Lemma D'

Gegeben ist ein s-t-Netzwerk $G = (V, E)$. Sei f ein Fluß, der während der Ausführung der Edmonds–Karp–Variante vorkommt und f' der Fluß nach der anschließenden Augmentierung.

Dann gilt: $\delta'(s, v) \geq \delta(s, v)$ für alle $v \in V$.

Hierbei sind $\delta(u, v)$ (bzw. $\delta'(u, v)$) die Kosten des billigsten Pfads von u nach v in G_f (bzw. in $G_{f'}$).

Beweis:

Wie Lemma D

Theorem

Die Edmonds–Karp–Variante findet einen Fluß mit Betrag B und minimalen Kosten in endlich vielen Schritten.

Beweis

- Solange die Kosten der augmentierenden Pfade gleichbleiben, werden unter ihnen kürzeste gewählt und der Algorithmus arbeitet wie der Edmonds–Karp–Algorithmus.

  $\Rightarrow$ Nach $O(|E| \cdot |V|)$ Iterationen, wird ein Pfad mit höheren Kosten gewählt (Lemma D').

- Es gibt nur endlich viele, mögliche augmentierende Pfade und damit nur endlich viele mögliche Kosten.

Bipartites Matching maximalen Gewichts

Gegeben:

Ein bipartiter, ungerichteter Graph (V_1, V_2, E) und eine Gewichtsfunktion $b\colon E \to \mathbf{R}_0^+$.

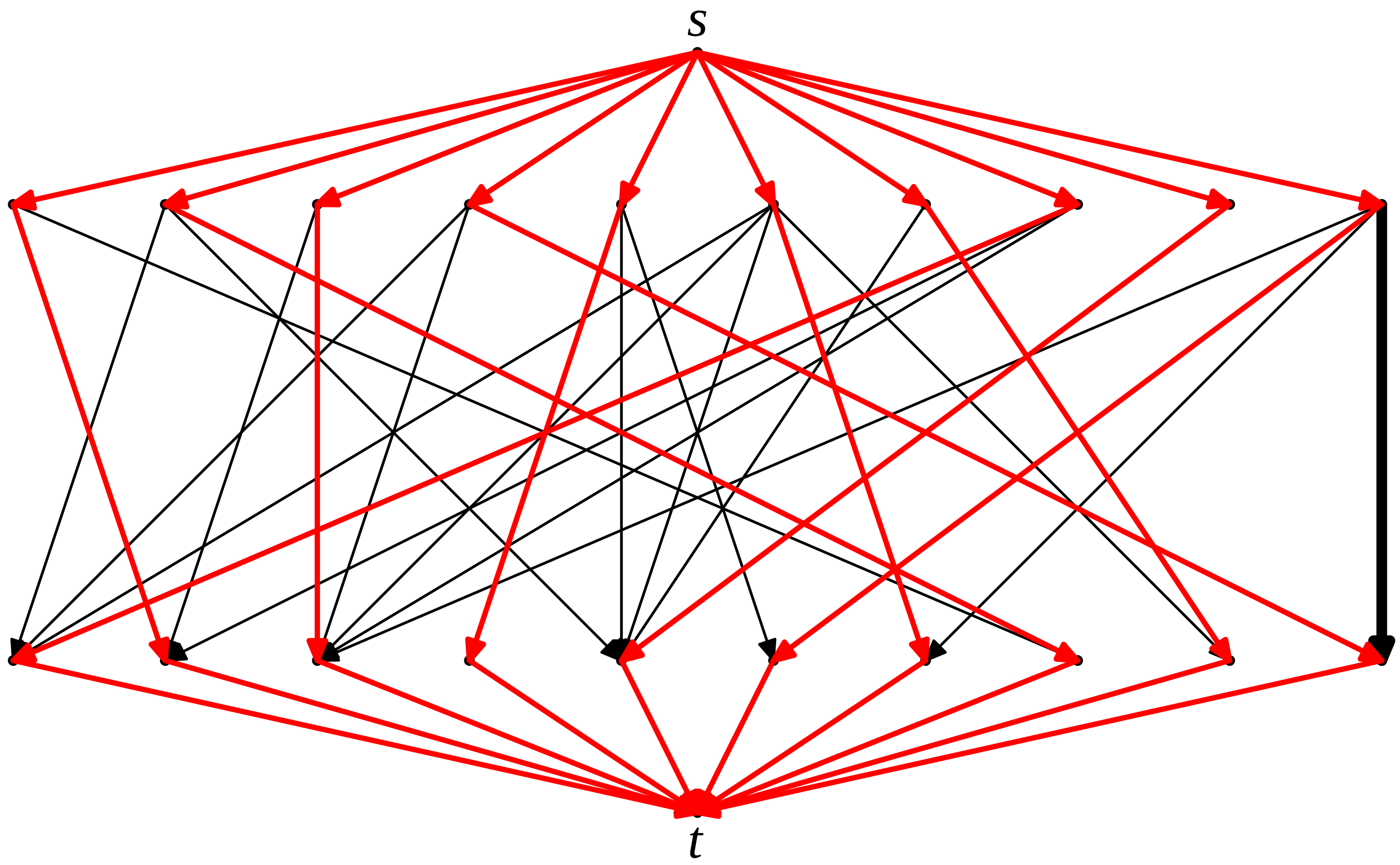
Gesucht:

Ein Matching mit maximalem Gewicht.

Das Gewicht eines Matchings M ist

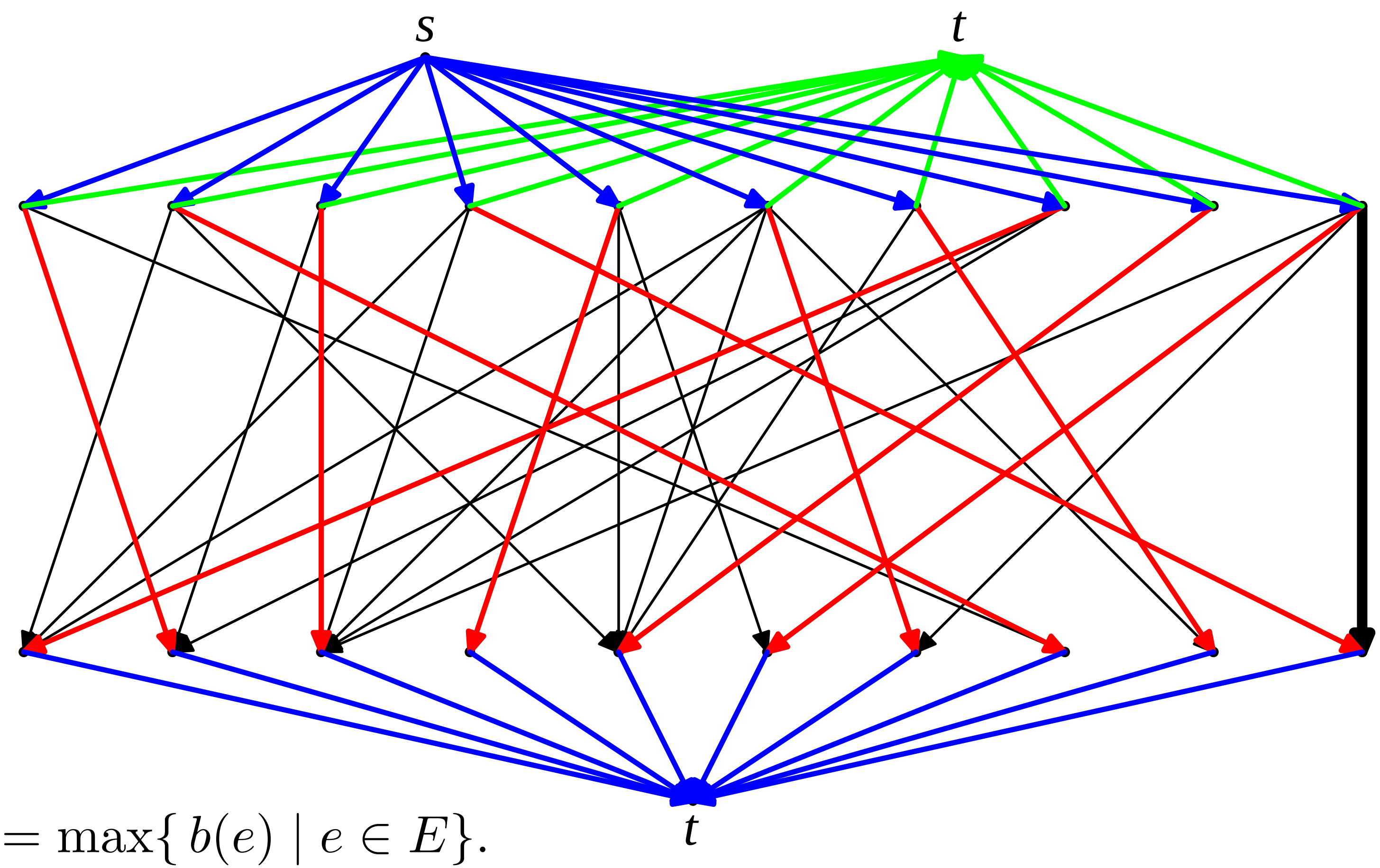
$$b(M) = \sum_{e \in M} b(e).$$

Beispiel



Alle Kanten haben Gewicht 1, nur die dicke Kante hat 3.

Hat das perfekte Matching auch maximales Gewicht?

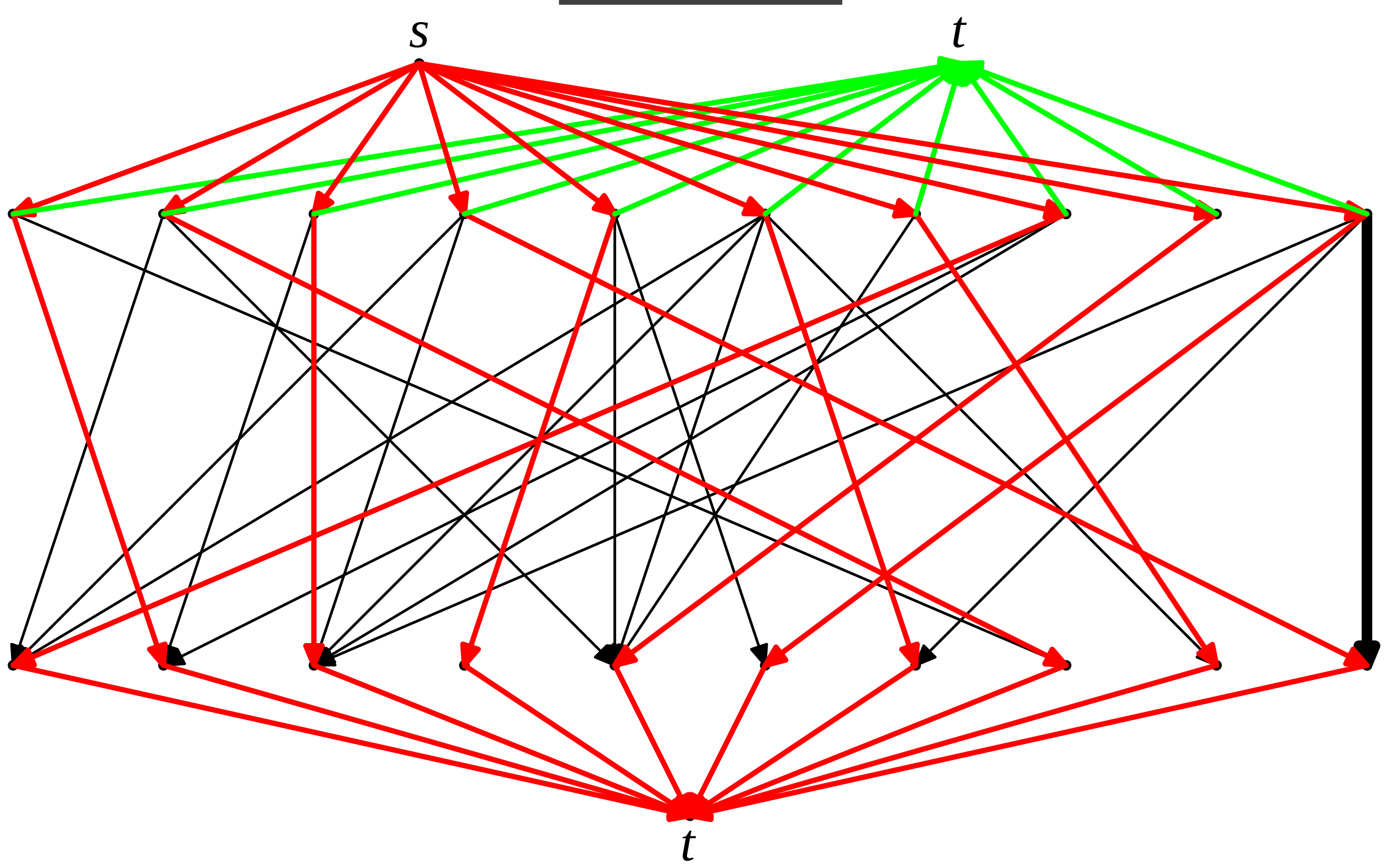


$w^* = \max\{\, b(e) \mid e \in E \}$.

Blaue Kanten haben Kosten 0, grüne Kanten haben Kosten w^*,
eine Matching-Kante e hat Kosten $w^* - b(e)$.

Matching mit Gewicht $b \Longrightarrow$ Fluß mit Kosten $nw^* - b$.

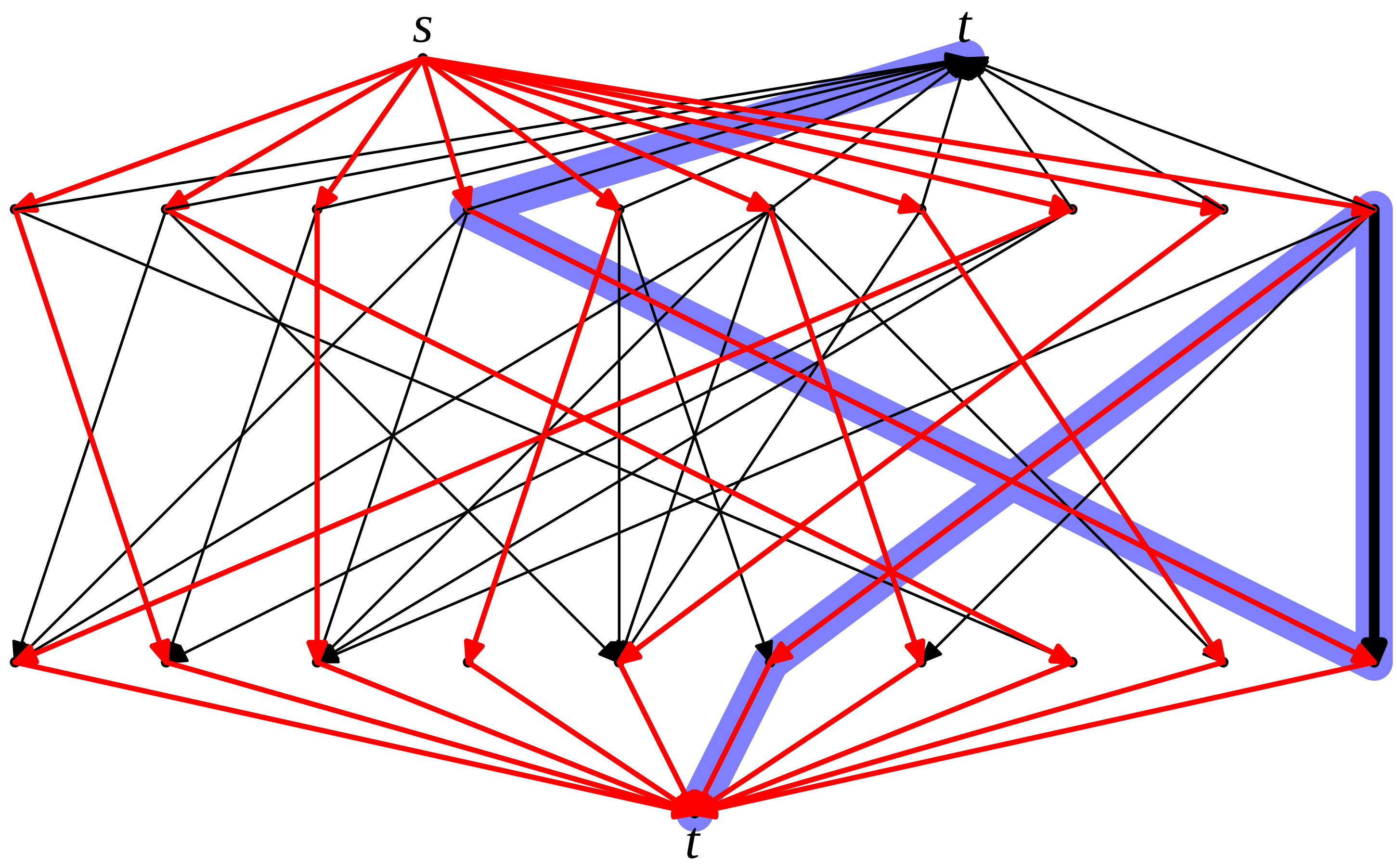
Beispiel



Dem perfekten Matching entspricht der rote Fluß mit Betrag 10.

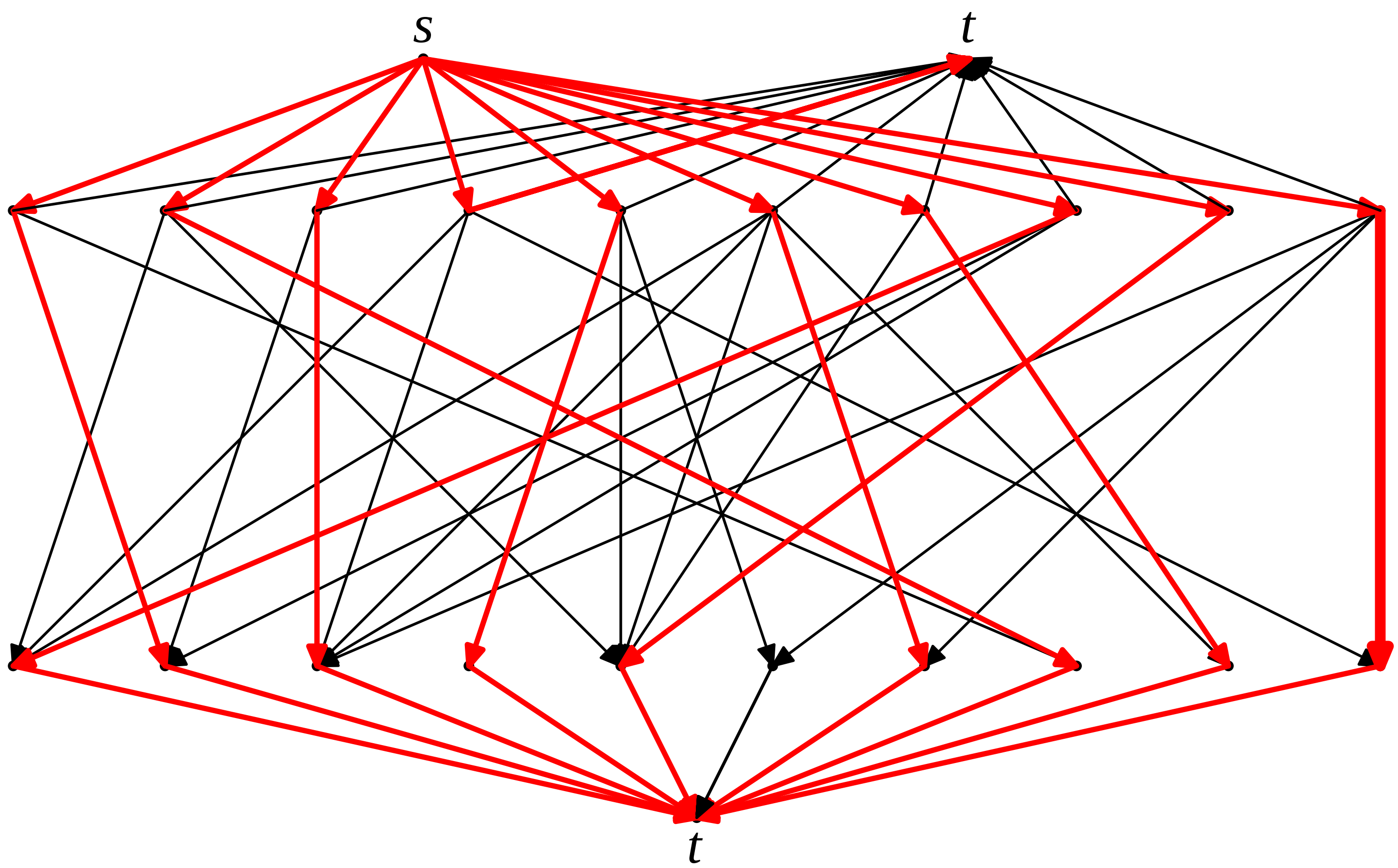
Wie groß sind die Kosten? ($w^* = 3$)

Beispiel



Die Kosten sind nicht minimal, denn es gibt einen Kreis in G_f mit negativen Kosten.

Beispiel



Jetzt hat der Fluß Betrag 10 und minimale Kosten. Das entsprechende Matching ist kleiner, hat aber maximales Gewicht.

Bipartites Matching maximalen Gewichts

Bipartites Matching maximalen Gewichts läßt sich mittels Flusses minimaler Kosten lösen:

Die Kosten des Flusses sind

- $(n - |M|)w^*$ und
- $\sum_{e \in M} (w^* - b(e))$

Zusammen sind es $nw^* - b(M)$ (mit $n = |V_1| \geq |V_2|$)

Dieser Wert ist minimal, wenn $b(M)$ maximal ist.$\square$

Lineares Programmieren

Eingabe:

Eine Menge von linearen Ungleichungen $S_i\colon \sum_{j=1}^{m} A_{ij}x_j \leq b_i$.

Eine lineare Optimierungsfunktion $f(x) = \sum_{j=1}^{m} c_j x_j$.

Problem:

Gesucht ist eine Belegung $x \geq 0$, für die alle Ungleichungen erfüllt sind und dir $f(x)$ maximiert.

Beispiel

Gegeben ein s-t-Netzwerk $G = (V, E)$.

Maximiere $\sum_{v \in V} f(s, v)$ unter den Bedingungen

- $f(u, v) = -f(u, v)$ für alle $(u, v) \in E$,
- $\sum_{u \in V} f(v, u) = 0$ für alle $u \in V - \{s, t\}$,
- $f(u, v) \leq c(u, v)$.

$\Rightarrow$ Lineares Programm für das Maximale-Fluß-Problem.

Modellierung als Lineares Programm

Maximiere $c^T x$ unter $Ax \leq b$ und $x \geq 0$.

Dabei ist $x \in \mathbf{R}^n$, $c \in \mathbf{R}^n$, $b \in \mathbf{R}^m$ und $A \in \mathbf{R}^{m \times n}$.

Viele abweichende Probleme lassen sich in diese Form bringen:

- Minimiere $c^T x \quad \mapsto \quad$ Maximiere $-c^T x$
- $a^T x \geq b \quad \mapsto \quad -a^T x \leq -b$
- Unbeschränktes $x_i \quad \mapsto \quad x_i' - x_i''$ mit $x_i', x_i'' \geq 0$

Beispiel: Wassermischen

Wir brauchen 100 Liter Wasser mit 60°C und nehmen

- x_1 Liter mit 20°C zu 1 DM/l
- x_2 Liter mit 50°C zu 2 DM/l
- x_3 Liter mit 100°C zu 3 DM/l

Minimiere $x_1 + 2x_2 + 3x_3$ (die Kosten) unter

- $x_1 + x_2 + x_3 = 100$
- $20x_1 + 50x_2 + 100x_3 = 6000$

Beispiel: Wassermischen

$x_1 + x_2 + x_3 = 100$

$20x_1 + 50x_2 + 100x_3 = 6000$