Transportation Problem

North-West Corner, Minimum Cost, Vogel Approximation and MODI

English

Wolfgang Boehme

wolfgang.boehme@economic-modeling.de

www.economic-modeling.de

25 October 2026

This document introduces the transportation problem and explains all four solution methods with a fully worked example.

The documentation on this website was developed with the support of the AI Claude (Anthropic).

Table of Contents

1 Introduction and Problem Statement **2**

2 Solution Methods: Overview **3**

3 Initial Solution: North-West Corner Method **3**

4 Initial Solution: Minimum Cost Method **3**

5 Initial Solution: Vogel Approximation (VAM) **4**

6 Optimisation: MODI Method (Stepping Stone) **4**

7 Complete Example **5**
- 7.1 Problem Statement . . . 5
- 7.2 Method 1: North-West Corner-Initial solution . . . 5
- 7.3 Method 2: Minimum Cost-Initial solution . . . 6
- 7.4 Method 3: Vogel Approximation . . . 6
- 7.5 MODI-Optimierung der North-West Corner-Solution . . . 7
  - 7.5.1 MODI-Iteration 1 . . . 8
  - 7.5.2 MODI-Iteration 2 . . . 8
  - 7.5.3 MODI-Iteration 3: Optimalitätsprüfung . . . 9
- 7.6 Optimale Solution . . . 9
- 7.7 Transportnetzwerk . . . 10
- 7.8 Methodnvergleich . . . 10

8 Tool Description **10**
- 8.1 Eingabe . . . 10
- 8.2 Methodn und Ausgabe . . . 11

9 Summary **11**

1 Introduction and Problem Statement

Das **Transportation problem** ist ein klassisches Optimierungsproblem der linearen Programmierung. Es modelliert den kostengünstigsten Transport eines homogenen Gutes von m Suppliern zu n Consumern.

Typische Anwendungen

- Verteilung von Gütern aus Lagerhäusern an Einzelhandelsgeschäfte
- Mineralöltransport von Raffinerien zu Tankstellen
- Personalzuweisung von Standorten zu Projekten
- Rohstofftransport von Minen zu Fabriken

Definition: Transportation Problem

Gegeben sind:

- m **Supplier** $A_1, \ldots, A_m$ mit Supplysmengen $a_1, \ldots, a_m > 0$
- n **Consumer** $N_1, \ldots, N_n$ mit Nachfragemengen $b_1, \ldots, b_n > 0$
- **Transport cost** $c_{ij} \geq 0$ pro Einheit von A_i nach N_j

Gesucht sind Transport quantityn $x_{ij} \geq 0$, die folgendes LP minimisen:

$$\min \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

subject to den Nebenbedingungen:

$$\sum_{j=1}^{n} x_{ij} = a_i \quad \forall\, i = 1, \ldots, m \quad \text{(Supplysbedingung)}$$

$$\sum_{i=1}^{m} x_{ij} = b_j \quad \forall\, j = 1, \ldots, n \quad \text{(Nachfragebedingung)}$$

$$x_{ij} \geq 0 \quad \forall\, i, j$$

Definition: Balancedes Transportation problem

Das Transportation problem heißt **balanced** (balanciert), wenn:

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$$

Ist es nicht balanced, fügt man einen **fiktiven Supplier** (falls $\sum b_j > \sum a_i$) oder **fiktiven Consumer** (falls $\sum a_i > \sum b_j$) mit Cost 0 hinzu.

Satz 1: Lösbarkeit und Optimalität

1. Jedes balancede Transportation problem besitzt eine Optimal solution.
2. Die Anzahl der Basic variablen (Belegungen > 0) in einer nicht-entarteten Basic solution beträgt genau $m + n - 1$.
3. Das Transportation problem ist total unimodular: Bei ganzzahligen a_i, b_j ist die Optimal solution ebenfalls ganzzahlig.

2 Solution Methods: Overview

Die Solution erfolgt in zwei Phasen:

1. **Initial solution** (feasiblee Basic solution): Eine von drei Methodn liefert eine erste feasiblee Solution.
2. **Optimierung** (MODI-Method): Iterative Verbesserung bis zur Optimalität.

3 Initial Solution: North-West Corner Method

Algorithm: North-West Corner Method

1. Beginne in der **nordwestlichen Ecke** des Tableaus: Zelle $(1, 1)$.
2. Weise zu: $x_{ij} = \min(a_i, b_j)$.
3. Reduziere a_i und b_j um x_{ij}.
4. Falls $a_i = 0$: gehe eine Row nach unten. Falls $b_j = 0$: gehe eine Column nach rechts. Falls $a_i = b_j = 0$ (Entartung): gehe diagonal.
5. Wiederhole bis alle Mengen aufgeteilt sind.

Bewertung: Einfach und schnell, ignoriert aber die Transport cost vollständig – liefert oft weit vom Optimum entfernte Solutionen.

4 Initial Solution: Minimum Cost Method

Algorithm: Minimum Cost Method

1. Wähle die Zelle (i^*, j^*) mit den **niedrigsten Transport cost** $c_{i^*j^*}$.
2. Weise zu: $x_{i^*j^*} = \min(a_{i^*}, b_{j^*})$.
3. Streiche die erschöpfte Row oder Column.
4. Wiederhole mit der nächst-günstigsten verfügbaren Zelle.

Bewertung: Besser als North-West Corner, da Cost berücksichtigt werden. Kann aber in Spezialfällen suboptimale Initial solutionen liefern.

5 Initial Solution: Vogel Approximation (VAM)

Algorithm: Vogel Approximation

1. Berechne für jede Row und Column die **Strafkosten** = Differenz zwischen dem kleinsten und zweitkleinsten Costwert.
2. Wähle die Row/Column mit der **höchsten Strafkosten**.
3. Weise in der gewählten Row/Column der Zelle mit den niedrigsten Cost zu: $x_{ij} = \min(a_i, b_j)$.
4. Streiche erschöpfte Row/Column und berechne Strafkosten neu.
5. Wiederhole bis alle Mengen verteilt sind.

Bewertung: Liefert subject to den drei Startmethoden die beste Solution – oft bereits nahe am Optimum oder sogar optimal.

Intuition der Strafkosten: Die Strafkosten messen den *Verlust*, der entsteht, wenn man die günstigste Zelle einer Row/Column *nicht* wählt. Eine hohe Strafe signalisiert: Hier sollte man unbedingt die günstigste Option wählen, sonst wird es teuer.

6 Optimisation: MODI Method (Stepping Stone)

Algorithmus: MODI-Method (Modified Distribution Method)

Grundprinzip: Berechne for all Nicht-Basiszellen die **reduzierten Cost**. Falls alle ≥ 0, ist die Solution optimal. Sonst verbessere durch einen Loopntausch.

Stepe pro Iteration:

1. **Potentiale berechnen:** Setze $u_1 = 0$. Für alle Basiszellen gilt $c_{ij} = u_i + v_j$. Löse das Gleichungssystem für u_i und v_j.
2. **Reduzierte Cost:** Für alle Nicht-Basiszellen: $\delta_{ij} = c_{ij} - u_i - v_j$.

3. **Optimalitätsprüfung:** Gilt $\delta_{ij} \geq 0$ for all Nicht-Basiszellen? $\Rightarrow$ **Optimal, STOPP**.
4. **Enteringe Variable:** Wähle Zelle (i^*, j^*) mit $\delta_{i^*j^*} = \min_{ij} \delta_{ij} < 0$.
5. **Loop finden:** Finde eine geschlossene Loop durch (i^*, j^*) und Basiszellen (abwechselnd Rown- und Columnnbewegungen).
6. **Verschiebungsmenge:** $\theta = \min$ aller Belegungen an den (-1)-Positionen der Loop.
7. **Aktualisierung:** $(+1)$-Zellen $+\theta$, (-1)-Zellen $-\theta$. Die (-1)-Zelle mit Belegung θ verlässt die Basis.

Satz 2: Korrektheit der MODI-Method

1. Die reduzierten Cost δ_{ij} geben an, um wieviel sich die Total cost pro Einheit ändern, wenn x_{ij} in die Basis aufgenommen wird.
2. Ein $\delta_{ij} < 0$ zeigt eine Verbesserungsmöglichkeit: Die Cost sinken um $|\delta_{ij}| \cdot \theta$.
3. Bei jeder Iteration gilt: $\Delta K = \delta_{i^*j^*} \cdot \theta < 0$ (strenger Abstieg).
4. Die MODI-Method terminiert nach endlich vielen Stepen mit der Optimal solution.

7 Complete Example

7.1 Problem Statement

Beispiel: Güterverteilung (3 Supplier – 4 Consumer)

	N_1	N_2	N_3	N_4	Supply
A_1	2	3	1	5	120
A_2	7	3	4	6	80
A_3	5	4	7	3	100
Nachfrage	90	70	80	60	300

Das Problem ist balanced: $120 + 80 + 100 = 300 = 90 + 70 + 80 + 60$.

7.2 Method 1: North-West Corner-Initial solution

Allocation Step für Step:

1. $(1,1)$: $\min(120, 90) = 90 \Rightarrow x_{11} = 90$. $a_1 = 30$, $b_1 = 0$ (Column 1 erschöpft).
2. $(1,2)$: $\min(30, 70) = 30 \Rightarrow x_{12} = 30$. $a_1 = 0$, $b_2 = 40$ (Row 1 erschöpft).

3. $(2,2)$: $\min(80,40)=40 \Rightarrow x_{22}=40$. $a_2=40$, $b_2=0$.
4. $(2,3)$: $\min(40,80)=40 \Rightarrow x_{23}=40$. $a_2=0$, $b_3=40$.
5. $(3,3)$: $\min(100,40)=40 \Rightarrow x_{33}=40$. $a_3=60$, $b_3=0$.
6. $(3,4)$: $\min(60,60)=60 \Rightarrow x_{34}=60$. Fertig.

North-West Corner-Tableau:

	N_1	N_2	N_3	N_4	Supply
A_1	**90** (2)	**30** (3)	(1)	(5)	120
A_2	(7)	**40** (3)	**40** (4)	(6)	80
A_3	(5)	(4)	**40** (7)	**60** (3)	100
Nachfrage	90	70	80	60	

Total cost (NW): $90\cdot 2+30\cdot 3+40\cdot 3+40\cdot 4+40\cdot 7+60\cdot 3=180+90+120+160+280+180=$ **1010**

7.3 Method 2: Minimum Cost-Initial solution

Sortierung der Zellen nach Cost: $(1,3)=1$, $(1,1)=2$, $(1,2)=3$, $(2,2)=3$, ...

1. $c_{13}=1$: $x_{13}=\min(120,80)=80$. $a_1=40$, $b_3=0$.
2. $c_{11}=2$ (nächst-günstig): $x_{11}=\min(40,90)=40$. $a_1=0$, $b_1=50$.
3. $c_{22}=3$: $x_{22}=\min(80,70)=70$. $a_2=10$, $b_2=0$.
4. $c_{34}=3$: $x_{34}=\min(100,60)=60$. $a_3=40$, $b_4=0$.
5. $c_{21}=7$? Nein – nächste verfügbar: $c_{31}=5$ (da $b_1=50$ noch offen): $x_{31}=\min(40,50)=40$. $a_3=0$, $b_1=10$.
6. $c_{21}=7$: $x_{21}=\min(10,10)=10$. Fertig.

Total cost (MC): $40\cdot 2+80\cdot 1+10\cdot 7+70\cdot 3+40\cdot 5+60\cdot 3=80+80+70+210+200+180=$ **820**

7.4 Method 3: Vogel Approximation

Strafkosten-Computation (Iteration 1):

	N_1	N_2	N_3	N_4	Supply	**Strafe**
A_1	2	3	1	5	120	$3-1=$ **2**
A_2	7	3	4	6	80	$4-3=$ **1**
A_3	5	4	7	3	100	$4-3=$ **1**
Nachfrage	90	70	80	60		
Strafe	$5-2=3$	$3-3=0$	$4-1=$ **3**	$5-3=2$		

Höchste Strafe: Column N_1 und N_3 (Strafe $= 3$). Wähle N_3: günstigste Zelle ist $(1, 3)$ mit $c = 1$.

$x_{13} = \min(120, 80) = 80$. Column N_3 erschöpft.

Iteration 2: Ohne Column N_3:

	N_1	N_2	N_4	Supply	Strafe
A_1	2	3	5	40	$3 - 2 = 1$
A_2	7	3	6	80	$6 - 3 = 3$
A_3	5	4	3	100	$4 - 3 = 1$
Nachfrage	90	70	60		
Strafe	$5 - 2 = 3$	$3 - 3 = 0$	$5 - 3 = 2$		

Höchste Strafe: A_2 und N_1 (Strafe $= 3$). Wähle A_2: günstigste Zelle $(2, 2)$ mit $c = 3$.

$x_{22} = \min(80, 70) = 70$. $a_2 = 10$, Column N_2 erschöpft.

Iterationen 3–5: (analog, hier zusammengefasst)

- $x_{11} = \min(40, 90) = 40$. Row A_1 erschöpft.
- $x_{34} = \min(100, 60) = 60$. Column N_4 erschöpft.
- $x_{31} = \min(40, 50) = 40$. Row A_3 erschöpft.
- $x_{21} = 10$. Fertig.

VAM-Tableau:

	N_1	N_2	N_3	N_4	Supply
A_1	**40** (2)	(3)	**80** (1)	(5)	120
A_2	**10** (7)	**70** (3)	(4)	(6)	80
A_3	**40** (5)	(4)	(7)	**60** (3)	100
Nachfrage	90	70	80	60	

Total cost (VAM): $40 \cdot 2 + 80 \cdot 1 + 10 \cdot 7 + 70 \cdot 3 + 40 \cdot 5 + 60 \cdot 3 = \mathbf{820}$

Hier liefert VAM zufällig dieselbe Solution wie die Minimum Cost-Method.

7.5 MODI-Optimierung der North-West Corner-Solution

Wir optimieren die North-West Corner-Initial solution (Cost: 1010).

Basis: $x_{11} = 90, x_{12} = 30, x_{22} = 40, x_{23} = 40, x_{33} = 40, x_{34} = 60$ $(m + n - 1 = 3 + 4 - 1 = 6$ Basic variablen ✓)

7.5.1 MODI-Iteration 1

Step 1 – Potentiale: Setze $u_1 = 0$.

$$x_{11} : c_{11} = u_1 + v_1 \Rightarrow 2 = 0 + v_1 \Rightarrow v_1 = 2$$
$$x_{12} : c_{12} = u_1 + v_2 \Rightarrow 3 = 0 + v_2 \Rightarrow v_2 = 3$$
$$x_{22} : c_{22} = u_2 + v_2 \Rightarrow 3 = u_2 + 3 \Rightarrow u_2 = 0$$
$$x_{23} : c_{23} = u_2 + v_3 \Rightarrow 4 = 0 + v_3 \Rightarrow v_3 = 4$$
$$x_{33} : c_{33} = u_3 + v_3 \Rightarrow 7 = u_3 + 4 \Rightarrow u_3 = 3$$
$$x_{34} : c_{34} = u_3 + v_4 \Rightarrow 3 = 3 + v_4 \Rightarrow v_4 = 0$$

Step 2 – Reduzierte Cost $\delta_{ij} = c_{ij} - u_i - v_j$:

	N_1 $(v=2)$	N_2 $(v=3)$	N_3 $(v=4)$	N_4 $(v=0)$
A_1 $(u=0)$	Basis	Basis	$1-0-4=\mathbf{-3}$	$5-0-0=5$
A_2 $(u=0)$	$7-0-2=5$	Basis	Basis	$6-0-0=6$
A_3 $(u=3)$	$5-3-2=0$	$4-3-3=\mathbf{-2}$	Basis	Basis

Negativste reduzierte Cost: $\delta_{13} = -3 \Rightarrow$ Zelle $(1,3)$ tritt ein.

Step 3 – Loop für $(1,3)$**:**

$$\underbrace{(1,3)}_{+} \rightarrow \underbrace{(1,2)}_{-} \rightarrow \underbrace{(2,2)}_{+} \rightarrow \underbrace{(2,3)}_{-} \rightarrow (1,3)$$

$(-)$-Positionen: $x_{12} = 30$, $x_{23} = 40$. $\theta = \min(30, 40) = 30$.

Costreduktion: $\Delta K = \delta_{13} \cdot \theta = (-3) \cdot 30 = -90$.

Aktualisierung: $x_{13} = 0 + 30 = 30$, $x_{12} = 30 - 30 = 0$ (verlässt Basis), $x_{22} = 40 + 30 = 70$, $x_{23} = 40 - 30 = 10$.

Neue Cost: $1010 + (-90) = \mathbf{920}$

Tableau nach Iteration 1:

	N_1	N_2	N_3	N_4	Supply
A_1	**90** (2)	(3)	**30** (1)	(5)	120
A_2	(7)	**70** (3)	**10** (4)	(6)	80
A_3	(5)	(4)	**40** (7)	**60** (3)	100
Nachfrage	90	70	80	60	

7.5.2 MODI-Iteration 2

Step 1 – Potentiale: $u_1 = 0$, $v_1 = 2$, $v_3 = 1$, $u_2 = 3$, $v_2 = 0$, $u_3 = 6$, $v_4 = -3$.

Step 2 – Reduzierte Cost:

	N_1 $(v=2)$	N_2 $(v=0)$	N_3 $(v=1)$	N_4 $(v=-3)$
A_1 $(u=0)$	Basis	$3-0-0=3$	Basis	$5-0-(-3)=8$
A_2 $(u=3)$	$7-3-2=2$	Basis	Basis	$6-3-(-3)=6$
A_3 $(u=6)$	$5-6-2=\mathbf{-3}$	$4-6-0=\mathbf{-2}$	Basis	Basis

Negativste: $\delta_{31} = -3 \Rightarrow$ Zelle $(3,1)$ tritt ein.

Step 3 – Loop:

$$\underbrace{(3,1)}_{+} \to \underbrace{(3,3)}_{-} \to \underbrace{(1,3)}_{+} \to \underbrace{(1,1)}_{-} \to (3,1)$$

$(-)$-Positionen: $x_{33} = 40$, $x_{11} = 90$. $\theta = \min(40, 90) = 40$.

$\Delta K = (-3) \cdot 40 = -120$.

Aktualisierung: $x_{31} = 40$, $x_{33} = 0$ (verlässt), $x_{13} = 70$, $x_{11} = 50$.

Neue Cost: $920 - 120 = \mathbf{800}$

7.5.3 MODI-Iteration 3: Optimalitätsprüfung

Potentiale: $u_1 = 0$, $v_1 = 2$, $v_3 = 1$, $u_2 = 3$, $v_2 = 0$, $u_3 = -1$, $v_4 = 4$.

Reduzierte Cost:

	N_1 $(v=2)$	N_2 $(v=0)$	N_3 $(v=1)$	N_4 $(v=4)$
A_1 $(u=0)$	Basis	$3-0-0=3$	Basis	$5-0-4=1$
A_2 $(u=3)$	$7-3-2=2$	Basis	Basis	$6-3-4=-1$??
A_3 $(u=-1)$	Basis	$4-(-1)-0=5$	$7-(-1)-1=7$	Basis

Neuberechnung: $u_3 = c_{31} - v_1 = 5-2 = 3$... Korrektur der Potentiale nach Iteration 2:

Basis: $x_{11} = 50$, $x_{13} = 70$, $x_{22} = 70$, $x_{23} = 10$, $x_{31} = 40$, $x_{34} = 60$.

$u_1 = 0,\ v_1 = 2,\ v_3 = 1,\ u_2 = 3,\ v_2 = 0,\ u_3 = c_{31} - v_1 = 5-2 = 3,\ v_4 = c_{34} - u_3 = 3-3 = 0$

Reduzierte Cost (korrekt):

	N_1 $(v=2)$	N_2 $(v=0)$	N_3 $(v=1)$	N_4 $(v=0)$
A_1 $(u=0)$	Basis	$3-0-0=3$	Basis	$5-0-0=5$
A_2 $(u=3)$	$7-3-2=2$	Basis	Basis	$6-3-0=3$
A_3 $(u=3)$	Basis	$4-3-0=1$	$7-3-1=3$	Basis

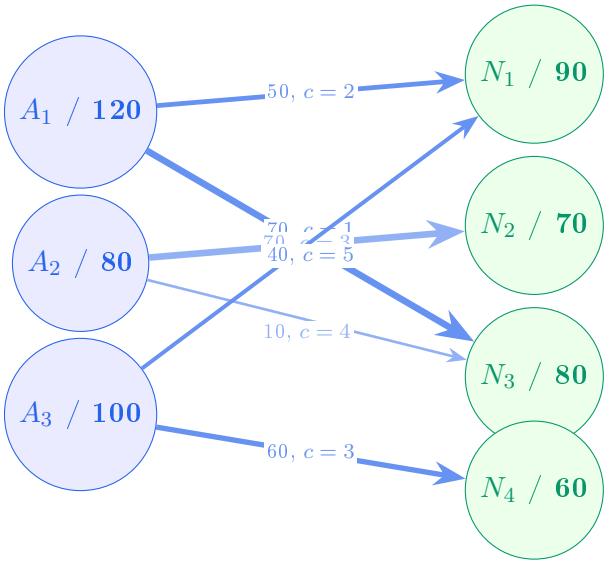
Alle $\delta_{ij} \geq 0 \Rightarrow$ Optimal solution erreicht!

7.6 Optimale Solution

	N_1	N_2	N_3	N_4	Supply
A_1	**50** (2)	(3)	**70** (1)	(5)	120
A_2	(7)	**70** (3)	**10** (4)	(6)	80
A_3	**40** (5)	(4)	(7)	**60** (3)	100
Nachfrage	90	70	80	60	

$K^* = 50 \cdot 2 + 70 \cdot 1 + 70 \cdot 3 + 10 \cdot 4 + 40 \cdot 5 + 60 \cdot 3 = 100 + 70 + 210 + 40 + 200 + 180 = \boxed{800}$

7.7 Transportnetzwerk



7.8 Methodnvergleich

Method	Startkosten	Abstand vom Optimum
North-West Corner	1010	+26,3%
Minimum Cost	820	+2,5%
Vogel Approximation	820	+2,5%
Optimum (MODI)	**800**	**0%**

8 Tool Description

Transport-Tool: Alle vier Methodn

Das Tool löst Transportation probleme mit bis zu 8×8 Zellen interaktiv und zeigt alle Stepe.

8.1 Eingabe

- **Größe:** Anzahl Supplier (2–8) und Consumer (2–8) einstellen.
- **Costmatrix:** Transport cost c_{ij} direkt in das Tableau eingeben.
- **Supply/Nachfrage:** Letzte Column (Supply a_i) und letzte Row (Nachfrage b_j).
- **Beispiele:** Zwei vorgeladene Beispiele und Zufallsgenerator verfügbar.
- **Ausgleich:** Unbalancierte Probleme werden automatisch durch fiktive Row/Column (Cost 0) balanced.

8.2 Methodn und Ausgabe

- **Initial solution berechnen:** Liefert die Initial solution der gewählten Method mit vollständigem Allocationsprotokoll (jeder Step einzeln mit Begründung).
- **Optimal solution (MODI):** Führt alle MODI-Iterationen durch. Jede Iteration zeigt:
  - Step 1: Potentiale u_i, v_j (mit Computation)
  - Step 2: Reduzierte Cost δ_{ij} for all Nicht-Basiszellen
  - Step 3: Loop mit [+]/[−]-Markierungen, θ-Wert, Vorher/Nachher der Zellen
- **Tableau-Farbcodierung:**
  - **Grün** (Basiszellen): aktuelle Allocation
  - **Orange** (eintretende Zelle): wird in Basis aufgenommen
  - **Rot** (verlassende Zelle): verlässt die Basis
  - Kleine Zahl oben rechts: Transport cost c_{ij}
  - δ-Werte: in Nicht-Basiszellen eingeblendet
- **Transportnetzwerk:** SVG-Visualisierung mit Linienbreite proportional zur Transport quantity. Supplier blau links, Consumer grün rechts.

9 Summary

Kernpunkte: Transportation problem

1. Das Transportation problem sucht die **kostengünstigste Verteilung** eines Gutes von m Suppliern zu n Consumern.
2. Eine balancede Basic solution hat genau $m + n - 1$ **Basic variablen**.
3. Die **North-West Corner-Method** ignoriert Cost – einfach, aber oft weit vom Optimum.
4. Die **Minimum Cost-Method** wählt stets die günstigste Zelle – bessere Initial solution.
5. Die **Vogel Approximation** verwendet Strafkosten – beste Initial solution der drei Methodn.
6. Die **MODI-Method** optimiert iterativ über reduzierte Cost und Loopntausch.
7. **Optimalitätskriterium:** Alle reduzierten Cost $\delta_{ij} \geq 0$.
8. Jede MODI-Iteration reduziert die Cost um $|\delta_{i^*j^*}| \cdot \theta$.
9. Bei ganzzahligen Supplys- und Nachfragemengen ist die Optimal solution ebenfalls ganzzahlig (totale Unimodularität).