1. Equality Constraints

\[ f(\vec{x}) \rightarrow \text{min}, \quad g_j(\vec{x}) = 0 \]

- Elimination:
  \[ x_1 \cdot x_2 \rightarrow \text{min}, \quad x_1 + x_2 = 1 \]
  \[ x_1(1 - x_1) \rightarrow \text{min} \]
  (not always possible)

- Lagrange multipliers:
  \[ f(\vec{x}, \lambda) = f(\vec{x}) + \sum_{j=1}^{m} \lambda_j g_j(\vec{x}) \]

To find \( \vec{x}^* \):

\[ \frac{\partial f}{\partial x_1} = 0, \quad \frac{\partial f}{\partial \lambda} = 0 \]

(system of equations)

Example:

\[ f(x_1, x_2) = -\frac{1}{2} x_1 x_2 \rightarrow \text{min} \]
\[ x_1^2 + x_2^2 - r^2 = 0 \]
\[ \Rightarrow f(\vec{x}, \lambda) = -\frac{1}{2} x_1 x_2 + \lambda(x_1^2 + x_2^2 - r^2) \]

Result:

\[ x_1^* = x_2^* = \frac{r}{\sqrt{2}}, \quad \lambda^* = \frac{1}{2} \]

2. Inequality Constraints

- In case of explicit (constant) bounds:
  Transformation

\[ x_i \geq 0 \quad \iff \quad x_i = \frac{y_i^2}{\exp(y_i)} \]
\[ 0 \leq x_i \leq 1 \quad \iff \quad x_i = \sin^2 y_i \]
\[ a_i \leq x_i \leq b_i \quad \iff \quad x_i = a_i + (b_i - a_i) \sin^2 y_i \]

Optimization under Constraints 2.3

- In case of implicit bound:
  Penalty function

\[ f(\vec{x}, r^{(p)}) = f(\vec{x}) + r^{(p)} \sum_{j=1}^{m} \left[ \frac{w_j}{g_j(\vec{x})} \right] \]
\[ + \frac{1}{r^{(p)}} \sum_{k=1}^{l} \left[ w_k h_k^2(\vec{x}) \right] \rightarrow \text{min} \]

Original problem:

\[ f(\vec{x}) \rightarrow \text{min} \]
\[ g_j(\vec{x}) \geq 0 \]
\[ h_k(\vec{x}) = 0 \]

Sequence of optimizations, for \( p = 1, 2, \ldots \)

\[ r^{(p)} > r^{(p+1)} > 0 \]

SUMT:
  Sequential
  Unconstrained
  Minimization
  Technique
  (Fiacco & McCormick)

Visualization of SUMT Principles:
Special case:

Searching for a feasible point as starting point for optimization.

\[ f(\vec{x}) = - \sum w_i g_i(\vec{x}) \rightarrow \min \]

Where \( w_i = \begin{cases} 1 & \text{if } g_i(\vec{x}) < 0 \\ 0 & \text{otherwise} \end{cases} \)

Stop, if \( \vec{f}(\vec{x}) = 0 \)

Continue with optimization method, which can handle inequality constraints

---

Optimization under Constraints 2.7

**Multiple Optimization Criteria**

- **Motivation:**
  Multiplying conflicting measures of performance
  \[ \vec{f}: M \rightarrow \mathbb{R}^k, \quad \vec{f}(\vec{x}) = (f_1(\vec{x}), ..., f_k(\vec{x})) \]

- **Improvement in any combination of objectives not possible without degradation in the remaining:** Solution \( \vec{x}_i \) is Pareto-optimal (non-dominated)
  \[ \Leftrightarrow \beta \vec{x}_i: \vec{f}(\vec{x}_i) \preceq \vec{f}(\vec{x}) \text{, where} \]
  \[ \vec{f}(\vec{x}_j) \prec \vec{f}(\vec{x}_i) \Leftrightarrow \forall p \in \{1, ..., k\}: f_p(\vec{x}_j) \leq f_p(\vec{x}_i) \land \exists p \in \{1, ..., k\}: f_p(\vec{x}_j) < f_p(\vec{x}_i) \]
  (assuming minimization)

- **Pareto-set:** Set of all Pareto-optimal solutions

- **Cost assignment with multiple objectives:**
Decision making problem, human preferences

---

**Multiple Optimization Criteria**

- **MCDM:** Multiple criteria decision making
  Vector-optimization
  \[ f: M \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^k \]
  \( I \) (conflicting) goals \( f_1(\vec{x}), ..., f_k(\vec{x}) \)

- **Goal:** Find efficient / Pareto-optimal / non-dominated solutions

---

Optimization under Constraints 2.8

**Example:** 2-dim. criterion space

**CRITERIA SPACE**
(cost versus reliability)

- Efficient solutions
- Single-criteria optima

---

**Bioinformatics and Optimization**

---

**Bioinformatics and Optimization**
Multiple Optimization Criteria

a) “Pathological case” of identical objectives

b) “Classical case” of identical objectives

c) Pareto set can be unconnected

Efficient solutions: \( \bar{x} \)

None of the goals can be improved anymore, without making at least one other goal worse

\[
\forall \bar{x} \in M : \quad f_k(\bar{x}') \leq f_k(\bar{x}) \quad \forall k \in \{1, \ldots, l\}
\]

and

\[
f_k(\bar{x}') < f_k(\bar{x})
\]

for at least one \( k \)

Also called: Non-dominated

A further selection from the Pareto-set requires subjective criteria in addition

\( \Rightarrow \) Decision maker!

Optimization under Constraints 2.11

Traditional MCDM approaches:

- Weighted sum
  \[
f(\bar{x}) = \sum_{k=1}^{l} w_k f_k(\bar{x})
  \]
  \( w_k \): Subjectively chosen weight factors

- Minimax approach / goal programming
  \[
  \Phi(\bar{f}(\bar{x})) = \max_{i=1,\ldots,k} \left\{ \frac{f_i(\bar{x}) - g_i}{w_i} \right\} \rightarrow \min
  \]
  - \( g_i \): are goals specified by the user goal attainment
  - \( w_i \) smaller \( \Rightarrow \) Objective \( i \) becomes harder
  - Often results in non-smooth fitness functions
  - Not guaranteed to produce strictly non-dominated solutions

- Target vector approach
  \[
  \Phi(\bar{f}(\bar{x})) = \| (\bar{f}(\bar{x}) - \bar{g}) \cdot \bar{w}^{-1} \|_2 \rightarrow \min
  \]
  - Minimizes (weighted) distance from target vector
  - \( g_i \): Target vector, specified by user
  - \( \bar{w} \): (Diagonal) weight matrix
  - Euclidean norm: \( \alpha = 2 \)

Optimization under Constraints 2.12

Additional Difficulties / Aspects:

- Discrete variables (e.g., integer, \( \{0, 1\} \))

\( \Rightarrow \) Minimize (estimated) expectation

\( \Rightarrow \) Stochastic Approximation algorithm (\( n = 1 \))

\[
x^{(k+1)} = x^{(k)} - 2\alpha^{(k)} \frac{F(x^{(k)} + c^{(k)}) - F(x^{(k)} - c^{(k)})}{2\alpha^{(k)}}
\]

\[
a^{(k)} = \frac{1}{k} a^{(0)}, \quad a^{(0)} > 0
\]

\[
c^{(k)} = \frac{1}{\sqrt{k}} c^{(0)}, \quad c^{(0)} > 0
\]

Gradient method with trial \( c^{(k)} \) / work \( a^{(k)} \) steps
• Moving Optima

**Dynamic optimization**: Maintain an optimal condition in the face of varying (time) conditions of the environment

Permanent on-line optimum search
Information becoming obsolete over time
→ “forgetting” necessary

Examples:
- Traffic control optimization
- Elevator control optimization
- Free-flight aircraft routing

→ Adaptive control systems:
  Set up an internal model over time; learn
→ If output quantities themselves (i.e. objective function) used to adjust control system: Self-learning

---

**Efficiency**

1: Specialized method for a very restricted problem class
2: Total enumeration, Monte Carlo, random walk: Widely application, but bad efficiency
3: Robust heuristics: Widely applicable, with good efficiency

E.g.: Genetic Algorithms, evolution strategies, Simulated Annealing

---

**Optimization under Constraints**

**Applicability of the “school-method”?**

Necessary cond.: Determine 1\(^{st}\) derivative and its zero(\(\epsilon\)), i.e. \(f'(x) = 0\)

Sufficient cond.: Determine 2\(^{nd}\) derivative and its sign for zero(\(\epsilon\)) \(x^*\)

\[ f''(x^*) = \begin{cases} 
  > 0 & \Rightarrow x^* \text{ minimum} \\
  < 0 & \Rightarrow x^* \text{ maximum} \\
  = 0 & \Rightarrow ?
\end{cases} \]

\(f''(x^*) = 0\): Determine k\(^{th}\) derivative,
until \(f^{(k)}(x^*) \neq 0\)

\(k \geq \text{even}\) \(\Rightarrow\) \(f^{(k)}(x^*) > 0\) minimum
\(k \geq \text{odd}\) \(\Rightarrow\) \(f^{(k)}(x^*) < 0\) maximum

\((n = 1: \text{MacLaurin, 1742})\)
\((n > 1: \text{Scheefer & Stoiz 1886, 1894})\)

Precondition: \(f(x)\) k-fold (partially) differentiable, i.e. analytically given!
No constraints!

---

**Robustness - intuitively:**

1: combinatorial
2: unimodal
3: multimodal

1: Specialized method for a very restricted problem class
2: Total enumeration, Monte Carlo, random walk: Widely application, but bad efficiency
3: Robust heuristics: Widely applicable, with good efficiency

E.g.: Genetic Algorithms, evolution strategies, Simulated Annealing