Electromagnetic Design of Accelerator Magnets and ROXIE User's Course

Mathematical Optimization Techniques

Stephan Russenschuck, CERN, 2022

Mathematical Formulation of Optimization Problems

$$X \subseteq \mathbb{R}^n \qquad (x_1, x_2, ..., x_n)^T \in X$$

$$\min\{f(\mathbf{x})\} \qquad f : X \to \mathbb{R}$$

Subject to

$$g_i(\mathbf{x}) \leq 0, \qquad i = 1, 2, ..., m,$$

$$h_j(\mathbf{x}) = 0, \qquad j = 1, 2, ..., p,$$

$$x_{l,\text{lower}} < x_l < x_{l,\text{upper}}, \qquad l = 1, 2, ..., n$$

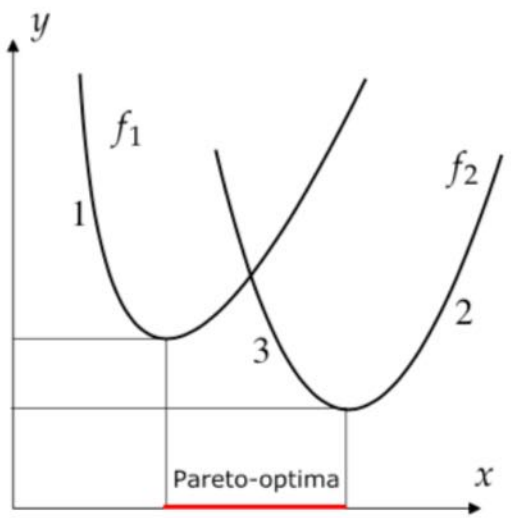
Pareto Optimality

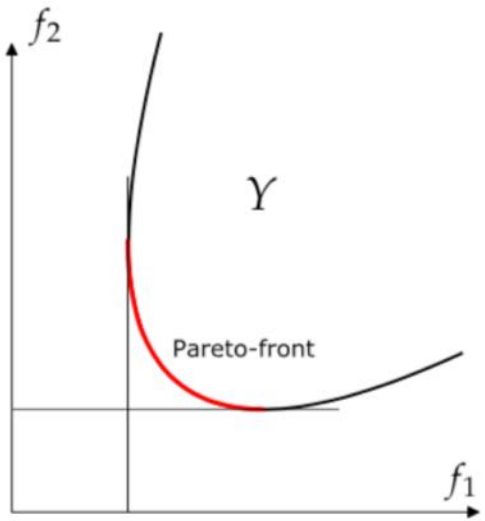
$$\mathrm{MIN}\,\{\mathbf{f}(\mathbf{x})\} = \mathrm{MIN}\,\{f_1(\mathbf{x}), f_2(\mathbf{x}), .., f_K(\mathbf{x})\}$$

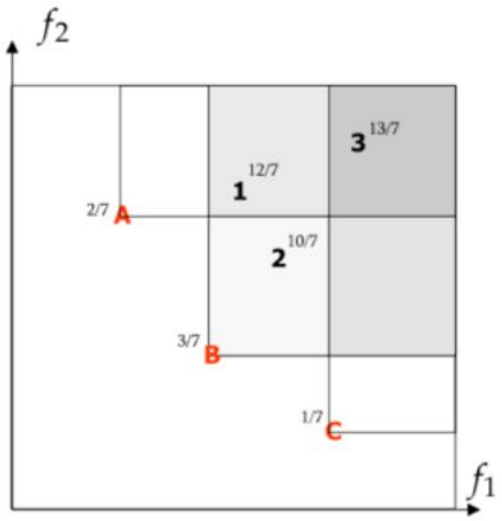
A Pareto optimal solution X* is given if there exists no solution with

$$f_k(\mathbf{x}) \leq f_k(\mathbf{x}^*) \qquad \forall\, k \in [1, K],$$

$$f_k(\mathbf{x}) < f_k(\mathbf{x}^*) \qquad \text{for at least one } k \in [1, K]$$







Real World Optimization Problems

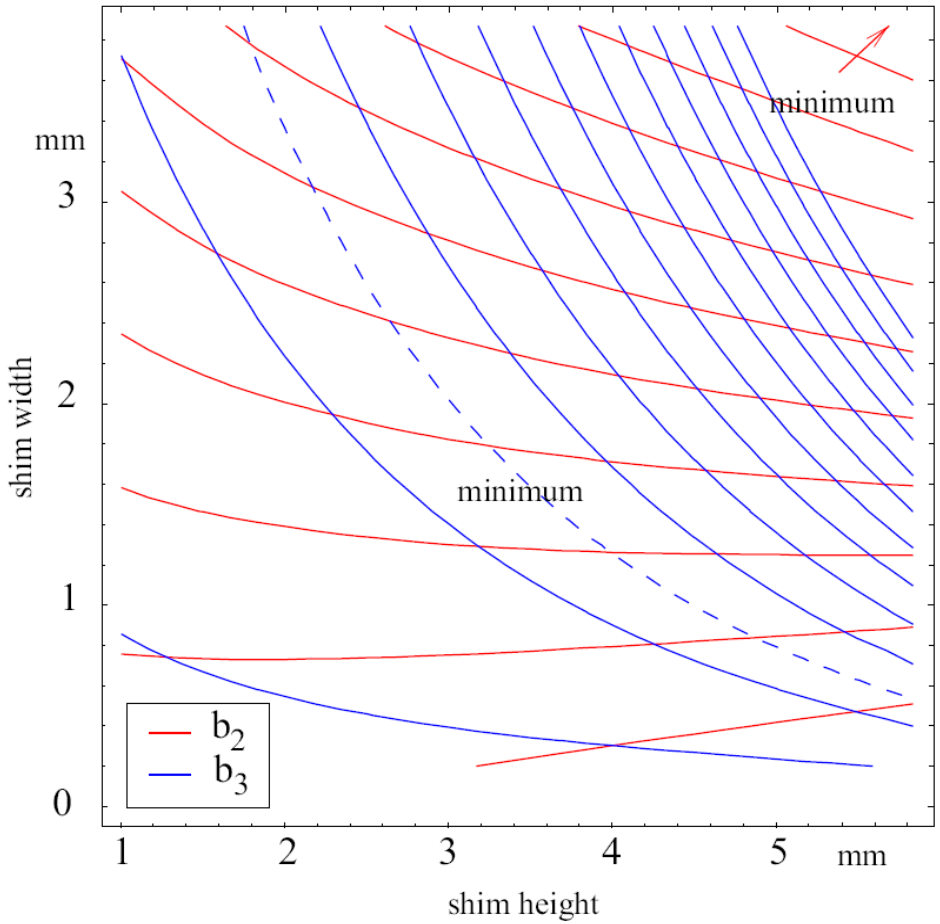
- There are only Pareto-optimal solutions
  - Decision making
  - Treatment of nonlinear constraints
  - Optimization algorithms
- The objective conflict is the characteristic of real world optimization problems
- Fuzzy objectives in the concept phase

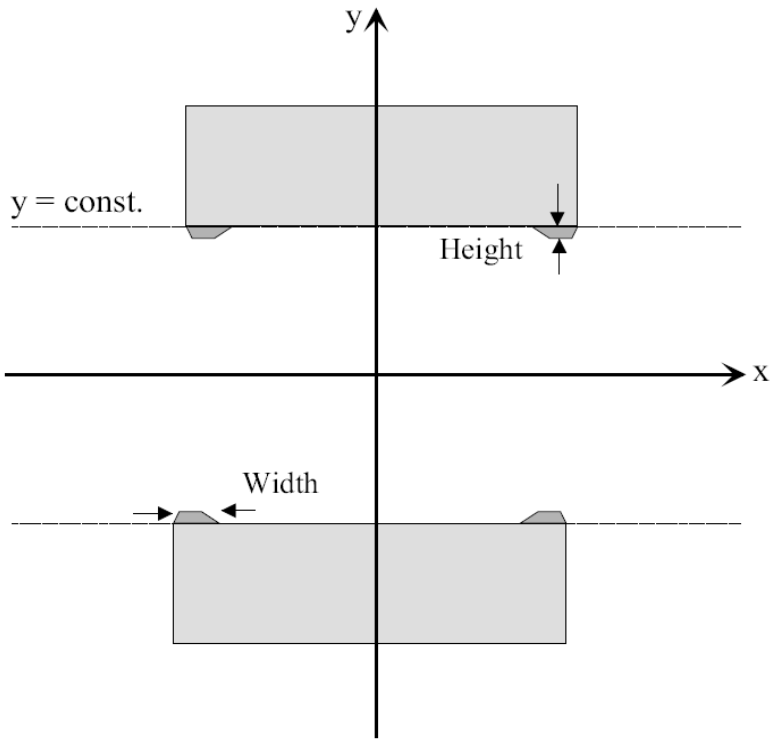
Optimization Problems in Magnet Design

	Decision-making method	Optimization algorithm	Typical no. of design variables	Typical no. of function evaluations
Conceptual coil design	Objective weighting	Genetic algorithms	15 (60 Bits)	7000
Coil cross section	Distance function	EXTREM	8	250
Conceptual yoke design (material distribution)	Distance function	Genetic algorithms	170 (170 Bits)	$20\,000 \cdot 4$ FEM
Yoke cross section	Objective weighting	EXTREM	10	$100 \cdot 6$ FEM
Sensitivity analysis	Lagrange multiplier estimation Payoff tables	Davidon–Fletcher–Powell algorithm EXTREM	10	10 $6 \cdot 100 \cdot 6$ FEM
3D Coil end optim.	Objective weighting	EXTREM	5	100
Inverse field calc.	Distance function	Levenberg–Marquard	50	700

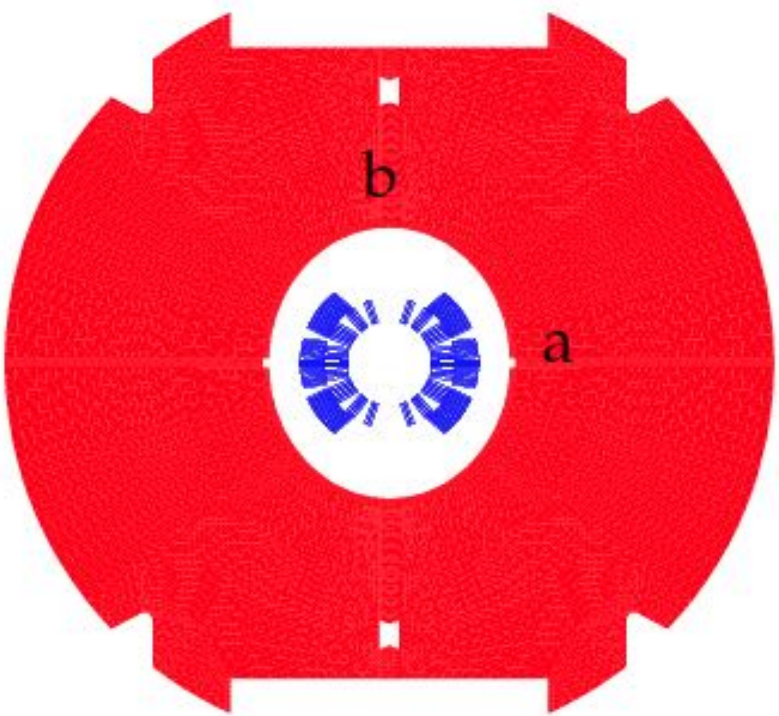
Load-line

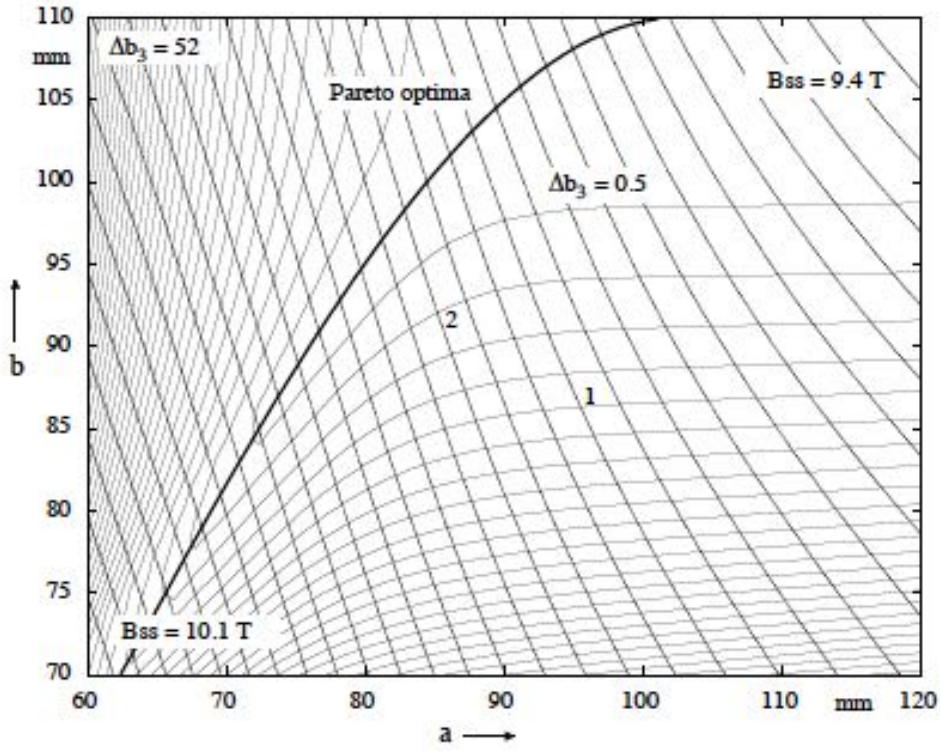
Objective Conflict





Objective Conflict Superconducting Magnets



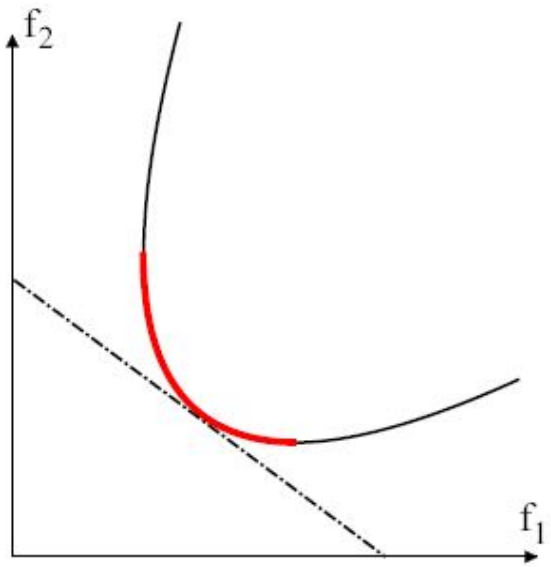


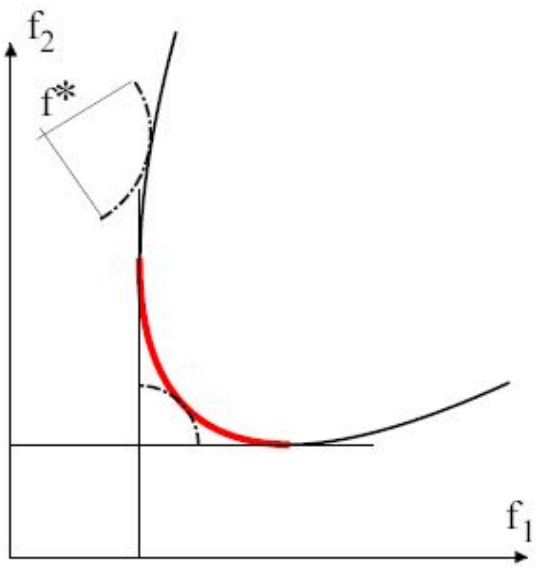
Objective Weighting, Distance Func., Constraint Form.

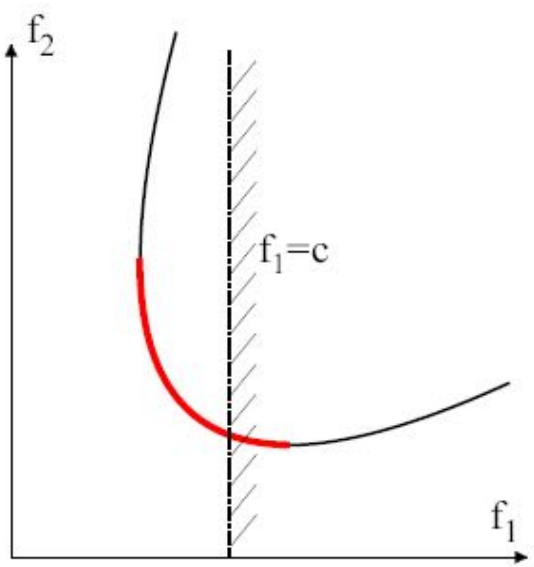
$$\min \left\{ u(\mathbf{f}(\mathbf{x})) := \sum_{k=1}^{K} t_k \, f_k(\mathbf{x}) \mid \mathbf{x} \in M \right\}$$

$$\min \left\{ \| \mathbf{z}(\mathbf{x}) \|_2^2 := \sum_{k=1}^{K} (t_k (f_k^*(\mathbf{x}) - f_k(\mathbf{x})))^2 \mid \mathbf{x} \in M \right\}$$

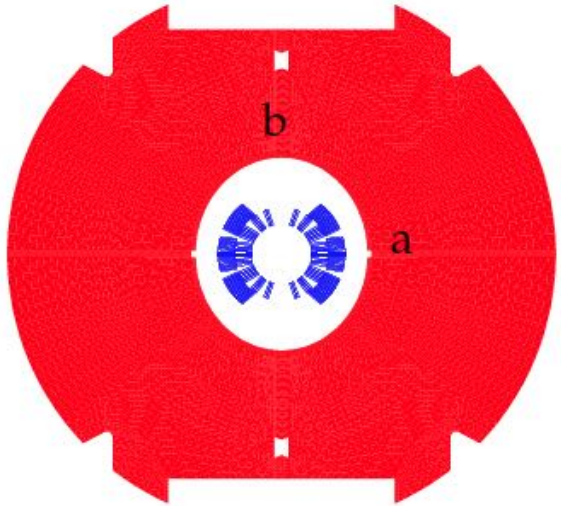
$$\min \{ f_i(\mathbf{x}) \} \quad \text{s.t.} \quad f_k(\mathbf{x}) - r_k \leq 0$$

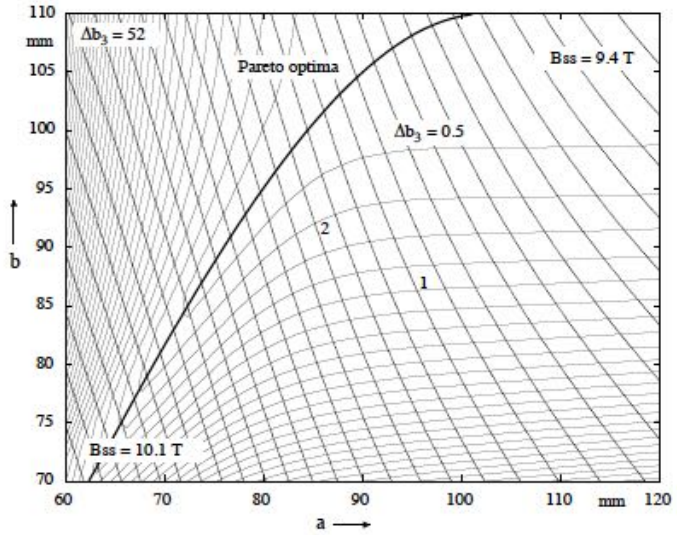


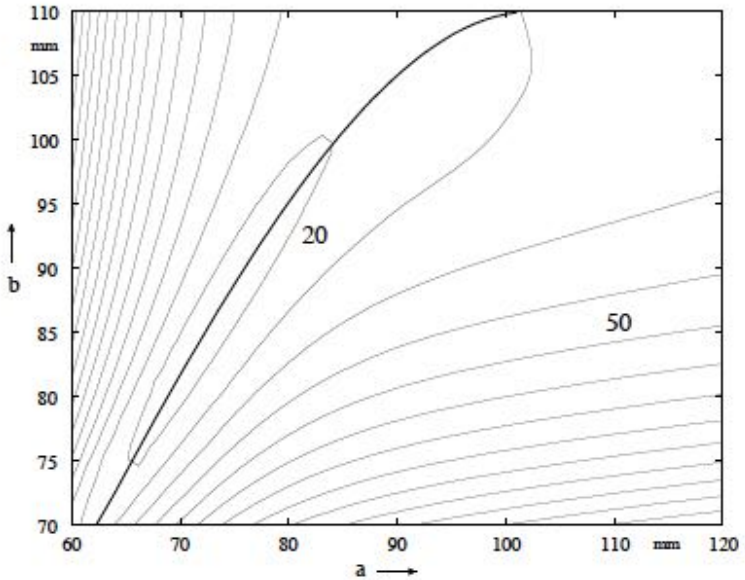


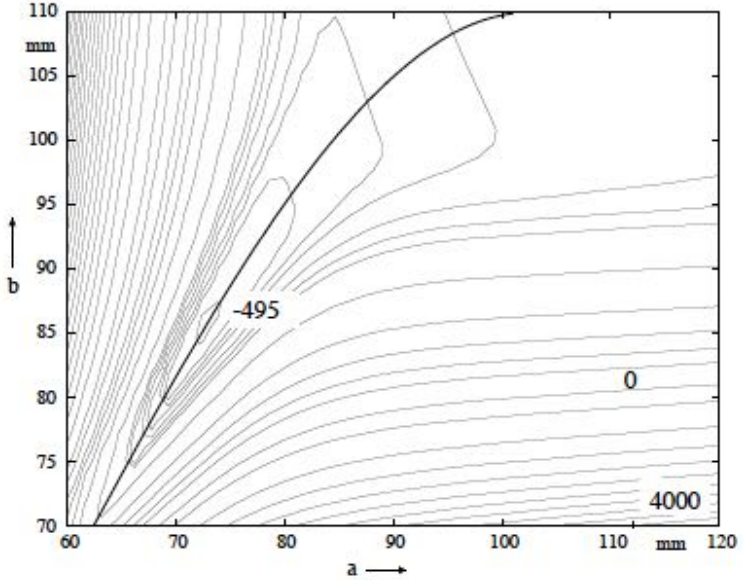


Objective Weighting and L2 Distance Function



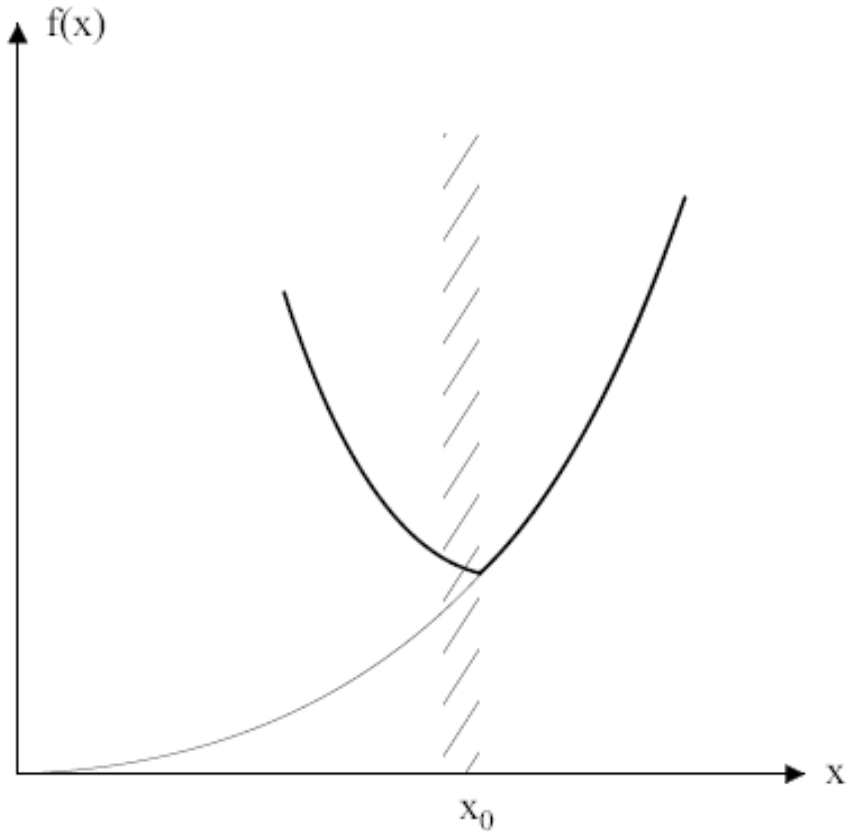






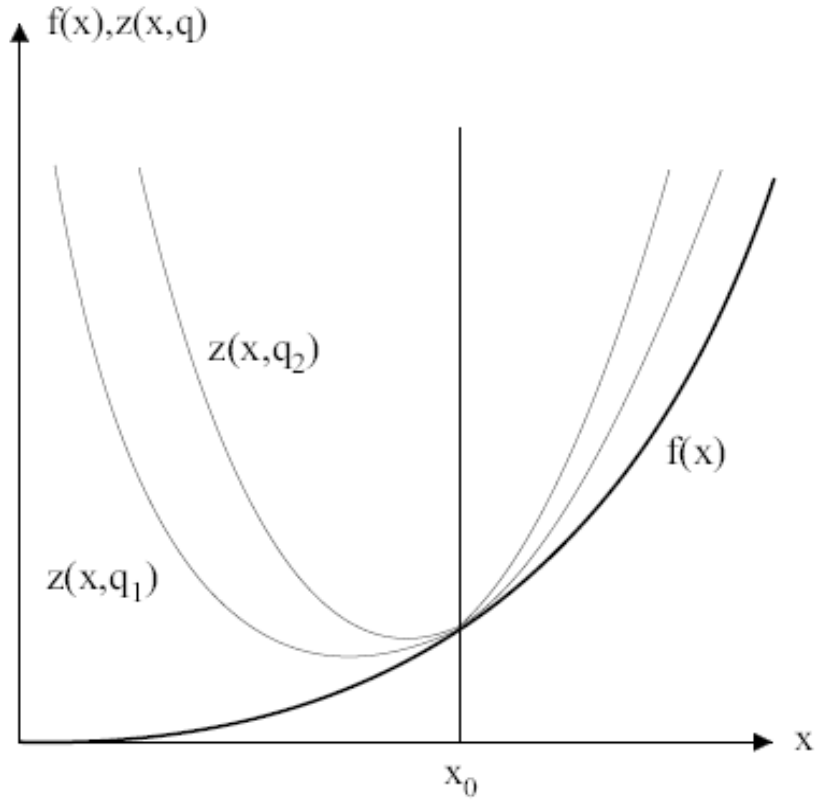
Treatment of Constraints (Box-Constraints)

$$p(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{no bound violated} \\ f(\mathbf{x}^*) + r(\mathbf{x}) & \text{bound violated} \end{cases}$$



$$r(\mathbf{x}) = \sum_l r_l \begin{cases} (x_l - x_{l,\text{upper}})^2 & \text{if } x_l > x_{l,\text{upper}} \\ (x_{l,\text{lower}} - x_l)^2 & \text{if } x_l < x_{l,\text{lower}} \\ 0 & \text{otherwise} \end{cases}$$

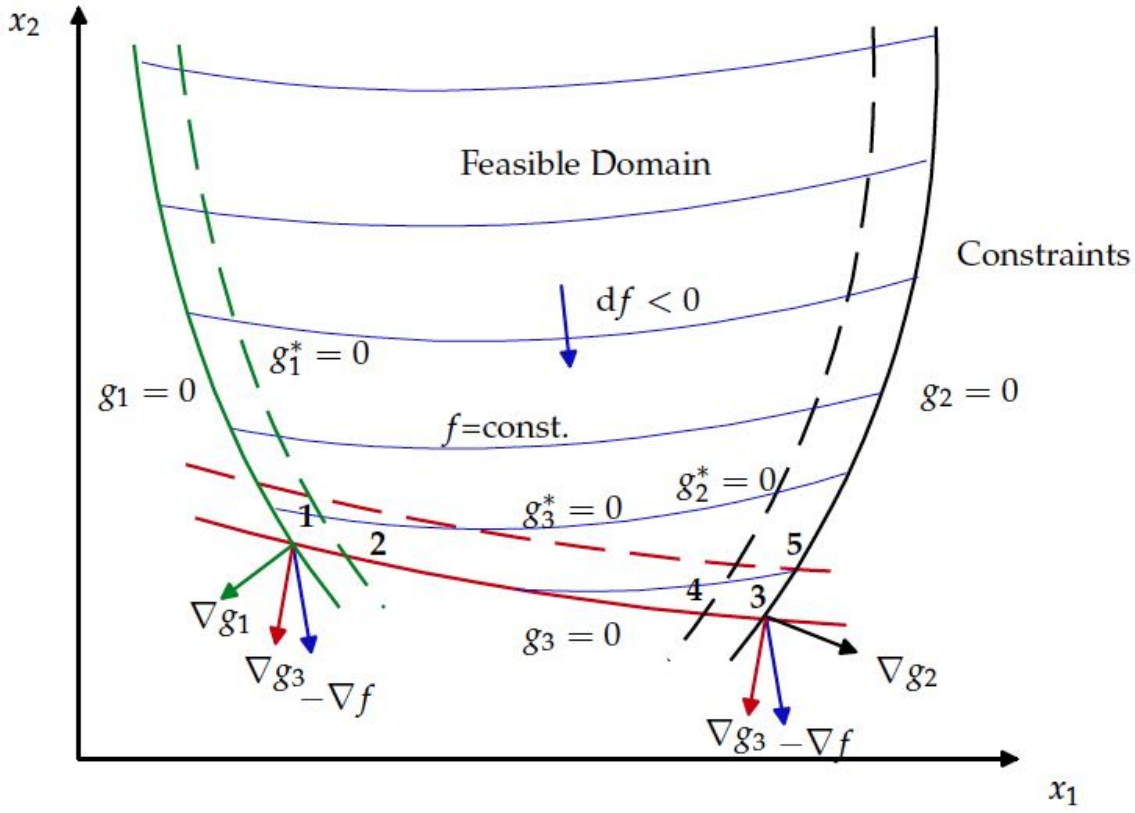
Penalty Transformation



$$z(\mathbf{x}, \mathbf{p}, \mathbf{q}) = f_i(\mathbf{x}) + \sum_{k=1}^{m+K-1} p_k \cdot \max^2\{(0.\,,\, g_k(\mathbf{x}) - d_k)\} + \sum_{j=1}^{p} q_j (h_j(\mathbf{x}) - c_j)^2$$

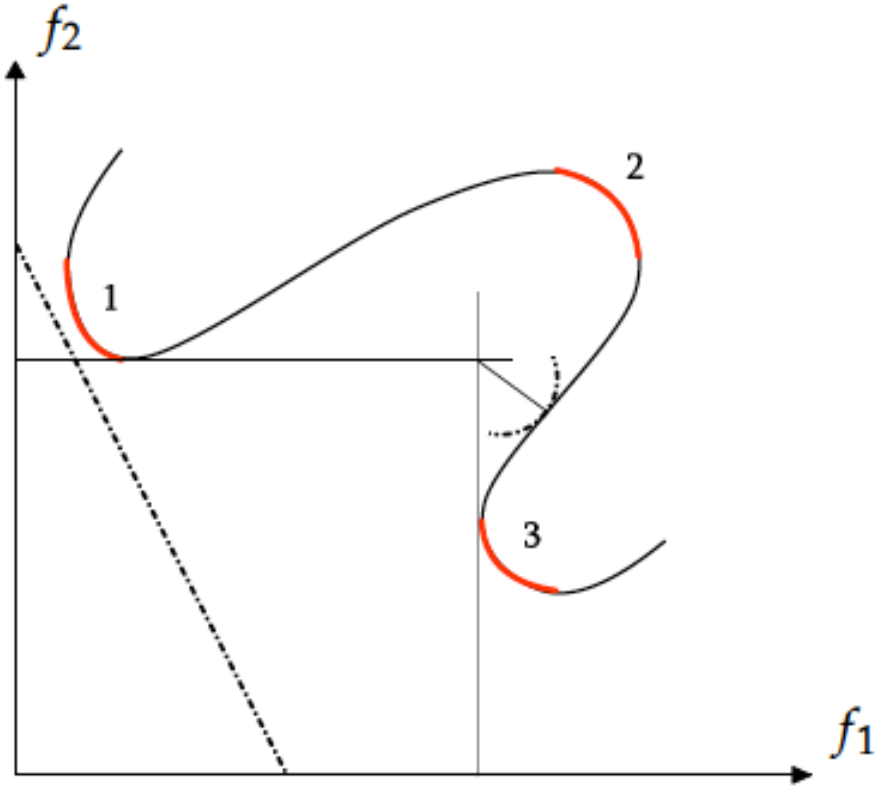
Sensitivity: The Kuhn Tucker Equations

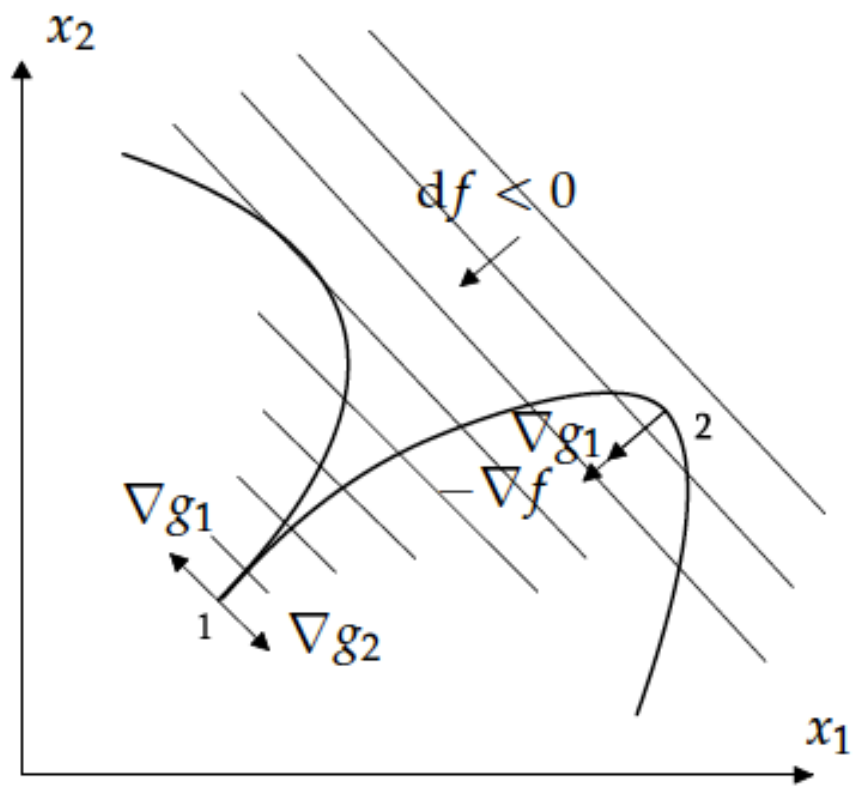
$$\nabla_{\mathbf{x}}\mathcal{L}(\mathbf{x}^*, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \nabla f_i(\mathbf{x}^*) + \boldsymbol{\alpha}^T \nabla \mathbf{g}(\mathbf{x}^*) + \boldsymbol{\beta}^T \nabla \mathbf{h}(\mathbf{x}^*) = \mathbf{0},$$
$$\mathbf{g}(\mathbf{x}^*) - \mathbf{d} = \mathbf{0},$$
$$\mathbf{h}(\mathbf{x}^*) - \mathbf{c} = \mathbf{0},$$
$$\alpha_i > 0.$$



$$\min_{\boldsymbol{\alpha},\boldsymbol{\beta}} \left\{ \left| \nabla f_i(\mathbf{x}^*) + \boldsymbol{\alpha}^T \nabla \mathbf{g}(\mathbf{x}^*) + \boldsymbol{\beta}^T \nabla \mathbf{h}(\mathbf{x}^*) \right| \right\}$$

Pitfalls for Non-Convex Functions

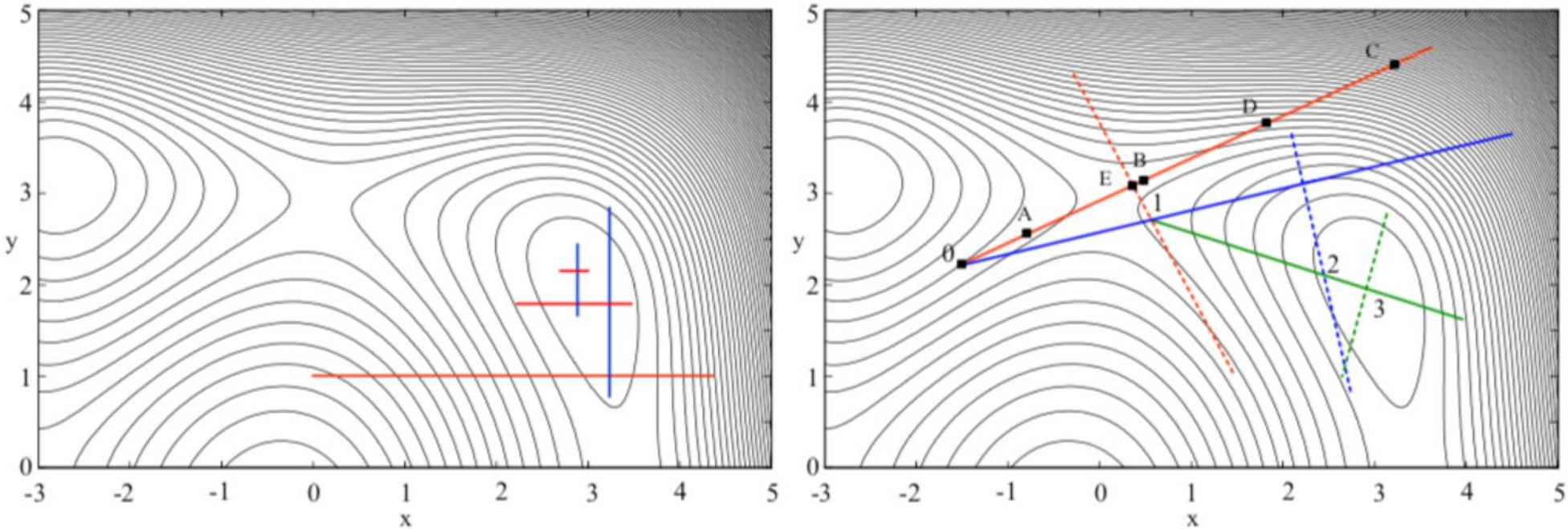




Optimization Algorithms

Search methods		
Direct search	Gauss-Seidel	
EXTREM	Jacob	1982
Rosenbrock	Rosenbrock	1960
Powell	Powell	1965
Flexible Polyhedron search	Nelder-Mead	1964
Hooke-Jeeves	Hooke-Jeeves	1962
Gradient methods		
Steepest descend	Cauchy	1847
Newton's method	Newton	1700
Levenberg-Marquard	Levenberg Marquard	1963
Conjugate gradient (CG)	Fletcher-Reeves	1964
Quasi-Newton	Davidon-Fletcher-Powell	1959
Stochastic and neural computing		
Evolutionary	Rechenberg	1964
Genetic algorithms	Fogel-Holland	1987
Neural computing (ANN)	Aarts-Korst	1989

Direct Search and EXTREM



Gradient Methods

$$\mathbf{g}_k := \nabla f(\mathbf{x}_k)^T$$

Steepest decent

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \lambda_k \mathbf{g}_k$$

Newtons method

Quadratic approximation

$$q(\mathbf{x}) = f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)(\mathbf{x} - \mathbf{x}_k) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_k)^T \mathbf{H}(\mathbf{x}_k)(\mathbf{x} - \mathbf{x}_k)$$

$$\nabla q(\mathbf{x})|_{\mathbf{x}_{k+1}}^T = \mathbf{0} = \mathbf{g}_k + \mathbf{H}(\mathbf{x}_k)(\mathbf{x}_{k+1} - \mathbf{x}_k)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [\mathbf{H}(\mathbf{x}_k)]^{-1} \mathbf{g}_k.$$

Quasi-Newton

SD Newton

$$\mathbf{S} = [\epsilon_k \mathbf{I} + \mathbf{H}(\mathbf{x}_k)]^{-1}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \lambda_k \mathbf{S} \mathbf{g}_k,$$

Levenberg Marquard

$$\min\{z(\mathbf{x})\} = \min\left\{\frac{1}{2}\sum_{i=1}^{m} q_i^2(\mathbf{x})\right\} = \min\left\{\frac{1}{2}\mathbf{f}(\mathbf{x})^T\mathbf{f}(\mathbf{x})\right\}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [\epsilon_k\mathbf{I} + \mathbf{H}(\mathbf{x}_k)]^{-1}\mathbf{g}_k.$$

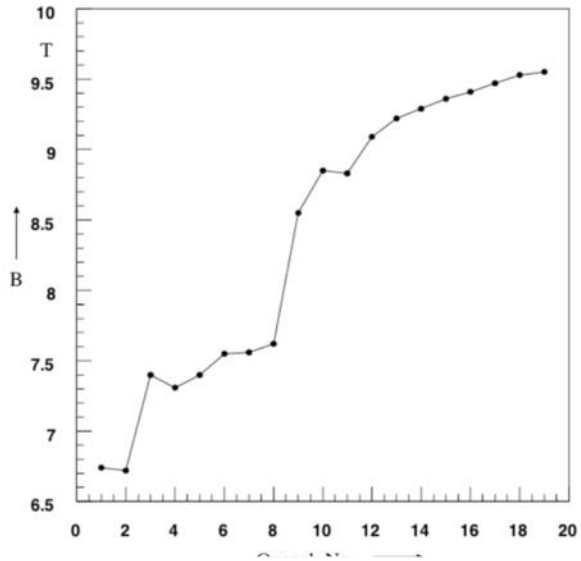
$$\nabla z(\mathbf{x}_k)^T = [\mathbf{J}(\mathbf{x}_k)]^T\mathbf{f}(\mathbf{x}_k),$$

Because of the quadratic nature of the optimization problem

$$\mathbf{H}(\mathbf{x}_k) = [\mathbf{J}(\mathbf{x}_k)]^T\mathbf{J}(\mathbf{x}_k) + \frac{\partial\mathbf{J}(\mathbf{x}_k)}{\partial\mathbf{x}_k}\mathbf{f}(\mathbf{x}_k).$$

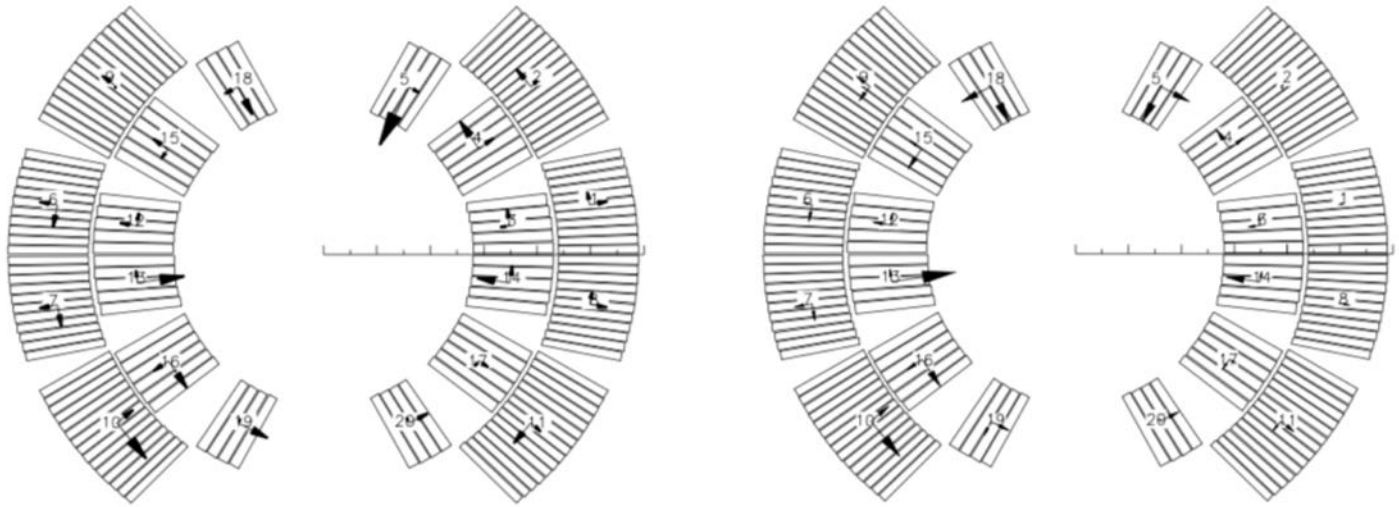
$$\mathbf{x}_{k+1} = \mathbf{x}_k - [\epsilon\mathbf{I} + (\mathbf{J}(\mathbf{x}_k))^T\mathbf{J}(\mathbf{x}_k)]^{-1}\ [\mathbf{J}(\mathbf{x}_k)]^T\mathbf{f}(\mathbf{x}_k)$$

Tracing of Manufacturing Errors



$$\min\left\{\sum_{i=1}^{9} p_i \cdot (b_i^*(\mathbf{x}) - b_i)^2 + q_i \cdot (a_i^*(\mathbf{x}) - a_i)^2\right\}$$

	Before		After		Intrinsic	
n	b	a	b	a	b	a
2	0.378	0.634	0.463	−0.229	0.248	0.000
3	−2.072	0.094	−1.246	0.117	−0.901	0.000
4	−0.055	0.151	−0.028	0.118	0.112	0.000
5	0.247	0.035	0.170	0.013	0.018	0.000
6	0.018	0.006	0.010	0.011	−0.002	0.000
7	0.032	−0.006	0.034	−0.003	0.005	0.000
8	−0.001	0.000	0.000	0.000	0.000	0.000
9	−0.001	0.000	−0.001	−0.001	0.000	0.000



Genetics and Evolution Theory

➔ Darwin (1860)
- Survival of the fittest
- Variations between individuals of species
- Reproductive populations
- Evolutionary computation (Rechenberg, Schwefel 1964)

➔ Mendel (1850)
- Genetic basis of variation
- Coding (Discrete units)
- Genetic algorithms (Holland 1970)
- Niching decreases selective pressure
- Niching genetic algorithms (Mahfoud 1995)

The Bee Orchid

Genetic Optimization

- Genetic operators
  - Crossover
  - Mutation
  - Selection
- Fairy wheel selection
- Niching genetic algorithms
- Results of coil block optimizations

Genetic Algorithms

Phenotype	$x_1^{(1)}$	$x_2^{(1)}$	$x_3^{(1)}$	$\ldots x_n^{(1)}$	Population
	$x_1^{(j)}$	$x_2^{(j)}$	$x_3^{(j)}$	$\ldots x_n^{(j)}$	(Elder)
	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	
Gene pool	(0111)	(1001)	(1101)	(1011)	
(Genotype)	(0011)	(0001)	(1110)	(1101)	
	$\downarrow$	Gray coding		$\downarrow$	
Chromosomes	(0100110110111110)				
	(0010000110011011)				
	$\downarrow$				Recombination,
Crossover, Mutation,	Genetic Operators				Selection, Niching,
Genetic drift	$\downarrow$				Migration
Chromosomes	(0100110110111110)				
	(0110110010110010)				
	$\downarrow$	Gray decoding		$\downarrow$	
Genotype	(0110)	(1011)	(0101)	(0110)	
	(0100)	(1000)	(1101)	(0011)	
	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	
Phenotype	$x_1^{(1)*}$	$x_2^{(1)*}$	$x_3^{(1)*}$	$\ldots x_n^{(1)*}$	Offspring
	$x_1^{(j)*}$	$x_2^{(j)*}$	$x_3^{(j)*}$	$\ldots x_n^{(j)*}$	

Encoding

$$g_i = \begin{cases} b_i & \text{if } i = 1 \\ b_{i-1} \oplus b_i & \text{if } i \geq 2 \end{cases}$$

$$\mathbf{b}(13) = (1101)$$

$$(1101) \oplus (0110) = (1011)$$

Decimal	Binary		Gray
0	(0000)	$\leftrightarrow$	(0000)
1	(0001)		(0001)
2	(0010)		(0011)
3	(0011)		(0010)
4	(0100)		(0110)
5	(0101)		(0111)
6	(0110)	$\leftrightarrow$	(0101)

Decimal	Binary		Gray
7	(0111)	$\leftrightarrow$	(0100)
8	(1000)		(1100)
9	(1001)		(1101)
10	(1010)		(1111)
11	(1011)		(1110)
12	(1100)		(1010)
13	(1101)	$\leftrightarrow$	(1011)

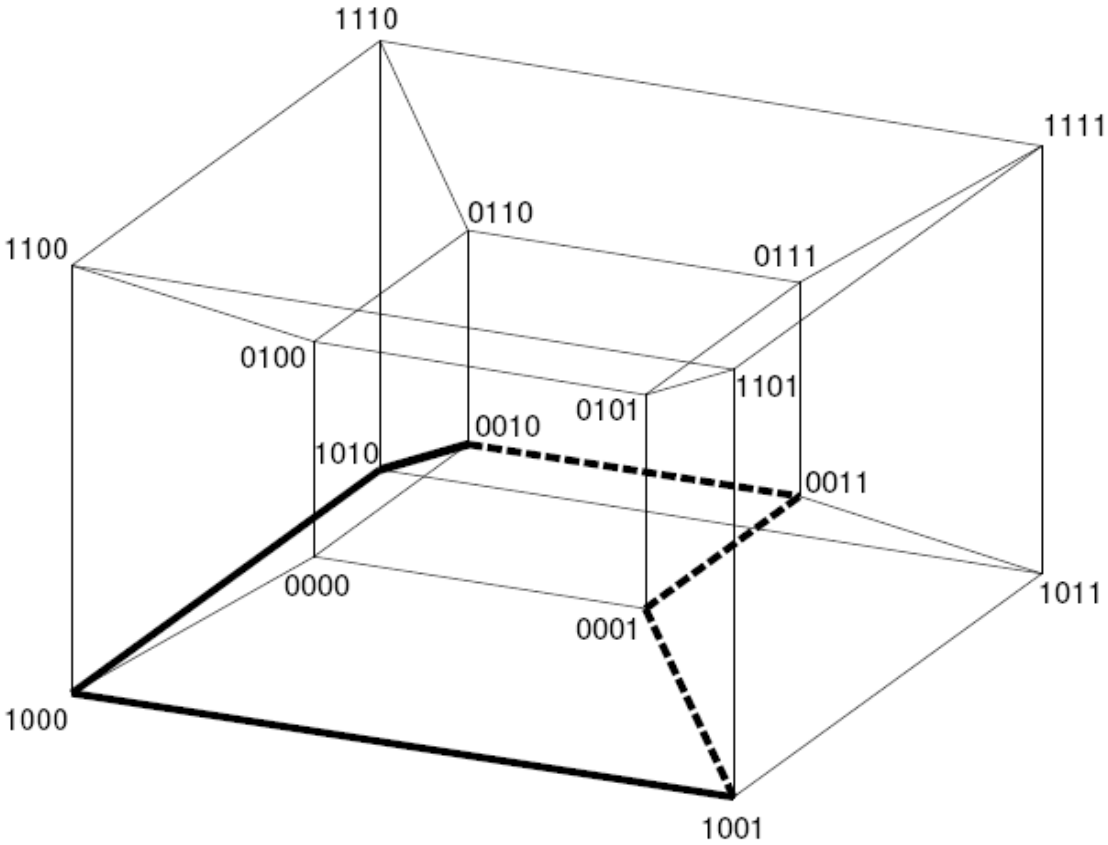
$$b_i = \bigoplus_{j=1}^{i} g_j$$

$$\mathbf{g} = (1011)$$

$$(1011) \oplus (0101) \oplus (0010) \oplus (0001) = (1101)$$

Genetic Operators: Crossover

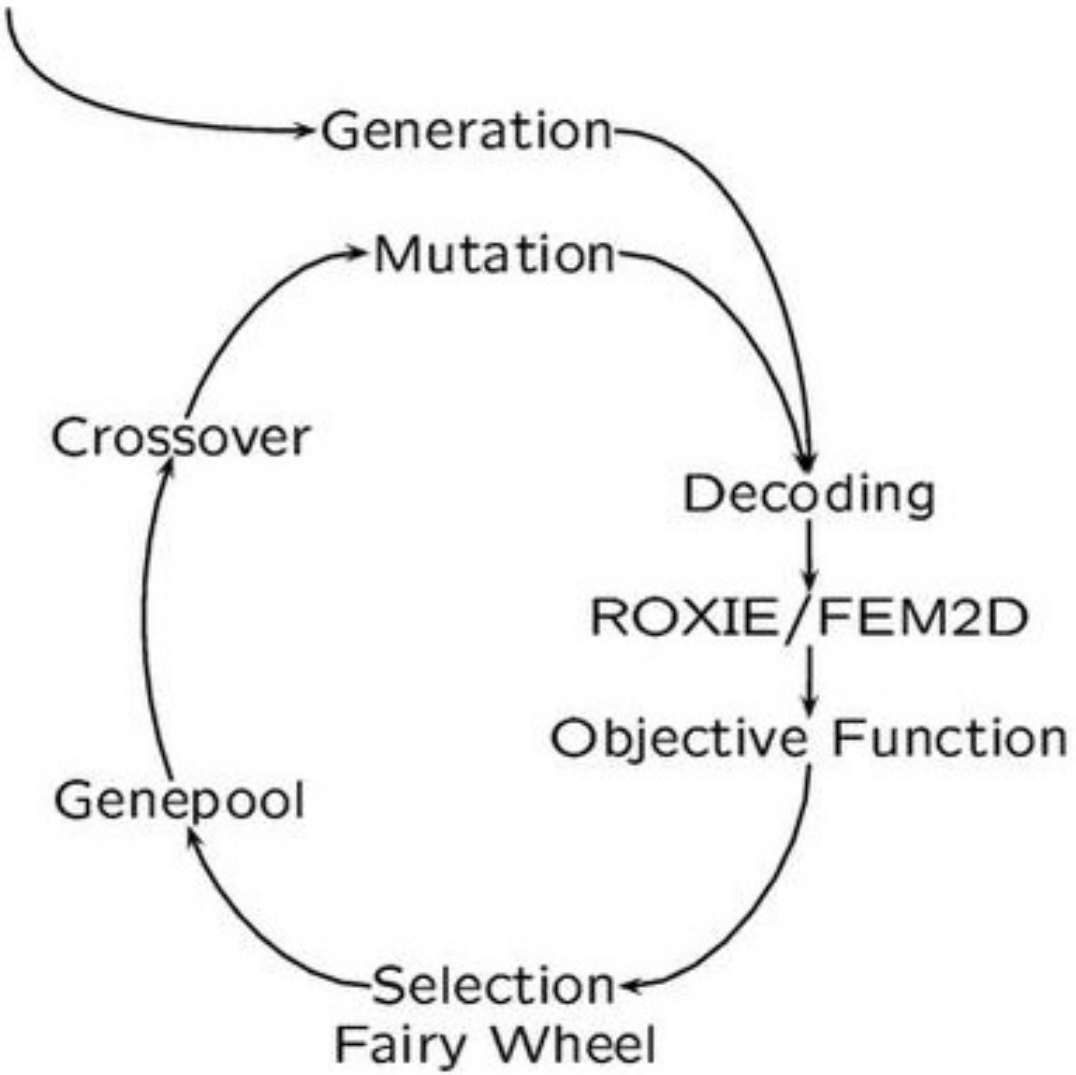
Chromosome A:	(0101001|101)	(0101001)	(011)	(0101001|011)
	:	×	:	
Chromosome B:	(1011010|011)	(1011010)	(101)	(1011010|101)



Genetic Operators: Mutation

	10	7	2	1	1	0	Phenotype, k
	1111	0100	0011	0001	0001	0000	Genotype, $\mathbf{g}(k)$
a)	10	8	2	1	1	0	Phenotype, k
	1111	**1**100	0011	0001	0001	0000	Genotype, $\mathbf{g}(k)$
b)	13	7	2	1	1	0	Phenotype, k
	1**0**11	0100	0011	0001	0001	0000	Genotype, $\mathbf{g}(k)$

Royal Road Algorithm



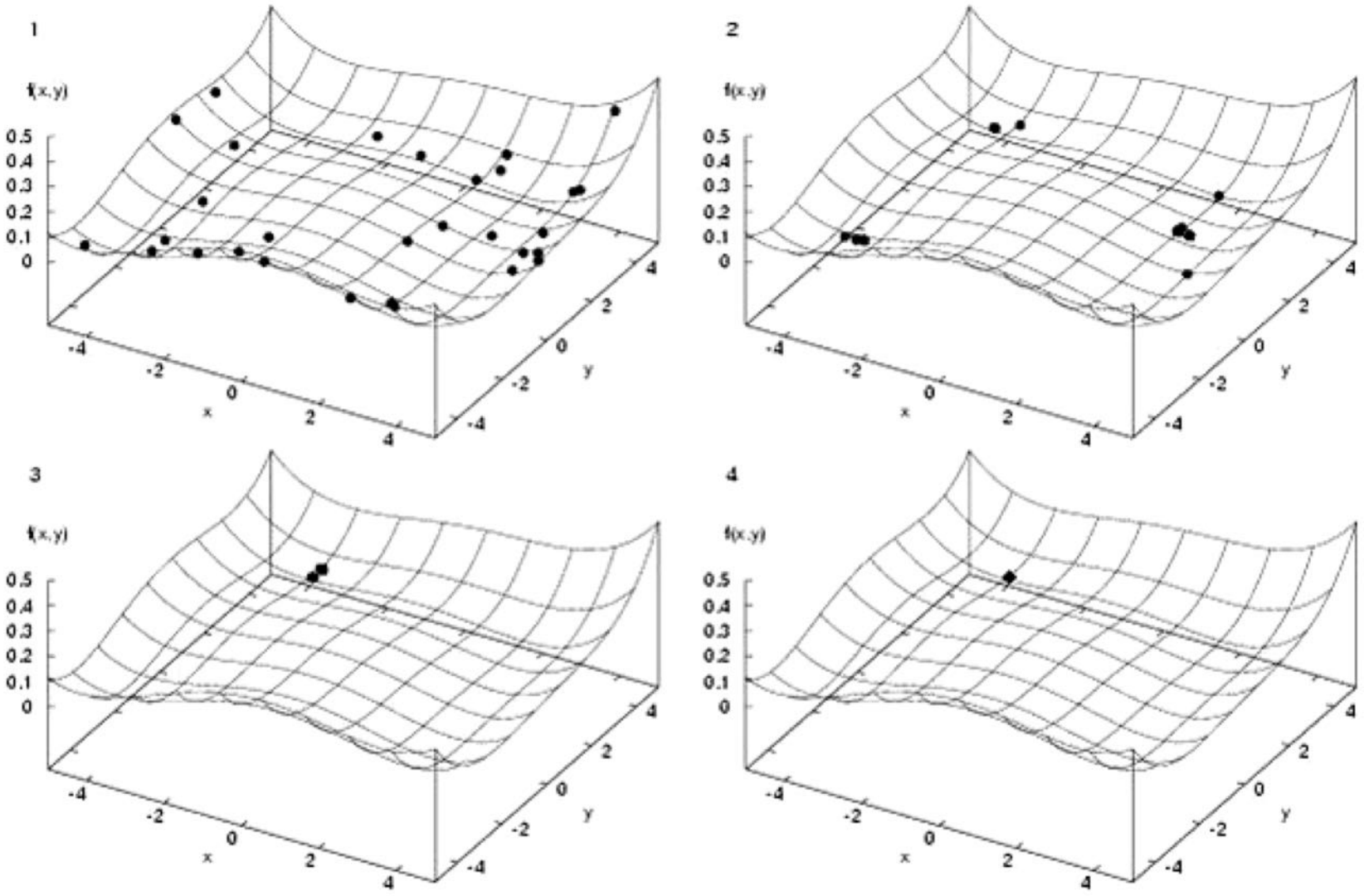
Fairy Wheel Selection

Index	Parent Population	Objective func. val.	Fitness value		Index of selected Chromosome	Child Population
1	(1000111010)	0.3	0.30		3	(1001101101)
2	(1110101101)	0.4	0.22		1	(1000111010)
3	(1001101101)	0.5	0.18		4	(1011010010)
4	(1011010010)	0.8	0.11	:	1	(1000111010)
5	(0111001100)	0.9	0.10		2	(1110101101)
6	(0111011010)	1.7	0.05		3	(1001101101)
7	(0011000101)	2.6	0.04		1	(1000111010)

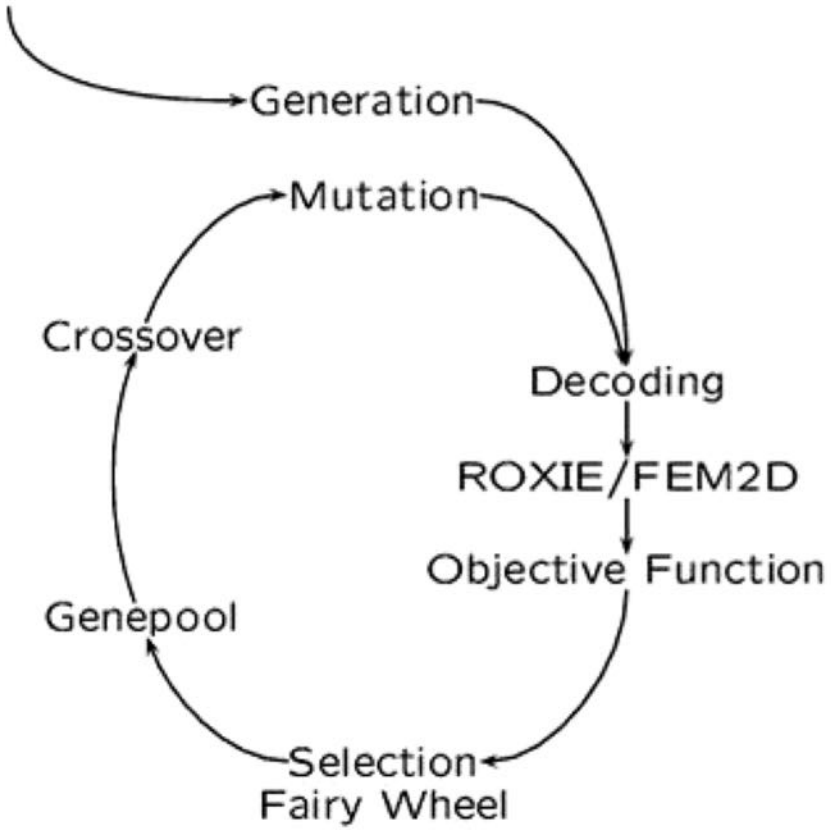
Fitness $e(\mathbf{b}) = f(\mathbf{x}) + c$

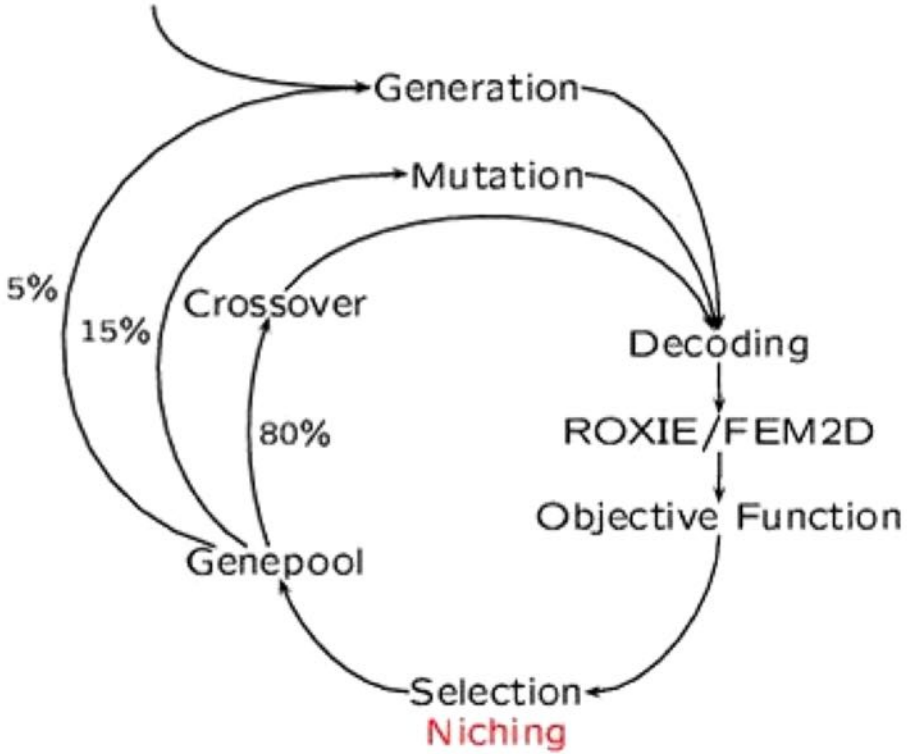
Likelihood of reproduction $p_i := \dfrac{e(\mathbf{b}_i)}{\sum_{j=1}^{n} e(\mathbf{b}_j)}$

Local Properties



Niching Genetic Algorithm





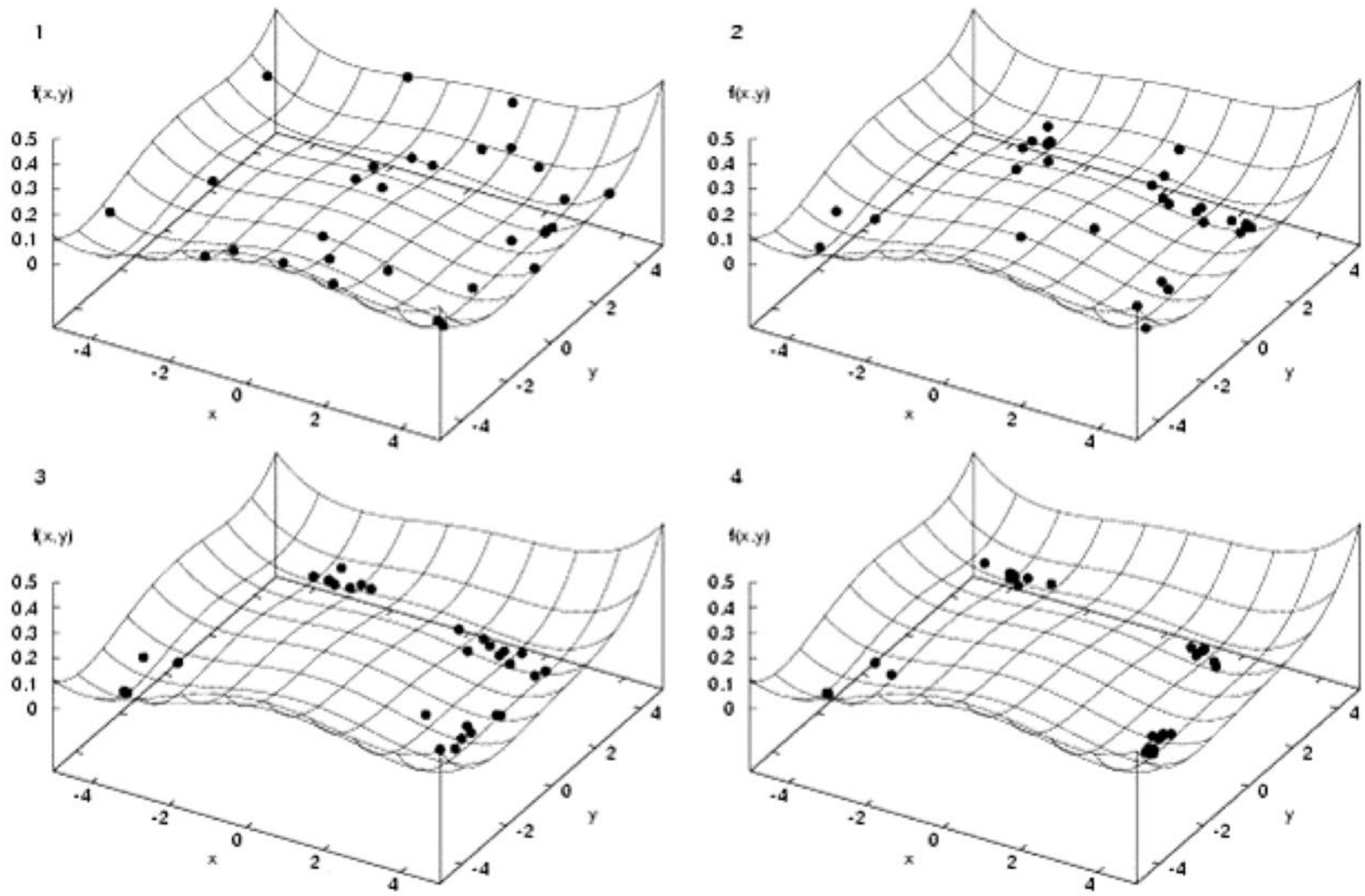
Niching Selection

Index	Parent Population	Objective func. val.	Hamming distance		Index of selected Chromosome	Child Population
1	(1000111010)	0.3	5		1	(1001101101)
2	(1110101101)	0.4	7		2	(1000111010)
3	(1001101101)	0.5	6		3	(1011010010)
4	(1011010010)	0.8	3	:	4	(1000111010)
5	(0111001100)	0.9	3		5	(0111011010)
6	(0111011010)	1.7	2		New	(0011001010)
7	(0011000101)	2.6	4		7	(1000111010)
New	(0011001010)	1.1	0			

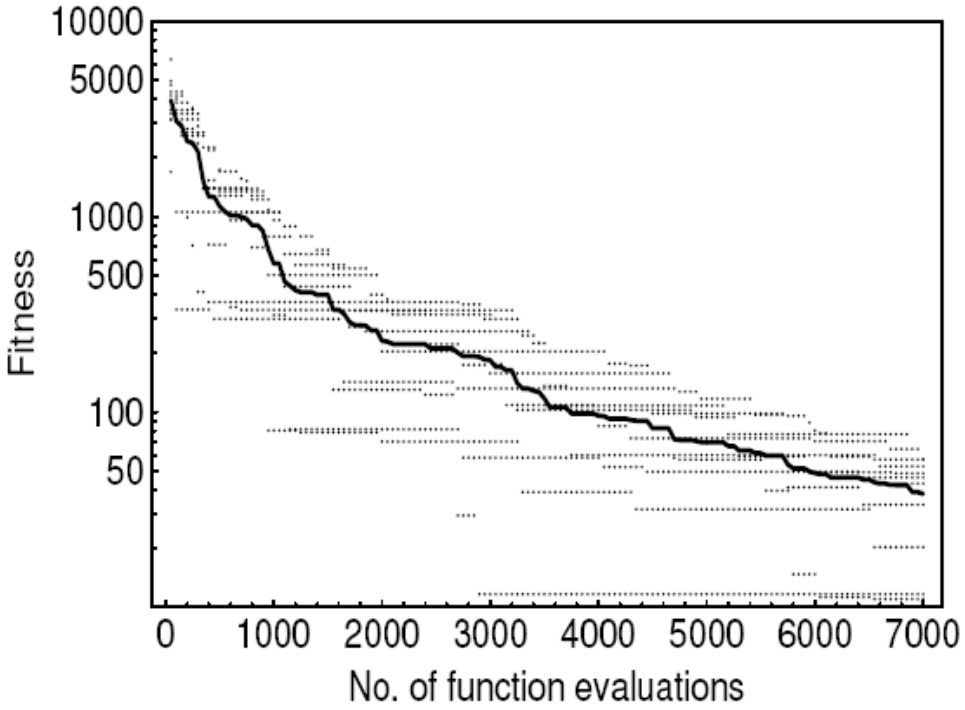
Fitness value

$$F_s(\mathbf{x}_i) := \frac{f(\mathbf{x}_i)}{\sum_{j=1}^{n} s(d(\mathbf{x}_i, \mathbf{x}_j))}$$

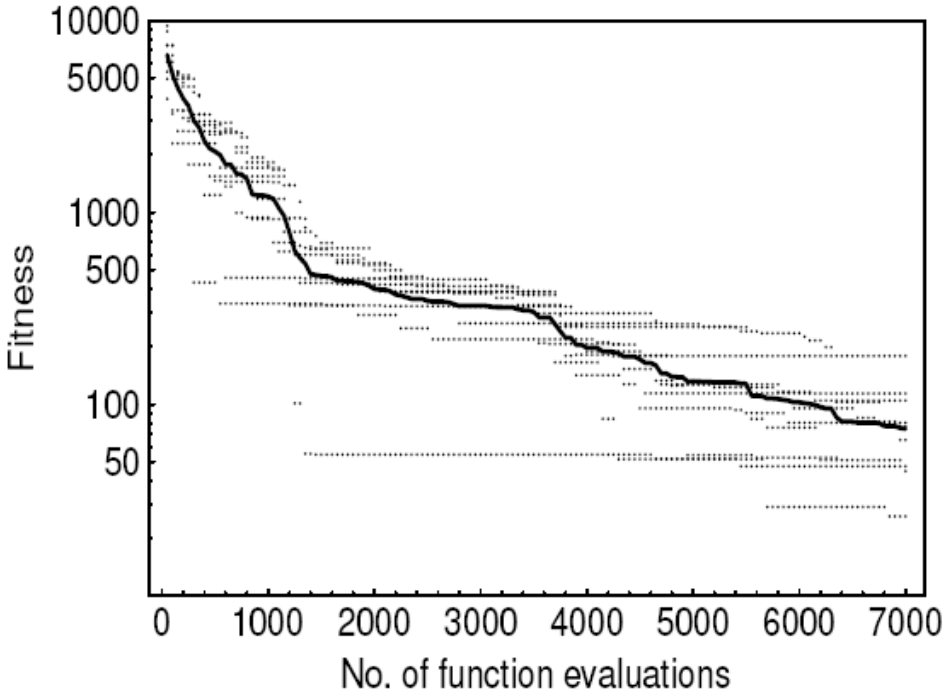
Global Properties



Convergence



Crossover rate 0.8
Mutation rate 0.15
Generation rate 0.05



Crossover rate 0.6
Mutation rate 0.35
Generation rate 0.05

Different Solutions found by Niching Genetic Algorithms