# FTP_OPTIMIZATION

## 1. Introduction

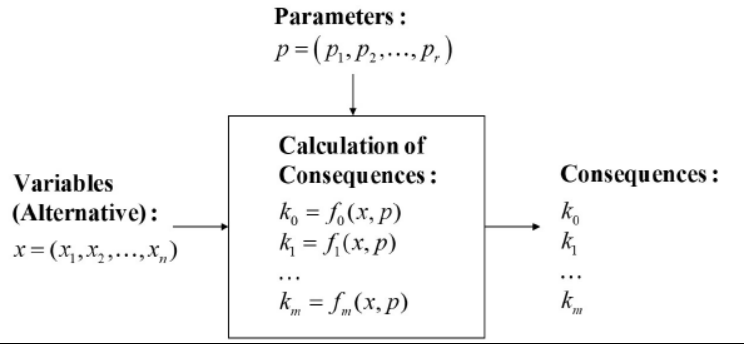
| | |
|---|---|
| **Two main areas** | Optimization of **business process** (production, logistics, services, operations, management)<br>Optimization of **technical processes** (engineering) |
| **Quantitative vs. Qualitative** | **Quantitative** analysis and optimization (numerical, measurable data, mathematical models/algorithms)<br>**Qualitative** analysis and optimization (informal facts, verbal description of processes and procedures) |
| **typically progress** | Phase 1: Qualitative analysis (up to 80%, unclear problem description, mess of information)<br>Phase 2: Qualitative or quantitative -> handled in this course (need for decision support) |
| **Types**<br><br><br>this course -> | **Continuous Optimization**: **infinitely** solutions, represented by **continuous variables**<br>local optimization based on differential information (1st (gradient) & 2nd derivative),<br>very difficult if non-continuous or non-differentiable, more difficult if constraints (Nebenbedingungen)<br>**Discrete Optimization (DO)**: **finitely** solutions, represented by **integer variables**<br>trivial algorithm (enumartion), can solve real world problems since invention of computers<br>find an "efficient" algorithm in a "reasonable" time to solve a specific problem -> **Complexity Theory** |
| **Importance of Linearity** | Finite set of solutions -> Discrete optimization<br>Solution can be represented by a list of variables $vector\ \vec{x} = (x_1, x_2, \dots, x_n)^T$<br>Solution is a finite set of points in n-dimensional space (e.g. convex hull)<br>Finite mesh implicite linearity! |
| **Decision Problems** | **Decision Support** (Entscheidungsunterstützung)<br>a) quantitative models<br>b) qualitative approach<br>**Decision maker** (Entscheidungsträger)<br>**Alternatives** (multiple possible decisions) with associated **consequences** (**deterministic** or **stochastic**)<br>**Evaluation** (Bewertung) of alternatives with regard to their consequences |
| **Evaluation of Consequences** | **Satisfication**: Consequences has to fullfil certain **constraints** (Ger: Restriktionen),<br>in order to have a **feasible** (Ger: zulässig) alternative.<br>**Optimization**: Consequences has to reach best possible value, most be **optimal** among all alternatives. |
| **Introduction Examples** | 1. Frequency Assignment in Mobile Networks<br>2. Product Mixture in an Oil Refinery<br>3. Vehicle Dispatching in a Car Rental Company<br>4. Shift Planning in a Department Store<br>5. Design of a Regional Optical Fiber Network<br>6. Sudoku |

## 2+3. Mathematical Models

### Descriptive Models

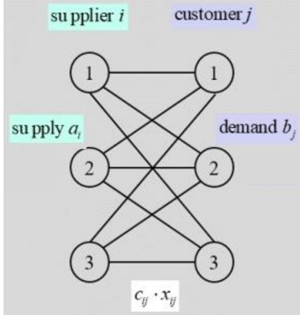
also called: "**Evaluation Models**"
Question: "What if?"

Calculates for a given alternative the resulting consequences.
e.g. Problem 3: vehicle dispatching (see Excel spreadsheet)
Variables: User specified
Parameters: Given
Consequences: Calculate




**5 things to notice with vehicle displatching problem**

| | | |
|---|---|---|
| Sets | $I$ | Set of locations |
| Parameters | $a_i$<br>$b_j$<br>$c_{ij}$ | Number of available vehicles at location $i$<br>Number of requested vehicles at location $j$<br>Distance (km) from location $i$ to location $j$ |
| Variables / Alternatives | $x_{ij}$ | Number of vehicles transferred from location $i$ to location $j$ |
| Consequences | $k_0$<br>$k_i^{Out}$<br>$k_j^{In}$ | Total distance (km) of all transfer<br>Number of vehicles transferred out of location $i$<br>Number of vehicles transferred into location $j$ |
| Model | $k_0=\sum_{i\in I}\sum_{j\in I} c_{ij}x_{ij}$ | $k_i^{Out}=\sum_{j\in I} x_{ij}$    $k_j^{In}=\sum_{i\in I} x_{ij}$ |



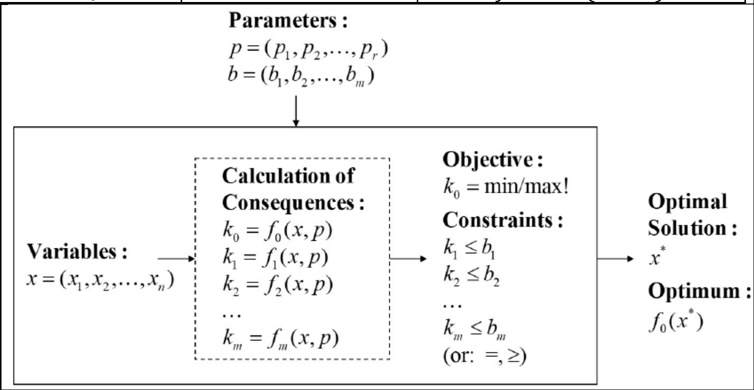

$i\in I=\{1\dots n\}$
$j\in I=\{1\dots n\}$

### Optimization Models

also called: "**Prescriptive Models**"
Question: "What's best?"

Calculates in the set of all **feasible** alternatives an **optimal** alternative
Set of all feasible solutions: **solution space**

Optimization algorithms needed!
-> Operations Research




**5 things to notice**

| | | | |
|---|---|---|---|
| Sets | $I$ | Set of locations | $\{1..n\}$ |
| Parameters | $a_i$<br>$b_j$<br>$c_{ij}$ | Number of available vehicles at location $i$<br>Number of requested vehicles at location $j$<br>Distance (km) from location $i$ to location $j$ | |
| Variables | $x_{ij}$ | Number of vehicles transferred from location $i$ to location $j$ | $i\in I$<br>$j\in I$ |
| Constraints | $\sum_{j\in I} x_{ij}\le a_i$ | $\sum_{i\in I} x_{ij}\ge b_j$    $x_{ij}\ge 0$ | |
| Objective Function | | $\min\sum_{i\in I}\sum_{j\in I} c_{ij}x_{ij}$ | |

**Example**

variables / alternative

| x_11 | x_12 | x_13 | x_21 | x_22 | x_23 | x_31 | x_32 | x_33 |
|---|---|---|---|---|---|---|---|---|
| 4 | 3 | 0 | 0 | 5 | 0 | 0 | 3 | 6 |

| | | | | | | | | | | | consequences | constraints | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| i=1 | 1 | 1 | 1 | | | | | | | a_1 | 7 | <= | 7 | available |
| i=2 | | | | 1 | 1 | 1 | | | | a_2 | 5 | <= | 5 | |
| i=3 | | | | | | | 1 | 1 | 1 | a_3 | 9 | <= | 9 | |
| j=1 | 1 | | | 1 | | | 1 | | | b_1 | 4 | >= | 4 | requested |
| j=2 | | 1 | | | 1 | | | 1 | | b_2 | 11 | >= | 11 | |
| j=3 | | | 1 | | | 1 | | | 1 | b_3 | 6 | >= | 6 | |

parameters / constants

| c_11 | c_12 | c_13 | c_21 | c_22 | c_23 | c_31 | c_32 | c_33 |
|---|---|---|---|---|---|---|---|---|
| 0 | 20 | 35 | 20 | 0 | 19 | 35 | 24 | 0 |

objective
min
**132**

## General Optimization Model

| | |
|---|---|
| General optimization problem | $\Pi: \max\{f(\vec{x}): \vec{x} \in S\}$ |
| Decision variables | $\vec{x} = (x_1 \dots x_n)^T \in \mathbb{R}^n$ |
| (Feasible) solutions | $\vec{x} \in S$ |
| Solution space | $S \subseteq \mathbb{R}^n$ |
| Objective function | $f: S \to \mathbb{R}$ |
| Optimal solution (Optimizer) | $\vec{x}^* \in S$ such that $f(\vec{x}^*) \geq f(\vec{x})$ for all $\vec{x} \in S$ |
| Optimum (Optimal value) | $f: (\vec{x}^*)$ |

## Conditions for Existence of Optimum

| | |
|---|---|
| Feasibility | $S \neq \emptyset$ |
| Ex. Infeasibility | $\max\{x_1: 2x_1 + 4x_2 = 5, \vec{x} \in \mathbb{Z}^2\}$ |
| Boundedness | feasible, $\exists \omega: f(\vec{x}) \leq \omega$ for all $\vec{x} \in S$ |
| Ex. Unboundedness | $\max\{x_1: 2x_1 + 4x_2 = 5, \vec{x} \in \mathbb{R}^2\}$ |
| Closedness | feasible, bounded, optimum exists |
| Ex. Unclosedness | $\max\{x_1: x_1 < 1, \vec{x} \in \mathbb{R}\}$ |

## Example

A company produces different types of **feed** for farm animals by mixing several **ingredients**. Each ingredient contains a certain amount of **protein** and **calcium** (given in gram per kg), and each type of feed requires a **minimum total amount of protein and calcium** (given in gram per kg). Furthermore, the **purchase price for each ingredient** is given (in dollar per kg), and the **sales price for each type of feed** is given (in dollar per kg).
Finally, the production quantity of each feed type should not exceed a specified limit (in kg).
Formulate a linear programming model which calculates an optimal production plan, i.e. a production plan that maximizes total profit.

**Sets :**

$I$ Set of ingredients, $I = \{1, \dots, m\}$

$J$ Set of feed types, $J = \{1, \dots, n\}$

**Parameters :**

$a_i^{\text{Prot}}$ Amount of protein (gram per kg) contained in ingredient $i$, $i \in I$

$a_i^{\text{Calc}}$ Amount of calcium (gram per kg) contained in ingredient $i$, $i \in I$

$d_i^{\text{Prot}}$ Total amount of protein (gram per kg) required for feed type $j$, $j \in J$

$d_i^{\text{Calc}}$ Total amount of calcium (gram per kg) required for feed type $j$, $j \in J$

$f_i$ Purchase price (dollar per kg) for ingredient $i$, $i \in I$

$c_j$ Sales price (dollar per kg) for feed type $j$, $j \in J$

$b_j$ Maximum production quantity (kg) for feed type $j$, $j \in J$

**Variables :**

$x_{ij}$ Amount (kg) of ingredient $i$ mixed into feed type $j$, $i \in I, j \in J$

$$\max \sum_{j \in J} c_j \sum_{i \in I} x_{ij} - \sum_{i \in I} f_i \sum_{j} x_{ij}$$

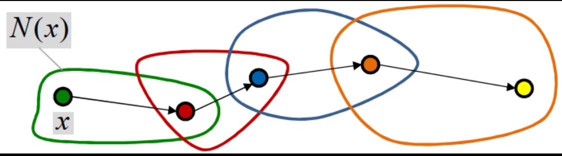
$$\sum_{i \in I} a_i^{\text{Prot}} x_{ij} \geq d_j^{\text{Prot}} \sum_{i \in I} x_{ij}, \quad j \in J$$

$$\sum_{i \in I} a_i^{\text{Calc}} x_{ij} \geq d_j^{\text{Calc}} \sum_{i \in I} x_{ij}, \quad j \in J$$

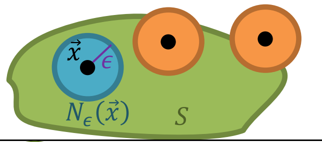
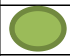
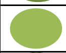
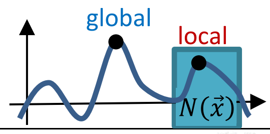
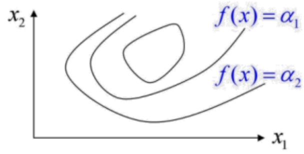
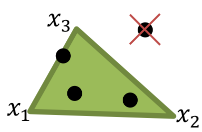
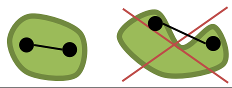
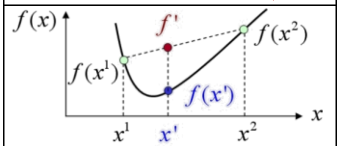
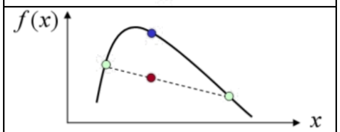
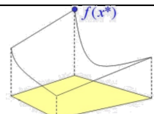
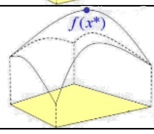
$$\sum_{i \in I} x_{ij} \leq b_j, \quad j \in J$$

$$x_{ij} \geq 0, \quad i \in I, j \in J$$

## Basic Concepts

| | | |
|---|---|---|
| **Problem and Problem Instances** | Problem | $e.g.: \max \sum_{j=1}^{n} c_j x_j$ |
| | Problem Instance | $e.g.: \max\{4x_1 + 7x_2 : 3x_1 + 5x_2 \le 17, \vec{x} \in \mathbb{R}^2\}$ |
| **Powerset** | $S = \{1,2,3\}$<br>$P(S) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\}$ | $\lvert P(S) \rvert = 2^{\lvert S \rvert}$ |
| **Neighborhood** user defined | Neighborhood | $N: S \to P(S)$ |
| | Neighbor solutions | $N(\boldsymbol{x}) \subseteq S$ |
| | Per definition | I am my own neighbor. |
| | Usage: Local Search Metaheuristics - search in my neighborhood for better solutions, repeat until best.<br>Usage: Euclidean Neighborhood |  |
| **Types of models** | Unconstrained vs Constrained<br>Global vs Local<br>Differentiable vs Non-Differentiable<br>Discrete vs Continuous | Convex vs Non-Convex<br>Linear vs Non-Linear<br>Exact vs Heuristic<br>General vs Problem specific |

## Notations

| | | | |
|---|---|---|---|
| Interior point $\vec{x}$ | $N_\epsilon(\vec{x}) \subset S$ | for some $\epsilon > 0$ |  |
| Boundary point $\vec{x}$ | $N_\epsilon(\vec{x}) \cap S \neq \emptyset \text{ and } N_\epsilon(\vec{x}) \cap (\mathbb{R}^n - S) \neq \emptyset$ | for all $\epsilon > 0$ | |
| S closed | all boundary points of S are in S | |  |
| S open | all points of S are interior points | |  |
| S bounded | $S \subset \{\vec{x} \in \mathbb{R}^n : \vec{a} \le \vec{x} \le \vec{b}\}$ | |  |
| Global Optima | $\vec{x}^* : f(\vec{x}) \ge f(\vec{x})$ | for all $\vec{x} \in S$ |  |
| Local Optima | $\vec{x}^* : f(\vec{x}) \ge f(\vec{x})$ | for all $\vec{x} \in N(\vec{x}) \cap S$ | |
| Graph | $H = \{(\vec{x}, f(\vec{x})) : \vec{x} \in S\}$ | | |
| Level set for level $\alpha$ | $L_\alpha = \{\vec{x} \in S : f(\vec{x}) = \alpha\}$ | |  |
| Convex combination | 2D: $\vec{x} = \lambda\vec{x^1} + (1-\lambda)\vec{x^2}$<br>$\vec{x} = \sum_{i=1}^{k} \lambda_i \vec{x}^i \text{ for some } \vec{\lambda} \in \mathbb{R}^k$ | $0 \le \vec{\lambda} \le 1$<br>$\sum_{i=1}^{k} \lambda_i = 1$ |  |
| Convex set $S \subset \mathbb{R}^n$ | $\lambda\vec{x^1} + (1-\lambda)\vec{x^2} \in S$<br>intersection of convex sets is convex | $\vec{x^1}, \vec{x^2} \in S$<br>$0 \le \lambda \le 1$ |  |
| Convex function | $f\left(\lambda\vec{x^1} + (1-\lambda)\vec{x^2}\right) \le \lambda f\left(\vec{x^1}\right) + (1-\lambda) f\left(\vec{x^2}\right)$ | |  |
| Concave function | $f\left(\lambda\vec{x^1} + (1-\lambda)\vec{x^2}\right) \ge \lambda f\left(\vec{x^1}\right) + (1-\lambda) f\left(\vec{x^2}\right)$ | |  |
| Linear function | is convex and concave | | |
| Convex optimization problem | $\max f(\boldsymbol{x}) : \boldsymbol{x} \in S$<br>-> every local optimum is on boundary | with $f$ convex<br>and S convex |  |
| | $\max f(\boldsymbol{x}) : \boldsymbol{x} \in S$<br>-> local optimum is the global optimum | with $f$ concave<br>and S convex |  |

## 4+5. Linear Programming

| | |
|---|---|
| **Problem Formulation** | $\boldsymbol{x} \in \mathbb{R}^n = \begin{Bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{Bmatrix}$ ; $\boldsymbol{A} \in \mathbb{R}^{m \times n} = \begin{Bmatrix} A_{11} & \dots & A_{1n} \\ \dots & \dots & \dots \\ A_{m1} & \dots & A_{mn} \end{Bmatrix}$ ; $\boldsymbol{b} \in \mathbb{R}^m = \begin{Bmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{Bmatrix}$ ; $\boldsymbol{c} \in \mathbb{R}^n = \begin{Bmatrix} c_1 \\ c_2 \\ \dots \\ c_n \end{Bmatrix}$ ; $I = \{1, \dots, m\}$, $J = \{1, \dots, n\}$ <br> $\boldsymbol{a}^i \in \mathbb{R}^n = \{A_{i1} \quad \dots \quad A_{in}\}$ <br> $\boldsymbol{A}_j \in \mathbb{R}^m = \begin{Bmatrix} A_{1j} \\ \dots \\ A_{mj} \end{Bmatrix}$ |
| **Linear function** | $f: \mathbb{R}^n \to \mathbb{R}$ <br> $f(\boldsymbol{x}) = a_1x_1 + a_2x_2 + \dots + a_nx_n = \sum_{j=1}^{n} a_jx_j = \boldsymbol{a}^T\boldsymbol{x}, \quad \boldsymbol{a} \in \mathbb{R}^n$ <br> $f(\alpha\boldsymbol{x} + \beta\boldsymbol{y}) = af(\boldsymbol{x}) + \beta f(\boldsymbol{y}), \alpha, \beta \in \mathbb{R}$ |
| **Linear inequality** | $\sum_{j=1}^{n} a_jx_j \leq b \quad , \quad \boldsymbol{a} \in \mathbb{R}^n, b \in \mathbb{R}$ <br> $\boldsymbol{a}^T\boldsymbol{x} \leq b$ |
| **System of linear inequalities** <br> $or \geq$ | $\sum_{j=1}^{n} a_{ij}x_j \leq b_i \quad , \quad i \in I$ <br> $\boldsymbol{a}^i\boldsymbol{x} \leq b_i$ <br> $\boldsymbol{Ax} \leq \boldsymbol{b}$ |
| **LP** (Linear Program) | Minimize **linear objective function** subject to **linear constraints** (linear (in)-equalities) |

### Form

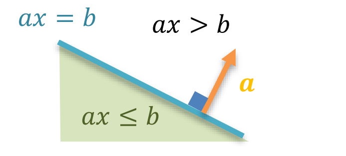
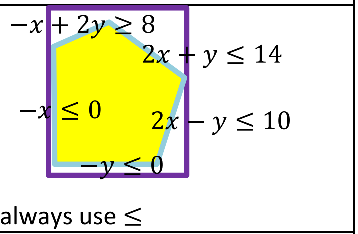
| | LP in General Form | in Canonical Form | | in Standard Form | LP in Inequality Form | |
|---|---|---|---|---|---|---|
| | $max, min\ \boldsymbol{c}^T\boldsymbol{x}$ | $\max \boldsymbol{c}^T\boldsymbol{x}$ | $\min \boldsymbol{c}^T\boldsymbol{x}$ | $max/min\ \boldsymbol{c}^T\boldsymbol{x}$ | $\max \boldsymbol{c}^T\boldsymbol{x}$ | $\min \boldsymbol{c}^T\boldsymbol{x}$ |
| equalities $i \in I$ | $\boldsymbol{a}^i\boldsymbol{x} \leq b_i$ <br> $\boldsymbol{a}^i\boldsymbol{x} = b_i$ <br> $\boldsymbol{a}^i\boldsymbol{x} \geq b_i$ | $\boldsymbol{a}^i\boldsymbol{x} \leq b_i$ | $\boldsymbol{a}^i\boldsymbol{x} \geq b_i$ | $\boldsymbol{a}^i\boldsymbol{x} = b_i$ | $\boldsymbol{a}^i\boldsymbol{x} \leq b_i$ | $\boldsymbol{a}^i\boldsymbol{x} \geq b_i$ |
| variables $j \in J$ | $x_j \geq 0$ <br> $x_j\ free$ <br> $x_j \leq 0$ | $x_j \geq 0$ | $x_j \geq 0$ | $x_j \geq 0$ | | |

### Inequalities transformations

1. replace variables (substitute in equalities, replace in variables)
2. transform equalities

| | | | | |
|---|---|---|---|---|
| equalities $i \in I$ | Inequality to Inequality <br> $\boldsymbol{a}^i\boldsymbol{x} \leq b_i \leftrightarrow -\boldsymbol{a}^i\boldsymbol{x} \geq -b_i$ | Equality to Inequality <br> $\boldsymbol{a}^i\boldsymbol{x} = b_i \leftrightarrow \begin{matrix} \boldsymbol{a}^i\boldsymbol{x} \leq b_i \\ \boldsymbol{a}^i\boldsymbol{x} \geq b_i \end{matrix}$ | Slack (Stillstand) variable <br> $\boldsymbol{a}^i\boldsymbol{x} \leq b_i \to \begin{matrix} \boldsymbol{a}^i\boldsymbol{x} + x_i^s = b_i \\ x_i^s \geq 0 \end{matrix}$ | Surplus (Überschuss) variable <br> $\boldsymbol{a}^i\boldsymbol{x} \geq b_i \to \begin{matrix} \boldsymbol{a}^i\boldsymbol{x} - x_i^s = b_i \\ x_i^s \geq 0 \end{matrix}$ |
| variables $j \in J$ | Nonpositive to nonnegative <br> $x_j \leq 0 \to \begin{matrix} x_j = -\overline{x_j} \\ \overline{x_j} \geq 0 \end{matrix}$ | Free to nonnegative <br> $x_j\ free \to \begin{matrix} x_j = x_j^+ - x_j^- \\ x_j^+, x_j^- \geq 0 \end{matrix}$ | | |

### Geometric aspects

| | | | |
|---|---|---|---|
| Halfspace | $H = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}^T\boldsymbol{x} \leq b\}$ | euclidean space divided by a plane <br> linear: $b = 0$, affine $b = arbitrary$ |   |
| Hyperplane | $H = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}^T\boldsymbol{x} = b\}$ | in 3D, the hyperplane is a 2d plane <br> linear: $b = 0$, affine $b = arbitrary$ | |
| Normal vector | $\boldsymbol{a}$ | | |
| Polyhedron | $P = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{Ax} \leq \boldsymbol{b}\}$ <br> $i = 1 \dots m$ | is the intersection of a finite number of halfspaces. can be unbounded. solution space of a system of linear equalities is also a polyhedron. polyhedron is a convex set. |   |
| Polytope | $P = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{Ax} \leq \boldsymbol{b}, \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u}\}$ | bounded polyhedron | always use $\leq$ |
| Eulerian Walk | | walk through a graph and use every vertex one's | If a graph has an Eulerian walk then the number of odd degree vertices is either 0 or 2. |
| Theorem | A LP with solution space P always has an optimal solution that is a vertex, as far as <br> - P has any vertices ("P is pointed") <br> - the optimum is finite | | |

## Simplex Algorithm

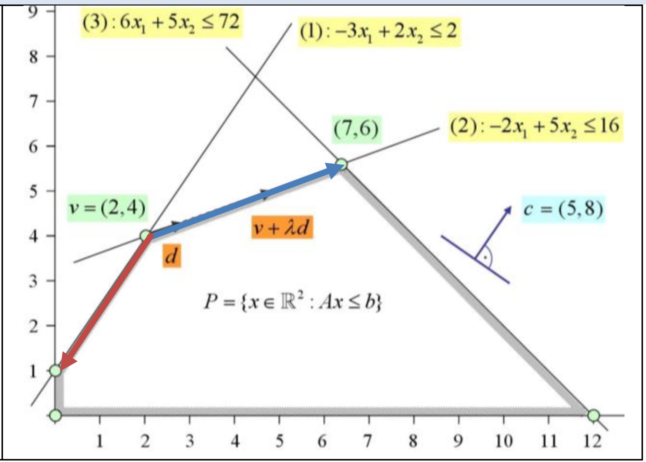
**Given**

| | | | |
|---|---|---|---|
| linear system (convert to $\leq$) | $P = \{x \in \mathbb{R}^2 : Ax \leq b\}$ | $-3x_1 + 2x_2 \leq 2$<br>$-2x_1 + 5x_2 \leq 16$<br>$6x_1 + 5x_2 \leq 72$<br>$-x_1 \leq 0$<br>$-x_2 \leq 0$ | (1)<br>(2)<br>(3)<br>(4)<br>(5) |
| level set | $\max\{c^T x : x \in P\}$ | $\max\{5x_1 + 8x_2\}$ | |

**Prepare**

| | |
|---|---|
| matrix $A$<br>vector $b$<br>level set $c$ | $A = \begin{pmatrix} -3 & 2 \\ -2 & 5 \\ 6 & 5 \\ -1 & 0 \\ 0 & -1 \end{pmatrix}, b = \begin{pmatrix} 2 \\ 16 \\ 72 \\ 0 \\ 0 \end{pmatrix}, c = (5, 8)^T$ |

These algorithm calc the max for min: $\min\{-cx\} = -\max\{cx\}$




**Algorithm**

| | | | |
|---|---|---|---|
| 1. choose **basic selection** | Basic selection $B$<br>Basis $A_B$<br>right hand side $b_B$ | $B \subseteq \{1, \ldots, m\}$ of $\lvert B \rvert = n$<br>$B = \{1,2\}$ | $A_B = \begin{pmatrix} -3 & 2 \\ -2 & 5 \end{pmatrix}$<br>$b_B = \begin{pmatrix} 2 \\ 16 \end{pmatrix}$ |
| 2. calc the inverse | inverse basis $\bar{A}$ | $(5, 2)$<br>$(-2, -3)$ | $\bar{A} = A_B^{-1} = \begin{pmatrix} -\frac{5}{11} & \frac{2}{11} \\ -\frac{2}{11} & \frac{3}{11} \end{pmatrix}$ |
| 3. calc vertex $v$ | basic solution $v$ | $v = \bar{A} b_B$ | $v = \begin{pmatrix} -\frac{5}{11} & \frac{2}{11} \\ -\frac{2}{11} & \frac{3}{11} \end{pmatrix} * \begin{pmatrix} 2 \\ 16 \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \end{pmatrix}$ |
| 4. calc vector $u^T$ | "reduced cost" $u^T$ | $u^T = c^T \bar{A}$ | $u^T = (5 \quad 8) \begin{pmatrix} -\frac{5}{11} & \frac{2}{11} \\ -\frac{2}{11} & \frac{3}{11} \end{pmatrix} = \left(-\frac{41}{11}, \frac{34}{11}\right)$ |
| 5. stop if | $u^T \geq 0^T$<br>in all inequalities | | $u^T \ngeq 0^T$ |
| 6. continue | $u_j < 0$ | choose $j$ so that $u_j < 0$<br>$d = -A_1$ | first element of $u_T$ is negativ<br>first element of $B$ is 1. $\rightarrow j = 1$ $\rightarrow d = \begin{pmatrix} 5/11 \\ 2/11 \end{pmatrix}$ |
| 7. determine $\lambda^*$ | $\lambda \in \mathbb{R}^0$ | $A(v + \lambda d) = Av + \lambda Ad \leq b$<br>$\lambda^* = \min\left\{ \frac{b_i - a^i v}{a^i d} : i \in \{1..m\}, a^i d > 0 \right\}$ | $\begin{pmatrix} -3 & 2 \\ -2 & 5 \\ 6 & 5 \\ -1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 2 \\ 4 \end{pmatrix} + \lambda \begin{pmatrix} -3 & 2 \\ -2 & 5 \\ 6 & 5 \\ -1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 5/11 \\ 2/11 \end{pmatrix} \leq \begin{pmatrix} 2 \\ 16 \\ 72 \\ 0 \\ 0 \end{pmatrix}$<br>$\begin{pmatrix} 2 \\ -16 \\ 32 \\ -2 \\ -4 \end{pmatrix} + \lambda \begin{pmatrix} -1 \\ 0 \\ 40/11 \\ -5/11 \\ -2/11 \end{pmatrix} \leq \begin{pmatrix} 2 \\ 16 \\ 72 \\ 0 \\ 0 \end{pmatrix} \rightarrow \lambda = \begin{pmatrix} 0 \\ \infty \\ 11 \\ -\frac{22}{5} \\ -22 \end{pmatrix}$<br>$k = 3, \quad \lambda^* = 11$ |
| 8. stop if | $\lambda^* = \infty$ | $\rightarrow Ad \leq 0$ | |
| 9. new basic selection | $B'$ | $B' = B - \{j\} \cup \{k\}$ | $B' = \{1,2\} - \{1\} \cup \{3\} = \{2,3\}$ |
| 10. calc vertex $v'$ | $v'$ | $v' = v + \lambda^* d$ | $v' = \begin{pmatrix} 2 \\ 4 \end{pmatrix} + 11 \begin{pmatrix} 5/11 \\ 2/11 \end{pmatrix} = \begin{pmatrix} 7 \\ 6 \end{pmatrix}$ |
| 11. goto step 1 | | | |

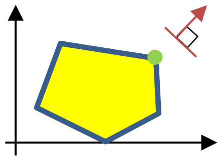
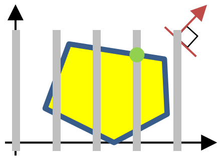
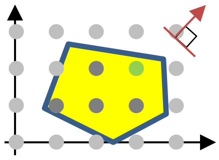
visualise:

| $i$ | $\{B\}$ | $b_B$ | $A_B$ | $A_B^{-1}$ | $v$ | $u$ | $j$ | $d$ | $\lambda^*$ | $\{j\}$ | $v'$ |
|---|---|---|---|---|---|---|---|---|---|---|---|

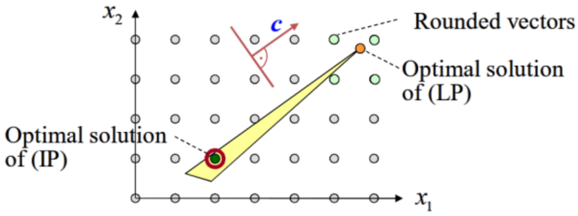
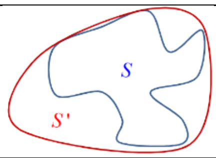
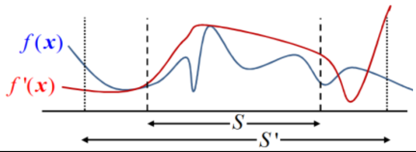
inverse a 2x2 matrix

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

## LP Overview

| Linear Program (LP) | Mixed Integer LP (MIP) | Integer LP (ILP or IP) |
|---|---|---|
| $\max\{\boldsymbol{c}^T\boldsymbol{x} : \boldsymbol{x} \in P\}$<br>$P = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}\}$ | $\max\{\boldsymbol{c}^T\boldsymbol{x} : \boldsymbol{x} \in P \cap \mathbb{Z}_K^n\}$<br>$P = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}\}$<br>$K \subseteq \{1 \ldots n\}$<br>$\mathbb{Z}_K^n = \{\boldsymbol{x} \in \mathbb{R}^n : x_j \in \mathbb{Z}\ for\ j \in K\}$ | $\max\{\boldsymbol{c}^T\boldsymbol{x} : \boldsymbol{x} \in P \cap \mathbb{Z}^n\}$<br>$P = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}\}$ |
|  |  |  |
| optimal solution on vertex | some variables are integer | all variables are integer |
| Simplex | | Commercial: Gurobi, CPLEX<br>Non-Commercial: GLPK, LPSOLVE, SCIP, …<br>Algebraic Model Lang: GAMS, AMPL, LPL, OPL, AIMMS |

## 6+7. Integer Linear Programming

<table>
<tr><td>Naive idea<br>not practicable</td><td>1. Solve problem with LP<br>2. round up/down to get integer solution<br><br>problems:<br>solution may not be a feasible solution<br>solution may be "far away" from optimal solution</td><td>
</td></tr>
<tr><td rowspan="2">Relaxations</td><td>Enlarge solution space<br>$S \rightarrow S'\ with\ S \subseteq S'$<br>$\max\{f(\boldsymbol{x}) : \boldsymbol{x} \in S'\} \geq \max\{f(\boldsymbol{x}) : \boldsymbol{x} \in S\}$<br>e.g. by removing constraints</td><td>
</td></tr>
<tr><td>Increase objective function<br>$f(\boldsymbol{x}) \rightarrow f'(\boldsymbol{x})\ with\ f'(\boldsymbol{x}) \geq f(\boldsymbol{x})\ for\ \boldsymbol{x} \in S$<br>$\max\{f'(\boldsymbol{x}) : \boldsymbol{x} \in S'\} \geq \max\{f(\boldsymbol{x}) : \boldsymbol{x} \in S\}$</td><td>
</td></tr>
</table>

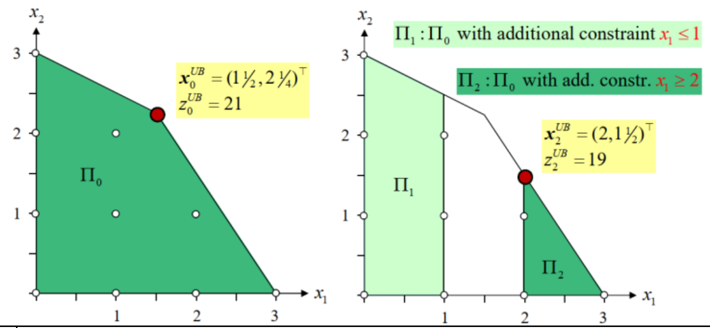
## Branch-and-Bound (B&B) Method (=Divide and Conquer)

**Branching**

If LP solution is all integer:
STOP -> Current node optimal solved

Otherwise: Choose some fractional variable and round up/down

$$\max\left\{\sum_{j\in J} c_j x_j : \sum_{j\in J} a_j x_j \le b, x_j \in \{0,1\}\right\}$$




**Given:**

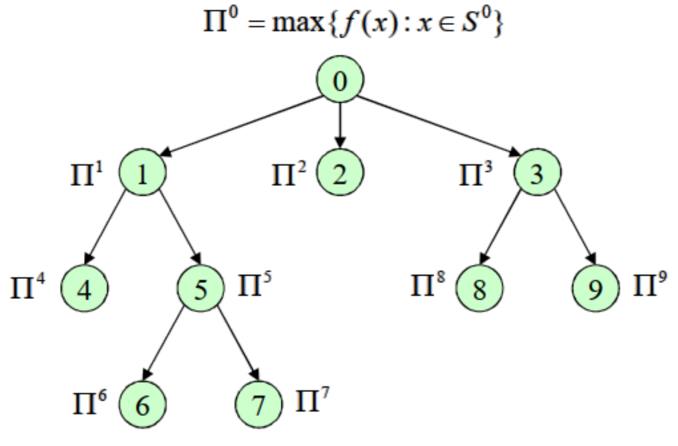
| Item $j \in J$ | A | B | C | D |
|---|---|---|---|---|
| Value $c_j$ | 10 | 12 | 28 | 21 |
| Volume $a_j$ | 7 | 4 | 8 | 9 |

$b = 18$

**Prepare:**

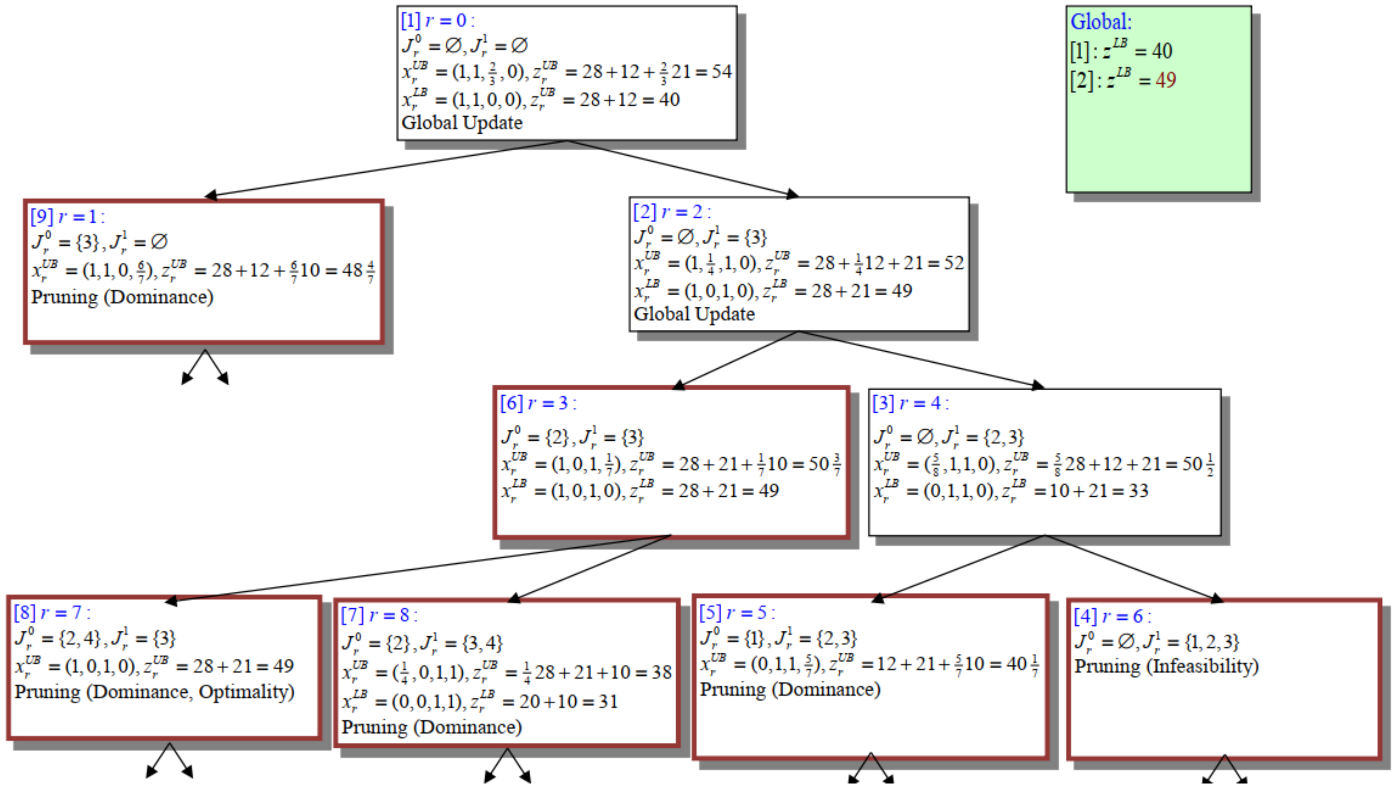
1. Calc benefit per volume: $c_j / a_j$
2. Sort in decreasing order

| Item $j \in J$ | C | B | D | A |
|---|---|---|---|---|
| new | 1 | 2 | 3 | 4 |
| Value $c_j$ | 28 | 12 | 21 | 10 |
| Volume $a_j$ | 8 | 4 | 9 | 7 |
| Benefit per Volume | $3\frac{1}{2}$ | 3 | $2\frac{1}{3}$ | $1\frac{3}{7}$ |

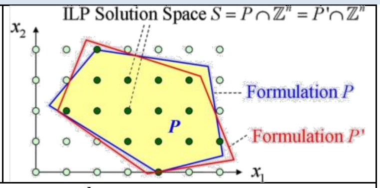
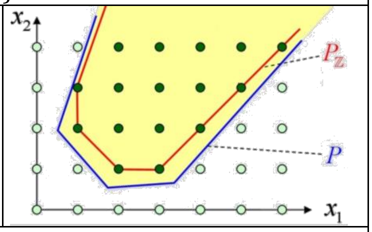
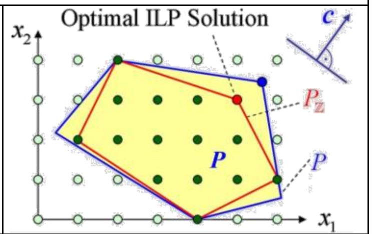
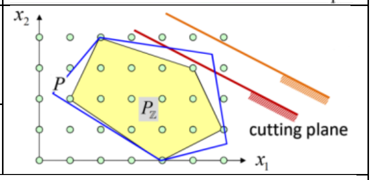
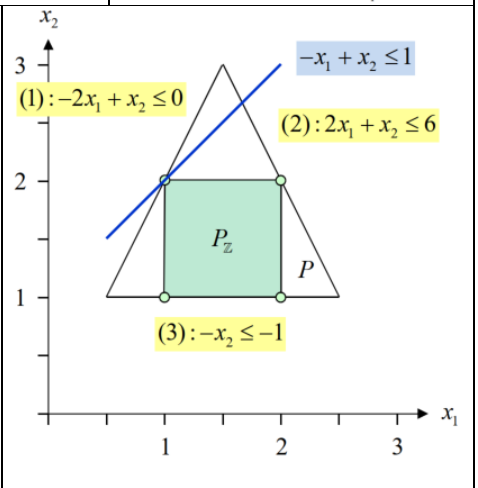



**Algorithm**

| | | | |
|---|---|---|---|
| **Root Node** | no fixed values in root node | $J_r^0 = \emptyset, J_r^1 = \emptyset$ | |
| **Per Node** | | | |
| check | if $sum\ of\ volume\ J_r^1 > b$ stop | "pruning (infeasible)" | |
| calc upper bound | add items from left to right<br>last item fractionally | $x^{UB} = \left(1,1,\frac{2}{3},0,0\right)^T$ | $z^{UB} = 28 + 12 + \frac{2}{3}21 = 54$ |
| check | if $current_{UB} < global_{LB}$ -> stop | "pruning (dominance)" | |
| calc lower bound | round down fractional item | $x^{LB} = (1,1,0,0,0)^T$ | $z^{LB} = 28 + 12 = 40$ |
| check | if $current_{LB} > global_{LB}$ -> add | "global update" | $[1]: z^{LB} = 40$ |
| check | if $current_{LB} = current_{UB}$ -> stop | "pruning (optimal)" | |
| branch (split by fractional item) | {3} | Leaf Node 1: $J_r^0 = \{3\}, J_r^1 = \emptyset$<br>Leaf Node 2: $J_r^0 = \emptyset, J_r^1 = \{3\}$ | |



## Cutting Planes

| | | |
|---|---|---|
| **Definition** | Let $S = P \cap \mathbb{Z}^n$ be the solution space for ILP $\Pi$.<br>A polyhedron $P' \subseteq \mathbb{R}^n$ is called an (ILP-)formulation for $\Pi$ if $P' \cap \mathbb{Z}^n = S$<br>$P'$ is called a better formulation for $\Pi$ if $P' \subseteq P$. |  |
| Convex hull | $$conv(S) = \left\{ \boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{x} = \sum_{i=1}^{k} \lambda_i x^i \ with\ \boldsymbol{x}^i \in S, \lambda_i \geq 0, i = 1 \dots k, \sum_{i=1}^{k} \lambda_i = 1 \right\}$$ $$S \subseteq conv(S)$$ is the smallest convex set containing S.<br>$$\max\{\boldsymbol{c}^T \boldsymbol{x} : \boldsymbol{x} \in S\} = \max\{\boldsymbol{c}^T \boldsymbol{x} : \boldsymbol{x} \in conv(S)\}$$ | |
| Integer hull | $$P_{\mathbb{Z}} = conv(P \cap \mathbb{Z}^n)$$ If $P \subseteq \mathbb{R}^n$ is a rational polyhedron **then $P_{\mathbb{Z}}$ is** a rational polyhedron and **the best ILP formulation**.<br>All vertices of $P_{\mathbb{Z}}$ are interger. |  |
| Rational polyhedron | $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ for some $A \in \mathbb{Q}^{m \times n}, b \in \mathbb{Q}^m$ | |
| Theorem | **Each ILP corresponds to some LP**<br>Let $P \subseteq \mathbb{R}^n$ be a rational polyhedron.<br>Suppose ILP $\max\{\boldsymbol{c}^T \boldsymbol{x} : \boldsymbol{x} \in P \cap \mathbb{Z}^n\}$ has a finite optimum.<br>Then $\max\{\boldsymbol{c}^T \boldsymbol{x} : \boldsymbol{x} \in P \cap \mathbb{Z}^n\} = \max\{\boldsymbol{c}^T \boldsymbol{x} : \boldsymbol{x} \in P_{\mathbb{Z}}\}$ |  |
| Valid inequality | Let $P \subseteq \mathbb{R}^n$ be a polyhedron. An inequality $\boldsymbol{a}^T \boldsymbol{x} \leq \beta$ is a **valid inequality** for P if $\boldsymbol{a}^T \boldsymbol{x} \leq \beta$ is valid for all $\boldsymbol{x} \in P$.<br>-> It does not cut any point inside P. |  |
| Cutting plane | A cutting plane for a polyhedron P is a valid inequality for $P_{\mathbb{Z}}$.<br>-> It does not cut any point inside $P_{\mathbb{Z}}$. | |
| Example | Given: 3 Inequalities<br>$(1): -2x_1 + x_2 \leq 0$<br>$(2): 2x_1 + x_2 \leq 6$<br>$(3): -x_2 \leq -1$<br>$\boldsymbol{A} = \begin{pmatrix} -2 & 1 \\ 2 & 1 \\ 0 & -1 \end{pmatrix}, \boldsymbol{b} = \begin{pmatrix} 0 \\ 6 \\ -1 \end{pmatrix}$<br>Combine inequalities: $\frac{3}{4} * (1) + \frac{1}{4} * (2)$<br>$$\frac{3}{4} * -2x_1 + \frac{3}{4} 1x_2 + \frac{1}{4} * 2x_1 + \frac{1}{4} * 1x_2 \leq \frac{3}{4} * 0 + \frac{1}{4} * 6$$ $$-x_1 + x_2 \leq \frac{3}{2} (\rightarrow can\ be\ rounded\ down)$$ resulting new inequality (Gomory-Chvatal-Cut)<br>$-x_1 + x_2 \leq 1$<br>$\boldsymbol{u} = \left(\frac{3}{4}, \frac{1}{4}, 0\right)^T \geq \boldsymbol{0}$<br>$\boldsymbol{\alpha} := \boldsymbol{u}^T \boldsymbol{A} = (-1,1)^T$<br>$\beta := \boldsymbol{u}^T \boldsymbol{b} = \frac{3}{2}$<br>G-C-Cut<br>$\boldsymbol{\alpha}^T \boldsymbol{x} \leq \lfloor \boldsymbol{\beta} \rfloor$ |  |

## 8+9. Nonlinear Continuous Optimization

**Minimization Problem**

Given the (continuous) function

$$f: \mathbb{R}^n \to \mathbb{R}$$
$$x \to f(x)$$

Find a point $x^*$ where f attains its minimum.

Remarks

Looking for the maximum of a function f is equivalent to finding the minimum of $-f$.
The optimization is called one-dimensional if $n = 1$ and multidimensional fi $n \geq 2$

**Gradient**

$$\nabla f(\vec{x}) = \begin{pmatrix} \frac{\delta f(\vec{x})}{\delta x_1} \\ \frac{\delta f(\vec{x})}{\delta x_2} \\ \dots \\ \frac{\delta f(\vec{x})}{\delta x_n} \end{pmatrix}$$

The gradient of a function f: $\mathbb{R}^n \to \mathbb{R}$ is the vector consisting of the n partial derivatives. At each point $\vec{x}$, the gradient $\nabla f(\vec{x})$ points in the direction of steepest ascent. Its norm $|\nabla f(\vec{x})|$ gives the slope.

$$f(x, y) = x^2 + y^2$$
$$\nabla f(x, y) = \begin{pmatrix} 2x \\ 2y \end{pmatrix}$$

stationary point

$$\nabla f(\vec{x_0}) = \vec{0} = \begin{pmatrix} 0 \\ \dots \\ 0 \end{pmatrix}$$

If the point $\vec{x_0}$ is an extremal point, the gradient must vanish (=0).
not every stationary point is an extremal point -> saddle point.

**Alg 1: Gradient descent**

Linear convergence -> Slow

is an algorithm to find a local minimum of an (unconstrained multidimensional) function $f$

```
start at random point x_0
iterate x_i → x_{i+1}
   determine gradient
   move by some amount β in opposite direction
repeat until gradient is approximately zero
```

$$x^{i+1} = x^i - \beta \nabla f(x^i)$$

Step size $\beta$

1: **Successive halving of the step size** (set $\beta = 1$)
- if worse -> half until better
- if better -> doubling while better

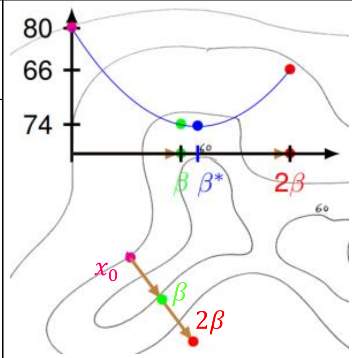
2: **Successive halving of the step size with subsequent parabola fitting** (choosing $x_{i+1}$ according to the minimum of the parabola).
Compute $P(t) = at^2 + bt + c$ such that

| | |
|---|---|
| $P(0) = f(x^i)$ | $= c$ |
| $P(\beta) = f\left(x^i - \beta \nabla f(x^i)\right)$ | $= a\beta^2 + b\beta + c$ |
| $P(2\beta) = f\left(x^i - 2\beta \nabla f(x^i)\right)$ | $= 4a\beta^2 + 2b\beta + c$ |

$P(t)$ attains its minimum in the intervall $[0, 2\beta]$ at

| | |
|---|---|
| $\beta^* = \frac{\beta}{2} * \frac{3 * P(0) - 4 * P(\beta) + P(2\beta)}{P(0) - 2 * P(\beta) + P(2\beta)}$ | $\beta^* = -\frac{b}{2a}$ |

choose better of $\beta$ or $\beta^*$




**Alg 2: Newton's Method** (finds zeros of a function)

Quadratic converg. -> Fast

Tangent t at $\left(x^i, f(x^i)\right)$ has slope $f'(x^i)$

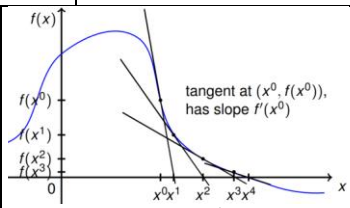
$$t(x) = f'(x^i)(x - x^i) + f(x^i) = 0$$
$$\to x = x^i - \frac{f(x^i)}{f'(x^i)}$$

apply to derivative $f'(x)$ approximates to zeros of $f'(x)$ -> extrem points

$$x^{i+1} = x^i - \frac{f'(x^i)}{f''(x^i)}$$

for multidimensional -> use Hessian-matrix

$$x^{i+1} = x^i - \left(H_f(x^i)\right)^{-1} \nabla f(x^i)$$




$$H_f(x^i) = \begin{pmatrix} \frac{\delta^2 f(x^i)}{\delta x_1 \delta x_1} & \dots & \frac{\delta^2 f(x^i)}{\delta x_1 \delta x_n} \\ \dots & \ddots & \dots \\ \frac{\delta^2 f(x^i)}{\delta x_n \delta x_1} & \dots & \frac{\delta^2 f(x^i)}{\delta x_n \delta x_n} \end{pmatrix}$$

Speed of convergence

| | | |
|---|---|---|
| Linear convergence | $c \in (0,1)$ and $i_0 \in \mathbb{N}$ such that for all $i \geq i_0$ | $\lvert x^* - x^{i+1} \rvert \leq c \lvert x^* - x^i \rvert$ |
| Superlinear convergence | sequence $\{c_i\}_{i \in \mathbb{N}}$ with $\lim_{n \to \infty} c_i = 0$ | $\lvert x^* - x^{i+1} \rvert \leq c_i \lvert x^* - x^i \rvert$ |
| Quadratic convergence | $c > 0$ and $i_0 \in \mathbb{N}$ sucht that for all $i \geq i_0$ | $\lvert x^* - x^{i+1} \rvert \leq c \lvert x^* - x^i \rvert^2$ |

Approximating partial derivates

Computing the partial derivates of f exactly may be impossible or computationally too expensive.

| | |
|---|---|
| first partial derivatives | $\frac{\delta f(x)}{\delta x_i} = \frac{f(x_1 \dots x_{i+\epsilon} \dots x_n) - f(x_1 \dots x_{i-\epsilon} \dots x_n)}{2\epsilon}$ |
| second partial derivates | $\frac{\delta^2 f(x)}{\delta x_i^2} = \frac{f(x_1 \dots x_{i+\epsilon} \dots x_n) - 2f(x_1 \dots x_n) + f(x_1 \dots x_{i-\epsilon} \dots x_n)}{\epsilon^2}$ |

**Alg 3: Broyden's Method**

Idea: Computing and inverting the Hessian matrix $H_f(x_i)$ exactly in the Newton's method is computationally expensive. The idea in quasi-Newton is to approximate the inverse of the Hessian $\left(H_f(x^i)\right)^{-1}$, by some matrix $(A^i)^{-1}$ that can be computed more efficiently.

$$A^i(x^i - x^{i-1}) = \nabla f(x^i) - \nabla f(x^{i-1})$$

$$A^i = A^{i-1} + \frac{\Bigg(\overbrace{\left(\nabla f(x^i) - \nabla f(x^{i-1})\right)}^{g^i} - A^{i-1}\overbrace{(x^i - x^{i-1})}^{d^i}\Bigg)\overbrace{(x^i - x^{i-1})}^{d^i}{}^T}{\Big|\underbrace{x^i - x^{i-1}}_{d^i}\Big|^2}$$

$$A^i = \frac{(g^i - A^{i-1}d^i){d^i}^T}{|d^i|^2}$$

They key insight of Broyden's method is that we do not need to invert $A^i$ explicitly in each step

$$x^{i+1} = x^i - (A^i)^{-1}\nabla f(x^i)$$

Instead we can compute $(A^i)^{-1}$ by updating $(A^{i-1})^{-1}$ according to the so-called **Sherman-Morrison** formula.

**computation**

1. Start with $x^0$.
   a) Compute $\nabla f(x^0)$ and set $(A^0)^{-1} := \left(H_f(x^0)\right)^{-1}$.
   b) Compute $x^1 = x^0 - (A^0)^{-1}\nabla f(x^0)$
2. Iteration step:
   a) Compute $\nabla f(x^i)$, $g^i)\nabla f(x^i) - \nabla f(x^{i-1})$ and $d^i = x^i - x^{i-1}$
   b) Compute $(A^i)^{-1}$ with Sherman-Morrison
   c) Set $x^{i+1} = x^i - (A^i)^{-1}\nabla f(x^i)$

**Sherman-Morrison**
$O(n^2)$ instead $O(n^3)$

Let A be a regular matrix, u and v two vectors. We can compute $(A^i)^{-1}$ directly from $(A^{i-1})^{-1}$

$$(A^i)^{-1} = (A^{i-1})^{-1} - \frac{\left((A^{i-1})^{-1}g^i - d^i\right)(d^i)^T(A^{i-1})^{-1}}{(d^i)^T(A^{i-1})^{-1}g^i}$$

**Alg 4: Aitken's acceleration method**

Aitken's method is not a new method, but can be used to improve the convergence speed of other existing methods.

Example with Pi

$$\pi = \sum_{k=0}^{\infty}\frac{4}{2k+1}(-1)^k = 3.14159265$$

| | | $i=0$ | $i=1$ | $i=2$ | $i=3$ | $i=4$ |
|---|---|---|---|---|---|---|
| normal formular | $x^i = \sum_{k=0}^{i}\frac{4}{2k+1}(-1)^k$ | 4 | 2.6667 | 3.4667 | 2.8952 | 3.3397 |
| Aitken | $y^i = x^i - \frac{(x^i - x^{i-1})^2}{x^i - 2x^{i-1} + x^{i-2}}$ | – | – | 3.1667 | 3.1333 | 3.1452 |

## 10+11. Graphs and Networks

see document 'Combinatorial Problems'
see document 'Graph Theory'

## 12-14. Heuristics

see document 'Combinatorial Problems'