Report on "Scalable Constrained Bayesian Optimization"

Son Luu

March 1, 2024

1 Summary

1.1 Review

The paper [EP21] proposed a constrained Bayesian optimization (CBO) algorithm, called SCBO, for black-box functions with black-box constraints. The algorithm is scalable for high-dimensional functions and able to support large batches for constrained problems even with asynchronous observations. To do this, the authors extended Thompson sampling to constrained optimization, transformed the objective and constrain functions to emphasize desired features, and used the trust region approach to keep samples local. The proposed method is shown in multiple experiment to match or outperform previous state-of-the-art methods on both high and low dimensions. In addition, two new high-dimensional constrained test problems of independent interest were introduced.

Many early works on CBO focus on the constrained expected improvement (cEI) criterion [SWJ98], obtained by multiplying expected improvement and probability of feasibility, in various settings [Sób+14; Par+12]. SCBO, on the other hand, uses Thompson sampling (TS) [Tho33] to large batch sizes, developed by [Her+17], to be scalable whereas methods based on predictive entropy search like PESC [Her+16] and Lagrangian relaxation like SLACK [Pic+16] do not generally scale well to higher dimensions due to their computational costs. [Wan+16; Eri+18; Rol+18] also used BO for high-dimensional problems with small sampling budgets but not in constrained setting. Most notably, the paper extended the trust region (TR) approach of TURBO [Eri+19], where samples are only collected from a hypercube around the best observation, to CBO context. However, the update criteria for the TR size seems too arbitrary and is not motivated by any prior works on TR based methods (see [Yua15] for a review). Another strength of [EP21] is the use of transformations such as Gaussian copula [WG10] and bilog to emphasize properties desired in the objective and constrains. Unfortunately, these transformations only benefited their proposed method. Other works such as [SGR03] and [Sno+14] also applied data-dependent transformations to black-box functions.

- *Originality:* The task of CBO is not a particularly new subject and there have been a few papers on the matter. Some aspects of the paper are new but most techniques utilized has been developed in previous works. The TR and TS aspect of SCBO have been well studied and is essentially an extension of TURBO to CBO setting. Other techniques such as the sample selection criterion and the transformations used on the objective and constraints are also well studied. However, the amalgamation of those techniques, in my opinion, is what gives SCBO an edge in performance compared to its predecessors. In addition, the paper provided a proof for the consistency of its proposed method that also extends to TURBO. Related works cited by the paper come from various approaches, which goes to show the authors are very familiar with the literature.
- *Quality:* The paper is technically sound and all of its claims are supported by experimental results. The experiments seem to cover many situations high dimensions (30D Keane bump function, 60D trajectory planning problem and 124D vehicle design problem) and low dimensions (3D tension-compression string problem, 4D pressure vessel design, 7D speed reducer problem, 10D Ackley problem and 12D robust multi-point optimization problem). The methods used are overall appropriate but there is room for improvement. The TR schedule could have been motivated by past works in more traditional optimization. There are many other criteria for sample selection that can overcome the weaknesses of the total constraint violation criterion stated in the paper. For these reasons, I would say the paper is still a work in progress. The authors did mention some weaknesses in the proposed method (total constraint violation susceptibility to scaling) but only to validate their subsequent decision to apply transformations to the target functions. Furthermore, the authors did not point out the reasoning behind the results of the ablation studies (section 4.7) for different acquisition functions. Specifically, the authors mentioned that TS outperforms EI for the 5D Rosenbrock function. Base on this, I think it is unfair that among the methods compared to SCBO, there does not seem to be any that uses TS.
- *Clarity:* The paper is well written and easy to follow. However, I think the technical details on the

TR and TS being placed before the algorithm summary is a little distracting and the order should be reversed. In the experiment section, it was not immediately clear which problems the authors introduced and what makes them interesting. Therefore, my advice is to briefly mention this information before going into the details of each problem. The experiments themselves are clearly stated and therefore, can be reproduced. However, the source code for the proposed method and experiments was not provided.

- *Significance:* The proposed method is significant in that it highlights the benefit of TR and function transformation. Although the transformation in SCBO has been shown to not have much effect on other methods, I believe future methods should adopt the TR approach when dealing high dimensional problems. In addition, SCBO has been shown to outperform many previous state-of-the-art works. Furthermore, it also introduced two new high-dimensional constrained test problems (12D robust multi-point optimization problem and a constraint-based extension of the 60D trajectory planning problem) that can be used to test future algorithms. The multi-point optimization problem, in particular, is a great example of naturally increasing the number of constraints to a problem while keeping its dimension fixed.

Questions:

- Is there any theoretical motivation to the choice of hyperparameters such as the thresholds for shrinking or expanding of TRs?
- How would SCBO perform compared to other algorithms when its competitors also make use of the trust region approach?
- How would SCBO perform for noisy functions?
- TURBO uses several TRs simultaneously but SCBO only uses one. Is there a reason for this change?

1.2 Methodological summary

1.2.1 Set up

The task is to find

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

where $f : \Omega \to \mathbb{R}$ and $c_l : \Omega \to \mathbb{R}, l = 1, \ldots, m$ are black-box functions, i.e. functions we can only query values from, defined over a compact set $\Omega \subset \mathbb{R}^d$. For CBO, the practice is to model the objective f and constrains c_l using a multivariate Gaussian process (GP) surrogate. At each round, a batch (or a single) points are chosen, using a policy based on the current dataset (sampled points and observed values), to be observed and added to the dataset. After a stopping criterion has been satisfied, the procedure returns the point with the lowest objective value that satisfies the constrains.

1.2.2 Methodology

The steps SCBO algorithm follows are

1. Evaluate an initial set of points and initialize the trust region, a hypercube with side length $L = L_{init}$, centered at a point of maximum utility.
2. Until the budget for samples is exhausted:
   - Fit GP models to the transformed observations.
   - Generate r candidate points $x_1, \ldots, x_r \in \Omega$ in the trust region.
   - Choose a batch of q points from these candidates to add to the dataset using TS for CBO.

- Evaluate the objective and constraints at the q new points.
- Adapt the trust region by moving the center or initialize a new one.

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

The specifics of these steps are as follow:

- **Thompson sampling in CBO:** SCBO uses TS as the main strategy for sample acquisition. To obtain a sample, TS starts by sampling a realization $(\hat{f}(x_i), \hat{c}_1(x_i), \ldots, \hat{c}_m(x_i))^T$ for each of the candidate point $x_i, i = 1, \ldots, r$ from the posterior distribution of the objective and constraints given the current dataset. These realizations are then compared and the feasible (satisfying all constraints) point with the lowest objective value is the chosen sample. When there are no feasible points, the one with the lowest total constraint violation (TCV)

$$\sum_{l=1}^{m} \max\{c_l(x), 0\} \tag{2}$$

  is chosen with objective value being the tiebreaker.
  While TCV is a natural selection criterion, it struggles when functions have significantly varying magnitudes. This is because the sum structure of TCV is sensitive to scaling.
  TS is the core of SCBO due to its simplistic formulation and ability to naturally balance between exploitation (choosing candidates with better objectives) and exploration (choosing uncertain candidates). Furthermore, TS can handle batches of GP models since each of its step can be parallelized by running them on multiple GPUs, making computation more scalable. However, since the method is based on sampling, there is a certain variation in performance depending on the amount of uncertainty in the posterior, making TS unreliable in some cases compared to more deterministic policies such as EI.

- **Maintaining the trust region:** When a new batch of samples is added, the center of a TR is shifted to the point of maximum utility (best objective or TCV). If at least one sample from the batch improves on the incumbent then the batch is considered a success, the success counter is increased by 1 and the failure counter resets to 0, otherwise it is considered a failure, the failure counter is increased by 1 and the success counter resets to 0. When a certain number of successes (failures) have been counted, the length L of the TR increases (decreases) by a factor of 2 and both counters reset. And when L falls below a certain threshold, the current TR is terminated and a new one is initialized.
  The main purpose of TR is to sample locally and, as a result, reduce the explorative behaviour of the policy. This addresses the phenomenon where samples of popular policies become too spread out and fail to zoom in on promising solutions when using BO in high dimensional settings. The local search makes it harder to find the global optimum within budget but the trade off is being able to find a good local optimum faster. The main challenge is to find a suitable update schedule for the TRs as well as the upper and lower limits to TR sizes. A large TR covers more grounds but makes samples more spread out while a small TR zooms in on good solutions better but cannot see as much of the search space.

- **Transformations of objective and constraints:** the paper applied a Gaussian copula, a function that operates on the data quantiles, to the objective and a bilog transformation to the constrains where

$$\text{bilog}(y) = \text{sgn}(y) \log(1 + |y|). \tag{3}$$

  The former magnifies differences between values at the end of the observed range for the objective while the latter emphasizes the change of sign as well as dampens large values for the constrains. This step is crucial to SCBO since it cover the scaling weakness that TCV has. However, these transformations were shown to not as noticeable an impact on other methods.

1.2.3 Consistency

Given 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 δ 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 $\gamma > 0$.

The paper also showed that SCBO is consistent in noise-less setting. They argued that conditions (2) and (4) make SCBO take a finite number of samples for any TR. Therefore, the algorithm will restart its TR infinitely often, creating an infinite subsequence $\{x_k(i)\}$ of initial points satisfying condition (1). Then global convergence follows from the proof of global convergence for random search under condition (3) [Spa05]. This proof also applies to TURBO since the finite sample argument works for both algorithms.

1.2.4 Crucial aspects and potential bottlenecks

The paper has shown that the trust region is crucial to the performance of SCBO. However, its update schedule is entirely based on empirical heuristics. Developing a more theoretical based schedule would improve the algorithm performance. Another weakness mentioned by the paper is the inability to handle highly varying functions of TCV. Because of this, the transformations used in SCBO are required and not optional like the other competitors. My suggestion is to use a variation of TCV that is more robust to scaling. Finally, the cost of learning the GP surrogate models tends to be the deciding factor to the method scalability. It is thanks to fast matrix vector multiplication implemented in the *GPyTorch* package [Gar+18] and the CUDA kernel constructed via *KeOPS* [CFG], that BO inference for thousands of samples can be done in minutes.

1.2.5 Alternative methods

Other methods that were used in CBO include:

1. **cEI** [Gar+14] multiplies the expected improvement with the probability of feasibility. This one of the most well-known and natural methods for CBO. However, in the case of no available feasible observations, cEI becomes ill-defined and have to rely on the probability of feasibility to find feasible points first.
2. **PESC** [Her+16] extends predictive entropy search to CBO and detailing how an approximation method that makes computation tractable. It is one of the first CBO methods that allows for decoupled evaluations of the objective and constraints. Its main drawback is the computation time since the acquisition function is approximated via Monte Carlo sampling.
3. **SLACK** [Pic+16] lifted the constraints into the objective via the Lagrangian relaxation. It introduces the Lagrange multipliers and slack variables as hyperparameters that accompany the constraints along with the objective to turn the original problem into one without constraints. This modified problem is then solved with the hyperparameters sequentially changed. This essentially turn a constrained optimization problem into several unconstrained ones, which can be costly in high dimension.
4. **CMA-ES** [Kra10] uses a covariance adaptation strategy to learn a second order model of the objective function. Furthermore, CMA-ES penalizes violations setting the fitness values of infeasible solutions to zero.

5. **COBYLA** [Pow94] also uses trust regions but model the objective and constrains using linear approximation.

2 Mini-project

2.1 Proposal

In constrained Bayesian optimization (CBO), the goal is to maximize an expensive objective function subjected to a number of constrains using Bayesian optimization. To accomplish this goal, one needs to construct an optimization policy that takes into account the constraints and favors the input region satisfying them, i.e. the feasible region. An interesting topic is how to adjust the level of favoritism for feasible points as infeasible points can still be useful by providing information about the behaviours of the feasible region. In this project, these adjustments are encoded into the policy by relaxing or tightening the original constraints.

To include the feasibility of constrains in their policies, [Gar+14; GSA14] used the expected improvement acquisition function weighted by the probability of feasibility while [Ber+11] used a function called integrated expected conditional improvement that computes expected reduction in improvement when a new point is evaluated. [GSA14] also considered the probability of feasibility as the acquisition function but this puts too much preference on finding the feasible region. [EP21] used the total constraint violation to choose the next observations. This, however, is sensitive to scaling and requires a transformation to stabilize the objective and constraints. [Kra10] used a fitness value to choose the next point and set the fitness of infeasible points to zero. It is important to note that most methods for CBO encode a fixed notion of preference for feasible points. This eliminates the need for extra hyperparameter tuning but the methods may behave suboptimally compared to ones with varied preference.

In this report, the original CBO problem is augmented by auxiliary parameters that control the looseness of the constraints. With this, I develop a new class of acquisition functions by weighting the notion of utility for these augmented problems. Finally, I propose an adaptive weighting scheme that changes the preference for feasible points based on observed dataset.

The main challenge for this scheme is, of course, how to set the weighting scheme. To do this, one needs to know to what extend changing the scheme would affect the level of favoritism for feasible points and the overall performance of the CBO policy. Too much emphasis on feasibility can lead to the objective not being optimized as quickly while to little emphasis may not yield any feasible points. Another concern is the computational complexities that come with the proposed method. There are elements that vary depending on the auxiliary variable, complicating any integration in the acquisition function. This means the acquisition function may not have a simple closed formula and instead needs to be approximated via sampling.

This project provides an example of how more information about the constraints can be incorporated into the optimization policy by relaxing and tightening the original CBO problem. I believe this approach is an intuitive way of viewing the trade-off between optimizing the objective and satisfying the constraints. In addition, I think the adaptation of this trade-off is a very useful practice in order to improve methods for CBO, similar to how the adaptation of the exploitation-exploration trade-off (e.g. trust region approach [Eri+19; EP21]) can benefit optimization strategies.

2.2 Weighted Constrained Expected Improvement (WCEI)

2.2.1 Set up

Consider the optimization problem

$$\text{Find } x^* = \underset{x\in\Omega}{\operatorname{argmax}} f(x) \text{ s.t. } \mathbf{c}(x) \leq \mathbf{0} \tag{4}$$

where $f : \Omega \to \mathbb{R},\ c = (c_1, \dots, c_m),\ c_l : \Omega \to \mathbb{R}, l = 1, \dots, m$ are black-box functions, $\Omega \subset \mathbb{R}^d$ and

$$\{\mathbf{c}(x) \leq \mathbf{0}\} := \{x :\ c_l(x) \leq 0,\ l = 1, \dots, m\}. \tag{5}$$

The strategy is the same as the one in the set up of the summary section: the objective f and constraints c_l are modelled by a multivariate Gaussian process (GP) surrogate, and at each iteration a policy is used to determine the next observation to add to the dataset until a stop condition is met. For this project, observations of f and $\mathbf{c}$ are exact and the policy is to choose an observation maximizing a so-called acquisition function.

2.2.2 Acquisition function

Consider the family of augmented CBO problems

$$\text{Find } x^* = \underset{x\in\Omega}{\text{argmax}} f(x) \text{ s.t. } \mathbf{c}(x) \leq \mathbf{a} \tag{6}$$

where $\mathbf{a} = (a_1, \ldots, a_m) \in \mathbb{R}^m$ and

$$\{\mathbf{c}(x) \leq \mathbf{a}\} := \{x : \ c_l(x) \leq a_l, \ l = 1, \ldots, m\}. \tag{7}$$

When $a_l > 0$, the corresponding constraint is relaxed and when $a_l < 0$, the constraint is tightened. The *Weighted Constrained Expected Improvement* (WCEI) algorithm I propose uses the following notion of utility

$$u(\mathcal{D}) = \int \left(\max_{\mathcal{D}\cap\{\mathbf{c}(x)\leq\mathbf{a}\}} f(x) \right) p(\mathbf{a}) d\mathbf{a} \tag{8}$$

where $\mathcal{D}$ is the current dataset and p is the distribution of the auxiliary variable $\mathbf{a}$, weighing the utility contribution of each problem in the augmented family. Note that this utility function reduces to that of constrained expected improvement (cEI) [Gar+14] when p is the point mass at $\mathbf{0}$. We can now write the improvement in utility when a new point is added to the dataset

$$I(x) := u(\mathcal{D}') - u(\mathcal{D}) = \int \max\{f(x) - f^*_{\mathbf{a}}, 0\} \mathbf{1}\{\mathbf{c}(x) \leq \mathbf{a}\} p(\mathbf{a}) d\mathbf{a} \tag{9}$$

where $\mathcal{D}'$ is the dataset when the observation at x is added, $f^*_{\mathbf{a}} := \max_{\mathcal{D}\cap\{\mathbf{c}(x)\leq\mathbf{a}\}} f(x)$ and $\mathbf{1}(A)$ is the indicator function of the set A. When $\mathcal{D} \cap \{\mathbf{c}(x) \leq \mathbf{a}\} = \emptyset$, we set

$$\max\{f(x) - f^*_{\mathbf{a}}, 0\} = 1 \tag{10}$$

This is similar to how cEI default to the probability of feasibility when there are no feasible observations. The acquisition function is the expectation of this improvement over the posterior distribution of f and $\mathbf{c}$

$$\alpha(x|\mathcal{D}) := E_{f,\mathbf{c}|x}(I(x)). \tag{11}$$

2.2.3 Computation of $\alpha(x|\mathcal{D})$

Going back to the formula in (9), note that the improvement function can be rewritten as

$$I(x) = \sum_i \max\{f(x) - f^{(i)}, 0\} \int_{A_i} \mathbf{1}\{\mathbf{c}(x) \leq \mathbf{a}\} p(\mathbf{a}) d\mathbf{a} \tag{12}$$

where $f^{(i)}$ denotes the i^{th} order statistic of the objective values in the dataset and

$$A_i := \{\mathbf{a} : \ f^*_{\mathbf{a}} = f^{(i)}\} \ \forall \ i > 0, \tag{13}$$

$$A_0 := \{\mathbf{a} : \ \mathcal{D} \cap \{\mathbf{c}(x) \leq \mathbf{a}\} = \emptyset\}, \tag{14}$$

$$\max\{f(x) - f^{(i)}, 0\} := 1. \tag{15}$$

Plugging (12) into the acquisition function formula (11) gives us

$$\alpha(x|\mathcal{D}) = \sum_i E_{f|x}(\max\{f(x) - f^{(i)}, 0\}) \int_{A_i} E_{\mathbf{c}|x}(\mathbf{1}\{\mathbf{c}(x) \leq \mathbf{a}\}) p(\mathbf{a}) d\mathbf{a} \tag{16}$$

$$= \sum_{i>0} EI(x; f^{(i)}) \int_{A_i} P(\mathbf{c}(x) \leq \mathbf{a}) p(\mathbf{a}) d\mathbf{a} + \int_{A_0} P(\mathbf{c}(x) \leq \mathbf{a}) p(\mathbf{a}) d\mathbf{a} \tag{17}$$

where

$$EI(x;y) = \Sigma(x)(Z\Phi(Z) + \phi(Z)) \tag{18}$$

with Φ, ϕ being the cdf and pdf of the standard normal distribution, $\mu(x)$ the posterior mean at x, $\Sigma(x)$ the posterior standard deviation at x and

$$Z = \frac{\mu(x) - y}{\Sigma(x)}. \tag{19}$$

Now we need to find the sets A_i and compute the integrals in (17).

- **Finding** A_i: Assuming that the dataset $\mathcal{D}$ has N observations, we can find A_i recursively starting with A_N. Since $f^{(N)} = \max_{\mathcal{D}} f(x)$, we have

$$f^*_{\mathbf{a}} = f^{(N)} \ \forall \ \mathbf{a} \geq \mathbf{c}^{(N)} \tag{20}$$

  where $\mathbf{c}^{(N)} = (c_1^{(N)}, \ldots, c_m^{(N)})$ is the constraint observation corresponding to $f^{(N)}$. Therefore

$$A_N = (\mathbf{c}^{(N)}, \infty) := (c_1^{(N)}, \infty) \times \ldots \times (c_m^{(N)}, \infty). \tag{21}$$

  Next, we have for all $i \in \{1, \ldots, N-1\}$

$$f^*_{\mathbf{a}} \geq f^{(i)} \ \forall \ \mathbf{a} \geq \mathbf{c}^{(i)} \tag{22}$$

  Therefore, we have the following recursive relationship between the A_i's

$$A_i = (\mathbf{c}^{(i)}, \infty) \backslash \left(\underset{j \geq i}{\cup} A_j \right) \quad \forall \ i \in \{1, \ldots, N-1\}. \tag{23}$$

  Finally, we have

$$A_0 = \mathbb{R}^m \backslash \left(\underset{j \geq 0}{\cup} A_j \right) \tag{24}$$

  In (12), note that we can also divide the integrals based on the observations of $\mathbf{c}$. The integral regions obtained will be hyperretangles

- **Computing the integrals in** (17): Due to the shapes the A_i's may take, the integrals in (17) need to be computed via Monte Carlo approximation. To do this, we first sample $\mathbf{a} \sim p$, sort the samples into the corresponding A_i and use these to approximate the individual integrals. The specific steps of this process are summarized in Algorithm 1.

2.2.4 Adapting WCEI

Going back to (17), we see that there are two types of terms: the ones in the sum where the integrals are multiplied by an EI term and the one where the integral stands alone. In the first type, the more we increase $\mathbf{a}$ the closer $P(\mathbf{c}(x) \leq \mathbf{a})$ is to one, making the term behave similarly to the unconstrained EI. Conversely, the more we decrease $\mathbf{a}$ the larger the set A_0 is, making the acquisition function behave similarly to the probability of feasibility. In other words, the more we relax the constraints the more the acquisition function favors maximizing the objective, and the more we tighten the constraints the more the acquisition function wants to obtain feasible points. To control the degree of looseness of the augmented problems, I proposed the adaptive policy in Algorithm 2. The idea is to tighten the problem when there are fewer feasible observations and relax the problem otherwise, similar to how an adaptive random walk Metropolis Hastings algorithm [RR09] increase and decrease its step size. This is achieved by rescaling the positive and negative weights separately.

Algorithm 1: Computation of the integrals in (17).

Input : surrogate model of f and $\mathbf{c}$, weight distribution p, number of samples r, dataset $\mathcal{D}$.
Output: approximations of $\int_{A_i} P(\mathbf{c}(x) \leq \mathbf{a})p(\mathbf{a})d\mathbf{a}, i = 0, \ldots, N$.
Sample $\mathbf{a}_1, \ldots, \mathbf{a}_r \sim p$.
for $i = N, \ldots, 0$ **do**
  Find $\hat{A}_i = \{\mathbf{a}_j : \mathbf{a}_j \in A_i\}$ via (21),(23) and (24).
  Compute

$$PF_i = \frac{1}{r} \sum_{j:\mathbf{a}_j \in \hat{A}_i} P(\mathbf{c}(x) \leq \mathbf{a}_j). \tag{25}$$

end
return $PF_0, \ldots, PF_N$.

Algorithm 2: Adaptive WCEI.

Input : surrogate model of f and $\mathbf{c}$, number of $\mathbf{a}$ samples r, dataset $\mathcal{D}$, number of iterations T.
Output: $x^* = \underset{x \in \mathcal{D}}{\text{argmax}} f(x)$ s.t. $\mathbf{c}(x) \leq \mathbf{0}$.
Sample $\mathbf{a}_1, \ldots, \mathbf{a}_r \sim N(\mathbf{0}, I_m)$.
for $i = 1, \ldots, T$ **do**
  Choose $x = \text{argmax}\alpha(x|\mathcal{D})$.
  Observe $y = f(x)$.
  Update $\mathcal{D} = \mathcal{D} \cup \{x, y\}$.
  if *feasible proportion* < 0.5 **then**
    Set $\mathbf{a}_i = 2^{1/d}\mathbf{a}_i \; \forall \; \mathbf{a}_i \leq \mathbf{0}$.
    Set $\mathbf{a}_i = 2^{-1/d}\mathbf{a}_i \; \forall \; \mathbf{a}_i > \mathbf{0}$.
  else
    Set $\mathbf{a}_i = 2^{-1/d}\mathbf{a}_i \; \forall \; \mathbf{a}_i \leq \mathbf{0}$.
    Set $\mathbf{a}_i = 2^{1/d}\mathbf{a}_i \; \forall \; \mathbf{a}_i > \mathbf{0}$.
  end
end
return $\mathcal{D}$.

2.3 Ablation study

Consider the 6 toy CBO problems in Figure 1. These problems represent various situations for a pair of objective and constraint. I investigate how tightening and relaxing the constraint affect the behaviour of WCEI compared to cEI, specifically the unconstrained reward, constrained reward and the proportion of feasible observation. For WCEI, I look at two weighting schemes where $\mathbf{a} \sim |N(0, 1)|$ and $\mathbf{a} \sim -|N(0, 1)|$ to represent a tightening only scheme and a relaxing only scheme, respectively. Figure 2 shows the mean curves over 20 runs of the 3 quantities being investigated for each of the problems. As expected, tighten only WCEI consistently obtained more feasible observations than the other two while relax only WCEI got better unconstrained reward. In the second and third problem, where constrain is beneficial to optimizing the objective or is easy to satisfy, all methods performed equally well with relax only WCEI being slightly worse than the other two. In the last three problems, where neither the objective, constraint or both are adversarial to optimization, we see that tighten only WCEI tends to get the best constrained reward in the early iterations and the effect seems to be more apparent when both the objective and constraint are adversarial. However, cEI tends to beat the other two in terms of constrained reward in the later iterations due to the inherent balance built into its design.

2.4 Simulation study

In this section, the adaptive policy in Algorithm 2 is compared to cEI for the Hartmann6 test function

$$f(x) = -\sum_{i=1}^{4} \alpha_i \exp\left(-\sum_{j=1}^{6} A_{ij}(x_j - P_{ij})^2\right) \tag{26}$$

where $x \in [0,1]^6$ and α_i, A_{ij}, P_{ij} are known constants. The problem is subjected to the constraint

$$c(x) = \|x\|_1 - 1.5. \tag{27}$$

Each method is run for 30 iterations and repeated 20 times. Figure 3 shows median curves of the runs along with their interquartile range (IQR) for each method. Overall, the adaptive scheme did slightly better than cEI but it is not clear as their IQRs have a lot of overlap.

2.5 Discussion

In this mini-project, I have devised and implemented a new acquisition function for CBO that augments the looseness of the original constrained optimization problem. This class of acquisition functions allows us to construct an adaptive policy that can adjust the level of favoritism for the feasible region. The adaptive policy devised here is relatively primitive and more understanding of WCEI is required to build a more effective policy. Note that this approach of controlling the constraint looseness can also be applied to other methods as well.

The idea of changing the looseness of a CBO problem also have some practical applications. Constraint tightening is beneficial when the constraints must be satisfied to observe the objective such as drug development satisfying safety constraints. On the other hand, in problems such as portfolio building, the budget constraints can be relaxed if it means the expected yield could be significantly improved.

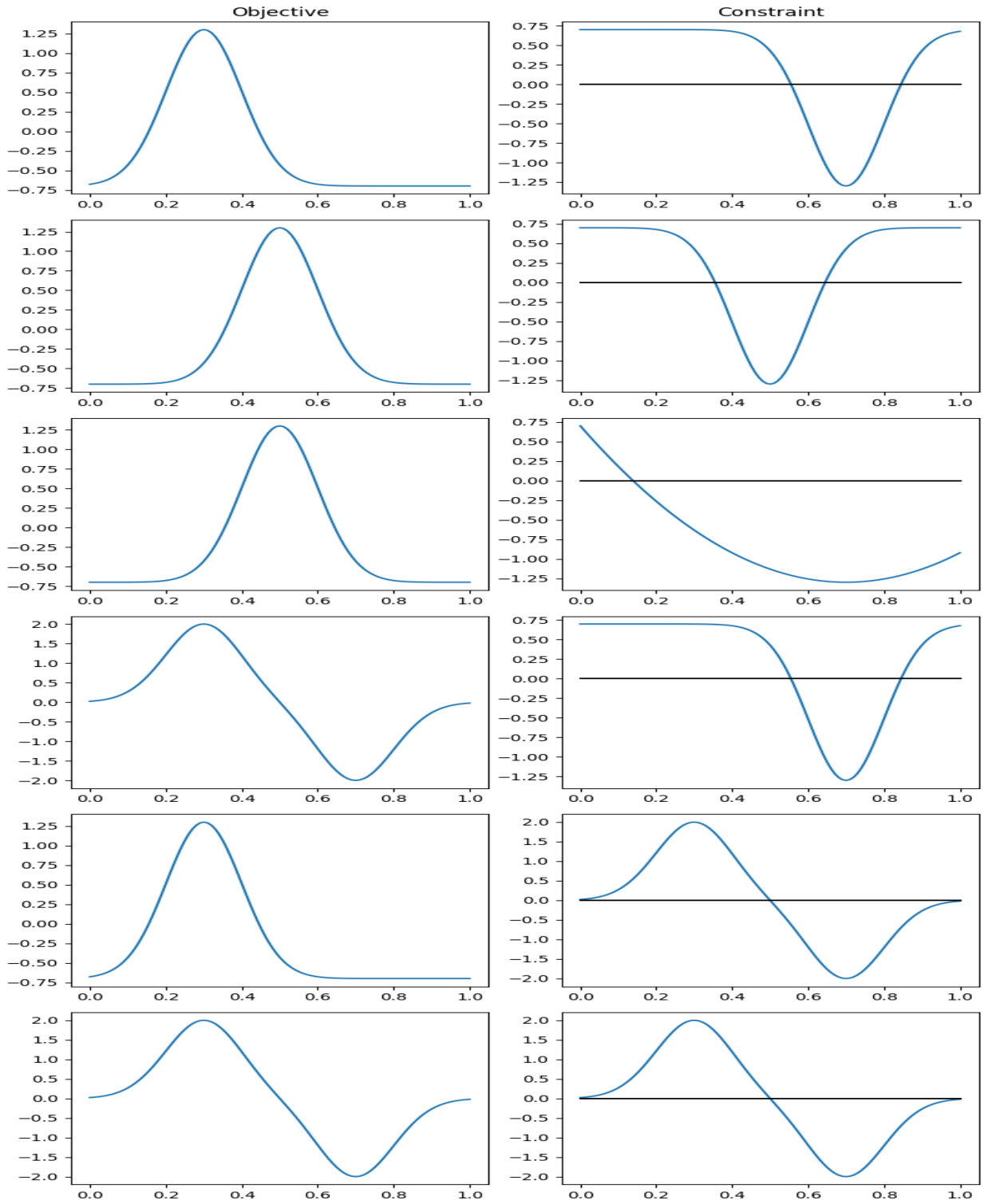

Figure 1: Toy problems for ablation studies. The region below the black lines are the feasible regions.

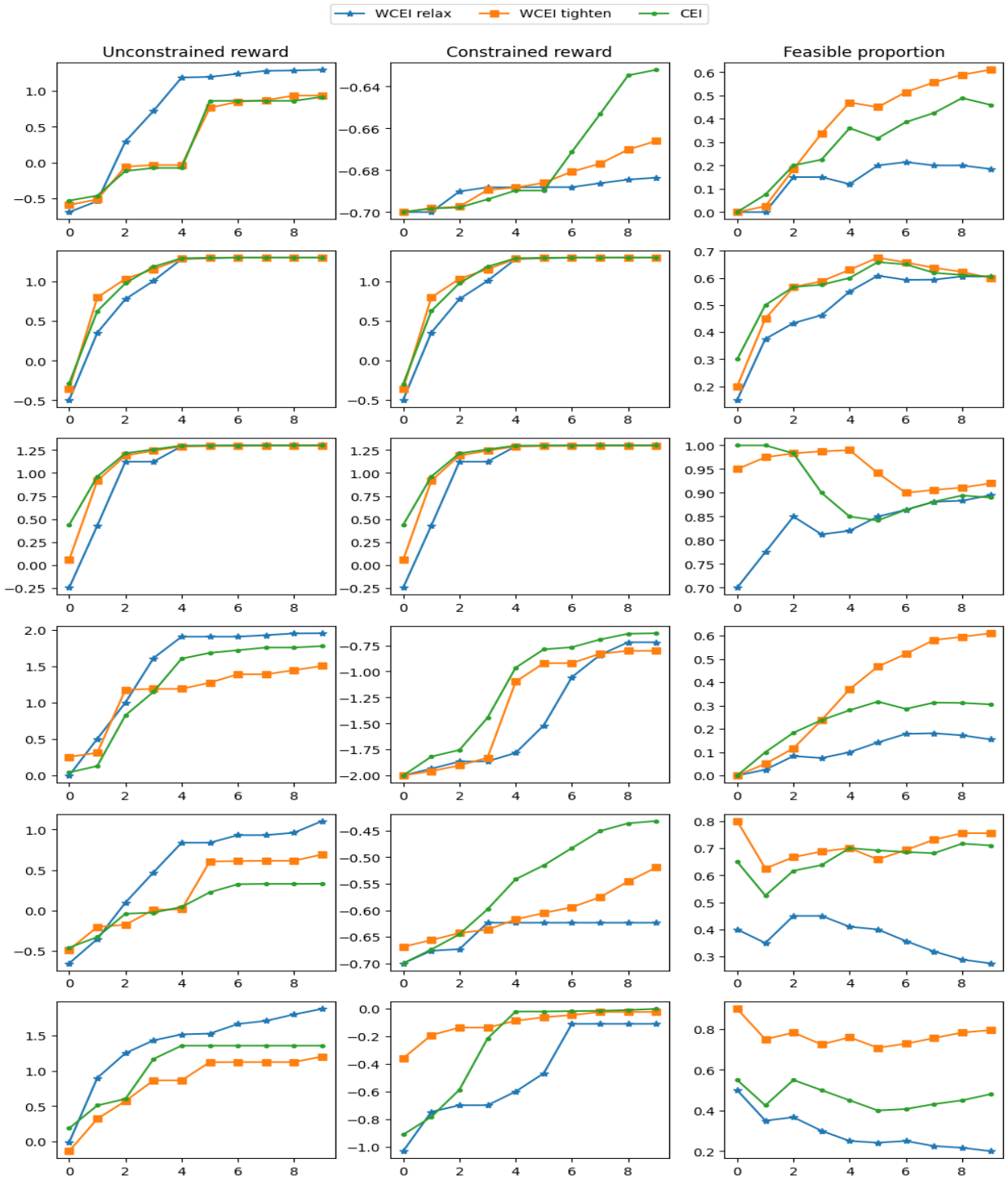

Figure 2: Optimization results for the toy problems. The x axis represent the number of iterations.

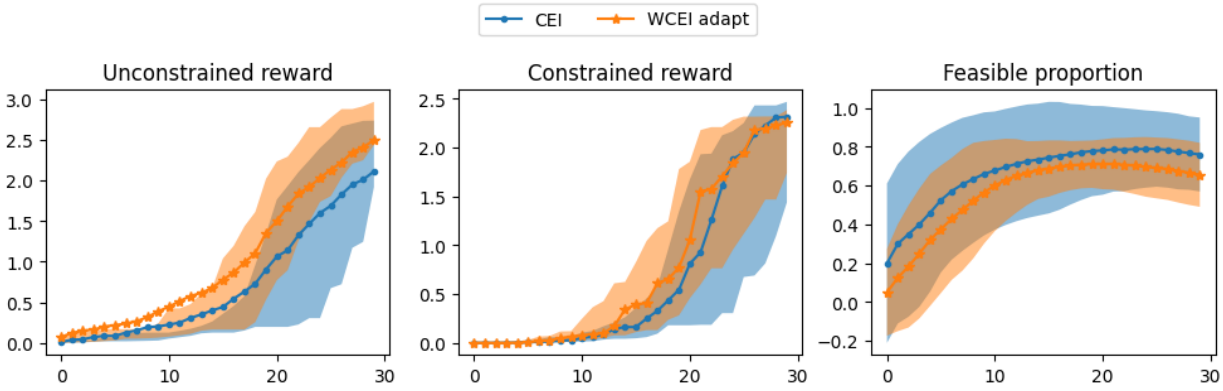

Figure 3: Optimization results for constrained Hartmann6 problem. The x axis represent the number of iterations.

References

[Ber+11] J. Bernardo et al. "Optimization under unknown constraints". In: *Bayesian Statistics* 9.9 (2011), p. 229.

[CFG] B. Charlier, J. Feydy, and J. Glaunés. "KeOps, 2018". In: *URL https://github. com/getkeops/keops* ().

[EP21] D. Eriksson and M. Poloczek. "Scalable constrained Bayesian optimization". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 730–738.

[Eri+18] D. Eriksson et al. "Scaling Gaussian process regression with derivatives". In: *Advances in neural information processing systems* 31 (2018).

[Eri+19] D. Eriksson et al. "Scalable global optimization via local Bayesian optimization". In: *Advances in neural information processing systems* 32 (2019).

[Gar+18] J. Gardner et al. "Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration". In: *Advances in neural information processing systems* 31 (2018).

[Gar+14] J. R. Gardner et al. "Bayesian optimization with inequality constraints." In: *ICML*. Vol. 2014. 2014, pp. 937–945.

[GSA14] M. A. Gelbart, J. Snoek, and R. P. Adams. "Bayesian optimization with unknown constraints". In: *30th Conference on Uncertainty in Artificial Intelligence, UAI 2014*. AUAI Press. 2014, pp. 250–259.

[Her+16] J. M. Hern et al. "A general framework for constrained Bayesian optimization using information-based search". In: *Journal of Machine Learning Research* 17.160 (2016), pp. 1–53.

[Her+17] J. M. Hernández-Lobato et al. "Parallel and distributed Thompson sampling for large-scale accelerated exploration of chemical space". In: *International conference on machine learning*. PMLR. 2017, pp. 1470–1479.

[Kra10] O. Kramer. "A review of constraint-handling techniques for evolution strategies". In: *Applied Computational Intelligence and Soft Computing* 2010 (2010), pp. 1–19.

[Par+12] J. M. Parr et al. "Infill sampling criteria for surrogate-based optimization with constraint handling". In: *Engineering Optimization* 44.10 (2012), pp. 1147–1166.

[Pic+16] V. Picheny et al. "Bayesian optimization under mixed constraints with a slack-variable augmented Lagrangian". In: *Advances in neural information processing systems* 29 (2016).

[Pow94] M. J. Powell. *A direct search optimization method that models the objective and constraint functions by linear interpolation*. Springer, 1994.

[RR09] G. O. Roberts and J. S. Rosenthal. "Examples of adaptive MCMC". In: *Journal of computational and graphical statistics* 18.2 (2009), pp. 349–367.

[Rol+18] P. Rolland et al. "High-dimensional Bayesian optimization via additive models with overlapping groups". In: *International conference on artificial intelligence and statistics*. PMLR. 2018, pp. 298–307.

[SWJ98] M. Schonlau, W. J. Welch, and D. R. Jones. "Global versus local search in constrained optimization of computer models". In: *Lecture notes-monograph series* (1998), pp. 11–25.

[SGR03] E. Snelson, Z. Ghahramani, and C. Rasmussen. "Warped gaussian processes". In: *Advances in neural information processing systems* 16 (2003).

[Sno+14] J. Snoek et al. "Input warping for Bayesian optimization of non-stationary functions". In: *International Conference on Machine Learning*. PMLR. 2014, pp. 1674–1682.

[Sób+14] A. Sóbester et al. "Engineering design applications of surrogate-assisted optimization techniques". In: *Optimization and Engineering* 15 (2014), pp. 243–265.

[Spa05] J. C. Spall. *Introduction to stochastic search and optimization: estimation, simulation, and control*. John Wiley & Sons, 2005.

[Tho33] W. R. Thompson. "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples". In: *Biometrika* 25.3-4 (1933), pp. 285–294.

[Wan+16] Z. Wang et al. "Bayesian optimization in a billion dimensions via random embeddings". In: *Journal of Artificial Intelligence Research* 55 (2016), pp. 361–387.

[WG10] A. G. Wilson and Z. Ghahramani. "Copula processes". In: *Advances in Neural Information Processing Systems* 23 (2010).

[Yua15] Y.-x. Yuan. "Recent advances in trust region algorithms". In: *Mathematical Programming* 151 (2015), pp. 249–281.