

Radial basis functions for constrained numerical global optimization

including the challenges of equality constraints

Wolfgang Konen

Cologne University of Applied Sciences (TH Köln)
Germany

June 2025

Work was done in cooperation with
Samineh Bagheri (formerly TH Köln, Germany),
Thomas Bäck (Leiden University, The Netherlands)

- 1 Introduction
- 2 Radial Basis Function Interpolation
- 3 SACOBRA
- 4 Equality Constraints
- 5 Results
- 6 Summary and Outlook
- 7 Appendix

Section 1

Introduction

Real-World Optimization Problems

$$\begin{array}{ll} \text{Min} & f(\vec{x}), \quad \vec{x} \in [\vec{a}, \vec{b}] \subset \mathbb{R}^d \\ \text{subject to} & g_i(\vec{x}) \leq 0, \quad i = 1, 2, \dots, m \\ & h_j(\vec{x}) = 0, \quad j = 1, 2, \dots, r \end{array}$$

COP:

Constraint Optimization Problem

Optimization problems in real world are often

- Non-linear

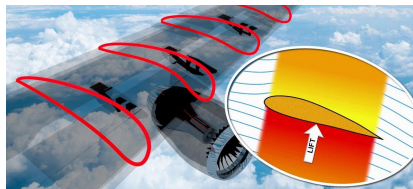
Real-World Optimization Problems

$$\begin{aligned} \text{Min} \quad & f(\vec{x}), \quad \vec{x} \in [\vec{a}, \vec{b}] \subset \mathbb{R}^d \\ \text{subject to} \quad & g_i(\vec{x}) \leq 0, \quad i = 1, 2, \dots, m \\ & h_j(\vec{x}) = 0, \quad j = 1, 2, \dots, r \end{aligned}$$

Optimization problems in real world are often

- Non-linear
- Multi-constrained
 - *equality* constraints
 - *inequality* constraints

COP:
Constraint Optimization Problem



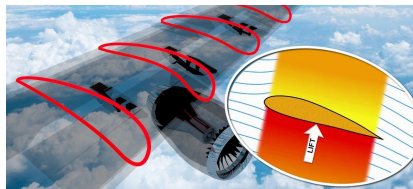
Real-World Optimization Problems

$$\begin{aligned} \text{Min} \quad & f(\vec{x}), \quad \vec{x} \in [\vec{a}, \vec{b}] \subset \mathbb{R}^d \\ \text{subject to} \quad & g_i(\vec{x}) \leq 0, \quad i = 1, 2, \dots, m \\ & h_j(\vec{x}) = 0, \quad j = 1, 2, \dots, r \end{aligned}$$

Optimization problems in real world are often

- Non-linear
- Multi-constrained
 - *equality* constraints
 - *inequality* constraints
- Blackbox

COP:
Constraint Optimization Problem



Real-World Optimization Problems

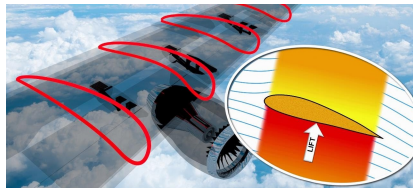
$$\begin{aligned} \text{Min} \quad & f(\vec{x}), \quad \vec{x} \in [\vec{a}, \vec{b}] \subset \mathbb{R}^d \\ \text{subject to} \quad & g_i(\vec{x}) \leq 0, \quad i = 1, 2, \dots, m \\ & h_j(\vec{x}) = 0, \quad j = 1, 2, \dots, r \end{aligned}$$

Optimization problems in real world are often

- Non-linear
- Multi-constrained
 - *equality* constraints
 - *inequality* constraints
- **Blackbox**
- Expensive to evaluate

COP:

Constraint Optimization Problem



Real-World Optimization Problems

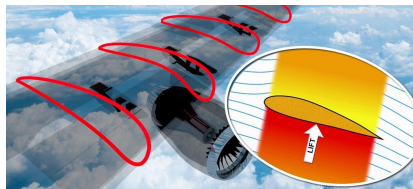
$$\begin{aligned} \text{Min} \quad & f(\vec{x}), \quad \vec{x} \in [\vec{a}, \vec{b}] \subset \mathbb{R}^d \\ \text{subject to} \quad & g_i(\vec{x}) \leq 0, \quad i = 1, 2, \dots, m \\ & h_j(\vec{x}) = 0, \quad j = 1, 2, \dots, r \end{aligned}$$

Optimization problems in real world are often

- Non-linear
- Multi-constrained
 - *equality* constraints
 - *inequality* constraints
- **Blackbox**
- Expensive to evaluate

COP:

Constraint Optimization Problem



Real-World Optimization Problems

$$\begin{aligned} \text{Min} \quad & f(\vec{x}), \quad \vec{x} \in [\vec{a}, \vec{b}] \subset \mathbb{R}^d \\ \text{subject to} \quad & g_i(\vec{x}) \leq 0, \quad i = 1, 2, \dots, m \\ & h_j(\vec{x}) = 0, \quad j = 1, 2, \dots, r \end{aligned}$$

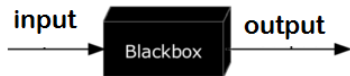
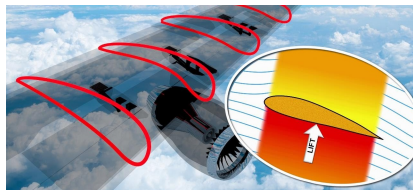
Optimization problems in real world are often

- Non-linear
- Multi-constrained
 - *equality* constraints
 - *inequality* constraints
- **Blackbox**
- Expensive to evaluate

Surrogate-assisted optimization algorithms aim to replace the black-box functions with mathematical models. This saves some expensive evaluations of the real functions

COP:

Constraint Optimization Problem

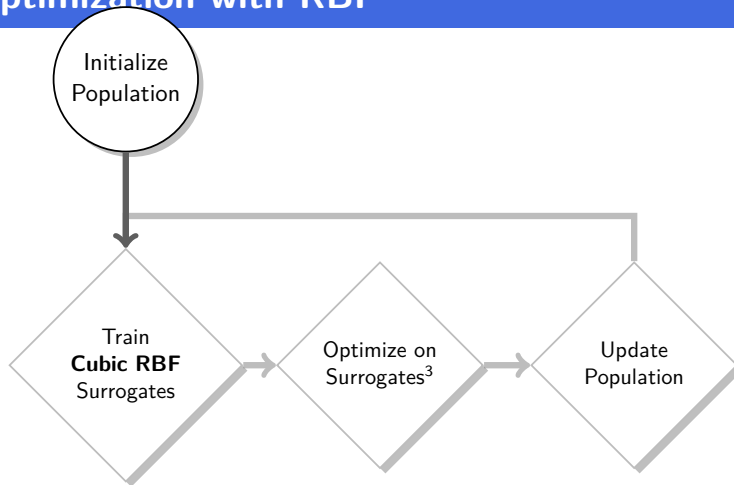


- COBRA¹ appeared to be an efficient solver
- SACOBRA², a modified version of COBRA in R, showed competitive results
- These techniques use **Radial Basis Functions (RBFs)** to model objective and constraint function(s)
- They solve an optimization problem on the surrogates
- Training **RBFs** is fast, even in a high dimensional space

¹R. G. Regis, “Constrained Optimization By Radial basis function Approximation for high-dimensional expensive black-box problems with infeasible initial points,” Engineering Optimization, 2013.

²S. Bagheri, W. Konen, M. Emmerich, and T. Bäck, “Self-adjusting parameter control for surrogate-assisted constrained optimization under limited budgets”, Applied Soft Computing 61 (2017): 377-393

SACOBRA: Self-Adjusting Constrained Black-Box Optimization with RBF



³In this step a new solution (infill point) is generated

Section 2

Radial Basis Function Interpolation

RBF Formulation

- Basic RBF: $s(x)$ models target function $f(x)$

$$s(x) = \sum_{j=1}^n \lambda_j \phi(\|x - x_j\|) \quad (1)$$

$$[\Phi][\lambda] = [f], \quad (2)$$

where $\Phi_{i,j} = \phi(\|x_i - x_j\|)$ and $f_i = f(x_i)$

- Augmented RBF

$$s(x) = \sum_{j=1}^n \lambda_j \phi(\|x - x_j\|) + \sum_{k=1}^M \gamma_k p_k(x) \quad (3)$$

$$\begin{pmatrix} \Phi & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ \gamma \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}, \quad (4)$$

where P is a matrix with entries $p_k(x_j)$ (**polynomial tail**)

Different types of Basis Function ϕ

Table: Commonly used radial basis functions

Type of basis function	$\phi(r)$
Parameter-free RBF	
Cubic	r^3
Thin plate spline	$r^2 \log r$
RBF with width parameter	
Gaussian	$e^{-(\frac{r}{w})^2}$
Multiquadric (MQ)	$\sqrt{1 + (\frac{r}{w})^2}$

- RBF interpolation was first invented by Hardy in 1971 for cartography problems.
- Many different radial basis functions can be used.
- RBF models are a really effective technique to model extrema
- They work efficiently in high dimensions
- They are easy to implement
- It is cheap to train them

Section 3

SACOBRA

- Earlier optimizers on COPs needed careful parameter adjustment for each problem
- SACOBRA replaces this by self-adjustment

- Earlier optimizers on COPs needed careful parameter adjustment for each problem
- **S**ACOBRA replaces this by **self**-**a**adjustment
 - As an example we show here the adjustment to large output ranges

- Earlier optimizers on COPs needed careful parameter adjustment for each problem
- **S**ACOBRA replaces this by **self**-**a**adjustment
 - As an example we show here the adjustment to large output ranges

Large Output Ranges (1)

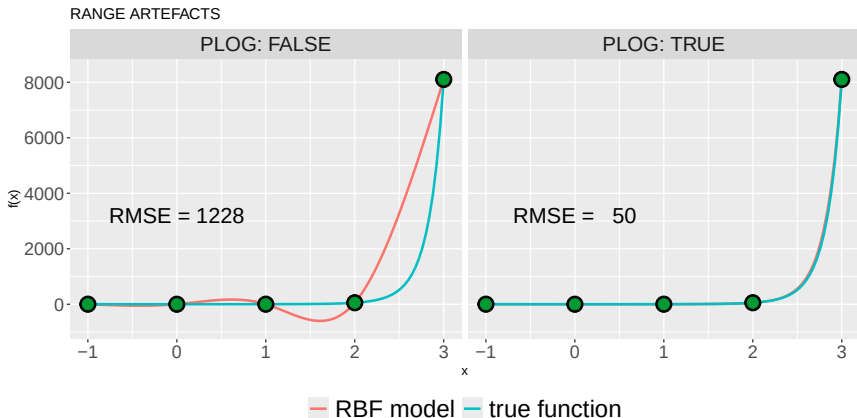


Figure: The influence of large output ranges. Left: Fitting the original function with a cubic RBF model. Right: The solution (see below).

Large Output Ranges (2)

Example problem:

A simple function with a large range:

$$f(x) = e^{x^2} \quad (5)$$

Analysis:

- RBF model using the green sample points in Fig. 1 results in oscillating behaviour.
- This is because the RBF model avoids large slopes.
- The RBF model fit is similar to a spline function.

Large Output Ranges (2)

Example problem:

A simple function with a large range:

$$f(x) = e^{x^2} \quad (5)$$

Analysis:

- RBF model using the green sample points in Fig. 1 results in oscillating behaviour.
- This is because the RBF model avoids large slopes.
- The RBF model fit is similar to a spline function.

Regis and Shoemaker (2012) defined the function

$$plog(f) = \begin{cases} +\ln(1+f) & \text{if } f \geq 0 \\ -\ln(1-f) & \text{if } f < 0 \end{cases} \quad (6)$$

We use this for a better modeling scheme:

- Transform $f(x)$ with $plog$ to smaller output range and smaller slopes.
- Make RBF model fit for $plog(f(x))$.
- Then back-transform with $plog^{-1}$, this takes care of the large slopes.

Advantage

Good modeling of large slopes

Disadvantage

Simple linear functions become nonlinear $\rightarrow$ harder to approximate

Regis and Shoemaker (2012) defined the function

$$plog(f) = \begin{cases} +\ln(1+f) & \text{if } f \geq 0 \\ -\ln(1-f) & \text{if } f < 0 \end{cases} \quad (6)$$

We use this for a better modeling scheme:

- Transform $f(x)$ with $plog$ to smaller output range and smaller slopes.
- Make RBF model fit for $plog(f(x))$.
- Then back-transform with $plog^{-1}$, this takes care of the large slopes.

Advantage

Good modeling of large slopes

Disadvantage

Simple linear functions become nonlinear $\rightarrow$ harder to approximate

Self-Adjustment of PLOG

When to apply PLOG?

SACOBRA measures the output range of the objective function

- It initializes PLOG accordingly
- And it controls PLOG online (with every iteration)

SACOBRA has a number of such adaptive mechanisms.

Self-Adjustment of PLOG

When to apply PLOG?

SACOBRA measures the output range of the objective function

- It initializes PLOG accordingly
- And it controls PLOG online (with every iteration)

SACOBRA has a number of such adaptive mechanisms.

Section 4

Equality Constraints

Problems with Equality Constraints

$$\begin{array}{ll}\text{Minimize} & f(\vec{x}), \\ \text{subject to} & g_i(\vec{x}) \leq 0, \quad i = 1, \dots, m \\ & h_j(\vec{x}) = 0, \quad j = 1, \dots, r\end{array}$$

- Initially, SACOBRA was not able to handle equality constraints
- Problem: COPs with equality constraints have **zero feasible volume**
- Equality-to-inequality transformation:

$$h_j(\vec{x}) = 0 \Rightarrow h_j(\vec{x}) \leq 0 \quad \wedge \quad h_j(\vec{x}) \geq 0$$

still zero feasible volume

- Manually choosing a feasible side

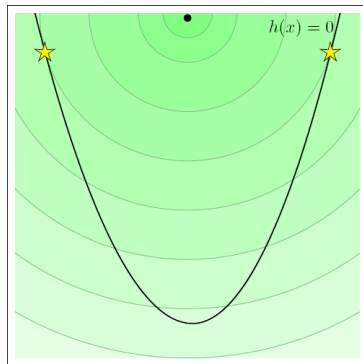
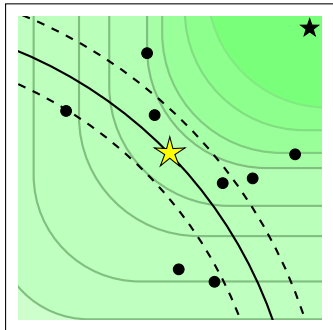


Figure: Example problem with equality constraint. Green shaded contours: objective function. Black curve: equality constraint.

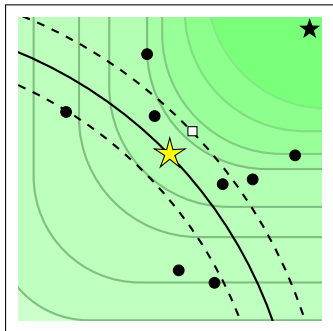
Margin-Based Equality Handling



Many equality handling techniques are **margin-based**

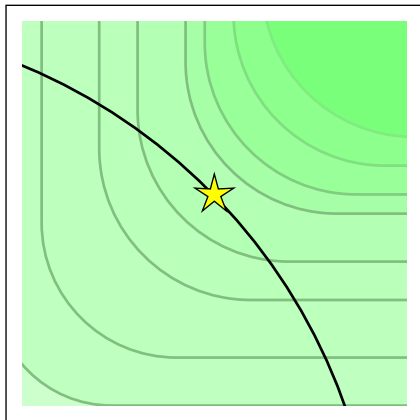
- constant margin ϵ_0
- gradually shrinking to a final margin ϵ_f

Dilemma with Margin-Based Equality Handling



- Most of such algorithms report solutions better than optimum (i.e. the white square).
- It is difficult (or even impossible) to compare solutions obtained with different margins.
- It is not clear what the right margin is.

Proposed Algorithm [3]⁴

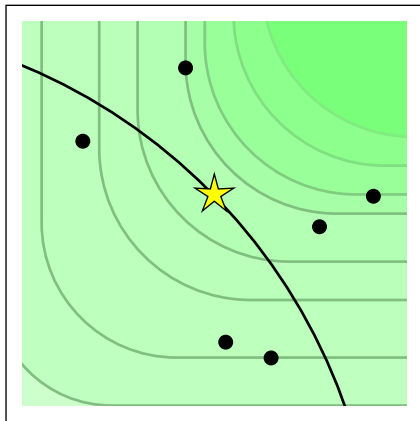


Repeat the following steps until the budget is exhausted

Figure: The black dots show the evaluated solutions.

⁴[Bagh16] S. Bagheri, W. Konen, and T. Bäck. Equality constraint handling for surrogate-assisted constrained optimization. In K. C. Tan, editor, *WCCI'2016, Vancouver*, pages 1924–1931. IEEE, 2016.

Proposed Algorithm [3]⁴



Repeat the following steps until the budget is exhausted

- Train new surrogates

Figure: The black dots show the evaluated solutions.

⁴[Bagh16] S. Bagheri, W. Konen, and T. Bäck. Equality constraint handling for surrogate-assisted constrained optimization. In K. C. Tan, editor, *WCCI'2016, Vancouver*, pages 1924–1931. IEEE, 2016.

Proposed Algorithm [3]⁴

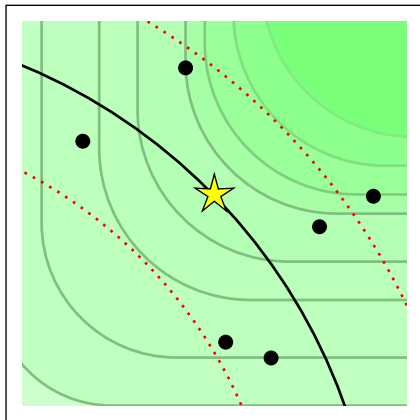


Figure: The black dots show the evaluated solutions.

Repeat the following steps until the budget is exhausted

- Train new surrogates
- Run optimization on surrogates

⁴[Bagh16] S. Bagheri, W. Konen, and T. Bäck. Equality constraint handling for surrogate-assisted constrained optimization. In K. C. Tan, editor, *WCCI'2016, Vancouver*, pages 1924–1931. IEEE, 2016.

Proposed Algorithm [3]⁴

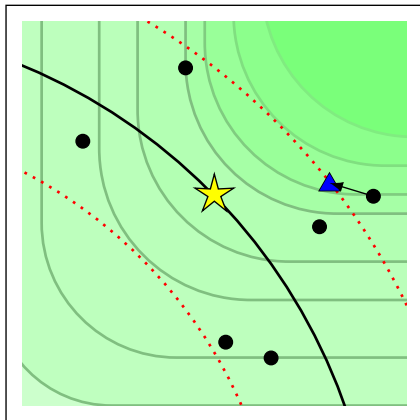


Figure: The black dots show the evaluated solutions.

Repeat the following steps until the budget is exhausted

- Train new surrogates
- Run optimization on surrogates
- Refine the solution

Refine Step:

$$\text{Min} \quad \sum_{j=1}^r (s_j^{(n)}(\vec{x}))^2$$

with surrogate $s_j^{(n)}$ for equality constraint j

⁴[Bagh16] S. Bagheri, W. Konen, and T. Bäck. Equality constraint handling for surrogate-assisted constrained optimization. In K. C. Tan, editor, *WCCI'2016, Vancouver*, pages 1924–1931. IEEE, 2016.

Proposed Algorithm [3]⁴

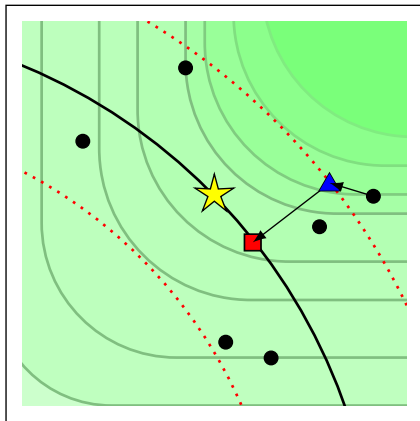


Figure: The black dots show the evaluated solutions.

Repeat the following steps until the budget is exhausted

- Train new surrogates
- Run optimization on surrogates
- Refine the solution
- **Reduce** the equality margin

Refine Step:

$$\text{Min} \quad \sum_{j=1}^r (s_j^{(n)}(\vec{x}))^2$$

with surrogate $s_j^{(n)}$ for equality constraint j

⁴[Bagh16] S. Bagheri, W. Konen, and T. Bäck. Equality constraint handling for surrogate-assisted constrained optimization. In K. C. Tan, editor, *WCCI'2016, Vancouver*, pages 1924–1931. IEEE, 2016.

Proposed Algorithm [3]⁴

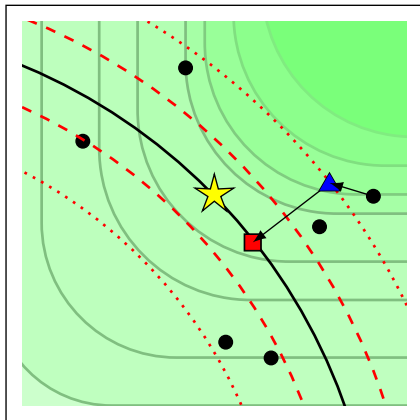


Figure: The black dots show the evaluated solutions.

Repeat the following steps until the budget is exhausted

- Train new surrogates
- Run optimization on surrogates
- Refine the solution
- **Reduce** the equality margin

Refine Step:

$$\text{Min} \quad \sum_{j=1}^r (s_j^{(n)}(\vec{x}))^2$$

with surrogate $s_j^{(n)}$ for equality constraint j

⁴[Bagh16] S. Bagheri, W. Konen, and T. Bäck. Equality constraint handling for surrogate-assisted constrained optimization. In K. C. Tan, editor, *WCCI'2016, Vancouver*, pages 1924–1931. IEEE, 2016.

Proposed Algorithm [3]⁴

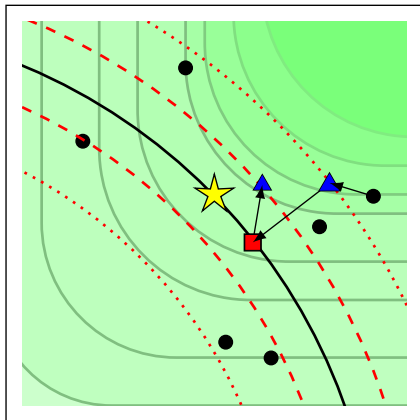


Figure: The black dots show the evaluated solutions.

Repeat the following steps until the budget is exhausted

- Train new surrogates
- Run optimization on surrogates
- Refine the solution
- **Reduce** the equality margin

Refine Step:

$$\text{Min} \quad \sum_{j=1}^r (s_j^{(n)}(\vec{x}))^2$$

with surrogate $s_j^{(n)}$ for equality constraint j

⁴[Bagh16] S. Bagheri, W. Konen, and T. Bäck. Equality constraint handling for surrogate-assisted constrained optimization. In K. C. Tan, editor, *WCCI'2016, Vancouver*, pages 1924–1931. IEEE, 2016.

Proposed Algorithm [3]⁴

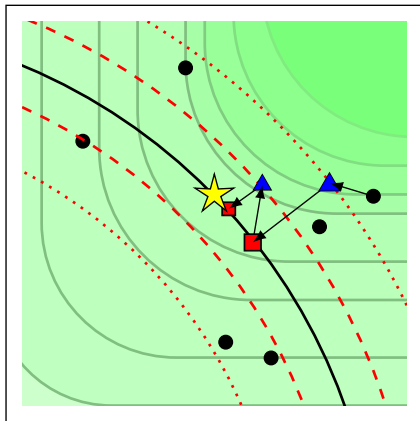


Figure: The black dots show the evaluated solutions.

Repeat the following steps until the budget is exhausted

- Train new surrogates
- Run optimization on surrogates
- Refine the solution
- **Reduce** the equality margin

Refine Step:

$$\text{Min} \quad \sum_{j=1}^r (s_j^{(n)}(\vec{x}))^2$$

with surrogate $s_j^{(n)}$ for equality constraint j

⁴[Bagh16] S. Bagheri, W. Konen, and T. Bäck. Equality constraint handling for surrogate-assisted constrained optimization. In K. C. Tan, editor, *WCCI'2016, Vancouver*, pages 1924–1931. IEEE, 2016.

Section 5

Results

The G-Problem Benchmark

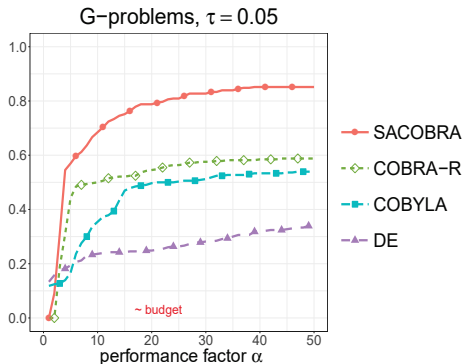
The **G-problem benchmark suite** originates from a CEC 2006 competition [Liang06]⁵.

- It is a set of 24 constrained optimization problems (COPs, G-problems) G01, . . . , G24 with various properties like dimension, number of equality / inequality constraints, feasibility ratio, etc.
- Eight of the 24 COPs have equality constraints.

⁵[Liang06] J. Liang, T. P. Runarsson, E. Mezura-Montes, M. Clerc, P. Suganthan, C. C. Coello, and K. Deb, "Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization," Journal of Applied Mechanics, vol. 41, p. 8, 2006.

Performance (Data Profiles) on G-Problems [5]⁶

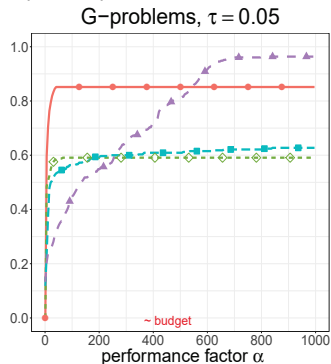
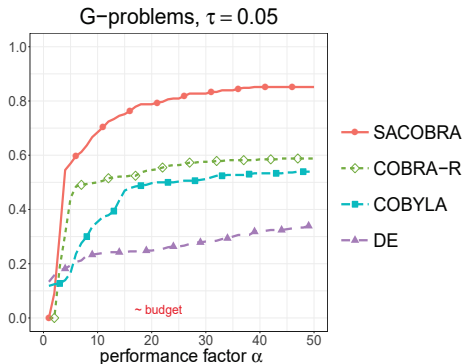
A data profile answers the question: **What is the fraction of problems that solver S can solve if the budget is $B = \alpha(d + 1)$?**



⁶[Bagh17] S. Bagheri, W. Konen, M. Emmerich, and T. Bäck. Self-adjusting parameter control for surrogate-assisted constrained optimization under limited budgets. *Applied Soft Computing*, 61:377–393, 2017.

Performance (Data Profiles) on G-Problems [5] ⁶

A data profile answers the question: **What is the fraction of problems that solver S can solve if the budget is $B = \alpha(d + 1)$?**



⁶[Bagh17] S. Bagheri, W. Konen, M. Emmerich, and T. Bäck. Self-adjusting parameter control for surrogate-assisted constrained optimization under limited budgets. *Applied Soft Computing*, 61:377–393, 2017.

Performance (Data Profiles) on G-Problems [5]⁶

A data profile answers the question: **What is the fraction of problems that solver S can solve if the budget is $B = \alpha(d + 1)$?**

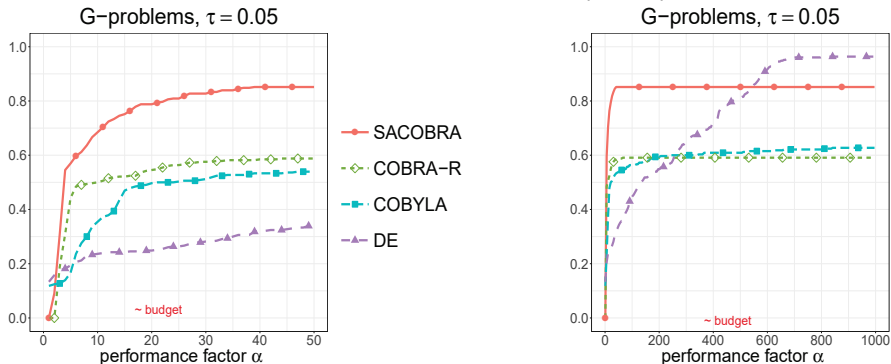


Figure: Data profiles for solvers SACOBRA [5], COBRA [11], DE [6, 14], and COBYLA [10] on G-problems G01-G11 (30 runs, each one differently seeded)

⁶[Bagh17] S. Bagheri, W. Konen, M. Emmerich, and T. Bäck. Self-adjusting parameter control for surrogate-assisted constrained optimization under limited budgets. *Applied Soft Computing*, 61:377–393, 2017.

Performance (Data Profiles) on G-Problems [5] ⁶

A data profile answers the question: **What is the fraction of problems that solver S can solve if the budget is $B = \alpha(d + 1)$?**

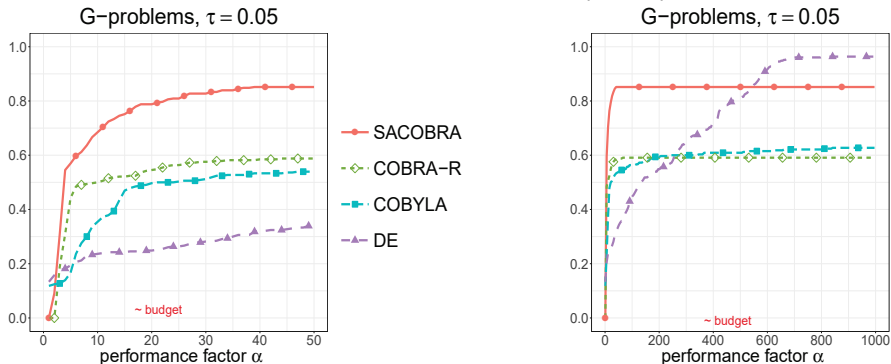


Figure: Data profiles for solvers SACOBRA [5], COBRA [11], DE [6, 14], and COBYLA [10] on G-problems G01-G11 (30 runs, each one differently seeded)

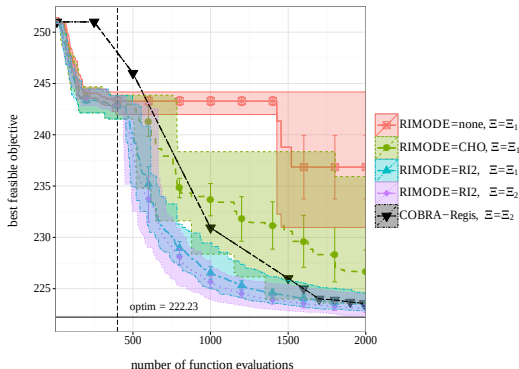
⁶[Bagh17] S. Bagheri, W. Konen, M. Emmerich, and T. Bäck. Self-adjusting parameter control for surrogate-assisted constrained optimization under limited budgets. *Applied Soft Computing*, 61:377–393, 2017.

Detailed Results on G-Problems G01-G11 [5]

Fct.	Optimum		SACOBRA [this work]	COBRA [11]	ISRES [13, 12]	RGA 10% [7]	COBYLA [10]	DE [6, 14]	CMAES [1]
G01	-15.0	m fe	-15.0 100	NA NA	-15.0 350000	-15.0 95512	-13.83 12743	-15.0 59129	NA NA
G02	-0.8036	m fe	-0.3466 400	NA NA	-0.7931 349600	-0.7857 331972	-0.197 97391	-0.8036 226994	NA NA
G03	-1.0	m fe	-1.0 300	-0.09 100	-1.001 349200	-0.9999 399804	-1.0 31069	-0.9999 211966	NA NA
G04	-30665.539	m fe	-30665.539 200	-30665.15 100	-30665.539 192000	-30665.539 26981	-30665.539 418	-30665.539 33963	NA NA
G05	5126.497	m fe	5126.498 200	5126.51 100	5126.497 195600	5126.498 39459	5126.498 194	5126.498 13375	NA NA
G06	-6961.81	m fe	-6961.81 100	-6834.48 100	-6961.81 168800	-6961.81 13577	-6961.81 134	-6961.81 2857	-6961.81 1060
G07	24.306	m fe	24.306 200	25.32 100	24.306 350000	24.471 428314	24.306 13072	24.306 94313	24.306 11283
G08	-0.0958	m fe	-0.0958 200	-0.1 100	-0.0958 160000	-0.0958 6217	-0.0282 553	-0.0958 990	NA NA
G09	680.630	m fe	680.761 300	3953.97 100	680.630 271200	680.638 388453	680.630 8973	680.630 34836	680.630 4106
G10	7049.248	m fe	7049.253 300	18031.74 100	7049.248 348800	7049.566 572629	7064.8 270840	7049.248 74875	7049.248 18781
G11	0.75	m fe	0.75 100	NA NA	0.75 137200	0.75 7215	0.75 11788	0.75 2190	NA NA
average		fe	218	100	261127	210012	40652	68681	8807

'MOPTA08' Automotive Benchmark [5]

COBRA-R Optimization for MOPTA08 Problem



- high-dimensional:
 $d = 124, n_{constr} = 68$

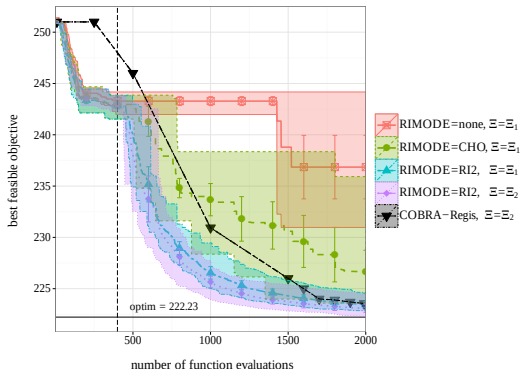


► Detail

⁷Don R. Jones, "Benchmark problems for constrained global optimization with high-dimensional black-box functions",
<https://github.com/donaldratnerjones/2025JonesBenchmarks>, 2025

'MOPTA08' Automotive Benchmark [5]

COBRA-R Optimization for MOPTA08 Problem



- high-dimensional:
 $d = 124, n_{constr} = 68$
- Our best result 222.4 is slightly better than 'best-known' 222.61 in [9]⁷

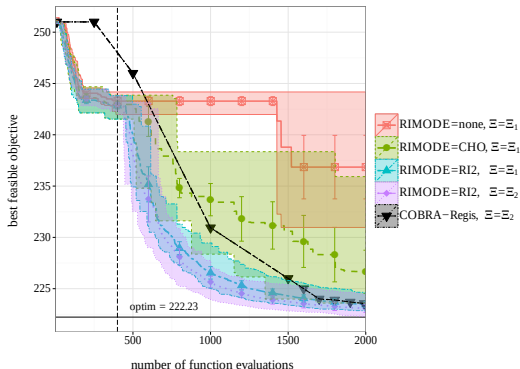


Detail

⁷Don R. Jones, "Benchmark problems for constrained global optimization with high-dimensional black-box functions",
<https://github.com/donaldratnerjones/2025JonesBenchmarks>, 2025

'MOPTA08' Automotive Benchmark [5]

COBRA-R Optimization for MOPTA08 Problem



- high-dimensional:
 $d = 124, n_{constr} = 68$
- Our best result 222.4 is slightly better than 'best-known' 222.61 in [9]⁷



Detail

⁷Don R. Jones, "Benchmark problems for constrained global optimization with high-dimensional black-box functions",
<https://github.com/donaldratnerjones/2025JonesBenchmarks>, 2025

Section 6

Summary and Outlook

Strengths:

- RBF surrogates stable and efficient for many COPs
- Equality constraint handling is not easy but doable

Strengths:

- RBF surrogates stable and efficient for many COPs
- Equality constraint handling is not easy but doable
- Function evaluations: Reduced by factor 10 – 100 (!)

Strengths:

- RBF surrogates stable and efficient for many COPs
- Equality constraint handling is not easy but doable
- Function evaluations: Reduced by factor 10 – 100 (!)
- Works on high-dimensional COPs

Strengths:

- RBF surrogates stable and efficient for many COPs
- Equality constraint handling is not easy but doable
- Function evaluations: Reduced by factor 10 – 100 (!)
- Works on high-dimensional COPs
- Adaptive elements: *one* parameter setting for all COPs

Strengths:

- RBF surrogates stable and efficient for many COPs
- Equality constraint handling is not easy but doable
- Function evaluations: Reduced by factor 10 – 100 (!)
- Works on high-dimensional COPs
- Adaptive elements: *one* parameter setting for all COPs

Limitations:

- Highly multimodal functions

Strengths:

- RBF surrogates stable and efficient for many COPs
- Equality constraint handling is not easy but doable
- Function evaluations: Reduced by factor 10 – 100 (!)
- Works on high-dimensional COPs
- Adaptive elements: *one* parameter setting for all COPs

Limitations:

- Highly multimodal functions
- Non-smooth or noisy functions

Strengths:

- RBF surrogates stable and efficient for many COPs
- Equality constraint handling is not easy but doable
- Function evaluations: Reduced by factor 10 – 100 (!)
- Works on high-dimensional COPs
- Adaptive elements: *one* parameter setting for all COPs

Limitations:

- Highly multimodal functions
- Non-smooth or noisy functions
- Hidden constraints (occasional NaNs)

Strengths:

- RBF surrogates stable and efficient for many COPs
- Equality constraint handling is not easy but doable
- Function evaluations: Reduced by factor 10 – 100 (!)
- Works on high-dimensional COPs
- Adaptive elements: *one* parameter setting for all COPs

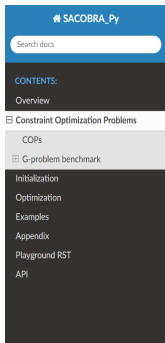
Limitations:

- Highly multimodal functions
- Non-smooth or noisy functions
- Hidden constraints (occasional NaNs)

... one last thing ... :-)

As of today:

... one last thing ... :-)
As of today:



Constraint Optimization Problems

This chapter defines COPs (Constraint Optimization Problems) and introduces the G-problem benchmark.

COPs

A constrained optimization problem (COP) for numerical and continuous quantities in $\mathbb{R}^d$ is defined as:

$$\begin{aligned} \text{Min} \quad & f(\vec{x}), \quad \vec{x} \in [\vec{a}, \vec{b}] \subset \mathbb{R}^d \\ \text{subject to} \quad & g_i(\vec{x}) \leq 0, \quad i = 1, 2, \dots, m \\ & h_j(\vec{x}) = 0, \quad j = 1, 2, \dots, r \end{aligned}$$

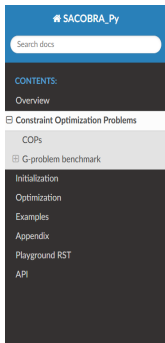
G-problem benchmark

The G-problem benchmark suite originates from a CEC 2006 competition [LiangRunar]. It is a set of 24 constrained optimization problems (COPs, G-problems) G01, ..., G24 with various properties like dimension, number of equality / inequality constraints, feasibility ratio, etc. Eight of the 24 COPs

The **SACOBRA Python port** is out!

- Beta version, not yet all features of the R-version
- GitHub: https://github.com/WolfgangKonen/SACOBRA_Py

... one last thing ... :-)
As of today:



Constraint Optimization Problems

This chapter defines COPs (Constraint Optimization Problems) and introduces the G-problem benchmark.

COPs

A constrained optimization problem (COP) for numerical and continuous quantities in $\mathbb{R}^d$ is defined as:

$$\begin{aligned} \text{Min} \quad & f(\vec{x}), \quad \vec{x} \in [\vec{a}, \vec{b}] \subset \mathbb{R}^d \\ \text{subject to} \quad & g_i(\vec{x}) \leq 0, \quad i = 1, 2, \dots, m \\ & h_j(\vec{x}) = 0, \quad j = 1, 2, \dots, r \end{aligned}$$

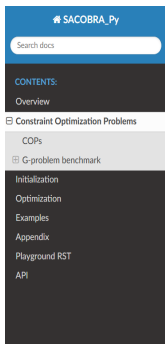
G-problem benchmark

The G-problem benchmark suite originates from a CEC 2006 competition [LiangRunar]. It is a set of 24 constrained optimization problems (COPs, G-problems) G01, ..., G24 with various properties like dimension, number of equality / inequality constraints, feasibility ratio, etc. Eight of the 24 COPs

The **SACOBRA Python port** is out!

- Beta version, not yet all features of the R-version
- GitHub: https://github.com/WolfgangKonen/SACOBRA_Py
- Faster than R by a factor of 4 – 40 (!)

... one last thing ... :-)
As of today:



Constraint Optimization Problems

This chapter defines COPs (Constraint Optimization Problems) and introduces the G-problem benchmark.

COPs

A constrained optimization problem (COP) for numerical and continuous quantities in $\mathbb{R}^d$ is defined as:

$$\begin{aligned} \text{Min} \quad & f(\vec{x}), \quad \vec{x} \in [\vec{a}, \vec{b}] \subset \mathbb{R}^d \\ \text{subject to} \quad & g_i(\vec{x}) \leq 0, \quad i = 1, 2, \dots, m \\ & h_j(\vec{x}) = 0, \quad j = 1, 2, \dots, r \end{aligned}$$

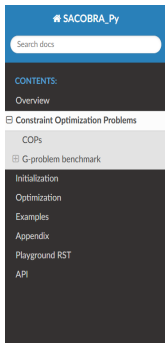
G-problem benchmark

The G-problem benchmark suite originates from a CEC 2006 competition [LiangRunar]. It is a set of 24 constrained optimization problems (COPs, G-problems) G01, ..., G24 with various properties like dimension, number of equality / inequality constraints, feasibility ratio, etc. Eight of the 24 COPs

The **SACOBRA Python port** is out!

- Beta version, not yet all features of the R-version
- GitHub: https://github.com/WolfgangKonen/SACOBRA_Py
- Faster than R by a factor of 4 – 40 (!)
- Streamlined in code

... one last thing ... :-)
As of today:



Constraint Optimization Problems

This chapter defines COPs (Constraint Optimization Problems) and introduces the G-problem benchmark.

COPs

A constrained optimization problem (COP) for numerical and continuous quantities in $\mathbb{R}^d$ is defined as:

$$\begin{aligned} \text{Min} \quad & f(\vec{x}), \quad \vec{x} \in [\vec{a}, \vec{b}] \subset \mathbb{R}^d \\ \text{subject to} \quad & g_i(\vec{x}) \leq 0, \quad i = 1, 2, \dots, m \\ & h_j(\vec{x}) = 0, \quad j = 1, 2, \dots, r \end{aligned}$$

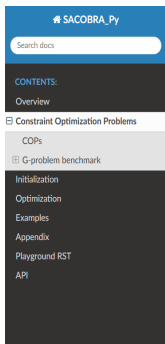
G-problem benchmark

The G-problem benchmark suite originates from a CEC 2006 competition [LiangRunar]. It is a set of 24 constrained optimization problems (COPs, G-problems) G01, ..., G24 with various properties like dimension, number of equality / inequality constraints, feasibility ratio, etc. Eight of the 24 COPs

The **SACOBRA Python port** is out!

- Beta version, not yet all features of the R-version
- GitHub: https://github.com/WolfgangKonen/SACOBRA_Py
- Faster than R by a factor of 4 – 40 (!)
- Streamlined in code
- Tested on many G-functions (not yet finished)

... one last thing ... :-)
As of today:



Constraint Optimization Problems

This chapter defines COPs (Constraint Optimization Problems) and introduces the G-problem benchmark.

COPs

A constrained optimization problem (COP) for numerical and continuous quantities in $\mathbb{R}^d$ is defined as:

$$\begin{aligned} \text{Min} \quad & f(\vec{x}), \quad \vec{x} \in [\vec{a}, \vec{b}] \subset \mathbb{R}^d \\ \text{subject to} \quad & g_i(\vec{x}) \leq 0, \quad i = 1, 2, \dots, m \\ & h_j(\vec{x}) = 0, \quad j = 1, 2, \dots, r \end{aligned}$$

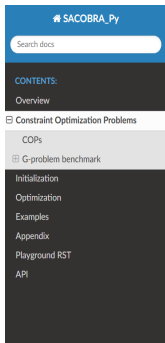
G-problem benchmark

The G-problem benchmark suite originates from a CEC 2006 competition [LiangRunar]. It is a set of 24 constrained optimization problems (COPs, G-problems) G01, ..., G24 with various properties like dimension, number of equality / inequality constraints, feasibility ratio, etc. Eight of the 24 COPs

The **SACOBRA Python port** is out!

- Beta version, not yet all features of the R-version
- GitHub: https://github.com/WolfgangKonen/SACOBRA_Py
- Faster than R by a factor of 4 – 40 (!)
- Streamlined in code
- Tested on many G-functions (not yet finished)
- Documentation on readthedocs.io.

... one last thing ... :-)
As of today:



Constraint Optimization Problems

This chapter defines COPs (Constraint Optimization Problems) and introduces the G-problem benchmark.

COPs

A constrained optimization problem (COP) for numerical and continuous quantities in $\mathbb{R}^d$ is defined as:

$$\begin{aligned} \text{Min} \quad & f(\vec{x}), \quad \vec{x} \in [\vec{a}, \vec{b}] \subset \mathbb{R}^d \\ \text{subject to} \quad & g_i(\vec{x}) \leq 0, \quad i = 1, 2, \dots, m \\ & h_j(\vec{x}) = 0, \quad j = 1, 2, \dots, r \end{aligned}$$

G-problem benchmark

The G-problem benchmark suite originates from a CEC 2006 competition [LiangRunar]. It is a set of 24 constrained optimization problems (COPs, G-problems) G01, ..., G24 with various properties like dimension, number of equality / inequality constraints, feasibility ratio, etc. Eight of the 24 COPs

The **SACOBRA Python port** is out!

- Beta version, not yet all features of the R-version
- GitHub: https://github.com/WolfgangKonen/SACOBRA_Py
- Faster than R by a factor of 4 – 40 (!)
- Streamlined in code
- Tested on many G-functions (not yet finished)
- Documentation on readthedocs.io.

**Thanks For
Your Attention**



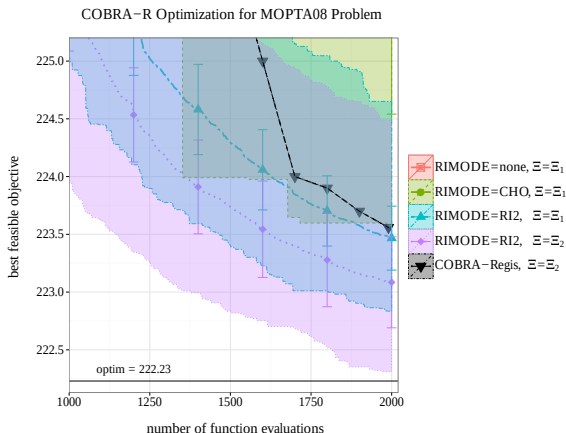
Section 7

Appendix

Detailed Results on Equality G-Problems [3]

Fct. Optimum			SACOBRA [this work]	Zhang [14]	ISRES [13]	RGA [7]	Jiao [8]	DE [6, 14]
G03	-1.0	m fe	-1.0 100	-1.0005 25493	-1.001 349200	-0.9999 399804	-1.0005 19534	-0.8414 13325
G05	5126.497	m fe	5126.497 300	5126.497 21363	5126.497 195600	5126.498 39459	5126.497 2050	5126.498 8108
G11	0.750	m fe	0.750 100	0.749 6609	0.750 137200	0.750 7215	0.749 135	0.750 2099
G13	0.0539	m fe	0.0539 300	0.0539 19180	0.0539 223600	– –	0.0539 3103	0.068 23637
G14	-47.765	m fe	-47.759 500	-47.765 34825	– –	– –	-47.765 6093	-47.761 72015
G15	961.715	m fe	961.715 500	961.715 11706	– –	– –	961.715 757	961.715 5666
G17	8853.534	m fe	8853.794 500	8868.539 43369	– –	– –	8853.534 3203	8867.606 37532
G21	193.724	m fe	193.725 500	193.735 23631	– –	– –	193.724 46722	193.790 35559
average		fe	350	23272	226400	148826	10200	24742

'MOPTA08' Automotive Benchmark – Figure Detail



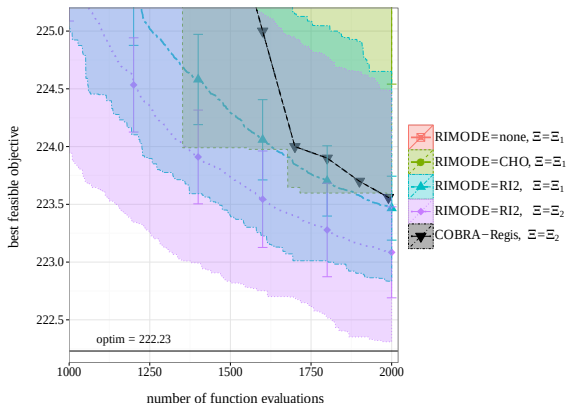
- high-dimensional:
 $d = 124, n_{constr} = 68$

Back

⁸Don R. Jones, "Benchmark problems for constrained global optimization with high-dimensional black-box functions",
<https://github.com/donaldratnerjones/2025JonesBenchmarks>, 2025

'MOPTA08' Automotive Benchmark – Figure Detail

COBRA-R Optimization for MOPTA08 Problem



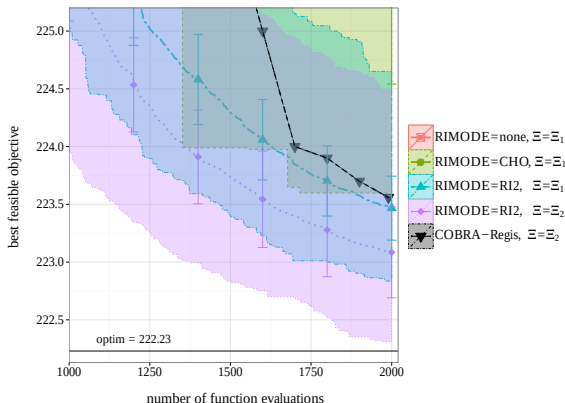
- high-dimensional:
 $d = 124, n_{constr} = 68$
- Our best result 222.4 is slightly better than 'best-known' 222.61 in [9]⁸

▶ Back

⁸Don R. Jones, "Benchmark problems for constrained global optimization with high-dimensional black-box functions",
<https://github.com/donaldratnerjones/2025JonesBenchmarks>, 2025

'MOPTA08' Automotive Benchmark – Figure Detail

COBRA-R Optimization for MOPTA08 Problem



- high-dimensional:
 $d = 124, n_{constr} = 68$
- Our best result 222.4 is slightly better than 'best-known' 222.61 in [9]⁸

▶ Back

⁸Don R. Jones, "Benchmark problems for constrained global optimization with high-dimensional black-box functions",
<https://github.com/donaldratnerjones/2025JonesBenchmarks>, 2025

Comparing Kriging and RBF Surrogates [4, 2]

- **Kriging:**

- PRO: uncertainty info, leading to EI
- CON: often crashes (with current software)
- CON: time-consuming (for many dimensions or many points)

- **RBF:**

- PRO: fast to compute
- PRO: robust (very seldom crashes)
- PRO: many models available
- CON: lacks uncertainty info and thus EI

⁹**SOCU: Surrogate-Assisted Optimization encompassing Constraints and Uncertainties** [2]

Comparing Kriging and RBF Surrogates [4, 2]

- **Kriging:**
 - PRO: uncertainty info, leading to EI
 - CON: often crashes (with current software)
 - CON: time-consuming (for many dimensions or many points)
- **RBF:**
 - PRO: fast to compute
 - PRO: robust (very seldom crashes)
 - PRO: many models available
 - CON: lacks uncertainty info and thus EI
- In [4] ("Comparing Kriging and RBF Surrogates") and [2] we propose a solution **SOCU-RBF**⁹ to the last CON which makes EI (Expected Improvement) available for RBF.

⁹**SOCU**: Surrogate-Assisted Optimization encompassing Constraints and Uncertainties [2]

Comparing Kriging and RBF Surrogates [4, 2]

- **Kriging:**

- PRO: uncertainty info, leading to EI
- CON: often crashes (with current software)
- CON: time-consuming (for many dimensions or many points)

- **RBF:**

- PRO: fast to compute
- PRO: robust (very seldom crashes)
- PRO: many models available
- CON: lacks uncertainty info and thus EI

- In [4] ("Comparing Kriging and RBF Surrogates") and [2] we propose a solution **SOCU-RBF**⁹ to the last CON which makes EI (Expected Improvement) available for RBF.
- In [2] we showed that SOCU-RBF performs well on the G-problem benchmark, but is inferior to SACOBRA.

⁹**SOCU**: Surrogate-Assisted Optimization encompassing Constraints and Uncertainties [2]

Comparing Kriging and RBF Surrogates [4, 2]

- **Kriging:**

- PRO: uncertainty info, leading to EI
- CON: often crashes (with current software)
- CON: time-consuming (for many dimensions or many points)

- **RBF:**

- PRO: fast to compute
 - PRO: robust (very seldom crashes)
 - PRO: many models available
 - CON: lacks uncertainty info and thus EI
- In [4] ("Comparing Kriging and RBF Surrogates") and [2] we propose a solution **SOCU-RBF**⁹ to the last CON which makes EI (Expected Improvement) available for RBF.
 - In [2] we showed that SOCU-RBF performs well on the G-problem benchmark, but is inferior to SACOBRA.

⁹**SOCU**: Surrogate-Assisted Optimization encompassing Constraints and Uncertainties [2]

- [1] D. V. Arnold and N. Hansen.
A (1+1)-CMA-ES for constrained optimisation.
In Proceedings of the 14th conference on Genetic and evolutionary computation (GECCO), pages 297–304. ACM, 2012.
- [2] S. Bagheri, W. Konen, R. Allmendinger, J. Branke, K. Deb, J. Fieldsend, D. Quagliarella, and K. Sindhya.
Constraint handling in efficient global optimization.
In P. A. N. Bosman, editor, Genetic and Evolutionary Computation Conference 2017 (GECCO'17), Berlin, page 1. ACM, 2017.
- [3] S. Bagheri, W. Konen, and T. Bäck.
Equality constraint handling for surrogate-assisted constrained optimization.
In K. C. Tan, editor, WCCI'2016, Vancouver, pages 1924–1931. IEEE, 2016.

Further Reading II

- [4] S. Bagheri, W. Konen, and T. Bäck.
Comparing Kriging and radial basis function surrogates.
In F. Hoffmann and E. Hüllermeier, editors, *Proceedings 27. Workshop Computational Intelligence*, pages 243–259.
Universitätsverlag Karlsruhe, 2017.
- [5] S. Bagheri, W. Konen, M. Emmerich, and T. Bäck.
Self-adjusting parameter control for surrogate-assisted constrained optimization under limited budgets.
Applied Soft Computing, 61:377–393, 2017.
- [6] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Zumer.
Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems.
Evolutionary Computation, IEEE Transactions on, 10(6):646–657, 2006.

Further Reading III

- [7] P. Chootinan and A. Chen.
Constraint handling in genetic algorithms using a gradient-based repair method.
Computers & Operations Research, 33(8):2263–2281, 2006.
- [8] L. Jiao, L. Li, R. Shang, F. Liu, and R. Stolkin.
A novel selection evolutionary strategy for constrained optimization.
Information Sciences, 239:122–141, 2013.
- [9] D. R. Jones.
Benchmark problems for constrained global optimization with high-dimensional black-box functions.
Technical report, University of Michigan, USA, 2025.

[10] M. J. D. Powell.

A direct search optimization method that models the objective and constraint functions by linear interpolation.

In Advances In Optimization And Numerical Analysis, pages 51–67. Springer, 1994.

[11] R. G. Regis.

Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points.

Engineering Optimization, 46(2):218–243, 2014.

[12] T. P. Runarsson and X. Yao.

Stochastic ranking for constrained evolutionary optimization.

IEEE Transactions on Evolutionary Computation, 4(3):284–294, 2000.

- [13] T. P. Runarsson and X. Yao.

Search biases in constrained evolutionary optimization.

IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 35(2):233–243, 2005.

- [14] H. Zhang and G. Rangaiah.

An efficient constraint handling method with integrated differential evolution for numerical and engineering optimization.

Computers & Chemical Engineering, 37:74 – 88, 2012.