

# Computer Science

## Jerzy Świątek

# Systems Modelling and Analysis

*Choose yourself and new technologies*

## L.6 Constrained optimization methods



**HUMAN CAPITAL**  
HUMAN – BEST INVESTMENT!



Wrocław University of Technology

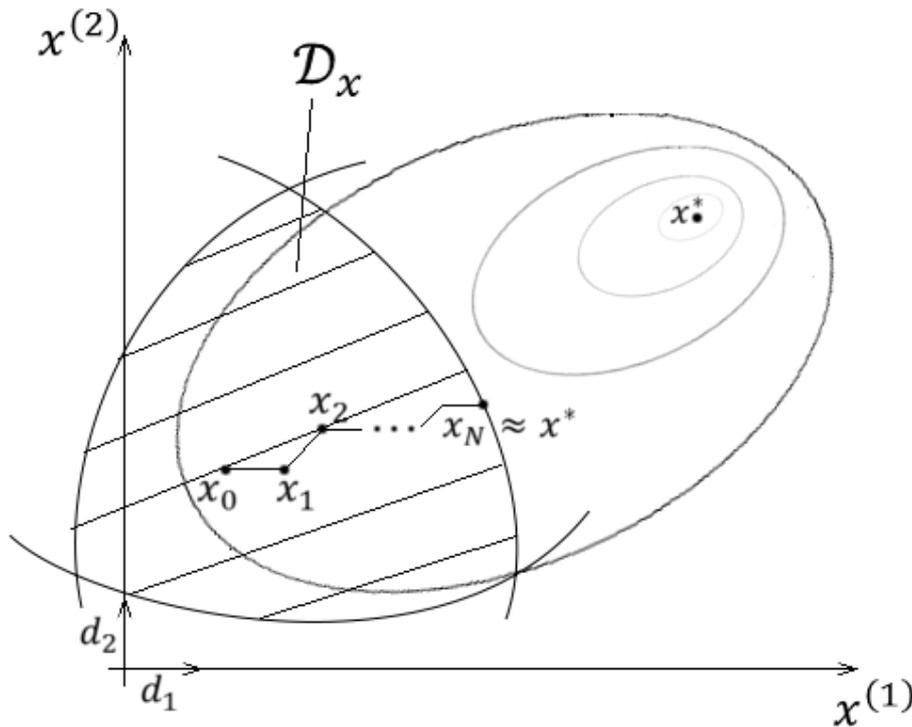
EUROPEAN  
SOCIAL FUND



Project co-financed from the EU European Social Fund



# Numerical constrained optimization methods



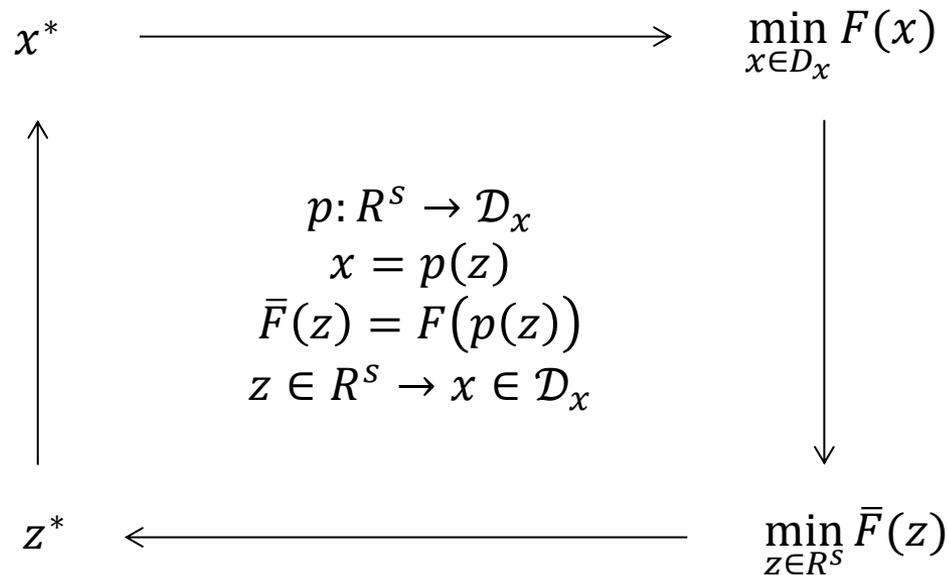
$$x^* \rightarrow F(x^*) = \min_{x \in \mathcal{D}_x} F(x)$$

1. Elimination of constraints
2. Penalty function method
  - exterior penalty
  - barrier function
3. Methods of feasible directions
4. Other approaches



# Elimination of constraints

$$x^* \rightarrow F(x^*) = \min_{x \in \mathcal{D}_x} F(x)$$





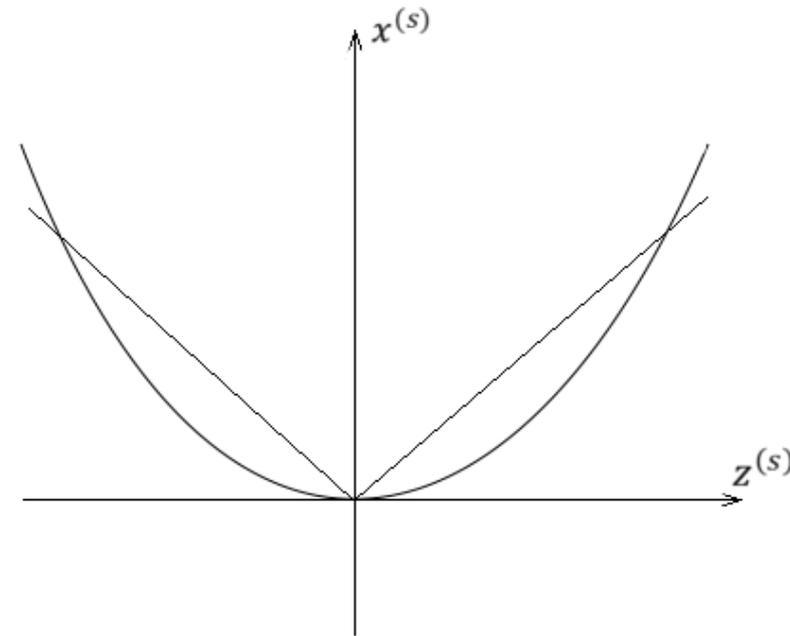
## Example

$$x^* \rightarrow F(x^*) = \min_{x \in \mathcal{D}_x} F(x) \quad F(x) = \sum_{s=1}^S (x^{(s)} + 5)^2$$

$$\mathcal{D}_x = \{x \in \mathbb{R}^S, \quad x^{(s)} \geq 0, \quad s = 1, 2, \dots, S\}$$

$$x^{(s)} = (z^{(s)})^2 \text{ or } x^{(s)} = |z^{(s)}|$$

$$z^{(s)} \in \mathbb{R} \rightarrow x^{(s)} \in [0, \infty)$$



$$F(x) = \sum_{s=1}^S (x^{(s)} + 5)^2 \rightarrow \bar{F}(z) = \sum_{s=1}^S (|z^{(s)}| + 5)^2 \quad z^* \rightarrow \bar{F}(z^*) = \min_{z \in \mathbb{R}^S} \bar{F}(z)$$



## Example

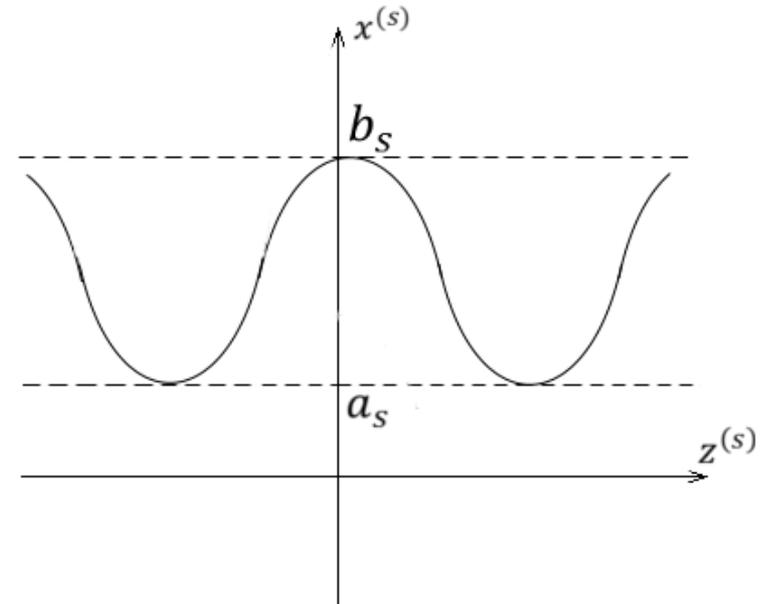
$$x^* \rightarrow F(x^*) = \min_{x \in \mathcal{D}_x} F(x)$$

$$F(x) = \sum_{s=1}^S (x^{(s)})^2$$

$$\mathcal{D}_x = \{x \in \mathbb{R}^S, \quad a_s \leq x^{(s)} \leq b_s, \quad s = 1, 2, \dots, S\}$$

$$x^{(s)} = a_s + (b_s - a_s) \sin^2 z^{(s)}$$

$$z^{(s)} \in \mathbb{R}^1 \rightarrow x^{(s)} \in [a_s, b_s]$$



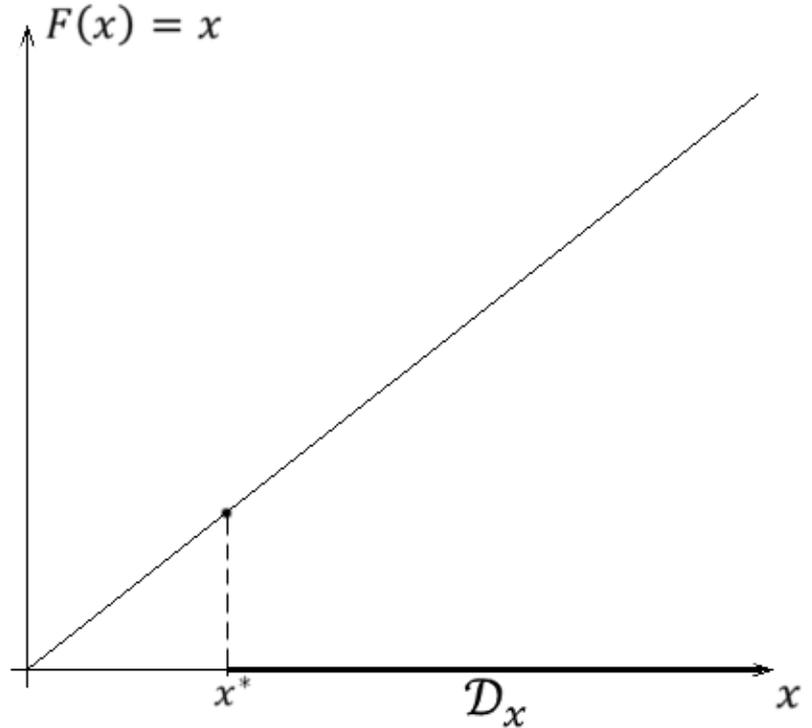
$$F(x) = \sum_{s=1}^S (x^{(s)})^2 \rightarrow \bar{F}(z) = \sum_{s=1}^S (a_s + (b_s - a_s) \sin^2 z^{(s)})^2 \quad z^* \rightarrow \bar{F}(z^*) = \min_{z \in \mathbb{R}^S} \bar{F}(z)$$



# Example

$$x^* \rightarrow F(x^*) = \min_{x \in \mathcal{D}_x} F(x)$$

$$F(x) = x \quad \mathcal{D}_x = \{x \in \mathbb{R}^1 \mid x \geq 1\}$$



$$x^* \rightarrow \min_{x \geq 1} x$$

$$x = z^2 + 1 = p(z)$$

$$z \in \mathbb{R} \rightarrow x \in [1, \infty)$$

$$\bar{F}(z) = z^2 + 1$$

$$z^* \rightarrow \min_{z \in \mathbb{R}} \bar{F}(z)$$

$$z^* \rightarrow \min_{z \in \mathbb{R}} (z^2 + 1)$$

$$(z^2 + 1)' = 2z = 0$$

$$z^* = 0$$

$$x^* = p(z^*) = (z^*)^2 + 1 = 1$$



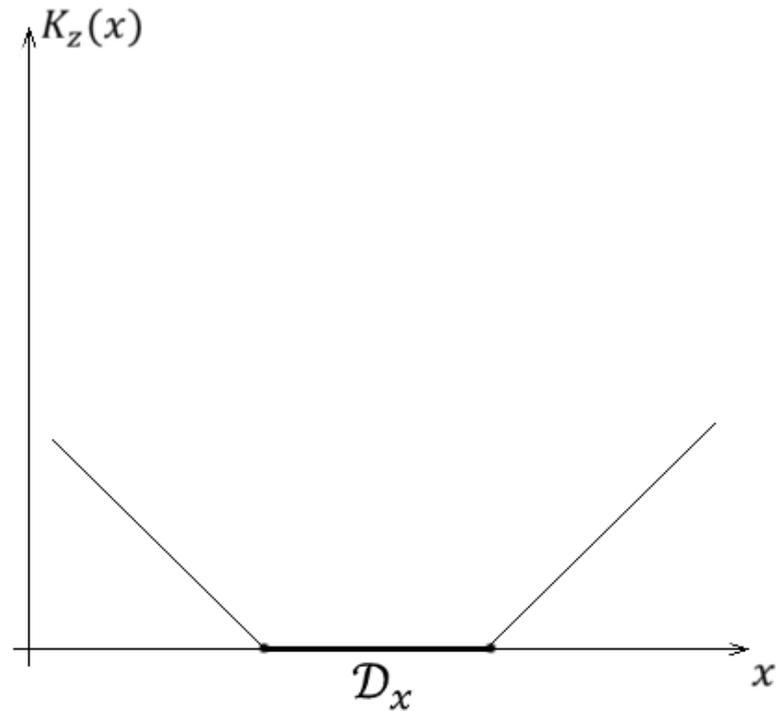
# Penalty function method

## Exterior penalty function

$$x^* \rightarrow F(x^*) = \min_{x \in \mathcal{D}_x} F(x)$$

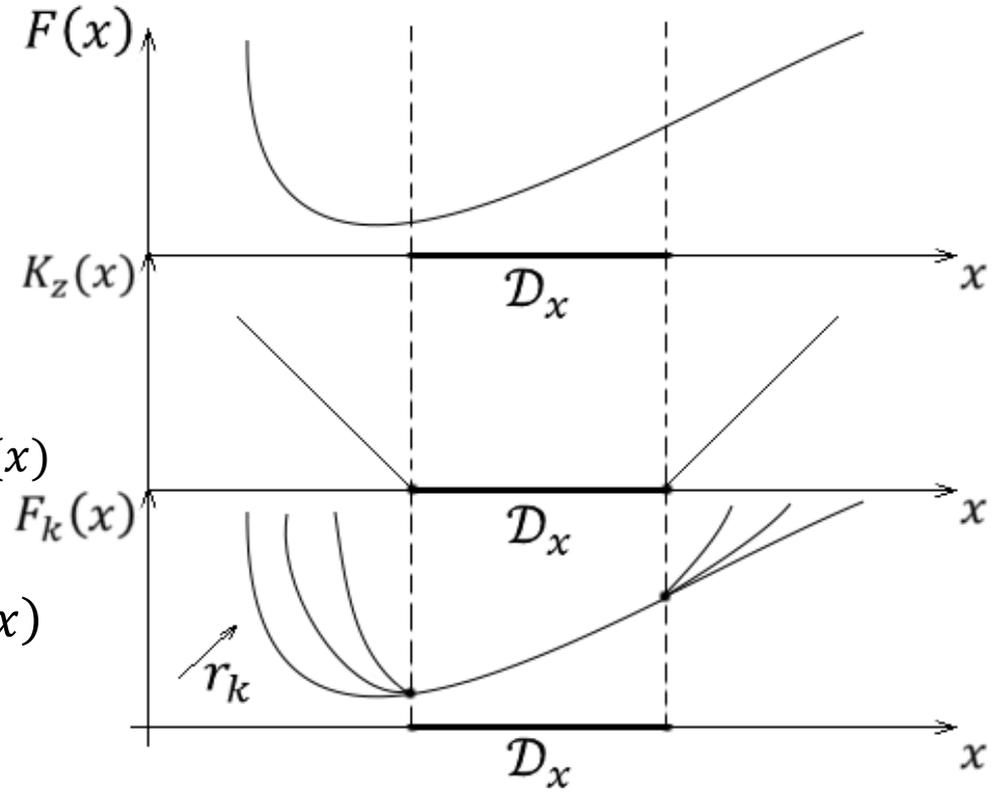
$$F_k(x) = F(x) + r_k K_z(x)$$

$$K_z(x) \begin{cases} = 0 & x \in \mathcal{D}_x \\ > 0 & x \notin \mathcal{D}_x \end{cases}$$





$$x^* \rightarrow F(x^*) = \min_{x \in \mathcal{D}_x} F(x)$$



$$F_k(x) = F(x) + r_k K_w(x)$$

$$x_k^* \rightarrow F(x_k^*) = \min_{x \in \mathcal{R}^S} F_k(x)$$

$$r_k > 0$$

$$\lim_{k \rightarrow \infty} r_k = \infty$$



# Example of exterior penalty function

$$x^* \rightarrow F(x^*) = \min_{x \in \mathcal{D}_x} F(x)$$

$$\mathcal{D}_x = \{x \in R^s, \varphi_l(x) = 0, \quad l = 1, 2, \dots, L, \quad \psi_m(x) \leq 0, \quad m = 1, 2, \dots, M\}$$

$$\varphi_l(x) \rightarrow K_{lz}(x) = (\varphi_l(x))^2, \quad l = 1, 2, \dots, L$$

$$\psi_m(x) \rightarrow K_{mz}(x) = (\max\{0, \psi_m(x)\})^2, \quad m = 1, 2, \dots, M$$

$$K_w(x) = \sum_{l=1}^L r_l (\varphi_l(x))^2 + \sum_{m=1}^M \rho_m \max\{0, \psi_m(x)\}^2$$



# Example

$$x^* \rightarrow F(x^*) = \min_{x \in \mathcal{D}_x} F(x)$$

$$F(x) = (x^{(1)})^2 + (x^{(2)})^2$$

$$\mathcal{D}_x = \{x \in \mathbb{R}^2, x^{(1)} + x^{(2)} - 2 = 0\}$$

$$F_k(x) = (x^{(1)})^2 + (x^{(2)})^2 + r_k(x^{(1)} + x^{(2)} - 2)^2$$

$$\nabla_x F_k(x) = 0_2$$

$$2x^{(1)} + 2r_k(x^{(1)} + x^{(2)} - 2) = 0$$

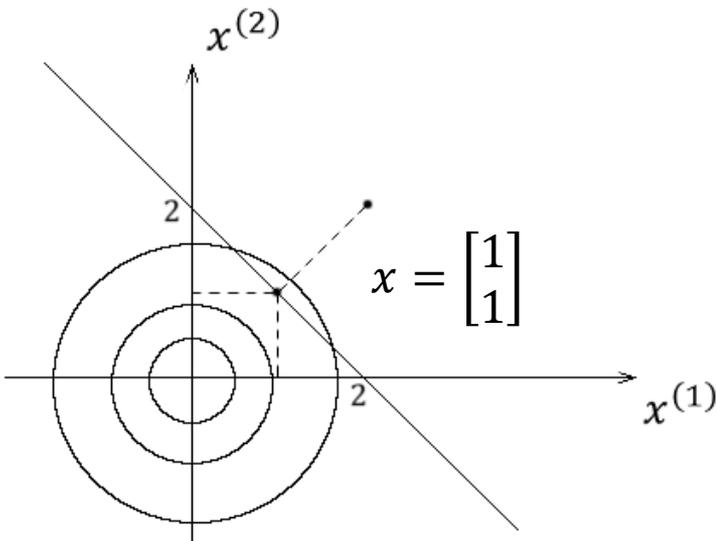
$$2x^{(2)} + 2r_k(x^{(1)} + x^{(2)} - 2) = 0$$

$$x^{(1)} = x^{(2)}$$

$$+2r_k(x^{(1)} + x^{(2)} - 2) = 0$$

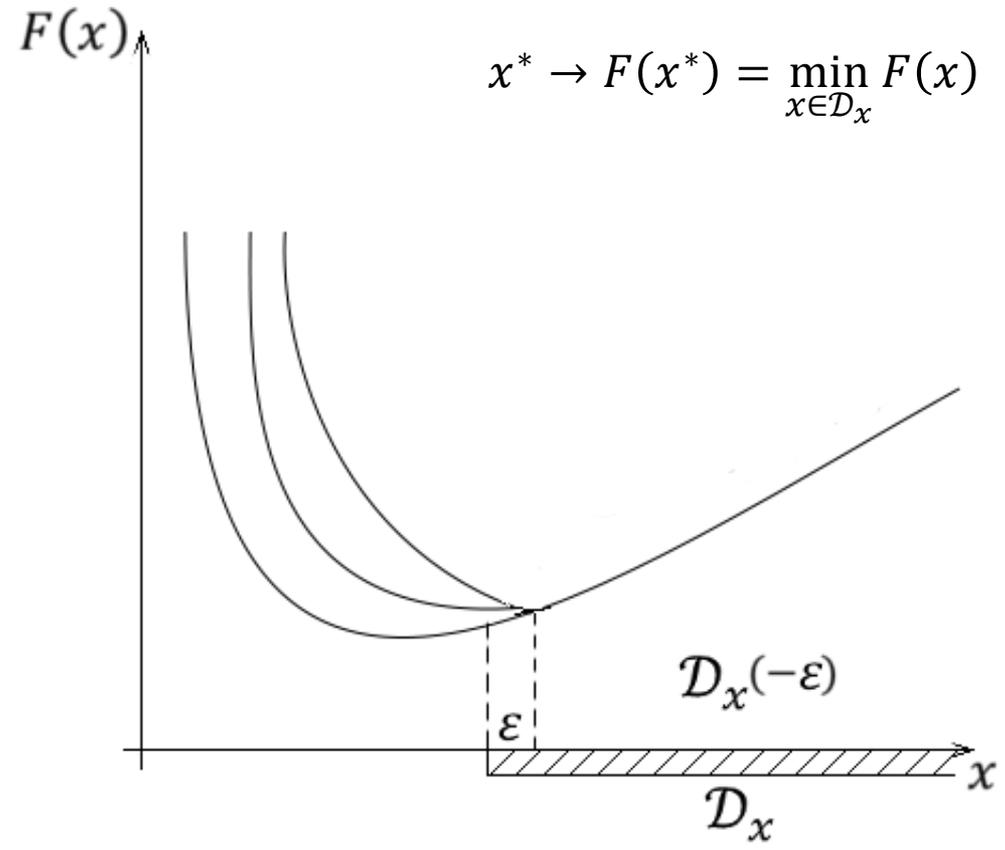
$$x^{(1)} = x^{(2)} = \frac{2r_k}{2r_k + 1}$$

$$x^{(1)} = x^{(2)} = \lim_{r_k \rightarrow \infty} \frac{2r_k}{2r_k + 1} = \lim_{r_k \rightarrow \infty} \frac{2}{2 + \frac{1}{r_k}} = 1, x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$





# Modifications





# Barrier function

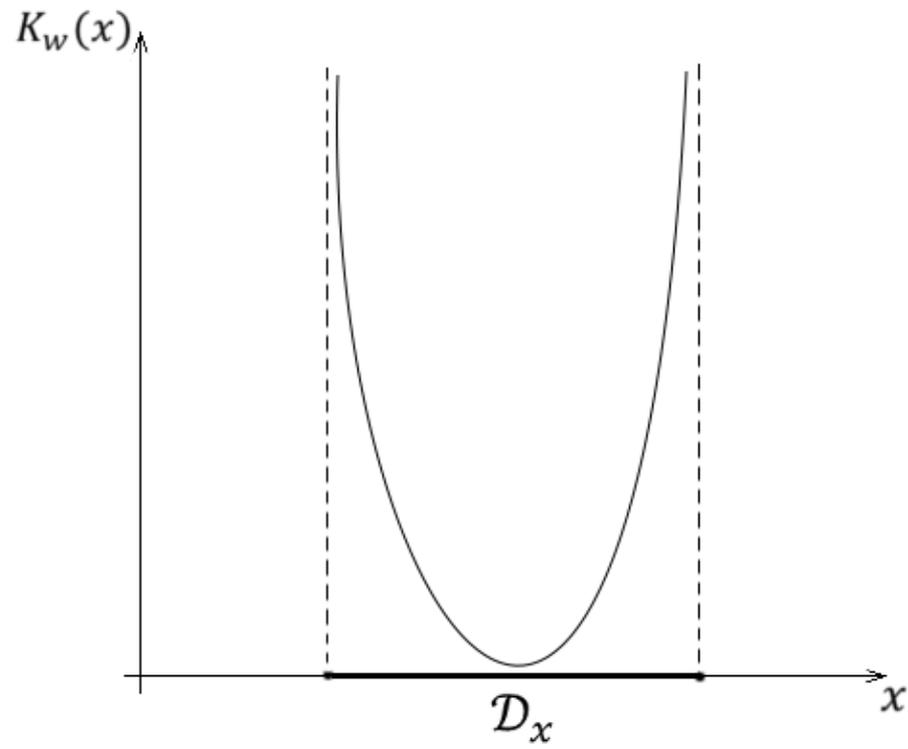
$$x^* \rightarrow F(x^*) = \min_{x \in \mathcal{D}_x} F(x)$$

$$F_k(x) = F(x) + r_k K_w(x)$$

$K_w(x)$  – such a function, that

$$\exists x_1, x_2, \dots, x_n \in \mathcal{D}_x \quad \lim_{k \rightarrow \infty} x_k = x \in \overline{\mathcal{D}_x}$$

$$\exists_k \quad K_w(x_k + 1) > K_w(x_k)$$

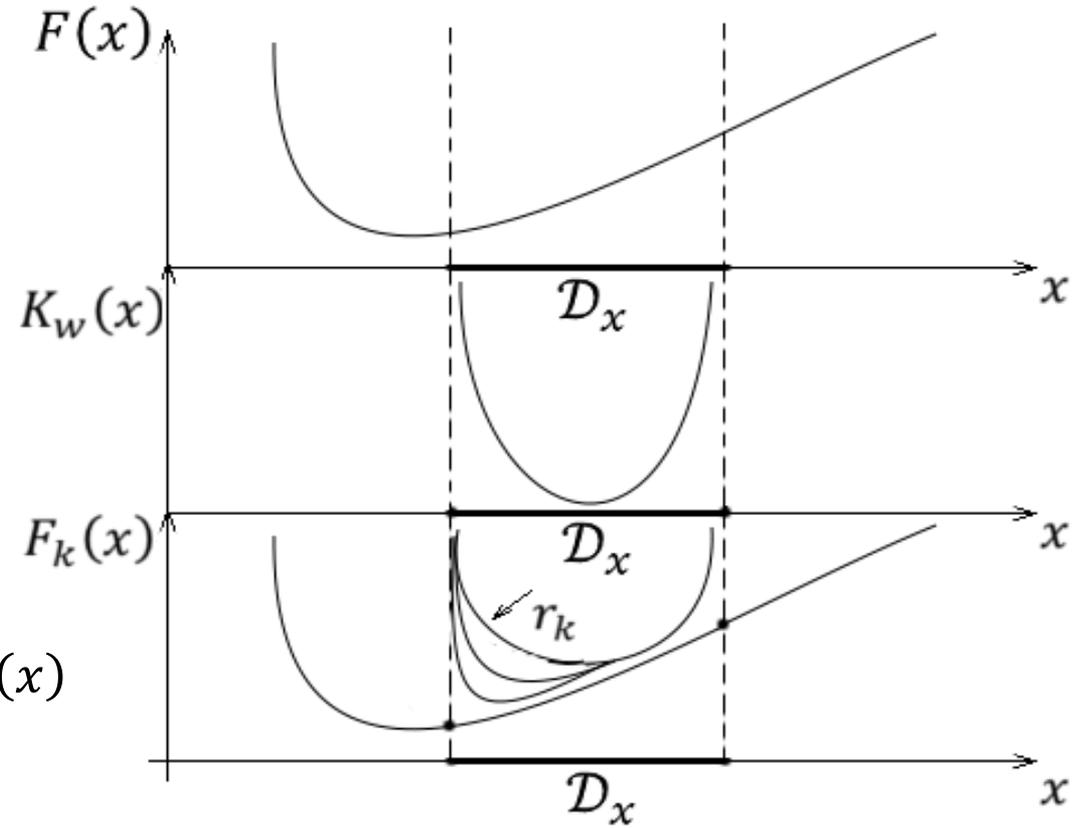




$$x^* \rightarrow F(x^*) = \min_{x \in \mathcal{D}_x} F(x)$$

$$F_k(x) = F(x) + r_k K_w(x)$$

$$x_k^* \rightarrow F(x_k^*) = \min_{x \in \mathcal{R}^S} F_k(x)$$



$$r_k > 0$$

$$\lim_{k \rightarrow \infty} r_k = 0$$



$$x^* \rightarrow F(x^*) = \min_{x \in \mathcal{D}_x} F(x)$$

$$\mathcal{D}_x = \{x \in R^s, \quad \psi_m(x) \leq 0, \quad m = 1, 2, \dots, M\}$$

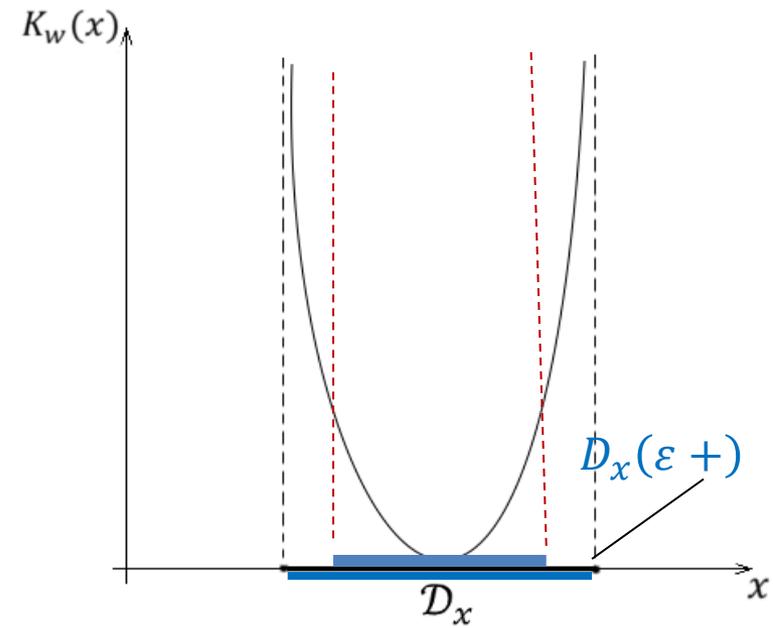
$$K_{Wm}(x) = \frac{-1}{\psi_m(x)}$$

$$K_w(x) = \sum_{m=1}^M r_m K_{Wm}(x) = \sum_{m=1}^M r_m \frac{-1}{\psi_m(x)}, \quad m = 1, 2, \dots, M$$



# Modifications

$$x^* \rightarrow F(x^*) = \min_{x \in \mathcal{D}_x} F(x)$$

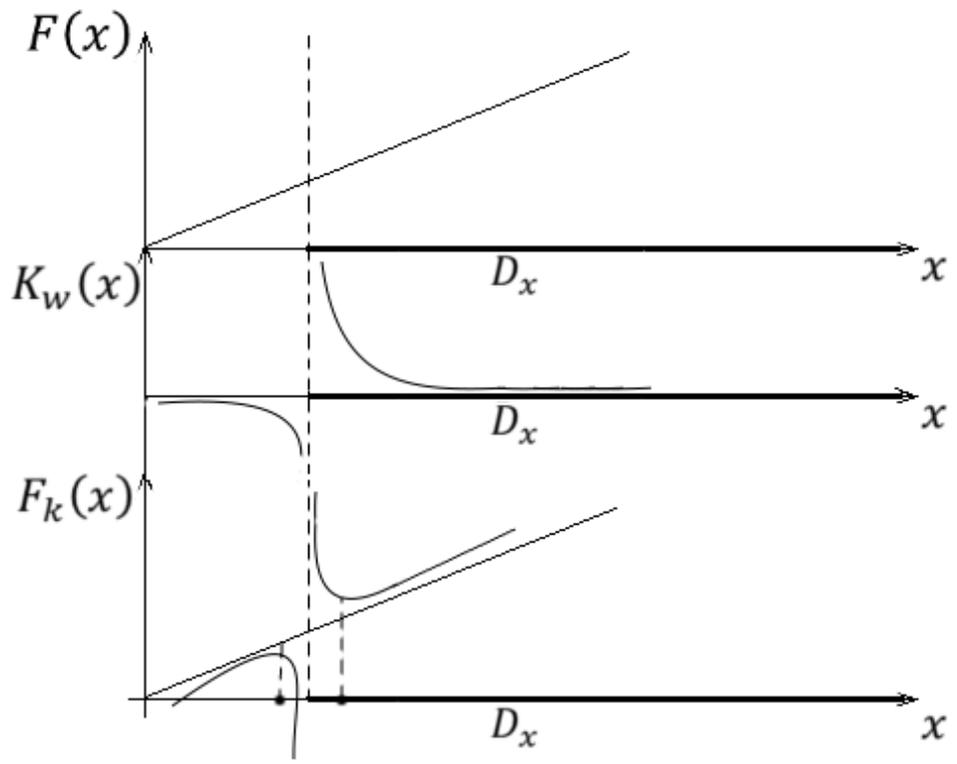




# Example

$$x^* \rightarrow F(x^*) = \min_{x \in \mathcal{D}_x} F(x)$$

$$F(x) = x; \quad \mathcal{D}_x = \{x \in \mathbb{R}^1, x \geq 1\} \equiv \{x \in \mathbb{R}^1, 1 - x \leq 0\}$$



$$K_w(x) = \frac{-1}{1-x} = \frac{1}{x-1}$$

$$F_k(x) = x + r_k \frac{1}{x-1}$$

$$F'(x) = 1 + \frac{-r_k}{(x-1)^2} = 0$$

$$x_k = 1 \pm \sqrt{r_k}$$

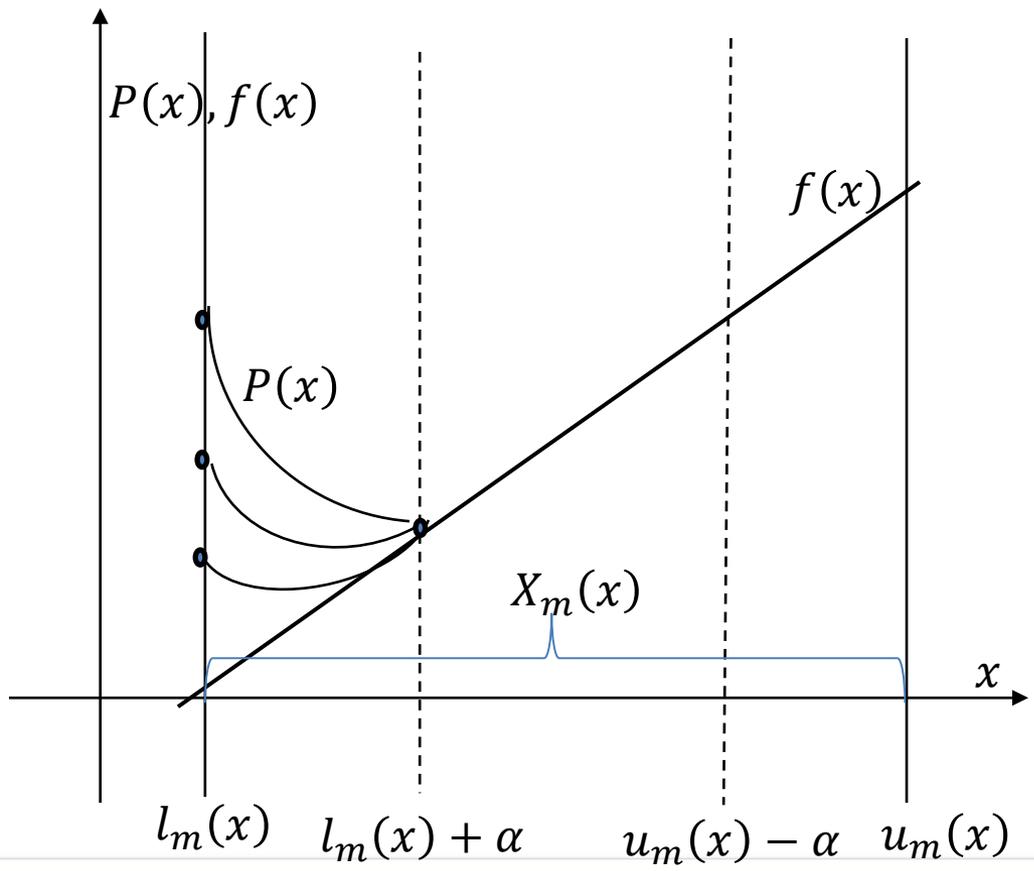
$$x_k = 1 - \sqrt{r_k} \notin \mathcal{D}_x$$

$$x_k = 1 + \sqrt{r_k} \in \mathcal{D}_x$$

$$\lim_{n \rightarrow \infty} x_n = 1$$



# Rozebrock Method





# Rozebrock Method

$$x^* \rightarrow F(x^*) = \min_{x \in \mathcal{D}_x} F(x)$$

$$D_x = \{l_m(x) \leq X_m(x) \leq u_m(x) \quad m = 1, 2, \dots, M\}$$

$$l_m(x) \leq X_m(x) \leq l_m(x) + \alpha$$

$$u_m(x) \leq X_m(x) \leq u_m(x) - \alpha$$

$$P(x) = f(x) - (f(x) - f^*)(3\mu - 4\mu^2 + 2\mu^3)$$

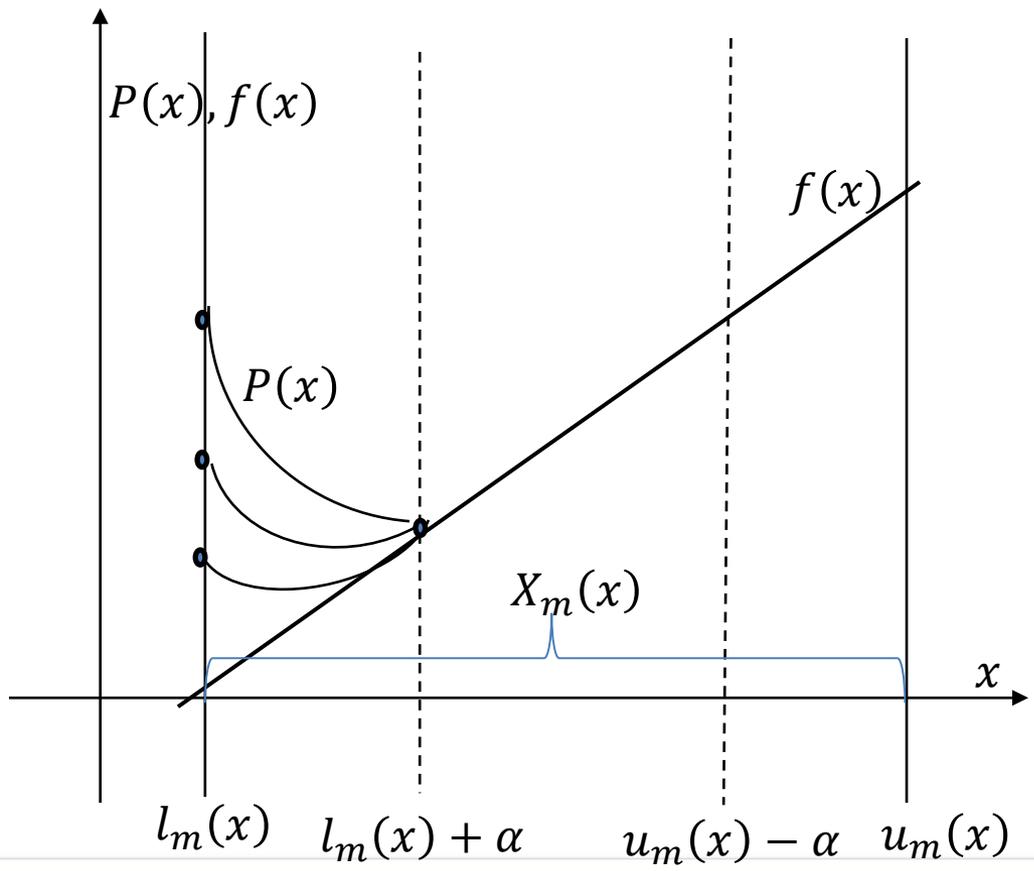
$$\mu = \frac{l_m(x) + \alpha - X_m(x)}{\alpha} \quad \text{or} \quad \mu = \frac{X_m(x) - u_m(x) + \alpha}{\alpha}$$

$$l_m(x) + \alpha \leq X_m(x) \leq u_m(x) - \alpha$$

$$P(x) = f(x)$$



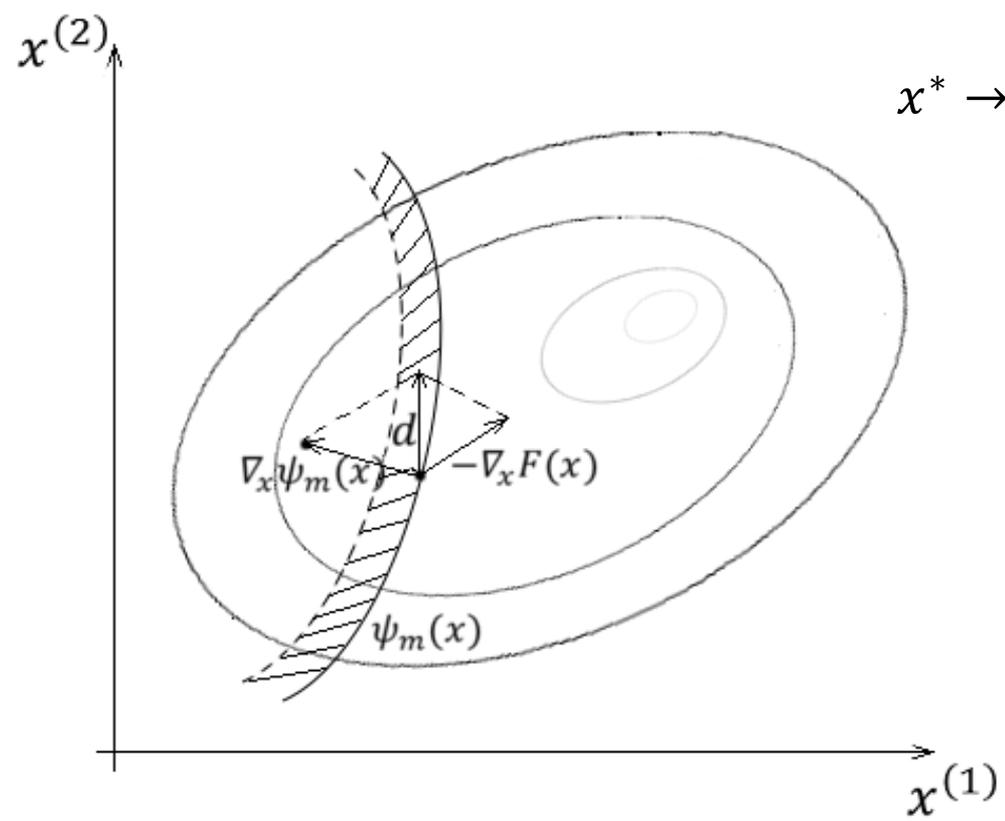
# Rozebrock Method





# Feasible directions method

$$x^* \rightarrow F(x^*) = \min_{x \in \mathcal{D}_x} F(x)$$

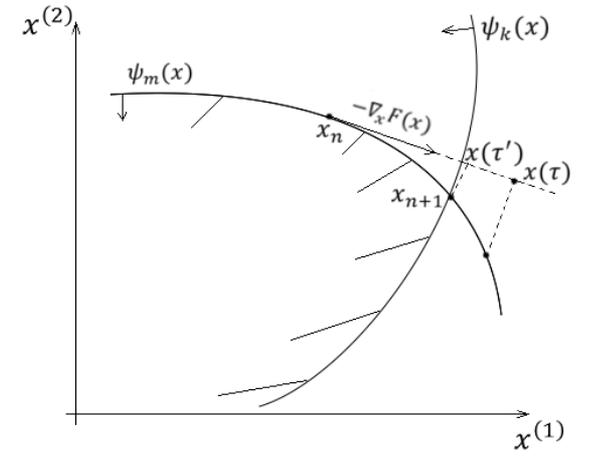
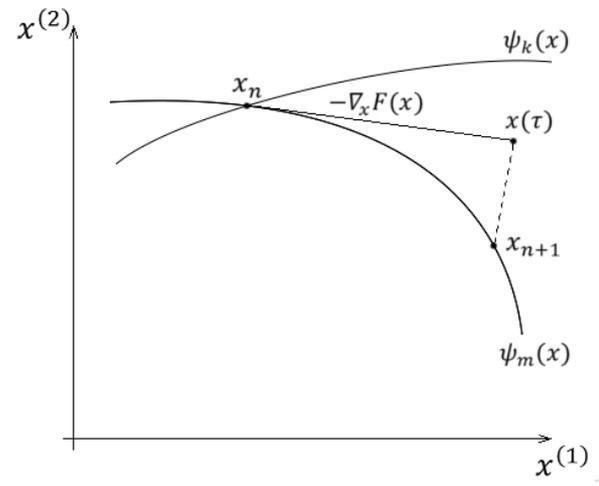
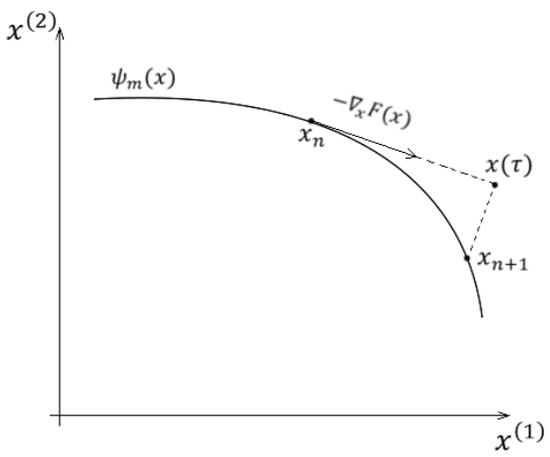


$$d = \frac{\nabla_x \psi_m(x)}{\|\nabla_x \psi_m(x)\|} - \frac{\nabla_x F(x)}{\|\nabla_x F(x)\|}$$

$$x: \psi(x) - \delta \leq 0$$



# Gradient projection method of Rosen

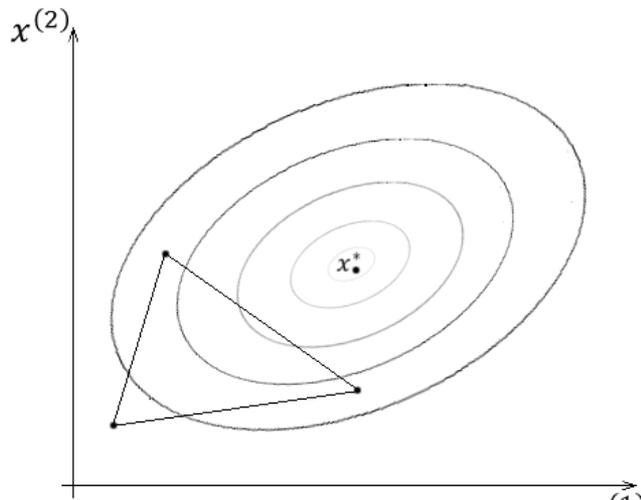




# Complex Method (Box)

## Nelder-Mead Method (simplex)

$x_1 x_2 \dots x_K$  - complex in  $S$  - domine space  $K \geq S + 1$



$$x_H \rightarrow F(x_H) = \max_{1 \leq s \leq K} F(x_s)$$

$$\bar{x} = \frac{1}{S} \sum_{s=1, s \neq H}^K x_s$$

$$x^* = (1 + \alpha)\bar{x} - x_H$$

$$D_x: l_s \leq x^{(s)} \leq u_s \quad s = 1, 2, \dots, S$$

$$l_m \leq x_m(x) \leq u_m \quad m = 1, 2, \dots, M$$

$x_k = l_k + r_k |u_k - l_k|$ ,  $r_k$  - random number  $\in [0, 1]$ ,  $k = 1, 2, \dots, K$

If  $x_k$  is not element of  $D_x$  then move point in the direction of centroid of accepted points



# Complex Method

Data:  $x_0, c, \varepsilon, K, \alpha$  recommended  $K = 2S, \alpha = 1.3$

Step 0:  $x_1 x_2 \dots x_K \in D_x$  - initial complex,  $n = 0$

Step 1: Determine the radius of the minimum ball containing the complex -  $\rho_{min}$

Step 2: Review if  $\rho_{min} < \varepsilon$  If yes then stop, otherwise go to step 3

Step 3:  $x_H \rightarrow F(x_H) = \max_{1 \leq s \leq S+1} F(x_s),$

$\bar{x} = \frac{1}{S} \sum_{\substack{s=1 \\ s \neq H}}^K x_s$  review if  $\bar{x} \in D_x$  if not add point to the complex

Step 4:  $x^* = (1 + \alpha)\bar{x} + \alpha x_H$

Step 5: Review if  $x^* \in D_x$  if yes then go to step 7, otherwise

Step 6:  $x^* = \bar{x} - (\bar{x} - x^*)/2$  until  $x^* \in D_x$

Step 7: If  $F(x^*) < F(x_H)$  go to step 2  
otherwise go to step 6



# Random search - Down Hill method

Data:  $F(x), x_0, D_x, N$

Step 0:  $n=0, x^* = x_n$

Step 1: Generate point  $x_{n+1}$  in the set  $D_x$  with unity probability density

Step 2: IF  $F(x_{n+1}) < F(x^*)$  THEN  $x^* = x_{n+1}$

Step 3: IF  $n < N$  THEN  $n = n + 1$  GO TO STEP 1

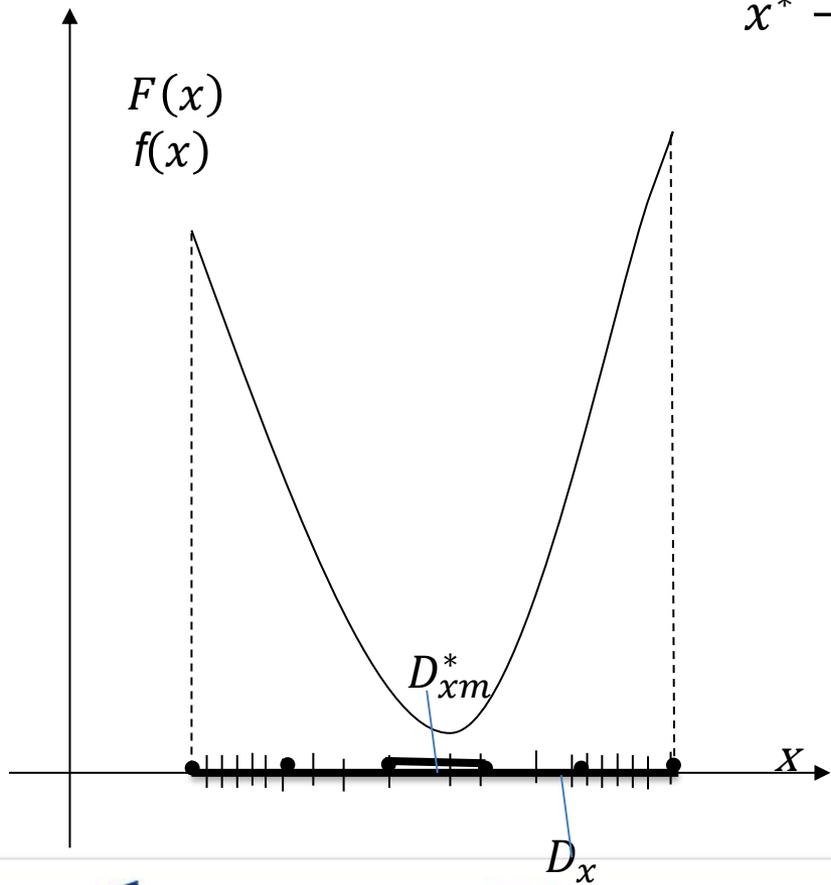
Step 4:  $x^* = x_N$



# Random search

$$x^* \rightarrow F(x^*) = \min_{x \in D_x} F(x)$$

$$f(x) = \frac{F(x)}{\int_{D_x} F(x) dx}$$





# Random search

Data:  $F(x), D_x, \varepsilon, N, M$

Step 1: Generate  $N$  point in set  $D_x$  with probability density

$$f(x) = \frac{F(x)}{\int_{D_x} F(x) dx}$$

Step 2: Divide set  $D_x$  for  $M$  disjoint sets such that

$$D_x = \bigcup_{m=1}^M D_{xm}, \quad \|D_{xm}\| = \frac{1}{M} \|D_x\|$$

Step 3. Count points in all sets  $D_{xm}$

$N_m$  – number of points in the set  $D_{xm}$

Step 4. For the next division choose such set that

$$D_{xm}^* \rightarrow N_m^* = \min_{1 \leq m \leq M} \{N_m\} \text{ if } \|D_{xm}\| < \varepsilon \text{ stop, } \forall \text{ point } \in D_{xm}^* \text{ solution}$$

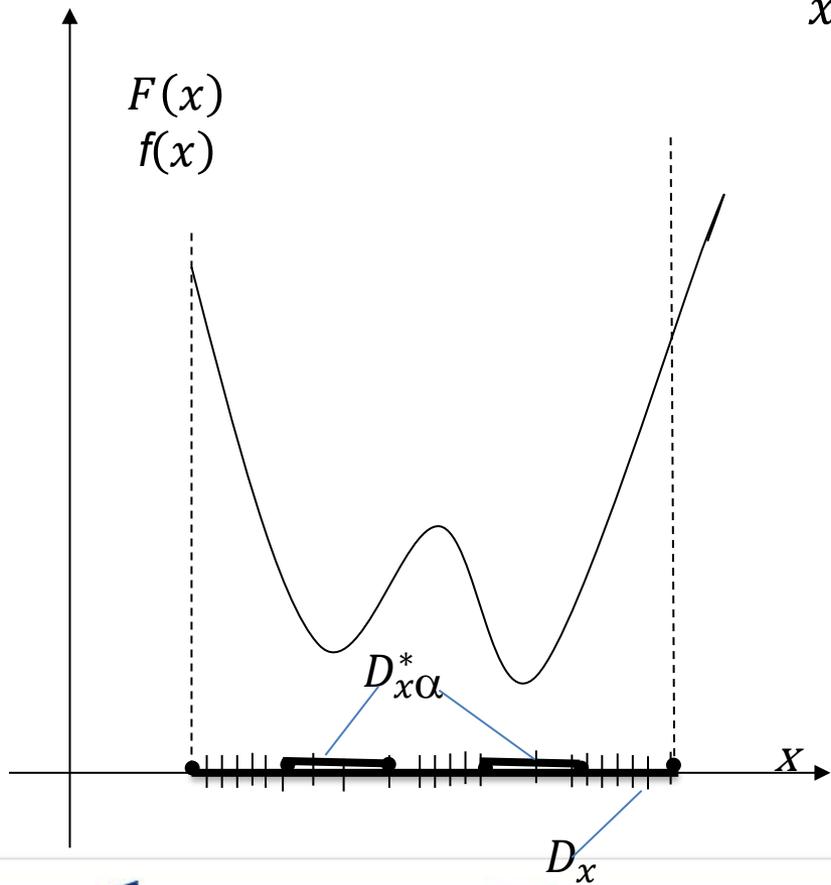
Otherwise i.e.  $\|D_{xm}^*\| \geq \varepsilon$  in the place  $D_x := D_{xm}^*$  and go to step 1.



# Random search

$$x^* \rightarrow F(x^*) = \min_{x \in D_x} F(x)$$

$$f(x) = \frac{F(x)}{\int_{D_x} F(x) dx}$$





# Random search

Data:  $F(x), D_x, \varepsilon, N, M$

Step 1: Generate  $N$  point in set  $D_x$  with probability density

$$f(x) = \frac{F(x)}{\int_{D_x} F(x) dx}$$

Step 2: Divide set  $D_x$  for  $M$  disjoint sets such that

$$D_x = \bigcup_{m=1}^M D_{xm}, \quad \|D_{xm}\| = \frac{1}{M} \|D_x\|$$

Step 3. Count points in all sets  $D_{xm}$

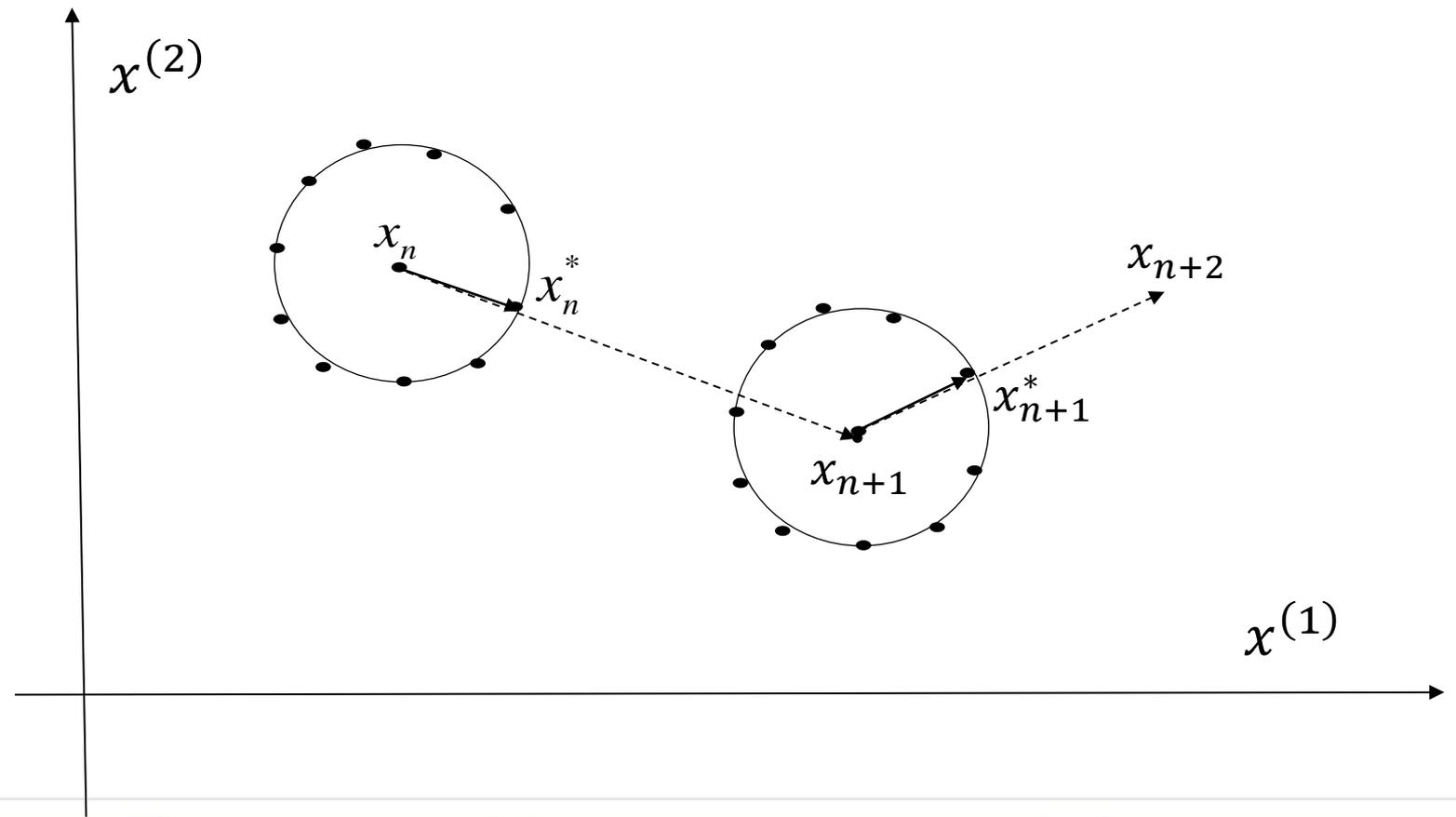
$N_m$  – number of points in the set  $D_{xm}$

Step 4. For the next division choose such set that

$$D_{xm\alpha}^* \rightarrow N_{m\alpha}^* \leq \alpha, \quad D_{x\alpha}^* = \bigcup_m D_{xm\alpha}^* \text{ if } \|D_{x\alpha}^*\| < \varepsilon \text{ stop, } \forall \text{ point } \in D_{x\alpha}^* \text{ solution}$$

Otherwise i.e.  $\|D_{xm}^*\| \geq \varepsilon$  in the place  $D_x := D_{xm}^*$

and go to step 1.





# Random search – random direction choice

Data:  $F(x), x_0, r, N, \varepsilon$

Step 0:  $n=0, \quad x^* = x_n$

Step 1: Generate  $N$  points with unity probability density on the cycle around point  $x_n$

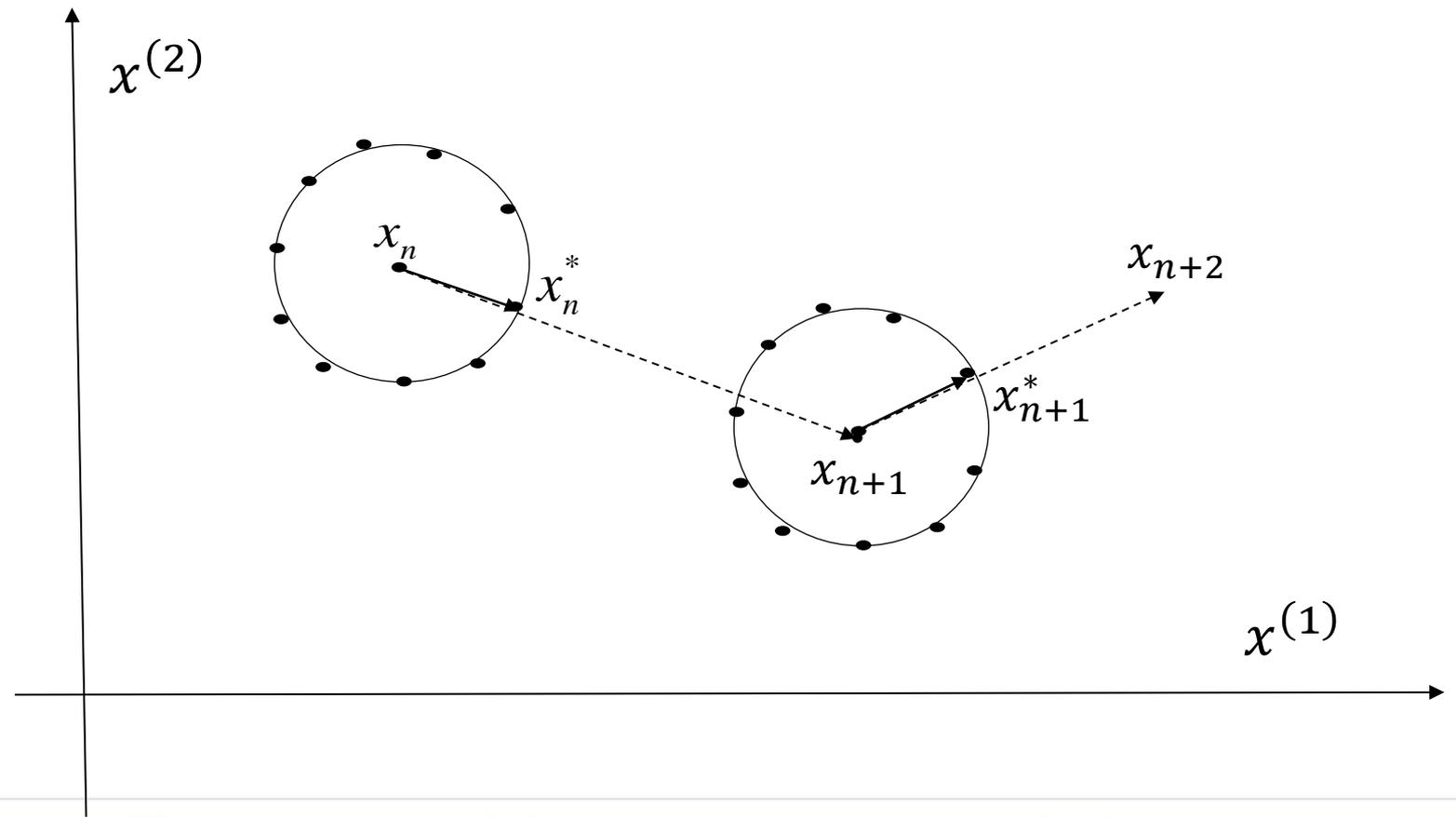
Step 2: Select from generated points point  $x_n^*$  for which the goal function obtain minimum

Step 3: determine direction 
$$d = \frac{x_n^* - x_n}{\|x_n^* - x_n\|}$$

Step 4: Determine point  $x_{n+1}$  as minimum in direction  $d$  from point  $x_n$

Step 5: IF minimum THEN STOP  $x_n \approx x^*$  ELSE

$n=n+1$  GO TO STEP 1





# *Nature-Inspired Algorithms*

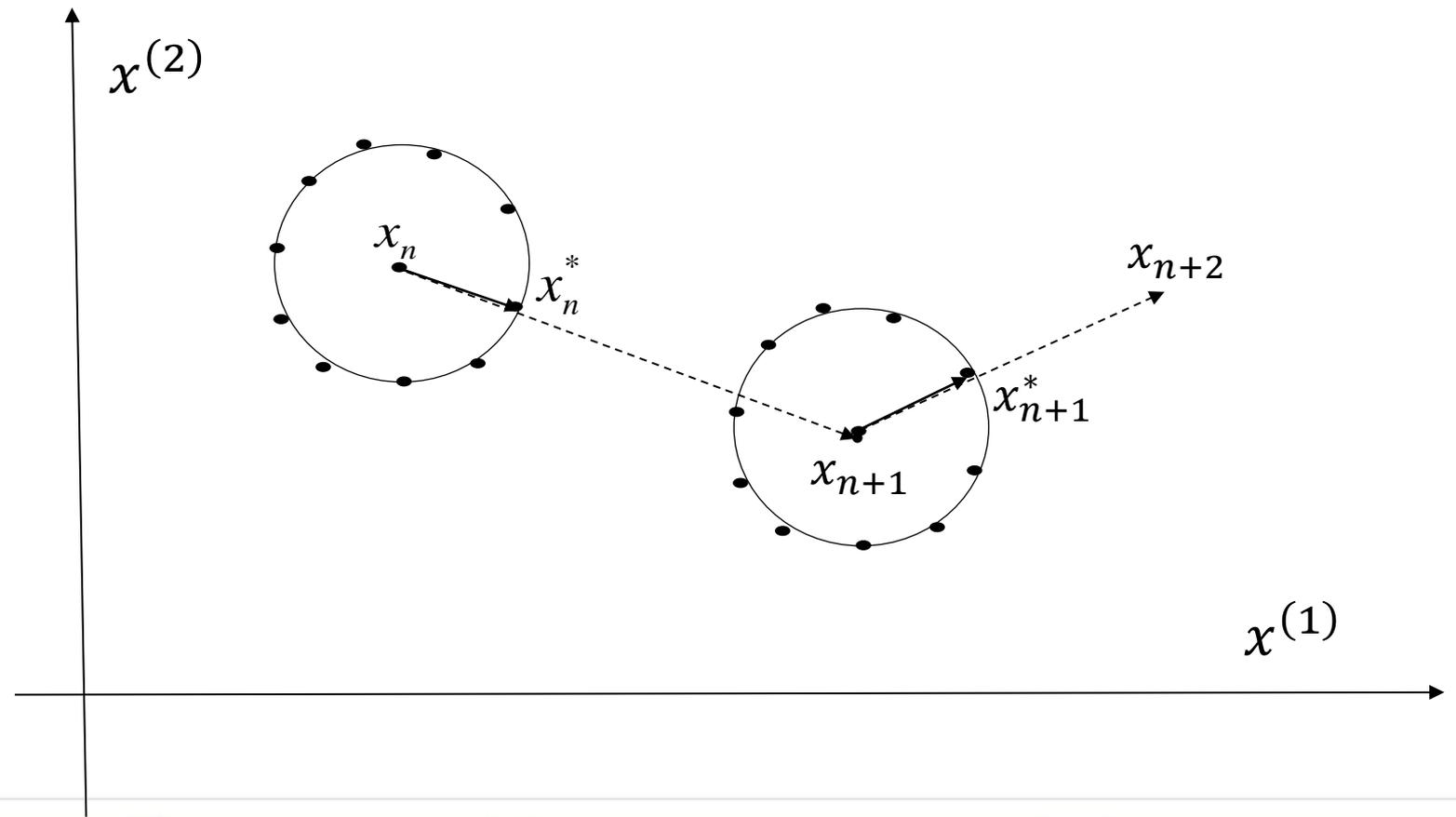
## Bibliography

- *Clever Algorithms: Nature-Inspired Programming Recipes*, Jason Brownlee
- *Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning*, Shumeet Baluja, School of Computer Science Carnegie Mellon University Pittsburgh, 1994
- The Bees Algorithm – A Novel Tool for Complex Optimisation Problems, D.T. Pham, A. Ghanbarzadeh, E. Koç et. al, Cardiff University, 2006
- Zastosowanie Algorytmów Rojowych do Optymalizacji Parametrów w Modelach Układów Regulacji, Mirosław Tomera, Zeszyty Naukowe Wydziału Elektrotechniki i Automatyki Politechniki Gdańskiej Nr 46, 2015
- Automatic Tuning of a Retina Model for a Cortical Visual Neuroprosthesis Using a Multi-Objective Optimization Genetic Algorithm, Antonio Martínez-Álvarez, Rubén Crespo-Cano, Ariadna Díaz-Tahoces et. al., International Journal of Neural Systems 26/7, 2016



# Ant algorithm

- Ants search the space randomly.
- If they come across "food", i.e. a solution – they return to the starting point, leaving a trace of pheromone.
- Encountering a pheromone trace, they follow the designated path, thereby increasing the amount of pheromone.
- The pheromone evaporates over time. Long distances lose the intensity of the pheromone faster.





# Bee algorithm

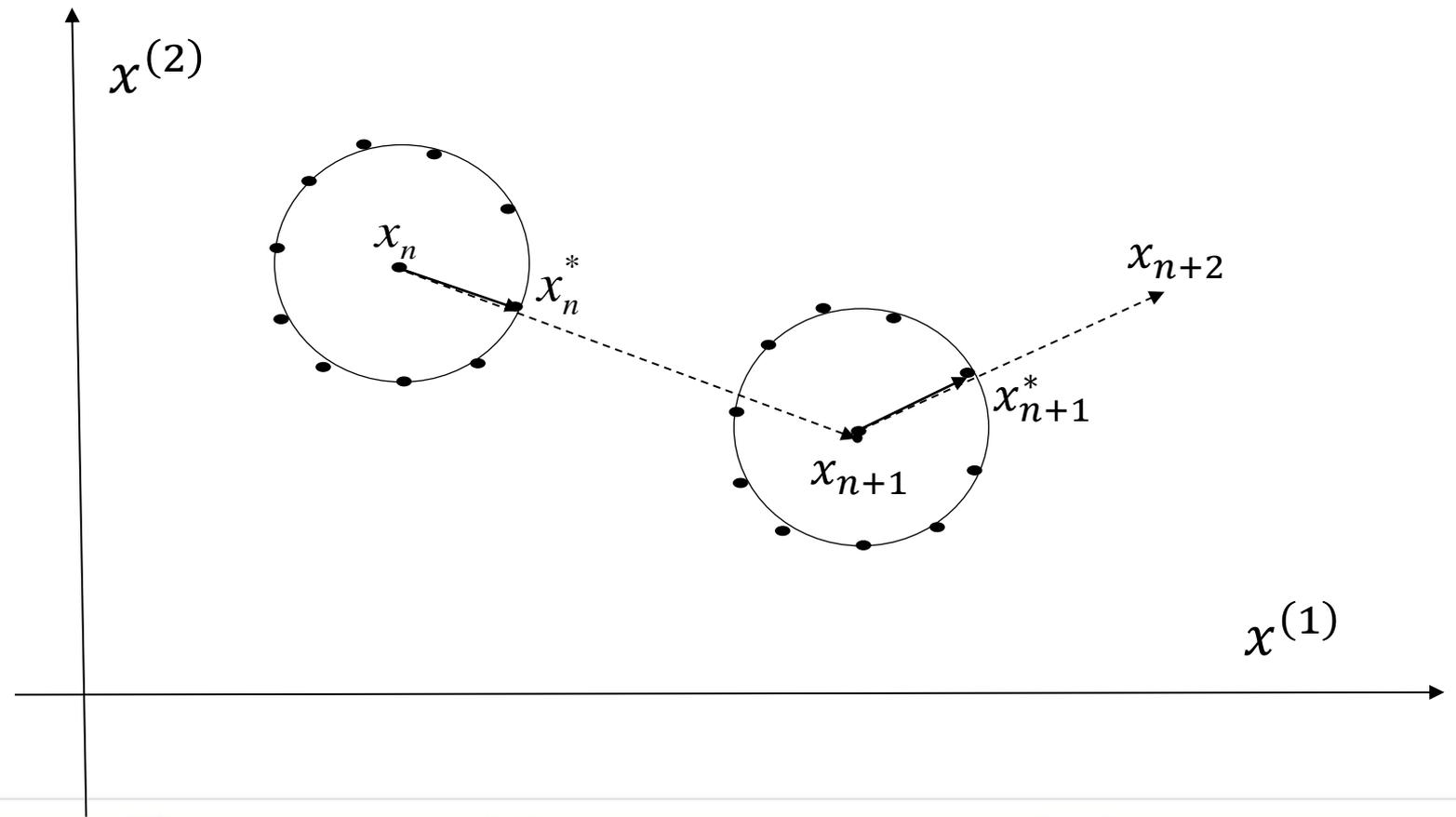
We distinguish groups:

- **Scouts** – search the solutions randomly
- **Employees** – they bring information about the quality of the solutions to which they are currently assigned.
- **Observers** – choose the next place based on information from *employees*.



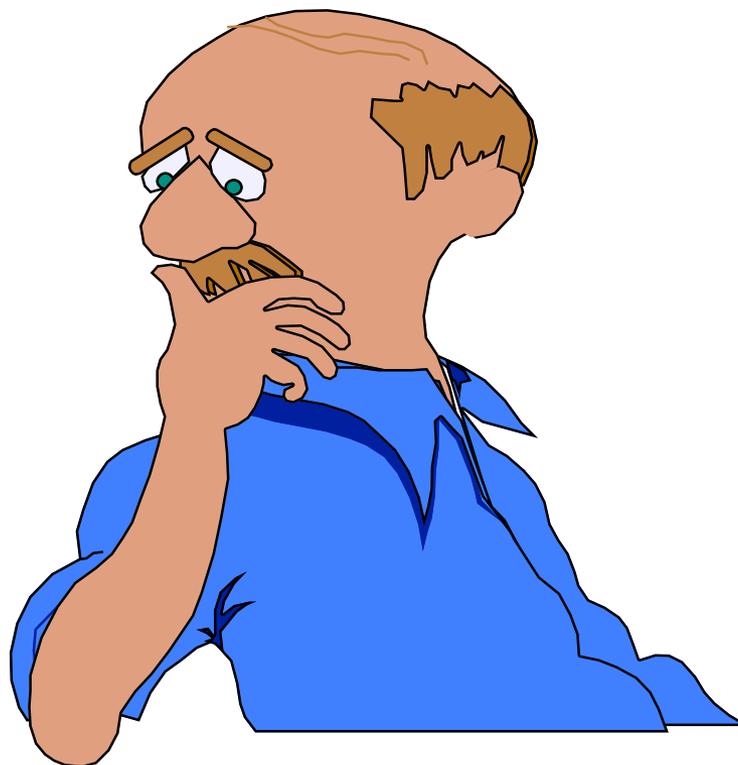
# Bee algorithm

- Scouts are sent to random places.
- In the vicinity of the found good values, more individuals are sent to study the local optimum.
- In each iteration, scouts are sent to further explore the solution space.





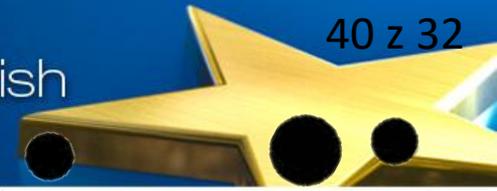
# Thank you for attention



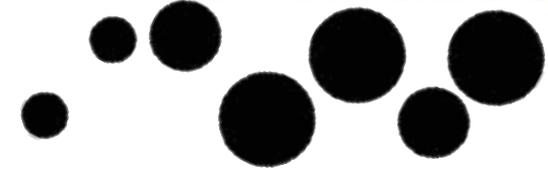


# Particle Swarm Algorithm

- The goal of the algorithm is to gather all particles around the optimum of multidimensional hyperspace.
- Molecules are assigned a random position and a low initial velocity.
- The motion of molecules is simulated based on their velocity, position, best known solution, and global optimum.
- The molecules converge into the optimum/optima.



# Algorytm Roju Cząstek



Celem algorytmu jest zgromadzenie wszystkich cząstek wokół optimum wielowymiarowej hiperprzestrzeni.

- Cząsteczkom przypisuje się losowe położenie i niską prędkość początkową.
- Symulowany jest ruch cząsteczek na podstawie jej prędkości, położenia, najlepszego znanego rozwiązania oraz optimum globalnego.
- Cząsteczki zbiegają do optimum/optimów.