

TU3A-5

A Fast Rank-Revealing Method for Solving High-Dimensional Global Optimization Problems

Dan Jiao

Elmore Family School of Electrical and Computer Engineering
Purdue University, West Lafayette, IN, USA

- **Problem Statement**
- **Proposed Method**
- **Validation and Application**
- **Summary**

Design Parameters

$$\mathbf{x} = [x_1, x_2, \dots, x_p]^T$$

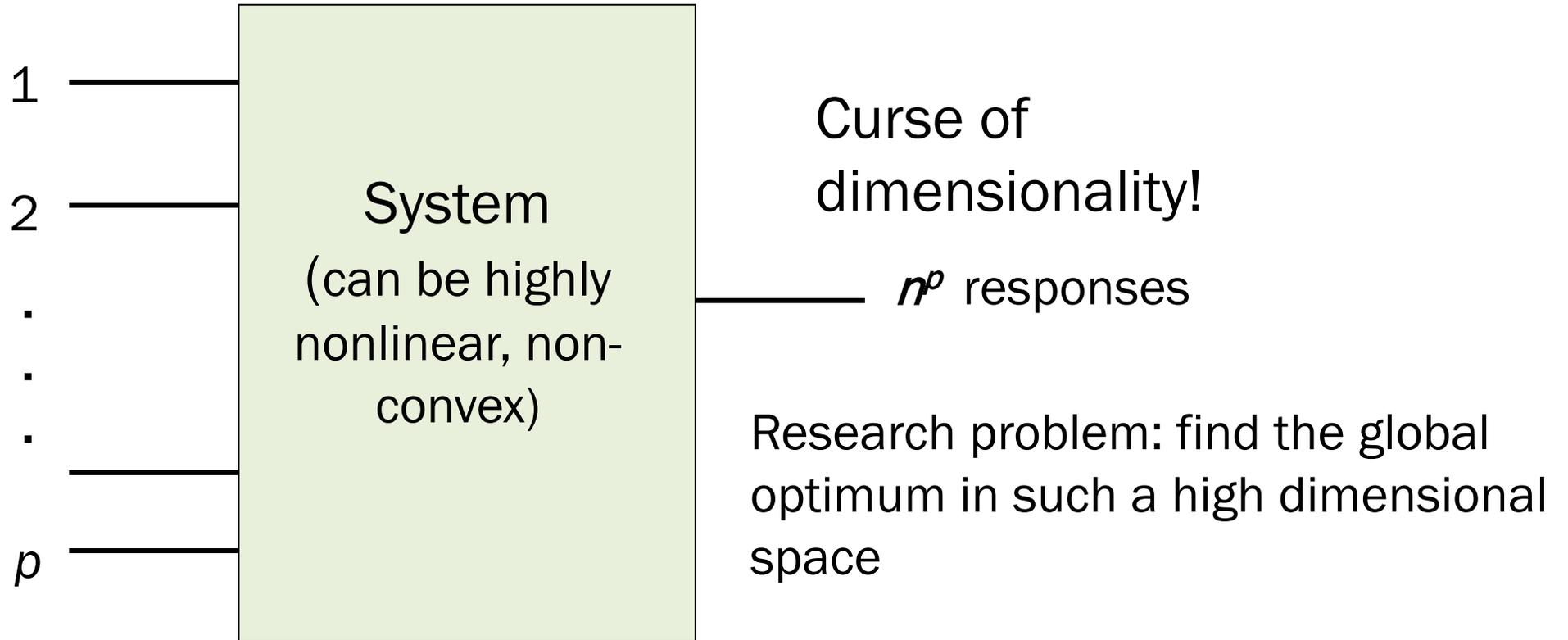
Design Space

$$X = \{\mathbf{x} \in R^p : \mathbf{g}(\mathbf{x}) \leq \mathbf{0}, \mathbf{h}(\mathbf{x}) = \mathbf{0}, x_i \in \Gamma_i, i = 1, \dots, p\}$$

Optimization Problem:

$$\min_{\mathbf{x} \in X} \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_o(\mathbf{x})]^T$$

A Different View



p parameters, each having
' n ' choices

Research Objective

- High-dimension global optimization is broadly used in system optimization, decision making, design automation, etc.
- Develop a fast and accurate method for high-dimensional global optimization, and use it for system design applications

Existing Approaches

- Machine learning
- Bayesian Optimization
- Genetic Algorithm
- Swarm-based methods
- Branch and cut
- Local optimization
- Others

Questions Addressed

- ❑ Questions that have not been well addressed by the state of the art methods
 - Has the global optimum been found?
 - Does the process converge?
 - Can the number of steps be reduced, i.e., be faster?
- ❑ We have developed a genuinely new algorithm for solving generic high-dimensional optimization problem, suitable for nonlinear, non-convex systems as well as linear and convex ones

All Responses:

$$\mathbf{R}_{1 \times n^p} = (f(\mathbf{x}_1) \quad f(\mathbf{x}_2) \quad \cdots \quad f(\mathbf{x}_{n^p}))$$

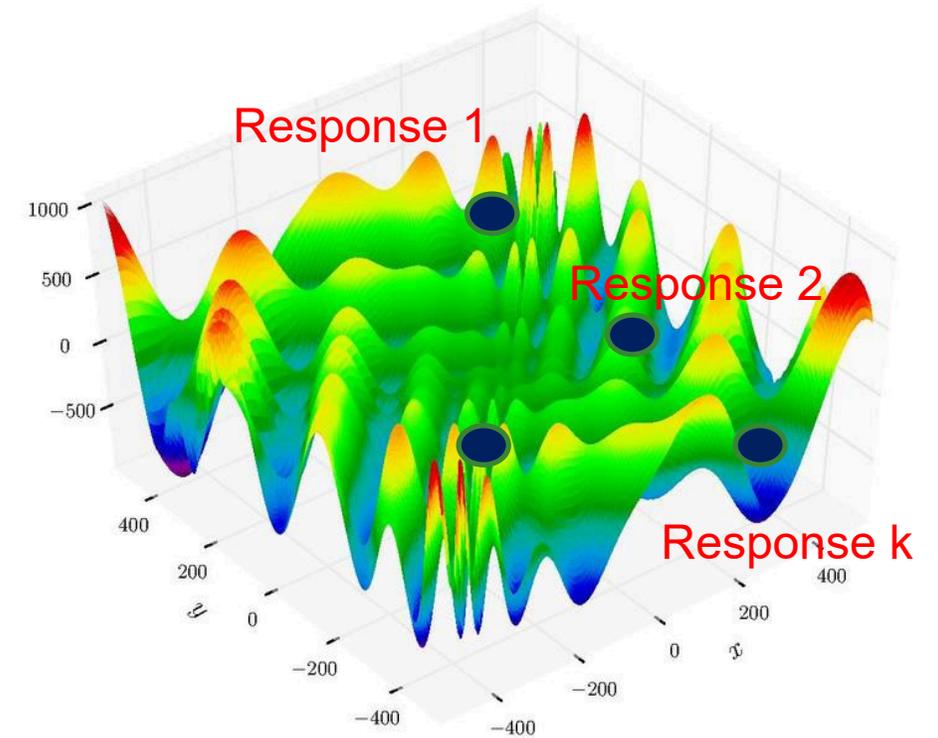
Reshaped Response Matrix:

$$\mathbf{R}_{r \times n^{p-1}} = \begin{pmatrix} f(x_\nu^1, \bar{\mathbf{x}}_1) & f(x_\nu^1, \bar{\mathbf{x}}_2) & \cdots & f(x_\nu^1, \bar{\mathbf{x}}_{n^{p-1}}) \\ f(x_\nu^2, \bar{\mathbf{x}}_1) & f(x_\nu^2, \bar{\mathbf{x}}_2) & \cdots & f(x_\nu^2, \bar{\mathbf{x}}_{n^{p-1}}) \\ \vdots & \vdots & \ddots & \vdots \\ f(x_\nu^r, \bar{\mathbf{x}}_1) & f(x_\nu^r, \bar{\mathbf{x}}_2) & \cdots & f(x_\nu^r, \bar{\mathbf{x}}_{n^{p-1}}) \end{pmatrix}$$

$$\mathbf{R}_{r \times n^{p-1}} = R_{:, \hat{j}} (R_{\hat{i}, \hat{j}})^{-1} \begin{bmatrix} R_{\hat{i}, \hat{j}} & R_{\hat{i}, J \neq \hat{j}} \end{bmatrix}$$

k distinct responses

Other $n^{p-1} - k$ responses overlap with the k responses, thus no need to compute



- To find the following factorization

$$\mathbf{R}_{r \times n^p} = \mathbf{R}_{:,j} (\mathbf{R}_{\hat{I},\hat{J}})^{-1} \mathbf{R}_{\hat{I},:}$$

- Existing methods

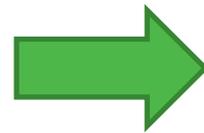
- n^p function evaluations

- $O(n^p)$ complexity (Full Cross Approximation (FCA), Adaptive CA, ...)

- Proposed method

- $O(k)$ evaluations

- $O(rk^2)$ complexity



Independent of n^p .
Overcome curse of
dimensionality!

- Rewrite \mathbf{R} into

$$\mathbf{R}_{r \times n^{p-1}} = \mathbf{S}_{r \times m} \mathbf{B}_{m \times n^{p-1}} + \mathbf{C},$$

where $m = n \times (p - 1)$

$$\mathbf{B}(:, j) = (e_{q_1} \ e_{q_2} \ \cdots \ e_{q_{p-1}})^T$$

$$\mathbf{S}(i, :) = \begin{pmatrix} f \left(x_\nu = x_\nu^i, \{x_{c1} = x_{c1}^j\}, \{\text{Other } x_i = x_i^{ref}\} \right) \\ f \left(x_\nu = x_\nu^i, \{x_{c2} = x_{c2}^j\}, \{\text{Other } x_i = x_i^{ref}\} \right) \\ \vdots \\ f \left(x_\nu = x_\nu^i, \{x_{c_{p-1}} = x_{c_{p-1}}^j\}, \{\text{Other } x_i = x_i^{ref}\} \right) \end{pmatrix}^T$$

Proposed method

Step 1

$$E^0 = \begin{bmatrix} 2 & 0 & -1 \\ 1 & 4 & 0 \\ -1 & 1 & 3 \end{bmatrix}$$

X1 on
X2 on
X3 on

Objective 1
Objective 2
Objective 3

$$i_1 = 2, j_1 = 6$$

$$a_1 = \begin{bmatrix} 2 \\ 5 \\ 0 \end{bmatrix},$$

$$c_1 = \frac{1}{5} [1 \quad 4 \quad 0]$$

Full Cross Approximation (FCA)

$$E^0 = \begin{bmatrix} -1 & 0 & -1 & 2 & 1 & 2 & 1 \\ 0 & 4 & 4 & 1 & 1 & 5 & 5 \\ 3 & 1 & 4 & -1 & 2 & 0 & 3 \end{bmatrix}$$

$$i_1 = 2, j_1 = 6$$

$$a_1 = \begin{bmatrix} 2 \\ 5 \\ 0 \end{bmatrix},$$

$$b_1 = \frac{1}{5} [0 \quad 4 \quad 4 \quad 1 \quad 1 \quad 5 \quad 5]$$

Proposed method

Step 2

$$\mathbf{E}_1^0 = \mathbf{E}^0 - \mathbf{a}_1 \times \mathbf{c}_1^T \times \mathbf{B}$$

$$= \begin{bmatrix} 1.6 & -1.6 & -1 \\ 0 & 0 & 0 \\ -1 & 1 & 3 \end{bmatrix}$$

$$i_2 = 3, j_2 = 3$$

$$\mathbf{a}_2 = \begin{bmatrix} -2.6 \\ 0 \\ 4 \end{bmatrix},$$

$$\mathbf{c}_2 = \frac{1}{4} \begin{bmatrix} -1 & 1 & 3 \end{bmatrix}$$

Full Cross Approximation (FCA)

$$\mathbf{E}_1^0 = \mathbf{E}^0 - \mathbf{a}_1 \times \mathbf{b}_1^T$$

$$= \begin{bmatrix} -1 & -1.6 & -2.6 & 1.6 & 0.6 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 1 & 4 & -1 & 2 & 0 & 3 \end{bmatrix}$$

$$i_3 = 3, j_3 = 3$$

$$\mathbf{a}_2 = \begin{bmatrix} -2.6 \\ 0 \\ 4 \end{bmatrix},$$

$$\mathbf{b}_2 = \frac{1}{4} \begin{bmatrix} 3 & 1 & 4 & -1 & 2 & 0 & 3 \end{bmatrix}$$

Proposed method

Step 3

$$\mathbf{E}_2^0 = \mathbf{E}_1^0 - \mathbf{a}_2 \times \mathbf{c}_2^T \times \mathbf{B}$$

$$= \begin{bmatrix} 0.95 & -0.95 & 0.95 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$i_3 = 1, j_3 = 5$$

$$\mathbf{a}_3 = \begin{bmatrix} 1.9 \\ 0 \\ 0 \end{bmatrix},$$

$$\mathbf{c}_3 = \frac{1}{1.9} \begin{bmatrix} 0.95 & -0.95 & 0.95 \end{bmatrix}$$

Full Cross Approximation (FCA)

$$\mathbf{E}_2^0 = \mathbf{E}_1^0 - \mathbf{a}_2 \times \mathbf{b}_2^T$$

$$= \begin{bmatrix} 0.95 & -0.95 & 0 & 0.95 & 1.9 & 0 & 0.95 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$i_3 = 1, j_3 = 5$$

$$\mathbf{a}_3 = \begin{bmatrix} 1.9 \\ 0 \\ 0 \end{bmatrix},$$

$$\mathbf{b}_3 = \frac{1}{1.9} \begin{bmatrix} 0.95 & -0.95 & 0 & 0.95 & 1.9 & 0 & 0.95 \end{bmatrix}$$

Proposed method

Full Cross Approximation (FCA)

Step 4

$$\mathbf{E}_3^0 = \mathbf{E}_2^0 - \mathbf{a}_3 \times \mathbf{c}_3^T \times \mathbf{B}$$

$$= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{E}_3^0 = \mathbf{E}_2^0 - \mathbf{a}_3 \times \mathbf{b}_3^T$$

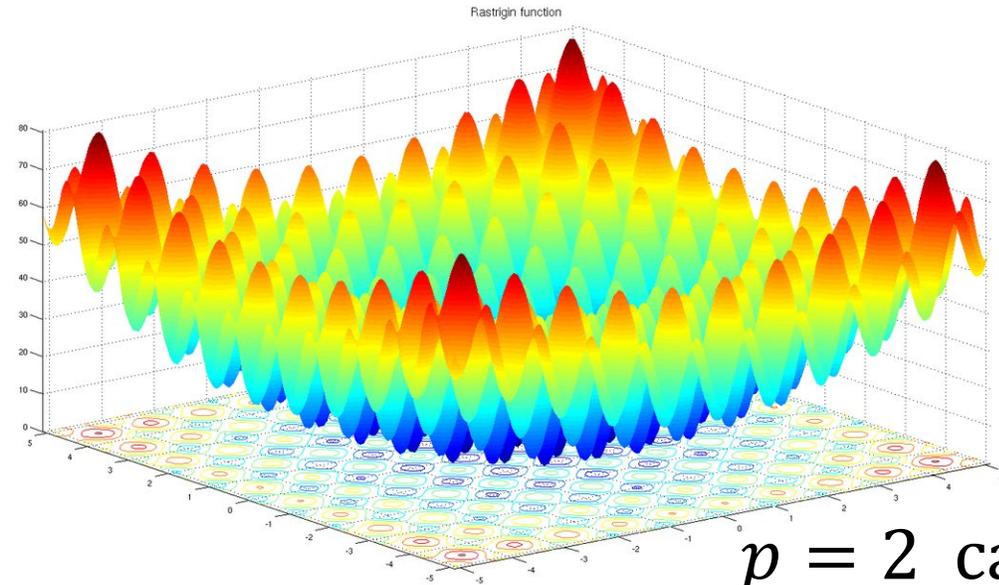
$$= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- FCA needs to compute 1-by- n^p b_i^T and r -by-1 a_i , costing $O(r \times n^p)$ at each step
- Proposed method computes 1-by- m c_i^T and r -by-1 a_i , costing $O(r \times n)$ at each step, while obtaining the same result

- Rastrigin function

$$f(\{x\}) = pA + \sum_{i=1}^p x_i^2 - A \cos(2\pi x_i)$$

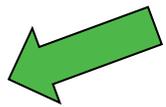
$$x_i \in (-5.12, 5.12)$$



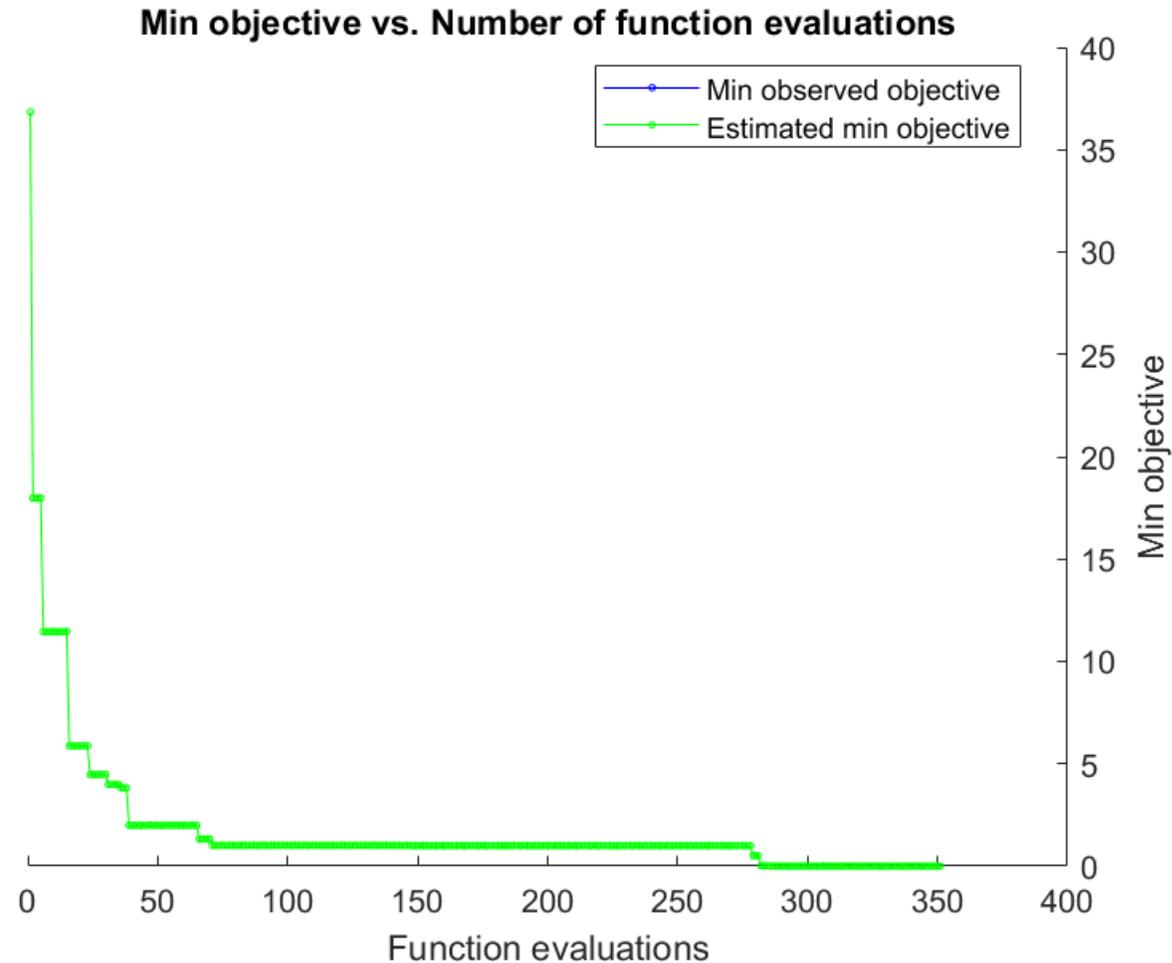
Example 1 ($p=2$): This Method

ranknum=7 numApp=21 finalRank=8
min value =0.000000e+00
min value occurs at i=1 coord=0.000000e+00
min value occurs at i=2 coord=0.000000e+00

tEnd =

0.1406  (seconds)

Example 1 ($p=2$): B0 Method



Time: $2.6351e+03$ s; 300 iterations (Matlab B0)

Example 1 (p=2): B0 Method

Iter	Eval	Objective	Objective	BestSoFar	BestSoFar	x1	x2
	result		runtime	(observed)	(estim.)		
281	Accept	0.71501	0.0013015	0.5082	0.5082	-0.060357	-0.0020672
282	Best	0.022552	0.0015578	0.022552	0.022555	0.0075754	-0.0075039
283	Best	0.017886	0.0015069	0.017886	0.017868	-0.0071297	0.0062717
284	Accept	0.13794	0.0013597	0.017886	0.017868	-0.01201	-0.023497
285	Accept	38.427	0.0014442	0.017886	0.017868	-2.9838	-4.2698
286	Best	0.011758	0.001423	0.011758	0.011765	0.00068291	0.0076689
287	Accept	42.041	0.0015494	0.011758	0.011765	-2.4825	3.97
288	Best	0.00073832	0.0014768	0.00073832	0.00063452	-0.0010001	0.0016496
289	Accept	52.39	0.0019471	0.00073832	0.00063375	-4.2478	2.611
290	Accept	36.155	0.0012744	0.00073832	0.000634	-2.2786	1.6375
291	Accept	73.909	0.0012375	0.00073832	0.00063135	-4.656	-4.3837
292	Accept	47.854	0.0015809	0.00073832	0.00063104	-4.2499	-3.239
293	Accept	55.335	0.0016832	0.00073832	0.00063034	-1.4944	-4.3229
294	Accept	47.507	0.0015426	0.00073832	0.00063024	1.3693	-5.118
295	Accept	51.091	0.0017145	0.00073832	0.00062977	0.42312	-4.307
296	Accept	75.693	0.0015095	0.00073832	0.00062664	4.4124	4.4056

Time: 2.6351e+03 s; 300 iterations (Matlab B0)

Example 1 (p=21): This Method

$x_i \in (-5.12, 5.12)$
 $n = 21, j = 51$
 51^{21} choices

ranknum=41 numApp=101 finalRank=41
min value = -2.100000e+02
min value occurs at i=1 coord=0.000000e+00
min value occurs at i=2 coord=0.000000e+00
min value occurs at i=3 coord=0.000000e+00
min value occurs at i=4 coord=0.000000e+00
min value occurs at i=5 coord=0.000000e+00
min value occurs at i=6 coord=0.000000e+00
min value occurs at i=7 coord=0.000000e+00
min value occurs at i=8 coord=0.000000e+00
min value occurs at i=9 coord=0.000000e+00
min value occurs at i=10 coord=0.000000e+00
min value occurs at i=11 coord=0.000000e+00
min value occurs at i=12 coord=0.000000e+00
min value occurs at i=13 coord=0.000000e+00
min value occurs at i=14 coord=0.000000e+00
min value occurs at i=15 coord=0.000000e+00
min value occurs at i=16 coord=0.000000e+00
min value occurs at i=17 coord=0.000000e+00
min value occurs at i=18 coord=0.000000e+00
min value occurs at i=19 coord=0.000000e+00
min value occurs at i=20 coord=0.000000e+00
min value occurs at i=21 coord=0.000000e+00

tEnd =  (seconds)
4.546875000000000

Example 1 (p=100): This Method

```
ranknum=17 numApp=0 finalRank=17
min value =0.000000e+00
min value occurs at i=1 coord=0.000000e+00
min value occurs at i=2 coord=0.000000e+00
min value occurs at i=3 coord=0.000000e+00
min value occurs at i=4 coord=0.000000e+00
min value occurs at i=5 coord=0.000000e+00
min value occurs at i=6 coord=0.000000e+00
min value occurs at i=7 coord=0.000000e+00
min value occurs at i=8 coord=0.000000e+00
min value occurs at i=9 coord=0.000000e+00
min value occurs at i=10 coord=0.000000e+00
min value occurs at i=11 coord=0.000000e+00
min value occurs at i=12 coord=0.000000e+00
min value occurs at i=13 coord=0.000000e+00
min value occurs at i=14 coord=0.000000e+00
min value occurs at i=15 coord=0.000000e+00
min value occurs at i=16 coord=0.000000e+00
min value occurs at i=17 coord=0.000000e+00
min value occurs at i=18 coord=0.000000e+00
min value occurs at i=19 coord=0.000000e+00
min value occurs at i=20 coord=0.000000e+00
min value occurs at i=21 coord=0.000000e+00
min value occurs at i=22 coord=0.000000e+00
min value occurs at i=23 coord=0.000000e+00
min value occurs at i=24 coord=0.000000e+00
min value occurs at i=25 coord=0.000000e+00
min value occurs at i=26 coord=0.000000e+00
min value occurs at i=27 coord=0.000000e+00
min value occurs at i=28 coord=0.000000e+00
min value occurs at i=29 coord=0.000000e+00
min value occurs at i=30 coord=0.000000e+00
min value occurs at i=31 coord=0.000000e+00
min value occurs at i=32 coord=0.000000e+00
min value occurs at i=33 coord=0.000000e+00
min value occurs at i=34 coord=0.000000e+00
min value occurs at i=35 coord=0.000000e+00
min value occurs at i=36 coord=0.000000e+00
min value occurs at i=37 coord=0.000000e+00
min value occurs at i=38 coord=0.000000e+00
min value occurs at i=39 coord=0.000000e+00
min value occurs at i=40 coord=0.000000e+00
min value occurs at i=41 coord=0.000000e+00
min value occurs at i=42 coord=0.000000e+00
min value occurs at i=43 coord=0.000000e+00
min value occurs at i=44 coord=0.000000e+00
min value occurs at i=45 coord=0.000000e+00
min value occurs at i=46 coord=0.000000e+00
min value occurs at i=47 coord=0.000000e+00
min value occurs at i=48 coord=0.000000e+00
min value occurs at i=49 coord=0.000000e+00
min value occurs at i=50 coord=0.000000e+
```

```
00
min value occurs at i=51 coord=0.000000e+00
min value occurs at i=52 coord=0.000000e+00
min value occurs at i=53 coord=0.000000e+00
min value occurs at i=54 coord=0.000000e+00
min value occurs at i=55 coord=0.000000e+00
min value occurs at i=56 coord=0.000000e+00
min value occurs at i=57 coord=0.000000e+00
min value occurs at i=58 coord=0.000000e+00
min value occurs at i=59 coord=0.000000e+00
min value occurs at i=60 coord=0.000000e+00
min value occurs at i=61 coord=0.000000e+00
min value occurs at i=62 coord=0.000000e+00
min value occurs at i=63 coord=0.000000e+00
min value occurs at i=64 coord=0.000000e+00
min value occurs at i=65 coord=0.000000e+00
min value occurs at i=66 coord=0.000000e+00
min value occurs at i=67 coord=0.000000e+00
min value occurs at i=68 coord=0.000000e+00
min value occurs at i=69 coord=0.000000e+00
min value occurs at i=70 coord=0.000000e+00
min value occurs at i=71 coord=0.000000e+00
min value occurs at i=72 coord=0.000000e+00
min value occurs at i=73 coord=0.000000e+00
min value occurs at i=74 coord=0.000000e+00
min value occurs at i=75 coord=0.000000e+00
min value occurs at i=76 coord=0.000000e+00
min value occurs at i=77 coord=0.000000e+00
min value occurs at i=78 coord=0.000000e+00
min value occurs at i=79 coord=0.000000e+00
min value occurs at i=80 coord=0.000000e+00
min value occurs at i=81 coord=0.000000e+00
min value occurs at i=82 coord=0.000000e+00
min value occurs at i=83 coord=0.000000e+00
min value occurs at i=84 coord=0.000000e+00
min value occurs at i=85 coord=0.000000e+00
min value occurs at i=86 coord=0.000000e+00
min value occurs at i=87 coord=0.000000e+00
min value occurs at i=88 coord=0.000000e+00
min value occurs at i=89 coord=0.000000e+00
min value occurs at i=90 coord=0.000000e+00
min value occurs at i=91 coord=0.000000e+00
min value occurs at i=92 coord=0.000000e+00
min value occurs at i=93 coord=0.000000e+00
min value occurs at i=94 coord=0.000000e+00
min value occurs at i=95 coord=0.000000e+00
min value occurs at i=96 coord=0.000000e+00
min value occurs at i=97 coord=0.000000e+00
min value occurs at i=98 coord=0.000000e+00
min value occurs at i=99 coord=0.000000e+00
min value occurs at i=100 coord=0.000000e+00
```

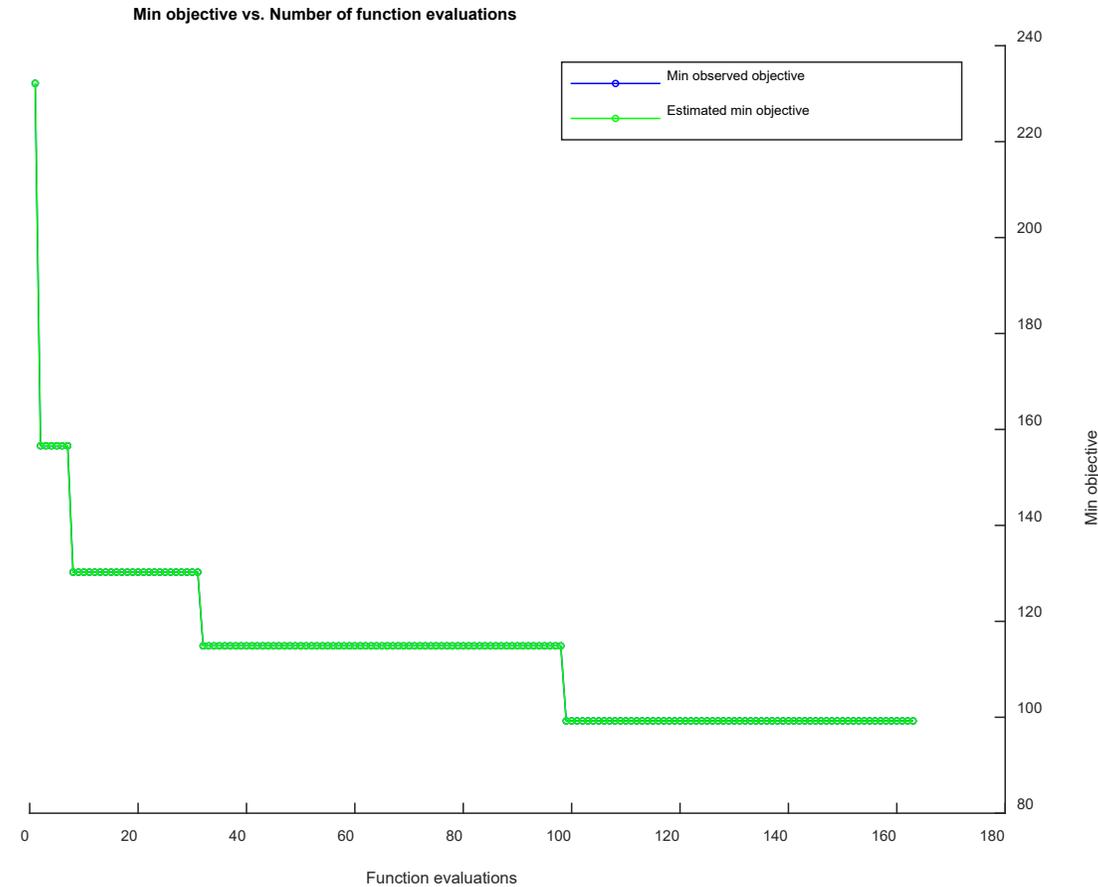
tEnd =

2.0625



(seconds)

Example 1 ($p=100$): BO Method



Time: 20,437 s

HD Rastrigin: BO, 100 parameters

Iter	Eval result	Objective	Objective runtime	BestSoFar (observed)	BestSoFar (estim.)	x1	x2	
161	Accept	201.02	0.0029775	99.242	99.242	-1.8806	2.4469	
162	Accept	147.28	0.002952	99.242	99.242	-1.7034	-0.78204	
163	Accept	105.27	0.0024795	99.242	99.242	-0.89105	-2.9409	
		x3	x4	x5	x6	x7	x8	x9
		-2.3406	2.1018	3.6423	0.59926	-0.38798	-2.6194	-3.3586
		-3.1055	-0.69451	1.4142	-3.5915	0.64184	-3.0901	0.35911
		-2.3201	-0.85062	2.0734	-0.44491	-1.909	-0.60215	-1.2276
		x94	x95	x96	x97	x98	x99	x100
		0.47819	4.8244	0.85113	-4.3041	3.4342	-0.94505	0.10861
		0.59073	1.0257	2.1115	-2.1486	4.8156	3.8577	-4.1579
		0.68268	4.2235	1.9016	4.3682	0.097302	0.23375	-4.8345

HD Rastrigin: GA, 100 parameters

Parameters of the best solution : [6.00000000e-02 -9.70000000e-01 -2.28000000e+00 1.94000000e+00

-9.80000000e-01 -4.00000000e-02 9.40000000e-01 -9.90000000e-01
-9.50000000e-01 1.00000000e-02 -2.03000000e+00 -1.04000000e+00
2.00000000e+00 -1.97000000e+00 1.76000000e+00 -8.60000000e-01
-2.00000000e-02 -3.92000000e+00 -6.00000000e-02 -4.00000000e-02
2.00000000e+00 -1.04000000e+00 -1.91000000e+00 9.50000000e-01
-8.70000000e-01 -8.70000000e-01 9.10000000e-01 9.40000000e-01
2.11000000e+00 7.60000000e-01 -1.10000000e-01 -1.04000000e+00
-2.20000000e-01 -9.00000000e-02 -1.95000000e+00 -2.20000000e-01
-2.87000000e+00 -1.03000000e+00 -1.96000000e+00 -1.40000000e-01
-9.00000000e-02 4.00000000e-02 -1.93000000e+00 9.20000000e-01
1.14000000e+00 -1.07000000e+00 8.00000000e-02 -1.83000000e+00
1.07000000e+00 -8.50000000e-01 -1.92000000e+00 -2.00000000e-02
-4.00000000e-02 1.50000000e-01 -9.50000000e-01 -2.93000000e+00
-1.87000000e+00 1.98000000e+00 -1.06000000e+00 8.20000000e-01
-1.00000000e+00 -1.06000000e+00 -4.00000000e-02 -1.00000000e-02
1.92000000e+00 2.16000000e+00 -8.00000000e-02 -1.99000000e+00
-4.14000000e+00 9.50000000e-01 -1.00000000e-02 1.93000000e+00
-1.05000000e+00 -1.03000000e+00 6.00000000e-02 4.04000000e+00
1.00000000e-02 -1.02000000e+00 -1.17000000e+00 4.00000000e-02
9.40000000e-01 1.02000000e+00 1.02000000e+00 -1.98000000e+00
-1.92000000e+00 -1.22000000e+00 -1.09245946e-13 2.03000000e+00
-3.07000000e+00 -9.70000000e-01 2.10000000e+00 1.08000000e+00
-2.04000000e+00 -9.40000000e-01 1.28000000e+00 -1.08000000e+00
-2.03000000e+00 1.16000000e+00 -1.02000000e+00 -4.00000000e-02]

Fitness value of the best solution = 0.002539938683377189

Predicted output based on the best solution : 393.71029172655693

Runtime: 1184.060890539

Used 100k generations

Comparison on the same Python platform

BO, GA, This

- 10 Parameter case
- Theoretical minimum: 0 at all 0

BO:

No. of optimization run: 480
New results are: 93.38890075683594
Minimum f(x): 93.38890075683594 at x: (-
4.050782203674316, 2.354004383087158, -
1.0819282531738281, -
1.0227537155151367, -
0.8088850975036621,
1.7891125679016113, -
1.9831023216247559, -
2.1536715030670166, -
1.892641305923462, 2.800482749938965).
Runtime: **6933.030862368**

GA:

Generation = 146 - Best result = 0.0
Generation = 147 - Best result = 0.0
Generation = 148 - Best result = 0.0
Generation = 149 - Best result = 0.0
Generation = 150 - Best result = 0.0
Minimum f(x): 0.0 at x: (-1.092459456231154e-
13, -1.092459456231154e-13, -
1.092459456231154e-13, -
1.092459456231154e-13, -
1.092459456231154e-13, -
1.092459456231154e-13, -
1.092459456231154e-13, -
1.092459456231154e-13, -
1.092459456231154e-13). Runtime:
61.28406199599999

This:

Maximum: [402.94241783]
Minmum: [0.] Runtime:
1.176050676
Coordinates of Minimum Point:
[[0.]
[0.]
[0.]
[0.]
[0.]
[0.]
[0.]
[0.]
[0.]
[0.]

HD Rastrigin: This method, 200 and 400 parameters

- $p=200, n=51$
- $p=400, n=51$

Time: 25 s, and 48 s respectively

Example 2

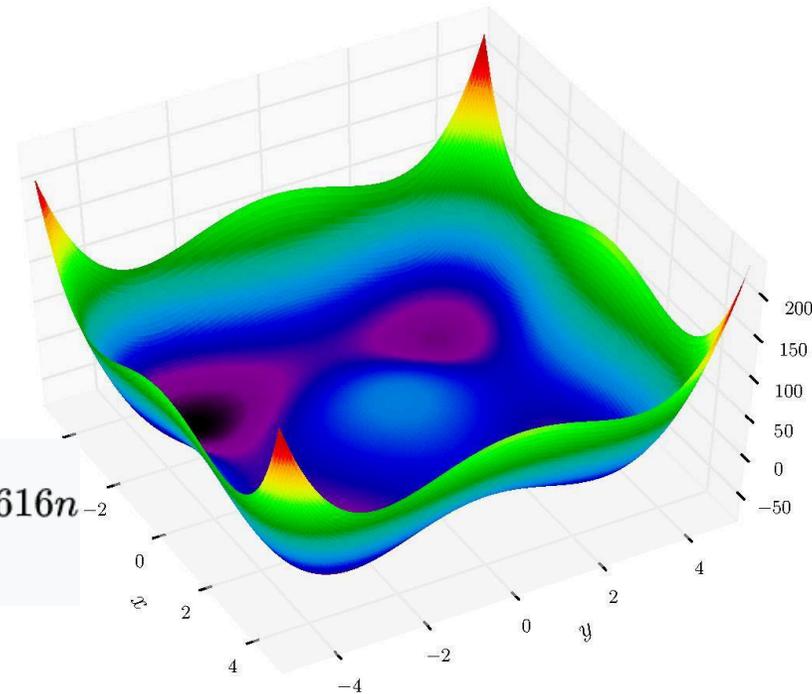
- Styblinski–Tang function

$$f(\{x\}) = \frac{\sum_{i=1}^n x_i^4 - 16x_i^2 + 5x_i}{2} \quad n = 2 \text{ case}$$

$$x_i \in (-5, 5)$$

Theoretical minimum:

$$-39.16617n < \underbrace{f(-2.903534, \dots, -2.903534)}_{n \text{ times}} < -39.16616n^{-2}$$



Example 2

- Styblinski-Tang function

$$f(\{x\}) = \frac{\sum_{i=1}^n x_i^4 - 16x_i^2 + 5x_i}{2}$$

Tested Case:

$$x_i \in (-5, 5)$$

$$n = 11, j = 101$$

101^{11} choices

ranknum=73 numApp=101 finalRank=73

min value = -4.308255e+02

min value occurs at i=1 coord=-2.900000e+00

min value occurs at i=2 coord=-2.900000e+00

min value occurs at i=3 coord=-2.900000e+00

min value occurs at i=4 coord=-2.900000e+00

min value occurs at i=5 coord=-2.900000e+00

min value occurs at i=6 coord=-2.900000e+00

min value occurs at i=7 coord=-2.900000e+00

min value occurs at i=8 coord=-2.900000e+00

min value occurs at i=9 coord=-2.900000e+00

min value occurs at i=10 coord=-2.900000e+00

min value occurs at i=11 coord=-2.900000e+00

tEnd =

(seconds)

5.3593750000000000



Example 3:

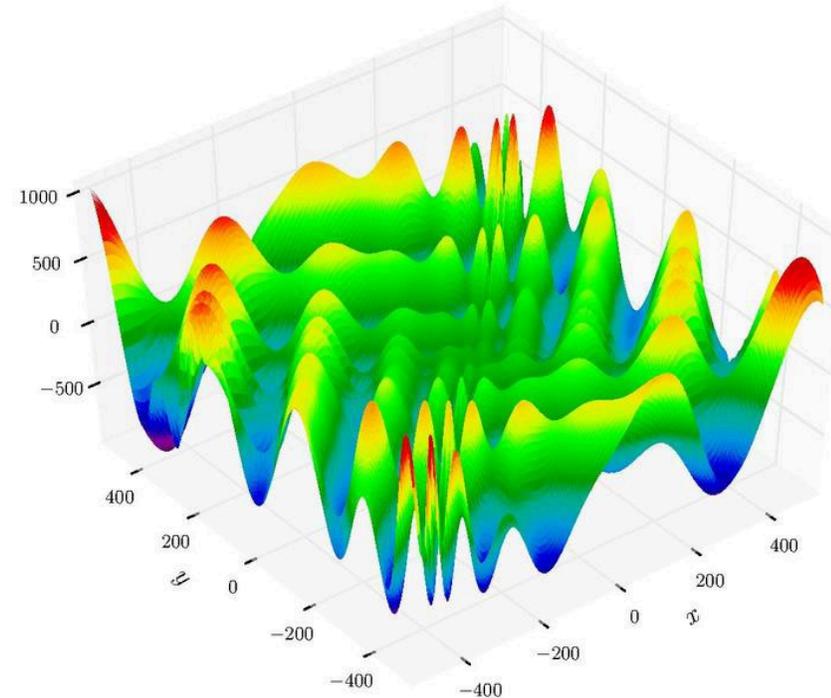
- Egg-holder function

$$f(x, y) = -(y + 47) \sin \sqrt{\left| \frac{x}{2} + (y + 47) \right|} - x \sin \sqrt{|x - (y + 47)|}$$

$$x_i \in (-512, 512)$$

Theoretical minimum:

$$f(512, 404.2319) = -959.6407$$



Optimization completed.
MaxObjectiveEvaluations of 200 reached.
Total function evaluations: 200
Total elapsed time: 254.8716 seconds.
Total objective function evaluation time: 0.40958

Best observed feasible point:

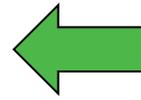
xp	yp
348.139674212863	499.908249451638

Observed objective function value = -888.8498
Estimated objective function value = -888.8498
Function evaluation time = 0.0011695

Best estimated feasible point (according to models):

xp	yp
348.139674212863	499.908249451638

Estimated objective function value = -888.8498
Estimated function evaluation time = 0.00122



BO Results

- in 200 steps still far from minimal value

Example 3 (This Method)

- Egg-holder function

$$f(x, y) = -(y + 47) \sin \sqrt{\left| \frac{x}{2} + (y + 47) \right|} - x \sin \sqrt{|x - (y + 47)|}$$

$$x_i \in (-512, 512)$$

Tested Case:

$$n = 2, j = 201$$

ranknum=200 numApp=201 finalRank=201

min value =-9.595705e+02

min value occurs at i=1 coord=5.120000e+02

min value occurs at i=2 coord=4.044800e+02

tEnd =

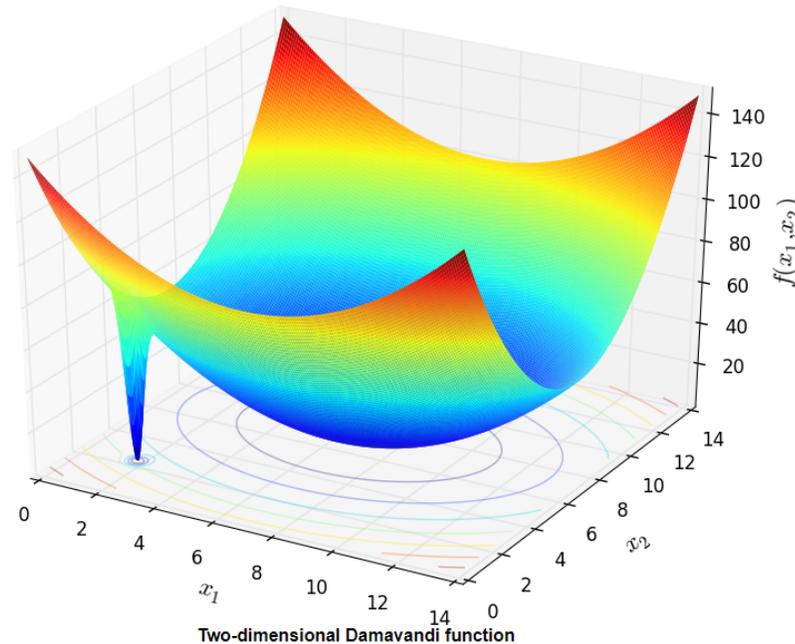
5.4688 (s)

Example 4: Damavandi

This class defines the Damavandi global optimization problem. This is a multimodal minimization problem defined as follows:

$$f_{\text{Damavandi}}(\mathbf{x}) = \left[1 - \left| \frac{\sin[\pi(x_1 - 2)] \sin[\pi(x_2 - 2)]}{\pi^2(x_1 - 2)(x_2 - 2)} \right|^5 \right] [2 + (x_1 - 7)^2 + 2(x_2 - 7)^2]$$

Here, n represents the number of dimensions and $x_i \in [0, 14]$ for $i = 1, \dots, n$.



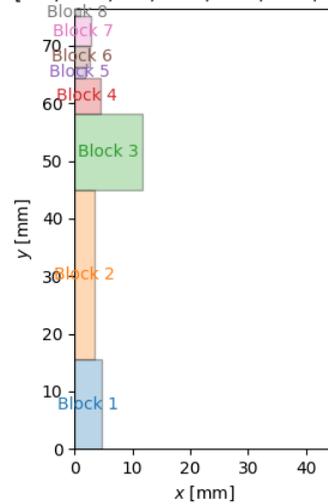
New_Opt_Damavandi
ranknum=362 numApp=401 finalRank=362
min value =3.172043e-02
min value occurs at i=1
coord=1.995000e+00
min value occurs at i=2
coord=1.995000e+00

tEnd =

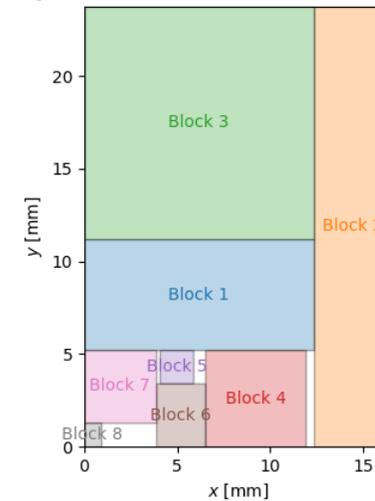
64.7344 s

Application: Floorplan and Placement

$r = [0.3, 0.12, 0.9, 0.69, 0.94, 0.72, 0.54, 0.69]$



$r = [2.07, 0.18, 0.98, 1.04, 1.03, 0.77, 0.99, 0.69]$

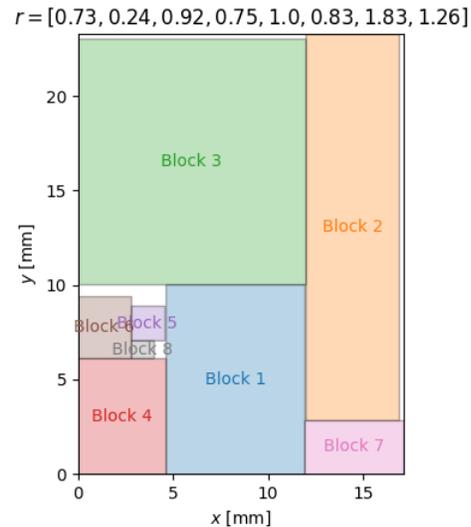


Application: Floorplan and Placement

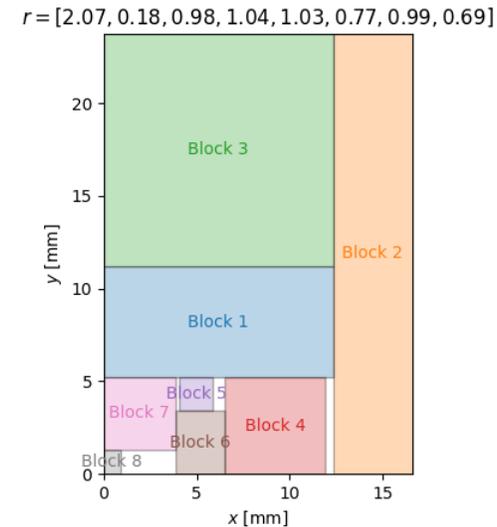
- Floor Plan Optimization

	Bounding box area [mm ²]
Benchmark	399.8574029187914
Floor plan & ratio optimized	394.6475438820303

Benchmark



Floor plan/Aspect Ratio optimized



Summary