.0 | NA | NA |
| | | Worst | **289.5** | 272.6 | 272.6 | NA | NA |
| 50 | | Median | **304.0** | 278.3 | 278.3 | NA | NA |
| | | Feasible | **9/10** | 4/10 | 2/10 | 0/10 | 0/10 |

*Table 1.* Results for the 12D multi-point optimization problem. The best result in every row is highlighted. We observe that SCBO finds the best robust policies over ten runs and scales best to larger numbers of constraints.

ditionally, we investigated the hardness of finding a feasible controller by random sampling: out of 50K samples, not a single policy satisfied all $m = 10$ constraints!

### 4.5. 60D Rover Trajectory Planning

We study a 60D route planning problem adapted from (Wang et al., 2018). The task is to position 30 waypoints that lead a rover on a path of minimum cost from its starting position to its destination, while avoiding collisions with obstacles. We propose a constraint-based formulation where each of the $m = 15$ constraints is met if and only if the rover does not collide with any of its associated impassable obstacles. The exact formulation of these constraints is given in the supplementary material. Note that the resulting feasible space is non-convex! Fig. 4 (middle) illustrates the setup and the best trajectory found by SCBO. Note that there are two types of terrain that vary in their cost: the green terrain can be traversed at cost zero, whereas the more difficult yellow terrain inflicts a certain cost. This problem turns out to be challenging for small sampling budgets. Thus, we have evaluated SCBO, cEI, CMA-ES, COBYLA, and RS for a total of 5000 evaluations with batch size $q = 100$ and 100 initial points. Fig. 4 (left) summarizes the performances. We see that SCBO outperforms the other methods by far on this hard benchmark.

### 4.6. 124D Vehicle Design with 68 Constraints

We evaluate the algorithms on a 124-dimensional vehicle design problem MOPTA08 (Anjos, 2008), where the goal is to minimize the mass of a vehicle subject to 68 constraints. The 124 variables describe gages, materials, and shapes. Here we ran all experiments with a budget of 2000 samples, batch size $q$=10, and 130 initial points. We point out that this benchmark showcases the scalability of the implementation of SCBO that uses GPyTorch (Gardner et al., 2018) and KeOPS (Charlier et al., 2018) to fit the 69 GP models in

a batch; see the supplement for details. Fig. 4 (right) shows SCBO, cEI, CMA-ES, and COBYLA over ten runs. SCBO finds a feasible point in 10/10 runs and the best solution has value 255.5. COBYLA found a feasible point in 4/10 runs, one which had objective value 241.8, which is better than any solution found by SCBO. The other baselines failed to find a feasible solution.

### 4.7. Ablation studies

We investigate how the various ideas in SCBO contribute to the excellent overall performance: specifically, how does the application of i) the transformations (Transformed, Untransformed), ii) the acquisition criterion (TS or EI), and iii) the use of a trust region (TR, Global) affect the performance. Keep in mind that the choice whether to use a trust region determines if samples are locally confined. Fig. 5 summarizes the performances of all eight combinations on three benchmarks. On the left, we see that the use of the trust region is critical for the 30D Keane function. Approaches that do not use a trust region struggle. Moreover, the transformation provides an additional gain, whereas the choice of the acquisition function has no noticeable effect. The center plot is for the 2D toy problem proposed by Hernández-Lobato et al. (2016) that has a smooth objective function and two easy constraints. Here BO without a trust region and EI should shine, and this is indeed what the observations indicate. The right plot considers the 5D Rosenbrock function with two poorly scaled constraints. Again the trust region is performance critical. Interestingly, TS achieves significantly better results than EI.

## 5. Conclusions

We studied the task of optimizing a black-box objective function under black-box constraints that has numerous applications in machine learning, control, and engineering. We found that the existing methods struggle in the face of multiple constraints and more than just a few decision variables. Therefore, we proposed the *Scalable Constrained Bayesian Optimization* (SCBO) algorithm that leverages tailored transformations of the underlying functions together with the trust region approach of Eriksson et al. (2019) and Thompson sampling (TS) to scale to high-dimensional spaces and large sampling budgets.

We performed a comprehensive experimental evaluation that compared SCBO to the state-of-the-art from machine learning, operations research, and evolutionary algorithms on a variety of benchmark problems that span control, multi-point optimization, and physics. We found that SCBO outperforms the state-of-the-art on high-dimensional benchmarks, and matches or beats the performance of the best baseline otherwise. We believe that these benchmarks are of independent interest. In the supplement, we also provide
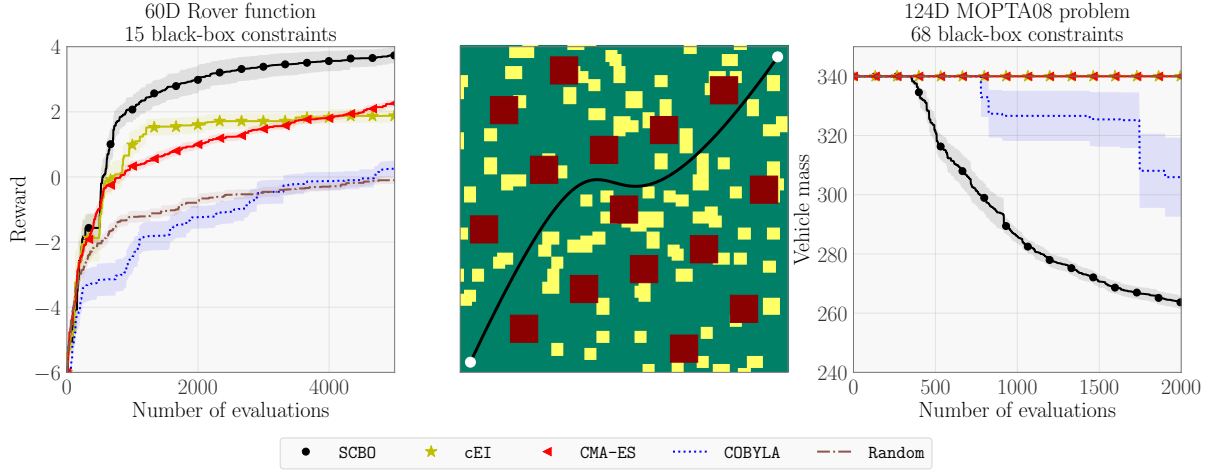
*Figure 4.* **(Left) 60D trajectory planning with 15 constraints:** SCBO finds excellent solutions quickly and outperforms the other methods. **(Middle) Illustration of the trajectory planning problem**: The black line is the best trajectory found by SCBO with a reward of 4.93. The green area can be traversed at no cost. Yellow squares denote terrain that inflicts a cost upon traversal. Red squares are impassable obstacles. **(Right) 124D Vehicle Design with 68 Constraints**: SCBO finds a feasible point in 10/10 runs and consistently finds good solutions.
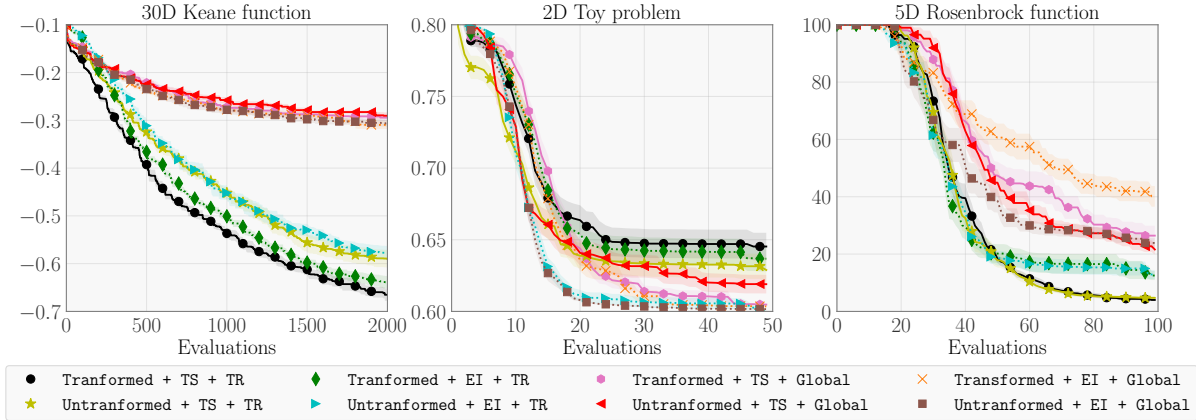


*Figure 5.* Ablation study where we investigate the effect of using the robust transformation, different acquisition functions (TS/EI), and using a trust region (TR). **(Left)** 30D Keane function with 2 black-box constraints. **(Middle)** 2D Toy problem with 2 black-box constraints of Hernández-Lobato et al. (2016). **(Right)** 5D Rosenbrock function with 2 poorly scaled black-box constraints.

an efficient GPU implementation of SCBO based on batch-GPs and a formal proof that SCBO converges to a global optimum.

For future work, we are interested in applications where the objective and constraints have substantial correlations. For example, consider the design of an aircraft wing: here the aerodynamic performance (e.g., lift and drag), the structural stability, and the fuel-burn will be related. If the airfoils geometry generates turbulent structures, the drag will increase and the fuel burn will suffer. The heterogeneity of the involved functions may make the adoption of a multi-output Gaussian process challenging. We believe that leveraging these correlations may pave an avenue towards solving problems with hundreds of constraints more efficiently. Constraints also arise naturally for combinatorial black-box functions (Baptista & Poloczek, 2018; Oh et al., 2019) that have exciting applications in engineering and science.

Moreover, we look forward to inter-disciplinary applications: SCBO's ability to optimize high-dimensional constrained problems will allow to optimize an airfoil described by a mesh or the parameters controlling a chemical process, e.g., for growing nanotubes or when searching for a solar cell material (Herbol et al., 2018; Ortoll-Bloch et al., 2019).

# References

Anjos, M. The MOPTA 2008 benchmark, 2008. URL http://www.miguelanjos.com/jones-benchmark.

Ariafar, S., Coll-Font, J., Brooks, D., and Dy, J. Admmbo: Bayesian optimization with unknown constraints using ADMM. *Journal of Machine Learning Research*, 20(123): 1–26, 2019.

Arnold, D. V. and Hansen, N. A (1+ 1)-CMA-ES for constrained optimisation. In *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, pp. 297–304. ACM, 2012.

Baptista, R. and Poloczek, M. Bayesian optimization of combinatorial structures. In *International Conference on Machine Learning*, pp. 462–471, 2018.

Binois, M., Ginsbourger, D., and Roustant, O. A warped kernel improving robustness in Bayesian optimization via random embeddings. In *International Conference on Learning and Intelligent Optimization*, pp. 281–286. Springer, 2015.

Binois, M., Ginsbourger, D., and Roustant, O. On the choice of the low-dimensional domain for global optimization via random embeddings. *Journal of global optimization*, 76(1):69–90, 2020.

Charlier, B., Feydy, J., and Glaunés, J. A. KeOps, 2018. URL https://github.com/getkeops/keops.

Cliff, S. E., Reuther, J. J., Saunders, D. A., and Hicks, R. M. Single-point and multipoint aerodynamic shape optimization of high-speed civil transport. *Journal of Aircraft*, 38(6):997–1005, 2001.

Coello, C. A. C. and Montes, E. M. Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Advanced Engineering Informatics*, 16(3):193–203, 2002.

Dong, K., Eriksson, D., Nickisch, H., Bindel, D., and Wilson, A. G. Scalable log determinants for Gaussian process kernel learning. In *Advances in Neural Information Processing Systems*, pp. 6327–6337, 2017.

Eriksson, D., Dong, K., Lee, E., Bindel, D., and Wilson, A. G. Scaling Gaussian process regression with derivatives. In *Advances in Neural Information Processing Systems*, pp. 6867–6877, 2018.

Eriksson, D., Pearce, M., Gardner, J., Turner, R. D., and Poloczek, M. Scalable global optimization via local Bayesian optimization. In *Advances in Neural Information Processing Systems 32*, pp. 5497–5508. 2019.

Frazier, P. I. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

Gardner, J., Pleiss, G., Weinberger, K. Q., Bindel, D., and Wilson, A. G. GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration. In *Advances in Neural Information Processing Systems*, pp. 7576–7586, 2018.

Gardner, J. R., Kusner, M. J., Xu, Z. E., Weinberger, K. Q., and Cunningham, J. P. Bayesian optimization with inequality constraints. In *ICML*, pp. 937–945, 2014.

Gelbart, M. A., Snoek, J., and Adams, R. P. Bayesian optimization with unknown constraints. In *30th Conference on Uncertainty in Artificial Intelligence, UAI 2014*, pp. 250–259. AUAI Press, 2014.

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

Hansen, N. The CMA evolution strategy: A comparing review. In *Towards a New Evolutionary Computation*, pp. 75–102. Springer, 2006. Code available at: https://github.com/CMA-ES/pycma. Last accessed on 02/03/2019.

Hedar, A.-R. and Fukushima, M. Derivative-free filter simulated annealing method for constrained continuous global optimization. *Journal of Global optimization*, 35(4):521–549, 2006.

Herbol, H. C., Hu, W., Frazier, P., Clancy, P., and Poloczek, M. Efficient search of compositional space for hybrid organic–inorganic perovskites via Bayesian optimization. *npj Computational Materials*, 4(1):1–7, 2018.

Hernández-Lobato, J. M., Hoffman, M. W., and Ghahramani, Z. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in Neural Information Processing Systems*, pp. 918–926, 2014.

Hernández-Lobato, J. M., Gelbart, M. A., Adams, R. P., Hoffman, M. W., and Ghahramani, Z. A general framework for constrained Bayesian optimization using information-based search. *The Journal of Machine Learning Research*, 17(1):5549–5601, 2016. Code available at: https://github.com/HIPS/Spearmint/tree/PESC. Last accessed on 02/03/2019.

Hernández-Lobato, J. M., Requeima, J., Pyzer-Knapp, E. O., and Aspuru-Guzik, A. Parallel and distributed Thompson sampling for large-scale accelerated exploration of chemical space. In *Proceedings of the International Conference on Machine Learning*, pp. 1470–1479, 2017.

Jameson, A. Automatic design of transonic airfoils to reduce the shock induced pressure drag. In *Proceedings of the 31st Israel annual conference on aviation and aeronautics, Tel Aviv*, pp. 5–17, 1990.

Jones, E., Oliphant, T., and Peterson, P. SciPy: Open source scientific tools for Python. 2014.

Kandasamy, K., Krishnamurthy, A., Schneider, J., and Póczos, B. Parallelised Bayesian optimisation via Thompson sampling. In *International Conference on Artificial Intelligence and Statistics*, pp. 133–142, 2018.

Keane, A. Experiences with optimizers in structural design. In *Proceedings of the conference on adaptive computing in engineering design and control*, volume 94, pp. 14–27, 1994.

Kramer, O. A review of constraint-handling techniques for evolution strategies. *Applied Computational Intelligence and Soft Computing*, 2010, 2010.

Lemonge, A., Barbosa, H., Borges, C., and Silva, F. Constrained optimization problems in mechanical engineering design using a real-coded steady-state genetic algorithm. *Mecánica Computacional*, 29:9287–9303, 2010.

Letham, B., Karrer, B., Ottoni, G., Bakshy, E., et al. Constrained Bayesian optimization with noisy experiments. *Bayesian Analysis*, 14(2):495–519, 2019.

Liem, R. P., Kenway, G. K., and Martins, J. R. Multimission aircraft fuel-burn minimization via multipoint aerostructural optimization. *AIAA Journal*, 53(1):104–122, 2014.

Liem, R. P., Martins, J. R. R. A., and Kenway, G. K. Expected drag minimization for aerodynamic design optimization based on aircraft operational data. *Aerospace Science and Technology*, 63:344–362, 2017.

Martins, J. R. R. A. Personal communication, 2018.

Mutny, M. and Krause, A. Efficient high dimensional Bayesian optimization with additivity and quadrature Fourier features. In *Advances in Neural Information Processing Systems*, pp. 9005–9016, 2018.

Nayebi, A., Munteanu, A., and Poloczek, M. A framework for Bayesian optimization in embedded subspaces. In *International Conference on Machine Learning*, pp. 4752–4761, 2019. The code is available at `https://github.com/aminnayebi/HesBO` and in `https://github.com/wujian16/Cornell-MOE`. Last accessed on 02/03/2019.

Oh, C., Gavves, E., and Welling, M. BOCK : Bayesian optimization with cylindrical kernels. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 3868–3877, 2018.

Oh, C., Tomczak, J., Gavves, E., and Welling, M. Combinatorial Bayesian optimization using the graph Cartesian product. In *Advances in Neural Information Processing Systems 32*, pp. 2910–2920. 2019.

Ortoll-Bloch, A. G., Herbol, H. C., Sorenson, B. A., Poloczek, M., Estroff, L. A., and Clancy, P. Bypassing solid-state intermediates by solvent engineering the crystallization pathway in hybrid organic–inorganic perovskites. *Crystal Growth & Design*, 2019.

Picheny, V. A stepwise uncertainty reduction approach to constrained global optimization. In *Artificial Intelligence and Statistics*, pp. 787–795, 2014.

Picheny, V., Gramacy, R. B., Wild, S., and Le Digabel, S. Bayesian optimization under mixed constraints with a slack-variable augmented Lagrangian. In *Advances in neural information processing systems (NIPS)*, pp. 1435–1443, 2016. Code available at: `https://cran.r-project.org/web/packages/laGP`. Last accessed on 02/03/2019.

Powell, M. J. A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in Optimization and Numerical Analysis*, pp. 51–67. Springer, 1994.

Powell, M. J. A view of algorithms for optimization without derivatives. *Mathematics Today-Bulletin of the Institute of Mathematics and its Applications*, 43(5):170–174, 2007.

Powell, W. B. A unified framework for stochastic optimization. *European Journal of Operational Research*, 275(3): 795–821, 2019.

Regis, R. G. and Shoemaker, C. A. A stochastic radial basis function method for the global optimization of expensive functions. *INFORMS Journal on Computing*, 19(4):497–509, 2007.

Rolland, P., Scarlett, J., Bogunovic, I., and Cevher, V. High-dimensional Bayesian optimization via additive models with overlapping groups. In *International Conference on Artificial Intelligence and Statistics*, pp. 298–307, 2018.

Schonlau, M., Welch, W. J., and Jones, D. R. Global versus local search in constrained optimization of computer models. *Lecture Notes-Monograph Series*, pp. 11–25, 1998.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and De Freitas, N. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

Snoek, J., Larochelle, H., and Adams, R. P. Practical Bayesian optimization of machine learning algorithms.

In *Advances in Neural Information Processing Systems*, pp. 2951–2959, 2012.

Spall, J. C. *Introduction to stochastic search and optimization: estimation, simulation, and control*, volume 65. John Wiley & Sons, 2005.

Thompson, W. R. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

Ubaru, S., Chen, J., and Saad, Y. Fast estimation of $tr(f(A))$ via stochastic Lanczos quadrature. *SIAM Journal on Matrix Analysis and Applications*, 38(4):1075–1099, 2017.

Wang, Z., Hutter, F., Zoghi, M., Matheson, D., and de Freitas, N. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55:361–387, 2016.

Wang, Z., Gehring, C., Kohli, P., and Jegelka, S. Batched large-scale Bayesian optimization in high-dimensional spaces. In *International Conference on Artificial Intelligence and Statistics*, pp. 745–754, 2018.

Wilson, A. G. and Ghahramani, Z. Copula processes. In *Advances in Neural Information Processing Systems*, pp. 2460–2468, 2010.

## Supplementary material

In Sect. A we describe how we leverage scalable GP regression to run SCBO and cEI with large sampling budgets. The value and scalability of our implementation is demonstrated by an experiment in Sect. B. We summarize the hyperparameters of SCBO in Sect. C and give additional details on how we shrink and expand the trust region. We prove a consistency result for SCBO in Sect. D. In Sect. E we show results for all baselines on the physics test problems where the objective and constraints have been transformed in the same fashion as in SCBO. Finally, Sect. F provides details on all benchmarks.

## A. Gaussian process regression

As usual, the hyperparameters of the Gaussian process (GP) model are fitted by optimizing the log-marginal likelihood. The domain is rescaled to $[0, 1]^d$ and the function values are standardized to have mean zero and variance one before fitting the GP. We use a Matérn-5/2 kernel with ARD and a linear mean function that we optimize using L-BFGS-B. The use of the linear mean function is important for the high-dimensional problems as it helps making progress early on. Following Snoek et al. (2012), a horseshoe prior is placed on the noise variance. We also learn a signal variance of the kernel.

Scaling BO to large number of evaluations is challenging due to the computational costs of inference. To compute the posterior distribution for $n$ observations, we need to solve linear systems with an $n \times n$ kernel matrix. This is commonly done via a Cholesky decomposition which has a computational complexity of $\Theta(n^3)$ flops. When there are $m$ constraints, the cost increases to $\Theta(mn^3)$ flops and thus may not scale to the large sampling budgets that we consider in this work. Thus, we leverage the parallelism of modern GPUs that allows to 'batch' several GP models which is provided in the GPyTorch package (Gardner et al., 2018). Relying only on fast matrix vector multiplication (MVM), we can solve linear systems with the kernel matrix using the conjugate gradient (CG) method and approximate the log-determinant via the Lanczos process (Dong et al., 2017; Ubaru et al., 2017). GPyTorch extends this idea to a batch of GPs by computing fast MVMs with a 3D tensor representing a kernel matrix of size $(m + 1) \times n \times n$, where $m+1$ is the number of batched GPs. The MVMs are further sped up by a compiled CUDA kernel constructed via KeOPS (Charlier et al., 2018). To the best of our knowledge, SCBO is the first Bayesian optimization algorithm to leverage batch GPs and KeOPS. Note that we also applied these ideas to scale cEI of Schonlau et al. (1998) to a large numbers of samples. Both implementations will be shared upon publication.

## B. Achieving Efficiency via Batch GPs

Next we describe an experiment that demonstrates that the Cholesky decomposition, which is commonly used in GP regression, does not scale to the large sampling budgets that are required for the demanding benchmarks that we study in this work. We consider training GPs with a different number of training points. The true function is standardized and we assume observations are subject to normally distributed noise with mean zero and variance $0.01$. We compare the computational cost of the Cholesky decomposition to the efficient batch GP implementation of GPyTorch, both in single precision. Table 2 provides run times for training, predictions, and sampling. We use $50$ gradient steps for training. All computations were performed on an NVIDIA RTX 2080 TI. The two rightmost columns also show the error for the mean and variance predictions. We see that batch GPs achieve a large speed-up while preserving a high accuracy. For example, with the batch GP implementation fitting one GP with 1000 points takes 2.33 seconds, while fitting 100 GPs in a batch only takes 11.15 seconds. Moreover, the Cholesky approach becomes impractical in the large-data regime: training 100 GPs with 8000 points takes 37 minutes, compared to just about 2 minutes for the batch GP implementation!

| Data size | | Cholesky decomposition | | | Scalable batch GPs | | | Prediction error | |
|---|---|---|---|---|---|---|---|---|---|
| #GPs | Training points | Training | Prediction | Sampling | Training | Prediction | Sampling | Mean MAE | Variance MAE |
| 1 | 1000 | 0.60 s | 0.06 s | 0.07 s | 2.09 s | 0.04 s | 0.39 s | 1.41e-03 | 4.44e-03 |
| 1 | 2000 | 1.03 s | 0.08 s | 0.09 s | 2.17 s | 0.05 s | 0.39 s | 2.12e-03 | 5.23e-04 |
| 1 | 4000 | 3.67 s | 0.16 s | 0.14 s | 2.26 s | 0.06 s | 0.41 s | 2.22e-04 | 4.19e-06 |
| 1 | 8000 | 22.87 s | 0.49 s | 0.32 s | 4.73 s | 0.13 s | 0.45 s | 3.94e-04 | 6.75e-05 |
| 10 | 1000 | 5.98 s | 0.47 s | 0.58 s | 2.49 s | 0.22 s | 1.16 s | 4.74e-02 | 1.00e-01 |
| 10 | 2000 | 10.13 s | 0.76 s | 0.77 s | 3.61 s | 0.31 s | 1.30 s | 5.00e-02 | 6.14e-02 |
| 10 | 4000 | 35.96 s | 1.53 s | 1.26 s | 7.81 s | 0.35 s | 0.95 s | 3.42e-04 | 9.89e-05 |
| 10 | 8000 | 227.30 s | 4.72 s | 2.88 s | 17.73 s | 0.70 s | 1.08 s | 5.24e-05 | 7.07e-06 |
| 50 | 1000 | 24.48 s | 2.40 s | 2.96 s | 7.37 s | 0.50 s | 2.77 s | 5.03e-03 | 3.38e-02 |
| 50 | 2000 | 49.02 s | 3.72 s | 3.88 s | 9.65 s | 0.90 s | 3.12 s | 4.06e-03 | 1.64e-03 |
| 50 | 4000 | 184.61 s | 7.90 s | 6.40 s | 21.15 s | 1.75 s | 3.57 s | 1.49e-04 | 2.10e-05 |
| 50 | 8000 | 1134.58 s | 25.37 s | 14.36 s | 66.50 s | 3.72 s | 4.40 s | 5.74e-04 | 1.15e-04 |
| 100 | 1000 | 55.94 s | 5.09 s | 6.11 s | 10.41 s | 1.03 s | 6.85 s | 1.85e-03 | 9.22e-04 |
| 100 | 2000 | 108.00 s | 8.42 s | 8.38 s | 18.59 s | 2.27 s | 8.61 s | 2.12e-02 | 2.40e-02 |
| 100 | 4000 | 365.88 s | 16.64 s | 12.54 s | 44.08 s | 3.91 s | 8.42 s | 4.56e-04 | 1.25e-04 |
| 100 | 8000 | 2303.86 s | 89.19 s | 28.48 s | 144.12 s | 13.14 s | 10.50 s | 1.24e-04 | 2.31e-05 |

*Table 2.* Computational cost for GP training, prediction, and sampling. The standard approach using the Cholesky decomposition in single precision is compared to a fast implementation using batch GPs. We take 50 gradient steps for training and predict/sample on 5000 test points. The mean and variance MAE between the two approaches is shown in the two rightmost columns.

## C. Details on SCBO

In all experiments we use the following hyperparameters for SCBO that were adopted from TURBO (Eriksson et al., 2019): $\tau_s = 3$, $\tau_f = \lceil d/q \rceil$, $L_{\min} = 2^{-7}$, $L_{\max} = 1.6$, $L_{\text{init}} = 0.8$, and perturbation probability $p_{\text{perturb}} = \min\{1, 20/d\}$, where $d$ is the number of dimensions and $q$ is the batch size. Recall that we assume the domain has been scaled to the unit hypercube $[0, 1]^d$. A *success* occurs if at least one evaluation in the batch improves on the incumbent. In this case, we increment the success counter and reset the failure counter to zero. If no point in the batch improves the current best solution, we set the success counter to zero and increment the failure counter. The discretized candidate set is also generated following the approach of Eriksson et al. (2019). In particular, we first create a scrambled Sobol sequence within the intersection of the TR and the domain $[0, 1]^d$. We use the value in the Sobol sequence with probability $p_{\text{perturb}}$ for a given candidate and dimension, and the value of the center otherwise.

## D. Global Consistency of SCBO

We prove that SCBO converges to a global optimum as the number of samples tends to infinity.

**Theorem 1.** *Suppose that* SCBO *with default parameters is used in a multi-start framework under the following conditions:*

1. *The initial points $\{y_i\}$ for* SCBO *are chosen such that for any $\delta > 0$ and $x \in [0, 1]^d$ there exists $\nu(x, \delta) > 0$ such that the probability that at least one point in $\{y_i\}$ ends up in a ball centered at $x$ with radius $\delta$ is at least $\nu(x, \delta)$.*

2. *The objective and constraints are bounded.*

3. *There is a unique global minimizer $x^*$.*

4. SCBO *considers any sampled point an improvement only if it improves the current best solution by at least some constant $\gamma > 0$.*

*Then,* SCBO *with noise-free observations converges to the global minimizer $x^*$.*

Note that condition (1) is met if the initial set is chosen uniformly at random. Conditions (2) and (3) hold almost surely under the prior for our domain and are common assumptions in global optimization (e.g., see Regis & Shoemaker, 2007; Spall, 2005). Condition (4) is a straightforward design decision of the algorithm.

*Proof.* First observe that SCBO will take only a finite number of samples for any trust region due to conditions (2) and (4). Thus, SCBO will restart infinitely often with a fresh trust region and hence there is an infinite subsequence $\{x_k(i)\}$ of initial points. This subsequence satisfies condition (1) by design. Thus, global convergence follows from the proof of global convergence for random search under condition (3) (e.g., see Spall, 2005). □

Note that this result also implies consistency for the TURBO algorithm of Eriksson et al. (2019).

# E. Results on the Transformed Physics Problems

Recall that SCBO applies the copula transformation to the objective function and the bilog transformation to each constraint. In this section we study on the four physics benchmarks how the performance of all baselines is affected by these transformations. Fig. 6 summarizes the performances. We note that the transformations do not lead to noticeable changes in performance for the baselines, except for PESC that benefits on the 3D tension-compression string problem and on the 4D welded beam design. The performance was comparable with and without the transformations for the other test problems.

# F. Details on the Benchmarks

In this section we provide additional information for the test problems. We refer the reader to the original papers for more details.

## F.1. 3D Tension-Compression String

The tension-compression string problem was described by Hedar & Fukushima (2006). The goal is to minimize the weight subject to constraints on minimum deflection, shear stress, surge frequency, limits on outside diameter, and on design variables (Coello & Montes, 2002). The first constraint is very sensitive to changes in the input parameters and cannot be modeled accurately by a global GP model.

## F.2. 4D Pressure Vessel Design

This problem was studied by Coello & Montes (2002) and has four constraints. The original problem does not specify bound constraints, so we use $0 \le x_1, x_2 \le 10$, $10 \le x_3 \le 50$, and $150 \le x_4 \le 200$. This domain contains the best solution found by Coello & Montes (2002). The goal is to minimize the total cost of designing the vessel, which includes the cost of the material, forming, and welding. The variables describe the thickness of the shell, thickness of the head, inner radius, and length of the cylindrical section of the vessel. The thickness of the shell and thickness of the head have to be multiples of 0.0625 and are rounded to the closest such value before evaluating the objective and constraints.

## F.3. 4D Welded Beam Design

This problem was considered by Hedar & Fukushima (2006) and has 5 constraints. The objective is to minimize the cost subject to constraints on shear stress, bending stress in the beam, buckling load on the bar, end deflection of the beam, and three additional side constraints.

## F.4. 7D Speed Reducer

The 7D speed reducer model has 11 black-box constraints and was described by Lemonge et al. (2010). The objective is to minimize the weight of a speed reducer. The design variables are the face width, the module of teeth, the number of teeth on pinion, the length of shaft one between the bearings, the length of shaft two between the bearings, the diameter of shaft one, and the diameter of shaft two.

## F.5. 10D Ackley

In this problem we consider the popular 10-dimensional Ackley function

$$f(x) = -20 \exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) -$$
$$\exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\right) + 20 + \exp(1)$$

in the domain $[-5, 10]^{10}$ subject to the constraints $c_1(x) = \sum_{i=1}^{10} x_i \le 0$ and $c_2(x) = \|x\|_2 - 5 \le 0$. This function is multi-modal and hard to optimize. Additionally, the size of the feasible region is small making this problem even more challenging. The optimal value is zero and is attained at the origin.

## F.6. 30D Keane Bump

For the Keane bump benchmark (Keane, 1994), the goal is to minimize

$$f(x) = -\left|\frac{\sum_{i=1}^{d}\cos^4(x_i) - 2\prod_{i=1}^{d}\cos^2(x_i)}{\sqrt{\sum_{i=1}^{d} i x_i^2}}\right|$$

subject to $c_1(x) = 0.75 - \prod_{i=1}^{d} x_i \le 0$ and $c_2(x) = \sum_{i=1}^{d} x_i - 7.5d \le 0$ over the domain $[0, 10]^d$. We consider the case $d = 30$ in our experiment. The Keane benchmark is notoriously famous for being challenging for Bayesian optimization as it is hard to model with a global GP model.

## F.7. 12D Robust Multi-point Optimization

The goal is to learn a robust controller for the lunar lander in the OpenAI gym[2]. The state space for the lunar lander is the position, angle, time derivatives, and whether or not either leg is in contact with the ground. For each frame there are four possible actions: firing a booster engine left, right, up, or doing nothing. The original objective was to maximize the average final reward over $m$ randomly generated terrains, initial positions, and velocities. We extend this
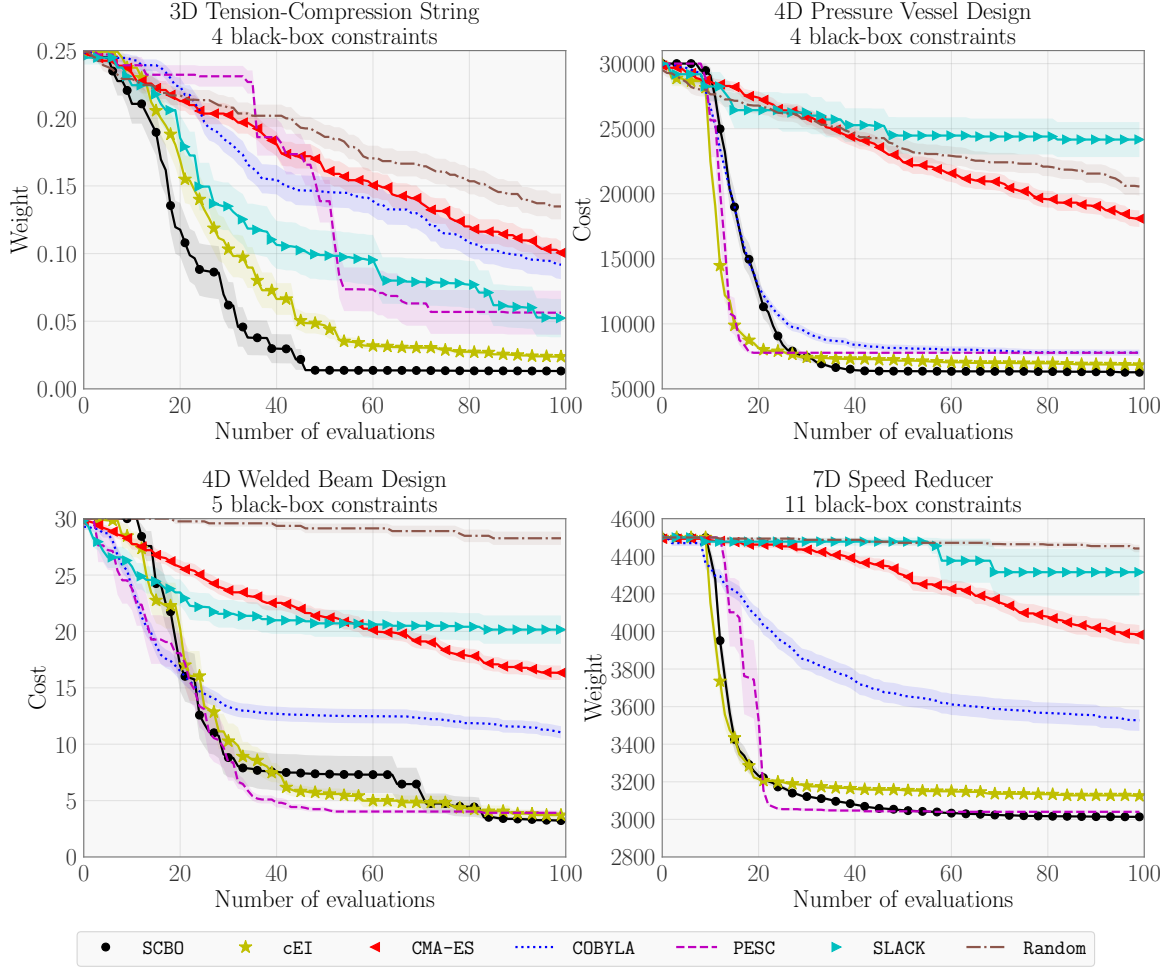
---

[2]https://gym.openai.com/envs/LunarLander-v2

Figure 6. **(Upper left)** `SCBO` and `cEI` outperform the other methods on the Tension-compression string problem. **(Upper right)** `SCBO` finds the best solutions on the pressure vehicle design problem, followed by `cEI`, `PESC`, and `COBYLA`. **(Lower left)** `PESC`, and `cEI` are eventually outperformed by `SCBO` on the welded beam design problem. **(Lower right)** `SCBO` and `PESC` perform the best on the speed reducer problem.

formulation to be more robust by adding $m$ constraints that no individual reward is below 200, which asserts that the lunar lander successfully lands in every scenario. Moreover, we fix the $m$ terrains throughout the optimization process therefore making the function evaluations deterministic (that is, without noise).

### F.8. 60D Rover Trajectory Planning

This problem was considered by Wang et al. (2018). The goal is to optimize the trajectory of a rover, where the trajectory is determined by fitting a B-spline to 30 design points in a 2D plane. The reward function is $f(\vec{x}) = c(\vec{x}) + 5$, where $c(\vec{x})$ penalizes any collision with an object along the trajectory by $-20$. We add the constraint that the trajectory has to start and end at the pre-specified start and end locations. Additionally, we add 15 additional obstacles that

are impassable and introduce constraints $c_i(x)$ for each $i$-th obstacle $o_i$ based on the final trajectory $\gamma(x)$ as follows:

$$c_i(x) = \begin{cases} -d(o_i, \gamma(x)) & \text{if } \gamma(x) \cap o_i = \emptyset, \\ \max_{\alpha \in \gamma(x) \cap o_i} \min_{\beta \in \partial o_i} d(\alpha, \beta) & \text{otherwise,} \end{cases}$$

where $\partial o_i$ is the boundary of $o_i$. That is, trajectories that do not collide with the object will be feasible under this constraint with a constraint value equal to the minimum distance between the trajectory and the object. Trajectories that collide with the object will be given a larger constraint value if they intersect close to the center of the object.

### F.9. 124D Vehicle Design with 68 Constraints (MOPTA08)

MOPTA08 is a large-scale multi-disciplinary optimization problem from the vehicle industry (Anjos, 2008). There are 124 variables that describe gages, materials, and shapes as well as 68 performance constraints. Note that a problem with this many input dimensions and black-box constraints is out of reach for existing methods.

### F.10. 2D Toy problem

This problem was proposed by Hernández-Lobato et al. (2016). The goal is to minimize the function $f(x) = x_1 + x_2$ subject to $c_1(x) = 1.5 - x_1 - 2x_2 - 0.5\sin(2\pi(x_1^2 - 2x_2)) \leq 0$ and $c_2(x) = x_1^2 + x_2^2 - 1.5 \leq 0$. The objective and constraints are all smooth low-dimensional functions. The domain is the unit square $[0, 1]^2$.

### F.11. 5D Rosenbrock function

The goal is to minimize the Rosenbrock function $f(x) = \sum_{i=1}^{4} \left[ 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$ subject to two constraints involving the Dixon-Price[3] (DP) function $c_1(x) = f_{\text{DP}}(x) - 10 \leq 0$ and the Levy[4] function $c_2(x) = f_{\text{Levy}}(x) - 10 \leq 0$. We created this problem to illustrate a setting where the objective and constraints are poorly scaled. SCBO excels on this problem as the use of the trust region and robust transformations makes it possible to quickly make progress. The domain is $[-3, 5]^5$.

---

[3]`https://www.sfu.ca/~ssurjano/dixonpr.html`
[4]`https://www.sfu.ca/~ssurjano/levy.html`