# TREGO: a Trust-Region Framework for Efficient Global Optimization

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

October 11, 2022

**Please cite as**: "Youssef Diouane, Victor Picheny, Rodolphe Le Riche and Alexandre Scotto Di Perrotolo, TREGO: a Trust-Region Framework for Efficient Global Optimization, Journal of Global Optimization, doi.org/10.1007/s10898-022-01245-w, October 2022"

**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, a trust-region framework for EGO (TREGO) is proposed and analyzed. 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), the proposed algorithm enjoys global convergence properties, while departing from EGO only for a subset of optimization steps. Using extensive numerical experiments based on the well-known COCO bound constrained problems, 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 black-box optimization methods.

**Keywords:** non-linear optimization; black-box 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 [26] to automatic machine learning [54] to crop selection [43]. 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 [34, 40] for early works or [14] for a recent review.

[*] Department of Mathematics and Industrial Engineering, Polytechnique Montréal. E-mail: youssef.diouane@polymtl.ca

[†] 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

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 $\Omega \subseteq \mathbb{R}^n$, with $n$ relatively small (say, 2 to 20). In this work, the objective function $f : \mathbb{R}^n \to \mathbb{R}$ is assumed observable exactly (i.e., without random noise), bounded from below in $\mathbb{R}^n$ and Lipschitz continuous near appropriate limit points. The constraints set $\Omega$ will be treated as explicit [i.e. not relying on estimates, as in 51] and non-relaxable [38], meaning that the objective function cannot be evaluated outside the feasible region. In our numerical experiments, $\Omega$ will be set as a bound constraints set.

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 [13, 35, 60] or incentivizing sampling away from the boundaries [42, 53]. 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 [58] showed that the sequence of evaluation points is dense in the search domain providing some strong assumptions on the objective function. Bull [16] built upon this result to provide a convergence rate for EGO when GP models with a Matérn kernel are used. However, the proposed convergence rate requires the addition of a well-calibrated epsilon-greedy strategy to EGO and it is 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) [5, 19]. 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 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 [36]. This is in line with recent propositions hybridizing BO and DFO [24, 48] that showed great promise empirically, but with limited theoretical guarantees. The proposed TREGO algorithm (Trust-Region framework for Efficient Global Optimization) benefits from both worlds: 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 [16, 58].

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. Intensive numerical experiments have been carried out using the COCO test bed [30]. They represent months of CPU time and have allowed to study TREGO and compare it with state-of-the-art alternatives. These experiments are reported in Section 5. Conclusions and perspectives are finally provided in Section 6. By default this paper uses $\ell_2$ norms.

## 2 The Efficient Global Optimization Framework

Efficient Global Optimization [EGO, 34] 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 [25];
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 $\Omega$;
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 assuming that $f$ is a realization of a Gaussian process (GP) $(Y_x)_{x\in\Omega} \sim \mathcal{GP}\left(m, c\right)$, with prior mean function $m(x) := \mathbb{E}(Y_x)$ and covariance function $c(x, x') := \operatorname{cov}(Y_x, Y_{x'})$, $x, x' \in \Omega$. Given a DoE of size $t \in \mathbb{N}^*$, i.e., $\mathcal{D}_t = \{x_1, x_2, \ldots, x_t\}$ and $\mathcal{Y}_t = \{f(x_1), f(x_2), \ldots, f(x_t)\}$, the posterior distribution of the process conditioned by $\mathcal{D}_t, \mathcal{Y}_t$ is Gaussian with mean and covariance given by [47]:

$$\begin{aligned} m_t(x) &:= m(x) + \lambda_t(x)\left(Y_t - M_t\right), \\ c_t(x, x') &:= c(x, x') - \lambda_t(x) c_t(x'), \end{aligned}$$

where $\lambda_t(x) := c_t(x)^\top C_t^{-1}$, $c_t(x) := (c(x, x_1), c(x, x_2), \ldots, c(x, x_t))^\top$, $C_t := (c(x_i, x_j))_{1\le i,j\le t}$, $Y_t := (f(x_1), f(x_2), \ldots, f(x_t))^\top$ and $M_t := (m(x_1), m(x_2), \ldots, m(x_t))^\top$.

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 $f$. Corresponding hyperparameters are often obtained as maximum likelihood estimates; see for example [47, 56] for the corresponding details.

Once the surrogate model is built, an acquisition function 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 the acquisition function only depends on the probabilistic surrogate model and usually integrates a trade-off between exploitation (i.e., low $m_t(x)$) and exploration (i.e., high $c_t(x, x)$) [27]. In the noise-free setting, the canonical acquisition is Expected Improvement (EI) [34], i.e.,

$$\mathrm{EI}_t(x) \quad := \quad (f_{\min} - m_t(x))\Phi\left(\frac{f_{\min} - m_t(x)}{\sqrt{c_t(x, x)}}\right) + \sqrt{c_t(x, x)}\phi\left(\frac{f_{\min} - m_t(x)}{\sqrt{c_t(x, x)}}\right),$$

where $f_{\min} = \min_{1\le i\le t}(f(x_i))$. The functions $\phi$ and $\Phi$ denote 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 [52] for a recent review. Note that while the focus here is on EI for simplicity, the proposed 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

$$x_{k+1}^{\text{global}} \in \underset{x \in \Omega}{\operatorname{argmax}}\ \alpha(x; \mathcal{D}_t). \tag{2}$$

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

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 ensures 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.

# 3 A Trust-Region Framework for EGO (TREGO)

In this section, we propose a modified version of EGO where a control parameter is included (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 [12, 19, 22, 57]. 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. 2. 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 ( the trust-region $\Omega_k$, detailed hereafter), so that

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

Associated with a proper management of the trust-region $\Omega_k$, 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 is not 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, some form of *sufficient decrease condition* is imposed on the objective function values to declare an iteration successful. Second, the size of the steps taken at each iteration is controlled using a parameter $\sigma_k$ that is updated depending on the sufficient decrease condition (increased if successful, decreased otherwise). Given a current best point $x_k^*$, at iteration $k$, a trust-region around $x_k^*$ is defined as

$$\Omega_k := \{x \in \Omega : d_{\min}\sigma_k \leq \|x - x_k^*\| \leq d_{\max}\sigma_k\}, \tag{4}$$

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 $\Omega_k$ 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. Note that the definition of the trust-region as given in (4) uses the $\ell_2$ norm, however other norms can be preferred depending on the nature of the constraints set $\Omega$. For instance, if $\Omega$ contains only bound constraints, it is more practical to use the $\ell_1$ norm as we will do in our experiments.

At each iteration of the local phase, the following sufficient decrease condition on the objective function is imposed:

$$f(x_{k+1}^{\text{local}}) \leq f(x_k^*) - \rho(\sigma_k), \tag{5}$$

where $\rho : \mathbb{R}^+ \to \mathbb{R}^+$ is a forcing function [36], i.e., a positive continuous nondecreasing function such that $\rho(\sigma)/\sigma \to 0$ when $\sigma \downarrow 0$ (for instance, $\rho(\sigma) = \sigma^2$). The step size parameter $\sigma_k$ is increased if the iteration is successful, i.e., $\sigma_{k+1} = \gamma\sigma_k$ with $\gamma \in (1, +\infty)$. An iteration is declared successful if the new iterate $x_{k+1}^*$ decreases sufficiently the objective function. In this case, the iterate $x_{k+1}^*$ can be updated either within the global phase, i.e., $x_{k+1}^* = x_{k+1}^{\text{global}}$, or the local one, i.e., $x_{k+1}^* = x_{k+1}^{\text{local}}$. If the sufficient decrease condition (5) is not satisfied, the current iterate is kept unchanged, i.e., $x_{k+1}^* = x_k^*$, and the step size is reduced, $\sigma_{k+1} = \beta\sigma_k$ with $\beta \in (0, 1)$. A classical scheme is to keep $\beta \in (0, 1)$ constant, and apply:

$$\begin{aligned} \sigma_{k+1} &= \frac{\sigma_k}{\beta} \quad \text{if the iteration is successful,} \\ \sigma_{k+1} &= \beta\sigma_k \quad \text{otherwise.} \end{aligned} \tag{6}$$

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

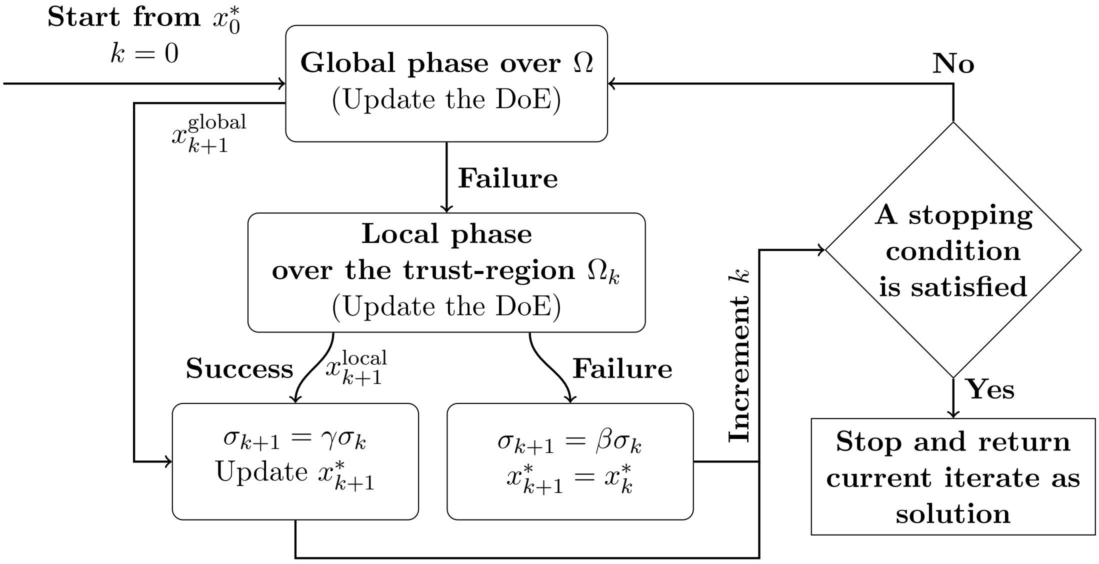

Figure 1: An overview of the TREGO framework. A detailed description is given in Algorithm 1.

### 3.2 Extensions

We now present several possible extensions to TREGO. Some of these extensions are tested in the ablation study of Section 5.3.

**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.

**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 [55] 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 [39]. In fact, as far as the condition (5) is used to decide whether the current iteration is successful or not, the convergence theory of the next section applies.

### 3.3 Related work

**TRIKE [48]** (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 of the expected improvement and the actual improvement, rather than sufficient decrease; 2) TRIKE does not have a global phase. In [48], TRIKE is associated with a restart strategy to ensure global search.

**TURBO [24]** (a TrUst-Region BO solver) carries out a collection of simultaneous BO runs using independent GP surrogate models, each within a different trust region. The trust-region radius is updated with a failure/success mechanism based on the progress made on the objective function[1]. At each iteration, managed by an implicit multi-armed bandit strategy, a global

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

phase 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 [12, 22, 19, 57] 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. In the non-smooth case, the first-order necessary conditions mean that, for any direction $d$, the Clarke generalized derivative [18] along the direction $d$ is non-negative.

In order to achieve our goal, the following additional assumption on the forcing function $\rho(\cdot)$ is made.

**Assumption 4.1** *The forcing function $\rho(\cdot)$ satisfies the following properties:*

- $\rho(\cdot)$ *is a positive continuous nondecreasing function such that $\rho(\sigma)/\sigma \to 0$ when $\sigma \downarrow 0$.*
- *there exist constants $\bar{\gamma}$ and $\bar{\beta}$ satisfying $0 < \bar{\beta} < 1 < \bar{\gamma}$, such that, for each $\sigma > 0$, one has*

$$\rho(\beta\sigma) \;\le\; \bar{\beta}\rho(\sigma) \qquad \textit{and} \qquad \rho(\gamma\sigma) \;\le\; \bar{\gamma}\rho(\sigma).$$

Such assumption is not restrictive as it holds in particular for the classical forcing functions of the form $\rho(\sigma) = c\sigma^q$ with $c > 0$ and $q \ge 1$. The next lemma shows that, as far as the objective function is bounded below, the series $\sum_{k=0}^{+\infty} \rho(\sigma_k)$ is bounded above. The proof of the lemma is inspired by what is done in DFO [4, 10, 11] when handling stochastic noisy estimates of the objective function.

**Lemma 4.1** *Under Assumption 4.1, consider TREGO without any stopping criterion. Let $f$ be bounded below by $f_{\text{low}} \in \mathbb{R}$. Then, one has*

$$\sum_{k=0}^{+\infty} \rho(\sigma_k) \le \frac{(\bar{\gamma} - \bar{\beta})\,(f(x_0^*) - f_{\text{low}}) + \rho(\sigma_0)}{1 - \bar{\beta}} < \infty.$$

**Proof.** For the sake of our proof, the following function is introduced

$$\phi_k \;:=\; \bar{\nu}(f(x_k^*) - f_{\text{low}}) + (1 - \bar{\nu})\rho(\sigma_k), \tag{7}$$

where $\bar{\nu} := \frac{\bar{\gamma}-\bar{\beta}}{1+\bar{\gamma}-\bar{\beta}} \in (0, 1)$. Then, if an iteration $k$ is unsuccessful $x_{k+1}^* = x_k^*$ and $\sigma_{k+1} = \beta\sigma_k$, this leads to

$$\phi_{k+1} - \phi_k = (1 - \bar{\nu})(\rho(\sigma_{k+1}) - \rho(\sigma_k)) \le (1 - \bar{\nu})(\bar{\beta} - 1)\rho(\sigma_k) = -\left(\frac{1 - \bar{\beta}}{1 + \bar{\gamma} - \bar{\beta}}\right)\rho(\sigma_k) \tag{8}$$

where $\rho(\beta\sigma_k) \le \bar{\beta}\rho(\sigma_k)$ and $\bar{\nu} = \frac{\bar{\gamma}-\bar{\beta}}{1+\bar{\gamma}-\bar{\beta}}$ were used.

Otherwise, if the iteration $k$ is successful, then $x^*_{k+1}$ is changed and $\sigma_{k+1} = \gamma\sigma_k$. Then, by using the fact that $\rho(\gamma\sigma_k) \le \bar{\gamma}\rho(\sigma_k)$ and $\bar{\nu} = \frac{\bar{\gamma}-\bar{\beta}}{1+\bar{\gamma}-\bar{\beta}}$, one has

$$\phi_{k+1} - \phi_k \quad \le \quad -\bar{\nu}\rho(\sigma_k) + (1-\bar{\nu})(\bar{\gamma}-1)\rho(\sigma_k) = -\left(\frac{1-\bar{\beta}}{1+\bar{\gamma}-\bar{\beta}}\right)\rho(\sigma_k). \tag{9}$$

Hence, from (8) and (9), one deduces that for any iteration $k$, one gets

$$\phi_{k+1} - \phi_k \;\le\; -\left(\frac{1-\bar{\beta}}{1+\bar{\gamma}-\bar{\beta}}\right)\rho(\sigma_k). \tag{10}$$

Thus, by applying the sum over the subscript $k$, one gets for a given iteration index $n$

$$\phi_{n+1} - \phi_0 = \sum_{k=0}^{n} \phi_{k+1} - \phi_k = \le -\left(\frac{1-\bar{\beta}}{1+\bar{\gamma}-\bar{\beta}}\right)\sum_{k=0}^{n} \rho(\sigma_k).$$

Since $\phi_{n+1} \ge 0$, one deduces that by taking $n \to \infty$

$$\sum_{k=0}^{+\infty} \rho(\sigma_k) \le \frac{(1+\bar{\gamma}-\bar{\beta})\phi_0}{1-\bar{\beta}} = \frac{(\bar{\gamma}-\bar{\beta})\left(f(x^*_0) - f_{\text{low}}\right) + \rho(\sigma_0)}{1-\bar{\beta}} < \infty.$$

■

From Lemma 4.1, one concludes that the full sequence $\{\rho(\sigma_k)\}$ must converge to zero. Then, by assuming that the forcing function is $\rho(\sigma) = c\sigma^q$ with $c > 0$ and $q \ge 1$, one gets $\lim_{k\to+\infty} \sigma_k = 0$. The result is stated in the next theorem.

**Theorem 4.1** *Consider TREGO without any stopping criterion and with a forcing function of the form $\rho(\sigma) = c\sigma^q$ with $c > 0$ and $q \ge 1$. Then, if the objective function $f$ is bounded below, one gets*

$$\lim_{k\to+\infty} \sigma_k = 0.$$

The following definition, similar to those in [2, 4, 5, 7], is now introduced to show the existence of convergent subsequences of TREGO iterates.

**Definition 4.1** *[4, Definition 5] A convergent subsequence $\{x^*_k\}_{k\in\mathcal{K}}$ of TREGO iterates (for some subset of indices $\mathcal{K}$) is said to be a refining subsequence, if and only if $\{\sigma_k\}_{k\in\mathcal{K}}$ converges to zero. The limit $x^*$ of $\{x^*_k\}_{k\in\mathcal{K}}$ is called a refined point.*

Assuming that TREGO is producing iterates that lie in a compact set, one can ensure the existence of a refining subsequence using the Bolzano-Weierstrass theorem.

**Theorem 4.2** *Consider TREGO without any stopping criterion and with a forcing function of the form $\rho(\sigma) = c\sigma^q$ with $c > 0$ and $q \ge 1$. Let $f$ be bounded below. If the sequence $\{x^*_k\}$ lies in a compact set, then there exists a convergent refining subsequence $\{x^*_k\}_{k\in\mathcal{K}}$.*

The proposed convergence analysis will rely on iterates from the local phase. Thus, in what comes next, the sequence $\{\hat{x}_k^{\text{local}}\}_{k\in\mathcal{K}'} \subseteq \{x_k^*\}_{k\in\mathcal{K}}$, where $\mathcal{K}' \subseteq \mathcal{K}$ is an infinite subset of indices, will be used to denote a refining subsequence associated with TREGO local phase iterates. 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, see [2, 4, 5, 7].

**Definition 4.2** *Consider a convergent refining subsequence associated with the TREGO local phase $\{\hat{x}_k^{\text{local}}\}_{k\in\mathcal{K}'}$ and its corresponding refined point $x^*$. Let $\{d_k\}_{k\in\mathcal{K}'}$ be a sequence such that, $d_k := (x_{k+1}^{\text{local}} - \hat{x}_k^{\text{local}})/\sigma_k$, for all $k \in \mathcal{K}'$. A direction $d$ is said to be a refining direction for $x^*$ if and only if there exists an infinite subset $\mathcal{L} \subseteq \mathcal{K}'$ such that $\lim_{k\in\mathcal{L}} d_k = d$.*

Note that by construction, one has $d_{\min} \leq \|d_k\| \leq d_{\max}$, for all $k \in \mathcal{K}'$. Thus, the existence of a refining direction $d$ is justified as the sequence $\{d_k\}_{k\in\mathcal{K}'}$ lies in a compact set.

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 [18], adapted by Jahn [33] 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 $\Omega$ at $x^*$. In the following definition from [2, 19], the notation $B(x; \Delta) := \{y \in \mathbb{R}^n : \|y - x\| < \Delta\}$ will be used to denote the open ball of radius $\Delta$ centered at $x$.

**Definition 4.3** *[2, Definition 3.3] 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 $\Omega$ 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 $\Omega$ at $x$, denoted by $T_\Omega^H(x)$, is the set of all hypertangent vectors to $\Omega$ at $x$. Then, the Clarke tangent cone to $\Omega$ at $x$ (denoted by $T_\Omega(x)$) can be defined as the closure of the hypertangent cone $T_\Omega^H(x)$. The Clarke tangent cone generalizes the notion of tangent cone in nonlinear programming [41]. In the following definition from [2, 5, 18, 19], the formal notion of the Clarke tangent cone is detailed.

**Definition 4.4** *[2, Definition 3.5] 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 $\Omega$ if for every sequence $\{y_k\}$ of elements of $\Omega$ 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 $y_k + t_k w_k \in \Omega$, for a sufficiently large $k$. The set $T_\Omega(x)$ of all Clarke tangent vectors to $\Omega$ at $x$ is called the Clarke tangent cone to $\Omega$ at $x$.*

If we assume that $f$ is Lipschitz continuous near $x^*$ and by using [2, Propostion 3.5], then for any direction $v$ in the Clarke tangent cone, one can consider the Clarke-Jahn generalized derivative to $\Omega$ at $x^*$ 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 Clarke stationary point is being equivalent to $\nabla f(x^*)^\top d \geq 0$ , $\forall d \in T_\Omega(x^*)$.

It remains now to state the next lemma which will be useful for the proof of the optimality result based on the Clarke derivative. The proof of this lemma is inspired by [4, Theorem 4].

**Lemma 4.2** *Consider TREGO without any stopping criterion and using a forcing function of the form $\rho(\sigma) = c\sigma^q$ with $c > 0$ and $q \geq 1$. Then, if the objective function $f$ is bounded below, one has*

$$\liminf_{k \to +\infty} \frac{f(x_k^*) - f(x_k^* + \sigma_k d_k)}{\sigma_k} \ \leq \ 0.$$

**Proof.** By contradiction, assume that there exists $\epsilon > 0$ such that,

$$\frac{f(x_k^*) - f(x_k^* + \sigma_k d_k)}{\sigma_k} \ \geq \ \epsilon, \qquad \text{for all } k \in \mathbb{N}. \tag{11}$$

From Theorem 4.1, one has $\lim_{k\to+\infty} \sigma_k = 0$, hence by using the forcing function properties one has also $\lim_{k\to+\infty} \frac{\rho(\sigma_k)}{\sigma_k} = 0$. This means that there exists $k_0 > 0$, such that

$$\rho(\sigma_k) \ \leq \ \epsilon\sigma_k, \qquad \text{for all } k \geq k_0. \tag{12}$$

By combining (11) and (12), one gets

$$f(x_k^*) - f(x_k^* + \sigma_k d_k) \geq \rho(\sigma_k), \qquad \text{for all } k \geq k_0.$$

Hence, for all $k \geq k_0$, the $k$-th iteration of TREGO is successful and $\sigma_{k+1} = \gamma\sigma_k$ with $\gamma > 1$. This contradicts $\lim_{k\to+\infty} \sigma_k = 0$ and thus the claim (11) is false. ■

The next theorem states the global convergence of TREGO. The obtained result is in the vein of those first established in [2, Theorem 3.2] for simple decrease and Lipschitz continuous functions and later generalized in [21, 59] for sufficient decrease and directionally Lipschitz functions.

**Theorem 4.3** *Let the assumptions made in Theorem 4.1 hold. Let $x^* \in \Omega$ be a refined point of a refining subsequence associated with the TREGO local phase $\{\hat{x}_k^{\text{local}}\}_{k\in\mathcal{K}'}$. Assume that $f$ is Lipschitz continuous near $x^*$ and that $T_\Omega^H(x^*) \neq \emptyset$. Let $d \in T_\Omega^H(x^*)$ be a refining direction associated with $\{d_k\}_{k\in\mathcal{K}'}$. Then, the Clarke-Jahn generalized derivative of $f$ at $x^*$ in the direction $d$ is nonnegative, i.e., $f^\circ(x^*; d) \geq 0$.*

**Proof.** In fact, from Lemma 4.2, there exists a subset $\mathcal{K}$ such that

$$\lim_{k\in\mathcal{K}} \frac{f(x_k^*) - f(x_k^* + \sigma_k d_k)}{\sigma_k} \ \leq \ 0.$$

From Theorem 4.1, one has also $\lim_{k\in\mathcal{K}} \sigma_k = 0$. Now, by using Theorem 4.2, there exists a subset $\mathcal{K}'' \subseteq \mathcal{K}$ such that $\lim_{k\in\mathcal{K}''} x_k^* = x^*$. Consider now $\mathcal{K}' \subseteq \mathcal{K}''$ an infinite subset of indices such that $\{\hat{x}_k^{\text{local}}\}_{k\in\mathcal{K}'} \subseteq \{x_k^*\}_{k\in\mathcal{K}''}$ is a refining subsequence associated with TREGO local phase iterates. Then, since the subsequence $\{d_k\}_{k\in\mathcal{K}'}$ lies in a compact set, there must exist a subset $\mathcal{L} \subseteq \mathcal{K}'$ such that $\{d_k\}_{k\in\mathcal{L}}$ converges to $d$ and

$$\lim_{k\in\mathcal{L}} \frac{f(\hat{x}_k^{\text{local}}) - f(\hat{x}_k^{\text{local}} + \sigma_k d_k)}{\sigma_k} \quad \leq \quad 0. \tag{13}$$

From the Lipschitz continuity of $f$ near $x^*$ and using [2, Proposition 3.9], one deduces that the Clarke generalized derivative is continuous with respect to $d$ on the Clarke tangent cone. Hence,

$$f^\circ(x^*; d) = \lim_{k\in\mathcal{L}} f^\circ(x^*; d_k)$$

Additionally, one has $\hat{x}_k^{\text{local}} + \sigma_k d_k \in \Omega$ for all $k \in \mathcal{L}$ sufficiently large, this leads to

$$\begin{aligned} f^\circ(x^*; d) \quad &= \quad \lim_{k\in\mathcal{L}} \quad \limsup_{\substack{x \to x^*, x \in \Omega \\ t \downarrow 0, x + td_k \in \Omega}} \quad \frac{f(x + td_k) - f(x)}{t}, \\ &\geq \quad \limsup_{k\in\mathcal{L}} \frac{f(\hat{x}_k^{\text{local}} + \sigma_k d_k) - f(\hat{x}_k^{\text{local}})}{\sigma_k}. \end{aligned} \tag{14}$$

Hence, by substituting (13) into (14), one gets

$$f^\circ(x^*; d) \quad \geq \quad \limsup_{k\in\mathcal{L}} \frac{f(\hat{x}_k^{\text{local}} + \sigma_k d_k) - f(\hat{x}_k^{\text{local}})}{\sigma_k} \geq 0.$$

■

# 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. TREGO is available both in the `R` package `DiceOptim` [2] and python library `trieste` [3].

## 5.1 Testing procedure using the BBOB benchmark

Our experiments are based on the COCO (COmparing Continuous Optimizers, [30]) software. COCO is a recent effort to build a testbed that allows the rigorous comparison of optimizers. We focus here on the noiseless Black-Box Optimization Benchmarking (BBOB) test suite in the *expensive objective function* setting [29] that contains 15 instances of 24 functions [15]; each function is defined for an arbitrary number of parameters to optimize. Each instance corresponds to a randomized modification of the original function by using 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

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

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

4 is often seen as the main target for Bayesian optimization [34]. The full description of the functions is available in Appendix B (see 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 BBOB 2009 [29]. In our experiments, the targets were set at the values reached by best09 after $[0.5, 1, 3, 5, 7, 10, 15, 20] \times n$ 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, the performance of a purely random search are also included to serve as a lower bound.

## 5.2 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) [44, 50]. Our setup aligns with current practices in BO [27, 52], 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 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 $\ell_1$ norm, see (4), to optimize the expected improvement using a multi-start L-BFGS scheme. Each experiment starts with an initial set of $2n + 4$ observations, generated using latin hypercube sampling improved through a maximin criterion [25]. 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 $2n + 1$ 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.

## 5.3 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 trust region contraction factor $\beta$, see (6)?
3. is using a local model beneficial?

| Acronym | Solvers |
|---|---|
| random | random search |
| best09 | best of all BBOB 2009 competitors at each budget [8] |
| TRIKE | TRIKE algorithm of [48] |
| SMAC | SMAC algorithm of [31] |
| DTS-CMA | DTS-CMA algorithm of [9] |
| EGO | original EGO algorithm of [34] |
| TREGO | default TREGO with $\beta = 0.9$, $\gamma = 1/\beta$, $\sigma_0 = \frac{1}{2}(1/5)^{1/n}$, $\rho(\sigma) = \sigma^2$, $d_{\max} = 1$, $d_{\min} = 10^{-6}$ , global/local ratio = 1 / 1 (i.e., $G = 1$ and $L = 1$), with no local GP model |
| gl1-10, gl1-4, gl4-1 and gl10-1 | TREGO with a global/local ratio of 1/10, 1/4, 4/1 and 10/1, respectively |
| smV0 and lgV0 | TREGO with small (i.e., $\sigma_0 = \frac{1}{2}(1/10)^{1/n}$) and large (i.e., $\sigma_0 = \frac{1}{2}(2/5)^{1/n}$) initial trust-region size |
| fstC | TREGO with fast contraction of the trust-region, i.e., $\beta = 0.5$ |
| fstCsmV0 | TREGO with fast contraction of the trust-region and small $\sigma_0$ |
| 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.

4. is there an optimal ratio of 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 $\beta$ is either 0.9 which is classical in DFO algorithms, or 0.5 which corresponds to an aggressive reduction of the trust region. The choice of the initial trust-region $\sigma_0$, within the default TREGO, corresponds to setting the initial trust-region volume to 20% of the search space. In this case, the initial trust-region volume is given by $(2\sigma_0)^n$. We test also as alternatives with a small initial trust-region, i.e., 10% of the search space, and a larger one, i.e., 40% of the search space. The global-local ratio varies from 10-1, which is expected to behave almost similarly to the original EGO, to 1-10, i.e., a very local behavior.

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 ($30n$ 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.

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.

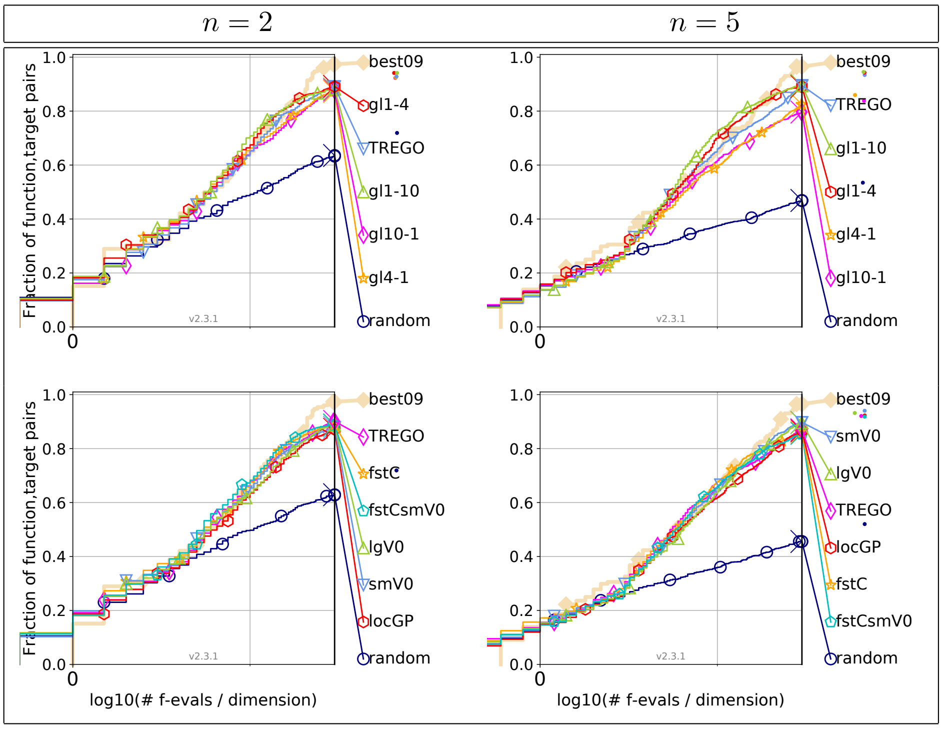

Figure 2: Effect of changing the amount of local and global iterations (top), and changing the other parameters of the TREGO algorithm (bottom). Performance is reported in terms of ERTD, averaged over the entire noiseless BBOB testbed in 2 (left) and 5 (right) dimensions. Run length is 30 × $n$.

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. The main visible results are:

- a slightly positive effect of the local GP on the groups 1 and 2 but a strong negative effect on unimodal functions with bad conditioning, 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 on highly multimodal functions 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.

The gl1-4 variant of TREGO is shown to offer the best trade-off over all groups between performance and robustness. In our comparison with the state-of-the-art BBO algorithms, we will use the name TREGO to refer to the gl1-4 solver.

### 5.4 Comparison with state-of-the-art BBO algorithms

Longer runs of length $50n$ (function evaluations) are made with TREGO in dimensions 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, DTS-CMA, Nomad and MCS. A COCO test campaign of such a set of algorithms up to dimension 10, with run length of $50n$ and 15 repetitions of the optimizations takes of the order of 3 weeks of computing time on a recent single processor.

DTS-CMA [9] 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 [31] 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 black-box optimizers. Nomad [6, 37] is a `C++` solver based on the mesh adaptive direct search method [2]. We have tested Nomad `version 4.2.0` via its provided Python interface where the variable neighborhood search (VNS) strategy was enabled to enhance its global exploration. Nomad enjoys similar convergence properties to those of TREGO, hence a comparison between the two solvers is meaningful. MCS [32] is a multilevel coordinate search solver that balances global and local search that uses quadratic interpolation. MCS is among the best DFO solvers on bound constrained optimization problems [49].

DTS-CMA, SMAC and MCS results are directly extracted from the COCO database. This is not the case of Nomad 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. 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 both trust regions algorithms, i.e., TREGO and TRIKE. This performance gap is limited for $n = 2$ but very visible for $n = 5$ and even higher for $n = 10$. It is also significant for any budget as soon as the shared initialization is done. The improvement is also visible for all function groups, see Figure 4, in particular for groups with strong structure. For the multimodal with weak structure group, the effect is mostly visible for the larger budgets.

**Nomad** has an overall performance comparable to TREGO. Nomad is shown in particular to be very efficient for small budgets but then it gets outperformed by TREGO as the evaluation budget gets larger. The performance gap between Nomad and TREGO is limited for $n = 2$

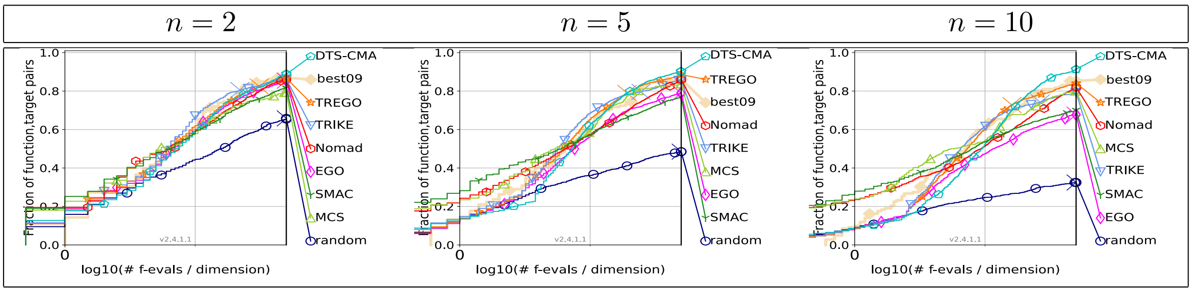

Figure 3: Comparison of TREGO with state-of-the-art optimization algorithms, averaged over the entire COCO testbed in 2, 5 and 10 dimensions. Run length = $50 \times n$.

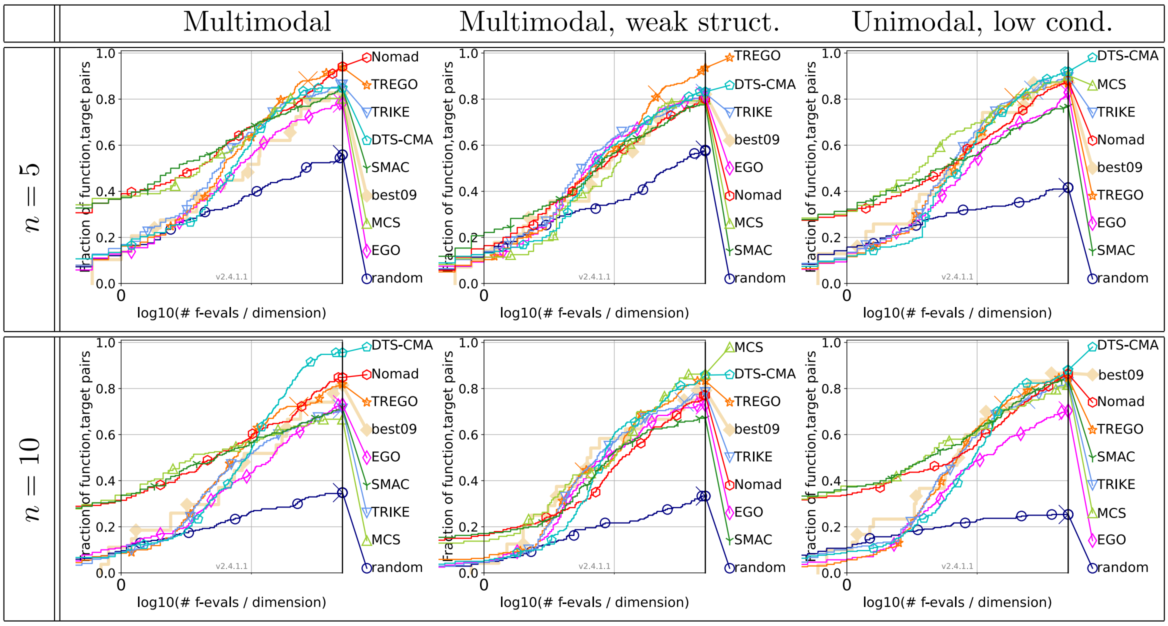

Figure 4: Comparison of TREGO with state-of-the-art optimization algorithms, averaged over the multi-modal functions with adequate (left, f15 to f19) and weak (middle, f20 to f24) global structure, unimodal functions with low conditioning (right), $n = 5$ (top row) and $n = 10$ (bottom row) dimensions.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

but very visible for $n = 5$ and even higher for $n = 10$, see Figure 3. The good start of Nomad can be explained by the fact that it requires only one point to start the optimization process, while Bayesian optimization solvers need a set of points (i.e., the initial DoE) to initiate the optimization process. Thanks to its variable neighborhood search strategy, Nomad seems to outperform most of the tested solvers on the group of multimodal optimization problems, see Figure 4.

**MCS** is outperformed by most of the tested solvers despite its very good performance at the early stages of the optimization process. In fact, the performance of MCS at the beginning seems to deteriorate very fast as the the budget is getting larger, particularly when the regarded optimization problems are multimodal, see Figure 4.

**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, i.e., 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, less than $20 \times n$. However, for large budgets and $n = 10$, DTS-CMA largely outperforms other methods on average. However, looking at Figure 4, DTS-CMA clearly outperforms the other methods, including the best09 baseline, on multimodal functions with strong structure for $n = 10$ and large budgets, while TREGO remains competitive in other cases.

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

**Overall performance** Overall, this benchmark does not reveal a universal winner. SMAC, Nomad and MCS excel with extremely limited budgets, while DTS-CMA outperforms the other methods for the largest dimensions and budgets. TREGO is 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 [42, 24], and narrowing the search space to the vicinity of the current best point naturally solves this issue. Thus, 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, i.e., TRIKE, can perform best in some cases (Figure 6), however our more balanced approach is almost as good, if better (Figure 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 Figure 4, all multimodal functions).

## 6 Conclusions and perspectives

In this work, the TREGO method is introduced: a Bayesian optimization algorithm based on a trust-region mechanism 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. 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. An extensive benchmark is then performed, which allowed us to form the following conclusions:

- TREGO benefits from having a relatively high proportion of local steps, but is otherwise almost 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. Our empirical evaluation focused on bound constrained BBO problems. However, TREGO readily applies to the case of explicit, non-relaxable constraints, which may be studied in the future. Moreover, inspired by e.g. [3, 20, 28] from the DFO community and [45, 51] from the BO one, TREGO can also be naturally extended to handle constraints that are allowed to be violated during the optimization process. Another important future work may include the extension of TREGO to the case of noisy observations, following recent results in DFO [1, 4, 17, 23] and established BO techniques [46].

## Data availability statements

The authors confirm that all data generated or analysed during this study are included in the paper.

## References

[1] S.-K. Anagnostidis, A. Lucchi, and Y. Diouane. Direct-search for a class of stochastic min-max problems. In *International Conference on Artificial Intelligence and Statistics*, pages 3772–3780, 2021.

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

[3] C. Audet and J. E. Dennis Jr. A progressive barrier for derivative-free nonlinear programming. *SIAM J. Optim.*, 20:445–472, 2009.

[4] C. Audet, K. J. Dzahini, M. Kokkolaras, and S. Le Digabel. Stochastic mesh adaptive direct search for blackbox optimization using probabilistic estimates. *Comput. Optim. Appl.*, 19:1–34, 2021.

[5] C. Audet and W. Hare. *Derivative-Free and Blackbox Optimization.* Springer, Cham, Philadelphia, 2017.

[6] C. Audet, S. Le Digabel, V. Rochon Montplaisir, and C. Tribes. Algorithm 1027: NOMAD Version 4: Nonlinear Optimization with the Mads Algorithm. *ACM Trans. Math. Softw.*, 48:1–22, 2022.

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

[8] A. Auger, S. Finck, N. Hansen, and R. Ros. BBOB 2009: Comparison Tables of All Algorithms on All Noiseless Functions. Technical Report RT-0383, INRIA, April 2010.

[9] L. Bajer, Z. Pitra, J. Repický, and M. Holena. Gaussian process surrogate models for the CMA evolution strategy. *Evol. Comput.*, 27:665–697, 2019.

[10] E. Bergou, Y. Diouane, V. Kungurtsev, and C. W. Royer. A stochastic Levenberg-Marquardt method using random models with complexity results. *SIAM-ASA J. Uncertain. Quantif.*, 10:507–536, 2022.

[11] J. Blanchet, C. Cartis, M. Menickelly, and K. Scheinberg. Convergence rate analysis of a stochastic trust region method via supermartingales. *INFORMS J. Optim.*, 1:92–119, 2019.

[12] A. J. Booker, J. E. Dennis Jr., P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset. A rigorous framework for optimization of expensive functions by surrogates. *Struct. Multidiscipl. Optim.*, 17:1–13, 1998.

[13] M. A. Bouhlel, N. Bartoli, R. G. Regis, A. Otsmane, and J. Morlier. Efficient global optimization for high-dimensional constrained problems by using the kriging models combined with the partial least squares method. *Eng. Optim.*, 50:2038–2053, 2018.

[14] E. Brochu, V. M. Cora, and N. De Freitas. 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.

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

[16] A. D. Bull. Convergence rates of efficient global optimization algorithms. *J. Mach. Learn. Res.*, 12:2879–2904, 2011.

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

[18] F. H. Clarke. *Optimization and Nonsmooth Analysis.* John Wiley & Sons, New York, 1983. Reissued by SIAM, Philadelphia, 1990.

[19] A. R. Conn, K. Scheinberg, and L. N. Vicente. *Introduction to Derivative-Free Optimization.* MPS-SIAM Series on Optimization. SIAM, Philadelphia, 2009.

[20] Y. Diouane. A merit function approach for evolution strategies. *EURO J. Comput. Optim.*, 9:100001, 2021.

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

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

[23] Y. Diouane, A. Lucchi, and V. Patil. A globally convergent evolutionary strategy for stochastic constrained optimization with applications to reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pages 3772–3780, 2022.

[24] D. Eriksson, M. Pearce, J. Gardner, R. D. Turner, and M. Poloczek. Scalable global optimization via local Bayesian optimization. In *Advances in Neural Information Processing Systems.*

[25] K.-T. Fang, R. Li, and A. Sudjianto. *Design and modeling for computer experiments.* CRC press, 2005.

[26] A. I. J. Forrester, A. Sóbester, and A. J. Keane. Multi-fidelity optimization via surrogate modelling. *Philos. Trans. A. Math. Phys. Eng. Sci.*, 463:3251–3269, 2007.

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

[28] S. Gratton and L. N. Vicente. A merit function approach for direct search. *SIAM J. Optim.*, 24:1980–1998, 2014.

[29] N. Hansen, A. Auger, R. Ros, S. Finck, and P. Pošík. Comparing results of 31 algorithms from the black-box optimization benchmarking bbob-2009. In *Annual Conference Companion on Genetic and evolutionary computation*, pages 1689–1696, 2010.

[30] N. Hansen, A. Auger, R. Ros, O. Mersmann, T. Tušar, and D. Brockhoff. COCO: a platform for comparing continuous optimizers in a black-box setting. *Optim. Methods Softw.*, 36:114–144, 2021.

[31] F. Hutter, H. H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *International Conference on Learning and Intelligent Optimization*, pages 507–523, 2011.

[32] W. Huyer and A. Neumaier. Global optimization by multilevel coordinate search. *J. Global Optim.*, 14:331–355, 1999.

[33] J. Jahn. *Introduction to the Theory of Nonlinear Optimization.* Springer-Verlag, Berlin, 1996.

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

[35] K. Kandasamy, J. Schneider, and B. Póczos. High dimensional Bayesian optimisation and bandits via additive models. In *International Conference on Machine Learning*, pages 295–304, 2015.

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

[37] S. Le Digabel. Algorithm 909: Nomad: Nonlinear optimization with the mads algorithm. *ACM Trans. Math. Softw.*, 37:44, 2011.

[38] S. Le Digabel and S.M. Wild. A Taxonomy of Constraints in Simulation-Based Optimization. Technical Report G-2015-57, Les cahiers du GERAD, 2015.

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

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

[41] J. Nocedal and S. J. Wright. *Numerical Optimization.* Springer-Verlag, Berlin, second edition, 2006.

[42] Ch. Y. Oh, E. Gavves, and M. Welling. BOCK: Bayesian optimization with cylindrical kernels. In *International Conference on Machine Learning*, pages 3868–3877, 2018.

[43] V. Picheny, P. Casadebaig, R. Trépos, R. Faivre, D. Da Silva, P. Vincourt, and E. Costes. Using numerical plant models and phenotypic correlation space to design achievable ideotypes. *Plant Cell Environ.*, 40:1926–1939, 2017.

[44] V. Picheny and D. Ginsbourger. Noisy Kriging-based optimization methods: a unified implementation within the DiceOptim package. *Comput. Stat. Data Anal.*, 71:1035–1053, 2014.

[45] V. Picheny, R. B. Gramacy, S. Wild, and S. Le Digabel. Bayesian optimization under mixed constraints with a slack-variable augmented lagrangian. In *Advances in Neural Information Processing Systems*, pages 1435–1443, 2016.

[46] V. Picheny, T. Wagner, and D. Ginsbourger. A benchmark of kriging-based infill criteria for noisy optimization. *Struct. Multidiscipl. Optim.*, 48:607–626, 2013.

[47] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning.* MIT press Cambridge, MA, 2006.

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

[49] L. Rios and N. Sahinidis. Derivative-free optimization: a review of algorithms and comparison of software implementations. *J. Global Optim.*, 56:1247–1293, 2013.

[50] O. Roustant, D. Ginsbourger, and Y. Deville. DiceKriging, DiceOptim: Two R Packages for the Analysis of Computer Experiments by Kriging-Based Metamodeling and Optimization. *J. Stat. Softw.*, 51, 2012.

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

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

[53] E. Siivola, A. Vehtari, J. Vanhatalo, J. González, and M. R. Andersen. Correcting boundary overexploration deficiencies in Bayesian optimization with virtual derivative sign observations. In *IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6, 2018.

[54] J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

[55] N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning*, 2010.

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

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

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

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

[60] Z. Wang, F. Hutter, M. Zoghi, D. Matheson, and N. de Feitas. Bayesian optimization in a billion dimensions via random embeddings. *J. Artif. Intell. Res.*, 55:361–387, 2016.

# A Pseudo-code of the TREGO algorithm

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

```
Data: Create an initial DoE $\mathcal{D}_{t_0} = \{x_1, x_2, \ldots, x_{t_0}\}$ of $t_0$ points in a given set $\Omega \subset \mathbb{R}^n$ with a given
      method. Set $\mathcal{Y}_{t_0} = \{f(x_1), f(x_2), \ldots, f(x_{t_0})\}$. 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 $\sigma_0$, $x_0^* \in \mathcal{D}_{t_0}$, choose the
      constants $\beta$, $\gamma$, $d_{\min}$ and $d_{\max}$ such that $0 < \beta < 1 < \gamma$ 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 $\Omega$:                                    */
    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 $x_{k+1}^{\text{global}}$ 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 $\sigma_{k+1} = \gamma\sigma_k$;
    else
        /* A local phase over the trust-region $\Omega_k$:                   */
        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 $\Omega_k$ is the trust-region given by $\Omega_k = \{x \in \Omega \mid d_{\min}\sigma_k \leq \|x - x_k^*\| \leq d_{\max}\sigma_k\}$;
            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 $x_{k+1}^{\text{local}}$ 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 $\sigma_{k+1} = \gamma\sigma_k$ ;
        else
            the local phase and iteration are not successful, set $x_{k+1}^* = x_k^*$, and $\sigma_{k+1} = \beta\sigma_k$;
        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 | $10^n$ local minima, spherical global structure |
| f4 | Büche-Rastrigin | $10^n$ 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 $\approx 100$, made of many plateaus |
| f8 | Original Rosenbrock | good points form a curved $n-1$ 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 $\approx 1000$ |
| f19 | Composite Griewank-Rosenbrock | highly multimodal version of Rosenbrock |
| multimodal functions with weak global structure | | |
| f20 | Schwefel | $2^n$ 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 $\approx 30$ |
| f22 | Gallagher's Gaussian 21-hi peaks | 21 optima with random positions and heights, conditioning $\approx 1000$ |
| f23 | Katsuura | highly rugged and repetitive function with more than $10^n$ 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

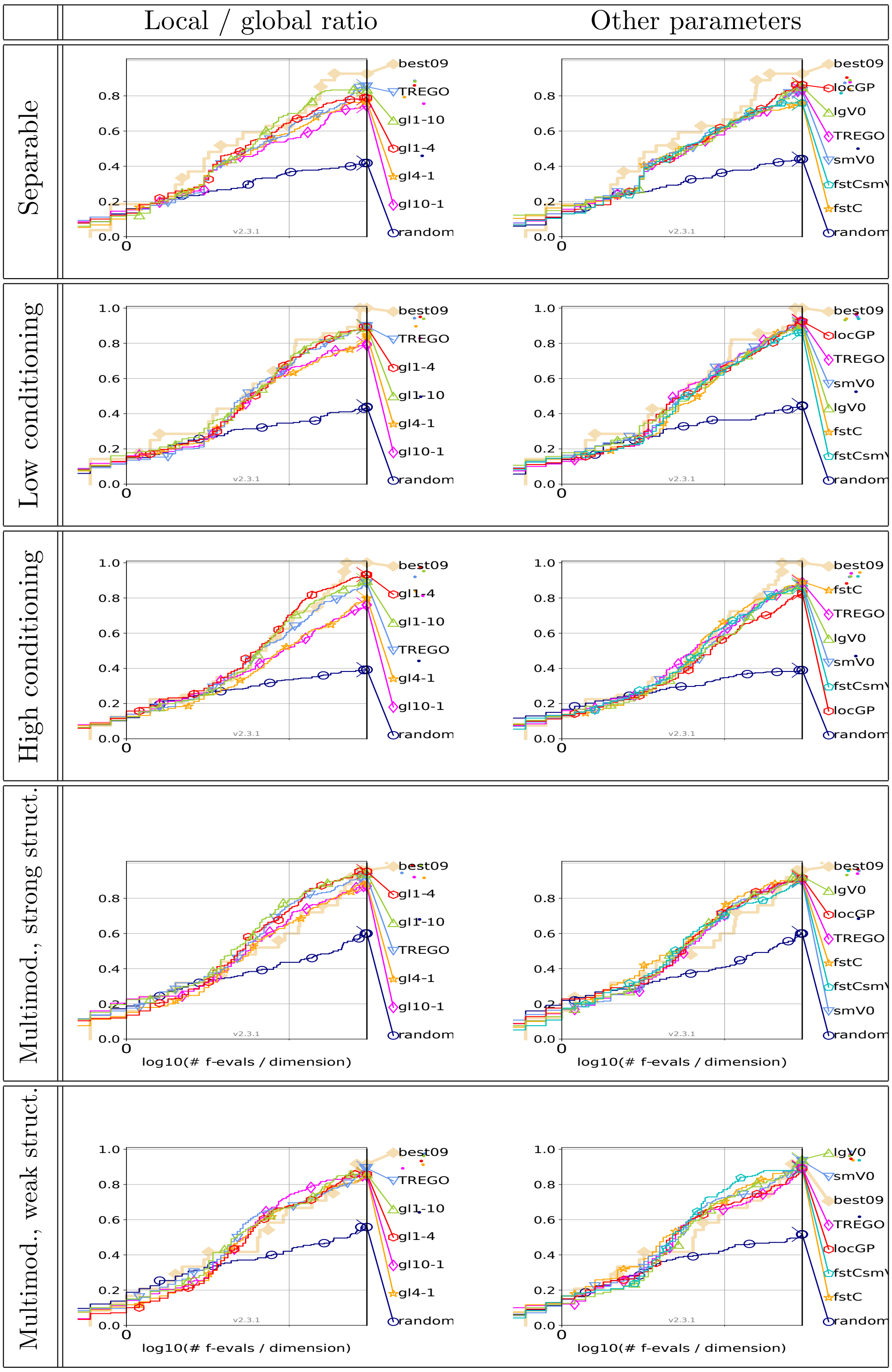

Figure 5: Effect of changing parameters of the TREGO algorithm, averaged by function groups for $n = 5$. Run length is $30 \times n$.

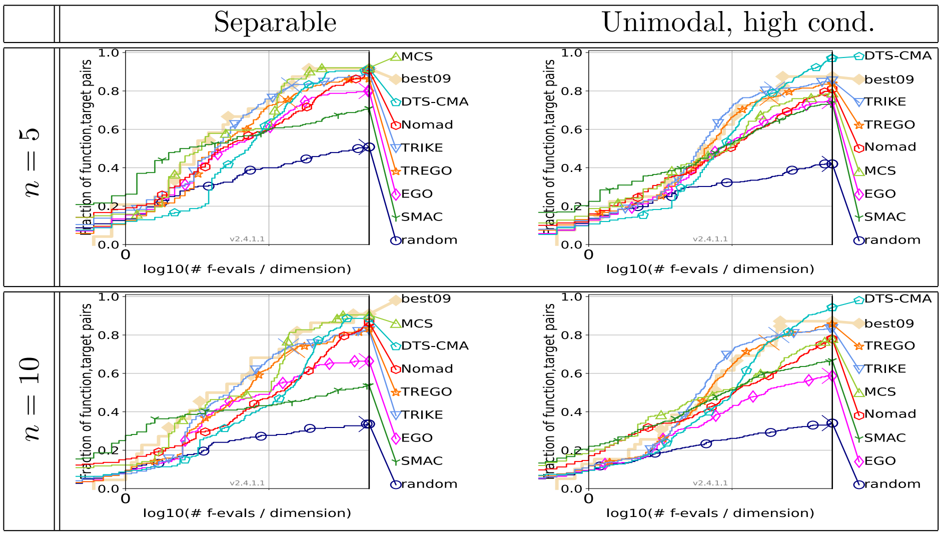

Figure 6: Comparison of TREGO with state-of-the-art optimization algorithms on separable (left) and unimodal with high conditioning functions (right), for $n = 5$ (top) and $n = 10$ (bottom). Run length $= 50 \times n$.