# Presentation of the bbob-constrained Test Suite

Paul Dufossé          Who else ?

February 17, 2022

## 1 Introduction

This document briefly describes the newly introduced bbob-constrained test suite implemented in the COCO software. It contains everything one needs to know if planning to benchmark an algorithm on the constrained test suite.

## 2 Problem definition

A constrained test problem is written in the standard form

$$\underset{x \in \mathcal{X}}{\text{minimize}} \quad f(x) \qquad \text{subject to} \quad g_k(x) \leq 0 \quad \text{for } k = 1, \ldots, m \tag{1}$$

where $\mathcal{X} = [-5, 5]^n$, $n$ is the dimension of the search space and $m$ is the number of generally non-linear constraints.

The construction of any problem of the test suite is detailed in Dufossé et al. (2022) and relies on Theorem 4.2.16 in (Bazaraa et al., 2006, p. 195) stating that, if $f$ is *pseudo-convex* at $y^\star$, and $g_k$ is differentiable and *quasi-convex* at $y^\star$ for all $k = 1, \ldots, m$, then the KKT conditions are sufficient to ensure that $y^\star$ is a global minimum of the constrained optimization problem. It is easy to show that, if $f$ is *strictly* pseudo-convex at $y^\star$, then the global minimum is also unique Dufossé et al. (2022). We refer to the textbook of Bazaraa et al. (2006) for the definitions and properties of pseudo- and quasi-convexity.

### 2.1 Principles of the construction

The theory allows to control the location of the optimum so it is possible to chose $y^\star$ such that the gradient is non-zero. In the construction, $f$ is an already shifted BBOB function, hence taking $y^\star = 0$ generally provides $\nabla f(y^\star) \neq 0$. From this point, following the gradient direction, an initial solution is determined

$$x^{\text{init}} = y^\star + \rho \nabla f(y^\star) + x^{\text{opt}} \tag{2}$$

where $x^{\text{opt}} \in \mathcal{X}$ defines another (random) translation such that the global minimum of the constrained problem is not in $y^\star$ and $\rho > 0$ is a scaling factor ensuring that $x^{\text{init}} \in \mathcal{X}$.

## 2.2 Construction of the feasible region

The algorithm to generate an instance of a constrained problem first builds a set of $m$ *linear* constraints, of the form $g_k(x) = \alpha_k a_k^\top (x - y^\star) + \beta_k$ by setting $a_k \in \mathbb{R}^n$ where $\alpha_k > 0, \beta_k \geq 0 \in \mathbb{R}$ are input parameters. The first constraint is chosen such that $a_1 = -\alpha_1 \nabla f(y^\star)/\|\nabla f(y^\star)\|$ and $\beta_1 = 0$ hence $g_1(y^\star) = 0$. If $m = 1$, this is the only choice to satisfy the stationarity condition in **??**. The remaining constraints are randomly sampled from an isotropic multivariate normal distribution. If $g_k(y^\star + \rho \nabla f(y^\star)) > 0$, the sign of $a_k$ is flipped to make sure that $y^\star + \rho \nabla f(y^\star)$ is feasible. A constraint is inactive at the constructed optimum if $\beta_k > 0$.

The Lagrange multipliers associated to $y^\star$ when the constraints are generated according to the aforementioned procedure are $[1, 0, \ldots, 0] \in \mathbb{R}^m$. A second step of the algorithm modifies the set of constraints gradients in a randomized manner such that the first constraint is not any more aligned with the gradient direction at the optimum. The constraints' indices are also shuffled.

## 2.3 Non-linear transformations

Let $T : \mathbb{R}^n \to \mathbb{R}^n$ be a bijective transformation of the search space with inverse $T^{-1}$. Then $T^{-1}(y^\star)$ is the global optimum of the problem

$$\underset{x \in \mathcal{X}}{\text{minimize}} \quad f(T(x)) \qquad \text{subject to} \quad g_i(T(x)) \leq 0 \quad i = 1, \ldots, m \tag{3}$$

if $y^\star$ is the global optimum of the original problem as described in Equation (1). This allows to apply non-linear transformations of the search space to the set of constructed linear constraints. For practical reasons, we consider in this work bijective transformations $T$ that have $y^\star$ as fixed point, i.e. $T(y^\star) = y^\star$. As a consequence, the optimum of the transformed problem in Equation (3) remains $y^\star$. This way, we avoid computing the inverse transformation at $y^\star$. The non-linear transformations implemented in the test suite are coordinate-wise, defined as $T^\beta_{\text{asy}}$, where $\beta > 0$, and $T_{\text{osz}}$, defined as follows:

$$T^\beta_{\text{asy}}(x) = \left( \begin{cases} x_i^{1+\beta \frac{i-1}{n-1}\sqrt{x_i}} & \text{if } x_i > 0 \\ x_i & \text{otherwise} \end{cases} \right)_{i=1,\ldots,n} , \tag{4}$$

$$T_{\text{osz}}(x) = \big(\text{sign}(x_i) \exp(\hat{x}_i + 0.049(\sin(c_1 \hat{x}_i) + \sin(c_2 \hat{x}_i)))\big)_{i=1,\ldots,n} , \tag{5}$$

with $\hat{x}_i = \log(|x_i|)\mathbb{1}_{\{x_i \neq 0\}}$ and $\text{sign}(x_i)$ is the sign function with convention $\text{sign}(0) = 0$. The value for the constants $c_1$ and $c_2$ are $c_1 = \begin{cases} 10 & \text{if } x_i > 0 \\ 5.5 & \text{otherwise} \end{cases}$, and $c_2 = \begin{cases} 7.9 & \text{if } x_i > 0 \\ 3.1 & \text{otherwise} \end{cases}$. The idea is that $T^\beta_{\text{asy}}$ introduces asymmetry, as it only changes positive coordinates, and $T_{\text{osz}} : \mathbb{R}^n \to \mathbb{R}^n$ oscillates around the origin. Both transformations are bijective, continuous, sign-preserving hence have fixed point 0.

## 2.4 Objective functions

The bbob-constrained test suite brings together the aforementioned methodology with several existing COCO "raw" functions already used in the unconstrained setting. They are described in Table 1.

To create the constrained problems, each one of the 9 objective functions may be composed with a non-linear transformation of the search space defined previously and is combined with $m' \in \{1, 2, 6, 6 + \lfloor n/2 \rfloor, 6 + n, 6 + 3n\}$ active constraints and $\lfloor m'/2 \rfloor$ inactive constraints, having overall $m = \lfloor 3m'/2 \rfloor$ constraints. This results in 54 constrained test problems, $f_i$, $i = 1, \ldots, 54$, summarized in Table 2. For practical considerations, the constrained problem obtained is shifted by a vector $x^{\text{opt}} \neq 0$ as mentioned in Section 2.1. In practice, $x^{\text{opt}}$ is sampled randomly to define different instances of the same constrained problem. Additionally, the objective functions are scaled down by a factor $c_{\text{scal}} > 0$ which is specific to each function. The final optimization problem writes:

$$\underset{x \in \mathcal{X}}{\text{minimize}} \quad c_{\text{scal}} f(v) \qquad \text{subject to} \quad g_i(v) \leq 0 \quad \text{for } i = 1, \ldots, m \tag{6}$$

where $v = T(x - x^{\text{opt}})$ and $T$ is the non-linear transformation reported in Table 2 along with $c_{\text{scal}}$ and the number of constraints associated to each problem. The considered dimensions are the same as for the bbob test suite, that is $n \in \{2, 3, 5, 10, 20, 40\}$. The first instance of each of the two-dimensional test problems is illustrated in Figures 1 to 4. Other instances in 2-D can be visualized at this link.

# 3 Performance Assessment

The target definition encompasses objective minimization and constraint satisfaction together using the merit function

$$\tilde{f}(x) := \max(f_{\text{opt}}, f(x)) + \sum_{k=0}^{m} \max(0, g_k(x)) \ , \tag{7}$$

where $f_{\text{opt}} = f(x^{\text{opt}})$. It is easy to see that $\min \tilde{f} = f_{\text{opt}}$.

The estimated Expected Runtime (ERT) Hansen et al. (2016) is computed from feasible points only and for 7 target values $f_{\text{target}} = f_{\text{opt}} + 10^i$, with $i \in \{1, 0, -1, -2, -3, -5, -6\}$.

Data points for the ECDFS are collected for 41 target values $f_{\text{target}} = f_{\text{opt}} + 10^i$, with $i$ evenly distributed in $[-6, 2]$.

# 4 Running an experiment

The interface yields the feasible starting point $x^{\text{init}}$ under problem.initial_solution. A code snippet of how to run a Python solver on a COCO constrained test problem is listed as Section 4. Running a full experiment requires, among other

| Function name | Formulation | Transformations |
|---|---|---|
| Sphere | $f_{\text{sphere}}(x) = z^\top z + f_{\text{uopt}}$ | $z = x - x^{\text{uopt}}$ |
| Separable ellipsoid | $f_{\text{ellipsoid}}(x) = \sum_{i=1}^{n} 10^{6\frac{i-1}{n-1}} z_i^2 + f_{\text{uopt}}$ | $z = x - x^{\text{uopt}}$ |
| Linear slope | $f_{\text{linear}}(x) = \sum_{i=1}^{n} 5\|s_i\| - s_i z_i + f_{\text{uopt}}$ | $s_i = \text{sign}(x_i^{\text{uopt}})10^{\frac{i-1}{n-1}}$ $z_i = \begin{cases} x_i & \text{if } x_i^{\text{uopt}} x_i < 5^2 \\ x_i^{\text{uopt}} & \text{otherwise} \end{cases}$ for $i = 1, \ldots, n$ |
| Rotated ellipsoid | $f_{\text{elli\_rot}}(x) = \sum_{i=1}^{n} 10^{6\frac{i-1}{n-1}} z_i^2 + f_{\text{uopt}}$ | $z = R(x - x^{\text{uopt}})$ |
| Discus | $f_{\text{discus}}(x) = 10^6 z_1^2 + \sum_{i=2}^{n} z_i^2 + f_{\text{uopt}}$ | $z = R(x - x^{\text{uopt}})$ |
| Bent cigar | $f_{\text{bent\_cigar}}(x) = z_1^2 + 10^6 \sum_{i=2}^{n} z_i^2 + f_{\text{uopt}}$ | $z = R(x - x^{\text{uopt}})$ |
| Different powers | $f_{\text{diff\_powers}}(x) = \sqrt{10^6 \sum_{i=1}^{n} \|z_i\|^{2+4\frac{i-1}{n-1}}} + f_{\text{uopt}}$ | $z = R(x - x^{\text{uopt}})$ |
| Rastrigin | $f_{\text{rastrigin}}(x) = 10\left(n - \sum_{i=1}^{n} \cos(2\pi z_i)\right) + z^\top z + f_{\text{uopt}}$ | $z = x - x^{\text{uopt}}$ |
| Rotated Rastrigin | $f_{\text{rast\_rot}}(x) = 10\left(n - \sum_{i=1}^{n} \cos(2\pi z_i)\right) + z^\top z + f_{\text{uopt}}$ | $z = R(x - x^{\text{uopt}})$ |

Table 1: COCO raw functions definition: formulation and search space transformations, $x^{\text{uopt}} \in \mathbb{R}^n$ is a randomly sampled vector locating the unconstrained minimum of the objective function such that $f(x^{\text{uopt}}) = f_{\text{uopt}} \in \mathbb{R}$. Considering rotations, $R \in \mathbb{R}^{n\times n}$ is a randomly sampled orthogonal matrix.

| Number of constraints | | | 1 | 3 | 9 | $9 + \lfloor 3n/4 \rfloor$ | $9 + \lfloor 3n/2 \rfloor$ | $9 + \lfloor 9n/2 \rfloor$ |
|---|---|---|---|---|---|---|---|---|
| Number of active constraints | | | 1 | 2 | 6 | $6 + \lfloor n/2 \rfloor$ | $6 + n$ | $6 + 3n$ |
| Objective | $T$ | $c_{\text{scal}}$ | Function IDs | | | | | |
| $f_{\text{sphere}}$ | id | 10 | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ |
| $f_{\text{ellipsoid}}$ | $T_{\text{osz}}$ | $10^{-4}$ | $f_7$ | $f_8$ | $f_9$ | $f_{10}$ | $f_{11}$ | $f_{12}$ |
| $f_{\text{linear}}$ | id | 10 | $f_{13}$ | $f_{14}$ | $f_{15}$ | $f_{16}$ | $f_{17}$ | $f_{18}$ |
| $f_{\text{elli\_rot}}$ | $T_{\text{osz}}$ | $10^{-4}$ | $f_{19}$ | $f_{20}$ | $f_{21}$ | $f_{22}$ | $f_{23}$ | $f_{24}$ |
| $f_{\text{discus}}$ | $T_{\text{osz}}$ | $10^{-4}$ | $f_{25}$ | $f_{26}$ | $f_{27}$ | $f_{28}$ | $f_{29}$ | $f_{30}$ |
| $f_{\text{bent\_cigar}}$ | $T_{\text{asy}}^{0.5}$ | $10^{-4}$ | $f_{31}$ | $f_{32}$ | $f_{33}$ | $f_{34}$ | $f_{35}$ | $f_{36}$ |
| $f_{\text{diff\_powers}}$ | id | $10^{-2}$ | $f_{37}$ | $f_{38}$ | $f_{39}$ | $f_{40}$ | $f_{41}$ | $f_{42}$ |
| $f_{\text{rastrigin}}$ | $T_{\text{asy}}^{0.2} \circ T_{\text{osz}}$ | 10 | $f_{43}$ | $f_{44}$ | $f_{45}$ | $f_{46}$ | $f_{47}$ | $f_{48}$ |
| $f_{\text{rast\_rot}}$ | $T_{\text{asy}}^{0.2} \circ T_{\text{osz}}$ | 10 | $f_{49}$ | $f_{50}$ | $f_{51}$ | $f_{52}$ | $f_{53}$ | $f_{54}$ |

Table 2: Identifiers of the bbob-constrained problems. The outer row indicates the number of total and active constraints while the outer column indicates the objective functions and the second left column the non-linear transformations from section 2.3 attached to each objective, where $\text{id}(x) = x$.

things, to attach an observer to the current problem, control the budget (number of objective and constraints evaluations) and loop over all problems of the test suite. If the experimental design includes restarts, another feasible starting can be sampled from $x^{\text{init}}$ following the procedure described in Algorithm 1. An exhaustive example of how to run a full experiment can be found in the COCO repository[1].

**Algorithm 1** Sampling a new feasible starting point

```
Require: $x^{\text{init}}$ and $\sigma > 0$
1: while True do
2:     Sample $z \sim \mathcal{N}(0, \text{id})$ and set $x = x^{\text{init}} + \sigma z$
3:     if $x \notin \mathcal{X}$ or any $g_k(x) \geq 0$ for $k = 1, \ldots, m$ then
4:         $\sigma \leftarrow \sigma/2$
5:     else
6:         return $x$ as the new starting point
```

```
import cocoex
from scipy.optimize import fmin_cobyla

def constraint(x):
```

[1]See coco/code-experiments/build/python/example_experiment2.py after installation of COCO

```
    #SciPy requires g(x) >= 0
    return - problem.constraint(x)

suite = cocoex.Suite('bbob-constrained', '', '')
problem = suite.next_problem()
x_start = problem.initial_solution
fmin_cobyla(
  problem, x_start, constraint,
  # solver parameters
  rhobeg=2, rhoend=1e-8, maxfun=1e4)
```

Listing 1: A simple run of SciPy's COBYLA on a COCO constrained test problem.

## References

Mokhtar S. Bazaraa, C. M. Shetty, and Hanif D. Sherali. 2006. *Nonlinear Programming.* Wiley. 871 pages.

Paul Dufossé, Nikolaus Hansen, Dimo Brockhoff, Phillipe R. Sampaio, Asma Atamna, Nguyen, and Anne Auger. 2022. Building Scalable Test Problems for Benchmarking Constrained Optimizers. *soon submitted to SIAM Journal of Optimization* (2022).

Nikolaus Hansen, Anne Auger, Dimo Brockhoff, Dejan Tušar, and Tea Tušar. 2016. COCO: Performance Assessment. *CoRR* (2016). arXiv:1605.03560

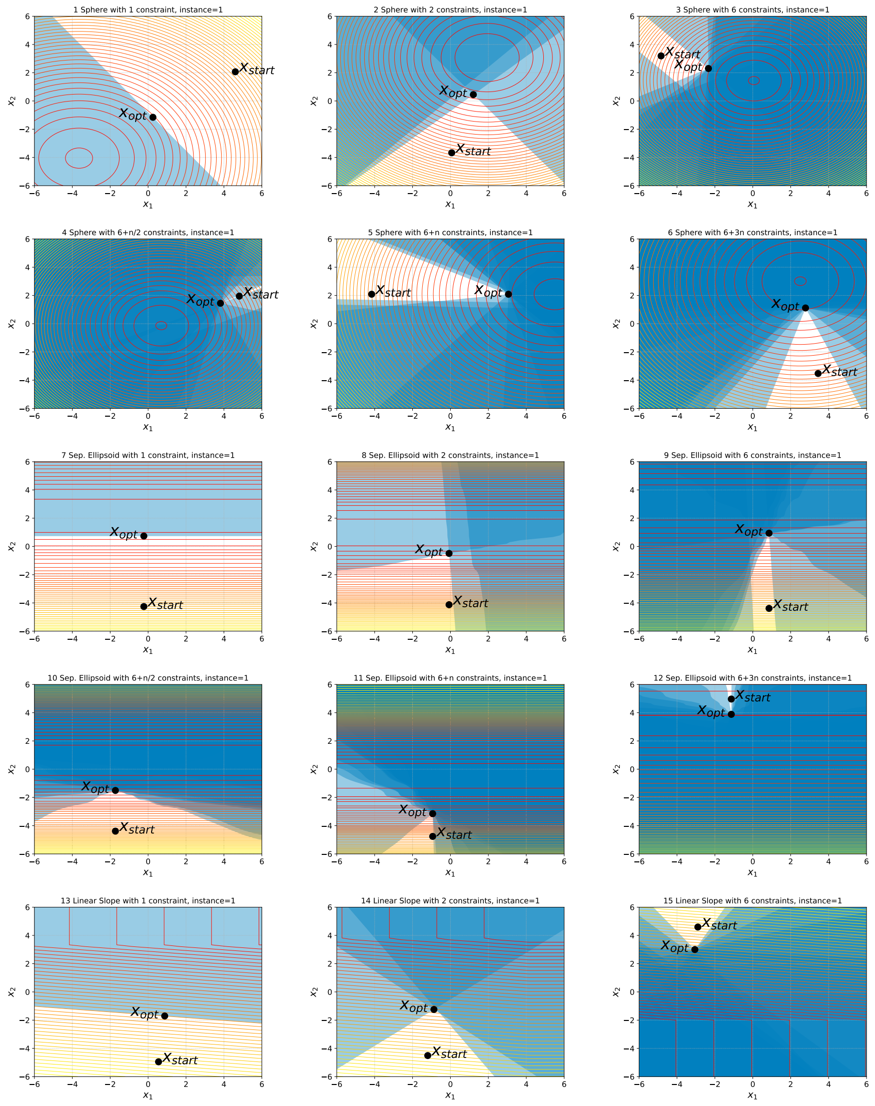

Figure 1: Contour plots of the first instance of `bbob-constrained` problems $f_1$ to $f_{15}$ in 2-D.

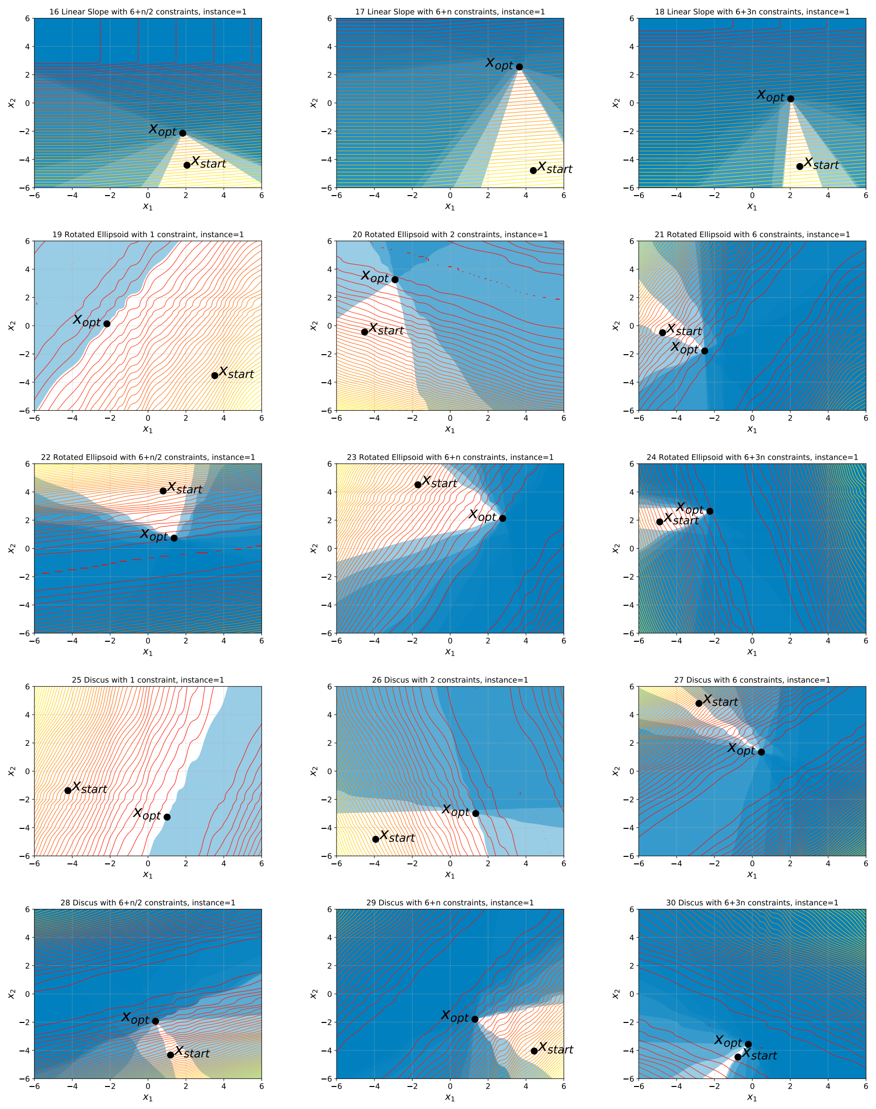

Figure 2: Contour plots of the first instance of bbob-constrained problems $f_{16}$ to $f_{30}$ in 2-D.

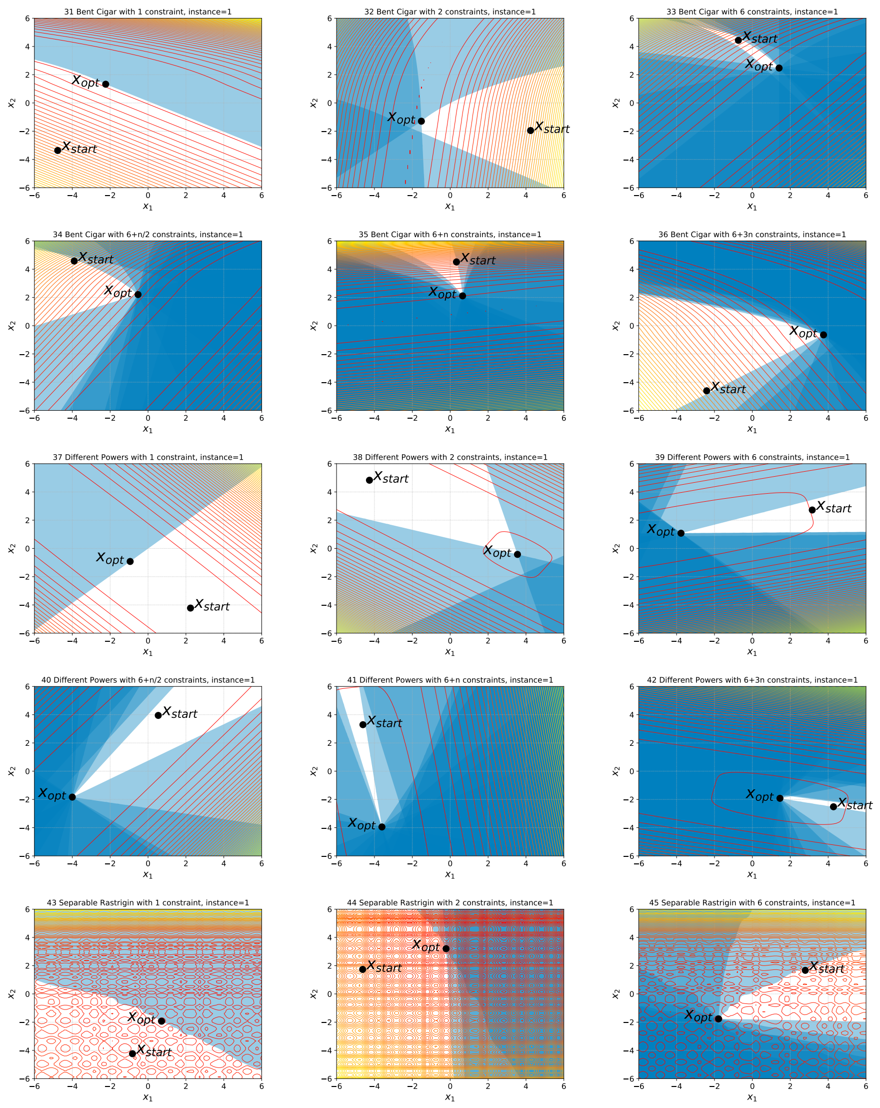

Figure 3: Contour plots of the first instance of bbob-constrained problems $f_{31}$ to $f_{45}$ in 2-D.

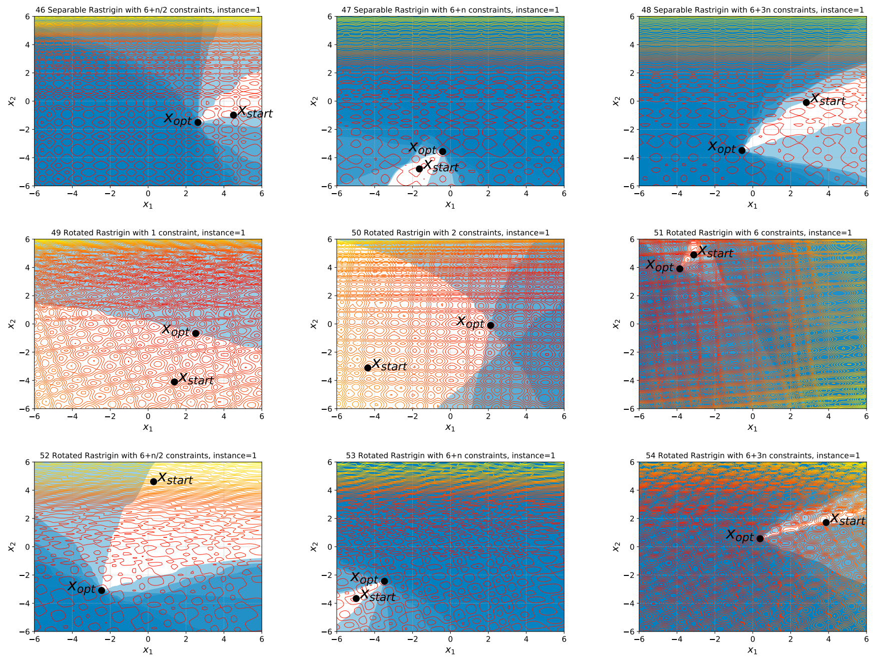

Figure 4: Contour plots of the first instance of bbob-constrained problems $f_{46}$ to $f_{54}$ in 2-D.