Non-uniform covering method for global optmization and approximation

Yu. G. Evtushenko, M.A. Posypkin

Dorodnicyn Computing Centre, FRC CSC RAS

«Optimization at Work», 14 April 2018

МФТИ

Non-Uniform Covering Method

1. Initially proposed for global optimization problems in 1971
   Ю. Г. Евтушенко, “Численный метод поиска глобального экстремума функций (перебор на неравномерной сетке)”, Ж. вычисл. матем. и матем. физ., 11:6 (1971), 1390–1403

2. Extened to multiobjective optimization in 1986
   Евтушенко Ю. Г., Потапов М. А. «Методы численного решения многокритериальных задач.» ДАН СССР, Т. 291, N 1, 1986, С. 25-39.

3. Extended to mathematical programming problems in 1992
   Yu. G. Evtushenko, M. A. Potapov, V. V. Korotkikh.
   Numerical methods for global optimization.
   In "Recent advances in global optimization”, Princeton University Press, pp. 274-297. 1992.

4. Parallel implementation 2007
   Ю. Г. Евтушенко, В. У. Малкова, А. А. Станевичюс. Распараллеливание процесса поиска глобального экстремума. Автоматика и телемеханика, № 5. С. 46-58, 2007.

Non-Uniform Covering Method

1. Advanced techniques for mathematical programming problems 2011
   Evtushenko Y., Posypkin M. A deterministic approach to global box-constrained optimization //Optimization Letters. – 2013. – Т. 7. – №. 4. – С. 819-829.
2. Advanced techniques for multiobjective optimization
   Evtushenko Y. G., Posypkin M. A. A deterministic algorithm for global multi-objective optimization //Optimization Methods and Software. – 2014. – Т. 29. – №. 5. – С. 1005-1019.
3. High-performance computing
   Evtushenko Y., Posypkin M., Sigal I. A framework for parallel large-scale global optimization //Computer Science-Research and Development. – 2009. – Т. 23. – №. 3-4. – С. 211-215.

   Evtushenko, Y., Golubeva, Y., Orlov, Y., Posypkin, M.Using simulation for performance analysis and visualization of parallel Branch-and-Bound methods //Russian Supercomputing Days. – Springer, Cham, 2016. – С. 356-368.
4. Approximation
   Evtushenko Y. G., Posypkin M. A. Effective hull of a set and its approximation //Doklady Mathematics. – Pleiades Publishing, 2014. – Т. 90. – №. 3. – С. 791-794.

   Evtushenko, Y., Posypkin, M., Rybak, L., Turkin, A.. Approximating a solution set of nonlinear inequalities //Journal of Global Optimization. – 2017. – С. 1-17.

GLOBAL OPTIMIZATION PROBLEM STATEMENT

$$f_* = f(x_*) = glob \min_{x \in X} f(x)$$

Solution set $\qquad X_* = \{x_* \in X : f(x_*) = f_*\}$

ε- solution set $\qquad X_\varepsilon = \{x_\varepsilon \in X : f_* \le f(x_\varepsilon) \le f_* + \varepsilon\}$

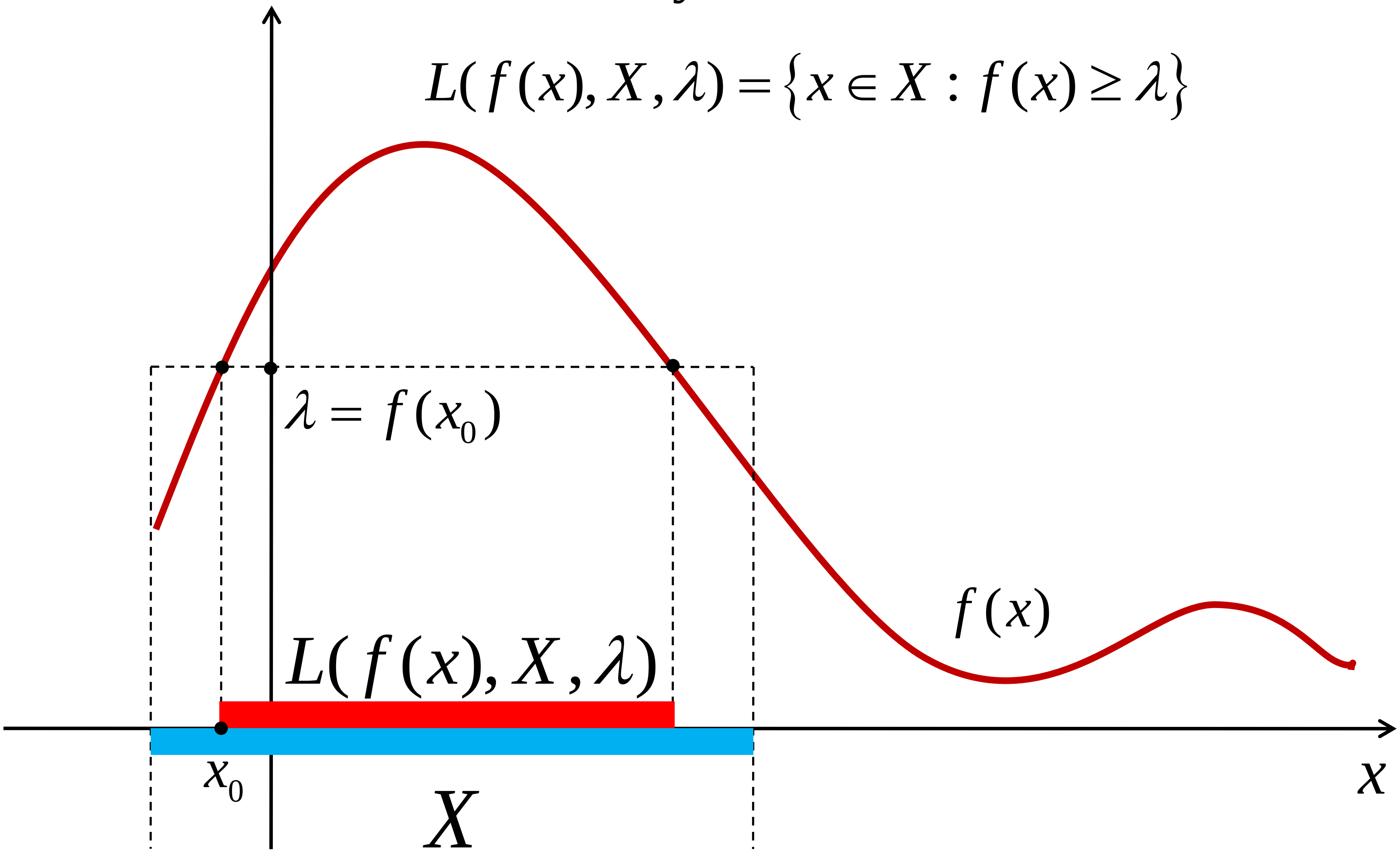
$$X_* \subseteq X_\varepsilon \subseteq X$$

$$\boxed{f_* \ge f(x_\varepsilon) - \varepsilon}$$

LEBESGUE SET

For arbitrary $\lambda \in R,\ X \subseteq R^n$ define

$$L(f(x), X, \lambda) = \{x \in X : f(x) \geq \lambda\}$$



RECORD POINTS AND VALUES

$$x_1, x_2, \ldots, x_i \in X$$

$$f(x_1), f(x_2), \ldots, f(x_i)$$

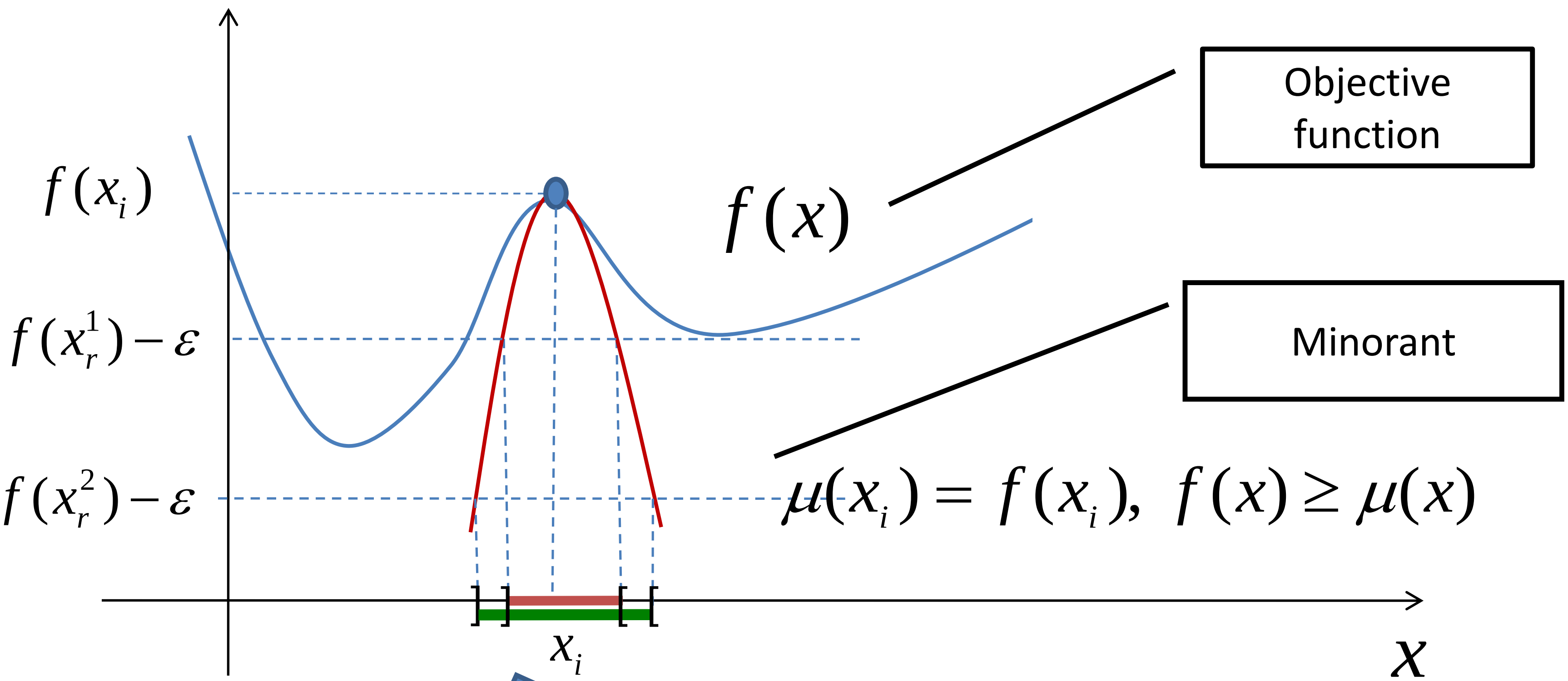
$$r_{i+1} = \min(r_i, f(x_{i+1}))$$

Current record value r_i

Current record point $x_j : f(x_j) = r_i$

$$1 \le j \le i$$

LOWER BOUNDS



$$\mu(x_i) = f(x_i),\ f(x) \geq \mu(x)$$

$$f(x_r^1) > f(x_r^2)$$

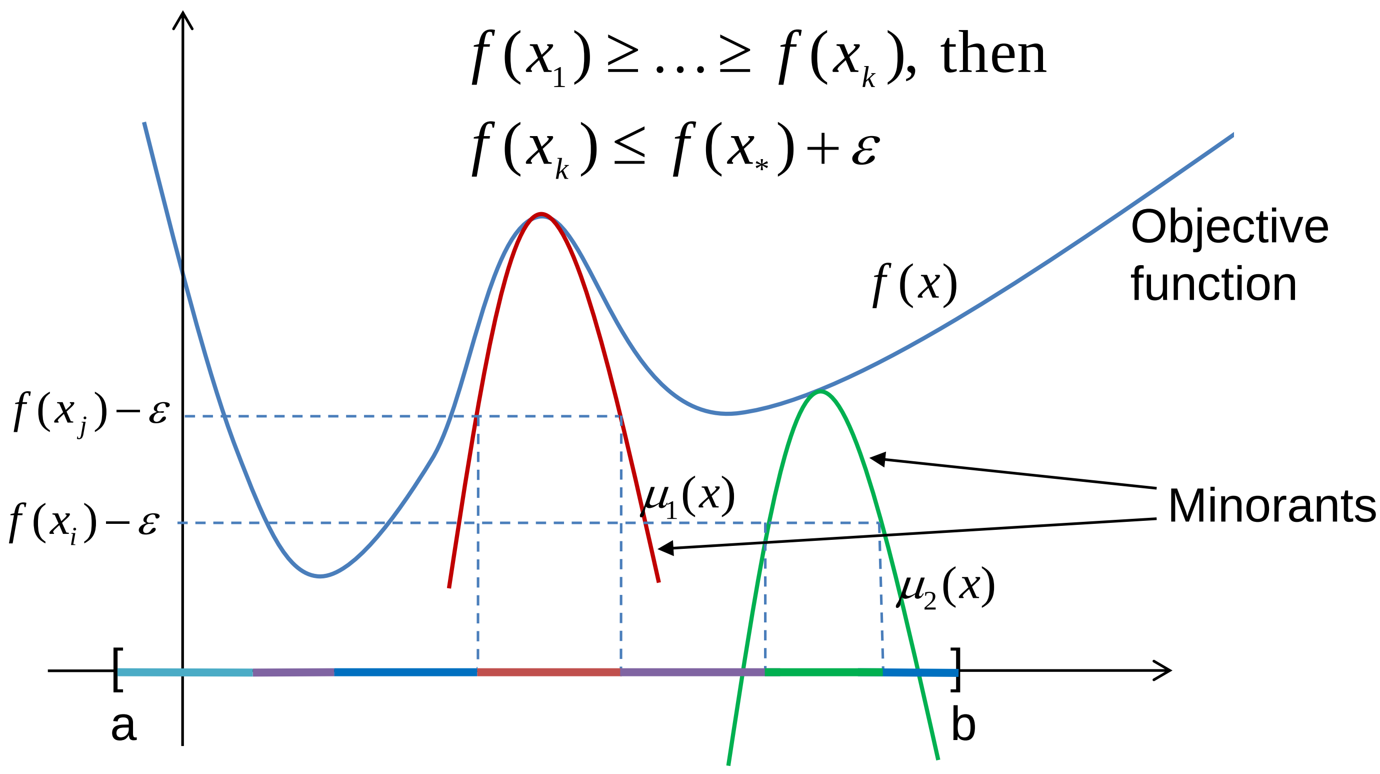
NON-UNIFORM COVERING METHOD

$$\text{If } X = \bigcup_{i=1}^{k} L(\mu_i(\cdot), X_i, f(x_i) - \varepsilon),$$

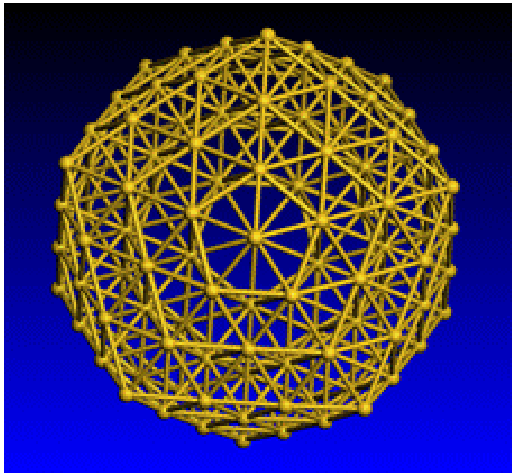
$$\text{where } X_i \subseteq X,\ i = 1,\dots,k$$

$$f(x_1) \geq \ldots \geq f(x_k), \text{ then}$$

$$f(x_k) \leq f(x_*) + \varepsilon$$



$$L(\mu(\cdot), A, \alpha) = \{x \in A : f(x) \geq a\}$$



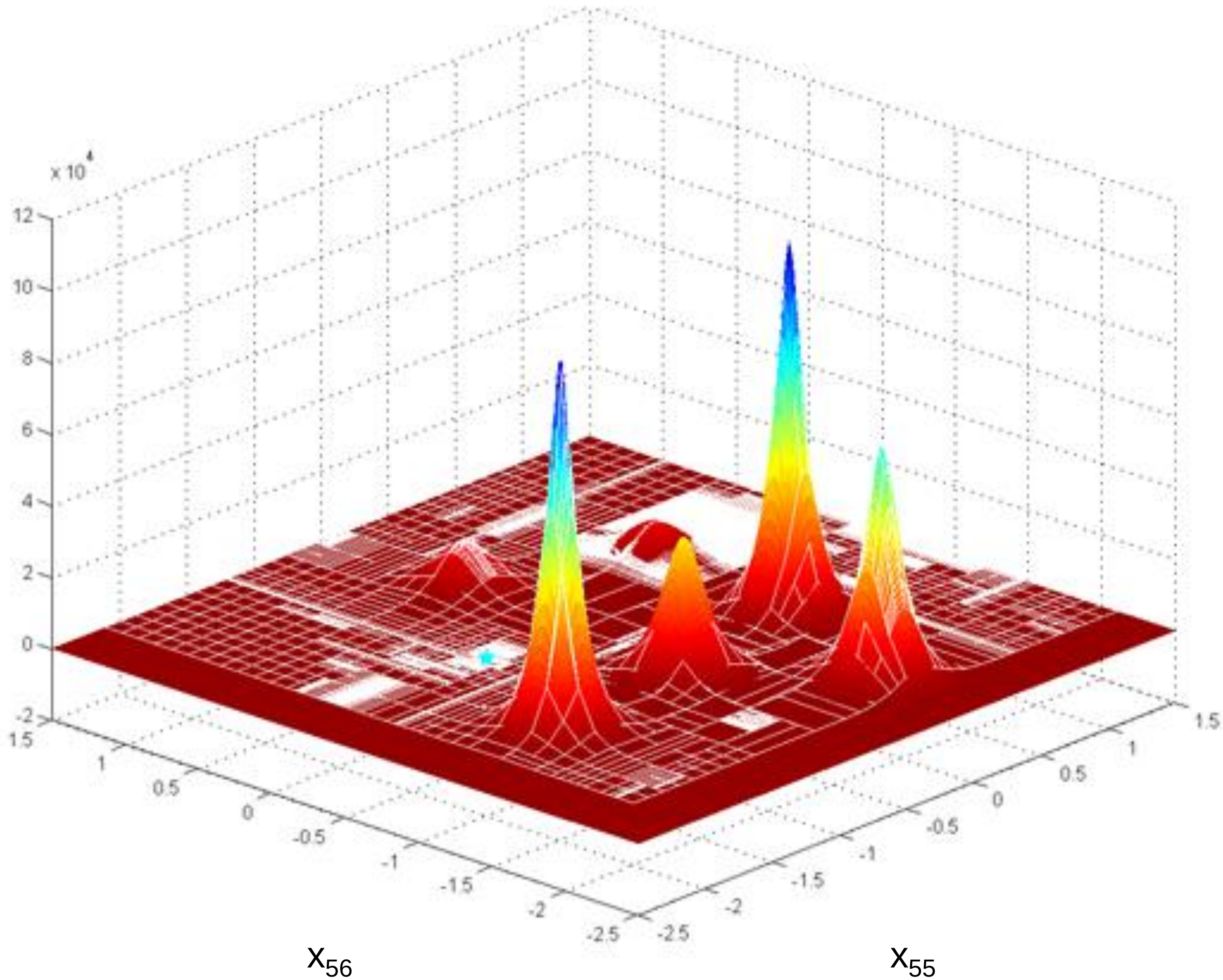
GLOBAL OPTIMIZATION PROBLEM EXAMPLE

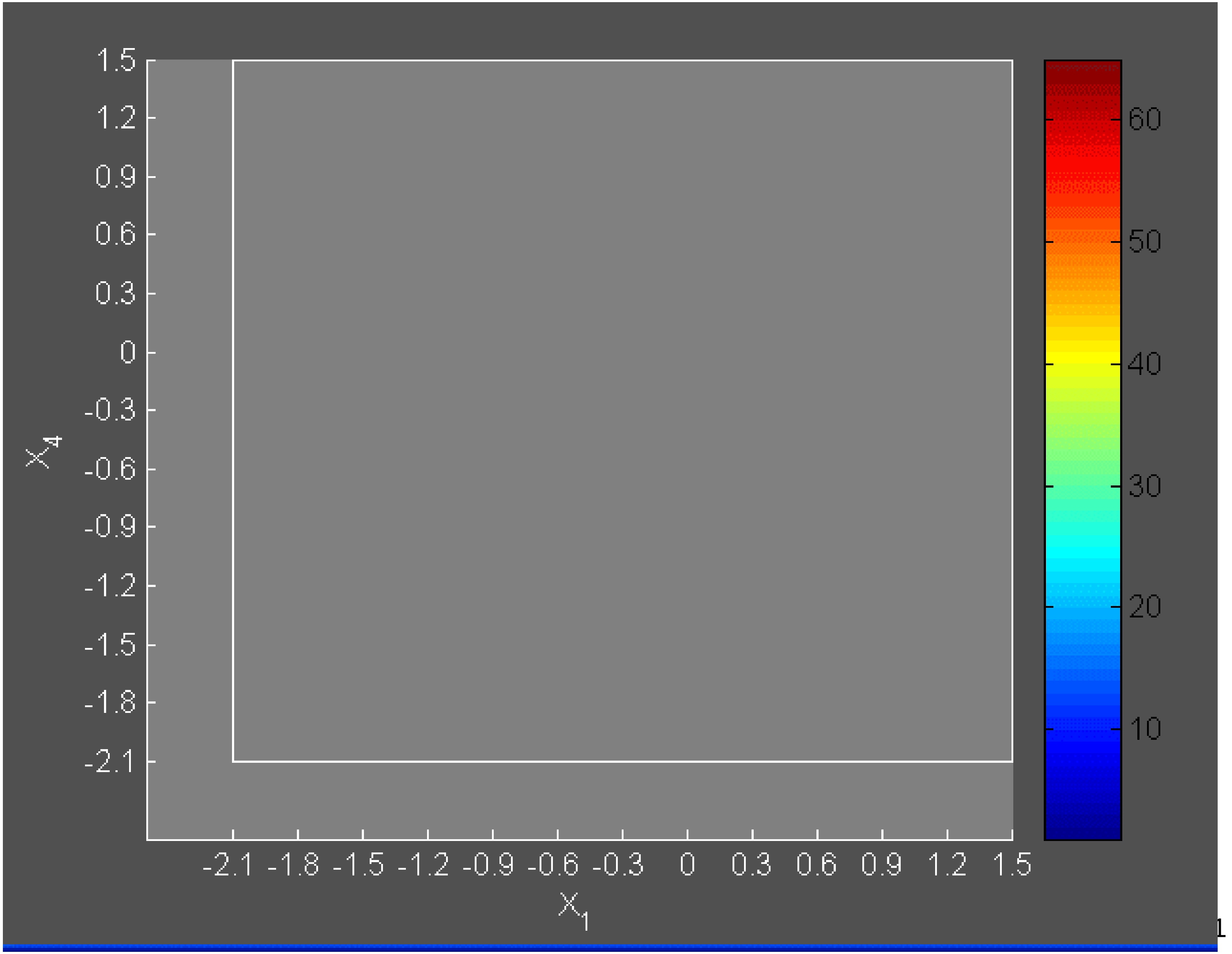
Morse atomic cluster energy function

$$F(\rho)=\sum_{i=1}^{n}\sum_{j=i+1}^{n}\left(\left(e^{\rho\left(1-\left\|X_i-X_j\right\|\right)}-1\right)^2-1\right)$$

***x**$_i$* и ***x**$_j$* coordinates of atoms

n - number of atoms


x 10^4
12
10
8
6
4
2
0
-2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
x_{56}
x_{55}



1.5
1.2
0.9
0.6
0.3
0
-0.3
-0.6
-0.9
-1.2
-1.5
-1.8
-2.1
X_4
-2.1 -1.8 -1.5 -1.2 -0.9 -0.6 -0.3 0 0.3 0.6 0.9 1.2 1.5
X_1
60
50
40
30
20
10

Efficient estimating of functions

Lets Hessian spectrum X_i is enclosed in $[k_i, K_i]$, then

$$f(x) \geq f(x_i) + \langle f_x(x_i), x - x_i \rangle + \frac{k_i}{2}\|x - x_i\|^2$$

$$f(x) \leq f(x_i) + \langle f_x(x_i), x - x_i \rangle + \frac{K_i}{2}\|x - x_i\|^2$$

$$\sigma(H) \subseteq [k_i, K_i]$$

$$-L_i \leq k_i \leq K_i \leq L_i$$

Spectrum bounding

$$k_i \geq \min_{i=1,..,n}\left(\underline{u_{ii}} - \sum_{j=1, j\neq i}^{n} v_{ij}\right), K_i \leq \max_{i=1,..,n}\left(\overline{u_{ii}} + \sum_{j=1, j\neq i}^{n} v_{ij}\right),$$

где

$$\underline{u_{ij}} \leq \min_{x\in X_i}\frac{\partial f(x)}{\partial x^i \partial x^j}, \quad \max_{x\in X_i}\frac{\partial f(x)}{\partial x^i \partial x^j} \leq \overline{u_{ij}},$$

$$v_{ij} = \max(|\underline{u_{ij}}|, |\overline{u_{ij}}|)$$

Accounting optimality conditions

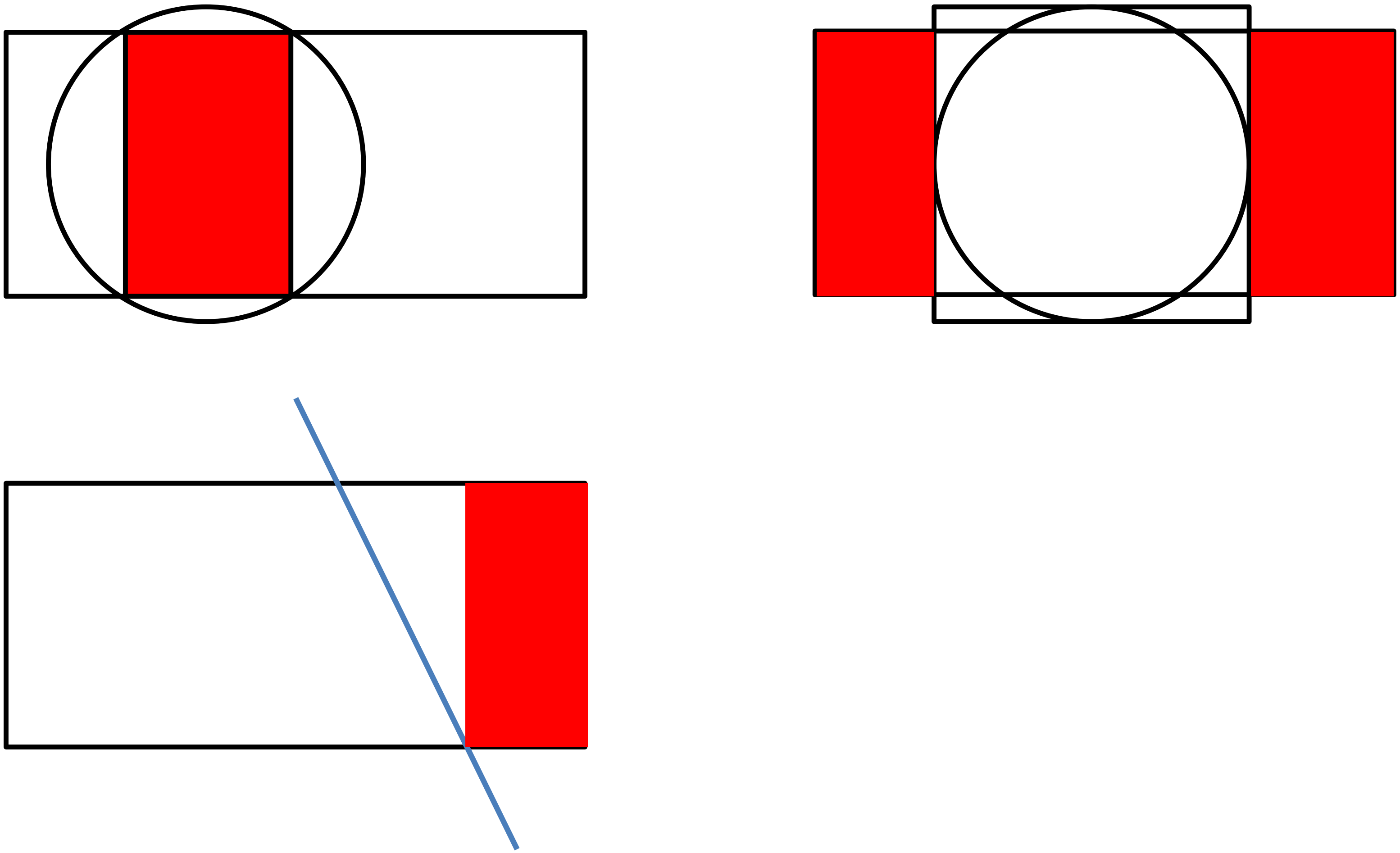
1. Underestimating on the set of optimal solutions

$$f(x) \geq \mu_i(x), x \in X_* \cap X_i$$

2. The following underestimation is valid on a set of optimal solutions

$$f(x) \geq f(x_i) - \frac{K_i}{2}\|x - x_i\|^2 \quad \text{if} \quad x \in X_* \cap X_i$$

Reducing the search space



Computational experiments

O0 $$f(x) \geq f(x_i) - l_i \|x - x_i\|$$

O1 $$f(x) \geq f(x_i) + \langle f_x(x_i), x - x_i \rangle - \frac{L_i}{2} \|x - x_i\|^2$$

O2 $$f(x) \geq f(x_i) + \langle f_x(x_i), x - x_i \rangle + \frac{k_i}{2} \|x - x_i\|^2$$

O3 $$f(x) \geq f(x_i) - \frac{K_i}{2} \|x - x_i\|^2$$

O3 + $$f(x) \geq f(x_i) - \frac{K_i}{2} \|x - x_i\|^2$$ + reduction rules

BR BARON

LG LINDOGLOBAL

Computational experiments

Series		O0	O1	O2	O3	O3+	BR	LG
1	AVR	5.399	0.73	0.49	0.08	0.05	1.07	5.42
	MAX	15.12	1.77	1.15	0.12	0.07	1.36	6.02
	MIN	0.66	0.33	0.11	0.07	0.04	0.65	3.43
2	AVR	T	11.24	8.52	0.54	0.29	4.86	E
	MAX	T	28.92	39.17	0.67	0.38	6.56	E
	MIN	T	4.03	2.09	0.46	0.24	3.68	E
3	AVR	T	T	T	2.2	1.38	A	E
	MAX	T	T	T	2.49	1.75	A	E
	MIN	T	T	T	1.9	1.15	A	E
4	AVR	T	107.31	44.15	0.94	0.54	3.51	23.63
	MAX	T	350.07	120.49	1.46	0.77	3.89	27.71
	MIN	T	22.03	10.49	0.76	0.41	3.02	20.54
5	AVR	T	T	T	11.72	5.85	A	E
	MAX	T	T	T	14.11	7.14	A	E
	MIN	T	T	T	9.93	4.7	A	E

GLOBAL MULTIOBJECTIVE OPTIMIZATION

Problem statement

$$F(x) \to min, x \in X \subseteq R^n$$

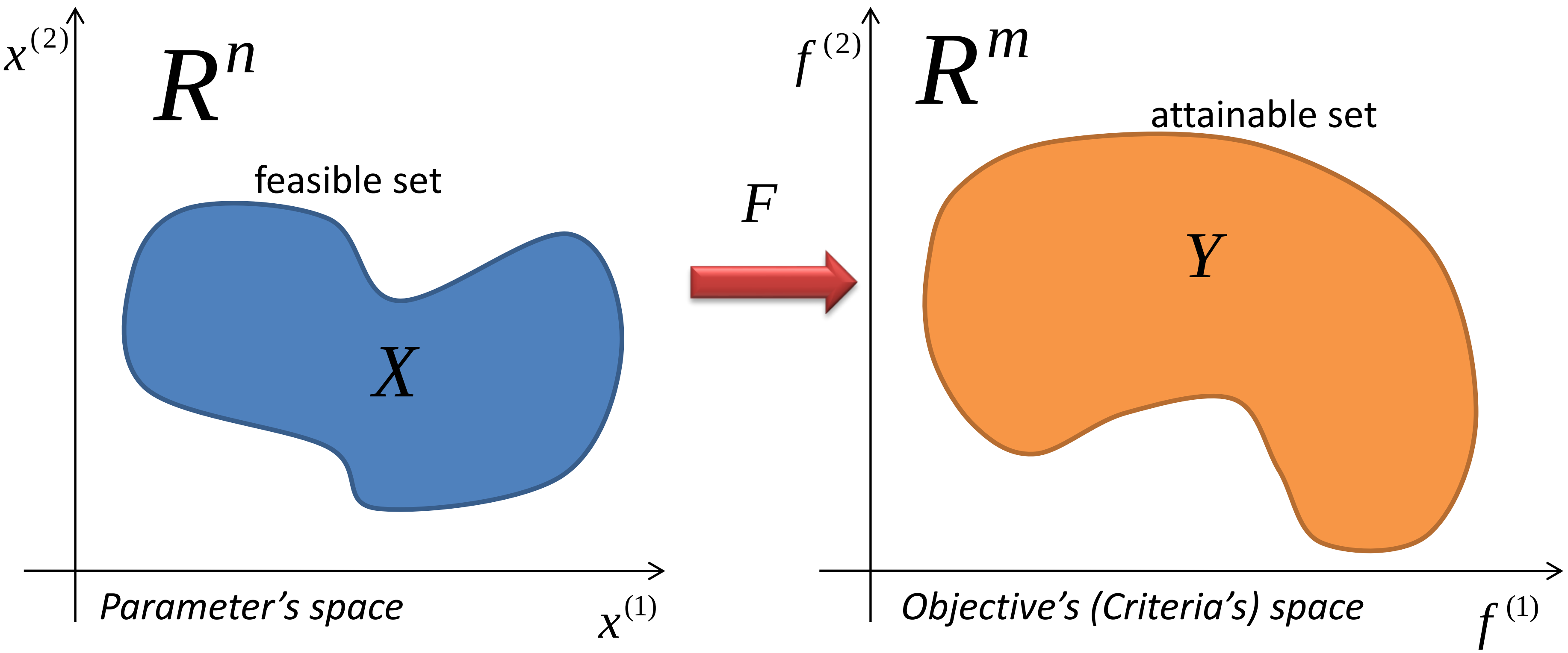
$$F\colon\ R^n \to R^m$$

$$F(x) = (f_1(x), \ldots, f_m(x))$$

$F(\cdot)$ - continuous function

X - compact set

MULTIOBJECTIVE OPTIMIZATION



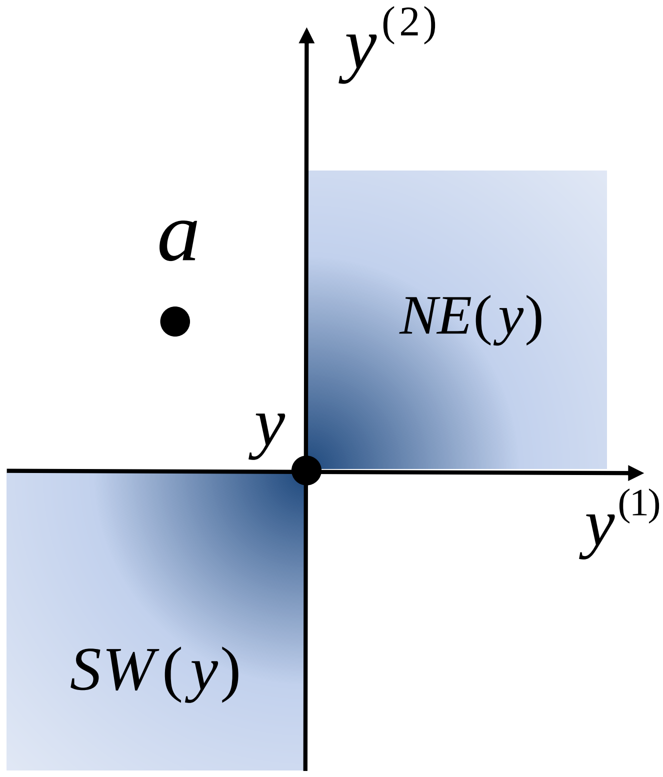
$$Y = F(X)$$

USEFUL NOTATIONS

1. $u \le y \Leftrightarrow u^{(i)} \le y^{(i)} \ \ for\, i = 1,\ldots,m$

2. $SW(y) = \left\{ u \in R^m : u \le y \right\}$

3. $NE(y) = \left\{ u \in R^m : u \ge y \right\}$

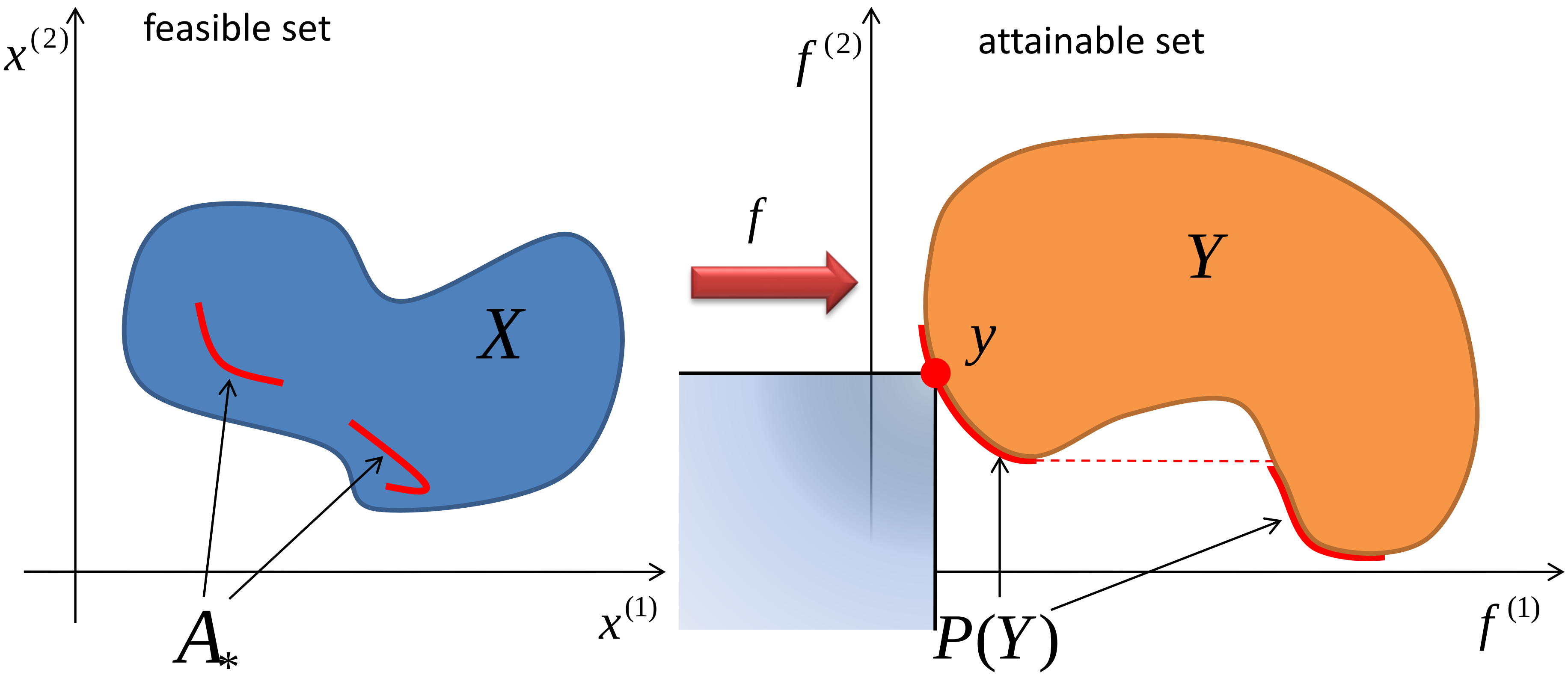


y dominates (better than) each point from **SW(y)**,

each point from **NE(y)** dominates **y**

y and **a** are not comparable

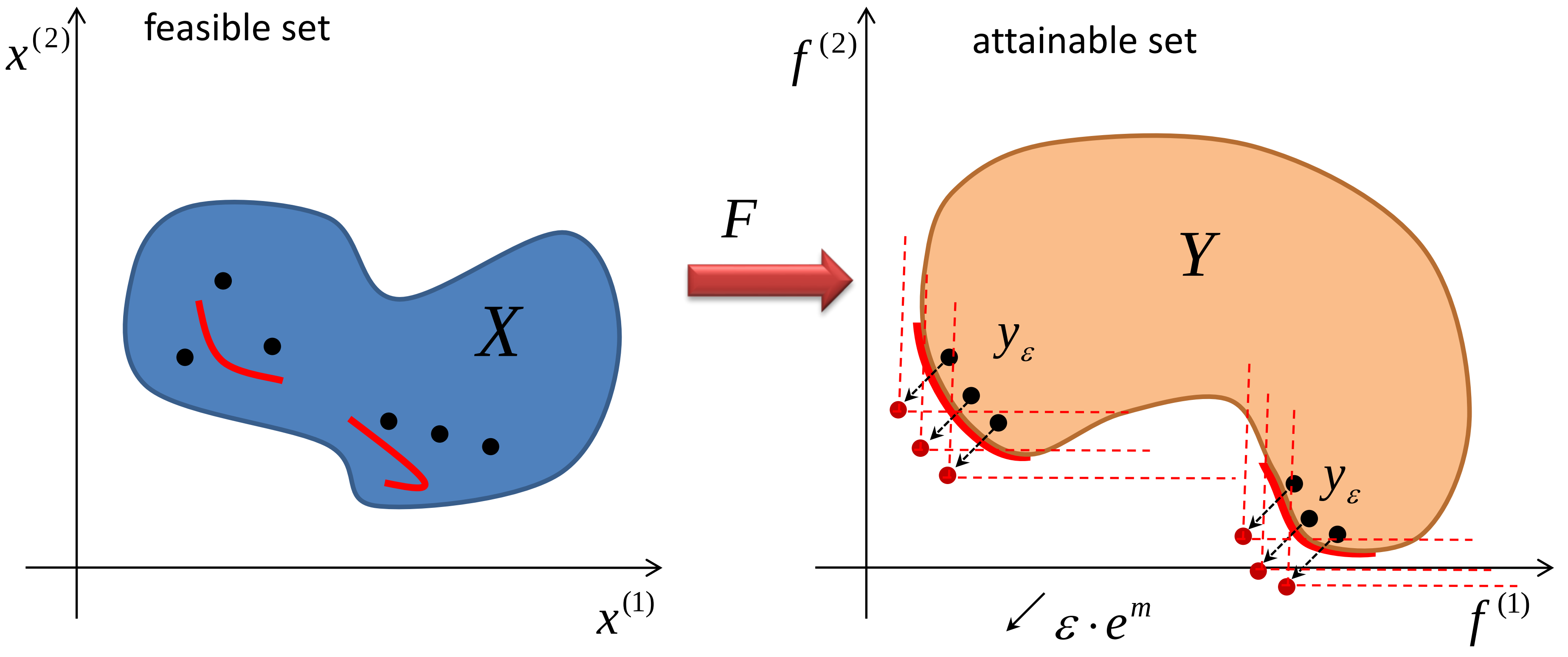
PARETO SET DEFINITION



$$P(Y) = \{y \in Y : SW(y) \cap Y = y\}$$

$$f(A_*) = P(Y)$$

ε-PARETO SET ILLUSTRATION



ε-PARETO SET DEFINITION

Y_ε is $\varepsilon-$Pareto set if

1. $Y_\varepsilon \subseteq Y$

2. $P(Y_\varepsilon) = Y_\varepsilon$

3. For each $y_* \in P(Y)$ exists $y_\varepsilon \in Y_\varepsilon$: $\boxed{y_* \geq y_\varepsilon - \varepsilon e_m}$

$A_\varepsilon - \varepsilon - solution:$

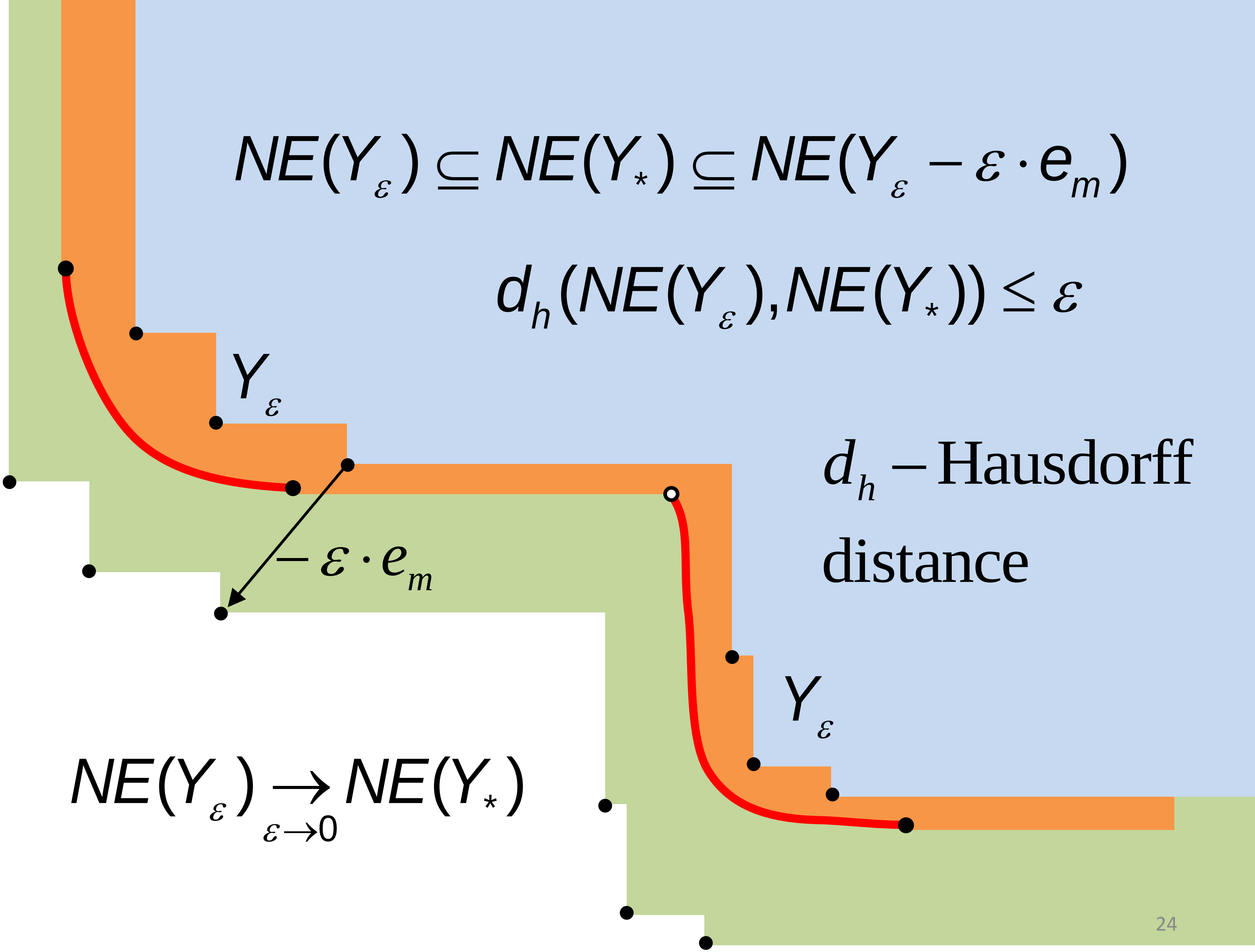
1. $A_\varepsilon \subseteq X,$

2. $f(A_\varepsilon) = Y_\varepsilon$

$$NE(Y_\varepsilon) \subseteq NE(Y_*) \subseteq NE(Y_\varepsilon - \varepsilon \cdot e_m)$$

$$d_h(NE(Y_\varepsilon), NE(Y_*)) \le \varepsilon$$

d_h – Hausdorff distance



$$NE(Y_\varepsilon) \xrightarrow[\varepsilon \to 0]{} NE(Y_*)$$

MCO WITH FUNCTIONAL CONSTRAINTS

$$F(x) \to \min,$$

$$x \in X = \{x \in R^n : G(x) \le 0\},$$

where

$$F(\cdot): R^n \to R^m, G(\cdot): R^n \to R^k$$

$$F(x) = \left(f^{(1)}(x), \ldots, f^{(m)}(x)\right), G(x) = (g^{(1)}(x), \ldots, g^{(k)}(x)).$$

$F(\cdot), G(\cdot)$ - continuous functions

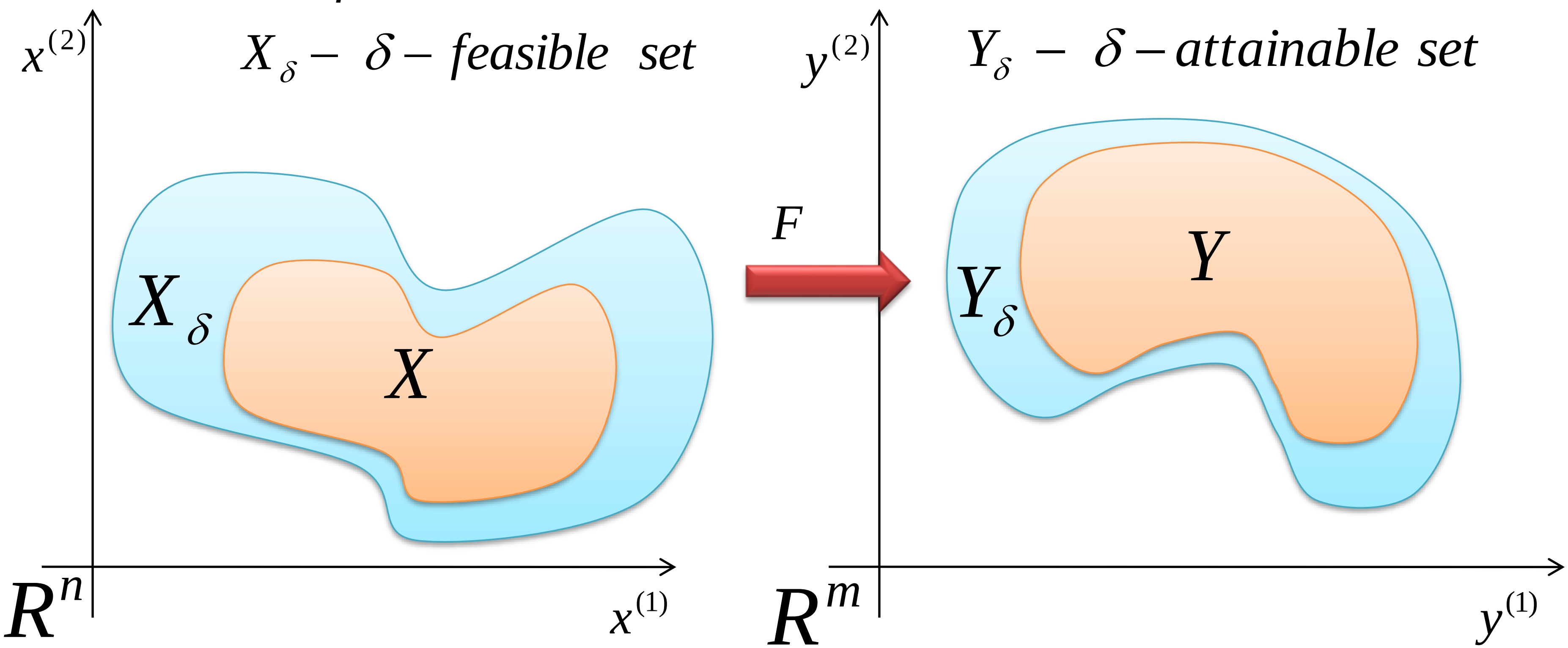
δ-feasible and δ-attainable sets

X – *feasible set*

X_δ – δ – *feasible set*

Y – *attainable set*

Y_δ – δ – *attainable set*



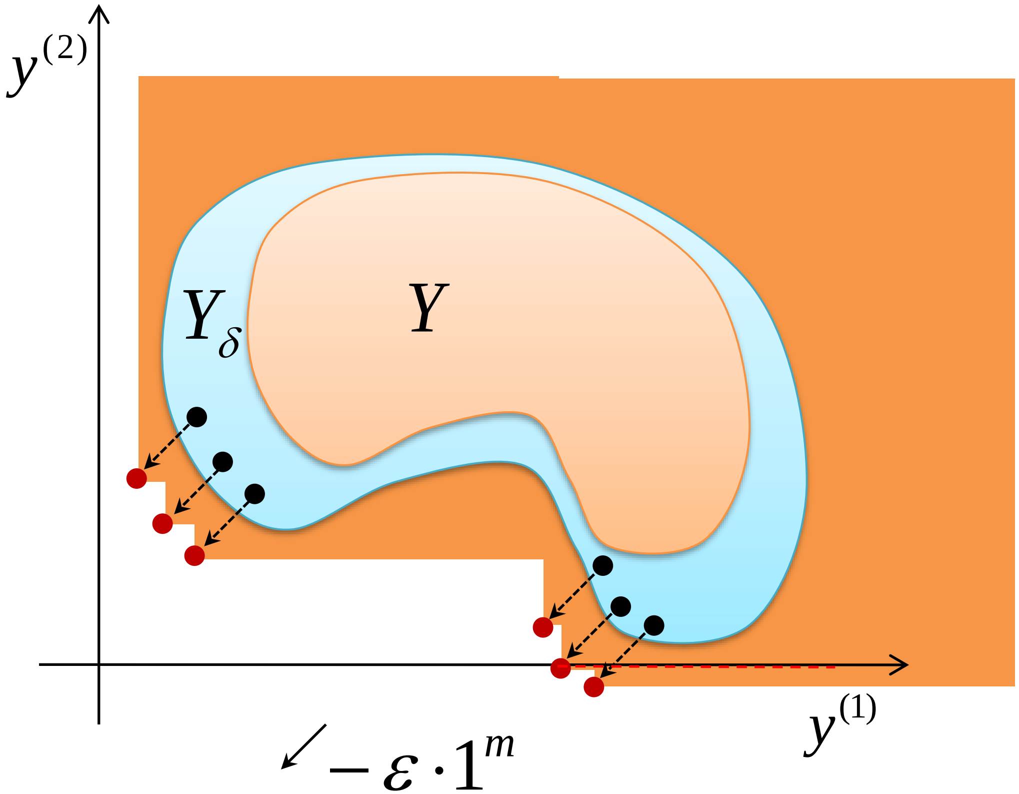
$$X = \{x \in R^n : G(x) \le 0\}$$

$$X_\delta = \{x \in R^n : G(x) \le \delta\},\ \delta \ge 0$$

$$Y = F(X)$$

$$Y_\delta = F(X_\delta)$$

ε,δ -Pareto Set



$Y_{\varepsilon,\delta}$ is $\varepsilon,\delta-$Pareto set if

1. $Y_{\varepsilon,\delta}\subseteq Y_\delta$
2. $P\,(Y_{\varepsilon,\delta})=Y_{\varepsilon,\delta}$
3. $Y\subseteq Y_{\varepsilon,\delta}-\varepsilon\cdot 1^m+R_+^m$

The goal: for given $\varepsilon>0,\delta>0$

find $\varepsilon,\delta-$Pareto set $Y_{\varepsilon,\delta}$ and $X_{\varepsilon,\delta}$

such that $F(X_{\varepsilon,\delta})=Y_{\varepsilon,\delta}$

COMPUTATIONAL EXPERIMENT

$$f^{(1)}(x) = x^{(1)},$$

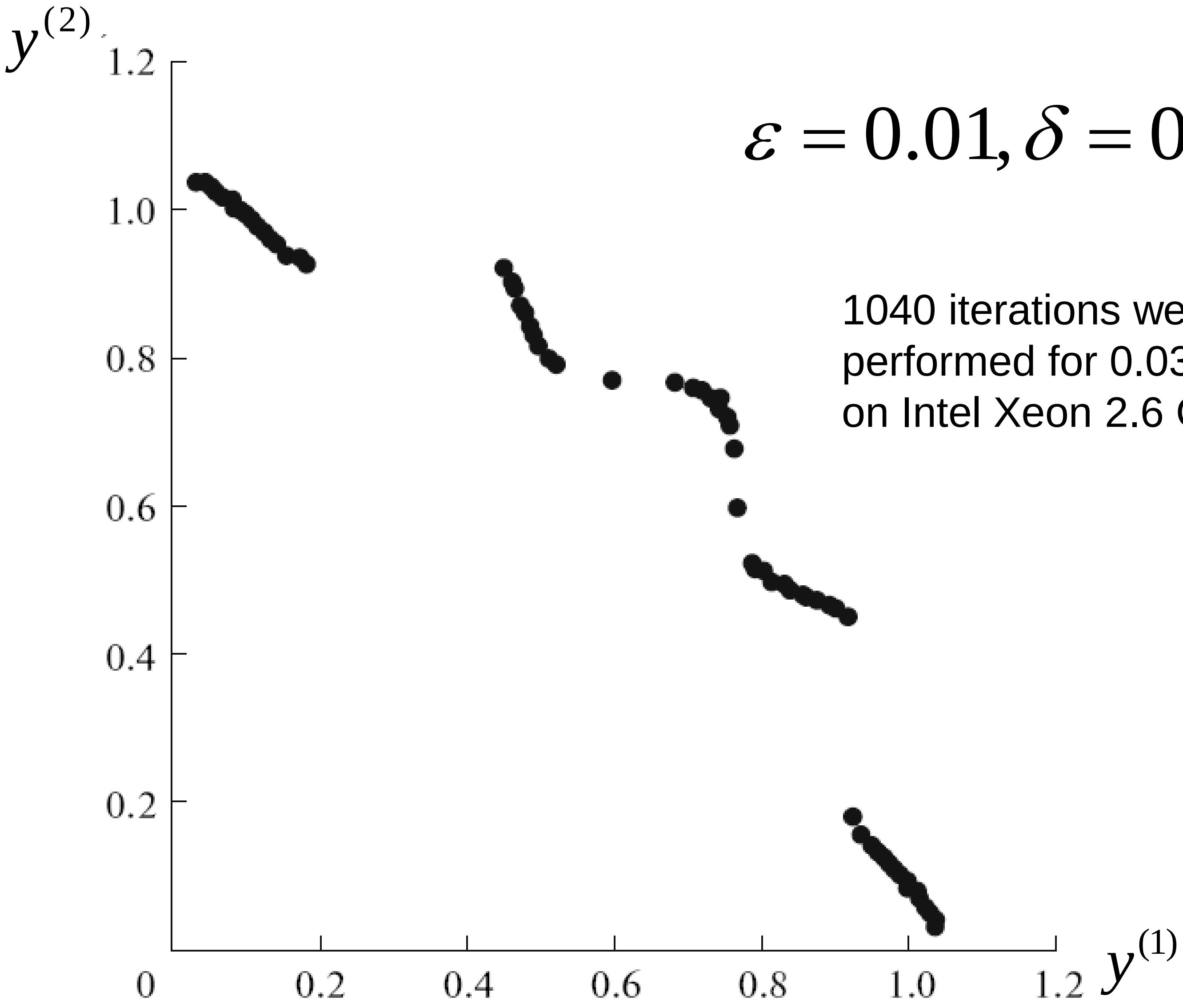
$$f^{(2)}(x) = x^{(2)},$$

$$-(x^{(1)})^2 - (x^{(2)})^2 + 1 + 0.1\cos\left(16\,\mathrm{arctg}\frac{x^{(1)}}{x^{(2)}}\right) \le 0,$$

$$(x^{(1)} - 0.5)^2 + (x^{(2)} - 0.5)^2 - 0.5 \le 0.$$

D. Chafekar, J. Xuan, K. Rasheed, Constrained Multiobjective Optimization Using Steady State Genetic Algorithms, In Proceedings of Genetic and Evolutionary Computation Conference, Chicago, Illinois, pp 813-824, 2003.

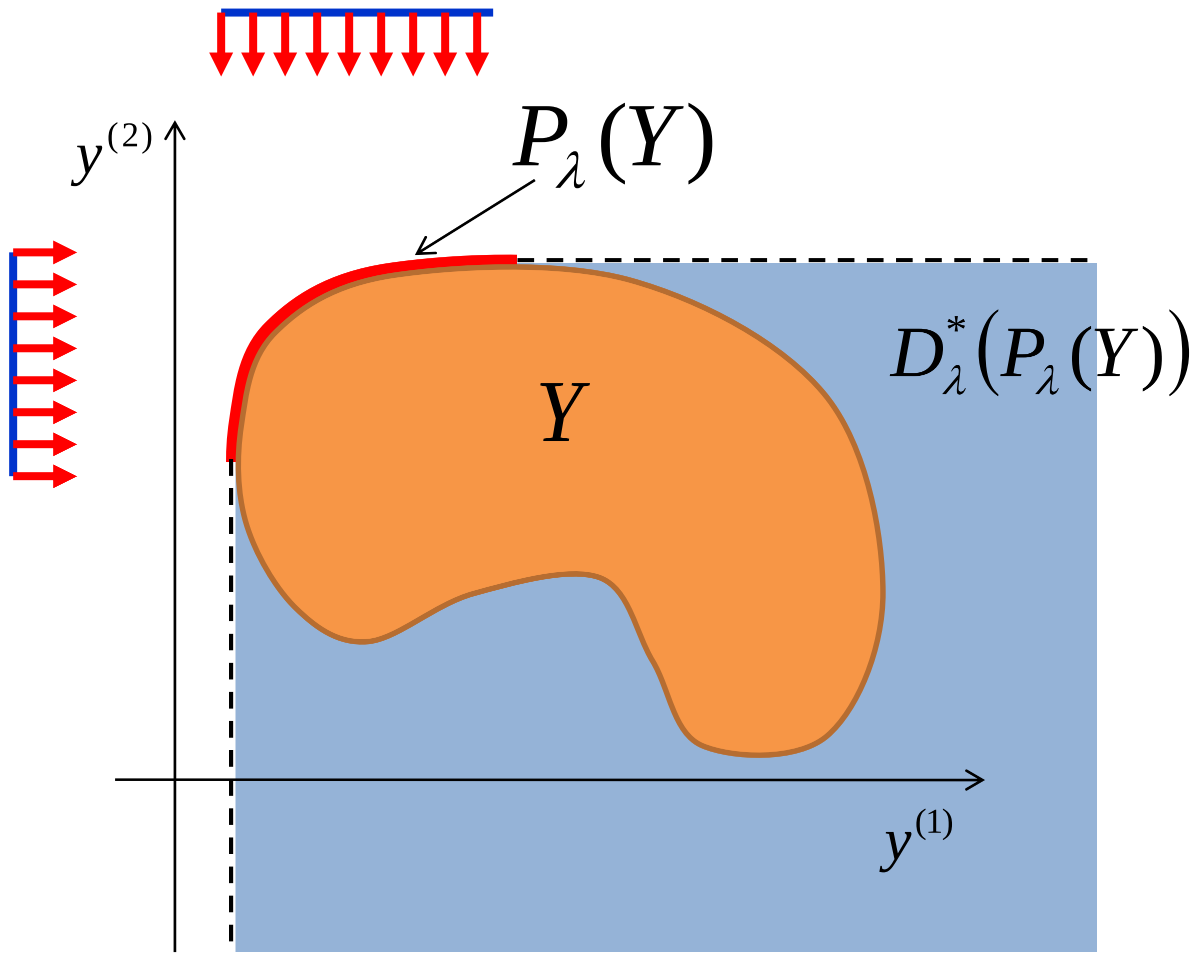
COMPUTATIONAL RESULTS



$$\varepsilon = 0.01, \delta = 0.01$$

1040 iterations were
performed for 0.03 s
on Intel Xeon 2.6 GHz PC

COMBINING MAXIMIZATION AND MINIMIZATION

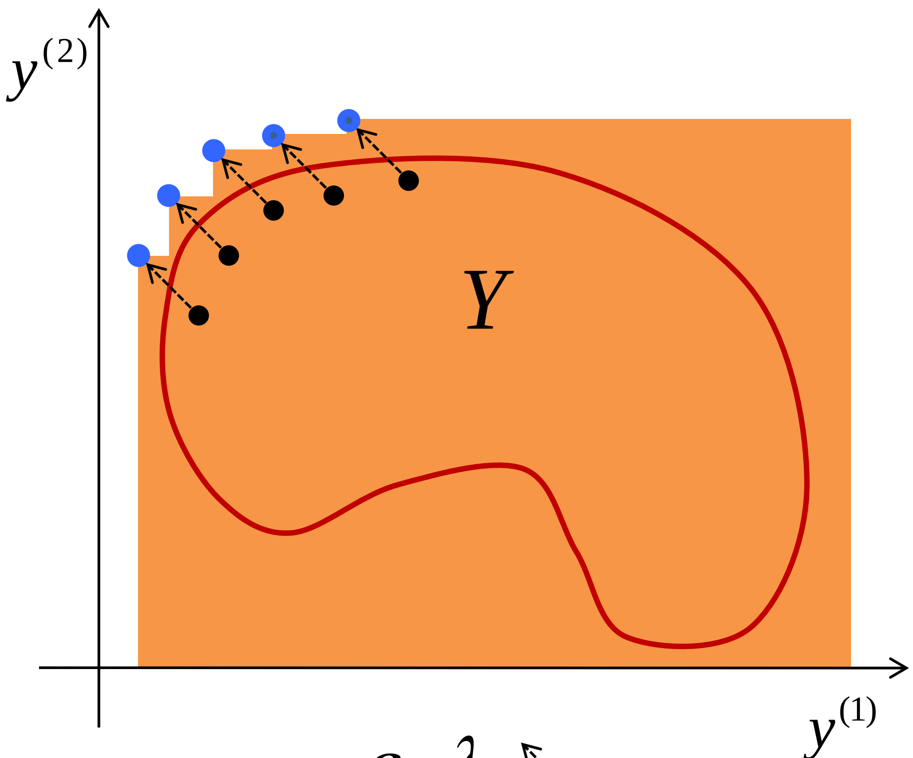


$$f^{(1)}(x) \to \min$$

$$f^{(2)}(x) \to \max$$

$$\lambda = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

GENERALIZATION OF A NOTION OF ε-PARETO SET



$\varepsilon \cdot \lambda$

$$1.\ Y_\varepsilon \subseteq Y$$

$$2.\ P_\lambda(Y_\varepsilon) = Y_\varepsilon$$

$$3.\ Y \subseteq D_\lambda^*\left(Y_\varepsilon + \varepsilon \cdot \lambda\right)$$

$$\lambda = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

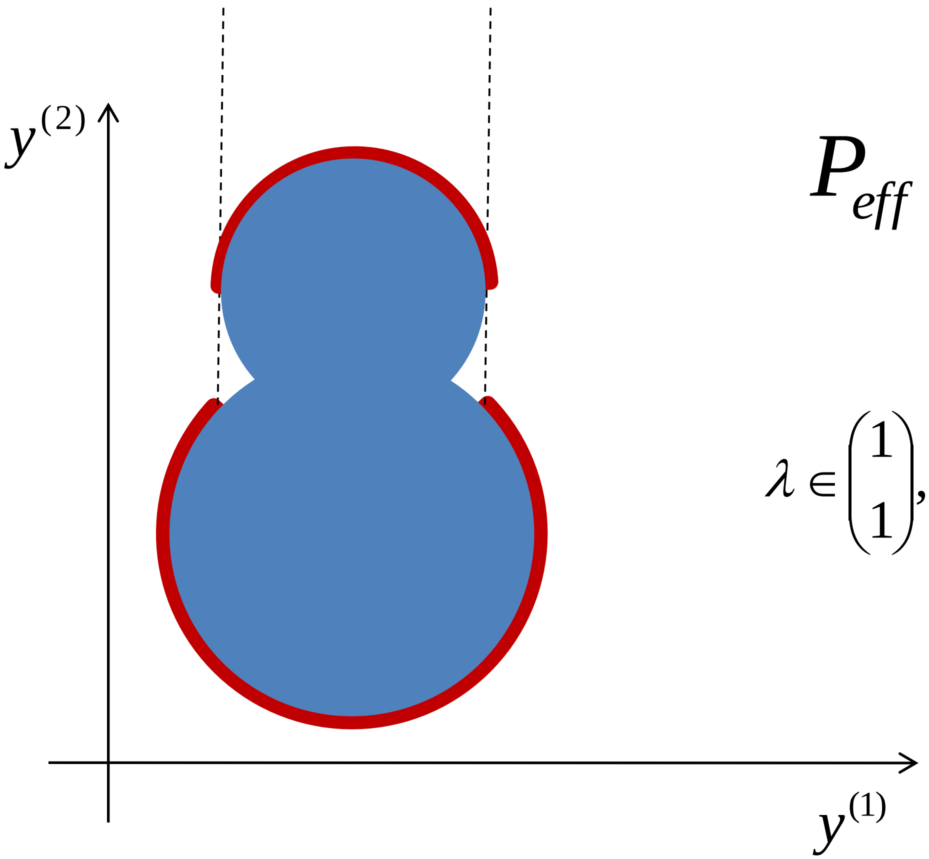
●●● Y_ε

●●● $Y_\varepsilon + \varepsilon \cdot \lambda$

$$f^{(1)}(x) \to \min$$

$$f^{(2)}(x) \to \max$$

EFFICIENT FRONTIER

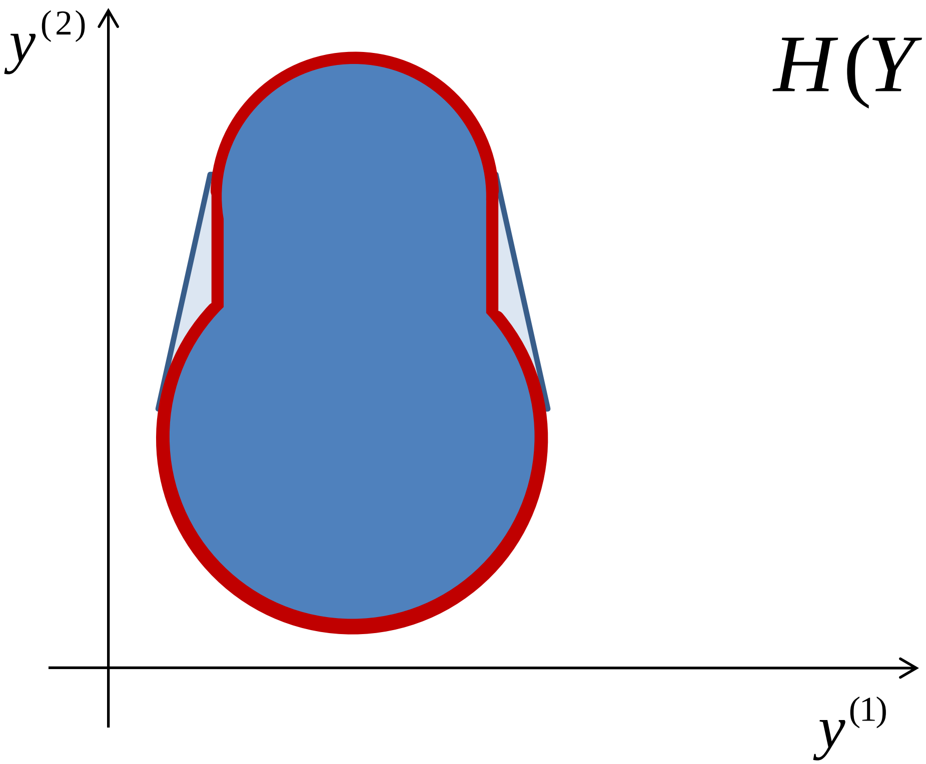


$$P_{eff} = \bigcup_{\lambda \in \{-1,1\}^m} P_\lambda(Y)$$

$$\lambda \in \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix}$$

Theorem. Efficient frontier of a strictly convex set coincides with its frontier.

EFFICIENT HULL

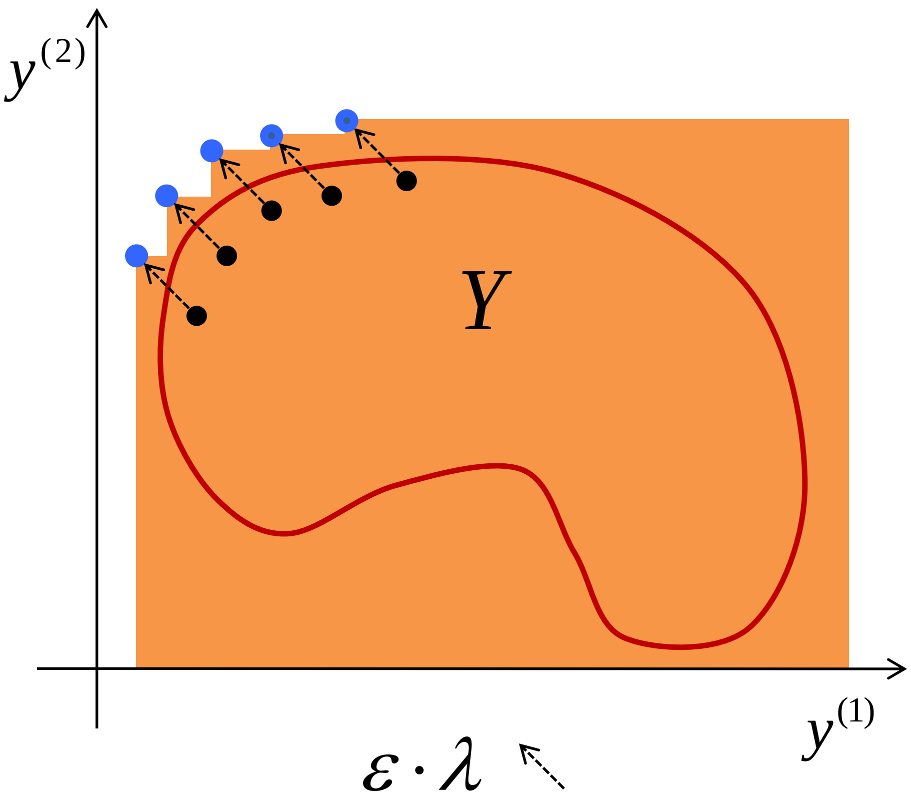


$$H(Y) = \bigcap_{\lambda \in \{-1,1\}^m} D_\lambda^*\left(P_\lambda(Y)\right)$$

Theorem. Convex hull of a set contains its efficient hull:

$$Y \subseteq H(Y) \subseteq Conv(Y)$$

GENERALIZATION OF A NOTION OF ε-PARETO SET



$$1.\ Y_\varepsilon \subseteq Y$$

$$2.\ P_\lambda(Y_\varepsilon) = Y_\varepsilon$$

$$3.\ Y \subseteq D_\lambda^*\left(Y_\varepsilon + \varepsilon \cdot \lambda\right)$$

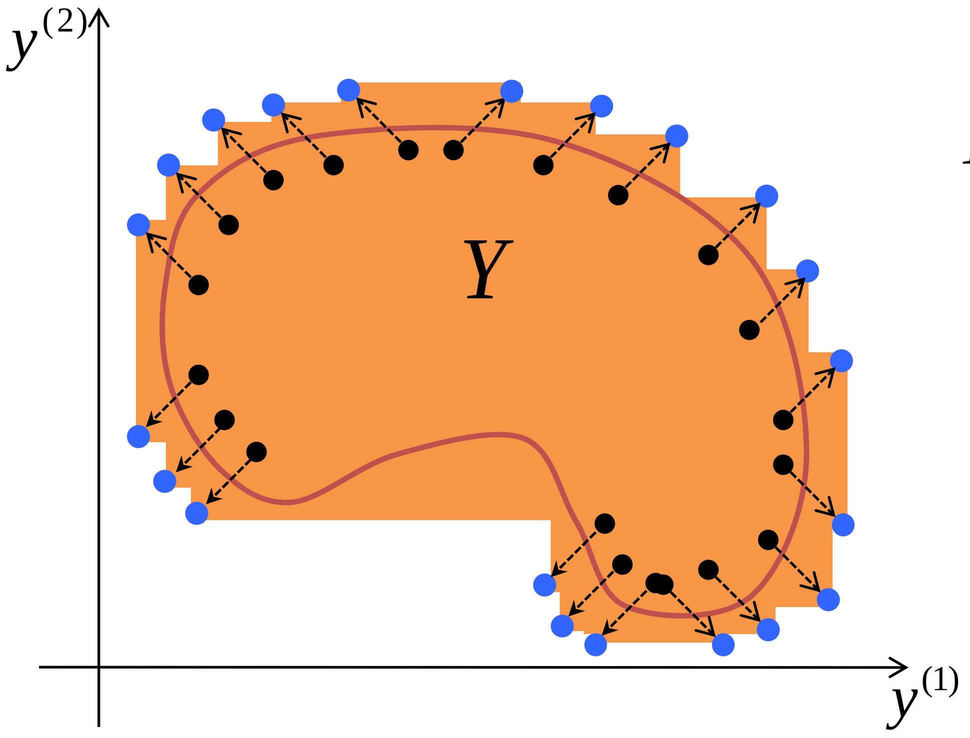
●●● Y_ε

●●● $Y_\varepsilon + \varepsilon \cdot \lambda$

$$f^{(1)}(x) \to \min$$

$$f^{(2)}(x) \to \max$$

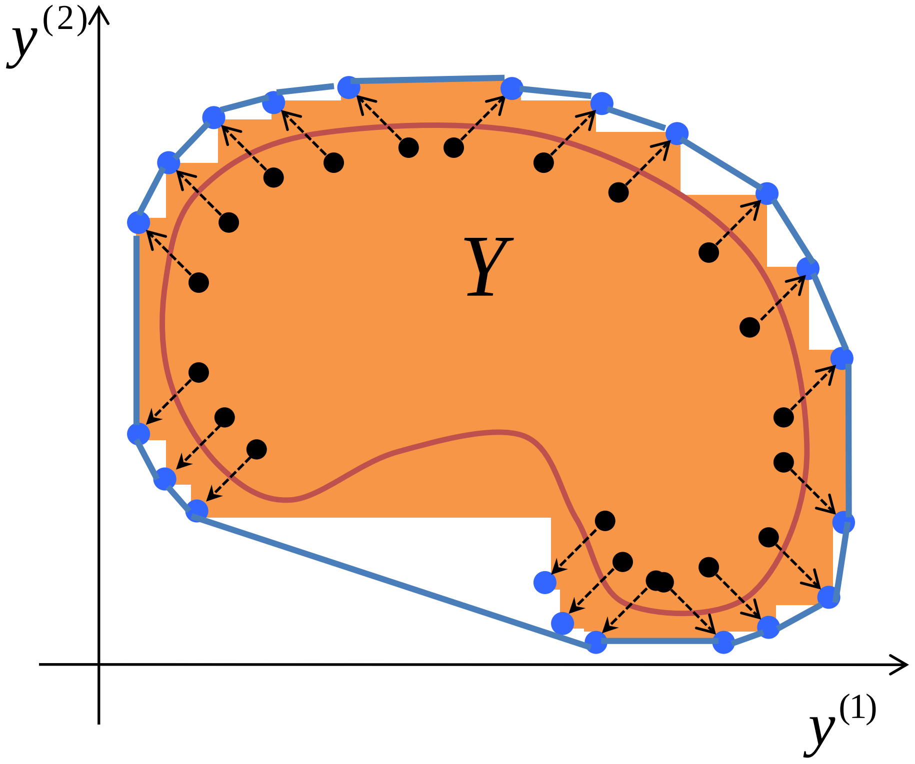
$\mathcal{E}$-EFFICIENT HULL



$$H_\varepsilon = \bigcap_{\lambda \in \{-1,1\}^m} D_\lambda^*(Y_\varepsilon^\lambda + \varepsilon \cdot \lambda)$$

- $B_\varepsilon = \bigcup_{\lambda \in \{-1,1\}^n} Y_\varepsilon^\lambda$ $-\ \varepsilon -$ *efficient frontier*
- $\overline{B}_\varepsilon = \bigcup_{\lambda \in \{-1,1\}^n} \left(Y_\varepsilon^\lambda + \varepsilon\lambda\right)$ $-$ external $\varepsilon -$ efficient frontier

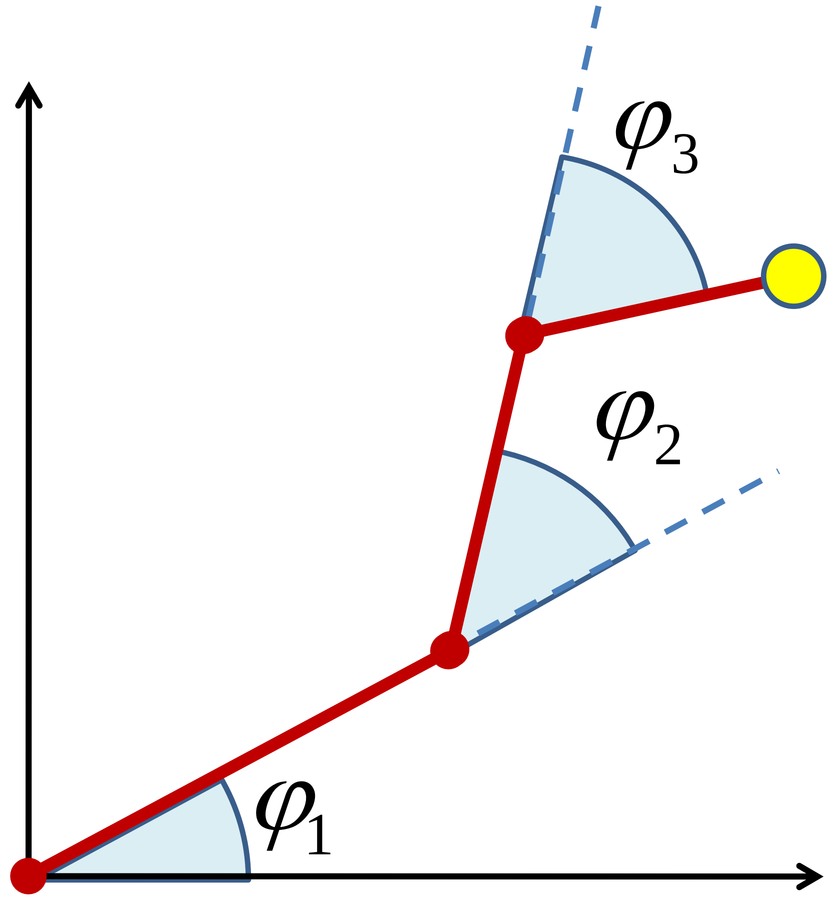
ε-EFFICIENT HULL and CONVEX HULL

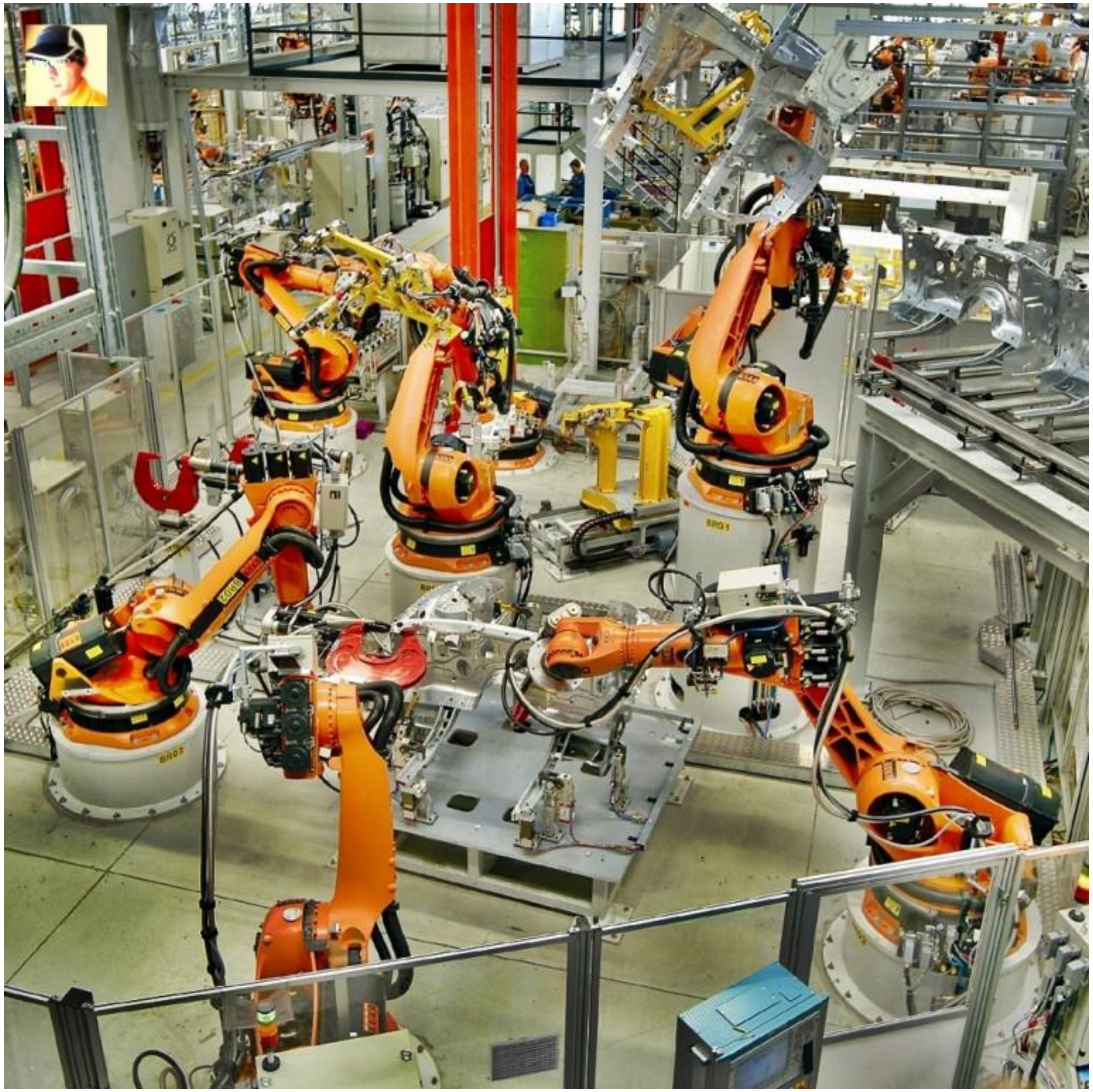


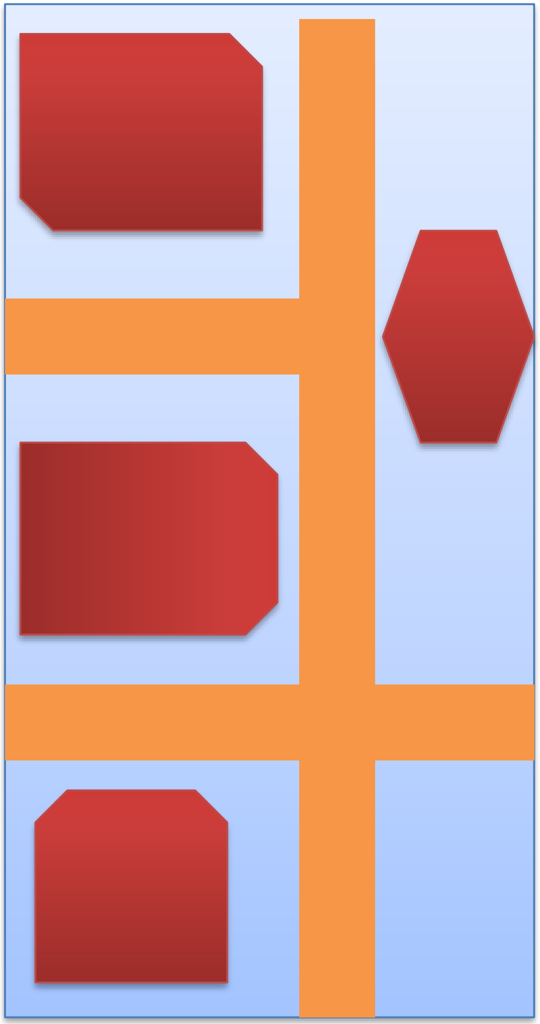
$$Y \subseteq H_\varepsilon \subseteq Conv(\overline{B}_\varepsilon)$$

- B_ε
- $\overline{B}_\varepsilon$

BOUNDING THE WORKING SET OF A ROBOTIC MANIPULATOR







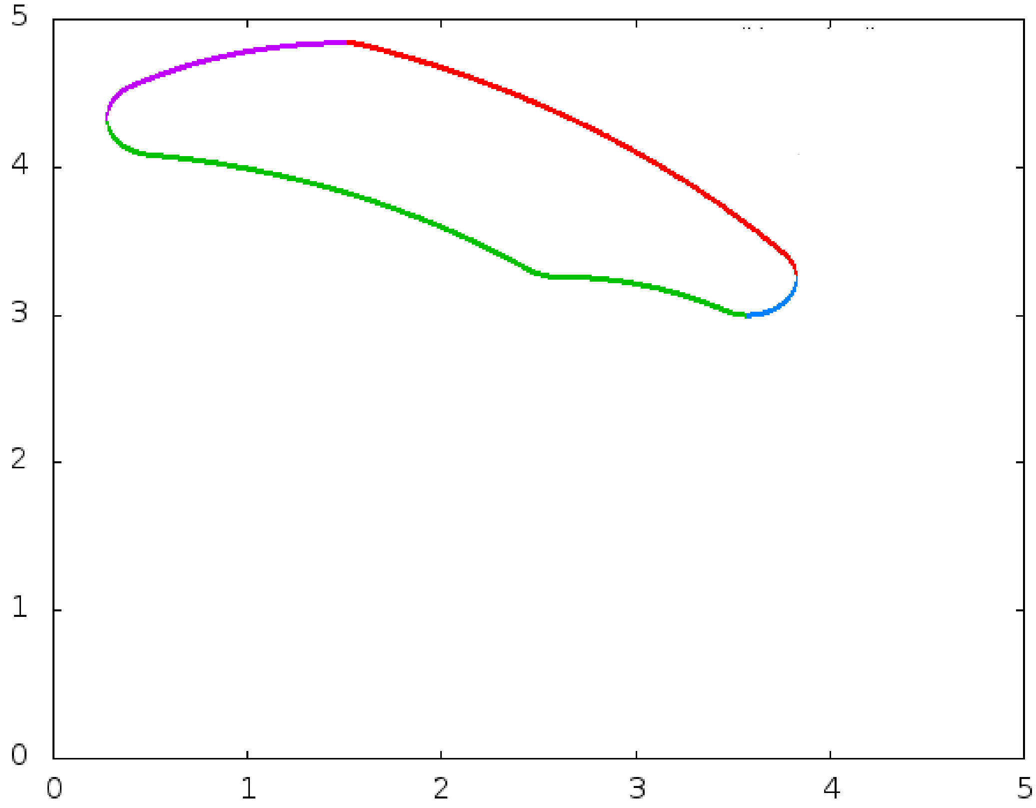
ROBOT WITH THREE LINKS

$$L_1 = 3, L_2 = 2, L_3 = 0.25,$$

$$\frac{\pi}{6} \le \varphi_1 \le \frac{\pi}{3},$$

$$\frac{\pi}{6} \le \varphi_2 \le \frac{\pi}{3},$$

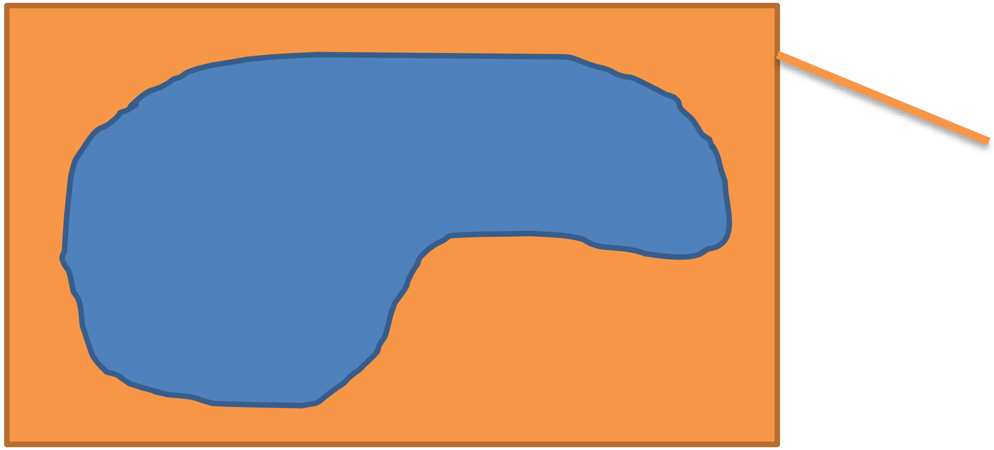
$$-\pi \le \varphi_3 \le \pi$$



SOLVING INEQUALITIES

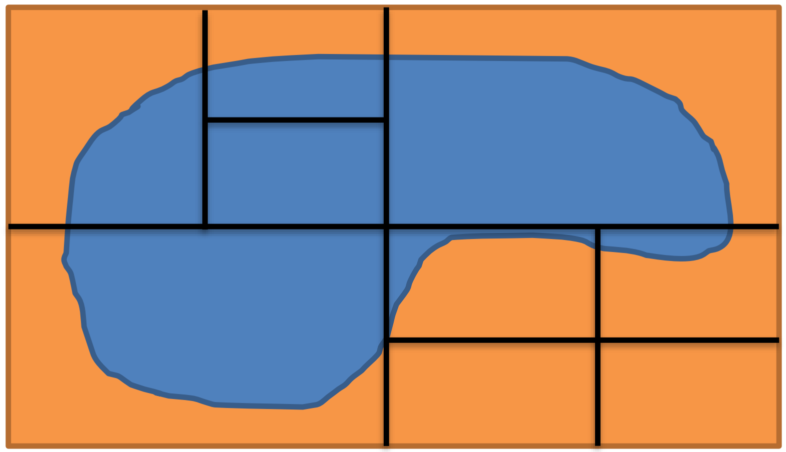
Find a set, defined by a system of inequalities

$$g_1(x) \le 0, \ldots, g_m(x) \le 0$$
$$x_i \in [a_i, b_i], i = 1, \ldots, n$$



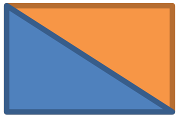
$$P = [a_1, b_1] \times \ldots \times [a_n, b_n]$$

NON-UNIFORM COVERINGS APPROACH



Inner  $\max_{x \in P_j} g_i(x) < 0 \quad \mathit{for\ all}\ \ i = 1,\dots,m$

Outer  $\mathit{for\ some}\ i \in 1,\dots,m \ \ \min_{x \in P_j} g_i(x) > 0$

Boundary  $\mathit{otherwise}$ ⇦ subject to split

ALGORITHM OUTLINE

1: **if** $L = \varnothing$ **then exit else take a box** P_j **from the list** L

2: **if** $\max_{x \in P_j} g_i(x) < 0$ **for all** $i = 1, \ldots, m$ **then**

3: **add** P_j **to the list** I ← Inner boxes

4: **else if for some** $i \in 1, \ldots, m$ $\min_{x \in P_j} g_i(x) > 0$ **then**

5: **add** P_j **to the list** O ← Outer boxes

6: **else if** $d(P_j) \leq \delta$ **then**

7: **add** P_j **to the list** B ← Boundary boxes

8: **else**

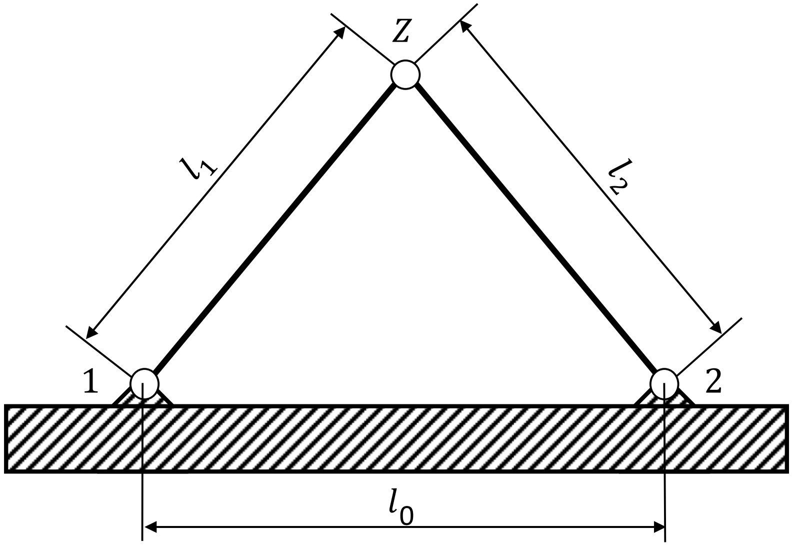
9: **bisect** P_j **into two boxes**

10: **and add them to the list** L

11: **endif**

12: **goto** 1

PRACTICAL EXAMPLE: APPROXIMATING WORKING SET OF A ROBOT

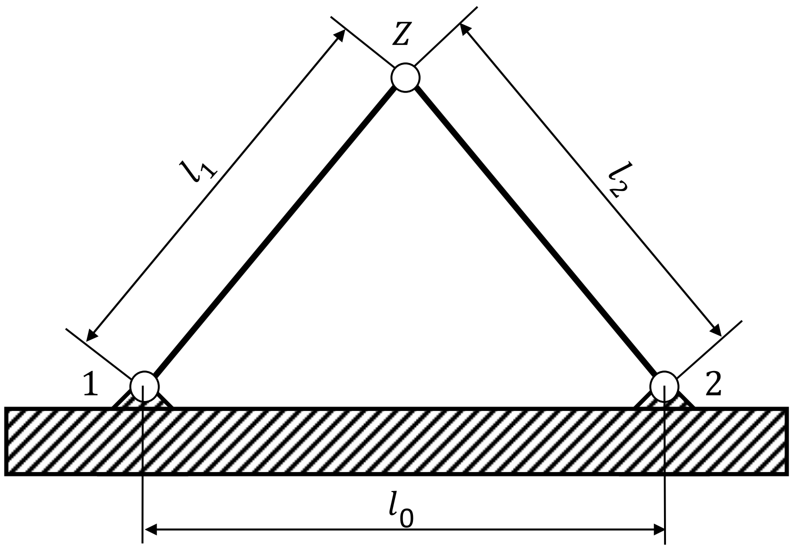


$Find\ all\ possible\ positions\ of\ Z\ where$

$linkages\ l_1,\ l_2\ can\ change\ length\ in\ given\ intervals$

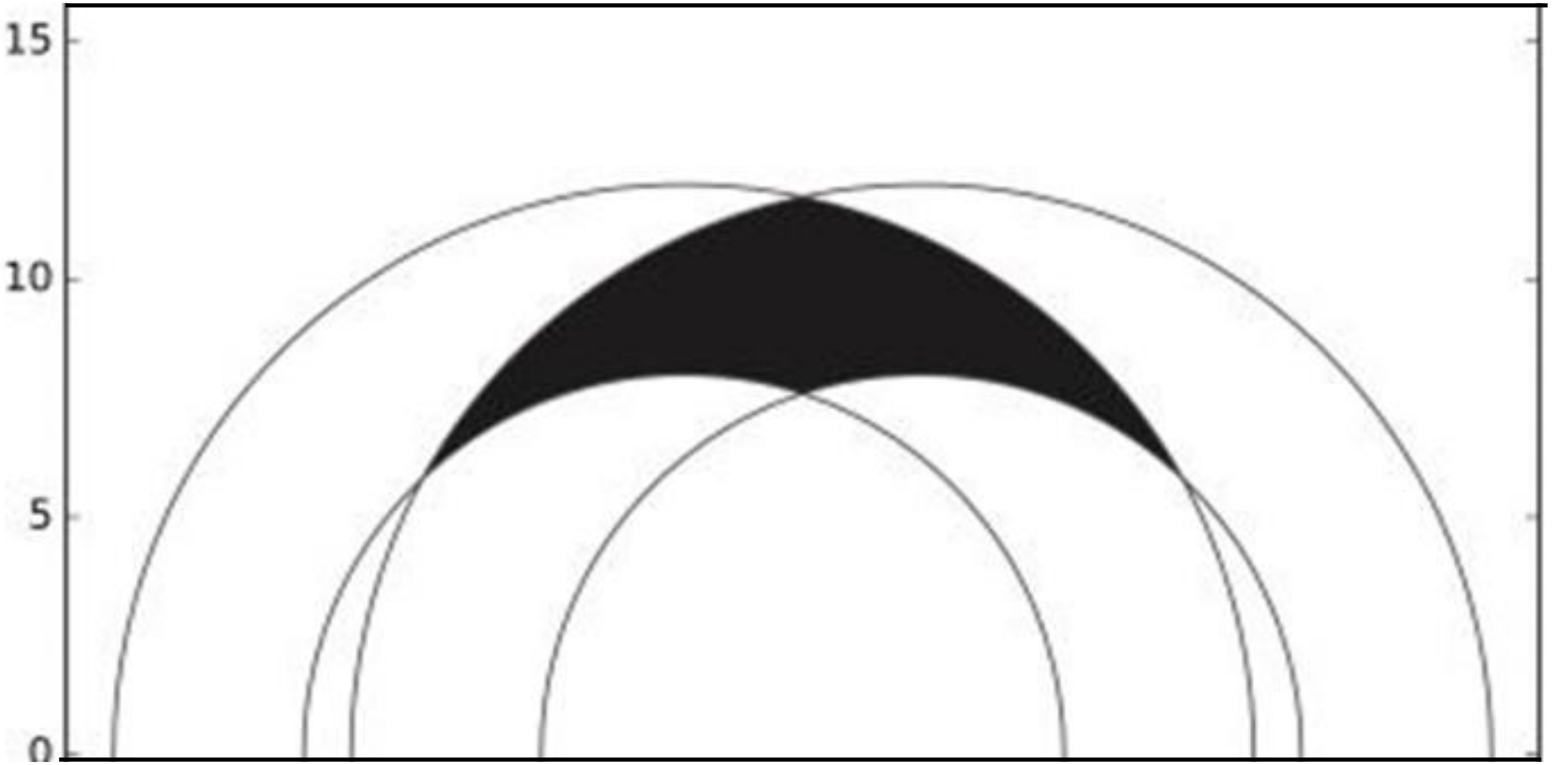
$$l_1^{\min} \le l_1 \le l_1^{\max},\ l_2^{\min} \le l_2 \le l_2^{\max}$$

SYSTEM OF INEQUALITIES



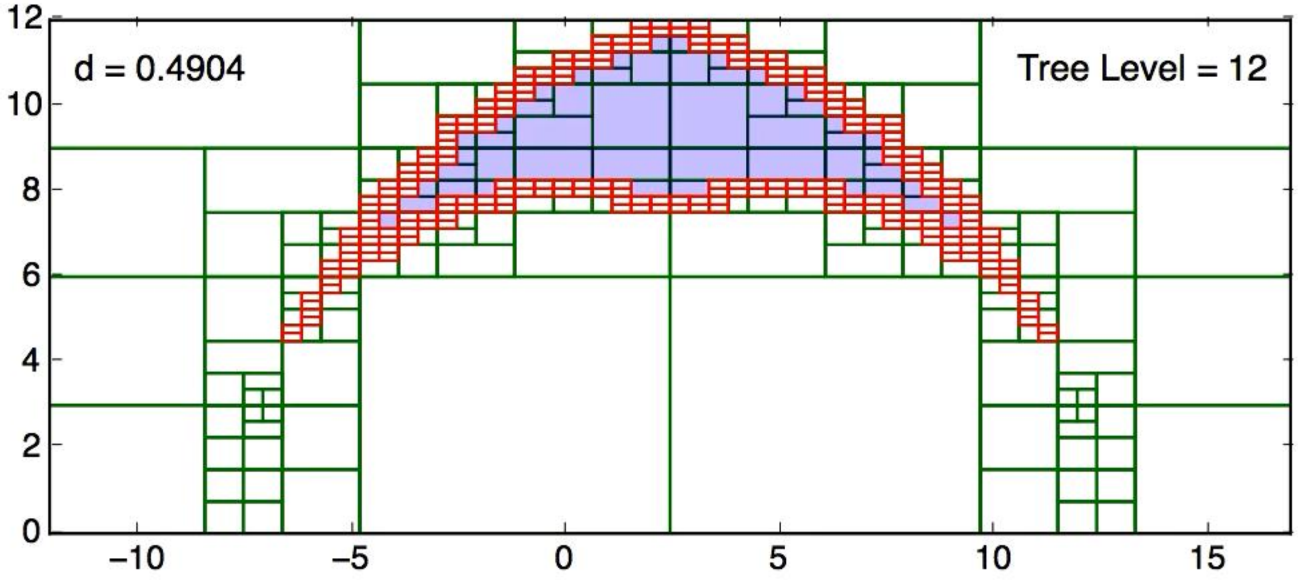
$$g_1(x_1, x_2) = x_1^2 + x_2^2 - (l_1^{max})^2,$$
$$g_2(x_1, x_2) = (l_1^{min})^2 - x_1^2 - x_2^2,$$
$$g_3(x_1, x_2) = (x_1 - l_0)^2 + x_2^2 - (l_2^{max})^2,$$
$$g_4(x_1, x_2) = (l_2^{min})^2 - (x_1 - l_0)^2 - x_2^2,$$
$$-l_1^{max} \leq x_1 \leq l_0 + l_2^{max},$$
$$0 \leq x_2 \leq \min(l_1^{max}, l_2^{max}),$$

ANALYTIC SOLUTION



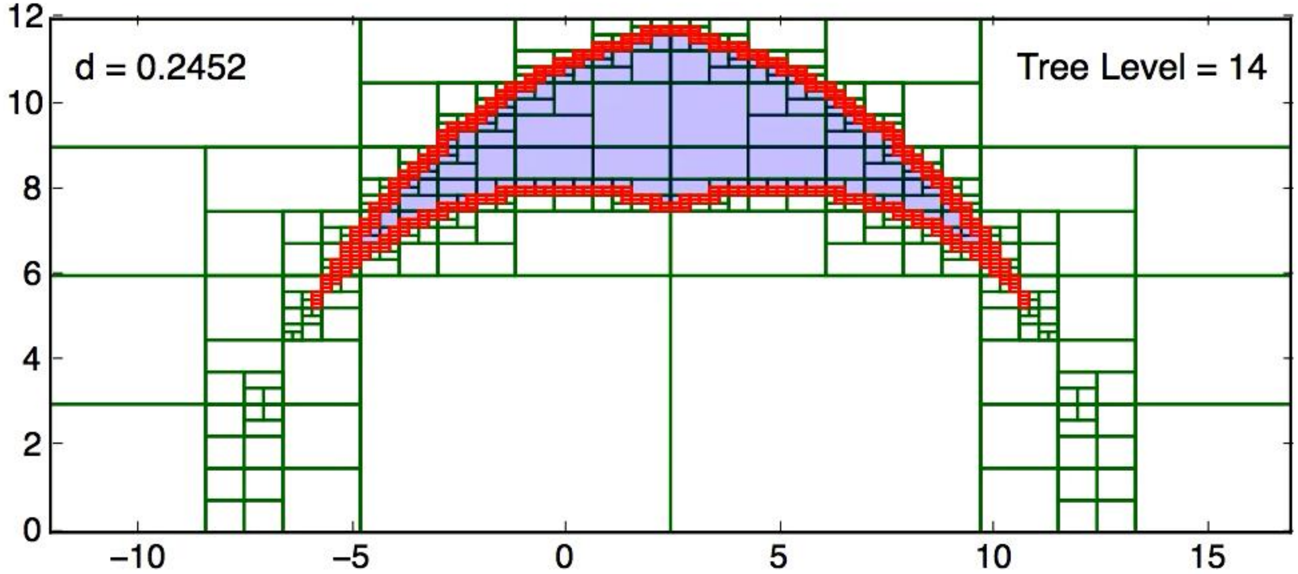
RESULTS

$$\delta = 0.5$$



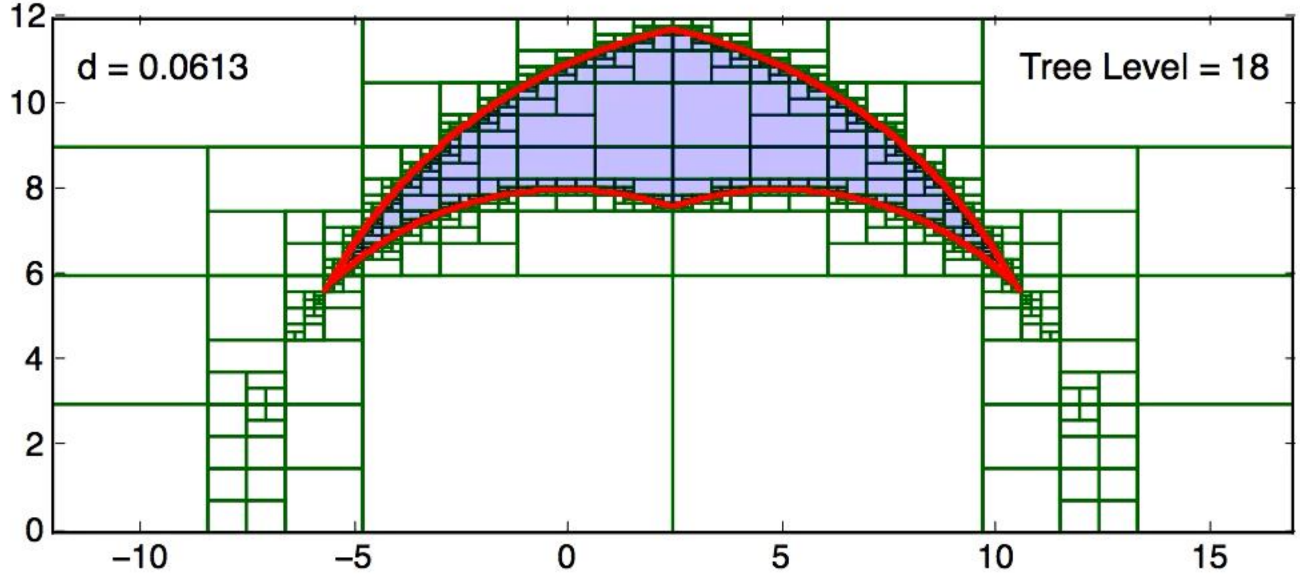
RESULTS

$$\delta = 0.25$$

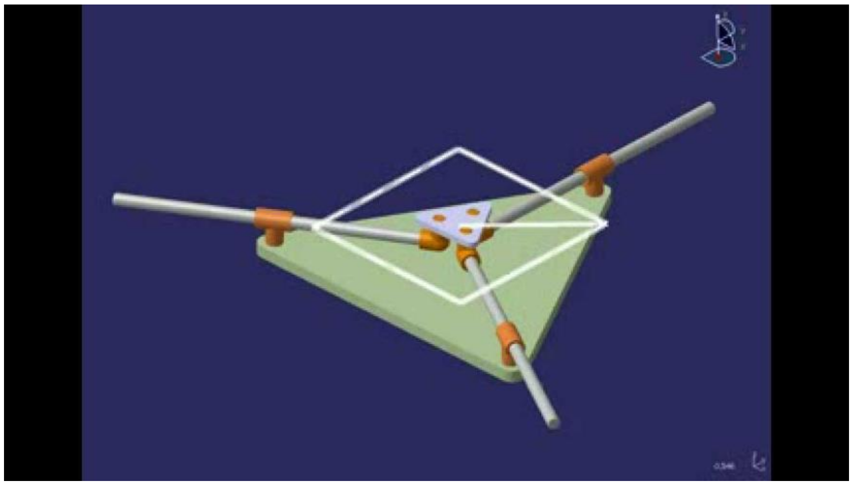


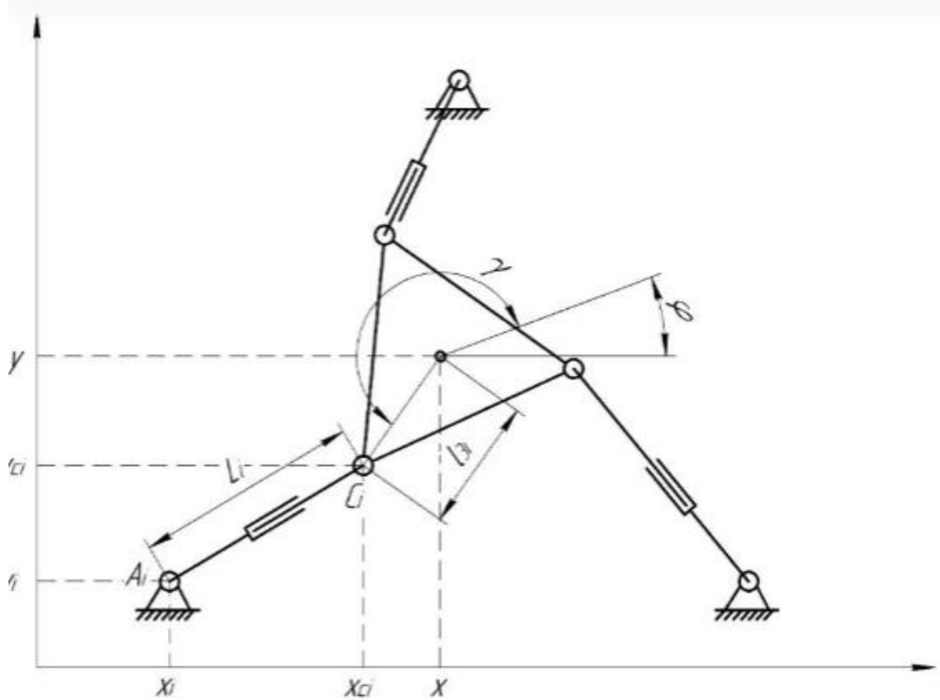
RESULTS

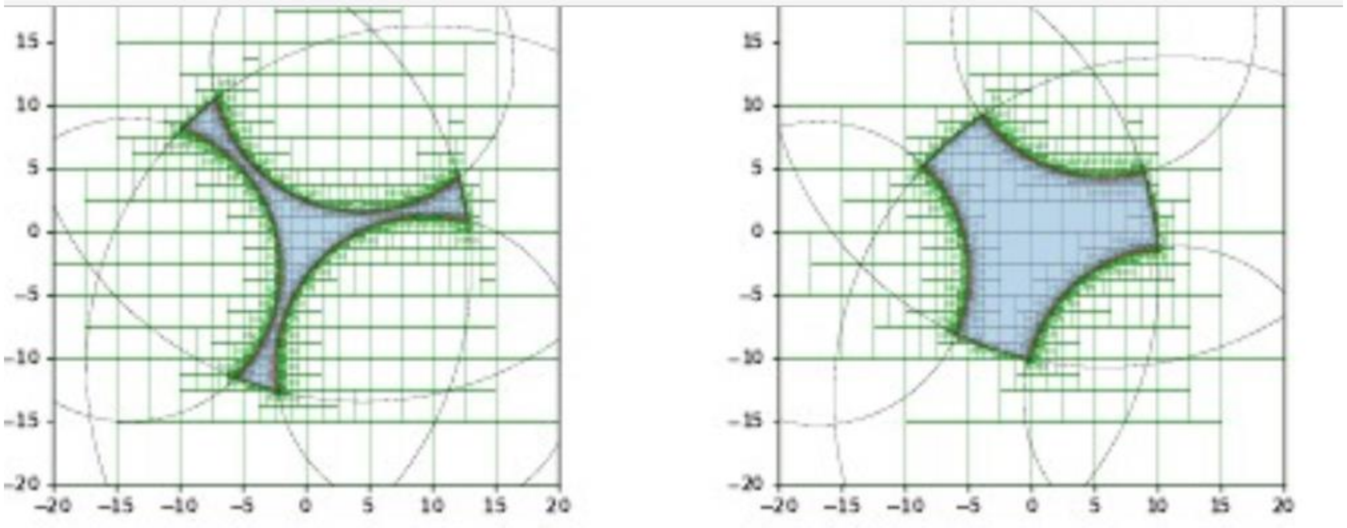
$$\delta = 0.07$$



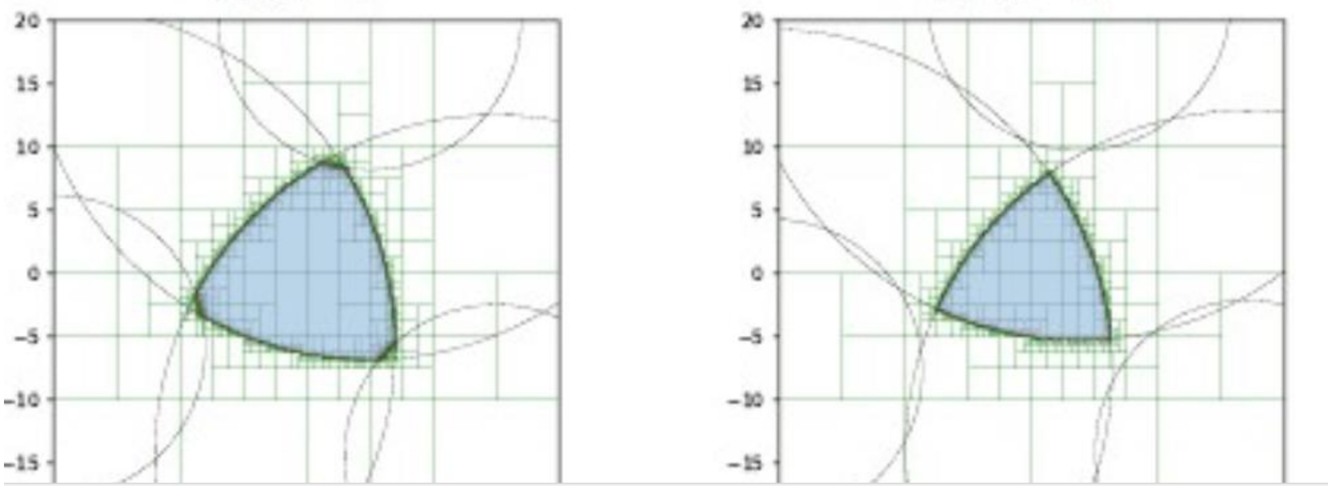
3-RPR Parallel Robot



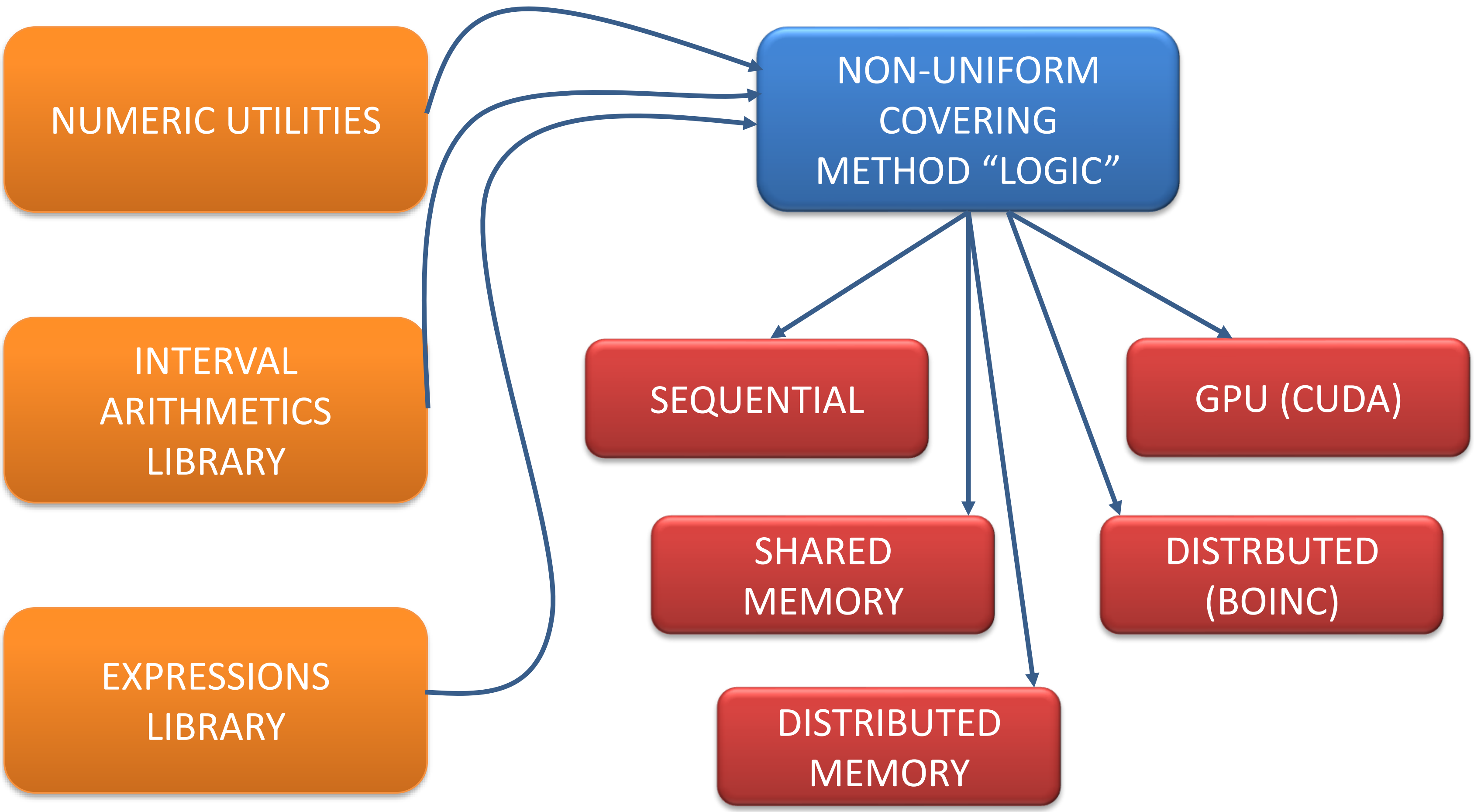




(a) $\xi_1 = 50$ (b) $\xi_2 = 80$



SOFTWARE INFRASTRUCTURE



EXPRESSIONS LIBRARY

Interval library supports only a limited set of algebraic functions: +, -, sin, cos, ….

We need a support for complex expressions, i.e.

$$f(x_1, x_2) = \sum_{k=1}^{n} \exp\left(\sin(x_1) + k \cdot \ln(x_1 + x_2)\right)$$

EXPRESSIONS LIBRARY

- Allows to declare complex formulas
- Supports automated evaluation of a function value, its derivatives, interval bounds and interval bounds for derivates

ROSENBROCK FUNCTION

```
template <class T>
Expr<T> Rosenbrock(int n)
{
    Expr<T> x;
    Iterator i(0, n-2);
    Expr<T> t = i;
    return loopSum(100*sqr(x[t+1]-sqr(x[i]))+sqr((x[i]-1)), i);
}
```

DropWave function

```
template <class T>
Expr<T> DropWave()
{
   Expr<T> x;
   Expr<T> a = sqr(x[0])+sqr(x[1]);
   Expr<T> b = 1+cos(12*sqrt(a));
   Expr<T> c = 0.5*a + 2;
   return –b/c;
}
```

Calling the calc method to get the value of the DropWave function at the given point (0.5, 1.5)

```
Expr<double> expr = DropWave<double>();
double result = expr.calc(FuncAlg<double>({0.5,
    1.5}));
std::cout << result;
```

Calculation of the interval estimate of DropWave function over the box [0.5, 1.5]x[-0.5, 0.5]

```
Expr<Interval<double>> expr =
        DropWave<Interval<double>>();
std::vector<Interval<double>> box =
        {{ 0.5, 1.5}, {-0.5, 0.5}};
Interval<double> result =
        expr.calc(InterEvalAlg<double>(box));
std::cout << result;
```

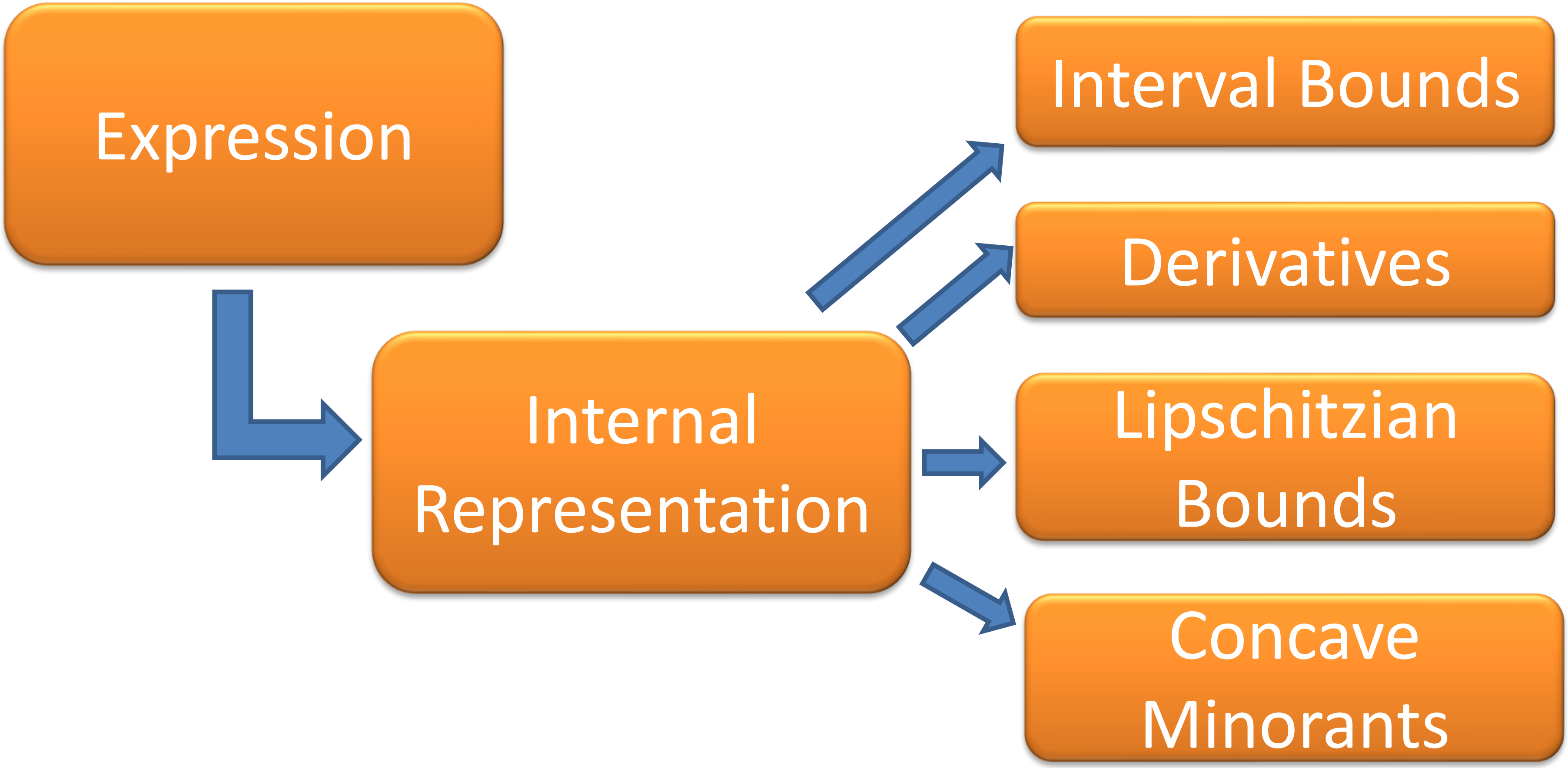
Calculate the gradient of the DropWave function at the point (0.5, 1.5)

```
Expr<ValDer<double>> expr =
        DropWave<ValDer<double>>();
ValDer<double> result = expr.calc(
        ValDerAlg<double>({0.5, 1.5}));
std::cout << result;
```

Calculations of the interval estimation of the gradient of the DropWave function on a box

```
Expr<IntervalDer<double>> expr =
        DropWave <IntervalDer<double>>();
std::vector<Interval<double>> box =
        {{ 0.5, 1.5}, {-0.5, 0.5}};
IntervalDer<double> result =
        expr.calc(IntervalDerAlg<double>(box));
std::cout << result;
```

EXPRESSION LIBRARY



SOFTWARE

The test suite for box-constrained optimization

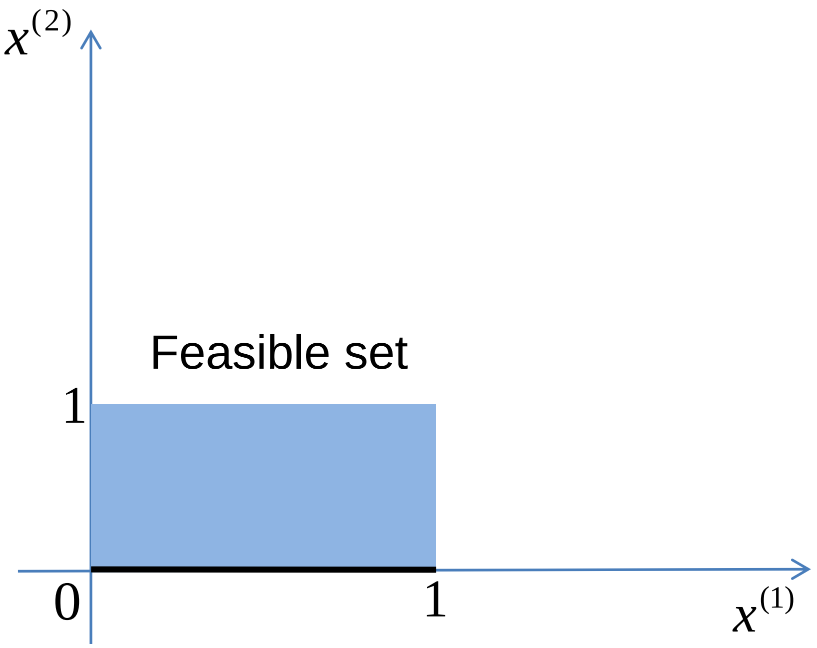
- https://github.com/alusov/mathexplib

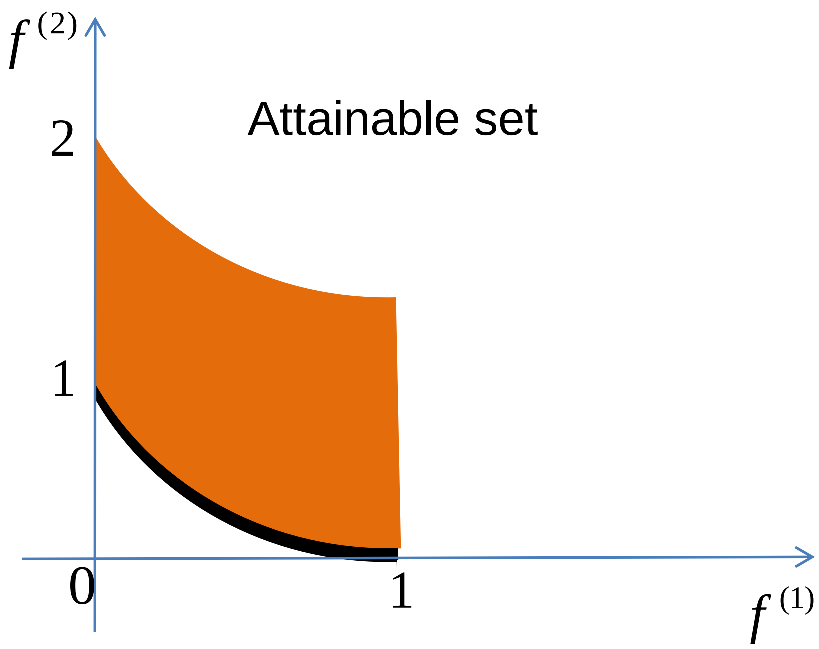
Team

- Acad. D. Sc. Yury Evtushenko (scientific management)
- D.Sc. Mikhail Posypkin (general management)
- Ph.D. Andrey Gorchakov (parallel computing)
- Ph.D. Andrey Turkin (applications to robotics)
- Ph.D. student Alexander Usov (expression library)
- Ms. student Andrey Igantov (distributed computing)

Thank you for attention!

EXAMPLE PROBLEM





$$f^{(1)}(x) = x^{(1)},$$
$$f^{(2)}(x) = (x^{(1)} - 1)^2 + x^{(2)}$$
$$0 \le x^{(1)} \le 1,$$
$$0 \le x^{(2)} \le 1$$

Convolution function

$$v(x) = \alpha_1 x^{(1)} + \alpha_2((x^{(1)} - 1)^2 + x^{(2)})$$
$$0 \le \alpha_1, 0 \le \alpha_2$$

IMPLEMENTATION

- Implemented using BNB-Solver library
  - Y. Evtushenko, M. Posypkin, I. Sigal, A framework for parallel large-scale global optimization // Computer Science - Research and Development 23(3), pp. 211-215, 2009
- Publicly available https://github.com/mposypkin/BNB-solver

INFORMATION SOURCES

1. Initially proposed for global optimization problems in 1971
   Ю. Г. Евтушенко, “Численный метод поиска глобального экстремума функций (перебор на неравномерной сетке)”, Ж. вычисл. матем. и матем. физ., 11:6 (1971), 1390–1403

2. Extened to multiobjective optimization in 1986
   Евтушенко Ю. Г., Потапов М. А. «Методы численного решения многокритериальных задач.» ДАН СССР, Т. 291, N 1, 1986, С. 25-39.

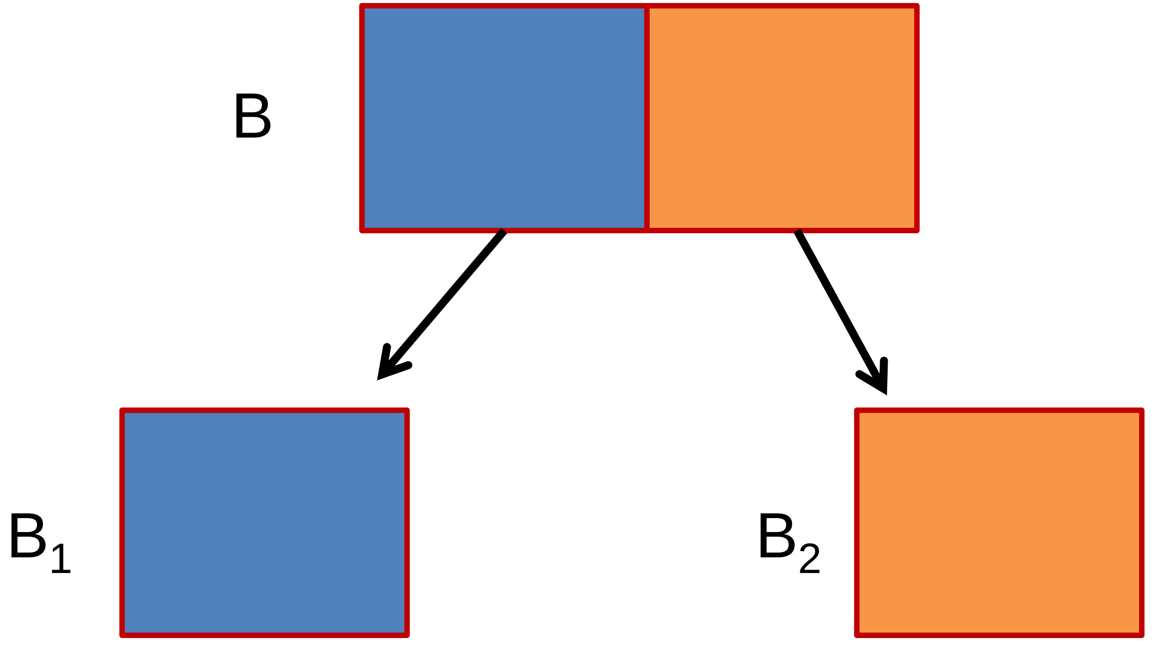
3. Parallel implementation 2009
   Y. Evtushenko, M. Posypkin, I. Sigal, A framework for parallel large-scale global optimization // Computer Science - Research and Development 23(3), pp. 211-215, 2009..

4. Advanced techniques for mathematical programming problems 2013
   Y. Evtushenko, M. Posypkin. A deterministic approach to global box-constrained optimization // Optimization Letters, 2013, Volume 7, Issue 4, pp 819-829.

5. Advanced techniques for multiobjective optimization
   Yu.G. Evtushenko. M.A. Posypkin, A deterministic algorithm for global multi-objective optimization // Optimization Methods and Software, Vol. 29, Iss. 5, pp. 1005-1019, 2014

BRANCHING RULE (SIMPLIFIED)

Divide by the longest edge:



"ARCHIVING"

$$\text{Procedure Update}(A, x)$$

$$\text{For all } a \in A \text{ do}$$

$$\quad \text{if } F(x) \leq F(a) \text{ then}$$

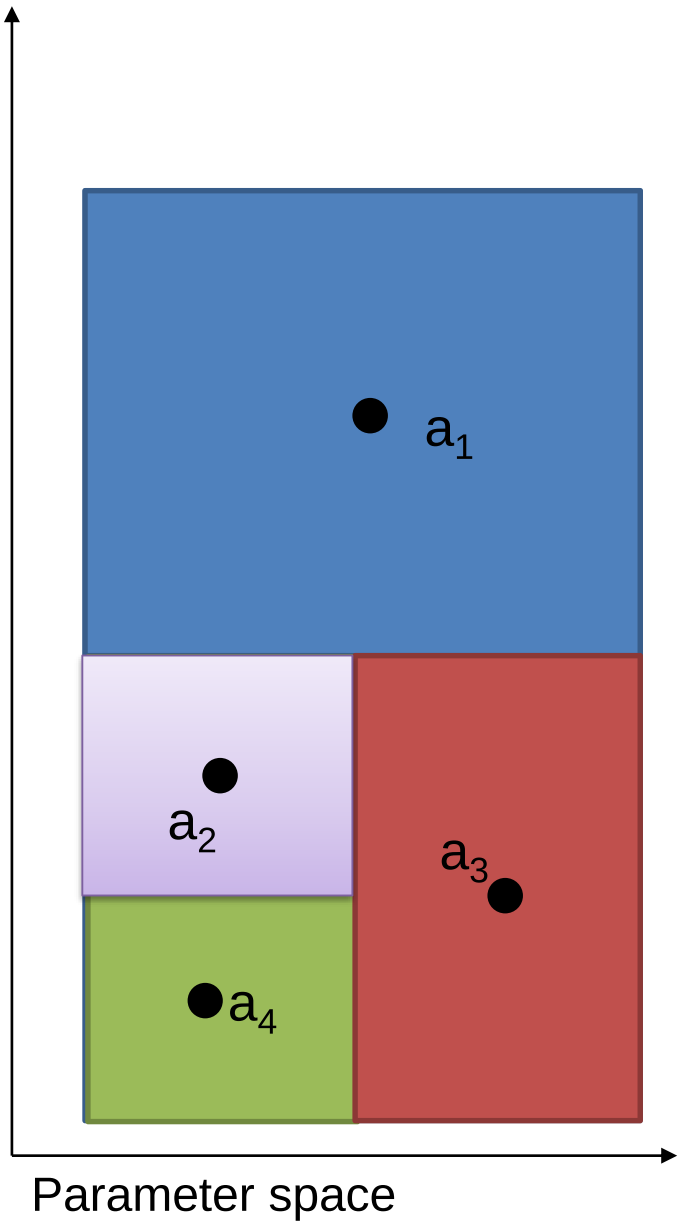
$$\qquad A = A \setminus \{a\}$$

$$\quad \text{if } F(a) \leq F(x) \text{ then}$$

$$\qquad \text{exit}$$

$$\text{end do}$$

$$A = A \bigcup \{x\}$$



BOUNDING CRITERIA FROM BELOW

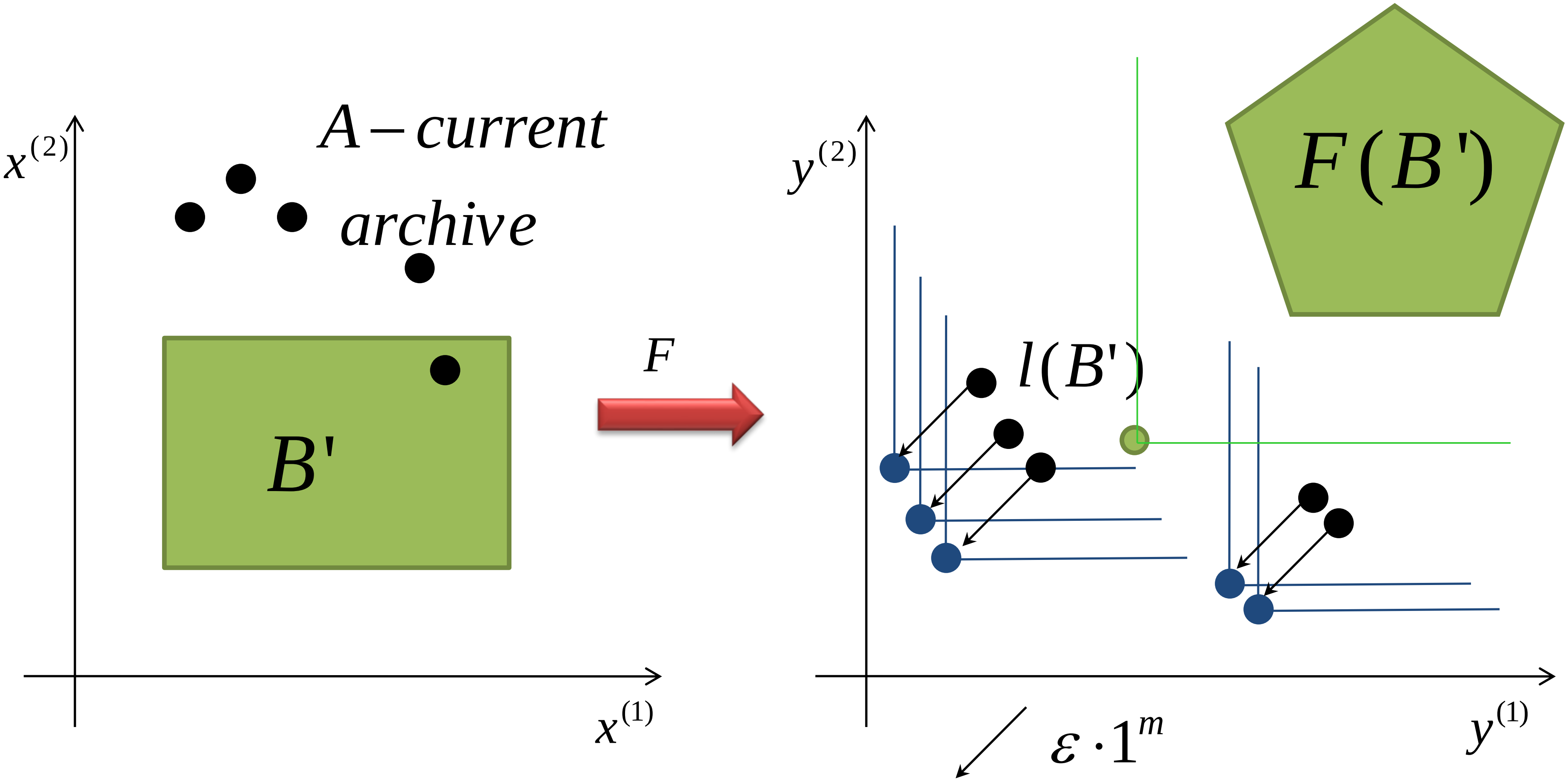
- Introduce a minorant for each criterion:

$$f^{(i)}(x) \geq \mu^{(i)}(x) \quad \text{for} \quad x \in B,\ i = 1,..,m$$

- Compute lower bound on some box B' - a point $l(B')$:

$$l(B') = (l^{(1)},\dots,l^{(m)}) \ \text{ where } \ l^{(i)} = \min_{x \in B'} \mu^{(i)}(x),\ i = 1,\dots,m$$

BOUNDING RULE



$$l(B') \in NE(F(A) - \varepsilon \cdot 1^m) \Rightarrow \text{discard } B'$$

ALGORITHM OUTLINE

1. $S = \{ X \}$, $A = \{\}$

2. If S is empty goto 8

3. Select (and remove) box B from S

4. Take point c – center of B

5. Update current archive A with c

6. Check **bounding rule** for B. If rule discards B then go to 2

7. Divide B into 2 boxes and add them to S, go to 2

8. End