1a. Einführung

Definitionen und Klassifizierung

Optimierung SoSe 2020

Dr. Alexey Agaltsov

Plan

- Definitionen
- Klassifizierung
- Konvexe Programmierung
- Lineare Programmierung
- Diskrete Optimierung

Mathematisches Programm

$$\text{Minimiere} \quad f(x) \quad \text{über } \overbrace{x}^{\text{Entscheidungsvariable}} \in \mathcal{B}$$

$$\text{u.d.N.} \quad \left.\begin{aligned} g_i(x) &\leq 0, \quad i = 1, \dots, m_U \\ h_j(x) &= 0, \quad j = 1, \dots, m_G \end{aligned}\right\} \text{Nebenbedingungen}$$

$f: \mathcal{B} \to \mathbb{R}$ heißt Zielfunktion

$\mathcal{B}$ heißt Grundmenge

In diesem Kurs $\mathcal{B} \in \{\mathbb{R}^n, \mathbb{Z}^n\}$

$g_i, h_j: \mathcal{B} \to \mathbb{R}$ beliebige funktionen

Zulässiger Bereich

- Zulässiger Bereich ist gegeben durch:

$$\mathcal{F} = \{x \in \mathcal{B}: g_i(x) \leq 0 \ \forall i, h_j(x) = 0 \ \forall j\}$$

- Punkte $x \in \mathcal{F}$ heißen zulässige Lösungen

- $x \in \mathcal{F}$ heißt optimale Lösung falls:

$$f(x) \leq f(y) \quad \forall y \in \mathcal{F}$$

Optimierungsverfahren

- Wir wollen eine oder alle optimale Lösungen finden
- Analytische Lösungen sind selten vorhanden
- Optimierungsverfahren erzeugen eine Folge $x_1, x_2, \ldots$ die gegen eine optimale Lösung konvergiert

Existenz von Lösungen

- Man muss zuerst sicherstellen, dass es eine optimale Lösung gibt
- Dafür braucht man handhabbare Bedingungen, die Existenz von Lösungen garantieren, z.B. Satz von Weierstrass:

$$\textit{Ist } \mathcal{F} \textit{ kompakt und } f \in C(\mathcal{F}), \textit{ so } \exists x_* \in \mathcal{F}\colon f(x_*) = \inf_{\mathcal{F}} f$$

$\Longrightarrow$ Kapitel 2

Optimalitätskriterien

- Sei x_* eine optimale Lösung. Wie kann man x_* charakterisieren?
- Notwendige Optimalitätsbedingungen:

Sei x_* eine optimale Lösung, so erfüllt x_* …

- Hinreichende Optimalitätsbedingungen:

Erfülle x_* …, so ist x_* eine Optimale Lösung

Plan

- Definitionen
- Klassifizierung
- Konvexe Programmierung
- Lineare Programmierung
- Diskrete Optimierung

Klassifizierung der Programme

Gibt es Nebenbedingungen?

nein → unrestringiertes Programm

ja → restringiertes Programm

Klassifizierung der Programme

$$\begin{aligned} \text{Minimiere} \quad & f(x) \quad \text{über } x \in \mathcal{B} \\ \text{u.d.N.} \quad & g_i(x) \leq 0, \quad i = 1, \dots, m_U \\ & h_j(x) = 0, \quad j = 1, \dots, m_G \end{aligned}$$

Sei $\mathcal{B} = \mathbb{R}^n$. Man bezeichnet das Programm als:

- glatt falls $f, g_i, h_j \in C^1(\mathbb{R}^n)$
- konvex falls f, g_i konvex sind, h_j affin sind ($h_j(x) = a_j^T x + c_j$)
- linear falls f, g_i, h_j affin sind

Ist $\mathcal{B} = \mathbb{Z}^n$, so bezeichnet man das Problem als ganzzahlig

Beispiel

Minimiere $f(x)$ über $x \in \mathbb{R}^n$

$f \in C^1(\mathbb{R}^n)$

Das ist ein unrestringiertes glattes Programm

$\Longrightarrow$ Kapitel 2

Beispiel

$$\begin{aligned} \text{Minimiere} \quad & f(x) \text{ über } x \in \mathbb{R}^n \\ \text{u.d.N.} \quad & g_i(x) \leq 0, \quad i = 1, \dots, m_U \\ & h_j(x) = 0, \quad i = 1, \dots, m_G \end{aligned}$$

$$f, g_i, h_j \in C^1(\mathbb{R}^n)$$

Das ist ein restringiertes glattes Programm

$\Longrightarrow$ Kapitel 8

Lösung von Optimierungsproblemen

Allgemeine Optimierungsprobleme sind extrem schwierig zu lösen

Effiziente Methoden sind in Spezialfällen vorhanden, z.B.:

- glatte konvexe Programme
- lineare Programme

Plan

- Definitionen
- Klassifizierung
- Konvexe Programmierung
- Lineare Programmierung
- Diskrete Optimierung

Konvexe Programmierung

- Wir betrachten ein glattes konvexes Programm:

$$\text{Minimiere } f(x) \text{ über } x \in \mathbb{R}^n$$
$$\text{u.d.N. } g_i(x) \leq b_i, \quad i = 1, \ldots, m_U$$
$$a_j^T x = b_j, \quad j = 1, \ldots, m_G$$

$f, g_i \in C^1(\mathbb{R}^n)$ konvex sind

$a_j \in \mathbb{R}^n, b_j \in \mathbb{R}$

Konvexe Programmierung

Konvexe Programme sind einfach zu lösen:

- Jedes lokales Minimum ist global
- Es gibt mehrere effiziente und verlässliche Methode

Konvexe Programmierung

- Nach Einfachheit des Lösungsverfahrens lassen sich glatte konvexe Programme auch in die folgende Hierarchie einordnen:

  1. Quadratische Programme mit Gleichungsnebenbedingungen
  2. Glatte konvexe Programme mit Gleichungsnebenbedingungen
  3. Allgemeine glatte konvexe Programme

Quadratische Programme mit GNB

$$\begin{aligned} \text{Minimiere } & \tfrac{1}{2}x^T Q x - c^T x + r \ \text{ über } x \in \mathbb{R}^n \\ \text{u.d.N. } & Ax = b \end{aligned}$$

$\Longrightarrow$ Kapitel 5

$$Q \in \mathbb{S}^n_{\succcurlyeq}, c \in \mathbb{R}^n, r \in \mathbb{R}$$
$$A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m$$

- Man kann eine optimale Lösung aus der KKT-Gleichung bestimmen:

$$\begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x_* \\ \lambda_* \end{bmatrix} = \begin{bmatrix} c \\ b \end{bmatrix}$$

Glatte konvexe Programme mit GNB

$$\text{Minimiere } \; f(x) \text{ über } x \in \mathbb{R}^n$$
$$\text{u.d.N. } \; Ax = b$$

$\Longrightarrow$ Kapitel 5

$$f \in C^1(\mathbb{R}^n) \text{ konvex}$$
$$A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$$

- Löse konsekutive quadratische Approximationen mithilfe der KKT-Gleichung $\Rightarrow$ Newton-Verfahren:

$$x_{k+1} = x_k + \Delta x_k$$

$$\begin{bmatrix} \nabla^2 f(x_k) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x_{k+1} \\ \lambda_k \end{bmatrix} = \begin{bmatrix} -\nabla f(x_k) \\ b \end{bmatrix}$$

Allgemeine glatte konvexe Programme

$$\begin{aligned} \text{Minimiere } & f(x) \text{ über } x \in \mathbb{R}^n \\ \text{u.d.N. } & g_i(x) \leq 0, i = 1, \dots, m_U \\ & Ax = b \end{aligned}$$

$\Longrightarrow$ Kapitel 10

$$f, g_i \in C^1(\mathbb{R}^n) \text{ konvex}$$
$$A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$$

- Löse eine Folge von konsekutiven Approximationen die nur GNB haben
  $\Rightarrow$ Barriere-Methode:

$$\begin{aligned} \text{Minimiere } & f(x) + \tfrac{1}{t} \textstyle\sum_{i=1}^{m_U} -\log\big(-g_i(x)\big) \\ \text{u.d.N. } & Ax = b \qquad\qquad t \to \infty \end{aligned}$$

Plan

- Definitionen
- Klassifizierung
- Konvexe Programmierung
- Lineare Programmierung
- Diskrete Optimierung

Lineare Programmierung

- Wir betrachten zunächst ein lineares Programm:

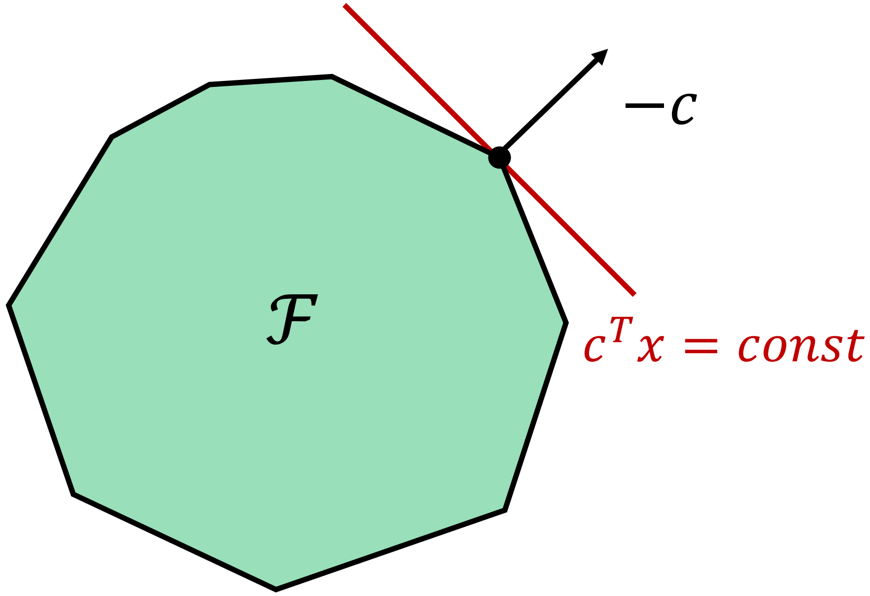
$$\begin{aligned} \text{Minimiere} \quad & c^T x \\ \text{u.d.N.} \quad & a_i^T x \leq b_i, \qquad i \in I_U \\ & a_i^T x = b_i, \qquad i \in I_G \end{aligned}$$

$$a_i, c \in \mathbb{R}^n, b_i \in \mathbb{R}$$

$$|I_U| < \infty, |I_G| < \infty$$

$\Longrightarrow$ Kapitel 7

Lineare Programmierung



$$\begin{aligned} &\text{Minimiere} && c^T x && \\ &\text{u.d.N.} && a_i^T x \le b_i, && i \in I_U \\ & && a_i^T x = b_i, && i \in I_G \end{aligned}$$

Fundamentaler Satz

Zulässigkeitsbereich ist ein Polyeder mit endlich vielen Ecken

Das Simplex-Verfahren

Das Simplex-Verfahren wurde im Jahre 1947 von G.B. Dantzig geprägt

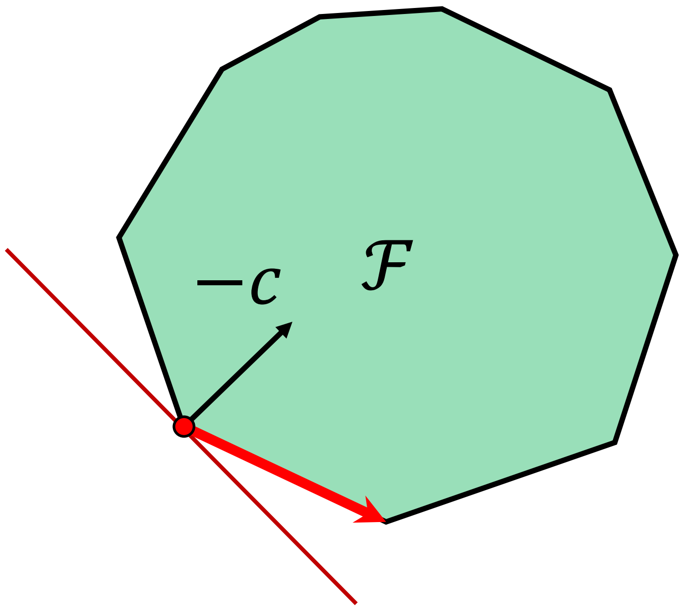
Später wurde es als eines der Top 10 Verfahren des XX Jahrhunderts ausgezeichnet

Das Simplex-Verfahren

Wähle eine Startecke

Finde eine Kante entlang welcher $c^T x$ kleiner wird

Gehe entlang dieser Kante zu einer benachbarten Ecke

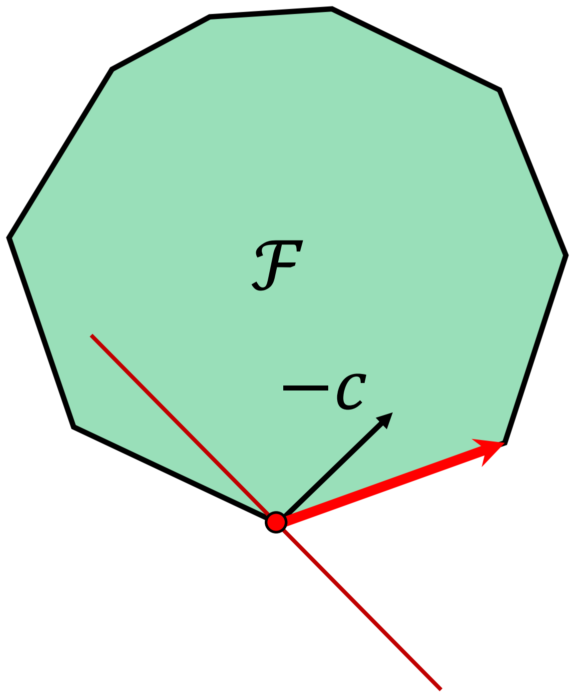


Das Simplex-Verfahren

Wähle eine Startecke

Finde eine Kante entlang welcher $c^T x$ kleiner wird

Gehe entlang dieser Kante zu einer benachbarten Ecke

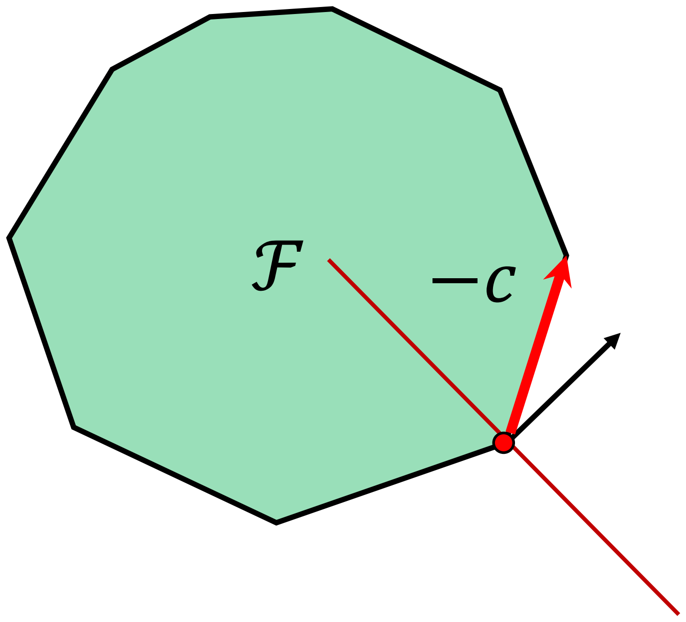


Das Simplex-Verfahren

Wähle eine Startecke

Finde eine Kante entlang welcher $c^T x$ kleiner wird

Gehe entlang dieser Kante zu einer benachbarten Ecke

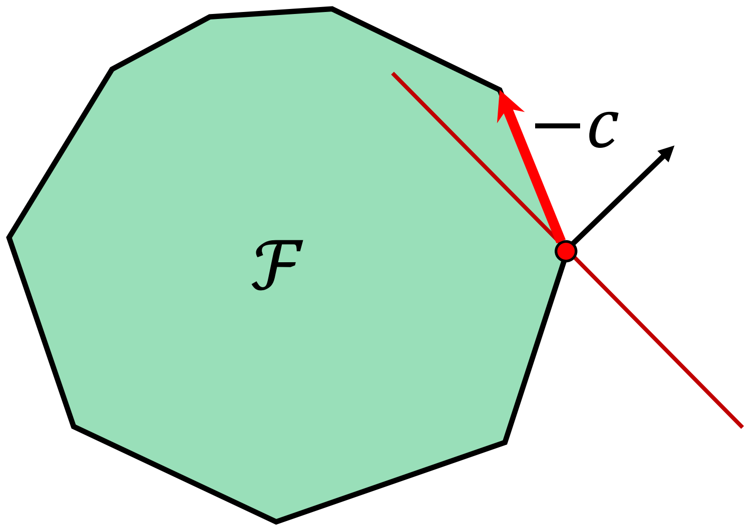


Das Simplex-Verfahren

Wähle eine Startecke

Finde eine Kante entlang welcher $c^T x$ kleiner wird

Gehe entlang dieser Kante zu einer benachbarten Ecke

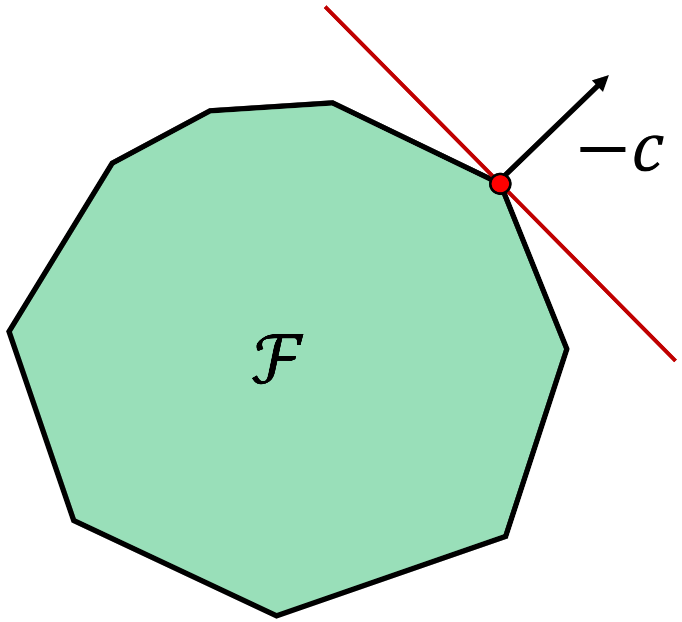


Das Simplex-Verfahren

Wähle eine Startecke

Finde eine Kante entlang welcher $c^T x$ kleiner wird

Gehe entlang dieser Kante zu einer benachbarten Ecke



Äquivalente Formulierungen

Mehrere nichtlineare Programme lassen sich äquivalent als ein lineares Programm formulieren

Beispiel: Maximum absoluter Beträge

Minimiere $\max_{i=1,\dots,m} \left| a_i^T x - y_i \right|$ über $x \in \mathbb{R}^n$

$a_i \in \mathbb{R}^n, y_i \in \mathbb{R}$

Minimiere t über $t \in \mathbb{R}, x \in \mathbb{R}^n$

u.d.N. $\left| a_i^T x - y_i \right| \leq t, \;\; i = 1, \dots, m$

Minimiere t über $t \in \mathbb{R}, x \in \mathbb{R}^n$

u.d.N. $a_i^T x - y_i \leq t \quad i = 1, \dots, m$

$-a_i^T x + y_i \leq t \quad i = 1, \dots, m$

Beispiel: Maximum einer Summe der Beträge

Minimiere $\sum_{i=1}^{m} |a_i^T x - y_i|$ über $x \in \mathbb{R}^n$

$a_i \in \mathbb{R}^n, y_i \in \mathbb{R}$

Minimiere $t_1 + \cdots + t_m$ über $x \in \mathbb{R}^n, t_i \in \mathbb{R}, i = 1, \ldots, m$

u.d.N. $\left|a_i^T x - y_i\right| \leq t_i, \quad i = 1, \ldots, m$

Minimiere $t_1 + \cdots + t_m$ über $x \in \mathbb{R}^n, t_i \in \mathbb{R}, i = 1, \ldots, m$

u.d.N. $a_i^T x - y_i \leq t_i \quad i = 1, \ldots m$

$-a_i^T x + y_i \leq t_i \quad i = 1, \ldots, m$

Plan

- Definitionen
- Klassifizierung
- Konvexe Programmierung
- Lineare Programmierung
- Diskrete Optimierung

Diskrete Optimierung

- Wir bezeichnen ein Optimierungsproblem als diskret falls Entscheidungsvariablen nur diskrete Werte annehmen dürfen

- Als spezielle Fälle unterscheiden wir kombinatorische und ganzzahlige Optimierungsprobleme

Kombinatorische Optimierung

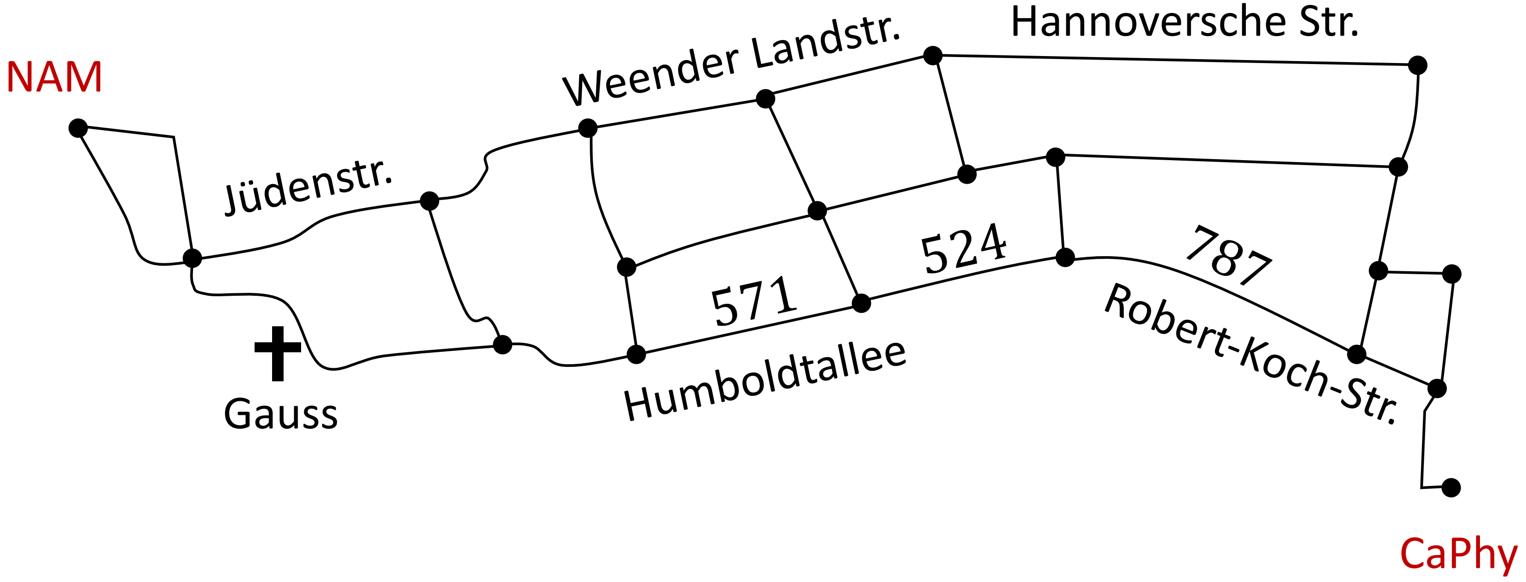
- Als kombinatorisches Optimierungsproblem bezeichnen wir diskrete Probleme, wobei es sich um graphänliche Strukturen handelt

  Problem des Handlungsreisenden

  Transportproblem

  Kürzester Pfad

Beispiel: Kürzester Pfad



Finde den kürzesten Weg von NAM nach CaPhy

Ganzzahlige Programmierung

- Als ganzzahliges Programm bezeichnen wir Optimierungsprobleme mit ganzzahligen Entscheidungsvariablen, d.h. $\mathcal{B} = \mathbb{Z}^n$

- Viele klassische Anwendungsprobleme lassen sich als ein ganzzahliges Programm formulieren

  Produktionsplanung

  Scheduling

  Rücksack-Problem

Beispiel

- Als lineares ganzzahliges Programm bezeichnen wir das Problem:

$$\text{Minimiere} \quad c^T x \quad \text{über } x \in \mathbb{Z}^n$$

$$\text{u.d.N.} \quad \begin{aligned} a_i^T x &= b_i,\ i \in I_G \\ a_i^T x &\leq b_i,\ i \in I_U \end{aligned}$$

$$c, a_i \in \mathbb{Z}^n, b_i \in \mathbb{Z}$$

$$|I_G| < \infty, |I_U| < \infty$$

Binäre Programmierung

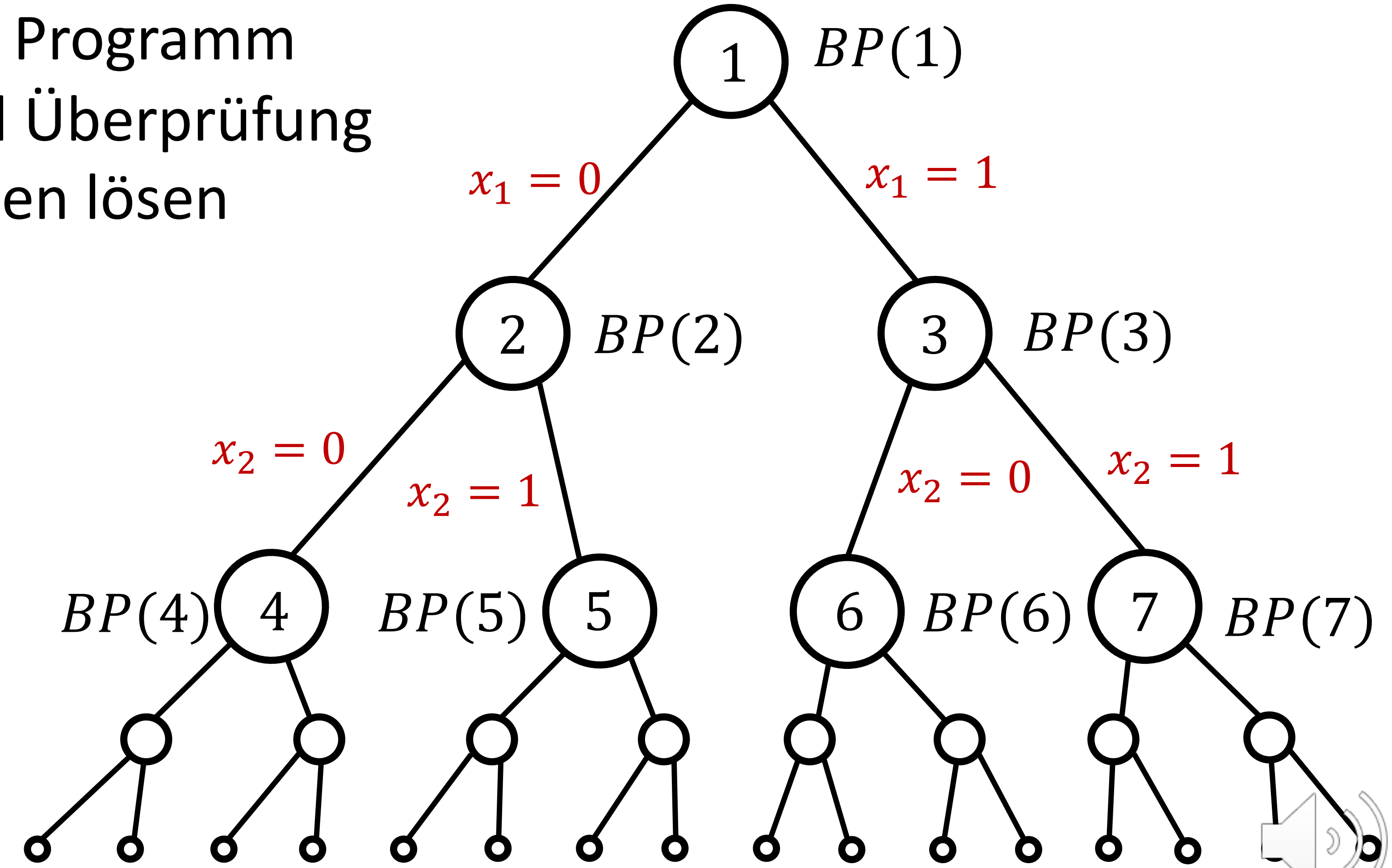
- Als Speziellen Fall unterscheiden wir $\{0,1\}$-lineare ganzzahlige Programme (oder einfach binäre Programme):

$$\begin{aligned} \text{Minimiere} \quad & c^T x \ \text{ über } x \in \{0,1\}^n \\ \text{u.d.N.} \quad & a_i^T x = b_i,\ i \in I_G \\ & a_i^T x \leq b_i,\ i \in I_U \end{aligned}$$

$$c, a_i \in \mathbb{Z}^n, b_i \in \mathbb{Z}$$
$$|I_G| < \infty, |I_U| < \infty$$

Binäre Programmierung

- Man kann ein binäres Programm durch Aufzählung und Überprüfung aller möglichen Punkten lösen



Binäre Programmierung

- Für n Entscheidungsvariablen gibt es 2^n Punkte zu überprüfen
- Angenommen, wir überprüfen $2^{30} \approx 10^9$ Punkte pro Sekunde

Variablen	30	40	50	60	70
Zeit	$1\ s$	$16.7\ m$	$12.1\ Tage$	$34\ Yahre$	$34{,}865\ Yahre$

Branch and Bound

Aufzählung aller Punkte ist langsam für meisten Probleme

Verzweigungsbaumverfahren (engl. Branch and bound, B&B) versucht es, durch eine intelligente Aufzählung viele Variante auszuschließen

Es wurde im Jahre 1954 von G. B. Dantzig, L. Ford und R. Fulkerson als ein Lösungsverfahren für das Problem des Handlungsreisenden geprägt

Plan

- Definitionen
- Klassifizierung
- Konvexe Programmierung
- Lineare Programmierung
- Diskrete Optimierung

Nächstes Video

- 1b. Einführung: Modellierung