TREGO: a Trust-Region Framework for Efficient Global Optimization

Youssef Diouane, Victor Picheny, Rodolphe Le Riche, Alexandre Scotto Di Perrotolo

TREGO: a Trust-Region Framework for Efficient Global Optimization

Y. Diouane* V. Picheny † R. Le Riche ‡ A. Scotto Di Perrotolo§

February 3, 2021

Abstract

Efficient Global Optimization (EGO) is the canonical form of Bayesian optimization that has been successfully applied to solve global optimization of expensive-to-evaluate black-box problems. However, EGO struggles to scale with dimension, and offers limited theoretical guarantees. In this work, we propose and analyze a trust-region-like EGO method (TREGO). TREGO alternates between regular EGO steps and local steps within a trust region. By following a classical scheme for the trust region (based on a sufficient decrease condition), we demonstrate that our algorithm enjoys strong global convergence properties, while departing from EGO only for a subset of optimization steps. Using extensive numerical experiments based on the well-known COCO benchmark, we first analyze the sensitivity of TREGO to its own parameters, then show that the resulting algorithm is consistently outperforming EGO and getting competitive with other state-of-the-art global optimization methods. The method is available both in the `R` package `DiceOptim` [1] and python library `trieste` [2].

Keywords: nonlinear optimization; Gaussian processes; Bayesian optimization; trust-region.

1 Introduction

In the past 20 years, Bayesian optimization (BO) has encountered great successes and a growing popularity for solving global optimization problems with expensive-to-evaluate black-box functions. Examples range from aircraft design [1] to automatic machine learning [2] to crop selection [3]. In a nutshell, BO leverages non-parametric (Gaussian) processes (GPs) to provide flexible surrogate models of the objective. Sequential sampling decisions are based on the GPs, judiciously balancing exploration and exploitation in search for global optima (see [4, 5] for early works or [6] for a recent review).

*ISAE-SUPAERO, Université de Toulouse, France. E-mail: youssef.diouane@isae-supaero.fr.
†Secondmind, 72 Hills Road, Cambridge, CB2 1LA, UK. E-mail: victor@secondmind.ai
‡CNRS LIMOS, Mines St-Etienne and UCA, France. E-mail: leriche@emse.fr
§ISAE-SUPAERO, Université de Toulouse, France. E-mail: alexandre.scotto-di-perrotolo@isae-supaero.fr

[1] https://cran.r-project.org/package=DiceOptim

[2] https://secondmind-labs.github.io/trieste/

BO typically tackles problems of the form:

$$\min_{x \in \Omega} f(x), \tag{1}$$

where f is a pointwise observable objective function defined over a continuous set $\Omega \subset \mathbb{R}^n$, with n relatively small (say, 2 to 20). In this work, we assume that $f : \mathbb{R}^n \to \mathbb{R}$ is observable exactly (i.e. without noise), bounded from below in $\mathbb{R}^n$ and Lipschitz continuous near appropriate limit points.

Despite its popularity and successes, BO suffers from a couple of important drawbacks. First, it is very sensitive to the curse of dimensionality, as with growing dimension exploration tends to overcome exploitation and learning an accurate model throughout the search volume is typically not feasible within a limited number of function evaluations. Several recent works have tackled this problem, either making strong structural assumptions ([7, 8, 9]) or incentivizing sampling away from the boundaries ([10, 11]). Second, the theoretical properties for BO are rather limited, in particular in the noiseless context. For BO algorithms based on the expected improvement acquisition function, Vazquez and Bect [12] showed that the sequence of evaluation points is dense in the search domain providing some strong assumptions on the objective function. Bull [13] built upon this result to calculate an upper bound on the simple regret of EGO for GP models with a Matérn kernel. However, these bounds require the addition of a well-calibrated epsilon-greedy strategy to EGO and they are valid for a limited family of objective functions.

Over the past two decades, there has been a growing interest in deterministic Derivative-Free Optimization (DFO) (see for reviews [14, 15]. DFO methods either try to build local models of the objective function based on samples of the function values, e.g. trust-region methods, or directly exploit a sample set of function evaluations without building an explicit model, e.g. direct-search methods. Motivated by the large number of DFO applications, researchers and practitioners have made significant progress on the algorithmic and theoretical aspects (in particular, proofs of global convergence) of the DFO methods.

In this paper, we propose to equip a classical BO method with known techniques from deterministic DFO using a trust region scheme, and a sufficient decrease condition to accept new iterates and ensure convergence [16]). This is in line with recent propositions hybridizing BO and DFO [17, 18] that showed great promise empirically, but with limited theoretical guarantees. Our TREGO algorithm (Trust Region framework for Efficient Global Optimization) benefits from both worlds: we show that TREGO rigorously achieves global convergence under reasonable assumptions, while enjoying the flexible predictors and efficient exploration-exploitation trade-off provided by the GPs. Contrary to the aforementioned propositions, TREGO maintains a global search step, ensuring that the algorithm can escape local optima and maintain the asymptotic properties of BO [12, 13].

The remainder of this article is organized as follows. Section 2 presents the classical BO framework. Section 3 describes our hybrid algorithm, and Section 4 its convergence properties. In Section 5 we report numerical experiments, including an ablation study and a broad comparison with other algorithms using the COCO test bed [19]. Conclusions and perspectives are finally provided in Section 6.

2 The Efficient Global Optimization Framework

Efficient Global Optimization [5, EGO] is a class of BO methods relying on two key ingredients: (i) the construction of a GP surrogate model of the objective function and (ii) the use of an acquisition function. EGO proceeds along the following steps:

1. an initial set of evaluations (often referred to as Design of Experiment, *DoE*) of the objective function is obtained, typically using a space-filling design [20];
2. a GP surrogate model is trained on this data;
3. a fast-to-evaluate acquisition function, defined with the GP model, is maximized over Ω;
4. the objective function is evaluated at the acquisition maximizer;
5. this new observation is added to the training set and the model is re-trained;
6. steps 3 to 5 are repeated until convergence or budget exhaustion.

The surrogate model is built by putting a Gaussian process (GP) prior on the objectives:

$$Y(.) \sim \mathcal{GP}\left(m(.), c\left(.,.\right)\right). \tag{2}$$

where the mean m and covariance c have predetermined parametric forms. Conditioning on a set of observations $\mathcal{D}_t = \{X_t, Y_t\}$, where $X_t = \{x_1, \ldots, x_t\}$ and $Y_t = \{f(x_1), \ldots, f(x_t)\}$, we have

$$\begin{aligned} m_t(x) &:= \mathbb{E}\left[Y(x)|\mathcal{D}_t\right] = m(x) + \lambda(x)\left(f - m(x)\right), \\ c_t(x,x') &:= \operatorname{cov}\left[Y(x), Y(x')|\mathcal{D}_t\right] = c(x,x') - \lambda(x)c(x',x_t), \end{aligned}$$

where

- $\lambda(x) := c(x,x_t)^\top c(X_t,X_t)^{-1}$,
- $c(x,x_t) := (c(x,x_1),\ldots,c(x,x_t))^\top$ and
- $c(X_t,x_t) := (c(x_i,x_j))_{1\leq i,j\leq n}$.

Typically, m is taken as constant or a polynomial of small degree and c belongs to a family of covariance functions such as the Gaussian and Matérn kernels, based on hypotheses about the smoothness of y. Corresponding hyperparameters are often obtained as maximum likelihood estimates; see for example [21, 22] for the corresponding details.

Once the surrogate model is built, an acquisition function (ic) is used to determine which point is most likely to enrich efficiently the model regarding the search for a global minimizer of the objective function f. The expression of ic only depends on the probabilistic surrogate model and usually integrates a trade-off between exploitation (low $\mu_t(x)$) and exploration (high $c_t(x,x)$). In the noise-free setting, the canonical acquisition is Expected Improvement [5, EI], the expected positive difference between $y_{\min} = \min_{1\leq i\leq n}(y_i)$, the minimum of the values observed so far, and the new potential observation $Y_{t+1}(x)$:

$$\begin{aligned} \mathrm{EI}_t(x) &= \mathbb{E}\left(\max\left((0, y_{\min} - Y(x)\right)|\mathcal{D}_t\right) \\ &= (f_{\min} - m_t(x))\Phi\left(\frac{y_{\min} - m_t(x)}{\sqrt{c_t(x,x)}}\right) + \sqrt{c_t(x,x)}\phi\left(\frac{y_{\min} - m_t(x)}{\sqrt{c_t(x,x)}}\right), \end{aligned}$$

with ϕ and Φ denoting the probability and cumulative density functions, respectively, of the standard normal variable. Note that many alternative acquisition functions have been proposed over the past 20 years, see for example [23] for a recent review. We stress that while we focus here on EI for simplicity, our framework described later is not limited to EI and other acquisitions can be used instead (see Section 4 for suitable choices).

Given $\mathcal{D}_t$ the set of observations available at iteration k, the next optimization iterate x_{k+1} is given by

$$x_{k+1}^{\text{global}} = \underset{x \in \Omega}{\text{argmax}}\ \alpha(x; \mathcal{D}_t). \tag{3}$$

where α corresponds to the chosen acquisition function at iteration k (for EGO, $\alpha(x; \mathcal{D}_t) = \text{EI}_t(x)$).

For most existing implementations of EGO, the stopping criterion relies typically on a maximum number of function evaluations. In fact, unlike gradient-based methods where the gradient's norm can be used as a relevant stopping criterion which ensure a first-order stationarity, derivative-free optimization algorithms have to cope with a lack of general stopping criterion and the EGO algorithm makes no exception.

We note also that, in the framework considered here, the constraints are treated as explicit [**?**, i.e. not relying on estimates, as in]]schonlau1998global and non-relaxable (meaning that the objective function cannot be evaluated outside the feasible region [24]). Typically, we assume that Ω is defined as bound constraints.

3 A Trust-Region framework for EGO (TREGO)

In this section, we propose a modified version of EGO where we include a control parameter (which depends on the decrease of the true objective function) to ensure some form of global convergence without jeopardizing the performance of the algorithm.

3.1 The TREGO algorithm

Our methodology follows the lines of the search/poll direct-search methods [25, 14, 26, 27]. In the context of EGO, this results in a scheme alternating between *local* and *global* phases. The global phase corresponds to running one iteration of the classical EGO algorithm over the whole design space as in Eq. 3. This phase ensures an efficient global exploration and aims at identifying the neighborhood of a global minimizer. The local phase corresponds to running one iteration of EGO, but restricting the search to the vicinity of the current best point (Ω_k, detailed hereafter), so that

$$x_{k+1}^{\text{local}} = \underset{x \in \Omega_k}{\text{argmax}}\ \alpha(x; \mathcal{D}_t). \tag{4}$$

Associated with a proper management of Ω_k, this phase ensures that the algorithm converges to a stationary point. All the trial points, whether coming from the global or from the local phase, are included in the DoE to refine the surrogate model of the objective function f.

By default, only the global phase is used. The local one is activated when the global phase isn't successful, that is when it fails to sufficiently reduce the best objective function value. In addition, the local phase consists of a fixed number of steps (typically only one), after which the algorithm reverts to the global phase. Consequently, the original EGO algorithm is entirely maintained over a subset of steps.

The local phase management follows two widely used techniques in the field of nonlinear optimization with and without derivatives. First, we impose some form of *sufficient decrease condition* on the objective function values to declare an iteration successful. Second, we control the size of the steps taken at each iteration using a parameter σ_k that is updated depending on the sufficient decrease condition (increased if successful, decreased otherwise). Given a current best point x_k^*, at iteration k, its neighborhood is defined as

$$\Omega_k = \{x \in \Omega \mid d_{\min}\sigma_k \leq \|x - x_k^*\| \leq d_{\max}\sigma_k\}, \tag{5}$$

where $d_{\min} < d_{\max}$ are any two strictly positive real values. The inclusion in the algorithm of the bounds $d_{\min}$ and $d_{\max}$ on the definition of Ω_k is essential to our convergence analysis. In practice, the constant $d_{\min}$ can be chosen very small and the upper bound $d_{\max}$ can be set to a very large number.

At each iteration of the local phase, we impose the following sufficient decrease on the objective function:

$$f(x_{k+1}^{\text{local}}) \leq f(x_k^*) - \rho(\sigma_k), \tag{6}$$

where $\rho : \mathbb{R}^+ \to \mathbb{R}^+$ is a forcing function [16], i.e., a positive nondecreasing function such that $\rho(t)/t \to 0$ when $t \downarrow 0$ (for instance, $\rho(t) = t^2$). The step size parameter σ_k can be kept unchanged or possibly increased if the iteration is successful, which happens if the new iterate x_{k+1}^{local} found in either the global or the local phase decreases the objective function. The step size is reduced if the sufficient decrease condition (6) is not satisfied, i.e., $\sigma_{k+1} = \beta_k\sigma_k$ with $\beta_k \in [\beta_{\min}, \beta_{\max}]$, with $\beta_{\min}, \beta_{\max} \in (0, 1)$. A classical scheme is to use a fixed parameter $\beta \in (0, 1)$, and apply:

$$\begin{aligned} \sigma_{k+1} &= \frac{\sigma_{k+1}}{\beta} \quad \text{if the iteration is successful} \\ \sigma_{k+1} &= \sigma_{k+1}\beta \quad \text{otherwise.} \end{aligned} \tag{7}$$

Figure 1 is a schematic illustration of the algorithm. The pseudo-code of the full algorithm is given in Appendix A.

3.2 Extensions

We now present several possible extensions to TREGO. Some of these extensions are tested in the ablation study of Section 5.2.1.

Local / global ratio: in the previous section, a single local step is performed when the global step fails. The local/global ratio can easily be controlled by forcing several consecutive steps of either the global or the local phase. For example, a "`gl3-5`" (see algorithms names later) tuning would first perform three global steps regardless of their success. If the last step fails, it then performs five local steps. Such modification will not alter the structure of the algorithm. Moreover, since the convergence analysis relies on a subsequence of unsuccessful iterations, the validity of the convergence analysis (see Section 4) is not called into question. In fact, during the local phase, we keep using the same sufficient decrease condition to decide whether the current iteration is successful or not.

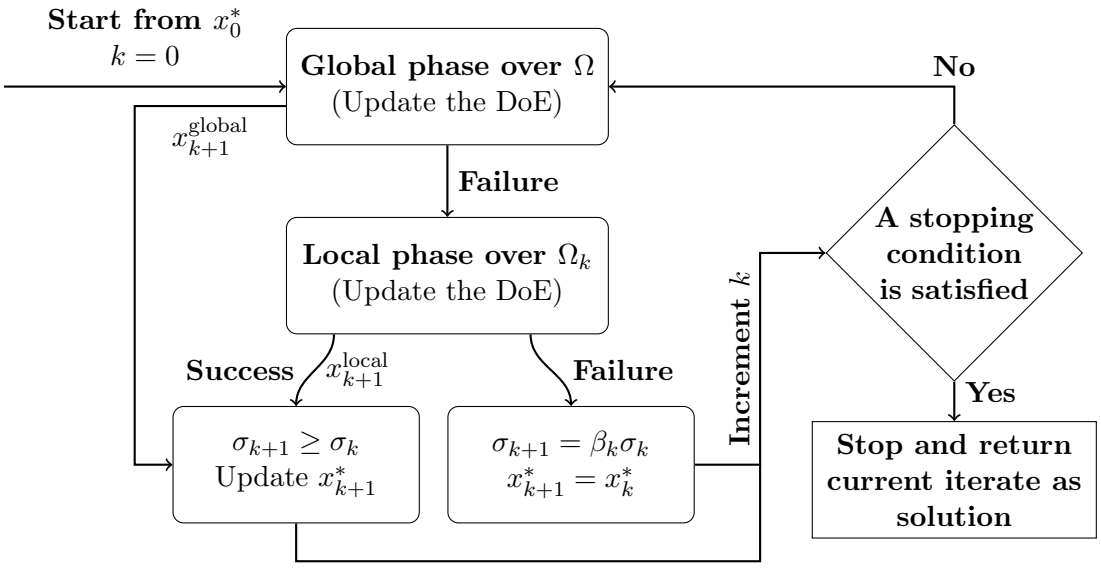

Figure 1: An overview of the TREGO framework (detailed in Algorithm 1).

Local acquisition function: our analysis (see Section 4) does not require using the same acquisition for the global and local steps. For example, as EI tends to become numerically unstable in the vicinity of a cluster of observations, it might be beneficial to use the GP mean or a lower confidence bound [28] as an acquisition function for the local step.

Local model: similarly, our approach does not require using a single model for the global and local steps. One could choose a local model that uses only the points inside the trust-region to allow a better fit locally, in particular for heterogeneously varying functions.

Non BO local step finally, our analysis holds when the algorithm employed for the local step is not Bayesian. For example, using BFGS would allow a more aggressive local search, which could prove beneficial [29]. In fact, as far as we impose the condition (6) to decide whether the current iteration is successful or not, the convergence theory of the next section applies.

3.3 Related work

TRIKE [17] (Trust-Region Implementation in Kriging-based optimization with Expected improvement) implements a trust-region-like approach where each iterate is obtained by maximizing the expected improvement acquisition function within some trust region. The two major differences with TREGO are: 1) the criterion used to monitor the step size evolution is based on the ratio between the expected improvement and the actual improvement, rather than sufficient decrease; 2) TRIKE does not have a global phase. In [17], TRIKE is associated with a restart strategy to ensure global search.

TURBO [18] (a TrUst-Region BO solver) carries out a collection of simultaneous BO runs using independent GP surrogate models, each within an different trust region. The trust-region

radius is updated with a failure/success mechanism based on the progress made on the objective function[3]. At each iteration, a global phase (managed by an implicit multi-armed bandit strategy) allocates samples between these local areas and thus decides which local optimizations to continue.

Both TRIKE and TURBO display very promising performances, in particular when solving high dimensional optimization problems. However, both rely on several heuristics that hinder theoretical guarantees. In contrast, the use of the search/poll direct-search algorithmic design [25, 14, 26, 27] allows TREGO to benefit from global convergence properties.

4 Convergence analysis of TREGO

Under appropriate assumptions, the global convergence of the proposed algorithm is now deduced. By global convergence, we mean the ability of a method to generate a sequence of points converging to a stationary point regardless of the starting DoE. A point is said to be stationary if it satisfies the first order necessary conditions, in the sense that the gradient is equal to zero if the objective function is differentiable or, in the non-smooth case, any directional derivative of the Clarke generalized derivatives [30] is non-negative.

The sketch of the convergence analysis is as follows. First, we prove that there exists a subsequence K of unsuccessful iterates driving the step size to zero (what is referred to as a refining subsequence in [31]) Because of the sufficient decrease in the objective function and the fact that the step size is significantly reduced (at least by $\beta_{\max}$) during unsuccessful iterations, one can guarantee that a subsequence of step sizes will converge to zero. Consequently, by assuming boundness of the sequence of iterates $\{x_k^*\}$, it is possible to assure the existence of a convergent refining subsequence. Our convergence analysis is concluded by showing that the limit point is a Clarke stationary point with respect to f.

Lemma 4.1 *Consider a sequence of iterations generated by Algorithm 1 without any stopping criterion. Let f be bounded below. Then* $\liminf_{k\to+\infty} \sigma_k = 0$.

Proof. Suppose that there exists a $\sigma > 0$ such that $\sigma_k > \sigma$ for all k.

If there is an infinite number of successful iterations, this leads to a contradiction to the fact that f is bounded below. In fact, since ρ is a non-decreasing positive function, one has $\rho(\sigma_k) \geq \rho(\sigma) > 0$. Hence, $f(x_{k+1}) \leq f(x_k^*) - \rho(\sigma)$ for all k, which obviously contradicts the boundedness below of f.

If no more successful iterations occur after a certain order, then this also leads to a contradiction as σ_k cannot stay larger than $\sigma > 0$. Thus, one must have a subsequence of iterations driving σ_k to zero. ■

From the fact that σ_k is only reduced in unsuccessful iterations by a factor not approaching zero, one can then conclude the following.

Lemma 4.2 *Consider a sequence of iterations generated by Algorithm 1 without any stopping criterion. Let f be bounded below.*

There exists a subsequence K of unsuccessful iterates for which $\lim_{k\in K} \sigma_k = 0$.

[3]Importantly, TURBO uses a simple decrease rule of the objective function, which is insufficient to ensure convergence to a stationary point with GP models.

If the sequence $\{x_k\}$ is bounded, then there exists an x^* and a subsequence K of unsuccessful iterates for which $\lim_{k\in K} \sigma_k = 0$ and $\lim_{k\in K} x_k^* = x^*$.

Proof. From Lemma 4.1, there must exist an infinite subsequence K of unsuccessful iterates for which σ_{k+1} goes to zero. In a such case we have $\sigma_k = (1/\beta_k)\sigma_{k+1}$, $\beta_k \in (\beta_{\min}, \beta_{\max})$, and $\beta_{\min} > 0$, and thus $\sigma_k \to 0$, for $k \in K$, too. The second part of the lemma is also proved by extracting a convergent subsequence of the subsequence K of the first part for which x_k converges to x^*. ■

The global convergence will be achieved by establishing that some type of directional derivatives are non-negative at limit points of refining subsequences along certain limit directions (known as refining directions). By refining subsequence [31], we mean a subsequence of unsuccessful iterates for which the step-size parameter converges to zero. When f is Lipschitz continuous near x^*, one can make use of the Clarke-Jahn generalized derivative along a direction d

$$f^\circ(x^*; d) \;=\; \limsup_{\substack{x \to x^*, x \in \Omega \\ t \downarrow 0, x + td \in \Omega}} \frac{f(x + td) - f(x)}{t}.$$

(Such a derivative is essentially the Clarke generalized directional derivative [30], adapted by Jahn [32] to the presence of constraints.) However, for the proper definition of $f^\circ(x^*; d)$, one needs to guarantee that $x + td \in \Omega$ for $x \in \Omega$ arbitrarily close to x^* which is assured if d is hypertangent to Ω at x^*. In the following definition we will use the notation $B(x; \Delta) = \{y \in \mathbb{R}^n : \|y - x\| \leq \Delta\}$.

Definition 4.1 *A vector $d \in \mathbb{R}^n$ is said to be a hypertangent vector to the set $\Omega \subseteq \mathbb{R}^n$ at the point x in Ω if there exists a scalar $\epsilon > 0$ such that*

$$y + tw \in \Omega, \quad \forall y \in \Omega \cap B(x; \epsilon), \quad w \in B(d; \epsilon), \quad \textit{and} \quad 0 < t < \epsilon.$$

The hypertangent cone to Ω at x, denoted by $T_\Omega^H(x)$, is then the set of all hypertangent vectors to Ω at x. Then, the Clarke tangent cone to Ω at x (denoted by $T_\Omega(x)$) can be defined as the closure of the hypertangent cone $T_\Omega^H(x)$ (when the former is nonempty, an assumption we need to make for global convergence anyway). The Clarke tangent cone generalizes the notion of tangent cone in Nonlinear Programming [33], and the original definition $d \in T_\Omega(x)$ is given below.

Definition 4.2 *A vector $d \in \mathbb{R}^n$ is said to be a Clarke tangent vector to the set $\Omega \subseteq \mathbb{R}^n$ at the point x in the closure of Ω if for every sequence $\{y_k\}$ of elements of Ω that converges to x and for every sequence of positive real numbers $\{t_k\}$ converging to zero, there exists a sequence of vectors $\{w_k\}$ converging to d such that $y_k + t_k w_k \in \Omega$.*

Given a direction v in the tangent cone, possibly not in the hypertangent one, one can consider the Clarke-Jahn generalized derivative to Ω at x^* [34] as the limit

$$f^\circ(x^*; v) \;=\; \lim_{d \in T_\Omega^H(x^*), d \to v} f^\circ(x^*; d).$$

A point $x^* \in \Omega$ is considered Clarke stationary if $f^\circ(x^*; d) \geq 0$, $\forall d \in T_\Omega(x^*)$. Moreover, when f is strictly differentiable at x^*, one has $f^\circ(x^*; d) = \nabla f(x^*)^\top d$. Hence in this case, if x^* is a Clark stationary point is being equivalent to $\nabla f(x^*)^\top d \geq 0$, $\forall d \in T_\Omega(x^*)$.

To state the global convergence result, it remains to define the notion of refining direction (see [34]), associated with a convergent refining subsequence K, as a limit point of $\{d_k/\|d_k\|\}$ for all $k \in K$ sufficiently large such that $x_k^* + \sigma_k d_k \in \Omega$ where one has $d_k = (x_{k+1}^{\rm local} - x_k^*)\sigma_k^{-1}$.

The following theorem is in the vein of those first established in [34] for simple decrease and Lipschitz continuous functions (and later generalized in [35, 36] for sufficient decrease and directionally Lipschitz functions).

Theorem 4.1 *Let $x^* \in \Omega$ be the limit point of a convergent subsequence of unsuccessful iterates $\{x_k^*\}_K$ for which $\lim_{k\in K} \sigma_k = 0$. Assume that f is Lipschitz continuous near x^* with constant $\nu > 0$ and that $T_\Omega^H(x^*) \neq \emptyset$.*

Let $d_k = (x_{k+1}^{\rm local} - x_k^*)/\sigma_k$. Assume that the directions d_k's are such that (i) $\sigma_k\|d_k\|$ tends to zero when σ_k does, and (ii) $\rho(\sigma_k)/(\sigma_k\|d_k\|)$ also tends to zero.

If $d \in T_\Omega^H(x^*)$ is a refining direction associated with $\{d_k/\|d_k\|\}_K$, then $f^\circ(x^*;d) \geq 0$.

If the set of refining directions associated with $\{d_k/\|d_k\|\}_K$ is dense in the unit sphere, then x^* is a Clarke stationary point.

Proof. Let d be a limit point of $\{d_k/\|d_k\|\}_K$. Then it must exist a subsequence K' of K such that $d_k/\|d_k\| \to d$ on K'. On the other hand, we have for all k that

$$x_{k+1}^{\rm local} \;=\; x_k^* + \sigma_k d_k,$$

and, for $k \in K'$, one has

$$f(x_k^* + \sigma_k d_k) \;>\; f(x_k^*) - \rho(\sigma_k).$$

Also, since the direction d_k is bounded above for all k, and so $\sigma_k\|d_k\|$ tends to zero when σ_k does.

Thus, from the definition of the Clarke generalized derivative,

$$f^\circ(x^*;d) \;=\; \limsup_{x\to x^*, t\downarrow 0} \frac{f(x+td)-f(x)}{t} \geq \limsup_{k\in K'} \frac{f(x_k^* + \sigma_k\|d_k\|(d_k/\|d_k\|)) - f(x_k^*)}{\sigma_k\|d_k\|} - r_k,$$

where, from the Lipschitz continuity of f near x^*,

$$r_k \;=\; \frac{f(x_k^* + \sigma_k d_k) - f(x_k^* + \sigma_k\|d_k\|d)}{\sigma_k\|d_k\|} \;\leq\; \nu \left\| \frac{d_k}{\|d_k\|} - d \right\|$$

tends to zero on K'. Finally, since $\|d_k\|$ is bounded away from zero in K',

$$\begin{aligned} f^\circ(x^*;d) &\geq \limsup_{k\in K'} \frac{f(x_k^* + \sigma_k d_k) - f(x_k^*) + \rho(\sigma_k)}{\sigma_k\|d_k\|} - \frac{\rho(\sigma_k)}{\sigma_k\|d_k\|} - r_k \\ &= \limsup_{k\in K'} \frac{f(x_k^* + \sigma_k d_k) - f(x_k^*) + \rho(\sigma_k)}{\sigma_k\|d_k\|} \\ &\geq 0. \end{aligned}$$

To prove the second part, we first conclude from the density of the refining directions on the unit sphere and the continuity of $f^\circ(x^*;\cdot)$ in $T_\Omega^H(x^*)$, that $f^\circ(x^*;d) \geq 0$ for all $d \in T_\Omega^H(x^*)$. Finally, we conclude that $f^\circ(x^*;v) = \lim_{d\in T_\Omega^H(x^*), d\to v} f^\circ(x^*;d) \geq 0$ for all $v \in T_\Omega(x^*)$. ■

The proposed algorithm converges to a Clarke stationary point under the assumption that the set of directions $\{d_k/\|d_k\|\}_k$ is dense in the unit sphere. In practice, such assumption can be

satisfied by switching to greedy search strategy for sufficiently small σ_k instead of maximizing the local acquisition function, this would allow a maximum exploration of the local design space. Another option can be to compute, after a given large number of iterations, x_{k+1}^{local} using a local direct-search method with orthogonal directions to cover the surface of the unit sphere more densely [37].

5 Numerical experiments

The objective of this section is twofold: first, to evaluate the sensitivity of TREGO to its own parameters and perform an ablation study; second, to compare our algorithm with the original EGO and other BO alternatives to show its strengths and weaknesses.

5.1 Design of experiments

5.1.1 Testing procedure using the BBOB benchmark

Our experiments are based on the `COCO` (COmparing Continuous Optimizers, [19]) software. COCO is a recent effort to build a testbed that allows the rigorous comparison of optimizers. We focus here on the noiseless BBOB test suite in the *expensive objective function* setting [38] that contains 15 instances of 24 functions [39]; each function is defined for an arbitrary number of parameters (≥ 2) to optimize. Each instance corresponds to a randomized modification of the original function (rotation of the coordinate system and a random translation of the optimum). The functions are divided into 5 groups: 1) separable, 2) unimodal with moderate conditioning, 3) unimodal with high conditioning, 4) multi-modal with adequate global structure, and 5) multi-modal with weak global structure. Note that group 4 is often seen as the main target for Bayesian optimization. The full description of the functions is available in Appendix (Table 2).

A problem is a pair [function, target to reach]. Therefore, for each instance of a function, there are several problems to solve of difficulty varying with the target value. The *Empirical Run Time Distributions* (ERTD) gives, for a given budget (i.e. number of objective function evaluations), the proportion of problems which are solved by an algorithm. This metric can be evaluated for a single function and dimension, or averaged over a set of functions (typically over one of the 5 groups or over the 24 functions).

To set the target values and more generally define a reference performance, COCO relies on a composite fake algorithm called `best09`. `best09` is made at each optimization iteration of the best performing algorithm of the Black-Box Optimization Benchmarking (BBOB) 2009 [38]. In our experiments, the targets were set at the values reached by `best09` after $[0.5, 1, 3, 5, 7, 10, 15, 20] \times d$ function evaluations.

Note that outperforming `best09` is a very challenging task, as it does not correspond to the performance of a single algorithm but of the best performing algorithm for each instance. In the following, the `best09` performance is added to the plots as a reference. In addition, we added the performance of a purely random search, to serve as a lower bound.

5.1.2 Sensitivity analysis and ablation study

TREGO depends on a number of parameters (see Section 3) and has some additional degrees of freedom worth exploring (see Section 3.2). The objective of these experiments is to answer the following questions:

1. is TREGO sensitive to the initial size of the trust region?

2. is TREGO sensitive to the contraction factor β (see Eq. 7) of the trust region?

3. is using a local model beneficial?

4. is there an optimal ratio between global and local steps?

To answer these questions, we run a default version of TREGO and 9 variants, as reported in Table 1. The contraction parameter β is either 0.9 (which is classical in DFO algorithms) or 0.5 (which corresponds to an aggressive reduction of the trust region). The default initial size of the trust region corresponds to 20% of the volume of the search space, and we test as alternatives 10% and 40%. The global:local ratio varies from 10:1 (which is expected to behave almost similarly to the original EGO) to 1:10 (very local).

Because of the cost of a full COCO benchmark with EGO-like algorithms, the interaction between these parameters is not studied. Also, the ablation experiments are limited to the problems with dimensions 2 and 5 and relatively short runs ($30d$ function evaluations). With these settings and 15 repetitions of each optimization run, an EGO algorithm is tested within a couple of days of computing time on a recent single processor.

5.1.3 Comparison with state-of-the-art algorithms

Longer runs of of length $50d$ (function evaluations) are made with the default TREGO and a version that stresses local search, gl1-4, in dimension 2, 5 and 10. The results are compared to state-of-the-art Bayesian optimization algorithms: a vanilla EGO, that serves as a baseline, TRIKE (see Section 3.3), SMAC and DTS-CMA. A COCO test campaign of an EGO-like algorithm up to dimension 10, with run length of $50d$ and 15 repetitions of the optimizations takes of the order of 3 weeks of computing time on a recent single processor.

DTS-CMA [40] is a surrogate-assisted evolution strategy based on a combination of the CMA-ES algorithm and Gaussian process surrogates. The DTS-CMA solver is known to be very competitive compared to the state-of-the-art black-box optimization solvers particularly on some classes of multimodal test problems. SMAC [41] (in its BBOB version) is a BO solver that uses an isotropic GP to model the objective function and a stochastic local search to optimize the expected improvement. SMAC is known to perform very well early in the search compared to the state-of-the-art blackbox optimizers. The DTS-CMA and SMAC results are directly extracted from the COCO database. This is not the case of TURBO and TRIKE. As TRIKE follows a relatively standard BO framework, we use our own implementation to compare TREGO against it. As TURBO has a complex structure and the available code is too computationally demanding to be used directly with COCO, it is left out of this study.

5.1.4 Implementation details

For a fair comparison, TREGO, EGO and TRIKE are implemented under a unique framework, based on the `R` packages `DiceKriging` (Gaussian process models) and `DiceOptim` (BO) [43, 44]. Our setup aligns with current practices in BO [45, 46], as we detail below.

All GP models use a constant trend and an anisotropic Matérn covariance kernel with smoothness parameter $\nu = 5/2$. The GP hyperparameters are inferred by maximum likelihood after each addition to the training set; the likelihood is maximized using a multi-start

Acronym	Solvers
`rando`	random search
`best09`	best of all BBOB 2009 competitors at each budget [42]
`TRIKE`	TRIKE algorithm of [17]
`SMAC`	SMAC algorithm of [41]
`DTS-CMA`	DTS-CMA algorithm of [40]
`EGO`	original EGO algorithm of [5]
`TREGO`	default TREGO with $\beta = 0.9$, $\sigma_1 = 0.2$, global/local ratio = 1 / 1, initial TR volume = 20% of the search space, and no local model
`gl1-10, gl1-4, gl4-1, gl10-1`	TREGO with a global/local ratio of 1/10, 1/4, 4/1 and 10/1, respectively
`smV0, lgV0`	TREGO with small (10%) and large (40%) initial TR size
`fstC`	TREGO with fast contraction of the TR, i.e., $\beta = 0.5$
`fstCsmV0`	TREGO with fast contraction of the TR and small initial TR
`locGP`	TREGO with a local GP model

Table 1: Names of the compared algorithms. For the TREGO variants, when not specified, the parameter values are the ones of the default, `TREGO`.

L-BFGS scheme. In case of numerical instability, a small regularization value is added to the diagonal of the covariance matrix.

Trust regions are defined using the ℓ_1 norm (see Eq.5), so that they are hyper-rectangles. This allow us to optimize the expected improvement using a multi-start L-BFGS scheme.

Each experiment starts with an initial set of $2d + 4$ observations, generated using latin hypercube sampling improved through a maximin criterion [20]. All BO methods start with the same DoEs, and the DoE is different (varying the seed) for each problem instance.

For `locGP`, the local model uses the same kernel and mean function as the global one, but its hyperparameters are inferred independently. To avoid numerical instability, the local model is always trained on at least $2d + 1$ points. If the trust-region does not contain enough points, the points closest to the center of the trust-region are also added to the training set.

5.2 Results

5.2.1 Sensitivity analysis and ablation study

Figure 2, top row, summarizes our study on the effect of the global versus local iterations ratio. There is measurable advantage of algorithms devoting more iterations to local rather than global search. gl1-4 and gl1-10 consistently outperform gl4-1 and gl10-1. gl1-4 and gl1-10 slightly outperform the TREGO baseline, the effect being more visible with higher dimension (see also Figure 3 for results with 10 dimensions).

By further splitting results into function groups (see Figure 5 in Appendix), it is observed that the performance gain due to having more local iterations happens on the unimodal function groups (the 2nd and 3rd, i.e., unimodal functions with low and high conditioning) when less difference can be observed on multimodal functions (first, fourth and fifth group). For multimodal functions with a weak global structure (fifth group, bottom right plot of Figure 5), gl10-1 is even on average (over the budgets) the best strategy. These findings are intuitive, as unimodal

function may not benefit at all from global steps, while on the other hand a too aggressively local strategy (e.g. gl1-10) may get trapped in a local optimum of a highly multimodal function. Overall on this benchmark, gl1-4 offers the best trade-off over all groups between performance and robustness.

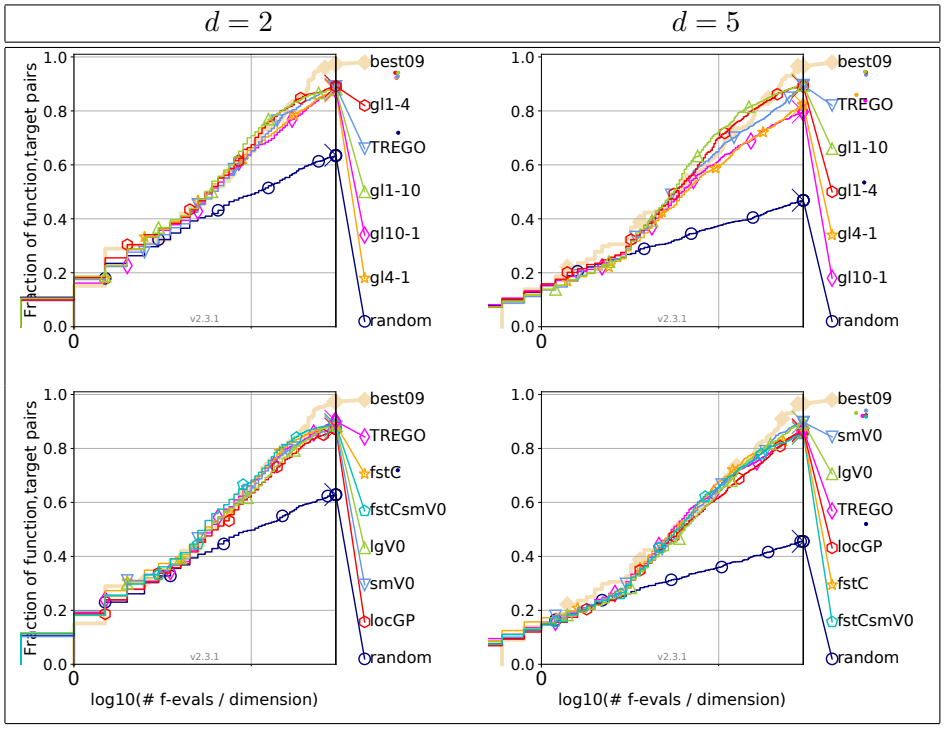

Figure 2: Effect of changing the amount of local and global iterations (top), and changing the other parameters of the TREGO algorithm (bottom). Performance is reported in terms of ERTD, averaged over the entire noiseless BBOB testbed in 2 (left) and 5 (right) dimensions. Run length is $30 \times d$.

Figure 2, bottom row, shows the average performance of other variants of TREGO. Overall, TREGO has very little sensitivity to its internal parameters, the average performances of all TREGO variants being similar in both dimensions. The robustness of TREGO performance with respect to the other parameters is an advantage of the method, and is in line with what is generally observed for trust region based algorithms.

The effects of the TREGO parameters are studied by function groups in Figure 5 (see Appendix). The main visible results are:

- a slightly positive effect of the local GP (locGP) on the groups 1 and 2 but a strong negative effect on unimodal functions with bad conditioning (group 3), and no effect on the remaining groups. Despite offering attractive flexibility in theory, the local GP provides in practice either limited gain or has a negative impact on performance. As this variant is also more complicated than TREGO, it may be discarded.

- a positive effect of fast contraction of the trust region (fstC and fstCsmV0) on highly multimodal functions (group 5) during early iterations. By making the trust region more

local earlier in the search, the fast contraction allows to reach the easy targets, but this early performance prevents the algorithm from finding other better targets later on (those variants being outperformed by others at the end of the runs).

5.2.2 Comparison with state-of-the-art Bayesian optimization algorithms

Figure 3 gives the average performance of the algorithms on all the functions of the testbed. Results in 5 and 10 dimensions split by function groups are provided in Figure 4.

EGO is significantly outperformed by all trust regions algorithms (TREGO, gl1-4 and TRIKE). This performance gap is limited for $d = 2$ but very visible for $d = 5$ and even higher for $d = 10$. It is also significant for any budget (as soon as the shared initialization is done). The improvement is also visible for all function groups (Fig. 4), in particular for groups with strong structure. For the multimodal with weak structure group, the effect is mostly visible for the larger budgets.

SMAC has an early start and is visibly able to start optimizing while all other methods are still creating their initial DoE. However, it is outperformed by all trust region variants before the number of evaluations reaches 10 times the problem dimension (vertical line on the graphs). This effect also increases with dimension.

DTS-CMA has conversely a slower start, so that it is slightly outperformed by trust regions for small budgets ($< 20 \times d$). However, for large budgets and $d = 10$, DTS-CMA largely outperforms other methods on average. However, looking at Fig. 4, DTS-CMA clearly outperforms the other methods (including the best09 baseline) on multimodal functions with strong structure for $d = 10$ and large budgets, while TREGO remains competitive in other cases.

TRIKE has an overall performance comparable to TREGO and gl1-4. For $d = 5$, it slightly outperforms the other methods for intermediate budget values, but looses its advantage for larger budgets. Figure 6 (see Appendix) reveals that this advantage is mainly achieved on the unimodal group with high conditioning, but on multi-modal problems, TREGO and gl1-4's ability to perform global steps offer a substantial advantage.

Overall performance Overall, this benchmark does not reveal a universal winner. SMAC excels with extremely limited budgets, while DTS-CMA outperforms the other methods for the largest dimensions and budgets. TREGO and gl1-4 are overall very competitive on intermediate values, in particular for multi-modal functions.

Discussion It appears clearly from our experiments that trust regions are an efficient way to improve EGO's scalability with dimension. EGO is known to over-explore the boundaries in high dimension [10, 18], and narrowing the search space to the vicinity of the current best point naturally solves this issue. And since EGO is outperformed for any budget, we can conclude that the gain obtained by focusing early on local optima is not lost later by missing the global optimum region. Trust regions also improve performance of EGO on problems for which GPs are not the most natural fit (i.e. unimodal functions). For this class of problems, the most aggressively local algorithm (TRIKE) can perform best in some cases (Fig. 6), however

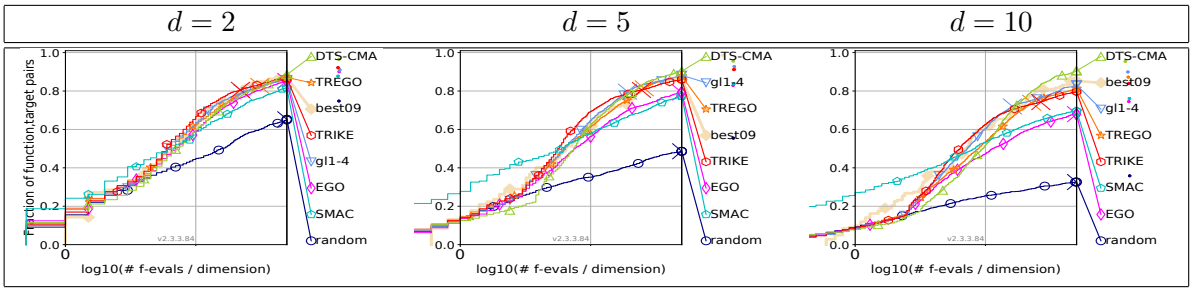

Figure 3: Comparison of the TREGO and gl1-4 with state-of-the-art optimization algorithms, averaged over the entire COCO testbed in 2, 5 and 10 dimensions. Run length $= 50 \times d$.

our more balanced approach is almost as good, if better (Fig. 6, unimodal functions with low conditioning). On the other hand, maintaining a global search throughout the optimization run allows escaping local optima and ultimately delivering better performance for larger budgets (see in particular Fig. 4, all multimodal functions).

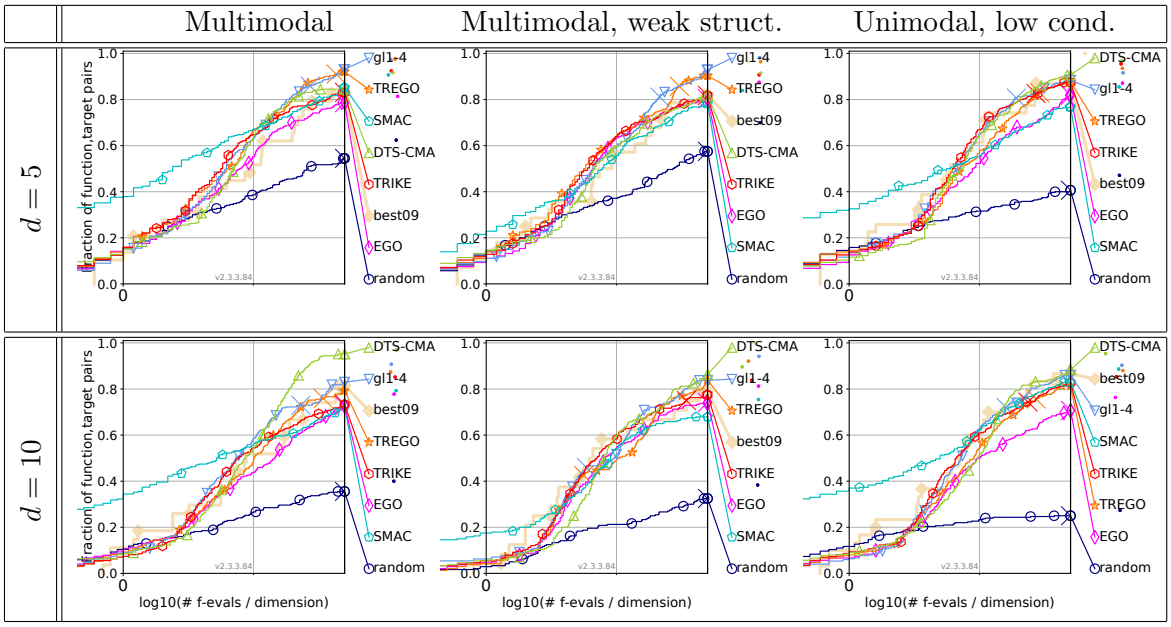

Figure 4: Comparison of TREGO and gl1-4 with state-of-the-art optimization algorithms, averaged over the multi-modal functions with adequate (left, f15 to f19) and weak (middle, f20 to f24) global structure, unimodal functions with low conditioning (right), $d = 5$ (top row) and $d = 10$ (bottom row) dimensions. Run length $= 50 \times d$. Results for the other groups are given in Appendix, Fig. 6.

6 Conclusions and perspectives

In this work, we introduced TREGO, a Bayesian optimization algorithm based on trust region for the optimization of expensive-to-evaluate black-box functions. TREGO builds on the celebrated EGO algorithm by alternating between a standard global step and a local step during which the search is limited to a trust region.

We showed that equipped with such a local step, TREGO rigorously achieves global convergence, while enjoying the flexible predictors and efficient exploration-exploitation trade-off provided by the GPs.

We then performed an extensive benchmark, which allowed us to form the following conclusions:

- TREGO benefits from having a relatively high proportion of local steps, but is otherwise insensitive to its other parameters.
- A more complex approach involving both a local and a global model, which is possible in the TREGO framework, does not provide any benefit.
- TREGO significantly outperforms EGO in all tested situations.
- TREGO is a highly competitive algorithm for multi-modal functions with moderate dimensions and budgets.

Making TREGO a potential overall winner on the experiments reported here is an avenue for future work. This would require improving its performance on unimodal functions with high conditioning, and improving its performance at very early steps, for example by leveraging SMAC for creating the initial DoEs. Another important future work may include the extension of TREGO to the case of noisy observations, following recent results in DFO [47, 48] and established BO techniques [49].

References

[1] Forrester, A.I.J., Sóbester, A., Keane, A.J.: Multi-fidelity optimization via surrogate modelling. Proceedings of the royal society a: mathematical, physical and engineering sciences **463**, 3251–3269 (2007)

[2] Snoek, J., Larochelle, H., Adams, R.P.: Practical Bayesian optimization of machine learning algorithms. In: Advances in neural information processing systems, pp. 2951–2959 (2012)

[3] Picheny, V., Casadebaig, P., Trépos, R., Faivre, R., Silva, D.D., Vincourt, P., Costes, E.: Using numerical plant models and phenotypic correlation space to design achievable ideotypes. Plant, cell & environment **40**, 1926–1939 (2017)

[4] Mockus, J.: Bayesian approach to global optimization: theory and applications. Springer Science & Business Media (2012)

[5] Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black-box functions. J. Global Optim. **13**, 455–492 (1998)

[6] Brochu, E., Cora, V.M., Freitas, N.D.: A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. arXiv preprint arXiv:1012.2599 (2010)

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

[8] Kandasamy, K., Schneider, J., Póczos, B.: High dimensional Bayesian optimisation and bandits via additive models. In: International conference on machine learning, pp. 295–304 (2015)

[9] Bouhlel, M.A., Bartoli, N., Regis, R.G., Otsmane, A., Morlier, J.: Efficient global optimization for high-dimensional constrained problems by using the kriging models combined with the partial least squares method. Engineering Optimization **50**(12), 2038–2053 (2018). DOI 10.1080/0305215X.2017.1419344

[10] Oh, C.Y., Gavves, E., Welling, M.: BOCK: Bayesian optimization with cylindrical kernels. In: International Conference on Machine Learning, pp. 3868–3877 (2018)

[11] Siivola, E., Vehtari, A., Vanhatalo, J., González, J., Andersen, M.R.: Correcting boundary over-exploration deficiencies in Bayesian optimization with virtual derivative sign observations. In: 2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP), pp. 1–6. IEEE (2018)

[12] Vazquez, E., Bect, J.: Convergence properties of the expected improvement algorithm with fixed mean and covariance functions. J. Stat. Plan. and Inference **140**, 3088–3095 (2010)

[13] Bull, A.D.: Convergence rates of efficient global optimization algorithms. Journal of Machine Learning Research **12**, 2879–2904 (2011)

[14] Conn, A.R., Scheinberg, K., Vicente, L.N.: Introduction to Derivative-Free Optimization. MPS-SIAM Series on Optimization. SIAM, Philadelphia (2009)

[15] Audet, C., Hare, W.: Derivative-Free and Blackbox Optimization. Springer, Cham, Philadelphia (2017)

[16] Kolda, T.G., Lewis, R.M., Torczon, V.: Optimization by direct search: New perspectives on some classical and modern methods. SIAM Rev. **45**, 385–482 (2003)

[17] Regis, R.G.: Trust regions in Kriging-based optimization with expected improvement. Eng. Optim. **48**, 1037–1059 (2016)

[18] Eriksson, D., Pearce, M., Gardner, J., Turner, R.D., Poloczek, M.: Scalable global optimization via local Bayesian optimization. In: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (eds.) Advances in Neural Information Processing Systems 32, pp. 5496–5507. Curran Associates, Inc. (2019). URL http://papers.nips.cc/paper/8788-scalable-global-optimization-via-local-bayesian-optimization.pdf

[19] Hansen, N., Auger, A., Mersmann, O., Tusar, T., Brockhoff, D.: Coco: A platform for comparing continuous optimizers in a black-box setting. arXiv preprint arXiv:1603.08785 (2016)

[20] Fang, K.T., Li, R., Sudjianto, A.: Design and modeling for computer experiments. CRC press (2005)

[21] Stein, M.L.: Interpolation of spatial data: some theory for Kriging. Springer Science & Business Media (2012)

[22] Rasmussen, C.E., Williams, C.K.I.: Gaussian processes for machine learning. MIT press Cambridge, MA (2006)

[23] Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., Freitas, N.D.: Taking the human out of the loop: A review of Bayesian optimization. Proceedings of the IEEE **104**, 148–175 (2015)

[24] Le Digabel, S., Wild, S.: A Taxonomy of Constraints in Simulation-Based Optimization. Tech. Rep. G-2015-57, Les cahiers du GERAD (2015). URL http://www.optimization-online.org/DB_HTML/2015/05/4931.html

[25] Booker, A.J., Dennis Jr., J.E., Frank, P.D., Serafini, D.B., Torczon, V., Trosset, M.W.: A rigorous framework for optimization of expensive functions by surrogates. Structural and Multidisciplinary Optimization **17**, 1–13 (1998)

[26] Vaz, A.I.F., Vicente, L.N.: A particle swarm pattern search method for bound constrained global optimization. J. Global Optim. **39**, 197–219 (2007)

[27] Diouane, Y., Gratton, S., Vicente, L.N.: Globally convergent evolution strategies for constrained optimization. Comput. Optim. Appl. **62**, 323–346 (2015)

[28] Srinivas, N., Krause, A., Kakade, S., Seeger, M.: Gaussian process optimization in the bandit setting: No regret and experimental design. In: Proceedings of the 27th International Conference on Machine Learning (2010)

[29] McLeod, M., Roberts, S., Osborne, M.A.: Optimization, fast and slow: optimally switching between local and Bayesian optimization. In: International Conference on Machine Learning, pp. 3443–3452 (2018)

[30] Clarke, F.H.: Optimization and Nonsmooth Analysis. John Wiley & Sons, New York (1983). Reissued by SIAM, Philadelphia, 1990

[31] Audet, C., Dennis Jr., J.E.: Analysis of generalized pattern searches. SIAM J. Optim. **13**, 889–903 (2002)

[32] Jahn, J.: Introduction to the Theory of Nonlinear Optimization. Springer-Verlag, Berlin (1996)

[33] Nocedal, J., Wright, S.J.: Numerical Optimization, second edn. Springer-Verlag, Berlin (2006)

[34] Audet, C., Dennis Jr., J.E.: Mesh adaptive direct search algorithms for constrained optimization. SIAM J. Optim. **17**, 188–217 (2006)

[35] Vicente, L.N., Custódio, A.L.: Analysis of direct searches for discontinuous functions. Math. Program. **133**, 299–325 (2012)

[36] Diouane, Y., Gratton, S., Vicente, L.N.: Globally convergent evolution strategies. Math. Program. **152**, 467–490 (2015)

[37] Abramson, M.A., Audet, C., Dennis, J.E., Digabel, S.L.: OrthoMADS: A Deterministic MADS Instance with Orthogonal Directions. SIAM J. Optim. **20**, 948–966 (2009)

[38] Hansen, N., Auger, A., Ros, R., Finck, S., Pošík, P.: Comparing results of 31 algorithms from the black-box optimization benchmarking bbob-2009. In: Proceedings of the 12th annual conference companion on Genetic and evolutionary computation, pp. 1689–1696. ACM (2010)

[39] Brockhoff, D.: Online description of the BBOB functions. `https://coco.gforge.inria.fr/` (2006)

[40] Bajer, L., Pitra, Z., Repický, J., Holena, M.: Gaussian Process Surrogate Models for the CMA Evolution Strategy. Evolutionary Computation **27**, 665–697 (2019)

[41] Hutter, F., Hoos, H.H., Leyton-Brown, K.: Sequential model-based optimization for general algorithm configuration. In: Proceedings of the 5th International Conference on Learning and Intelligent Optimization, LION'05, pp. 507–523. Springer-Verlag, Berlin, Heidelberg (2011)

[42] Auger, A., Finck, S., Hansen, N., Ros, R.: BBOB 2009: Comparison Tables of All Algorithms on All Noiseless Functions. Technical Report RT-0383, INRIA (2010). URL `https://hal.inria.fr/inria-00471251`

[43] Roustant, O., Ginsbourger, D., Deville, Y.: DiceKriging, DiceOptim: Two R Packages for the Analysis of Computer Experiments by Kriging-Based Metamodeling and Optimization. Journal of Statistical Software **51** (2012)

[44] Picheny, V., Ginsbourger, D.: Noisy Kriging-based optimization methods: a unified implementation within the DiceOptim package. Computational Statistics & Data Analysis **71**, 1035–1053 (2014)

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

[46] Frazier, P.I.: A tutorial on Bayesian optimization. arXiv preprint arXiv:1807.02811 (2018)

[47] Chen, R., Menickelly, M., Scheinberg, K.: Stochastic optimization using a trust-region method and random models. Math. Program. **169**, 447–487 (2018)

[48] Audet, C., Dzahini, K., Kokkolaras, M., Digabel., S.L.: StoMADS: Stochastic blackbox optimization using probabilistic estimates. Tech. rep., Les Cahiers du GERAD G-2019-30 (2019)

[49] Picheny, V., Wagner, T., Ginsbourger, D.: A benchmark of kriging-based infill criteria for noisy optimization. Structural and Multidisciplinary Optimization **48**(3), 607–626 (2013)

A Pseudo-code of the TREGO algorithm

Algorithm 1: A trust-Region framework for EGO (TREGO).

Data: Create an initial DoE $\mathcal{D}_{t_0}$ of t_0 points in a given set $\Omega \subset \mathbb{R}^n$ with a given method. Set $\mathcal{Y}_{t_0} = \{f(x_i) \mid \forall\, x_i \in \mathcal{D}_{t_0}\}$. Choose $G \geq 0$ the number of the global steps and $L \geq 1$ the number of the local steps. Initialize the step-size parameter σ_0, $x_0^* \in \mathcal{D}_{t_0}$, choose the constants $\beta_{\min}$ and $\beta_{\max}$ such that $0 < \beta_{\min} \leq \beta_{\max} < 1$ and $0 < d_{\min} < d_{\max}$. Select a forcing function $\rho(.)$ and set $k = 0$ and $t = t_0$;

while *some stopping criterion is not satisfied* **do**

`/* A global phase over Ω: */`

for $i = 1, \ldots, G$ **do**

Step 1 (global acquisition function maximization):
Set

$$x_t^{\text{global}} := \underset{x \in \Omega}{\text{argmax}}\ \alpha(x; \mathcal{D}_t);$$

Step 2 (update the DoE): Set $\mathcal{D}_{t+1} = \mathcal{D}_t \cup \left\{x_t^{\text{global}}\right\}$ and $\mathcal{Y}_{t+1} = \mathcal{Y}_t \cup \left\{f\left(x_t^{\text{global}}\right)\right\}$;
Increment t;

end

Let x_{k+1}^{global} be the best point (in term of f) in the DoE $\mathcal{D}_t$;
Step 3 (imposing sufficient decrease globally):
if $f(x_{k+1}^{\text{global}}) \leq f(x_k^*) - \rho(\sigma_k)$ **then**
the global phase is successful, set $x_{k+1}^* = x_{k+1}^{\text{global}}$ and $\sigma_{k+1} \geq \sigma_k$;
else

`/* A local phase over Ω_k: */`

for $i = 1, \ldots, L$ **do**

Step 4 (local acquisition function maximization):
Set

$$x_t^{\text{local}} := \underset{x \in \Omega_k}{\text{argmax}}\ \alpha(x; \mathcal{D}_t),$$

where $\Omega_k = \{x \in \Omega \mid d_{\min}\sigma_k \leq \|x - x_k^*\| \leq d_{\max}\sigma_k\}$;
Step 5 (update the DoE): Set $\mathcal{D}_{t+1} = \mathcal{D}_t \cup \left\{x_t^{\text{local}}\right\}$ and $\mathcal{Y}_{t+1} = \mathcal{Y}_t \cup \left\{f\left(x_t^{\text{local}}\right)\right\}$;
Increment t;

end

Let x_{k+1}^{local} be the best point (in term of f) in the DoE $\mathcal{D}_t$;
Step 6 (imposing sufficient decrease locally):
if $f(x_{k+1}^{\text{local}}) \leq f(x_k^*) - \rho(\sigma_k)$ **then**
the local phase and iteration are successful, set $x_{k+1}^* = x_{k+1}^{\text{local}}$ and $\sigma_{k+1} \geq \sigma_k$;
else
the local phase and iteration are not successful, set $x_{k+1}^* = x_k^*$, and $\sigma_{k+1} = \beta_k \sigma_k$ with $\beta_k \in (\beta_{\min}, \beta_{\max})$;
end

end

Increment k;

end

B Functions of the BBOB noiseless testbed

ID	name	comments
		separable functions
f1	Sphere	unimodal, allows to checks numerical accuracy at convergence
f2	Ellipsoidal	unimodal, conditioning $\approx 10^6$
f3	Rastrigin	10^d local minima, spherical global structure
f4	Büche-Rastrigin	10^d local minima, asymmetric global structure
f5	Linear Slope	linear, solution on the domain boundary
		functions with low or moderate conditioning
f6	Attractive Sector	unimodal, highly asymmetric
f7	Step Ellipsoidal	unimodal, conditioning ≈ 100, made of many plateaus
f8	Original Rosenbrock	good points form a curved $d-1$ dimensional valley
f9	Rotated Rosenbrock	rotated f8
		unimodal functions with high conditioning $\approx 10^6$
f10	Ellipsoidal	rotated f2
f11	Discus	a direction is 1000 times more sensitive than the others
f12	Bent Cigar	non-quadratic optimal valley
f13	Sharp Ridge	resembles f12 with a non-differentiable bottom of valley
f14	Different Powers	different sensitivities w.r.t. the x_i's near the optimum
		multimodal functions with adequate global structure
f15	Rastrigin	rotated and asymmetric f3
f16	Weierstrass	highly rugged and moderately repetitive landscape, non unique optimum
f17	Schaffers F7	highly multimodal with spatial variation of frequency and amplitude, smoother and more repetitive than f16
f18	moderately ill-conditioned Schaffers F7	f17 with conditioning ≈ 1000
f19	Composite Griewank-Rosenbrock	highly multimodal version of Rosenbrock
		multimodal functions with weak global structure
f20	Schwefel	2^d most prominent optima close to the corners of a shrinked and rotated rectangle
f21	Gallagher's Gaussian 101-me peaks	101 optima with random positions and heights, conditioning ≈ 30
f22	Gallagher's Gaussian 21-hi peaks	21 optima with random positions and heights, conditioning ≈ 1000
f23	Katsuura	highly rugged and repetitive function with more than 10^d global optima
f24	Lunacek bi-Rastrigin	highly multimodal function with 2 funnels, one leading to a local optimum and covering about 70% of the search space

Table 2: Functions of the BBOB noiseless testbed, divided in groups.

C Complementary experimental results

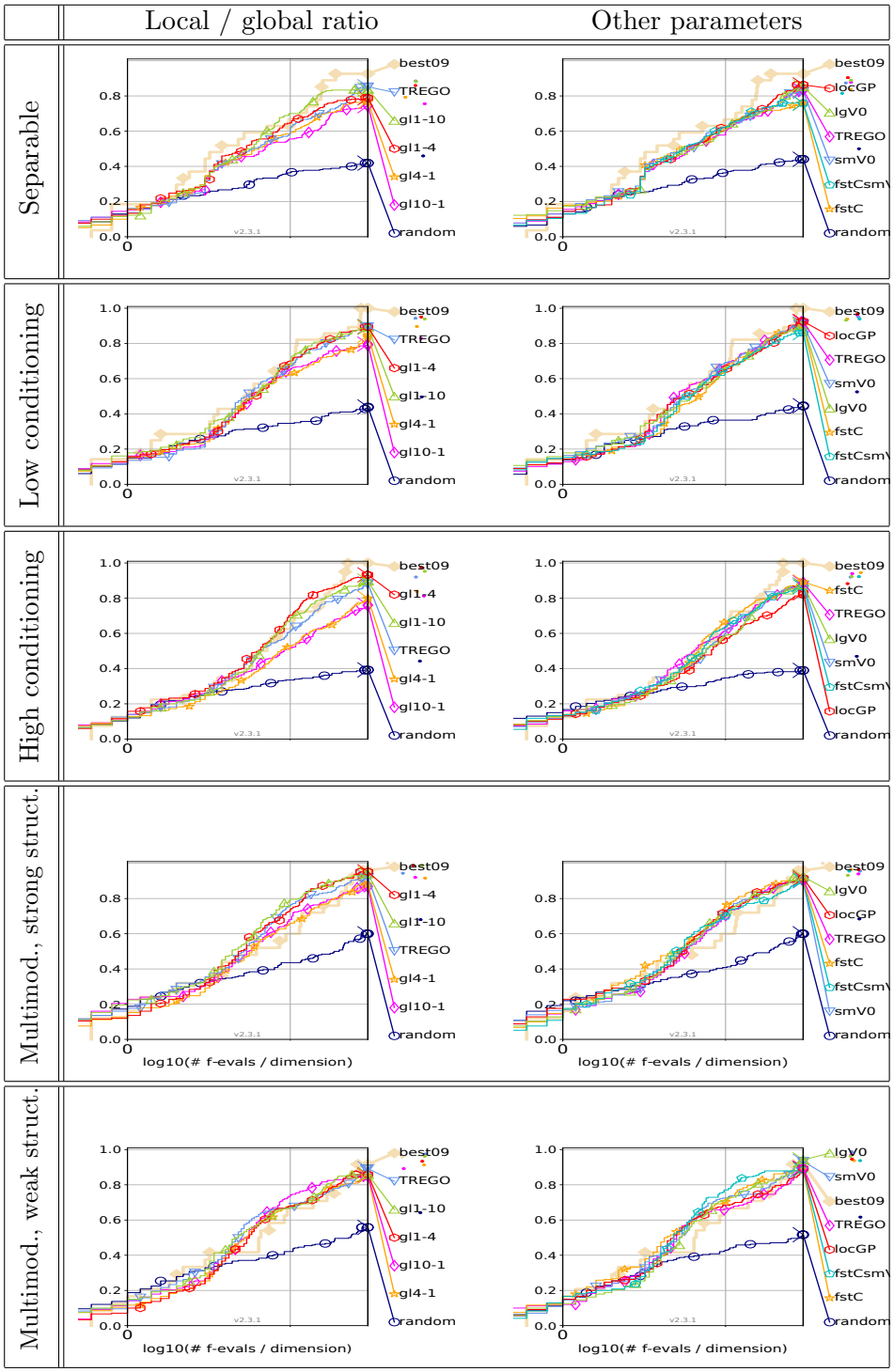

Figure 5: Effect of changing parameters of the TREGO algorithm, averaged by function groups for $d = 5$. Run length is $30 \times d$.

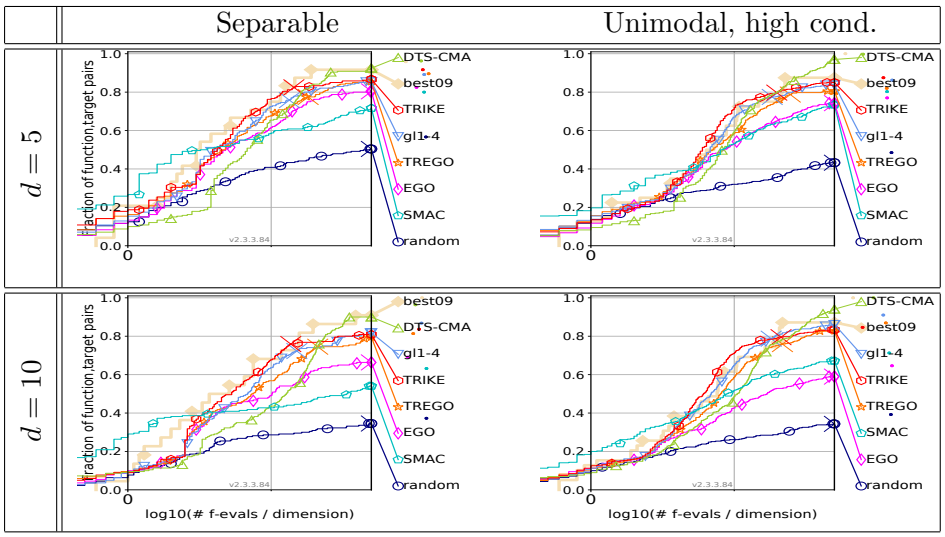

Figure 6: Comparison of TREGO and gl1-4 with state-of-the-art optimization algorithms on separable (left) and unimodal with high conditioning functions (right), for $d = 5$ (top) and $d = 10$ (bottom). Run length $= 50 \times d$.