Applied Numerical Optimization

Prof. Alexander Mitsos, Ph.D.

Constrained Optimization: Example

Aachener Verfahrenstechnik

RWTH AACHEN UNIVERSITY

Constrained Optimization

General formulation:

$$\min_{\boldsymbol{x} \in R^n} f(\boldsymbol{x})$$

$$\text{s.t.} \quad c_i(\boldsymbol{x}) = 0, i \in E$$

$$c_i(\boldsymbol{x}) \leq 0, i \in I$$

$\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T \in D$ a vector (point in *n*-dimensional space)

D host set, here $D = R^n$ for explicit treatment of constraints

$f : D \to R$ objective function

$c_i : D \to R$ constraint functions $\forall\, i \in E \cup I$

E the index set of equality constraints

I the index sets of inequality constraints

The constraints and the host set define the feasible set, i.e., the set of all feasible solutions:

$$\Omega = \{\boldsymbol{x} \in R^n \mid c_i(\boldsymbol{x}) \leq 0\ \forall\, i \in I, c_i(\boldsymbol{x}) = 0\ \forall\, i \in E\}$$

Equivalent formulation: $\boxed{\min_{x \in \Omega} f(\boldsymbol{x})}$

Recap on Optimality Conditions of Unconstrained Problems

Optimality conditions of unconstrained problems:

Theorem (Second-order necessary optimality conditions):

Let f be *twice continuously differentiable* and let $\boldsymbol{x}^* \in R^n$ be a *local minimizer* of f, then

1. $\boldsymbol{\nabla} f(\boldsymbol{x}^*) = \boldsymbol{0}$,
2. $\boldsymbol{\nabla}^2 f(\boldsymbol{x}^*)$ is positive semi-definite.

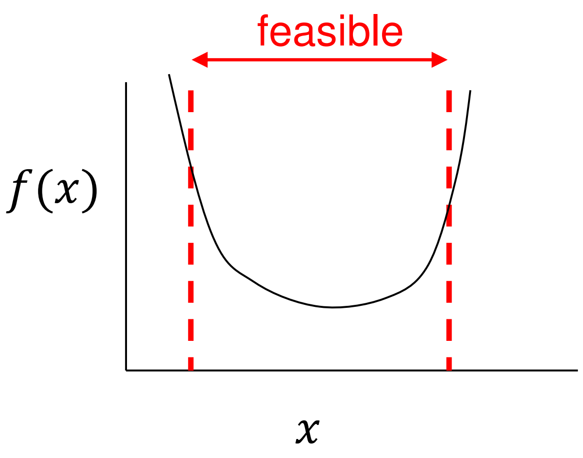
Theorem (Second-order sufficient optimality conditions):

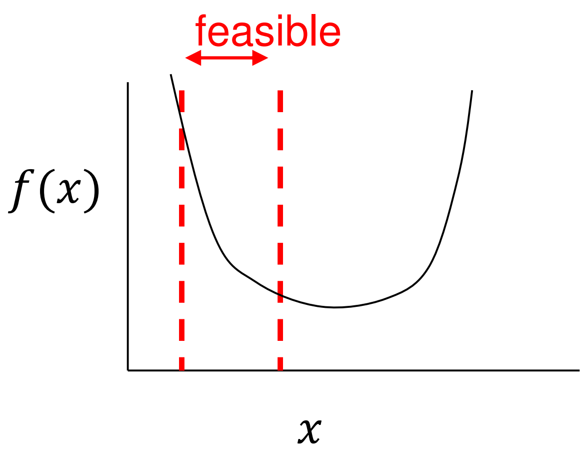
Let f be *twice continuously differentiable* and let $\boldsymbol{x}^* \in R^n$, if

1. $\boldsymbol{\nabla} f(\boldsymbol{x}^*) = \boldsymbol{0}$,
2. $\boldsymbol{\nabla}^2 f(\boldsymbol{x}^*)$ is positive definite.

then $\boldsymbol{x}^*$ is a strict local minimizer of f.

Unconstrained and Constrained Local Optima





Constraints not active

No modifications of optimality conditions

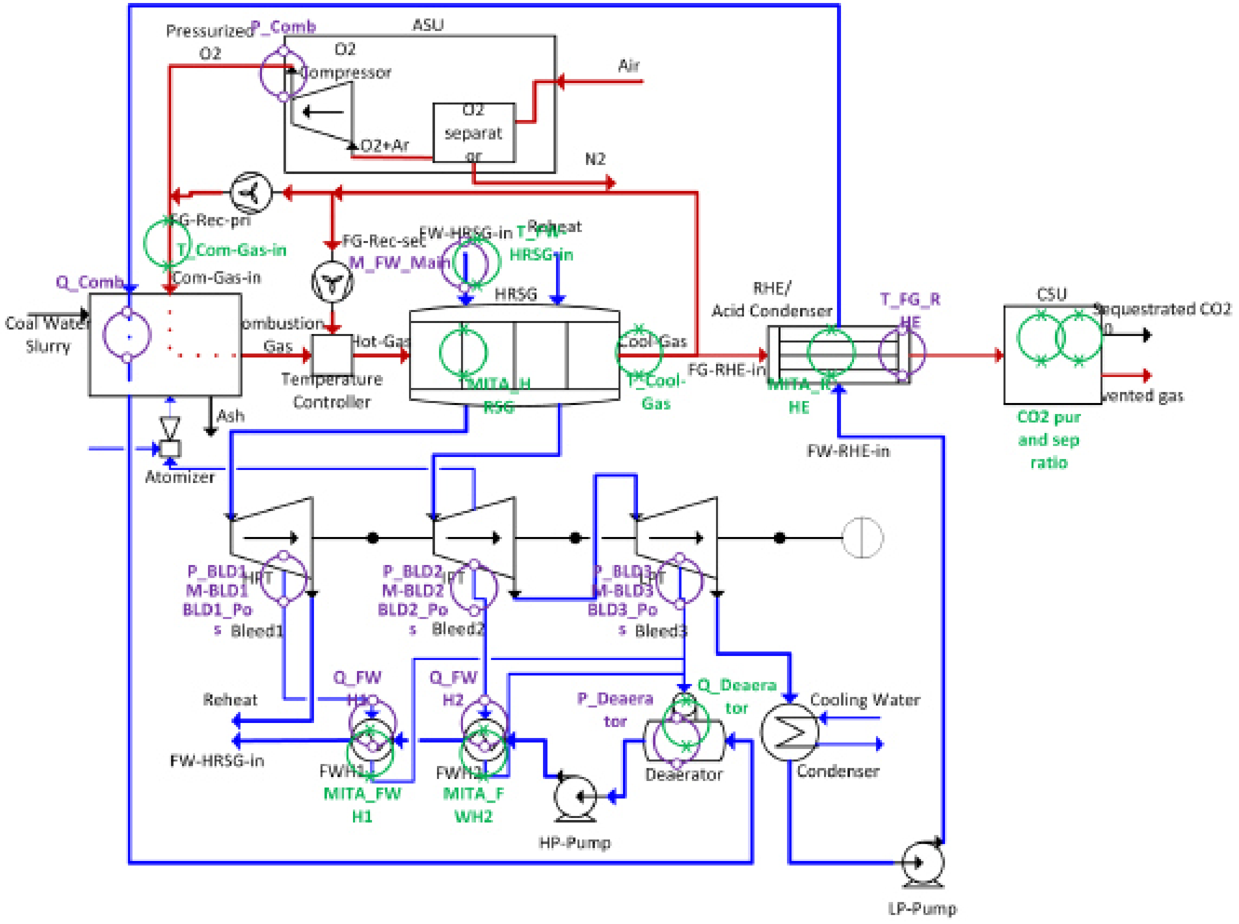
Which conditions have to apply here?

Example: Pressurized OxyCombustion – Basics

- Carbon capture: continue burning fossil fuels but capture the CO_2, i.e., do not emit to atmosphere
  - Pre-, post- oxy-combustion, fuel cells, …
  - Oxycombustion: combust with oxygen instead of air → combustion outlet is CO_2 and H_2O
- Why coal + carbon capture?
  - Coal cheap but dirty
  - Reserves will last for decades
  - Man-made global climate change serious threat now
- Problems: complicated systems, reduced efficiency, increased cost → **optimization**
- Research project moving to physical demonstration
  - Pressurized operation to increase efficiency
  - 300 MWe process (ENEL/ITEA)
  - Detailed model accounting for major losses

Example: Pressurized OxyCombustion – Problem Formulation

- Maximize thermal efficiency for a fixed fuel input
- 16 optimization variables (marked purple) including integer-valued variables
- Inequality constraints: design and economical considerations (marked green)
  - e.g., minimum temperature difference in heat exchangers, maximum temperature in combustor, etc.
- Model equations as equality constraints
  - e.g., mass balance, energy balance, reaction extent, turbine efficiency, heat transfer calculations, etc.
- Proved that some inequalities are active at minimum, eliminated some variable/constraint pairs

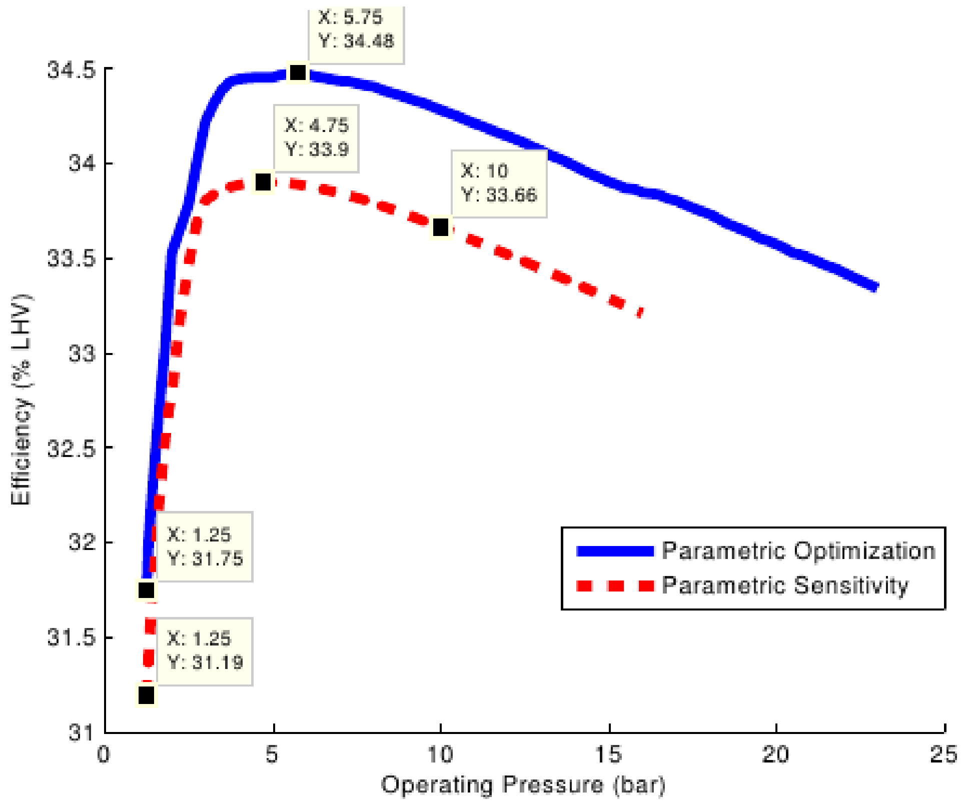


Aachener Verfahrenstechnik

RWTH AACHEN UNIVERSITY

Example: Pressurized OxyCombustion – Results

- Solution with built-in optimization tools in Aspen after extensive massaging, in particular elimination of variable constraints pairs by design specs
  - Not always a good idea but in this case only way to make it work
- Optimal pressure significantly lower than prior single-variable optimization
- Parametric optimization performed to identify trade-off between efficiency and operating pressure
- Optimization teaches about process → can re-solve without optimization
- Optimization gives qualitative and quantitative information and can be used to guide R&D (multi-million €)



Check Yourself

- Why are different optimality conditions needed if the optimization problem has constraints?

Applied Numerical Optimization

Prof. Alexander Mitsos, Ph.D.

Constrained optimization: optimality conditions for equalities

Aachener Verfahrenstechnik

RWTH AACHEN UNIVERSITY

Constrained Optimization: Only Equalities

General formulation:

$$\min_{\boldsymbol{x} \in R^n} f(\boldsymbol{x})$$

$$\text{s.t.} \quad c_i(\boldsymbol{x}) = 0, i \in E$$

$\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T \in D = R^n$ a vector (point in *n*-dimensional space)

D host set here $D = R^n$ (no inequalities, explicit treatment of equalities)

$f : D \to R$ objective function

$c_i : D \to R$ constraint functions $\forall\, i \in E$

E the index set of equality constraints

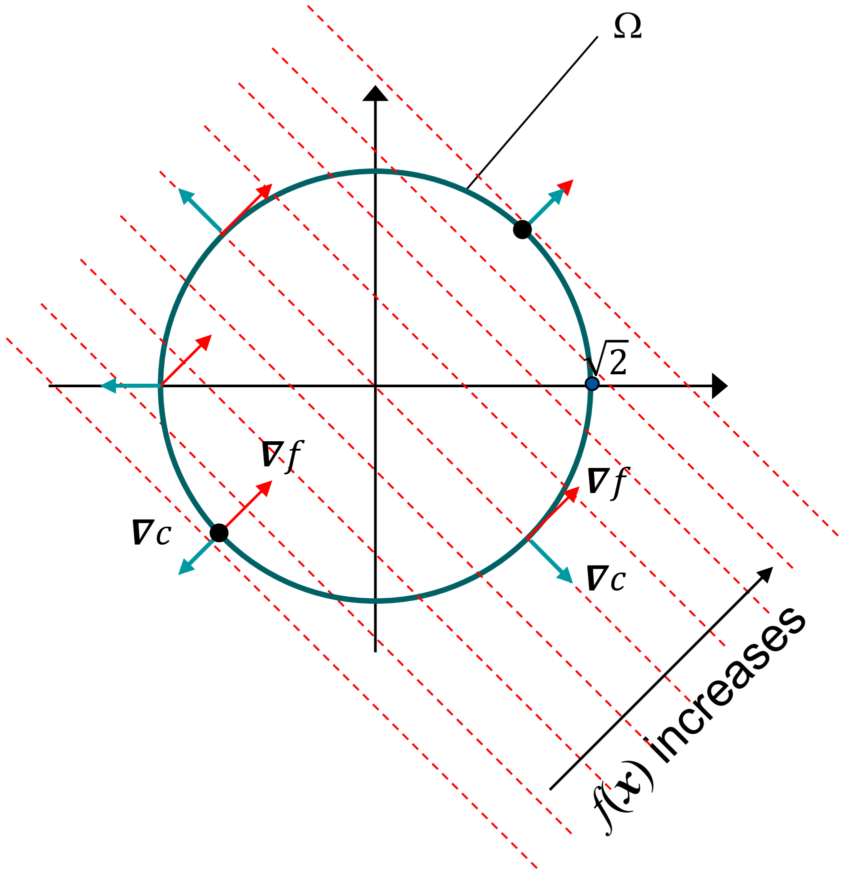
The constraints and the host set define the feasible set, i.e., the set of all feasible solutions:

$$\Omega = \{\boldsymbol{x} \in R^n \mid c_i(\boldsymbol{x}) = 0\ \forall\, i \in E\}$$

Single Equality Constraint: Example

Example:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & x_1 + x_2 && \leftarrow f(\boldsymbol{x}) \\ \text{s.t.} \quad & x_1^2 + x_2^2 - 2 = 0 && \leftarrow c(\boldsymbol{x}) \end{aligned}$$



Solution :

$$\boldsymbol{x}^* = (-1, -1)^T$$

$$\boldsymbol{\nabla} f(\boldsymbol{x}^*) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\boldsymbol{\nabla} c(\boldsymbol{x}^*) = \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix} = \begin{bmatrix} -2 \\ -2 \end{bmatrix}$$

Which conditions are fulfilled at the minimum ?

Single Equality Constraint: Derivation of Optimality Condition

- A current feasible point $\boldsymbol{x}$ is not optimal, if a step $\boldsymbol{p}$ can be found, such that
  - feasibility is retained
  - f can be decreased
- A decrease in f is achieved only along a descent direction $\boldsymbol{p}$ such that $\nabla f(\boldsymbol{x})^T \boldsymbol{p} < 0$
- The equality constraint has to retain feasibility, i.e.,

$$c(\boldsymbol{x}+\boldsymbol{p}) = c(\boldsymbol{x}) + \nabla c(\boldsymbol{x})^T \boldsymbol{p} + \cdots = 0$$

$$\Rightarrow \nabla c(\boldsymbol{x})^T \boldsymbol{p} = 0$$

- At the optimum there should exist no $\boldsymbol{p}$ such that $\nabla f(\boldsymbol{x})^T \boldsymbol{p} < 0$ and $\nabla c(\boldsymbol{x})^T \boldsymbol{p} = 0$

$$\Rightarrow \nabla c(\boldsymbol{x}) || \nabla f(\boldsymbol{x})$$

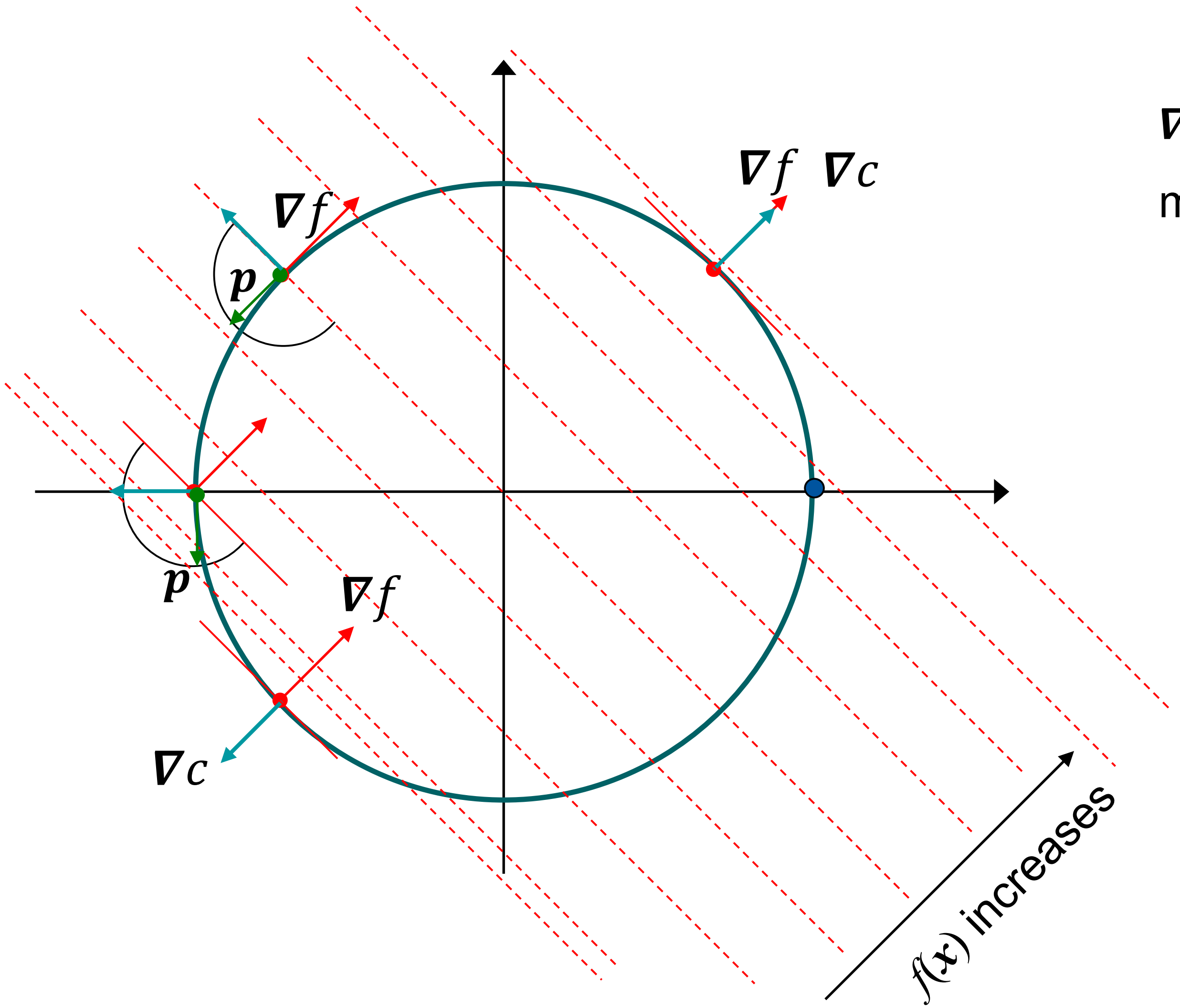
$$\boxed{\Rightarrow \nabla f(\boldsymbol{x}) + \lambda \nabla c(\boldsymbol{x}) = 0}$$

Single Equality Constraint: Illustration of Optimality Condition $\nabla f(x) + \lambda \nabla c(x) = 0$

$\nabla c\,(x)^T p = 0$

$\nabla f(x)^T p < 0$

not minimum



$\nabla f(x) + \lambda \nabla c(x) = 0$

maximum

$\nabla f(x) + \lambda \nabla c(x) = 0$

minimum

(Single) Equality Constraint: General Optimality Conditions

- Define the Lagrangian function: $L(\boldsymbol{x}, \lambda) = f(\boldsymbol{x}) + \lambda c(\boldsymbol{x})$
- Stationary points of L (more precisely saddle point) correspond to stationary points of the constrained problem

$$\min_{\boldsymbol{x}} f(\boldsymbol{x})$$
$$\text{s.t.} \quad c(\boldsymbol{x}) = 0$$

- First-order necessary optimality conditions:

$$\boldsymbol{\nabla} L(\boldsymbol{x}, \lambda) = \begin{bmatrix} \boldsymbol{\nabla}_{\boldsymbol{x}} L(\boldsymbol{x}, \lambda) \\ \nabla_\lambda L(\boldsymbol{x}, \lambda) \end{bmatrix} = \boldsymbol{0}$$

$$\begin{cases} \boldsymbol{\nabla}_{\boldsymbol{x}} L(\boldsymbol{x}, \lambda) = \boldsymbol{\nabla} f(\boldsymbol{x}) + \lambda \boldsymbol{\nabla} c(\boldsymbol{x}) = \boldsymbol{0} & \text{(stationarity)} \\ \nabla_\lambda L(\boldsymbol{x}, \lambda) = c(\boldsymbol{x}) = 0 & \text{(primal feasibility)} \end{cases}$$

- Multiple constraints: Lagrangian multiplier λ_i for each constraint c_i, sum over i: $\boldsymbol{\nabla} f(\boldsymbol{x}) + \sum_i \lambda_i \boldsymbol{\nabla} c_i(\boldsymbol{x}) = \boldsymbol{0}$ and enforce primal feasibility $\forall i$: $c_i(\boldsymbol{x}) = 0$

Single Equality Constraint: Application of Optimality Conditions to Example

Example:

$$\min_{\boldsymbol{x}} \quad x_1 + x_2 \qquad \leftarrow f(\boldsymbol{x})$$

$$\text{s.t.} \quad x_1^2 + x_2^2 - 2 = 0 \qquad \leftarrow c(\boldsymbol{x})$$

Solution :

Lagrangian function: $L(\boldsymbol{x}, \lambda) = f(\boldsymbol{x}) + \lambda c(\boldsymbol{x}) = (x_1 + x_2) + \lambda(x_1^2 + x_2^2 - 2)$

First-Order Necessary Optimality Conditions:

$$\begin{matrix} \boldsymbol{\nabla}_{\boldsymbol{x}} L(\boldsymbol{x}, \lambda) = 0 \\ \nabla_\lambda L(\boldsymbol{x}, \lambda) = 0 \end{matrix} \quad \Leftrightarrow \quad \begin{cases} \boldsymbol{\nabla} f(\boldsymbol{x}) + \lambda \boldsymbol{\nabla} c(\boldsymbol{x}) = \boldsymbol{0} \\ \qquad\qquad\qquad c(\boldsymbol{x}) = 0 \end{cases} \quad \Leftrightarrow \quad \begin{cases} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \lambda \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ x_1^2 + x_2^2 = 2 \end{cases}$$

Two solutions $\begin{cases} \text{min: } \lambda^* = 0.5, \ \boldsymbol{x}^* = (-1, -1) \\ \text{max: } \lambda^* = -0.5, \ \boldsymbol{x}^* = (1,1) \end{cases}$

Single Equality Constraint: Sign of Lagrange Multiplier

- In general we cannot conclude if λ is negative or positive for the minimum. Simply said, the sign of c and thus of ∇c and λ is arbitrary

- $$\min_{\boldsymbol{x}} \quad x_1 + x_2 \qquad \leftarrow f(\boldsymbol{x})$$

  $$\text{s.t.} \quad x_1^2 + x_2^2 - 2 = 0 \qquad \leftarrow c(\boldsymbol{x})$$

  is equivalent to

  $$\min_{\boldsymbol{x}} \quad x_1 + x_2 \qquad \leftarrow f(\boldsymbol{x})$$

  $$\text{s.t.} \quad -x_1^2 - x_2^2 + 2 = 0 \qquad \leftarrow c(\boldsymbol{x})$$

  but has opposite signs for c and ∇c and thus λ.

Check Yourself

- Define the Lagrangian function. Why is it introduced?
- Write down the optimality conditions for equality constraints

Applied Numerical Optimization

Prof. Alexander Mitsos, Ph.D.

Constrained optimization: optimality conditions for inequalities

Aachener Verfahrenstechnik
RWTH AACHEN UNIVERSITY

Constrained Optimization: Only Inequalities

General formulation:

$$\min_{\boldsymbol{x} \in R^n} f(\boldsymbol{x})$$

$$\text{s.t.} \quad c_i(\boldsymbol{x}) \leq 0, i \in I$$

$\boldsymbol{x} = [x_1, x_2, \dots, x_n]^T \in D = R^n$ a vector (point in *n*-dimensional space)

D host set here $D = R^n$ (no equalities, explicit treatment of inequalities)

$f : D \rightarrow R$ objective function

$c_i : D \rightarrow R$ constraint functions $\forall\, i \in I$

I the index set of inequality constraints

The constraints and the host set define the feasible set, i.e., the set of all feasible solutions:

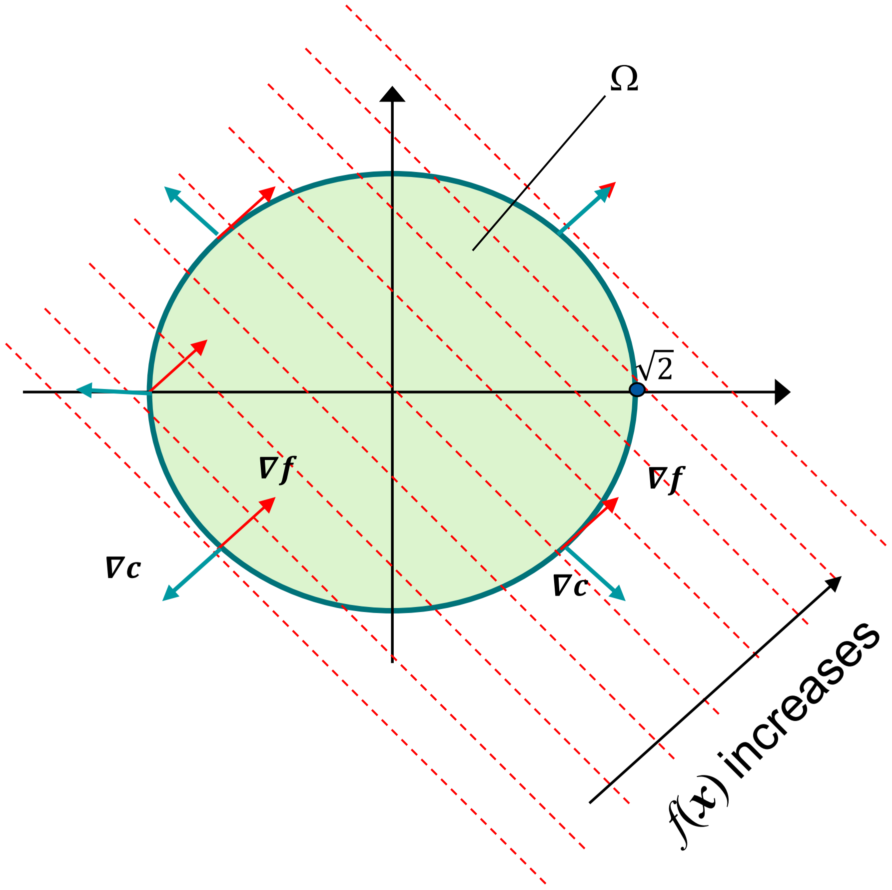
$$\Omega = \{\boldsymbol{x} \in R^n \mid c_i(\boldsymbol{x}) \leq 0 \;\forall\, i \in I\}$$

Single Inequality Constraint: Example

Example:

$$\min_{\boldsymbol{x}} \quad x_1 + x_2 \qquad \leftarrow f(\boldsymbol{x})$$

$$\text{s.t.} \quad x_1^2 + x_2^2 - 2 \leq 0 \qquad \leftarrow c(\boldsymbol{x})$$



Solution :

$$\boldsymbol{x}^* = (-1,-1)^T$$

$$\boldsymbol{\nabla} f(\boldsymbol{x}^*) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\boldsymbol{\nabla} c(\boldsymbol{x}^*) = \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix} = \begin{bmatrix} -2 \\ -2 \end{bmatrix}$$

Which conditions are fulfilled at the minimum ?

Single Inequality Constraint: Derivation of Optimality Conditions (1)

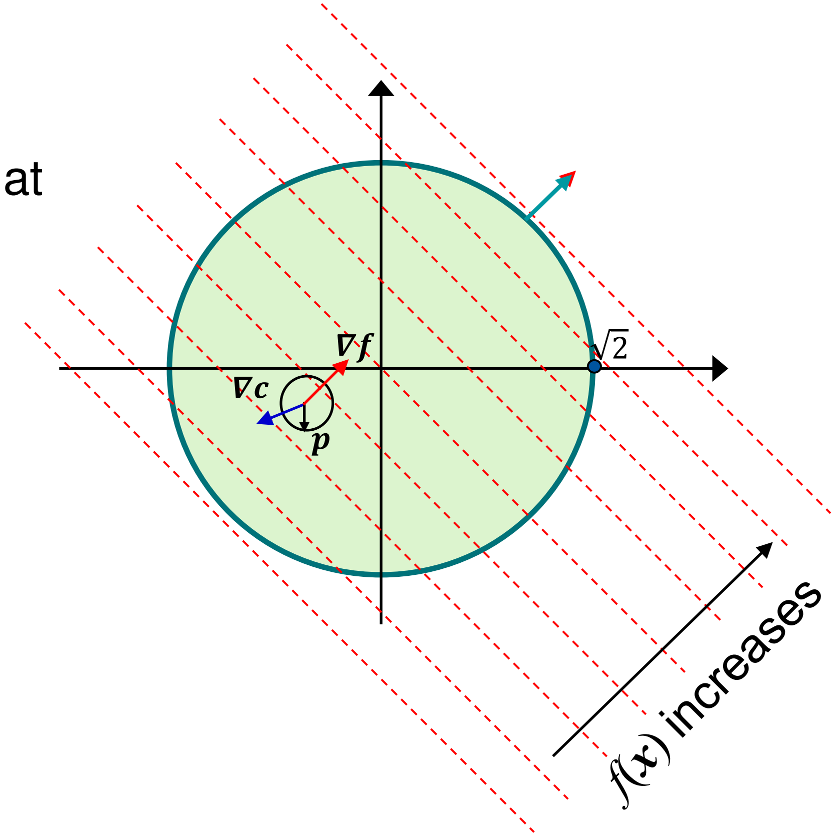
- A current feasible point $\boldsymbol{x}$ is not optimal, if a step $\boldsymbol{p}$ can be found, such that
  - feasibility is retained
  - f can be decreased
- A decrease in f is achieved only along a descent direction $\boldsymbol{p}$ such that

$$\nabla f(\boldsymbol{x})^T \boldsymbol{p} < 0 \qquad (1)$$

- The inequality constraint has to retain feasibility, i.e.,

$$0 \geq c(\boldsymbol{x}+\boldsymbol{p}) \approx c(\boldsymbol{x}) + \nabla c(\boldsymbol{x})^T \boldsymbol{p}$$

$$\Rightarrow c(\boldsymbol{x}) + \nabla c(\boldsymbol{x})^T \boldsymbol{p} \leq 0 \qquad (2)$$



- Two cases:
  1. Point lies in the interior of the disc $c(\boldsymbol{x}) < 0$ (constraint not active).

     $\Rightarrow$ (2) always satisfied for sufficiently small $\|\boldsymbol{p}\|$

     (1) is also satisfied unless $\boxed{\nabla f(\boldsymbol{x}) = \boldsymbol{0}}$

Single Inequality Constraint: Derivation of Optimality Conditions (2)

- Two cases:

  1. Point lies in the interior of the disc $c(\boldsymbol{x}) < 0$ (constraint not active).

     $$\boxed{\nabla f(\boldsymbol{x}) = \boldsymbol{0}}$$

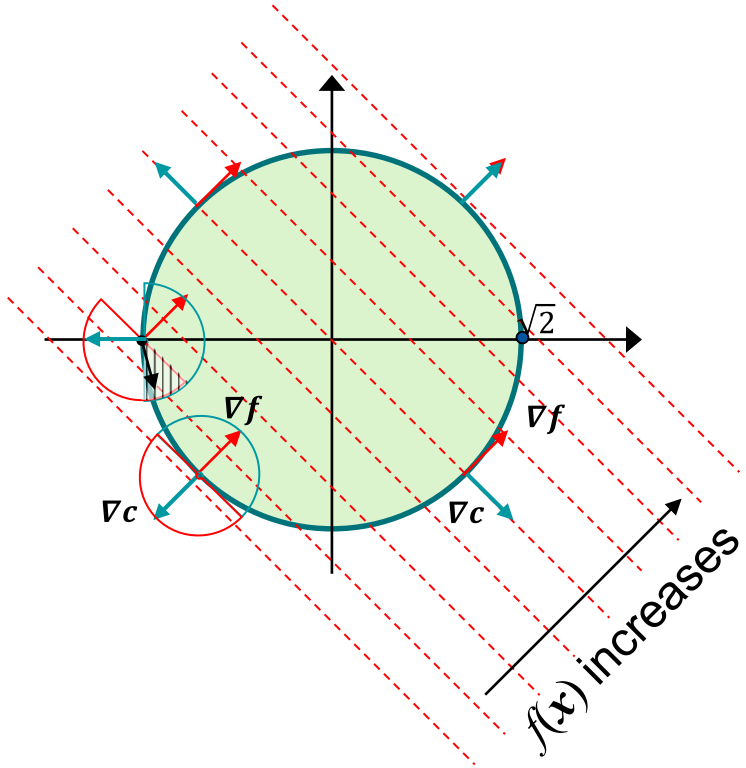
  2. Point lies on the circle $c(\boldsymbol{x}) = 0$ (constraint active).

     Descent direction $\boldsymbol{p}$: $\nabla f(\boldsymbol{x})^T \boldsymbol{p} < 0$ (1)

     Feasibility: $c(\boldsymbol{x}) + \nabla c(\boldsymbol{x})^T \boldsymbol{p} \leq 0$

     $\Rightarrow \nabla c(\boldsymbol{x})^T \boldsymbol{p} \leq 0$ (2)

     $\Rightarrow$ No descent possible if $\nabla f(\boldsymbol{x}) + \lambda \nabla c(\boldsymbol{x}) = 0,\ \lambda \geq 0$



- Both cases can be characterized using the Lagrangian function.

(Single) Inequality Constraint: Optimality Conditions

Example:

$$\min_{\boldsymbol{x}} \quad x_1 + x_2 \qquad \leftarrow f(\boldsymbol{x})$$

$$\text{s.t.} \quad x_1^2 + x_2^2 - 2 \leq 0 \qquad \leftarrow c(\boldsymbol{x})$$

Solution :

Lagrangian function: $L(\boldsymbol{x}, \lambda) = f(\boldsymbol{x}) + \lambda c(\boldsymbol{x}) = (x_1 + x_2) + \lambda(x_1^2 + x_2^2 - 2)$

First-Order Necessary Optimality Conditions:

$$\begin{cases} \boldsymbol{\nabla} f(\boldsymbol{x}) + \lambda \boldsymbol{\nabla} c(\boldsymbol{x}) = \boldsymbol{0} \leftarrow \text{stationarity condition} \\ c(\boldsymbol{x}) \leq 0 \leftarrow \text{primal feasibility} \\ \lambda\, c(\boldsymbol{x}) = 0 \leftarrow \text{complementarity condition} \\ \lambda \geq 0 \leftarrow \text{dual feasibility} \end{cases}$$

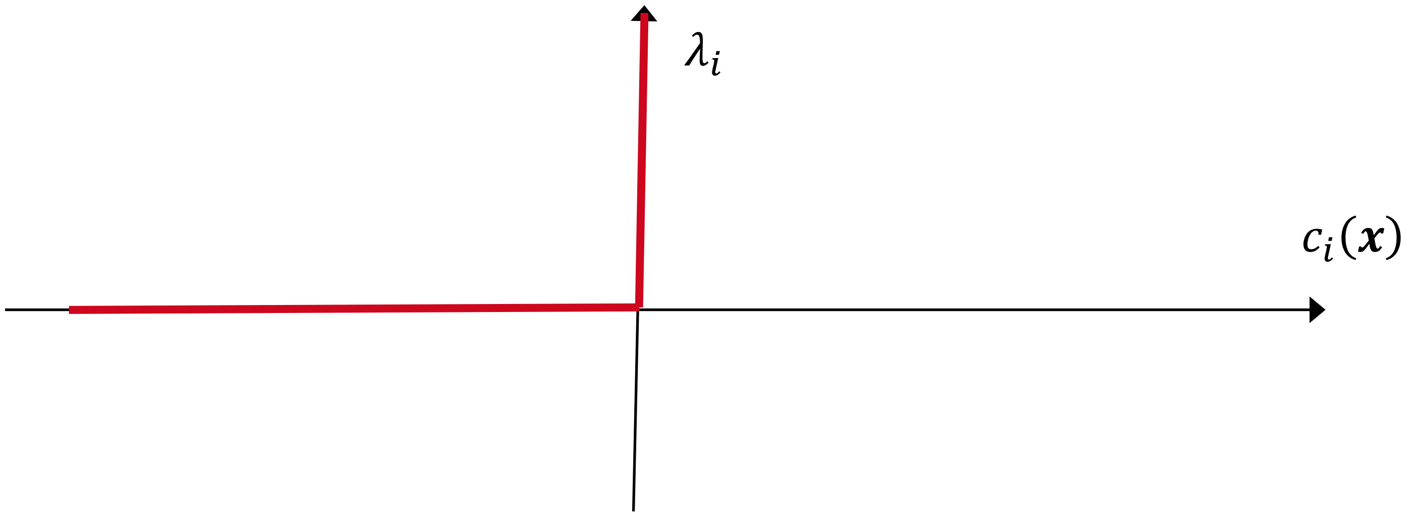
Multiple constraints: Lagrangian multiplier λ_i for each constraint c_i, sum over i: $\boldsymbol{\nabla} f(\boldsymbol{x}) + \sum_i \lambda_i \boldsymbol{\nabla} c_i(\boldsymbol{x}) = \boldsymbol{0}$ and enforce the other conditions $\forall i$: $c_i(\boldsymbol{x}) \leq 0$, $\lambda_i c_i(\boldsymbol{x}) = 0$, $\lambda_i \geq 0$

Complementarity Slackness (CS) of Inequalities

- CS is deceptive simple $\lambda_i \, c_i(\boldsymbol{x}) = 0, i \in I$

- Together with primal and dual feasibility, CS is equivalent to an if-then-else statement
  - constraint inactive: $c_i(\boldsymbol{x}) < 0 \Rightarrow \lambda_i = 0$
  - constraint active : $c_i(\boldsymbol{x}) = 0 \Rightarrow \lambda_i \geq 0$
  - Active set $A(\boldsymbol{x})$: set of active inequalities, i.e. $A(\boldsymbol{x}) = \{i \in I | c_i(\boldsymbol{x}) = 0\}$

- In some sense CS is a non-smooth constraint: feasible set is a corner with no interior!



Check Yourself

- Define the Lagrangian function. Why is it introduced?
- What are complementary conditions? What additional information do they provide on the solution?
- What does “active set” mean?

Applied Numerical Optimization

Prof. Alexander Mitsos, Ph.D.

Constrained optimization: optimality conditions

Constrained Optimization

General formulation:

$$\min_{x \in R^n} f(\boldsymbol{x})$$

$$\text{s.t.} \quad c_i(\boldsymbol{x}) = 0, i \in E$$
$$c_i(\boldsymbol{x}) \leq 0, i \in I$$

$\boldsymbol{x} = [x_1, x_2, \dots, x_n]^T \in D$ a vector (point in *n*-dimensional space)

D host set, here $D = R^n$ for explicit treatment of constraints

$f : D \to R$ objective function

$c_i : D \to R$ constraint functions $\forall\, i \in E \cup I$

E the index set of equality constraints

I the index sets of inequality constraints

The constraints and the host set define the feasible set, i.e., the set of all feasible solutions:

$$\Omega = \{\boldsymbol{x} \in R^n \mid c_i(\boldsymbol{x}) \leq 0 \ \forall\, i \in I, c_i(\boldsymbol{x}) = 0 \ \forall\, i \in E\}$$

Equivalent formulation: $\boxed{\min_{x \in \Omega} f(\boldsymbol{x})}$

First-Order Necessary Optimality Conditions: Definitions

$$\begin{aligned} \min_{x \in R^n} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & c_i(\boldsymbol{x}) = 0,\ i \in E \\ & c_i(\boldsymbol{x}) \leq 0,\ i \in I \end{aligned}$$

Definition active set:

The active set $A(\boldsymbol{x})$: set of the active inequalities and all equalities

$$A(\boldsymbol{x}) = E \cup \{i \in I | c_i(\boldsymbol{x}) = 0\}$$

Lagrangian function:

$$L(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \sum_{i \in E} \lambda_i c_i(\boldsymbol{x}) + \sum_{i \in I} \lambda_i c_i(\boldsymbol{x})$$

$$= f(\boldsymbol{x}) + \boldsymbol{\lambda}^T \boldsymbol{c}(\boldsymbol{x})$$

First-Order Necessary Optimality Conditions: KKT Conditions

Theorem (**K**arush-**K**uhn-**T**ucker (**KKT**)):

Let $\boldsymbol{x}^* \in R^n$ be a local minimizer. Assume $\boldsymbol{\nabla} c_i(\boldsymbol{x}^*)$, $i \in A(\boldsymbol{x}^*)$ are linearly independent. Then, there exist Lagrange multipliers, $\lambda_i^*, i \in E \cup I$:

(1) $\boldsymbol{\nabla} f(\boldsymbol{x}^*) + \sum_{i \in E \cup I} \lambda_i^* \boldsymbol{\nabla} c_i(\boldsymbol{x}^*) = \boldsymbol{0}$ ← stationarity

(2) $c_i(\boldsymbol{x}^*) = 0, \forall\, i \in E$ ← primal feasibility

(3) $c_i(\boldsymbol{x}^*) \leq 0, \forall\, i \in I$ ← primal feasibility

(4) $\lambda_i^* \geq 0, \forall\, i \in I$ ← dual feasibility

(5) $\lambda_i^* c_i(\boldsymbol{x}^*) = 0, \forall\, i \in I$ ← complementary slackness

Remarks:

- From (5) and inactive constraints ($c_i(\boldsymbol{x}^*) < 0$) the Lagrange multipliers are $\lambda_i^* = 0$. Therefore, (1) can be written as $\boldsymbol{0} = \boldsymbol{\nabla}_{\boldsymbol{x}} L(\boldsymbol{x}^*, \lambda^*) = \boldsymbol{\nabla} f(\boldsymbol{x}^*) + \sum_{i \in A(\boldsymbol{x}^*)} \lambda_i^* \boldsymbol{\nabla} c_i(\boldsymbol{x}^*)$
- (5) is trivially satisfied also for equalities since $c_i(\boldsymbol{x}^*) = 0$ by (2)
- (1) implies $\boldsymbol{\nabla} f(\boldsymbol{x}^*)$ is a linear combination of $\boldsymbol{\nabla} c_i(\boldsymbol{x}^*)$, $i \in A(\boldsymbol{x}^*)$. Geometry: $\boldsymbol{\nabla} f(\boldsymbol{x}^*)$ lies in the cone of $\boldsymbol{\nabla} c_i(\boldsymbol{x}^*)$
- Second order sufficient conditions exist.

Optimality Conditions for ≥ Inequalities

$$\min_{\boldsymbol{x}} \quad f(\boldsymbol{x})$$
$$\text{s.t.} \quad c_i(\boldsymbol{x}) = 0,\ i \in E$$
$$\boxed{c_i(\boldsymbol{x}) \leq 0,\ i \in I}$$
$$E = \{1, \dots, n_E\}$$
$$I = \{n_E + 1, \dots, n_E + n_I\}$$

$$\min_{\boldsymbol{x}} \quad f(\boldsymbol{x})$$
$$\text{s.t.} \quad c_i(\boldsymbol{x}) = 0,\ i \in E$$
$$\boxed{c_i(\boldsymbol{x}) \geq 0,\ i \in I}$$
$$E = \{1, \dots, n_E\}$$
$$I = \{n_E + 1, \dots, n_E + n_I\}$$

Lagrangian functions:

$$L(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \sum_{i \in E \cup I} \lambda_i c_i(\boldsymbol{x}) = f(\boldsymbol{x}) + \boldsymbol{\lambda}^T \boldsymbol{c}(\boldsymbol{x})$$

$$L(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) - \sum_{i \in E \cup I} \lambda_i c_i(\boldsymbol{x}) = f(\boldsymbol{x}) - \boldsymbol{\lambda}^T \boldsymbol{c}(\boldsymbol{x})$$

The KKT theorem is valid for both formulations, only one difference:

(3) $c_i(\boldsymbol{x}^*) \leq 0,\ \forall i \in I$

(3) $c_i(\boldsymbol{x}^*) \geq 0,\ \forall i \in I$

Lagrange Multipliers and Sensitivity

- Active constraints: $i \in A(\boldsymbol{x}^*)$

  Stationarity $\mathbf{0} = \boldsymbol{\nabla}_{\boldsymbol{x}} L(\boldsymbol{x}^*, \lambda^*) = \boldsymbol{\nabla} f(\boldsymbol{x}^*) + \sum_{i \in A(\boldsymbol{x}^*)} \lambda_i^* \boldsymbol{\nabla} c_i(\boldsymbol{x}^*)$:

  $f(\boldsymbol{x}) - f(\boldsymbol{x}^*) \approx (\boldsymbol{x} - \boldsymbol{x}^*)^T \boldsymbol{\nabla} f(\boldsymbol{x}^*) = -\sum_{i \in A(\boldsymbol{x}^*)} \lambda_i^* (\boldsymbol{x} - \boldsymbol{x}^*)^T \boldsymbol{\nabla} c_i(\boldsymbol{x}^*) \approx -\sum_{i \in A(\boldsymbol{x}^*)} \lambda_i^* [c_i(\boldsymbol{x}) - c_i(\boldsymbol{x}^*)]$.

  Thus, $\boxed{\delta f = -\sum_{i \in A(\boldsymbol{x}^*)} \lambda_i^* \delta c_i.}$

- Perturbation of active constraint has impact on optimal objective value! (unless by coincidence $\lambda_i^* = 0$)
- Perturbation of inactive constraint has no impact on optimal objective value.
- Sensitivity analysis: how important is each constraint

Check Yourself

- Define the Lagrangian function. Why is it introduced?
- What are complementary conditions? What additional information do they provide on the solution?
- What does “active set” mean?
- Write down the KKT-optimality conditions for a constrained optimization problem.
- Can you think of the problems which could arise if the gradients of the constraints are linearly dependent? (cf. KKT conditions)
- What influence does a perturbation of active constraints have on the solution? And that of inactive constraints?

Applied Numerical Optimization

Prof. Alexander Mitsos, Ph.D.

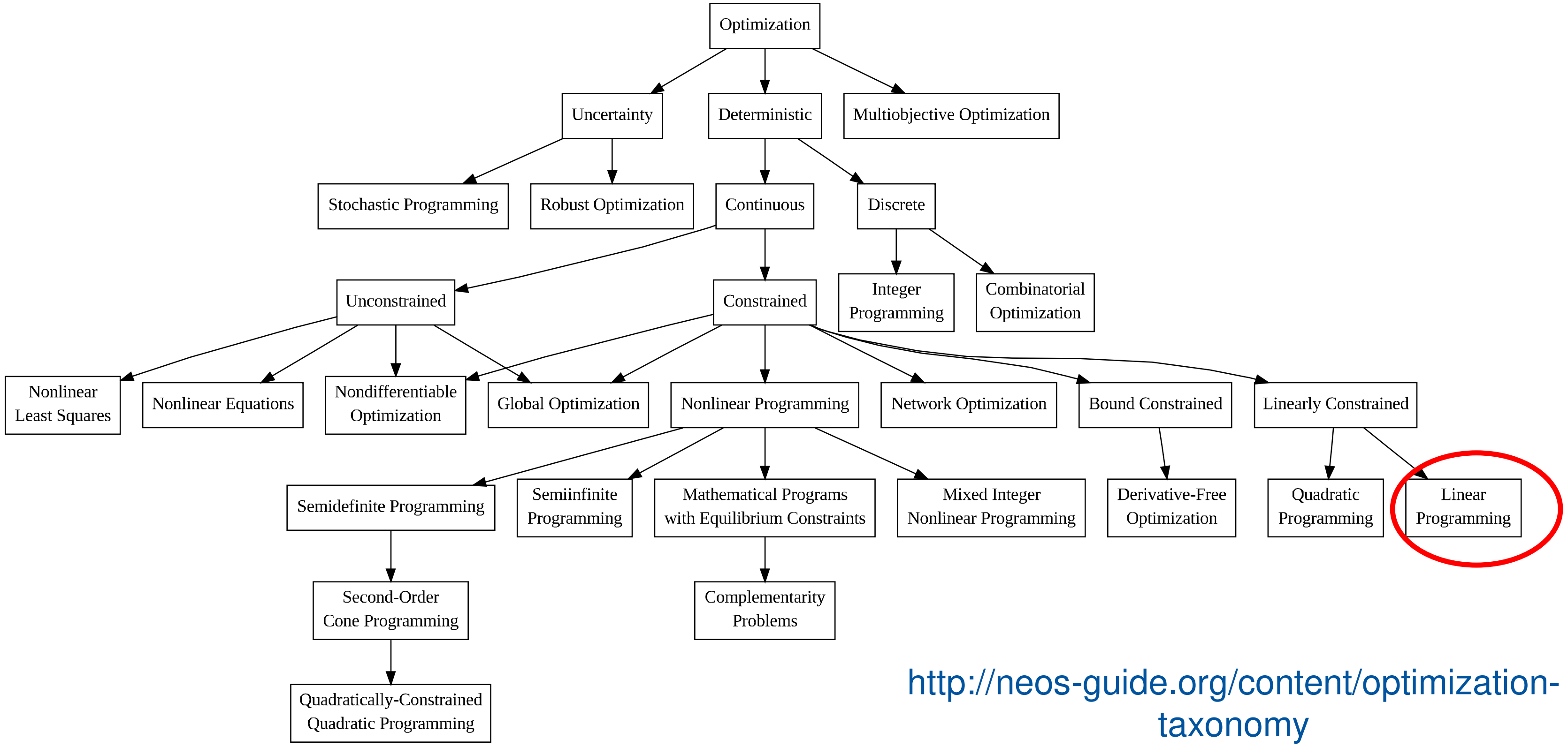
Linear programming (LP): definitions and example

Aachener Verfahrenstechnik
RWTH AACHEN UNIVERSITY

Linear Programming (LP)

- LP was developed in the 1940s and is still widely used
- Algorithms for other classes solve sequence of LPs (e.g., in global optimization)
- LP algorithms provide understanding for NLP
- Good software tools exist for LPs

Special case of the constrained (nonlinear) optimization problems



http://neos-guide.org/content/optimization-taxonomy

Example

- A refinery has two crude oils available as raw materials.
- It produces gasoline, kerosene and fuel oil.
- Profit from processing crude #1 is 1 €/kg and from crude #2 is 0.7 €/kg.
- What are their optimal daily feed rates?

product	yield percentage		maximum allowable product rate (kg/day)
	from Crude #1	from Crude #2	
Gasoline	70	31	6000
Kerosene	6	9	2400
Fuel oil	24	60	12000

Mathematical formulation:

$$\max_{x_1,x_2} 1x_1 + 0.7x_2$$

$$\begin{aligned} \text{s.t. } & 0.70x_1 + 0.31x_2 \leq 6000 \\ & 0.06x_1 + 0.09x_2 \leq 2400 \\ & 0.24x_1 + 0.60x_2 \leq 12000 \\ & x_1, x_2 \geq 0 \end{aligned}$$

$x_i, i \in \{1,2\}$ denotes the feed rate of crude i to the refinery

Edgar T. F., Himmelblau D. M., Optimization of Chemical Processes, 2nd Edition, McGraw-Hill, 2001

Standard Form of LPs

NLP, non-linear program

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \quad \leftarrow \text{Objective function}$$

$$\text{s.t. } c_i(\boldsymbol{x}) = 0, i \in E \quad \leftarrow \text{Equality constraints (ECs)}$$

$$c_i(\boldsymbol{x}) \leq 0, i \in I \quad \leftarrow \text{Inequality constraints (ICs)}$$

LP, linear program

$$\min_{\boldsymbol{x}} \boldsymbol{d}^T \boldsymbol{x} \quad \leftarrow \text{Linear objective function}$$

$$\text{s.t. } \boldsymbol{a}_i^T \boldsymbol{x} - \boldsymbol{b}_i = 0, i \in E \quad \leftarrow \text{Linear ECs}$$

$$-x_i \leq 0, i \in \{1, \dots, n\} \quad \leftarrow \text{Variable bounds}$$

Standard form:

$$\min_{\boldsymbol{x}} \boldsymbol{d}^T \boldsymbol{x}$$
$$\text{s.t. } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$
$$\boldsymbol{x} \geq \boldsymbol{0}$$

Transformation to Standard Form

$$\begin{aligned}\min_{\bar{\boldsymbol{x}}}\ & \bar{\boldsymbol{d}}^T \bar{\boldsymbol{x}} \\ \text{s.t.}\ & \boldsymbol{A}_0 \bar{\boldsymbol{x}} = \boldsymbol{b}_0 \\ & \boldsymbol{A}_1 \bar{\boldsymbol{x}} \geq \boldsymbol{b}_1 \\ & \boldsymbol{A}_2 \bar{\boldsymbol{x}} \leq \boldsymbol{b}_2 \\ & \bar{\boldsymbol{x}} \geq \boldsymbol{0}\end{aligned}$$

?

$$\begin{aligned}\min_{\boldsymbol{x}}\ & \boldsymbol{d}^T \boldsymbol{x} \\ \text{s.t.}\ & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0}\end{aligned}$$

Transformation to standard form:

- Introduce new “slack” variables for each IC

$$\boldsymbol{A}_1 \bar{\boldsymbol{x}} \geq \boldsymbol{b}_1 \Rightarrow \boldsymbol{A}_1 \bar{\boldsymbol{x}} - \boldsymbol{v}_1 = \boldsymbol{b}_1, \boldsymbol{v}_1 \geq \boldsymbol{0}$$

$$\boldsymbol{A}_2 \bar{\boldsymbol{x}} \leq \boldsymbol{b}_2 \Rightarrow \boldsymbol{A}_2 \bar{\boldsymbol{x}} + \boldsymbol{v}_2 = \boldsymbol{b}_2, \boldsymbol{v}_2 \geq \boldsymbol{0}$$

- Complement variables: $\boldsymbol{x} = [\bar{\boldsymbol{x}}^T\ \boldsymbol{v}_1^T\ \boldsymbol{v}_2^T]^T$ and combine equalities

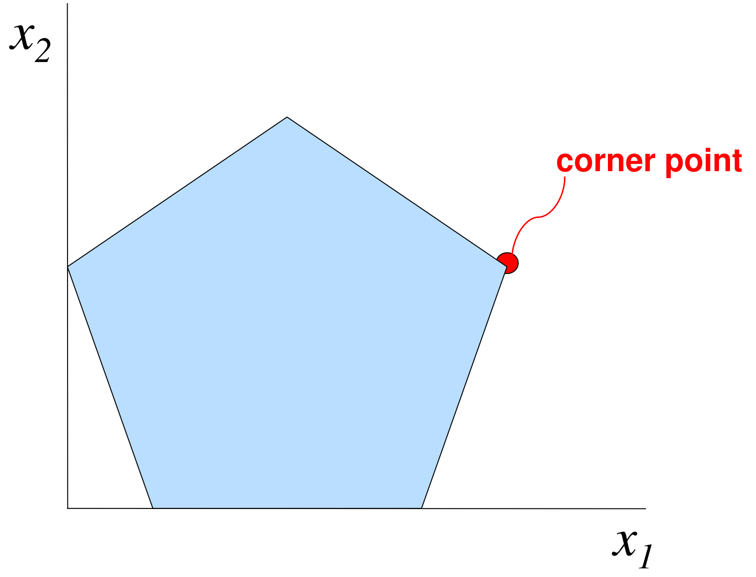
Basic Assumptions

- The LPs are given in standard form: $$\boxed{\begin{aligned} &\min_{\boldsymbol{x}} \boldsymbol{d}^T \boldsymbol{x} \\ &\text{s.t. } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ &\qquad \boldsymbol{x} \geq \boldsymbol{0} \end{aligned}}$$
- The model has more variables than equality constraints, i.e., we have degrees of freedom
- The matrix $\boldsymbol{A} \in R^{m \times n}$ has full row rank, i.e. $\text{rank}(\boldsymbol{A}) = m$
- For graphical illustration we typically take:

$$\boxed{\begin{aligned} &\min_{x_1, x_2} d_1 x_1 + d_2 x_2 \\ &\text{s.t. } \boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b} \\ &\qquad x_1 \geq 0 \\ &\qquad x_2 \geq 0 \end{aligned}}$$

Geometry of Linear Programming Problems

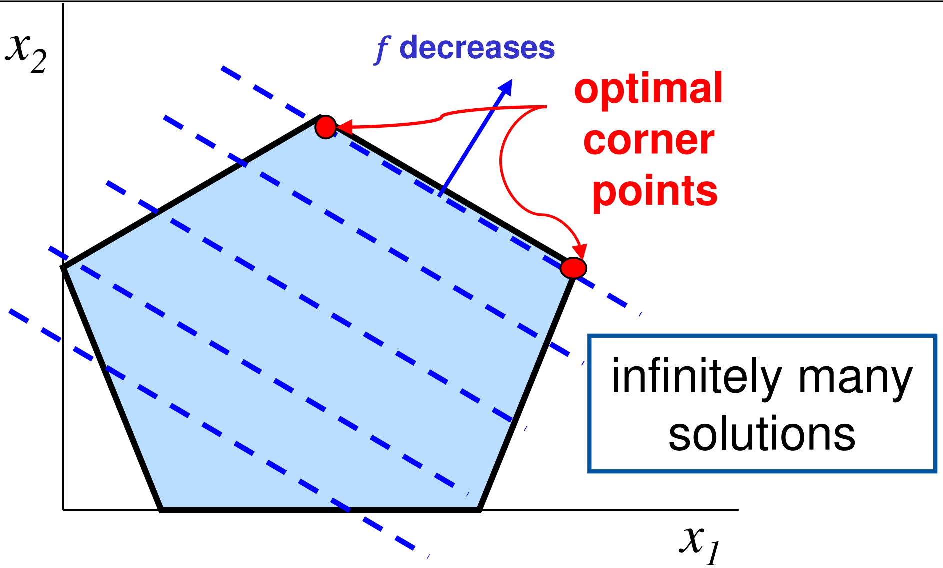
Feasible set : $\Omega = \{\boldsymbol{x} \in R^n \,|\boldsymbol{x} \geq \boldsymbol{0}, \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}\}$

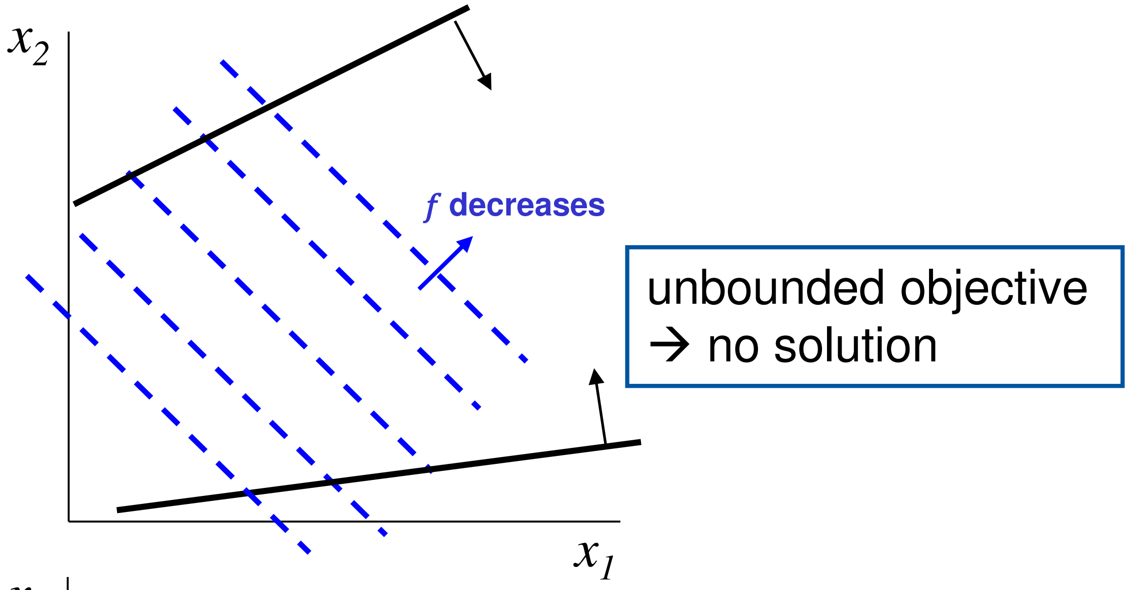


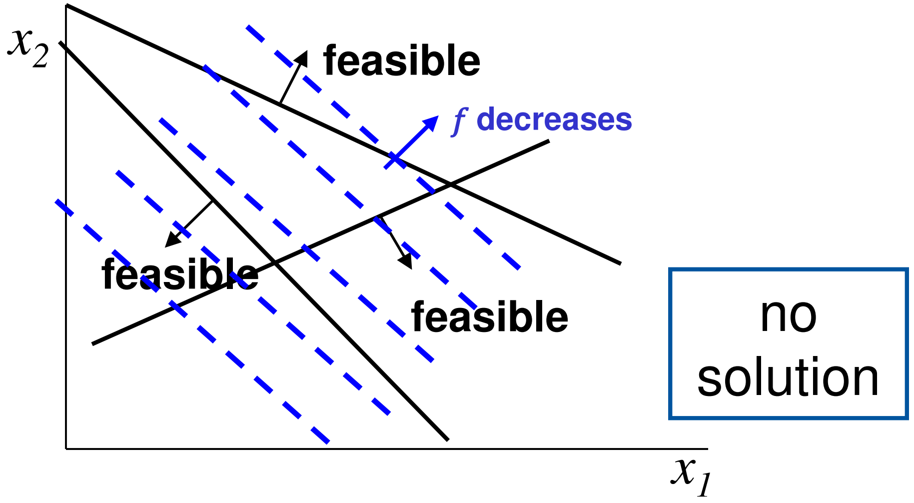
- Ω is a polytope, i.e. a multidimensional polygon.
- Several edges span the faces of the polytope
- A corner point **P** is the intersection of (at least two) active constraints.
- LPs are always convex. So any local solution is a global solution.
- If an optimal solution exists, then at least one corner point of the polytope is optimal

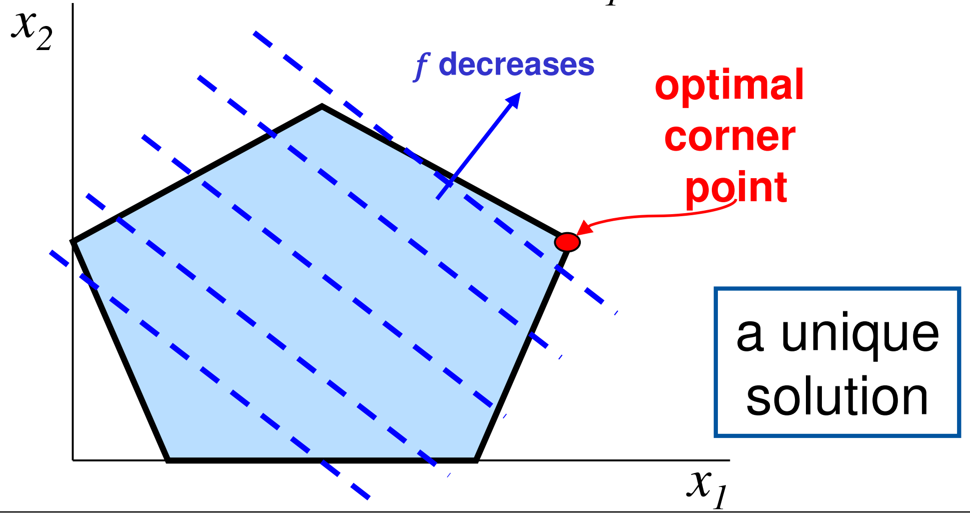
In figure, $\Omega = \{\boldsymbol{x} \in R_+^2 \,|\boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b}\}$

Are LPs Always Solvable?









Check Yourself

- What constitutes Linear Programming?
- Write down the standard form of the LP. How can we transform an arbitrary LP into standard form?
- What are basic assumptions met by the standard form?
- Typically, what represents the feasible set of the LP?

Applied Numerical Optimization

Prof. Alexander Mitsos, Ph.D.

Linear programming: the Simplex method

Definition Basic Feasible Point

We consider the LP in standard form:

$$\min_{\boldsymbol{x}} \boldsymbol{d}^T \boldsymbol{x}$$
$$\text{s.t.}\ \ \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$
$$\boldsymbol{x} \geq \boldsymbol{0}$$

where matrix $\boldsymbol{A}$ is a full rank $(m \times n)$ matrix, $n > m$

Definition: $\boldsymbol{x}$ is a basic feasible point if an index set $|T(\boldsymbol{x})| = m,\ T(\boldsymbol{x}) \subset \{1, \ldots, n\}$ can be chosen:

- $\boldsymbol{B} \coloneqq [\boldsymbol{a}_i]_{i \in T(\boldsymbol{x})}$ is regular basis matrix ($\boldsymbol{a}_i$ is i-th column of $\boldsymbol{A}$)
- $\boldsymbol{x}_B \coloneqq [x_i]_{i \in T(\boldsymbol{x})} \geq \boldsymbol{0}$ and $\boldsymbol{x}_N \coloneqq [x_i]_{i \notin T(\boldsymbol{x})} = \boldsymbol{0}$

Visualization of a Basic Feasible Point

Original, non-square linear system of equations:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \qquad \begin{bmatrix} a_{11} & \cdots & a_{1m} & a_{1,m+1} & \cdots & a_{1n} \\ \cdots & & & \cdots & \cdots & \cdots \\ a_{m1} & \cdots & a_{mm} & a_{m,m+1} & \cdots & a_{mn} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ \cdots \\ x_m \\ x_{m+1} \\ \cdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \cdots \\ b_m \end{bmatrix}$$

Select index vector and reorder columns:

$$\Leftrightarrow [\boldsymbol{B} \quad \boldsymbol{N}] \begin{bmatrix} \boldsymbol{x}_B \\ \boldsymbol{x}_N \end{bmatrix} = \boldsymbol{b} \qquad \begin{array}{cc} \text{basic columns} & \text{nonbasic columns} \\ \left[\begin{array}{ccc} b_{11} & \cdots & b_{1m} \\ \cdots & & \\ b_{m1} & \cdots & b_{mm} \end{array}\right. & \left.\begin{array}{ccc} n_{1,m+1} & \cdots & n_{1n} \\ \cdots & \cdots & \cdots \\ n_{m,m+1} & \cdots & n_{mn} \end{array}\right] \\ \boldsymbol{B} & \boldsymbol{N} \end{array} \cdot \begin{bmatrix} x_{B,1} \\ \cdots \\ x_{B,m} \\ 0 \\ \cdots \\ 0 \end{bmatrix} = \begin{bmatrix} b_1 \\ \cdots \\ b_m \end{bmatrix}$$

Properties of Basic Feasible Points

With $\boldsymbol{N} := [\boldsymbol{a}_i]_{i \notin T(\boldsymbol{x})}$

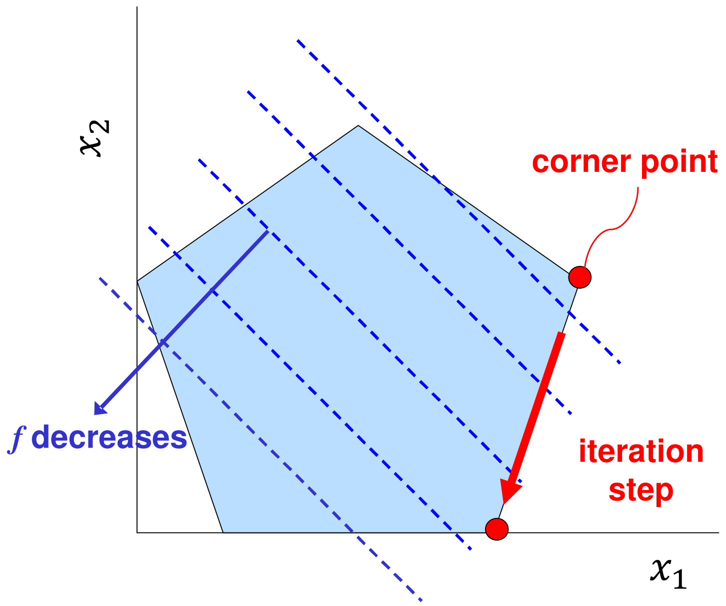
$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \Leftrightarrow \boldsymbol{B}\boldsymbol{x}_B + \boldsymbol{N}\boldsymbol{x}_N = \boldsymbol{b} \;\Rightarrow\; \boldsymbol{B}\boldsymbol{x}_B = \boldsymbol{b} \;\Rightarrow\; \boldsymbol{x}_B = \boldsymbol{B}^{-1}\boldsymbol{b}$$

Propositions:

- The basic feasible points are the corner points.
- If there is a feasible point, then there is a basic feasible point.
- If an optimal solution exist, then at least one basic feasible point is an optimal solution.

Simplex Method for LP – Overview

- Search optimal among the basic feasible points (i.e. the corner points of the polytope).
- Start from feasible corner
- Iterate by moving to neighboring corner point.
- At each move we decrease the objective function
  - In degenerate cases it may stay constant.
- Neighboring corner points of the polytope correspond to basic feasible points with one different index in $T(\boldsymbol{x})$.



- Which conditions must hold at the optimum?
- How to perform the iteration step?
- (How to find the initial feasible basic point?)

Simplex Method – Check of Optimality

- KKT-conditions are sufficient for global solution (convexity)

$$\boldsymbol{A}^T\boldsymbol{\lambda}_E + \boldsymbol{\lambda}_I = \boldsymbol{d}$$
$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$
$$\boldsymbol{x} \geq \boldsymbol{0}$$
$$\boldsymbol{\lambda}_I \geq \boldsymbol{0}$$
$$x_i\lambda_{I,i} = 0, \forall\, i = \{1, \dots, n\}$$

choose $T(\boldsymbol{x})$ $\Rightarrow$

$$[\boldsymbol{B} \quad \boldsymbol{N}]^T\boldsymbol{\lambda}_E + [\boldsymbol{\lambda}_{I,B}^T \quad \boldsymbol{\lambda}_{I,N}^T]^T = [\boldsymbol{d}_B^{\ T} \quad \boldsymbol{d}_N^{\ T}]^T$$
$$\boldsymbol{B}\boldsymbol{x}_B + \boldsymbol{N}\boldsymbol{x}_N = \boldsymbol{b}$$

choose $\boldsymbol{x}_N = \boldsymbol{0}$, and $\boldsymbol{\lambda}_{I,B} = \boldsymbol{0}$ (C.S. satisfied)

$$\Rightarrow \quad \boldsymbol{\lambda}_E = [\boldsymbol{B}^T]^{-1}\boldsymbol{d}_B$$
$$\Rightarrow \quad \boldsymbol{\lambda}_{I,N} = \boldsymbol{d}_N - \boldsymbol{N}^T\boldsymbol{\lambda}_E$$
$$\Rightarrow \quad \boldsymbol{x}_B = \boldsymbol{B}^{-1}\boldsymbol{b}$$

KKT conditions are satisfied if: $\boldsymbol{x}_B \geq \boldsymbol{0}$ and $\boldsymbol{\lambda}_{I,N} \geq \boldsymbol{0}$

Simplex Method - Iteration Sequence

Initialize with basic feasible point $\boldsymbol{x}$

Loop:

1. If $\boldsymbol{\lambda}_{I,N} \geq \boldsymbol{0}$ terminate
2. Choose an index q: $q \notin T^k(\boldsymbol{x})$, $\lambda_{I,q} = \min_{i \notin T^k(\boldsymbol{x})} \lambda_{I,i}$ (note $\lambda_{I,q} < 0$)
3. Initialize x_q^+=0, fix all other components of $\boldsymbol{x}_N^+$ to zero.
4. Increase x_q^+, following $\boldsymbol{A}\boldsymbol{x}^+ = \boldsymbol{b}$ until some x_p^+ with $p \in T(\boldsymbol{x})$ becomes zero.

$$\boldsymbol{A}\boldsymbol{x}^+ = \boldsymbol{B}\boldsymbol{x}_B^+ + \boldsymbol{a}_q x_q^+ = \boldsymbol{b} = \boldsymbol{A}\boldsymbol{x} = \boldsymbol{B}\boldsymbol{x}_B$$

$$\boldsymbol{x}_B^+ = \boldsymbol{x}_B - \boldsymbol{B}^{-1}\boldsymbol{a}_q x_q^+ \geq \boldsymbol{0} \quad \Rightarrow \quad x_p^+ = 0$$

5. Replace the index p with q in $T(\boldsymbol{x})$ and update $\boldsymbol{x} = \boldsymbol{x}^+$
6. Go to 1.

Check Yourself

- Why is the solution of the LP at the corner points of the feasible set?
- What is the main idea behind the Simplex Method for LP?
- Explain geometrically the procedure for updating the index set in the Simplex Method.

Applied Numerical Optimization

Prof. Alexander Mitsos, Ph.D.

Linear Programming (LP): Duality and Optimality Conditions

Aachener Verfahrenstechnik
RWTH AACHEN UNIVERSITY

First-Order Necessary Optimality Conditions (Recap)

$$\min_{\boldsymbol{x} \in R^n} f(\boldsymbol{x})$$

$$\text{s.t.} \quad c_i(\boldsymbol{x}) = 0, i \in E$$

$$c_i(\boldsymbol{x}) \leq 0, i \in I$$

Theorem (**K**arush-**K**uhn-**T**ucker (**KKT**)):

Let $\boldsymbol{x}^* \in R^n$ be a local minimizer. Assume $\boldsymbol{\nabla} c_i(\boldsymbol{x}^*)$, $i \in A(\boldsymbol{x}^*)$ are linearly independent. Then, there exist Lagrange multipliers, $\lambda_i^*, i \in E \cup I$:

1. $\boldsymbol{\nabla} f(\boldsymbol{x}^*) + \sum_{i \in E \cup I} \lambda_i^* \boldsymbol{\nabla} c_i(\boldsymbol{x}^*) = \mathbf{0}$ ← stationarity
2. $c_i(\boldsymbol{x}^*) = 0, \forall\, i \in E$ ← primal feasibility
3. $c_i(\boldsymbol{x}^*) \leq 0, \forall\, i \in I$ ← primal feasibility
4. $\lambda_i^* \geq 0, \forall\, i \in I$ ← dual feasibility
5. $\lambda_i^*\, c_i(\boldsymbol{x}^*) = 0, \forall\, i \in I$ ← complementary slackness

KKT Conditions of Optimality for LPs

- General problem:

$$\min_{x \in R^n} f(\boldsymbol{x})$$
$$\text{s.t.} \quad c_i(\boldsymbol{x}) = 0, i \in E$$
$$c_i(\boldsymbol{x}) \leq 0, i \in I$$

LP (standard form)

$$\min_{x \in R^n} \boldsymbol{d}^T \boldsymbol{x}$$
$$\text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$
$$\boldsymbol{x} \geq \boldsymbol{0}$$

- Lagrange function:

$$L(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \sum_{i \in E} \lambda_i c_i(\boldsymbol{x}) + \sum_{i \in I} \lambda_i c_i(\boldsymbol{x})$$

$$L(\boldsymbol{x}, \boldsymbol{\lambda}_E, \boldsymbol{\lambda}_I) = \boldsymbol{d}^T \boldsymbol{x} + \boldsymbol{\lambda}_E^T (\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}) - \boldsymbol{\lambda}_I^T \boldsymbol{x}$$

- KKT conditions:

$$\boldsymbol{\nabla}_{\boldsymbol{x}} L(\boldsymbol{x}^*, \lambda^*) = \boldsymbol{0}$$
$$c_i(\boldsymbol{x}^*) = 0, \forall\, i \in E$$
$$c_i(\boldsymbol{x}^*) \leq 0, \forall\, i \in I$$
$$\lambda_i^* \geq 0, \forall\, i \in I$$
$$\lambda_i^* c_i(\boldsymbol{x}^*) = 0, \forall\, i \in I$$

$$\Rightarrow$$

$$\boldsymbol{A}^T \boldsymbol{\lambda}_E^* + \boldsymbol{\lambda}_I^* = \boldsymbol{d}$$
$$\boldsymbol{A}\boldsymbol{x}^* = \boldsymbol{b}$$
$$\boldsymbol{x}^* \geq \boldsymbol{0}$$
$$\boldsymbol{\lambda}_I^* \geq \boldsymbol{0}$$
$$\lambda_{I,i}^* x_i^* = 0, \forall i = 1, \dots, n$$

Nonlinear equations!
With the inequalities can be rewritten as single nonlinear equation $\boldsymbol{x}^{*,T} \boldsymbol{\lambda}_I^* = 0$

Duality at the Optimum

- LP (standard form):
$$\min_{\boldsymbol{x}} \boldsymbol{d}^T\boldsymbol{x}$$
$$\text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$
$$\boldsymbol{x} \geq \boldsymbol{0}$$

- Lagrange function: $L(\boldsymbol{x}, \boldsymbol{\lambda}_E, \boldsymbol{\lambda}_I) = \boldsymbol{d}^T\boldsymbol{x} + \boldsymbol{\lambda}_E^T(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}) - \boldsymbol{\lambda}_I^T\boldsymbol{x}$

- KKT conditions:
$$\boldsymbol{A}^T\boldsymbol{\lambda}_E^* + \boldsymbol{\lambda}_I^* = \boldsymbol{d}$$
$$\boldsymbol{A}\boldsymbol{x}^* = \boldsymbol{b}$$
$$\boldsymbol{x}^* \geq \boldsymbol{0}$$
$$\boldsymbol{\lambda}_I^* \geq \boldsymbol{0}$$
$$\lambda_{I,i}^* x_i^* = 0, \forall\, i = \{1, \dots n\}$$

$$\begin{aligned}\boxed{\boldsymbol{d}^T\boldsymbol{x}^*} &= (\boldsymbol{A}^T\boldsymbol{\lambda}_E^* + \boldsymbol{\lambda}_I^*)^T\boldsymbol{x}^* \\ &= \boldsymbol{\lambda}_E^{*T}\boldsymbol{A}\boldsymbol{x}^* + \boldsymbol{\lambda}_I^{*T}\boldsymbol{x}^* \\ &= \boldsymbol{\lambda}_E^{*T}\boldsymbol{b} \\ &= \boxed{\boldsymbol{b}^T\boldsymbol{\lambda}_E^*}\end{aligned}$$

Strong Duality

Dual Linear Program

- The dual LP: $\max\limits_{\boldsymbol{\lambda}_E} \ \boldsymbol{b}^T\boldsymbol{\lambda}_E$ s.t. $\boldsymbol{A}^T\boldsymbol{\lambda}_E \leq \boldsymbol{d}$ $\Leftrightarrow$ $\min\limits_{\boldsymbol{\lambda}_E} \ -\boldsymbol{b}^T\boldsymbol{\lambda}_E$ s.t. $\boldsymbol{d} - \boldsymbol{A}^T\boldsymbol{\lambda}_E \geq \boldsymbol{0}$

- Lagrange function: $\bar{L}(\boldsymbol{\lambda}_E, \boldsymbol{x}) = -\boldsymbol{b}^T\boldsymbol{\lambda}_E + \boldsymbol{x}^T(\boldsymbol{A}^T\boldsymbol{\lambda}_E - \boldsymbol{d})$ Lagrange multipliers

- KKT conditions:

$$\boldsymbol{A}\boldsymbol{x}^* = \boldsymbol{b}$$
$$\boldsymbol{A}^T\boldsymbol{\lambda}_E^* \leq \boldsymbol{d}$$
$$\boldsymbol{x}^* \geq \boldsymbol{0}$$
$$x_i^*\left(\boldsymbol{a}_i^T\boldsymbol{\lambda}_E^* - d_i\right) = 0, \forall i \in \{1, \dots, n\}$$

($\boldsymbol{a}_i$ the i^th^ column of $\boldsymbol{A}$)

Choose: $\boldsymbol{\lambda}_I^*$: $\lambda_{I,i}^* = d_i - \boldsymbol{a}_i^T\boldsymbol{\lambda}_E^*$ $\Longleftrightarrow$

$$\boldsymbol{A}\boldsymbol{x}^* = \boldsymbol{b}$$
$$\boldsymbol{\lambda}_I^* \geq \boldsymbol{0}$$
$$\boldsymbol{x}^* \geq \boldsymbol{0}$$
$$x_i^*\lambda_{I,i}^* = 0, \forall\, i = \{1, \dots n\}$$

- Dual and primal problem share optimality conditions
- The dual of the dual is the primal

Strong Duality

- Primal problem: $\min_{\boldsymbol{x}} \boldsymbol{d}^T\boldsymbol{x}$ s.t. $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, $\boldsymbol{x} \geq \boldsymbol{0}$ Dual Problem: $\max_{\boldsymbol{\lambda}_E} \boldsymbol{b}^T\boldsymbol{\lambda}_E$ s.t. $\boldsymbol{A}^T\boldsymbol{\lambda}_E \leq \boldsymbol{d}$

- Strong duality: optimal objective values of primal and dual are equal $\boldsymbol{d}^T\boldsymbol{x}^* = \boldsymbol{b}^T\boldsymbol{\lambda}_E^*$
  - True for all convex problems.
  - In general weak duality: optimal objective value of primal is ≥ optimal objective value of dual
- Strong duality with primal and dual feasibility are necessary and sufficient optimality conditions for LP
  - Primal feasibility: $\boldsymbol{x}^* \geq \boldsymbol{0}$, dual feasibility $\boldsymbol{A}^T\boldsymbol{\lambda}_E^* \leq \boldsymbol{d}$**,** strong duality $\boldsymbol{d}^T\boldsymbol{x}^* = \boldsymbol{b}^T\boldsymbol{\lambda}_E^*$
  - Alternative to KKT. No Lagrange multipliers of the inequalities, no complementarity slackness

Check Yourself

- Write down the KKT-conditions for LP problems.
- How is the dual LP defined? How are the primal and the dual problems related?
- What is strong duality? How is it useful?

Applied Numerical Optimization

Prof. Alexander Mitsos, Ph.D.

Interior-point methods for linear programs (LP)

Aachener Verfahrenstechnik | RWTH AACHEN UNIVERSITY

Primal-Dual Method (PDM) – Basic Idea

- The following constraints imply KKT conditions of primal and dual problem

$$\begin{cases} (1)\ \boldsymbol{A}^T\boldsymbol{\lambda}_E + \boldsymbol{\lambda}_I = \boldsymbol{d} \\ (2)\ \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ (3)\ x_i\lambda_{I,i} = 0, i = 1,2,\ldots,n \\ (4)\ \boldsymbol{x} \geq \boldsymbol{0}, \boldsymbol{\lambda}_I \geq \boldsymbol{0} \end{cases}$$

- PDM is an interior-point method based on this set of constraints
- PDM finds a solution of the system (1)-(3) by applying a variant of Newton's method
- Equations (1) and (2) are always satisfied as they are linear
- The inequalities (4) are the main source of all complications in interior-point methods (IPM).

Primal-Dual Method: Full Iteration Step for Equations

- We define the function $\boldsymbol{F}(\boldsymbol{x}, \boldsymbol{\lambda}_E, \boldsymbol{\lambda}_I) = \begin{bmatrix} \boldsymbol{A}^T \boldsymbol{\lambda}_E + \boldsymbol{\lambda}_I - \boldsymbol{d} \\ \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \\ \boldsymbol{X}\boldsymbol{\Lambda}_I \boldsymbol{e} \end{bmatrix} = \boldsymbol{0}$

  where $\boldsymbol{X} = \begin{bmatrix} x_1 & & & \\ & x_2 & & \\ & & \ddots & \\ & & & x_n \end{bmatrix}$, $\boldsymbol{\Lambda}_I = \begin{bmatrix} \boldsymbol{\lambda}_{I,1} & & & \\ & \boldsymbol{\lambda}_{I,2} & & \\ & & \ddots & \\ & & & \boldsymbol{\lambda}_{I,n} \end{bmatrix}$ and $\boldsymbol{e} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$

- PDM employs Newton's method to $\boldsymbol{F}$ at the current point to find the search direction $(\delta\boldsymbol{x}, \delta\boldsymbol{\lambda}_{\mathrm{E}}, \delta\boldsymbol{\lambda}_{\mathrm{I}})$

$$\boldsymbol{J}\left(\boldsymbol{x}^{(k)}, \boldsymbol{\lambda}_E^{(k)}, \boldsymbol{\lambda}_I^{(k)}\right) \begin{bmatrix} \delta\boldsymbol{x} \\ \delta\boldsymbol{\lambda}_E \\ \delta\boldsymbol{\lambda}_I \end{bmatrix} = -\boldsymbol{F}\left(\boldsymbol{x}^{(k)}, \boldsymbol{\lambda}_E^{(k)}, \boldsymbol{\lambda}_I^{(k)}\right) \Rightarrow \begin{bmatrix} \boldsymbol{0} & \boldsymbol{A}^T & \boldsymbol{I} \\ \boldsymbol{A} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{\Lambda}_I & \boldsymbol{0} & \boldsymbol{X} \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{x} \\ \delta\boldsymbol{\lambda}_E \\ \delta\boldsymbol{\lambda}_I \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ -\boldsymbol{X}\boldsymbol{\Lambda}_I \boldsymbol{e} \end{bmatrix}$$

  where $\boldsymbol{J}(\boldsymbol{x}^{(k)}, \boldsymbol{\lambda}_E^{(k)}, \boldsymbol{\lambda}_I^{(k)})$ is the Jacobian matrix of $\boldsymbol{F}$.

Aachener Verfahrenstechnik

RWTH AACHEN UNIVERSITY

Primal-Dual Method: Actual Step for Iterations

- A full step would violate the bounds ($\boldsymbol{x} \geq \mathbf{0}, \boldsymbol{\lambda}_I \geq \mathbf{0}$).
  Therefore, a step-length $\alpha_k \in (0,1]$ is chosen

$$\left(\boldsymbol{x}^{(k+1)}, \boldsymbol{\lambda}_E^{(k+1)}, \boldsymbol{\lambda}_I^{(k+1)}\right) = \left(\boldsymbol{x}^{(k)}, \boldsymbol{\lambda}_E^{(k)}, \boldsymbol{\lambda}_I^{(k)}\right) + \alpha_k(\delta\boldsymbol{x}, \delta\boldsymbol{\lambda}_\mathrm{E}, \delta\boldsymbol{\lambda}_\mathrm{I})$$

- such that: $\boldsymbol{x}^{(k)} > \mathbf{0}, \boldsymbol{\lambda}_I^{(k)} > \mathbf{0}$

- The inequalities are strictly satisfied hence “interior point”

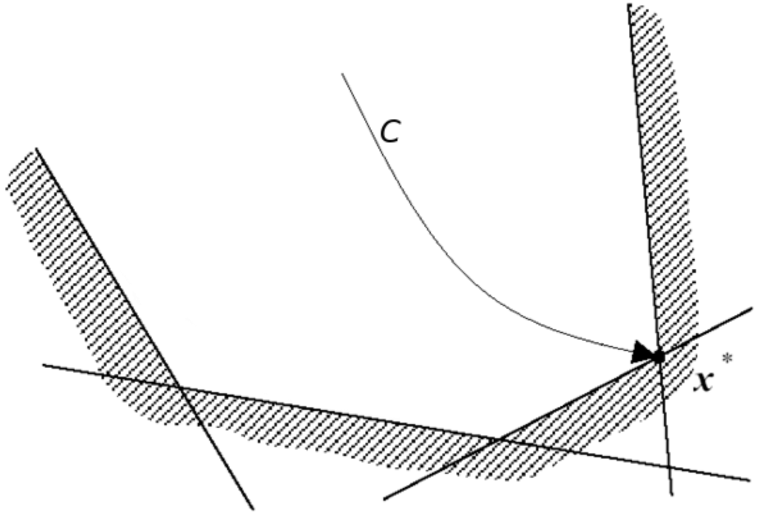
- The step-length α_k computed this way is often very small

- To achieve convergence, Newton’s method within the primal-dual framework is modified.

Adapted Newton's Method for LP (1)

- Each point $(\boldsymbol{x}_\tau, \boldsymbol{\lambda}_{E,\tau}, \boldsymbol{\lambda}_{I,\tau})$ on the central path C solves

$$\begin{cases} \boldsymbol{A}^T\boldsymbol{\lambda}_E + \boldsymbol{\lambda}_I = \boldsymbol{d} \\ \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ x_i\lambda_{I,i} = \tau, i = 1,2,\dots,n \\ \boldsymbol{x}_\tau > 0, \boldsymbol{\lambda}_{I,\tau} > 0 \end{cases}$$



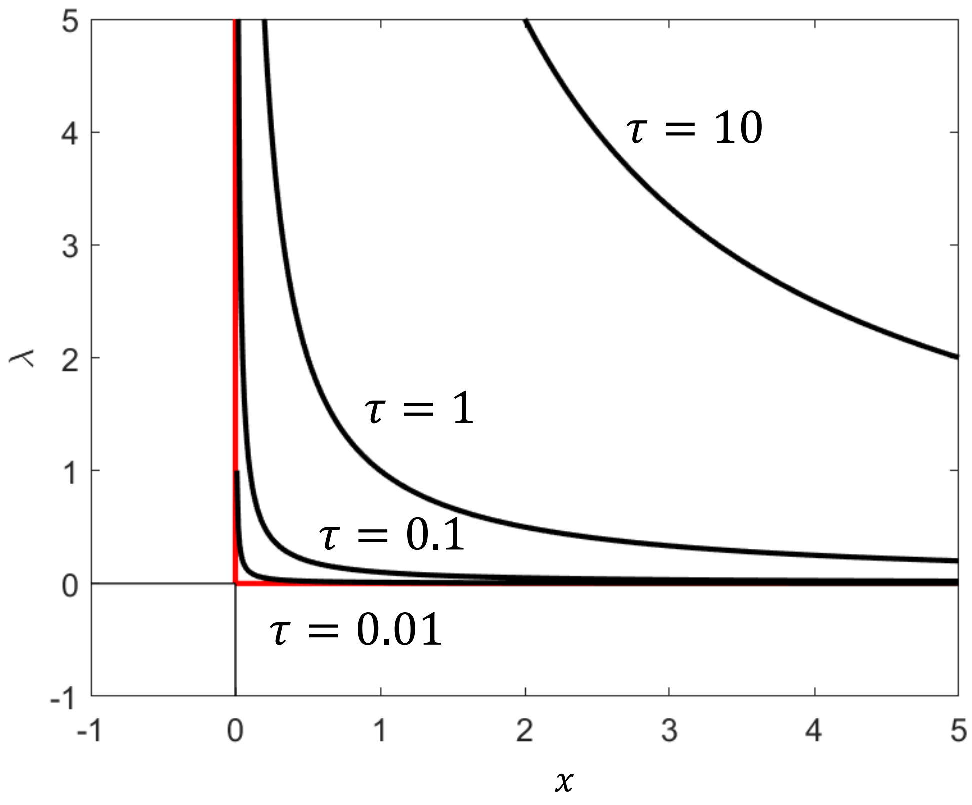
- We use the compact notation

$$\boldsymbol{F}(\boldsymbol{x}_\tau, \boldsymbol{\lambda}_{E,\tau}, \boldsymbol{\lambda}_{I,\tau}) = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \tau\boldsymbol{e} \end{bmatrix}, \ \boldsymbol{x}_\tau > \boldsymbol{0}, \boldsymbol{\lambda}_{I,\tau} > \boldsymbol{0} \qquad C = \{(\boldsymbol{x}_\tau, \boldsymbol{\lambda}_{E,\tau}, \boldsymbol{\lambda}_{I,\tau}) \mid \tau > 0\}$$

- The smaller τ, the better the approximation of the optimality conditions.

Graphic Interpretation of the Interior Point

- The complementary condition $x_i \lambda_{I,i} = 0$ is nonsmooth (corner)
- The smooth approximation $x_i \lambda_{I,i} = \tau$ gets closer to the corner for smaller τ
- The solutions are in in the interior of the inequality constraints, as both $\boldsymbol{x}_\tau > 0, \boldsymbol{\lambda}_{I,\tau} > 0$



Adapted Newton's Method for LP (2)

- Instead of pure a Newton step, the PDM points with $\tau > 0$ are taken → bigger steps
- For the technical implementation, two parameters σ and μ, with $\tau = \sigma\mu$ are introduced

$$\begin{bmatrix} \mathbf{0} & \boldsymbol{A}^T & \boldsymbol{I} \\ \boldsymbol{A} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{\Lambda}_I & \mathbf{0} & \boldsymbol{X} \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{x} \\ \delta\boldsymbol{\lambda}_E \\ \delta\boldsymbol{\lambda}_I \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ -\boldsymbol{X}\boldsymbol{\Lambda}_I\boldsymbol{e} + \sigma\mu\boldsymbol{e} \end{bmatrix}$$

- For $\sigma = 0$, we have the standard Newton step. With $\sigma = 1$ a point on the central path is taken. μ measures the average violation in the nonlinear constraints.

$$\sigma \in [0,1], \qquad \mu = \frac{1}{n}\sum_{i=1}^{n} x_i\lambda_{I,i} = \frac{\boldsymbol{x}^T\boldsymbol{\lambda}_I}{n}$$

Aachener Verfahrenstechnik RWTH AACHEN UNIVERSITY

Primal-Dual Method - Algorithm

- Given a feasible initial guess $(\boldsymbol{x}^0 > \boldsymbol{0}, \boldsymbol{\lambda}_E^0, \boldsymbol{\lambda}_I^0 > \boldsymbol{0})$
- for $k = 0,1,\ldots$
  - Solve $$\begin{bmatrix} \boldsymbol{0} & \boldsymbol{A}^T & \boldsymbol{I} \\ \boldsymbol{A} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{\Lambda}_I & \boldsymbol{0} & \boldsymbol{X} \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{x} \\ \delta\boldsymbol{\lambda}_E \\ \delta\boldsymbol{\lambda}_I \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ -\boldsymbol{X}\boldsymbol{\Lambda}_I\boldsymbol{e} + \sigma\mu\boldsymbol{e} \end{bmatrix}, \sigma^{(k)} \in [0,1],\ \mu^{(k)} = \frac{\boldsymbol{x}^{(k)^T}\boldsymbol{\lambda}_I^{(k)}}{n}$$
  - Set $$\left(\boldsymbol{x}^{(k+1)}, \boldsymbol{\lambda}_E^{(k+1)}, \boldsymbol{\lambda}_I^{(k+1)}\right) = \left(\boldsymbol{x}^{(k)}, \boldsymbol{\lambda}_E^{(k)}, \boldsymbol{\lambda}_I^{(k)}\right) + \alpha_k(\delta\boldsymbol{x}, \delta\boldsymbol{\lambda}_E, \delta\boldsymbol{\lambda}_I)$$

    computing α_k such that $\boldsymbol{x}^{(k+1)} > \boldsymbol{0}, \boldsymbol{\lambda}_I^{(k+1)} > \boldsymbol{0}$
- The choice of parameters is important.
- It is very difficult to find a strictly feasible initial solution.
- As the method works with the optimality conditions, it can be seen as an indirect method

Check Yourself

- Explain the primal-dual solution method.
- Explain what interior point means
- Define the central path and explain the adapted Newton’s method.

Applied Numerical Optimization

Prof. Alexander Mitsos, Ph.D.

Solvers for optimization with focus on linear programs

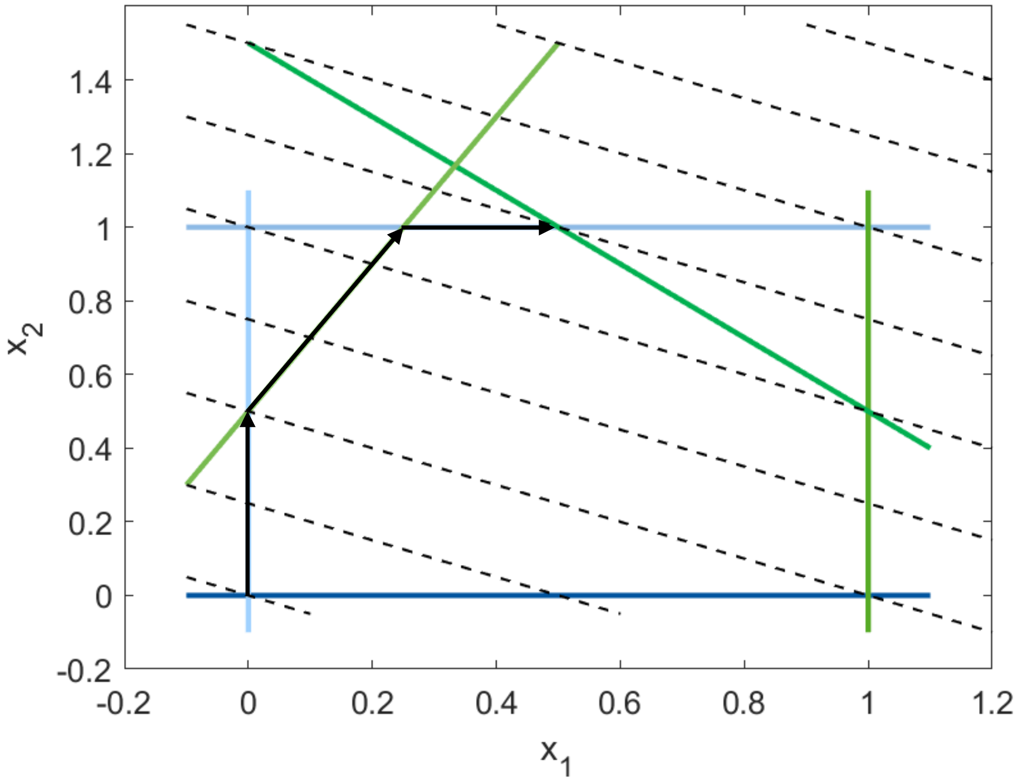
Aachener Verfahrenstechnik

RWTH AACHEN UNIVERSITY

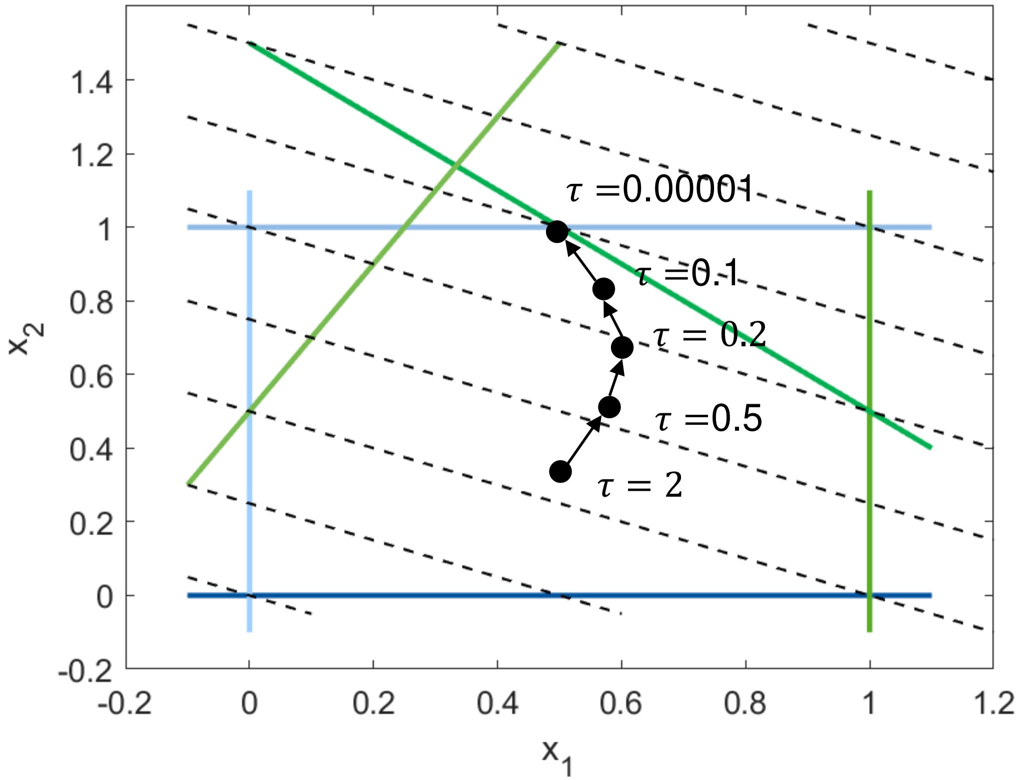
Simplex Method vs. Interior-Point Method

$$\begin{aligned} \min_{x_1,x_2} \quad & -x_1 - 2x_2 \\ \text{s.t.} \quad & x_1 + x_2 \leq 1.5 \\ & -2x_1 + x_2 \leq 0.5 \\ & 0 \leq x_1, x_2 \leq 1 \end{aligned}$$

Simplex method



Interior-point method



Complexity of Solution Methods

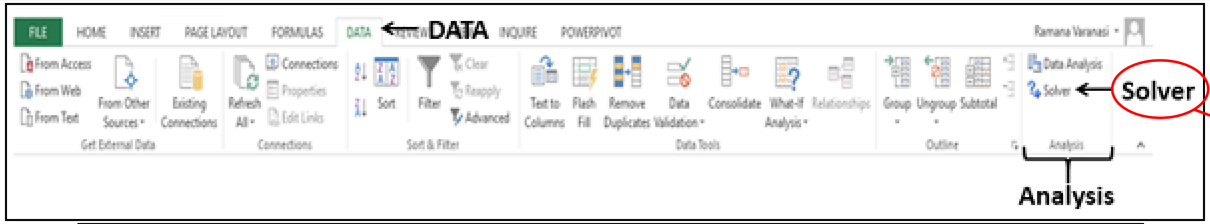
- Complexity of algorithms if often worst-case performance.
  Desired: CPU time is polynomial in the size of problem
  – Algorithms with factorial complexity are intractable for large-scale problems

- LPs are polynomial problems: algorithms exist with polynomial worst-case run time

- The Simplex method is an “active-set” method
  – Worst-case complexity is open problem.
  – In practice Simplex works exceptionally well for most LP instances
  – Simplex has polynomial average complexity

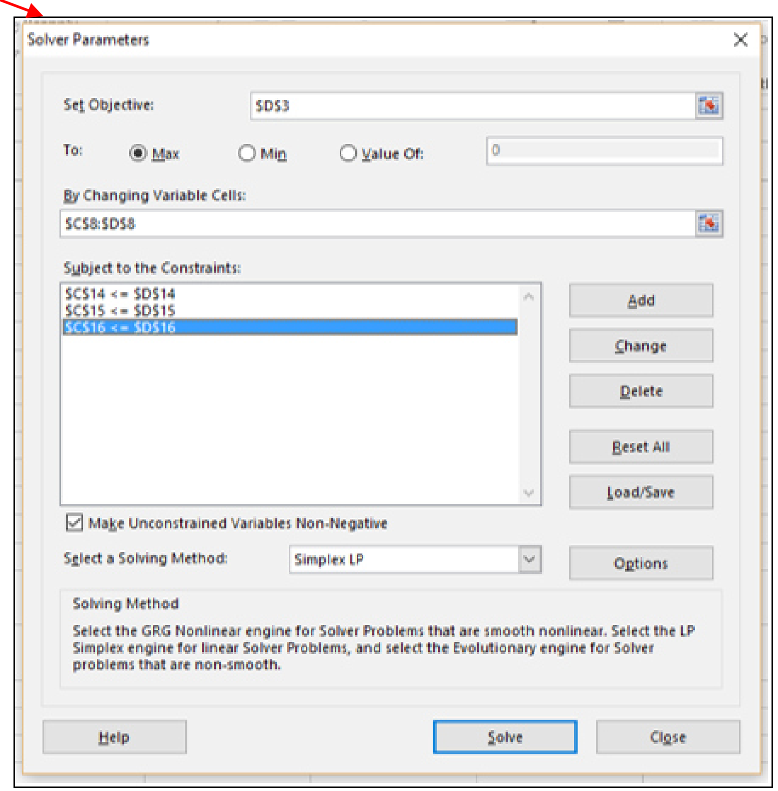
- Some interior-point methods have polynomial complexity for LPs
  – Most are in practice slow compared to the Simplex method
  – The best compete with Simplex for large-scale problems

Microsoft Excel



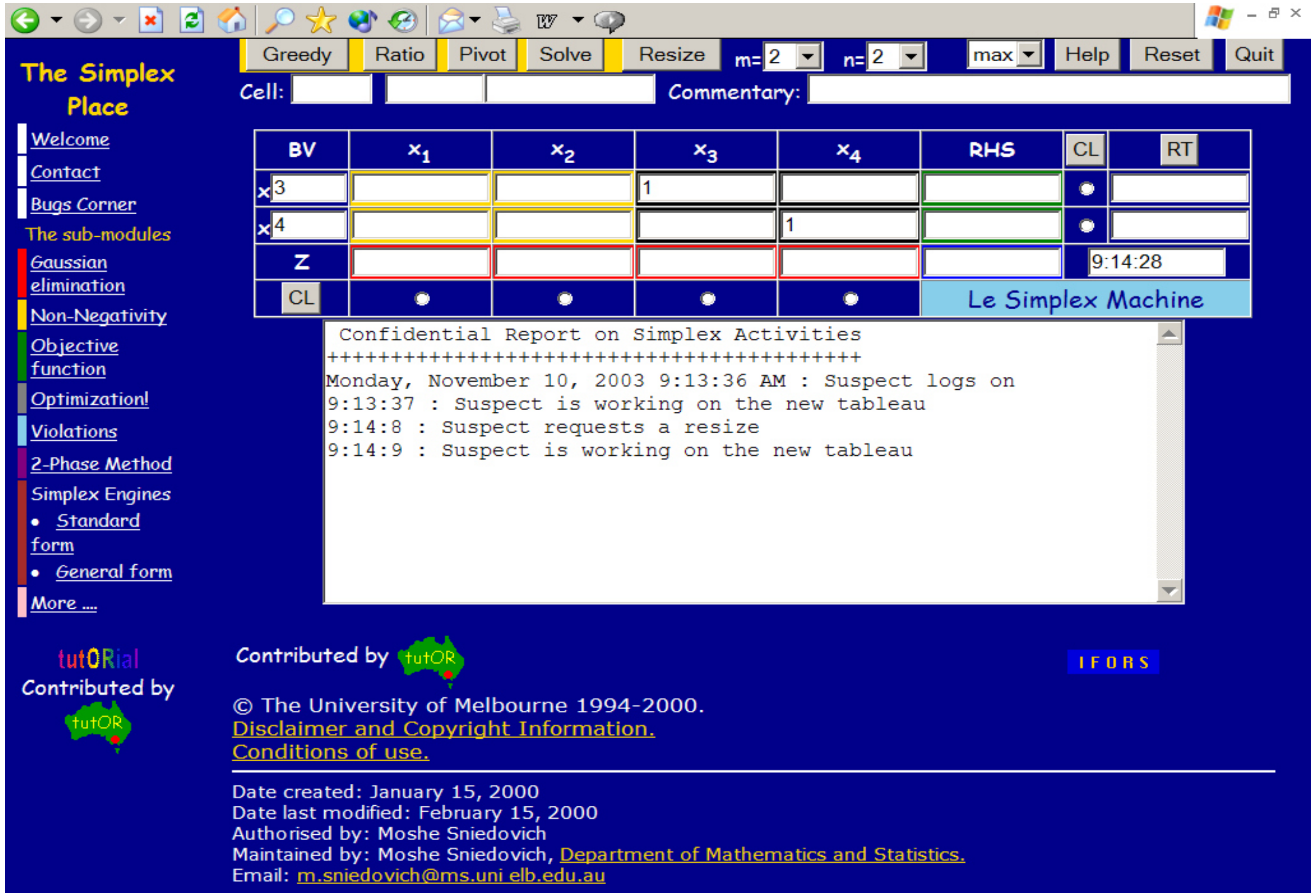
	A	B	C	D
1				
2		Unit Cost	50	Total Profit
3		Unit Price	100	=C12+D12
4		Adv. Cost per Unit	20	
5				
6			Quarter1	Quarter2
7		No. of Units Available	400	600
8		Adv. Budget	10000	10000
9		No. of Units Sold	=MIN(C8/C4,C7)	=MIN(D8/C4,D7)
10		Revenue	=C3*C9	=C3*D9
11		Expenses	=C2*C7+C8	=C2*D7+D8
12		Profit	=C10-C11	=D10-D11
13				
14		Total Adv. Budget	=C8+D8	20000
15		No. of Units sold in Quarter1	=C9	=C7
16		No. of Units sold in Quarter2	=D9	=D7

Applicable for small LPs



Pictures from: https://www.tutorialspoint.com

IFORS – The Simplex Place



Allows the user to participate in the solution process of an LP.

Great for educational purposes, but not suitable for solution of real problems!

Try it at: https://ifors.ms.unimelb.edu.au/tutorial/simplex/

The General Algebraic Modeling System (GAMS)

- GAMS is a state-of-the-art commercial modeling system for writing optimization problems. Input and output are represented by formalized text data.

```
*  A pipestill is a crude distillation unit, with several products
*  from the two (atmospheric and vacuum) towers
Table pipestill_yield(flow, crude)   pipestill yield (proportions by weight)
                    arabian-l   arabian-h   brega
 lv-naphtha       .035        .030        .045
 iv-naphtha       .100        .075        .135
 v-heat-oil       .390        .300        .430
 vacuum-dst       .285        .230         .280
 res-arab-l       .165
 res-arab-h                    .335
 res-brega                                  .100
```

```
purchase_cost ..
   purchase
      =E= sum(crude, crude_oil(crude)*crudedat(crude, 'price'))
         + flowrate('fuel-oil', 'in', 'fuel-imp')*fuel_imp_price;
transport_cost ..
   transport =E= sum(crude, crude_oil(crude)*crudedat(crude, 'transport'));
profit_equ ..
   profit =E= revenue - operating - purchase - transport;
model exxon /all/;
```

applicable for LPs with 10,000+ variables

- GAMS offers an **interface to all major commercial and academic solvers.**
- Demo/educational version available for free
- Similar systems: AMPL and AIMMS

PYOMO

- Pyomo is a Python-based open-source modeling language
- Python is popular language for computing
- Pyomo has interfaces to many major commercial and academic solvers
- A variety of optimization problems can be natively modeled and solved, including LP, NLP, stochastic and dynamic (pyomo.dae)
- User can implement their own solution algorithms
- Try it out: pyomo.org

JuMP

- JuMP is a Julia-based open-source modeling language
- Julia is a programming language that aims to combine high performance with ease of use
- JuMP supports several open-source and commercial solvers
- Several optimization problems can be modeled, e.g. LP, MILP, NLP, SDP
  - With InfiniteOpt.jl also infinite-dimensional, including stochastic, inverse problems, optimal control, dynamic optimization
- Try it out: jump.dev

Competitive Solvers

CPLEX[1]

X-PRESS[2]

Gurobi[3]

- Use a combination of Simplex, primal-dual, heuristics and trade secrets
- Free for academics, expensive for commercial purposes
- Available through library (more efficient) and various interfaces (Matlab, GAMS, AMPL, C/C++, …)
- Can solve both LP and Mixed-integer Linear Programs (stay tuned)
- Open-source alternatives exist and work fairly well for medium sized problems: COIN-OR, SCIP

[1] https://www.ibm.com/de-de/analytics/cplex-optimizer
[2] https://www.fico.com/de/products/fico-xpress-solver
[3] https://www.gurobi.com/products/gurobi-optimizer/

Check Yourself

- What are solution methods for LP?
- What are advantages and disadvantages?
- What are solvers for LP?
- What tools exist to model and solve optimization problems?