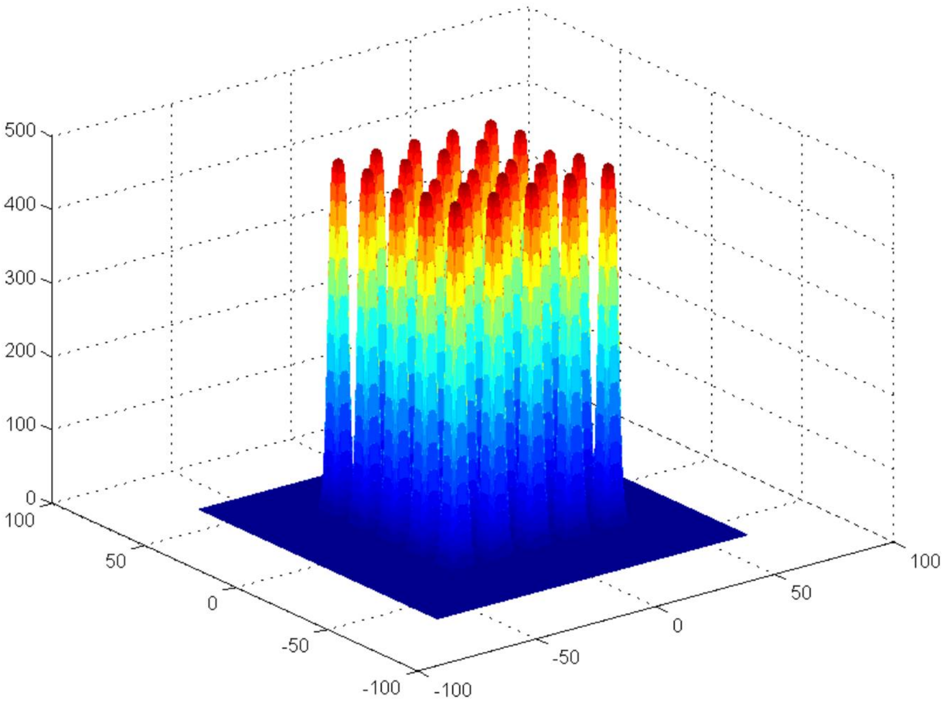
TMKT48 - Summary

LINKÖPING UNIVERSITY

Contents

- Elements of Optimization
- Optimization Algorithms
- Multi-Objective Optimization
- Sensitivity Analysis
- Surrogate Models



Elements of Optimization

Elements of Optimization: Design Variables

- Entities that the designer can change

- They could be:
  - Continuous: Free to assume any value
  - Discrete: Can assume only fixed values
  - Integer: Can only be integer values

LINKÖPING UNIVERSITY

Elements of Optimization: Objective Function

- The objective function prescribes the criterion that guides the search for the ”best solution”

ObjValue=f(x)

ObjValue is the quantity that should be maximized or minimized

$x=(x_1,x_2,..,x_n)$ represents the design variables

- Examples:
  – Minimize cost
  – Maximize efficiency
  – Minimize Weight

LINKÖPING UNIVERSITY

Elements of Optimization: Objective Function

- The choice of the objective function is crucial
  - Different objective functions produces different optima

- Problems can be single objective or multi objective
  - Multiple objectives are often conflicting

Elements of Optimization: Constraints

- Optimization constraints are numerical values of identified conditions that must be satisfied in order to achieve a feasible solution to a given problem.

- $h(x)=0$ Equality constraint
- $g(x)\leq 0$ Inequality constraint

- Constraints could be inactive or active at the optimum
  - Active => $g(x)=0$. Equality constraints are always active

Elements of Optimization: Design Space

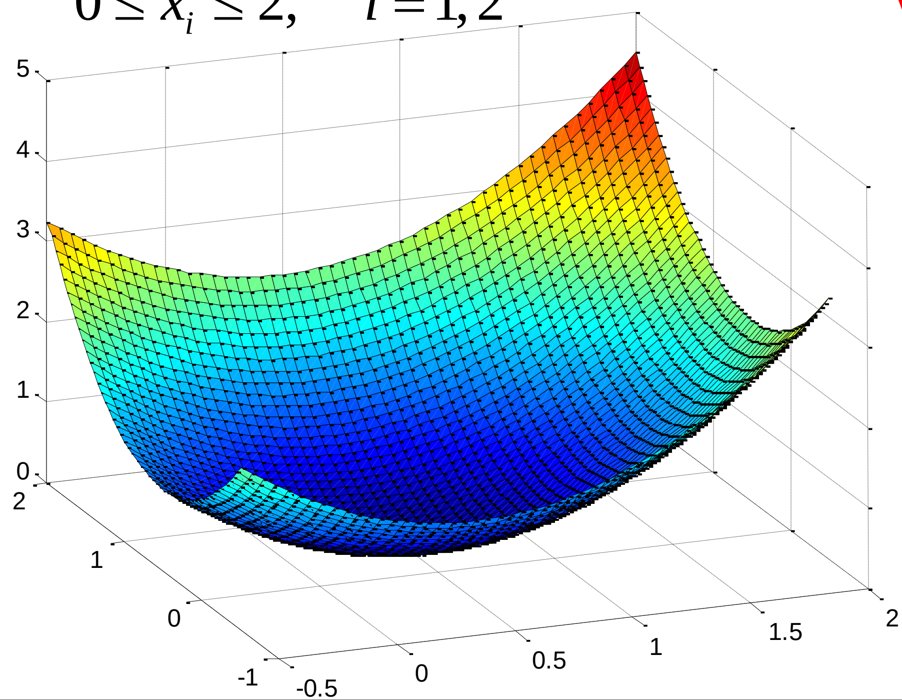
- The imaginary space where possible solutions can be found
- Usually created by assigning minimum and maximum values for each design variable.

Elements of Optimization: Example

$$\min_{\mathbf{x}} f(\mathbf{x}) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2$$ Objective Function

$$x_1 + x_2 - 2 \le 0$$ Constraint

$$0 \le x_i \le 2, \quad i = 1, 2$$ Variable Limits

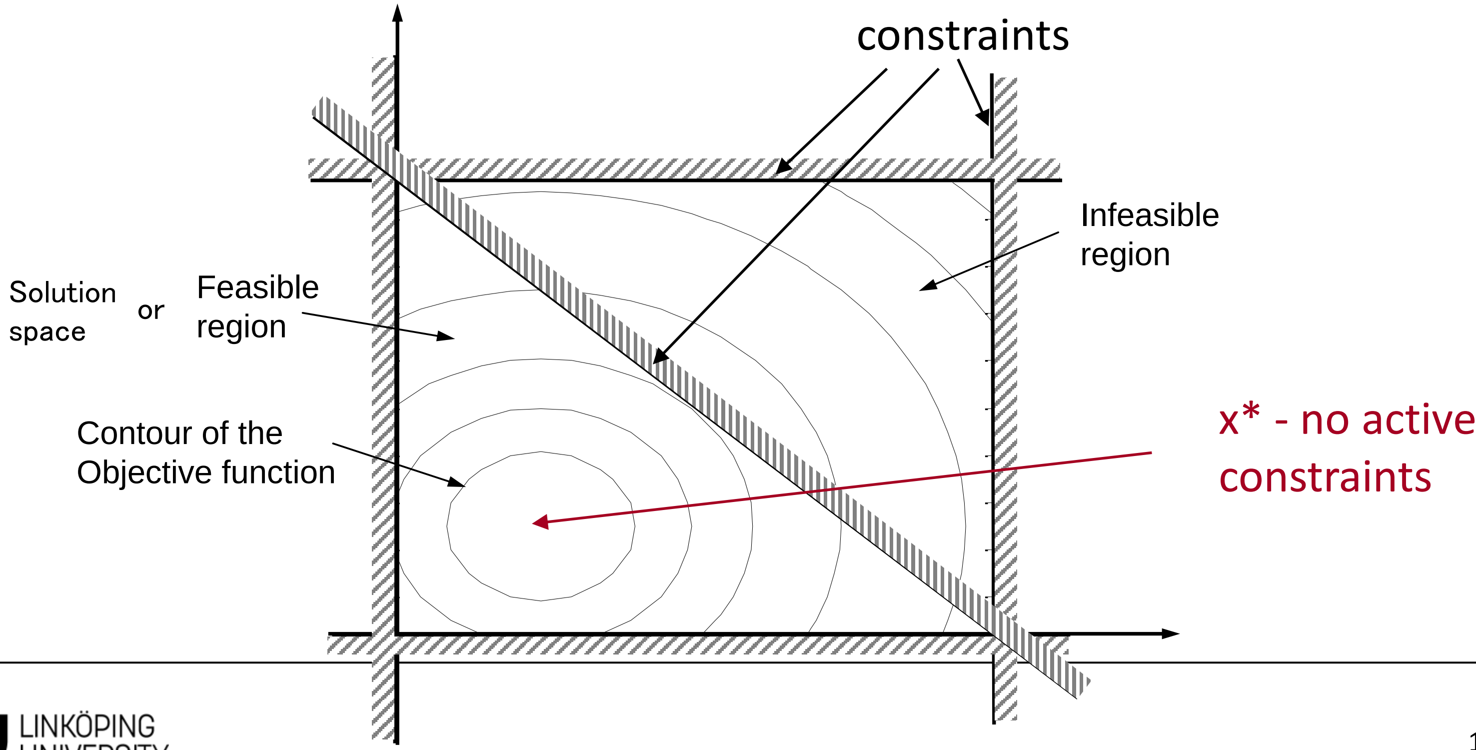


Elements of Optimization: Example

$$\min_{\mathbf{x}} f(\mathbf{x}) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2$$ Objective Function

$$x_1 + x_2 - 2 \le 0$$ Constraint

$$0 \le x_i \le 2, \quad i = 1, 2$$ Variable Limits

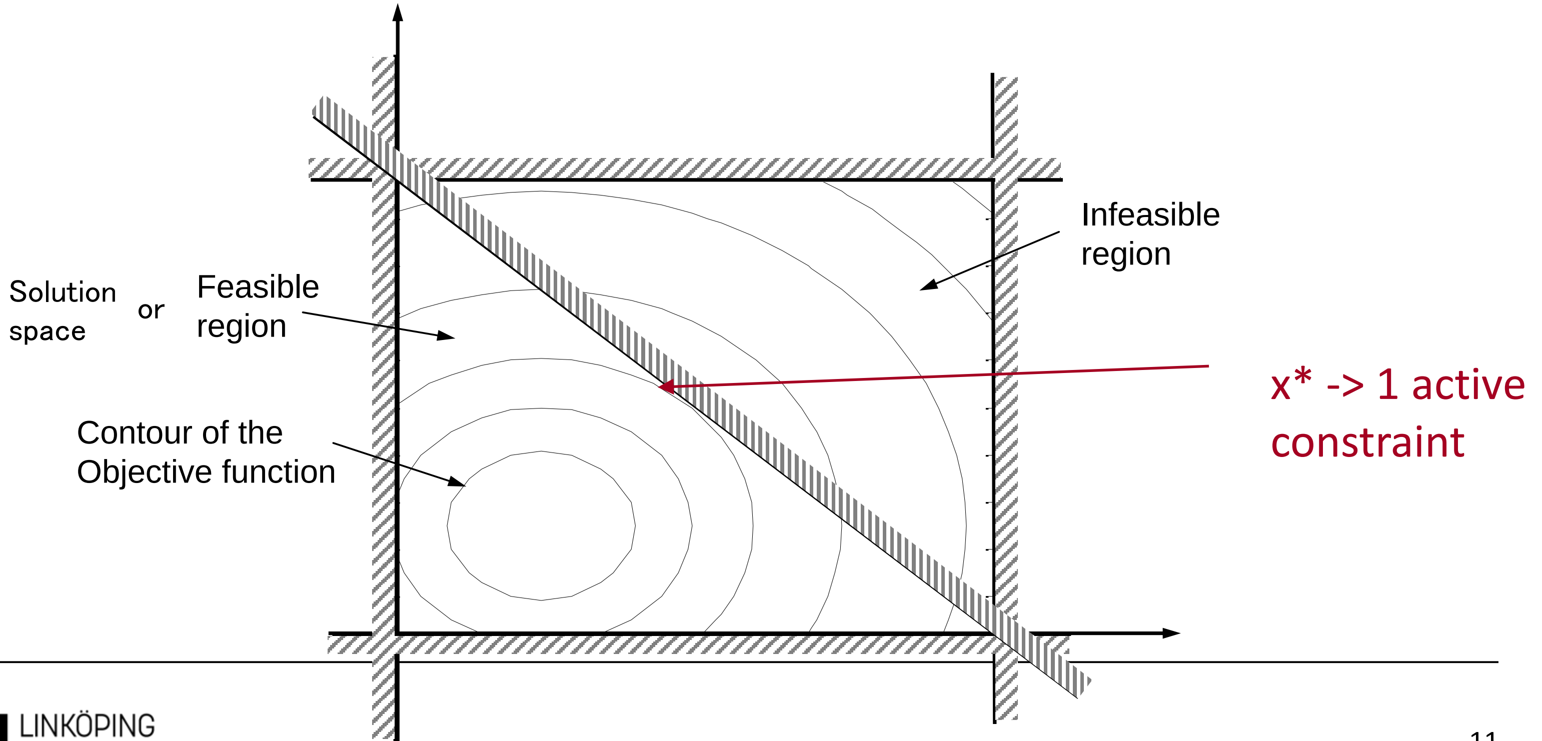


Elements of Optimization: Example

$$\min_{\mathbf{x}} f(\mathbf{x}) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2$$ Objective Function

$$x_1 + x_2 - 2 \leq 0$$ Constraint

$$0 \leq x_i \leq 2, \quad i = 1, 2$$ Variable Limits



The Optimization Problem

Design characteristics

$$\min \mathbf{F}(\mathbf{x}) = f\left(DC_1(\mathbf{x}), DC_2(\mathbf{x}), \ldots, DC_m(\mathbf{x})\right)$$ Objective function

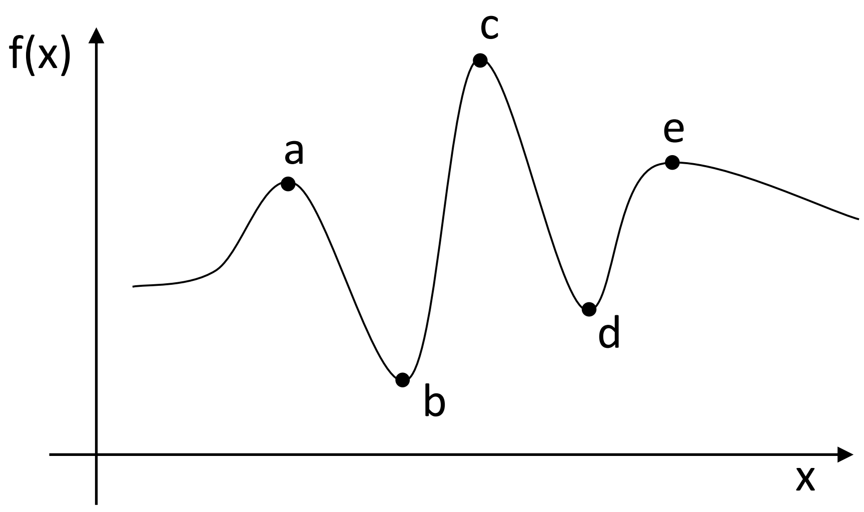
$$\mathrm{x}_\mathrm{i}^\mathrm{l} \le x_i \le \mathrm{x}_\mathrm{i}^\mathrm{u} \qquad i = 1, 2, \ldots, n$$ Variable limits

$$\mathrm{g}_\mathrm{j}(\mathbf{x}) \le 0 \qquad j = 1, 2, \ldots, r$$ Constraints

$$\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$$ Design variables

Global and Local Optima

- A function is unimodal if it has just one hump or depression within a defined interval, otherwise it is multi-modal (as below).



A point that is the best in its immediate vicinity is a local optima

The “best” point of all local optima is the global optima

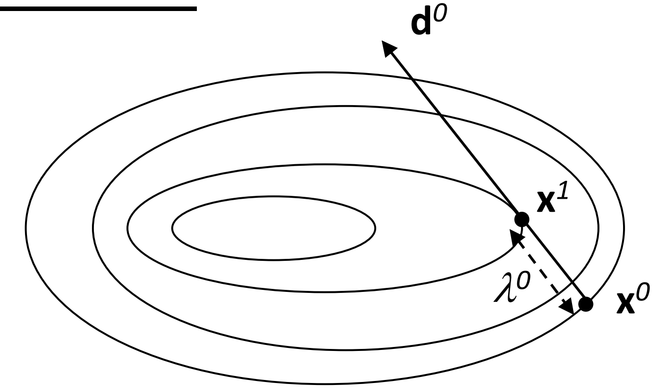
LINKÖPING UNIVERSITY

Optimization Algorithms

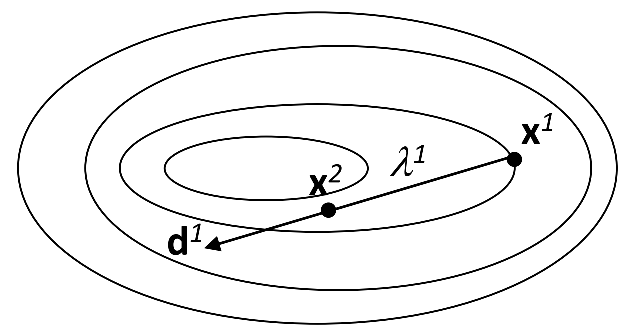
Gradient-Based Optimization Algorithms

- Requires starting point(s)
- Calculates the gradients at the current point
  - Sometimes Hessians as well
- Moves in the direction of the steepest descent

<u>First iteration</u>



<u>Second iteration</u>



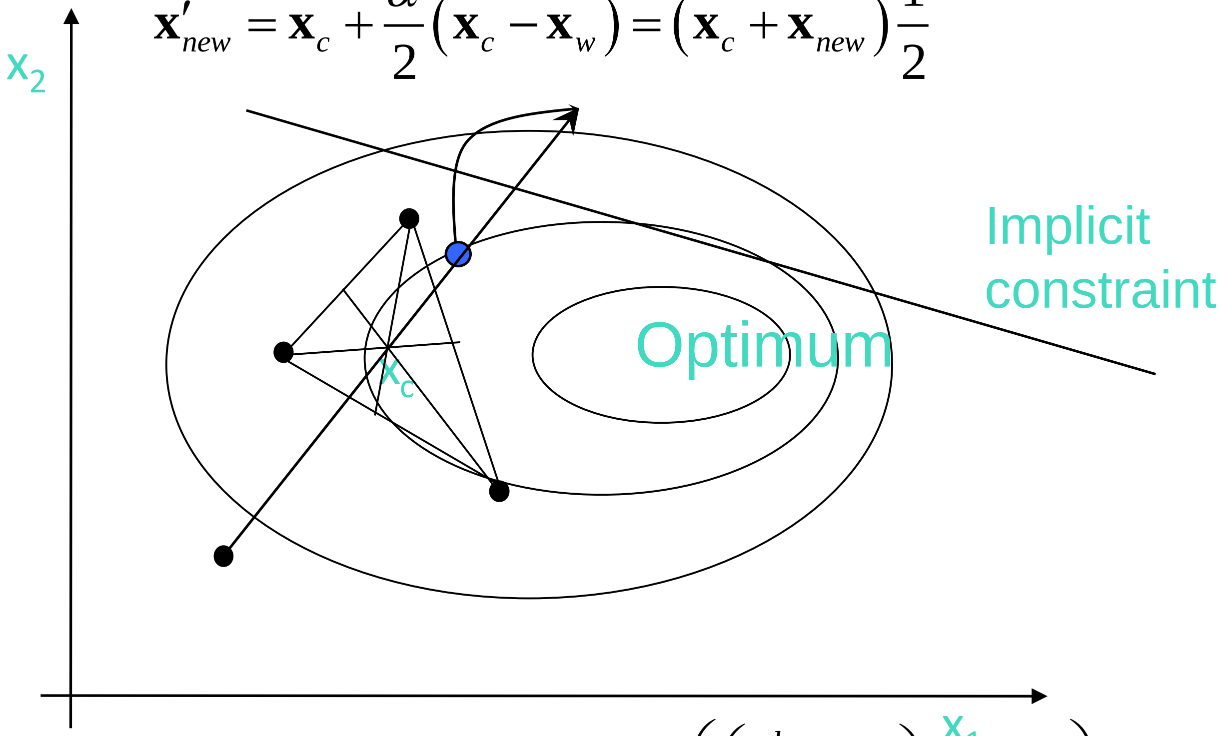
Gradient-Based Optimization Algorithms

- The gradients can be received by
  - Analytical derivatives
  - Finite differences
  - Complex step

The Complex Method

$$\mathbf{x}_{new} = \mathbf{x}_c + \alpha(\mathbf{x}_c - \mathbf{x}_w)$$

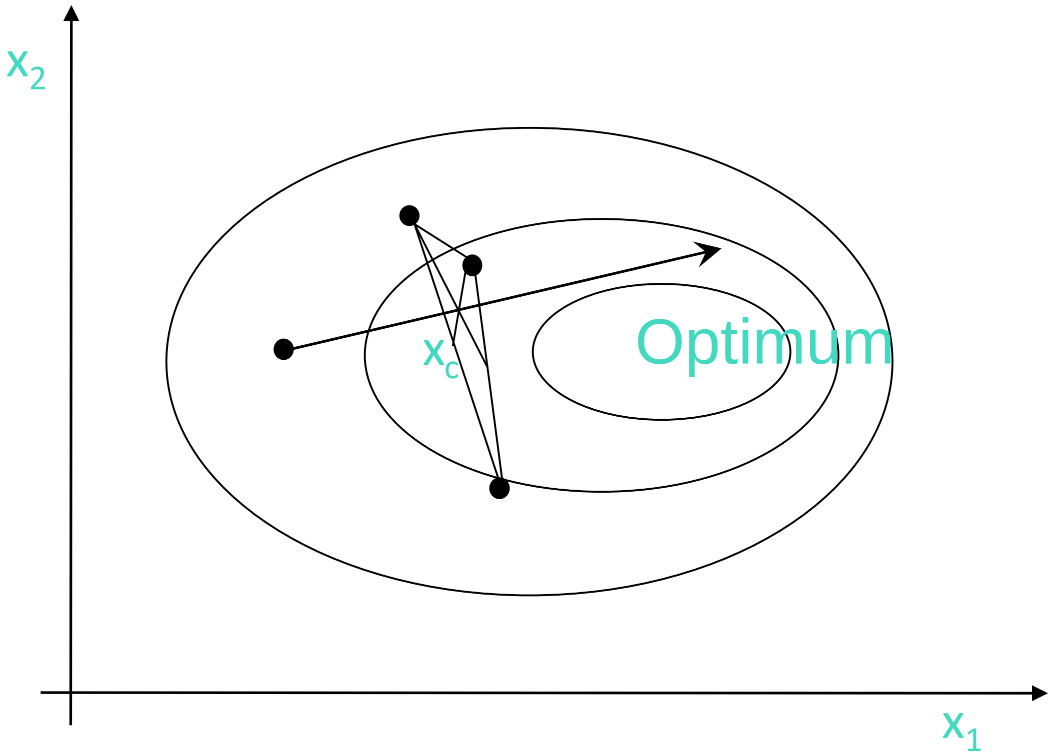
$$\mathbf{x}'_{new} = \mathbf{x}_c + \frac{\alpha}{2}(\mathbf{x}_c - \mathbf{x}_w) = (\mathbf{x}_c + \mathbf{x}_{new})\frac{1}{2}$$



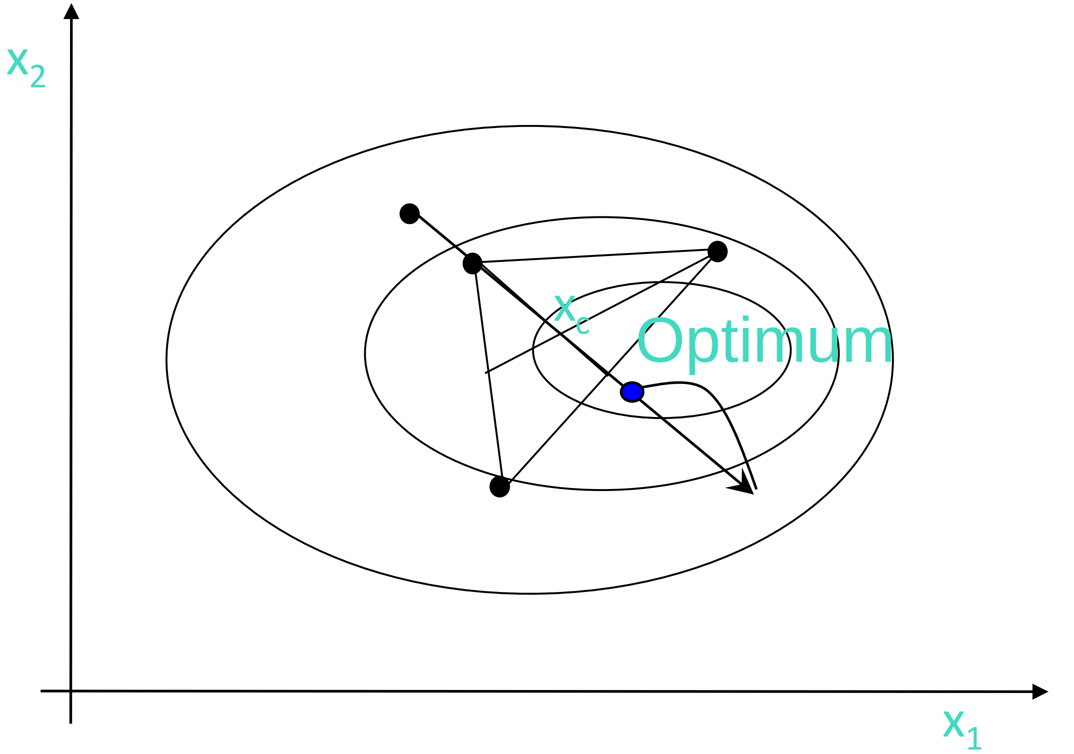
$$x_{c,j} = \frac{1}{k-1}\left(\left(\sum_{i=1}^{k} x_{i,j}\right) - x_{w,j}\right), j = 1 \ldots n$$

…

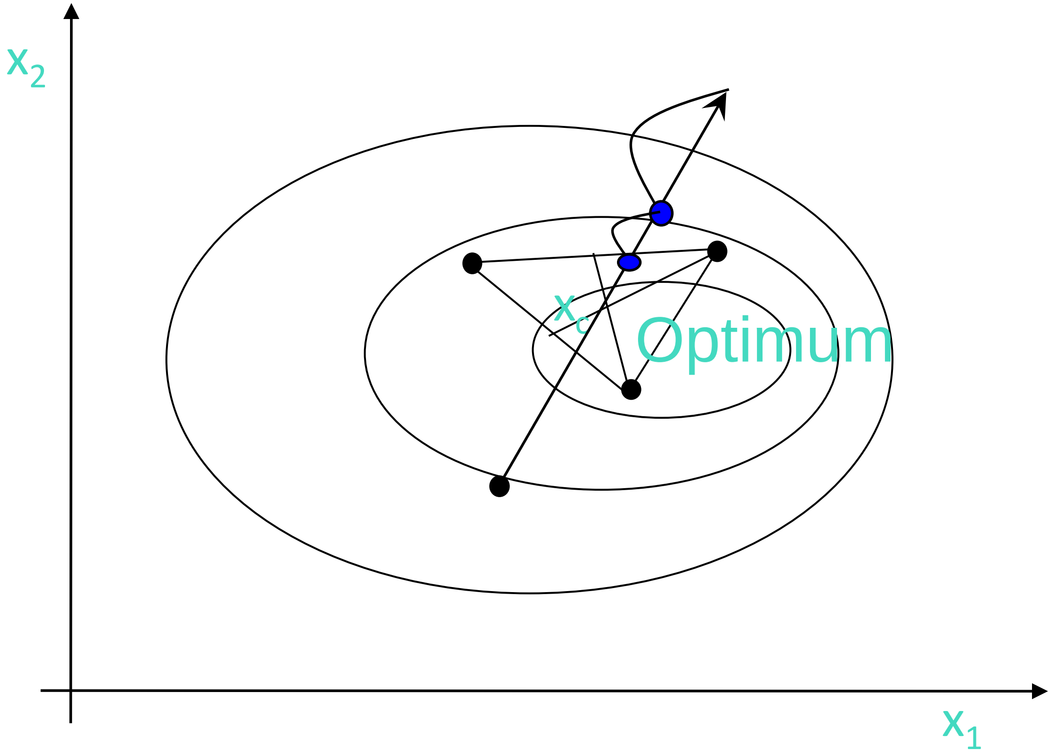
The Complex Method



The Complex method



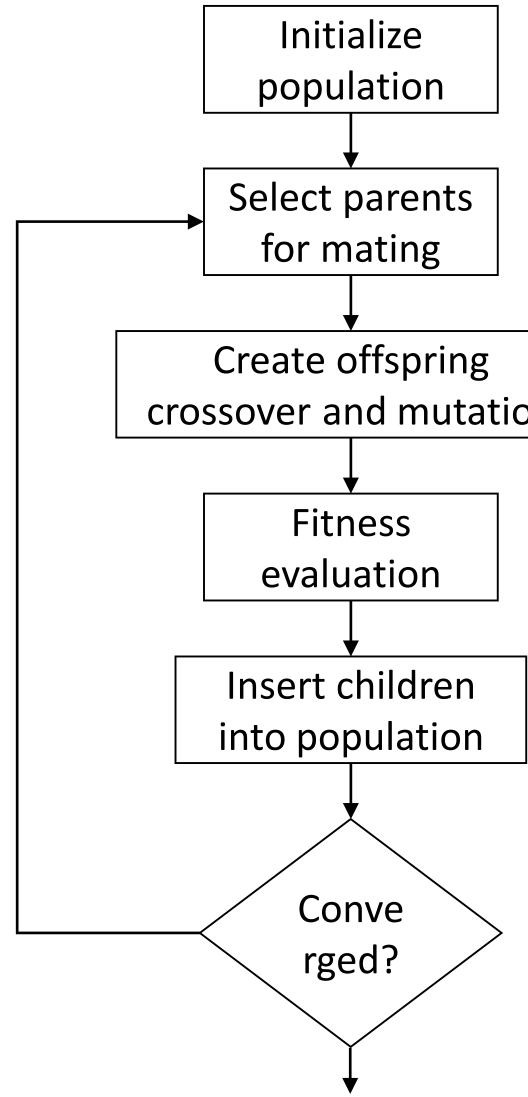
The Complex method



Genetic Algorithms

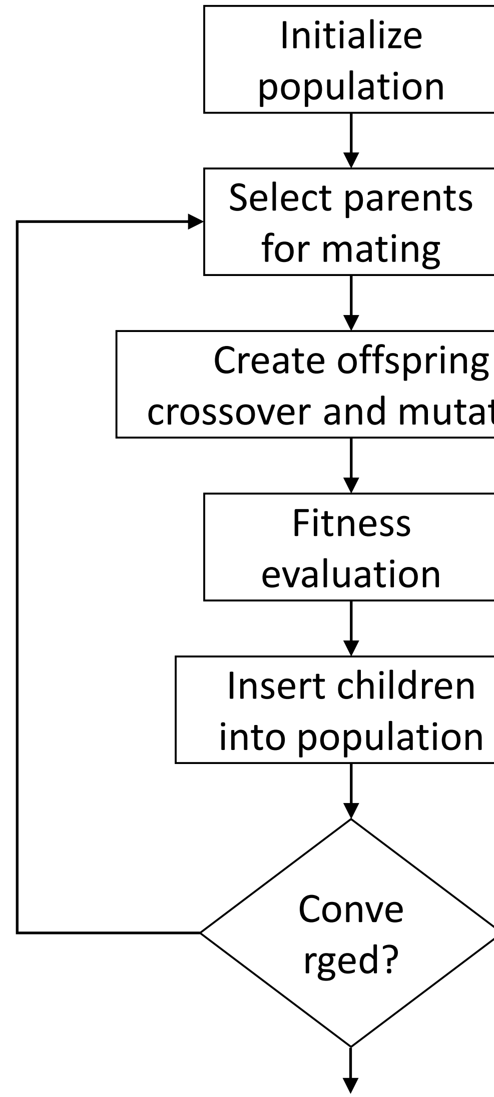
- Mimics Darwin's survival of the fittest

- Uses a fixed number of individuals in each generation



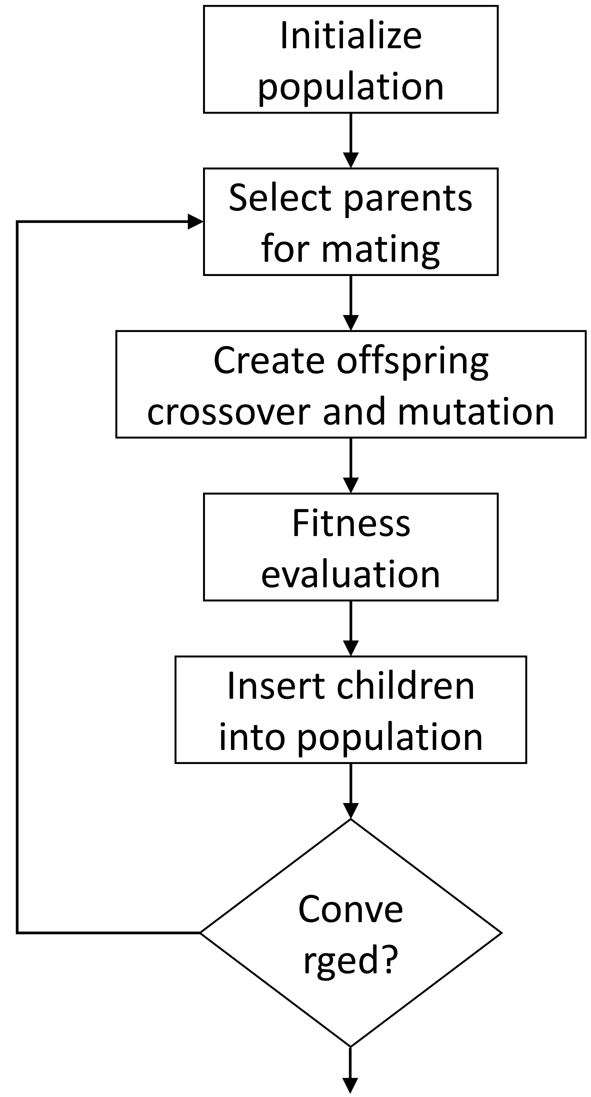
Genetic Algorithms

1. Spread out the initial population in the design space
2. Evaluate all individuals
3. Select the best individuals for mating
4. Create new individuals based on the parents



Genetic Algorithms

5. Evaluate the children
6. Replace the worst parents with the best children
7. Converged?
   – End or go to 3.



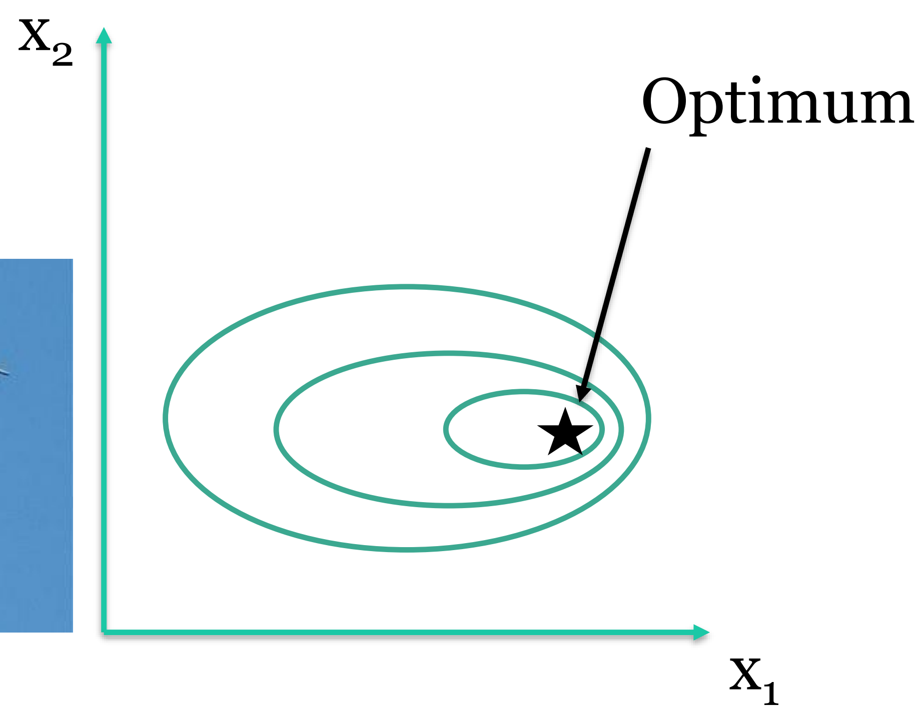
LINKÖPING UNIVERSITY

GA – Important Parameters

- Population Size
- Number of Generations
  - Number of evaluated designs = Population Size * Number of Generations * Generation Gap
- Mutation Rate
  - The chance that an individual has its variable values set randomly
  - Large value -> Explore larger parts of the design space, risque of no convergence
  - Small value -> Faster convergence, risque of missing global optimum

Particle Swarm Optimization

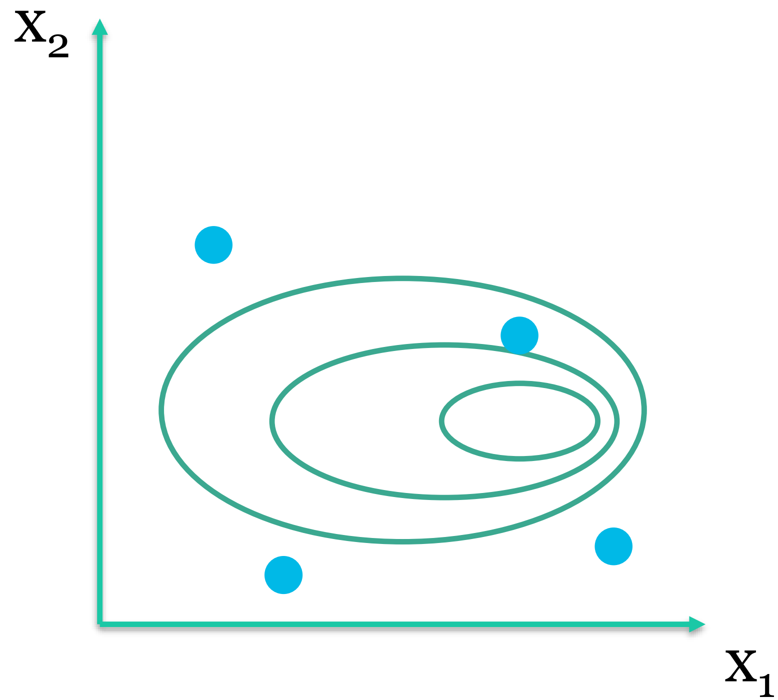
- Mimics animals that live in swarms / packs
- For example Seagulls



Particle Swarm Optimization

- The algorithm consists of swarm with a number of individuals that are constant during the optimization

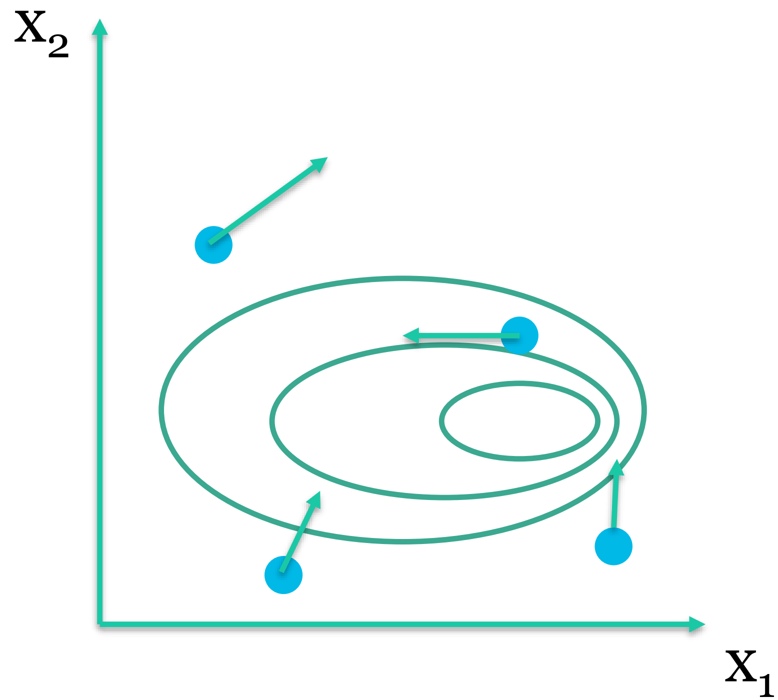
- The individuals start at different locations in the design space



Particle Swarm Optimization

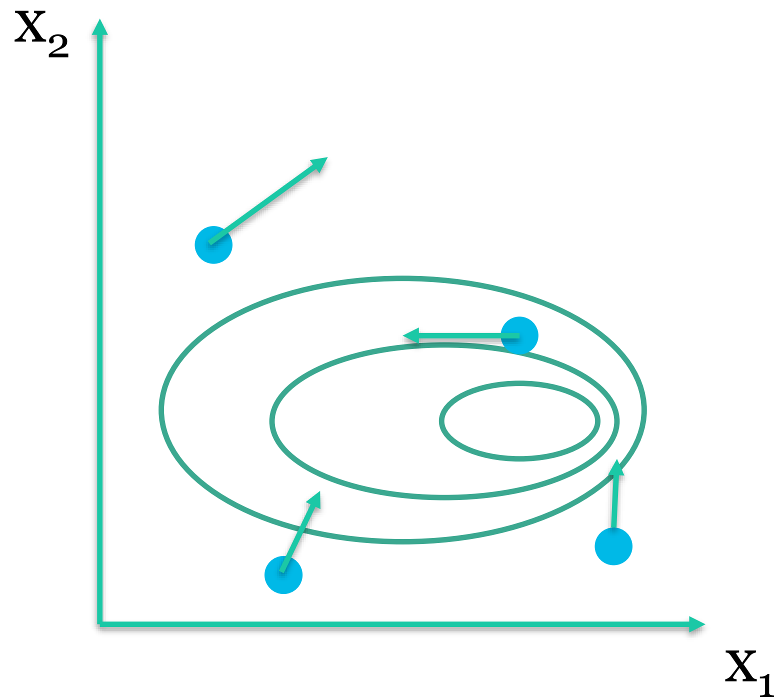
- Each individual is given an initial speed and direction

- The objective function value of each individual is also calculated



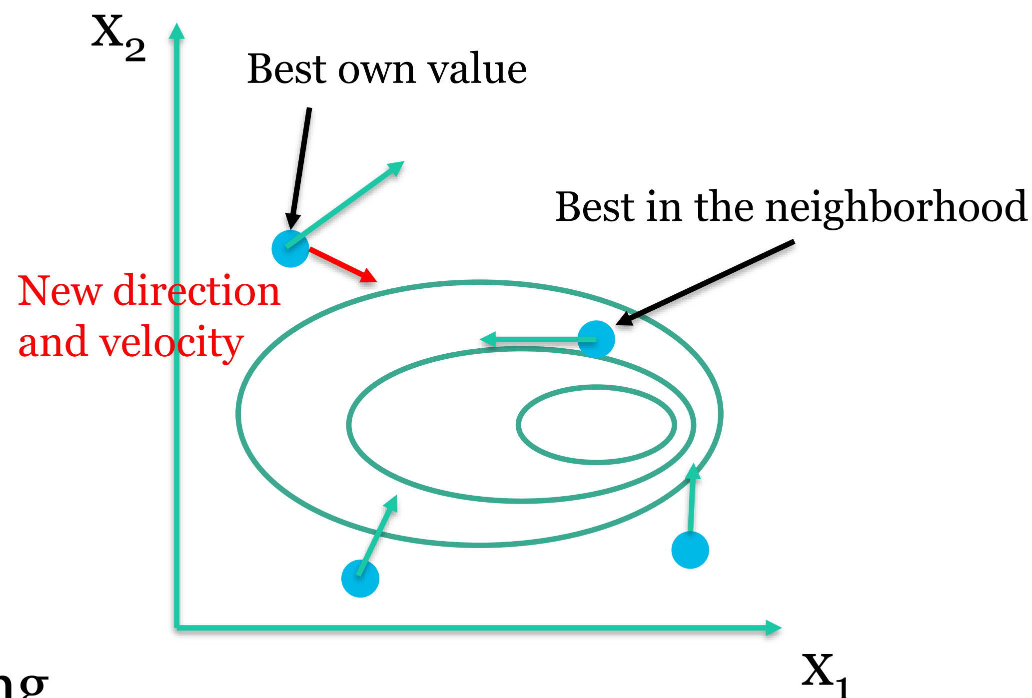
Particle Swarm Optimization

- Each individual will track its best position during the optimization



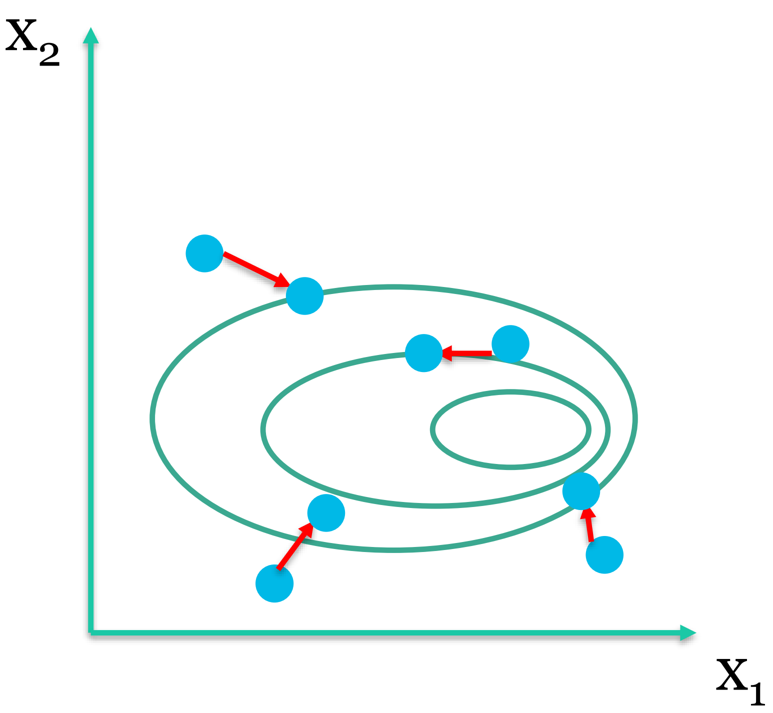
Particle Swarm Optimization

- The new velocity and direction will be a combination of
  - The previous velocity and direction
  - The best position the individual has visited
  - The best position that any neighboring individual has found



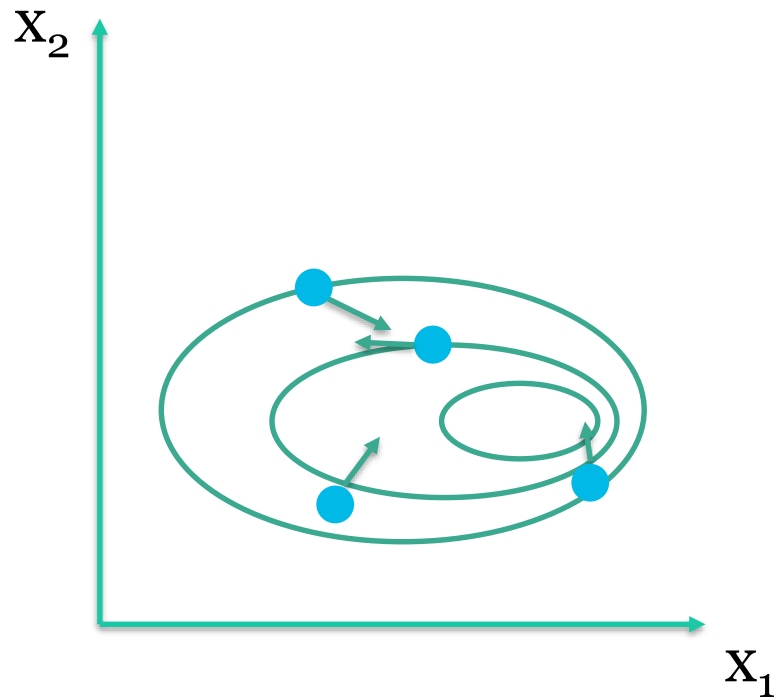
Particle Swarm Optimization

- Move all individuals to their new locations
- Evaluate their objective function values
- Update their best locations found



Particle Swarm Optimization

- The individuals will slowly move around towards the optimum until a stop criteria is met
  - No improvement in objective function value
  - Maximum number of evaluations



Summary Optimization Algorithms

	Gradient-Based	Population Based (GA, PSO)	Simlex / Complex
Finding the global optimum	Sometimes	Often	Medium
Number of evaluations	Low	High	Quite low
Comment	Sensitive to starting point(s)		Good trade-off between speed and accuracy

- Use a population based algorithm if you have the time to wait for the results
- Use gradient-based methods for simple problems that are thought to only have one optimum

Multi-Objective Optimization

Multi-Objective Optimization

$$\min \mathbf{F}(\mathbf{x}) = \left(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x}) \right)^T$$

$$s.t. \;\; \mathbf{x} \in S$$

$$\mathbf{x} = \left(x_1, x_2, \ldots, x_n \right)^T$$

k = *number of objectives*

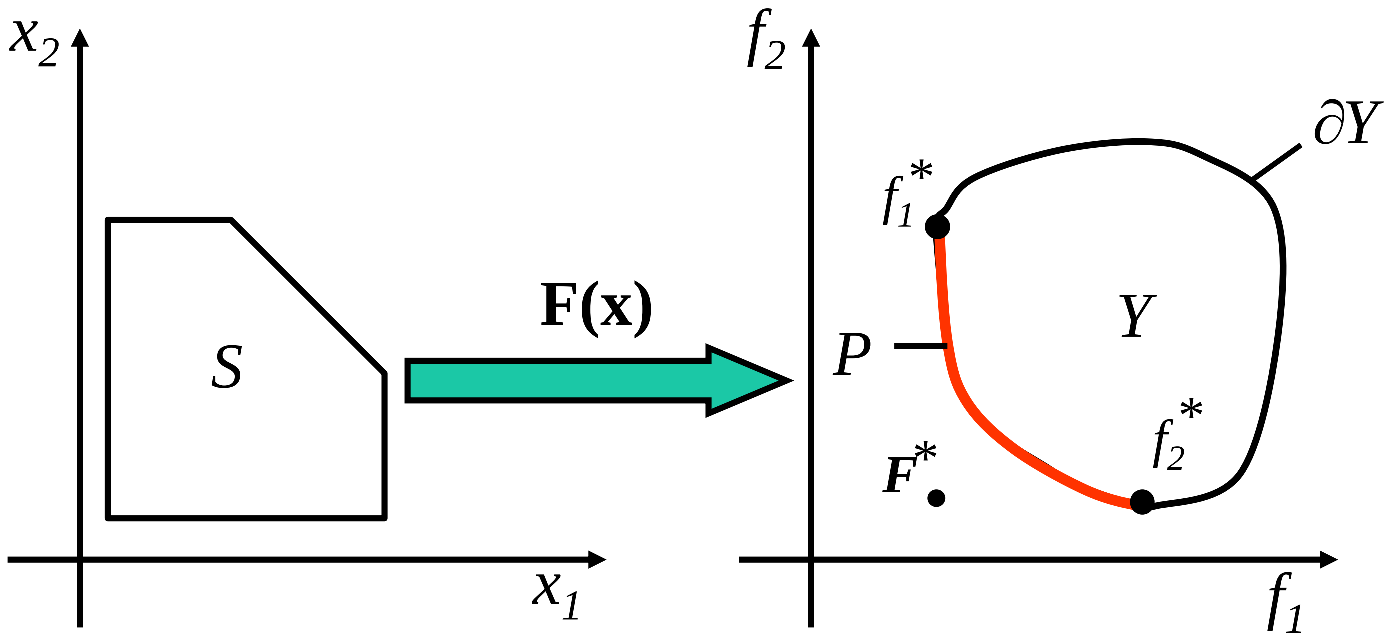
n = *number of parameters*

f_i = system characteristic, or sub-objective

x_i = optimization variables

S = solution space

Problem Visualisation



S = parameter space

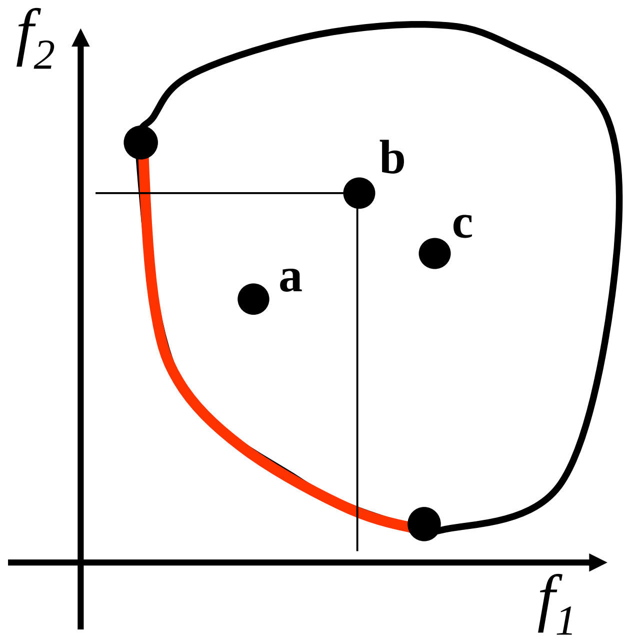
Y = objective or attribute space

f_i^*= individual optima

F^*= utopian solution

P = Pareto optimal front

Pareto Dominance



a is said to dominate b, ($\mathbf{a} \succ \mathbf{b}$), if:

$$\forall i \in \{1, 2, \ldots, k\} : f_i(\mathbf{a}) \le f_i(\mathbf{b}) \quad and \quad \exists j \in \{1, 2, \ldots, k\} : f_j(\mathbf{a}) < f_j(\mathbf{b})$$

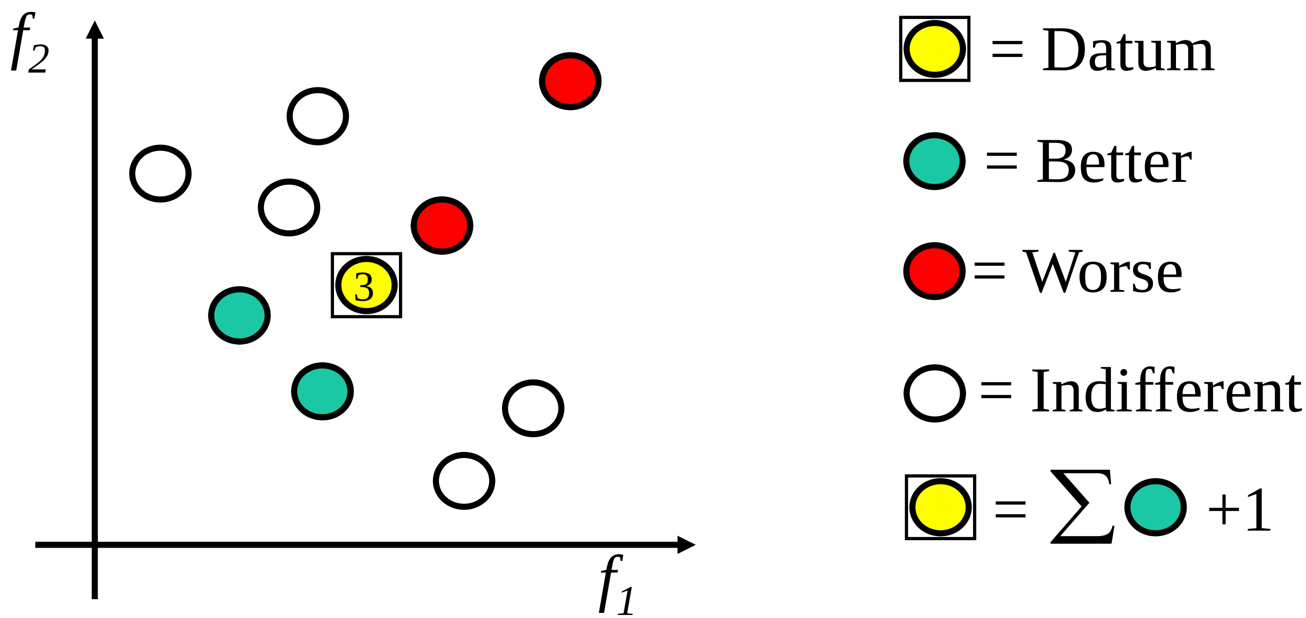
Multi-Objective Genetic Algorithms

- Tries to spread the population evenly on the Pareto front as the GA evolves.
- Identify the Pareto front in one optimization run.

- Two common
  - Non-dominated sorting GA (NSGA)
  - Multi-objective GA (MOGA)

MOGA

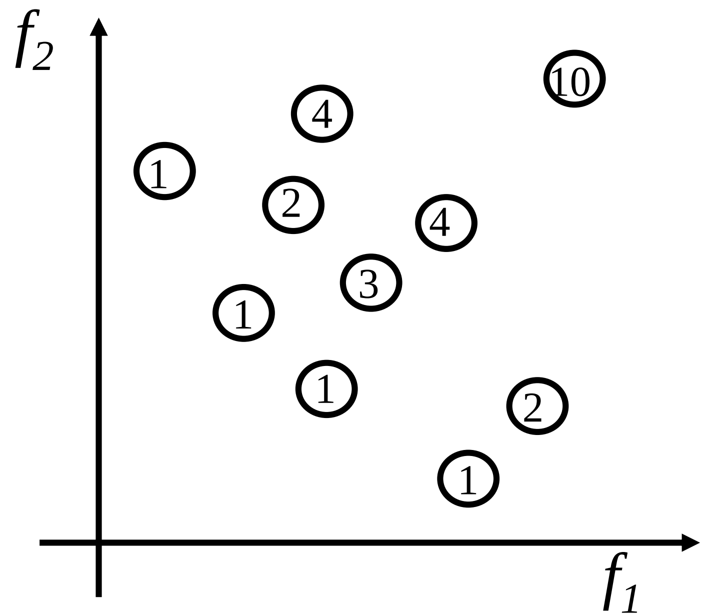
- Use Pareto dominance to rank the population



(Fonseca and Flemming, 1995)

MOGA

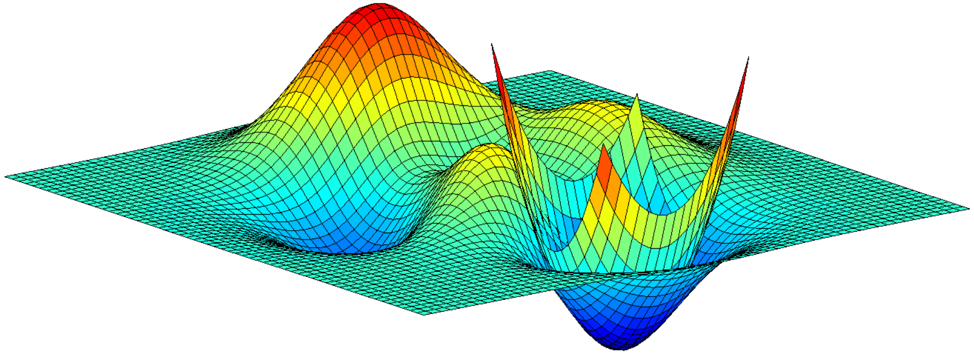
- Use Pareto dominance to rank the population



Surrogate Models

Surrogate Models

- Also known as metamodels (models of models)
- Numerically efficient reanimations of systems or other models
- Are used to model unknown systems
- Can replace computationally expensive models to enable optimization



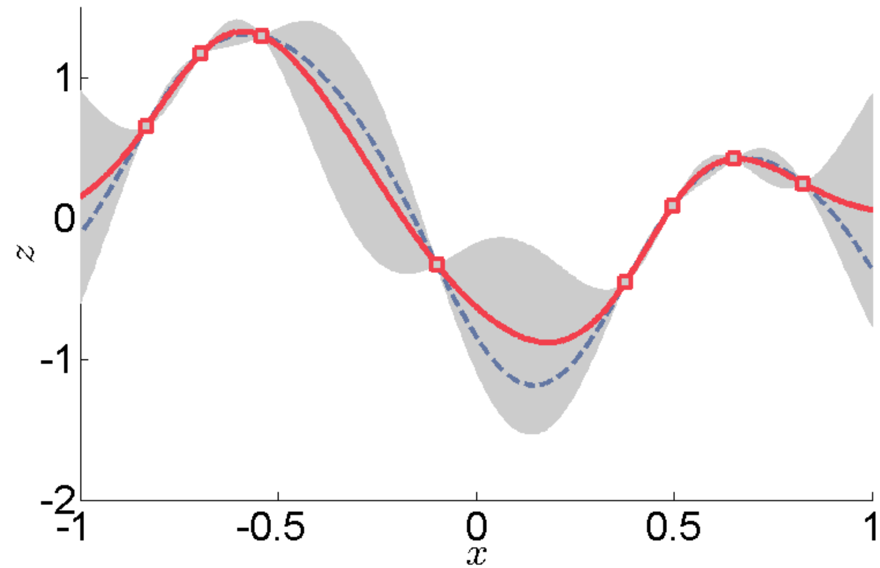
LINKÖPING UNIVERSITY

How to use surrogate models

- Collect the data needed
  - Experiments / Simulation
- Create a model that reanimates the data from the experiments
- Perform an optimization of the surrogate model to find an optimal design
- Verify the optimal design
  - Experiment / Simulation

Common Types of Surrogate Models

- Polynomial Response Surfaces
- Neural Networks
- Kriging
- Radial Basis Functions
- Support Vector Regression



Wikipedia: Kriging

Polynomial Response Surfaces

- Approximates the desired entity as a polynomial of desired degree

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2 = \mathbf{X\beta}$$

- Can be fitted with the linear least squares method

$$\mathbf{X\beta} = \mathbf{y}$$

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^T \mathbf{X}\right)^{-1} \mathbf{X}^T \mathbf{y}$$

Polynomial Response Surfaces

- Approximates the desired entity as a polynomial of desired degree

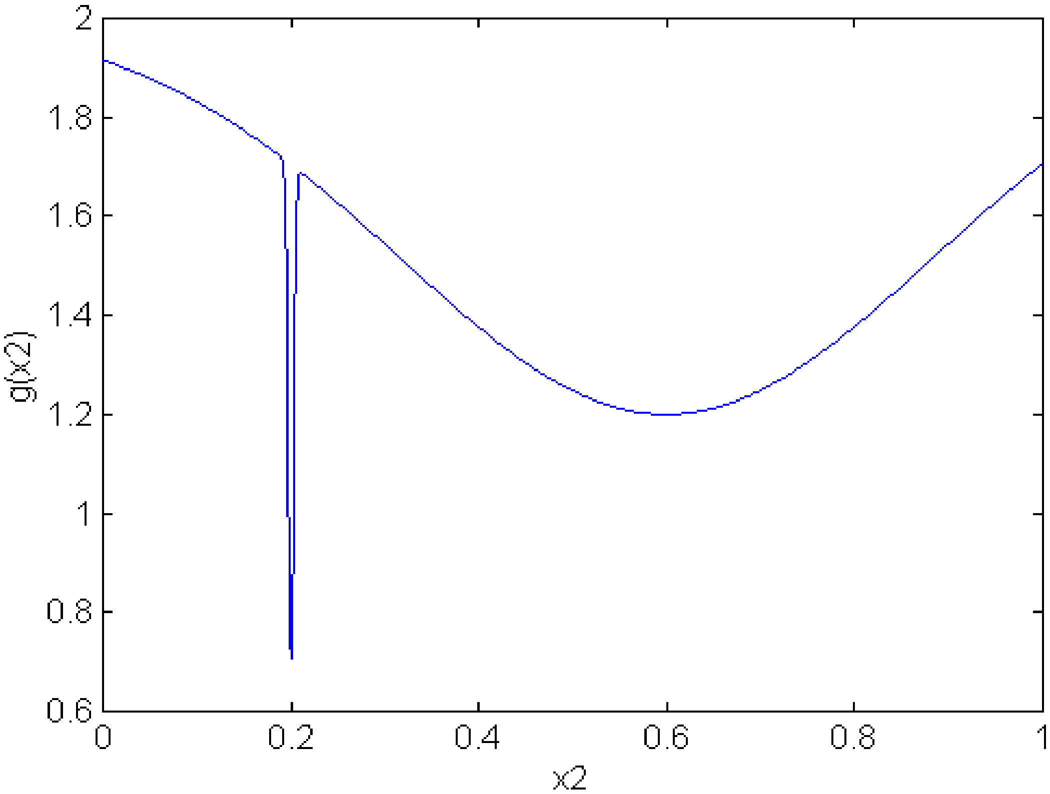
$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2 = \mathbf{X\beta}$$

- Pros
  - Easy to implement and understand
  - Computationally fast creation (matrix problem)
- Cons
  - Unsuitable for problems with many parameters
    - Too many samples are needed

Sensitivity Analysis

Sensitivity Analysis

- We want to see how robust a suggested solution is
  - Choose a robust solution
  - Identifies the variables that are not so important



Local Sensitivity Analysis

Changes in design characteristics = Sensitivity matrix * Changes in design variables

$$
\begin{bmatrix} \mathrm{D}DC_1 \\ \mathrm{D}DC_2 \\ \vdots \\ \mathrm{D}DC_m \end{bmatrix}
=
\begin{bmatrix}
\frac{\partial DC_1}{\partial x_1} & \frac{\partial DC_1}{\partial x_2} & \cdots & \frac{\partial DC_1}{\partial x_n} \\
\frac{\partial DC_2}{\partial x_1} & \frac{\partial DC_2}{\partial x_2} & \cdots & \frac{\partial DC_2}{\partial x_n} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial DC_m}{\partial x_1} & \cdots & \cdots & \frac{\partial DC_m}{\partial x_n}
\end{bmatrix}
\begin{bmatrix} \mathrm{D}x_1 \\ \mathrm{D}x_2 \\ \vdots \\ \mathrm{D}x_n \end{bmatrix}
$$

$$\Delta DC = S * \Delta x$$

Local Sensitivity Analysis

$$\Delta DC = S * \Delta x$$

- Analytical Derivative

$$S_{ij} = \frac{\partial DC_i}{\partial x_j}$$

- Partial Derivative

$$S_{ij} = \frac{DC_i(x+h) - DC_i(x)}{h}$$

Normalize the Sensitivities

- To enable comparison of variables and DC's of different orders of magnitudes

$$S_{norm,ij} = S_{ij}\frac{x_{j0}}{DC_{i0}} = \frac{\partial DC_i}{\partial x_j}\frac{x_{j0}}{DC_{i0}}$$

LINKÖPING UNIVERSITY

Questions?