

Standortplanung und strategisches Supply Chain Management

Ecko Tan

21. Februar 2018

Don't panic!

Inhaltsverzeichnis

1 Volkswirtschaftliche und deskriptive Standortmodelle	7
1.1 Volkswirtschaftliche Standortmodelle	7
1.1.1 Die Wahl kostenminimaler Wohnstandorte	7
1.1.2 Theorie der Boden- und Flächennutzung	8
1.2 Deskriptive überbetriebliche Standortmodelle	9
1.2.1 Prüflisten-Verfahren	9
1.2.2 Rangfolge-Verfahren	10
1.2.3 Vergleich	11
1.3 Standortplanung unter Wettbewerb	11
1.3.1 Modelle mit vollständiger Zuordnung	11
1.3.1.1 Voroni-diagramm	11
1.3.2 Modelle mit partieller Aufteilung	11
1.3.2.1 Gravitions-Modelle (Das Modell von Huff)	12
1.3.3 Leader-Follower-Modelle	13
1.3.3.1 Regeln	13
1.3.3.2 Probleme / Herausforderungen	13
1.3.3.3 Typische Strategien für Follower F	13
1.3.3.4 Verschiedene Strategien für Leader L	14
2 Standortplanung in der Ebene	17
2.1 Theorie der Standortplanung	17
2.2 Begriffe und Notationen	17
2.2.1 Distanzmessung	17
2.2.2 l_p -Metrik	18
2.3 1-Medianprobleme	18
2.3.1 Einleitung	18
2.3.1.1 Aufgabe	18
2.3.1.2 Dominanzkriterium	18
2.3.2 1-Medianprobleme mit l_1 -Metrik	19
2.3.2.1 Das 1-Medianproblem mit l_1 -Metrik auf der Linie	19
2.3.3 1-Medianprobleme mit l_2 -Metrik	20
2.3.4 1-Medianprobleme mit l_∞ -Metrik	21
2.4 1-Centerprobleme	21
2.4.1 Aufgabe	21
2.4.2 formel	22
2.4.3 1-Centerprobleme mit l_1 -Metrik	22
2.4.3.1 1-Centerprobleme mit l_∞ -Metrik	22
2.4.3.1.1 Der ungewichtete Fall	22

2.4.3.1.2 Der allgemeine geweichtete Fall	23
2.4.4 1-Centerprobleme mit l_2 -Metrik	25
2.4.4.1 Zielfunktion	25
2.4.4.2 Der ungewichtete Fall: Das Kreis-Überdeckungsproblem	25
2.5 Mehrstandortprobleme	26
2.5.1 Modelle mit Interaktion	27
2.5.1.1 Annahmen	27
2.5.1.2 Zielfunktion	27
2.5.1.3 Notationen	27
2.5.1.4 Das Interaktions-Modell mit l_1 -Metrik	27
2.5.1.5 Das Interaktions-Modell mit l_2 -Metrik	28
2.6 Zuordnungs-Modelle	28
2.6.1 Das p-Median Zuordnungsproblem	29
2.6.1.1 Das p-Median Zuordnungsproblem mit l_1 -Metrik	29
2.6.1.2 Das Zuordnungsproblem mit l_2 -Metrik	30
3 Standortplanung auf Netzwerken	31
3.1 Graphentheorie	31
3.2 1-Medianprobleme	32
3.2.1 1-Medianprobleme auf allgemeinen Graphen	33
3.2.1.1 Knotenbeschränktes 1-Medianproblem	33
3.2.1.2 Absolutes 1-Medianproblem	33
3.2.2 1-Medianprobleme auf Bäume	34
3.3 1-Centerprobleme	35
3.3.1 1-Centerprobleme auf allgemeinen Graphen	35
3.3.2 1-Centerprobleme auf Bäume	36
3.3.2.1 Das ungewichtete 1-Centerproblem	36
3.4 Mehrstandortprobleme	37
4 Diskrete Standortplanung	39
4.1 Klassifikation diskreter Standortprobleme	39
4.2 Das Warehouse Location Problem (WLP)	40
4.2.1 Modellierung	40
4.2.2 Heuristiken	41
4.2.2.1 Greedy-Heuristik	41
4.2.2.2 Interchange-Heuristik	43
4.2.3 Das DUALOC-Verfahren	43
4.2.4 Dual Ascent-Verfahren	44
4.2.5 Konstruktionsheuristik	48
4.2.5.1 Dual Adjustment-Verfahren	49
4.3 Hub-Location-Probleme	52
5 Gebietsplanung	53
5.1 Basismodell	53
5.1.1 Definitionen und Notationen	53
5.1.2 Modell-Kriterien	53
5.1.3 Ziel der Gebietsplanung	55
5.2 Vorgehensweisen zur Gebietsplanung	55
5.2.1 Notation	55

5.2.2	LP-Formulierung	56
5.3	Recursive-Partitioning-Algorithmus	56
5.3.1	Definitionen	56
5.3.2	Recursive-Partitioning	56
5.3.3	Algorithmus	57

Kapitel 1

Volkswirtschaftliche und deskriptive Standortmodelle

1.1 Volkswirtschaftliche Standortmodelle

1.1.1 Die Wahl kostenminimaler Wohnstandorte standortbezogenen Kosten

$$C(d) = Q \cdot p(d) + V \cdot k(d) \quad (1.1)$$

- d : die Entfernung des Standortes zum Stadtzentrum
- $C(d)$: die gesamten Standortkosten
- Q : die vorgegebene Größe der Wohnfläche
- $p(d)$: die standortabhängigen Mietkosten pro Flächeneinheit (z.B. Quadratmeter)
- V : die Anzahl der Fahrten ins Zentrum
- $k(d)$: standortabhängige Fahrtkosten

Annahme

- $p(d)$ nehmen mit zunehmender Entfernung zum Stadtzentrum exponentiell ab:

$$p(d) = P_Z \cdot e^{-rd} \quad (1.2)$$

- P_Z : der Mietpreis direkt im Zentrum
- r : Verfallskonstante für die Entfernung
- die Fahrtkosten sind proportional zur Entfernung zum Zentrum

$$k(d) = K \cdot d \quad (1.3)$$

- K : die Kosten pro Entfernungseinheit

Den kostenminimalen Standort findet man unter diesen Annahmen durch Minimierung der Funktion $C(d)$.

Ableiten der Funktion C nach d und anschließendem “Nullsetzen” der Ableitung \Rightarrow

$$d^* = \frac{1}{r} (\ln[r \cdot P_Z \cdot Q] - \ln[V \cdot K]) \quad (1.4)$$

Example 1.1.1. [Aufgabe 1](#), [Aufgabe 2](#)

1.1.2 Theorie der Boden- und Flächennutzung

von Thünens Modell

- das zu untersuchende Gebiet ist eine kreisförmige, isolierte Fläche gleichmäßiger Produktivität
- es liegt ein einziger Absatzmarkt im Zentrum der Fläche vor
- die Transportverbindungen sind überall gleichmäßig gut
- die Produktion der Produkte ist überall und zu gleichen Kosten möglich
- die Transportkosten steigen proportional zur Entfernung
- es ist eine Menge von Produktions-Aktivitäten und deren Output-Mengen gegeben

Unter dem Gesichtspunkt der Gewinn-Maximierung stellt sich nun die Frage:

“Welches der Produkte soll in welcher Entfernung vom Absatzmarkt hergestellt werden?”

Algorithm 1 Verfahren (zur Bestimmung der oberen Einhüllenden)

1: Stelle für jede Produktions-Aktivität die zugehörige Profitfunktion auf:

$$R(d) = \max_i R_i(d) = \max_i (p_i - k_i d)$$

- 2: Setze $d' = 0$ und $I = 1, \dots, n, L = \emptyset$
- 3: Bestimme der max. Profit $R_{i^*} = \max_{j=1, \dots, n} R_j(0)$ und damit die im Zentrum profitabelste Aktivität P_{i^*}
- 4: **if** (das Maximum ist nicht eindeutig) **then**
- 5: wähle unter den maximalen Aktivitäten diejenige mit dem kleinsten k_j .
- 6: **end if**
- 7: $I = I \setminus \{i^*\}, L = L \cup \{i^*\}$
- 8: **while** (Es gibt Schnittpunkte ($d_{ij} = \frac{P_i - P_j}{k_i - k_j}$) $\wedge (I \neq \emptyset)$) **do**
- 9: Schneide R_{i^*} mit allen $R_j(\cdot), j \in I$
- 10: Bestimme $S_{i^*j} = (d_{i^*j}, R_{i^* \setminus j}(d_{i^*j}))$ als den Schnittpunkt mit dem kleinsten Wert $d_{i^*j} > d'$
- 11: **if** (Schneiden sich mehrere Profitfunktionen bei d_{i^*j}) **then**
- 12: wähle unter den Aktivitäten diejenige mit dem kleinsten k .
- 13: **end if**
- 14: $d' = d_{i^*j}, i^* = j, I = I \setminus \{i^*\}$
- 15: Füge d und i^* am Ende der Liste L an
- 16: **end while**
- 17: Füge d^{\max} am Ende von L ein

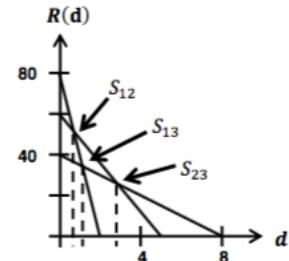
Output: Die Liste L enthält dann die gesuchten Entfernungsintervalle und die zugehörigen, profitablen Produktionsaktivitäten.

Example 1.1.2.**Beispiel**

Gegeben seien drei Produktions-Aktivitäten für Produkte P_1, P_2 und P_3 mit den folgenden Charakteristika

- $p_1 = 80, k_1 = 40 \quad d_1^{\max} = 2$
- $p_2 = 60, k_2 = 12 \quad d_2^{\max} = 5$
- $p_3 = 40, k_3 = 5 \quad d_3^{\max} = 8$

Wir haben $I = \{1,2,3\}$ und $d' = 0$.



Im Zentrum ist Aktivität P_1 am profitabelsten, d.h. $i^* = 1$.

Setze $I = I \setminus \{1\} = \{2,3\}$ und $L = \{1\}$.

Schneide R_1 mit R_2 und R_3 . Man erhält: $S_{12} = \left(\frac{5}{7}, \frac{360}{7}\right)$ und $S_{13} = \left(\frac{8}{7}, \frac{240}{7}\right)$

Da $d' < \frac{5}{7} < \frac{8}{7}$, haben wir $i^* = j = 2$, $d' = \frac{5}{7}$, $I = I \setminus \{2\} = \{3\}$ und $L = \left\{1, \frac{5}{7}, 2\right\}$.

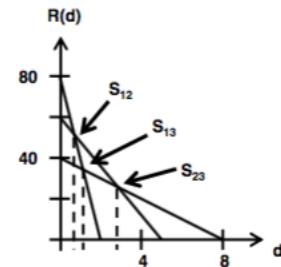
Schneide nun R_2 mit R_3 . Man erhält: $S_{23} = \left(\frac{20}{7}, \frac{180}{7}\right)$

Da $d' < \frac{20}{7}$, haben wir $i^* = j = 3$, $d' = \frac{20}{7}$, $I = I \setminus \{3\} = \emptyset$ und $L = \left\{1, \frac{5}{7}, 2, \frac{20}{7}, 3\right\}$.

Da $I = \emptyset$, füge $d^{\max} = 8$ in L und beende das Verfahren.

Wir erhalten: $L = \left\{1, \frac{5}{7}, 2, \frac{20}{7}, 3, 8\right\}$.

Damit ist im Entfernungsbereich $\left[0, \frac{5}{7}\right]$ Produkt 1 das profitabelste. Danach erzielt man mit Produkt 2 den größten Gewinn im Bereich $\left[\frac{5}{7}, \frac{20}{7}\right]$, und Produkt 3 ist schließlich im Bereich $\left[\frac{20}{7}, 8\right]$ am profitabelsten.

**Example 1.1.3. Aufgabe 3**

1.2 Deskriptive überbetriebliche Standortmodelle

Sowohl Prüflisten- als auch Rangfolge-Verfahren bestimmen den besten Standort für eine neue Einrichtung basierend auf Standortfaktoren.

1.2.1 Prüflisten-Verfahren

Vorgehensweise:

- Zusammenstellung der dem jeweiligen Problem angemessenen Standortfaktoren. Bei dieser Aufstellung bedarf es großer Umsicht.
- Für jeden, der als relevant erachteten Standortfaktoren, ist zu überprüfen und entsprechend zu kennzeichnen, z. B. durch ein Kreuz, an welchem Standort der Faktor künftig am besten erfüllt wird.

Bei dieser Vorgehensweise werden einige Standorte mehrere Kennzeichnungen auf sich vereinen und deshalb als besonders günstig erscheinen, während andere möglicherweise leer ausgehen.

Note 1. Wähle den Standort mit meisten Kreuze

Example 1.2.1.

Beispiel

Gegeben seien 6 Standorte, A – F, und 6 als relevant erkannte Faktoren (I – VI) zusammen mit den entsprechenden Markierungen, welche den höchsten Erfüllungsgrad unter allen Alternativen kennzeichnen.

Standort	relevante Standortfaktoren					
	I	II	III	IV	V	VI
A		X				
B	X			X		
C						X
D						
E					X	
F			X			

← Standort B ist der am besten geeignete

Abbildung 1.1: Prüflisten-Verfahren Bsp

1.2.2 Rangfolge-Verfahren

Vorgehensweise:

- Zusammenfassung der dem Problem angemessenen Standortfaktoren
- Ermittlung einer Gewichtungszahl für jeden Standortfaktoren
- Finde die höchste Gesamtwertigkeit

Example 1.2.2.

Beispiel

Gegeben seien 6 Standorte (A – F) sowie 2 Standortfaktoren (I und II) zusammen mit den entsprechenden **Gewichtungszahlen** (GZ), **Rangwerten** (RW) und den daraus resultierenden **Wertigkeitsziffern** (WZ).

Standort	Standortfaktor I			Standortfaktor II			Gesamt-wertigkeit
	RW	GZ	WZ	RW	GZ	WZ	
A	2	0.3	0.6	4	0.7	2.8	3.4
B	1	0.3	0.3	2	0.7	1.4	1.7
C	3	0.3	0.9	1	0.7	0.7	1.6
D	4	0.3	1.2	6	0.7	4.2	5.4
E	5	0.3	1.5	5	0.7	3.5	5.0
F	6	0.3	1.8	3	0.7	2.1	3.9

Standort D, welcher die **höchste Gesamtwertigkeit** aufweist, ist als optimal zu bezeichnen.

Abbildung 1.2: Rangfolge-Verfahren Bsp

1.2.3 Vergleich

1.3 Standortplanung unter Wettbewerb

Ziel: eine oder mehrere neue Einrichtung so zu platzieren, dass der Marktanteil und Gewinn der eigenen Firma maximiert wird.

1.3.1 Modelle mit vollständiger Zuordnung

Der Bedarf jedes Kunden wird von genau einer Einrichtung vollständig befriedigt. (Bei diesen Modellen lässt sich für jeden Kunden eindeutig entscheiden, welche Einrichtung für ihn die Beste oder Attraktivste ist.)

Definition 1.3.1. Indifferenzmenge

die Menge aller Punkte, welche gleichweit von den beiden Standorten x_A und x_B entfernt liegen

$$IS_{AB} := \{x \in \mathbb{R} \mid l_2(x; x_A) = l_2(x; x_B)\} \quad (1.5)$$

1.3.1.1 Voroni-diagramm

Example 1.3.1. Aufgabe 5

1.3.2 Modelle mit partieller Aufteilung

Bei diesen Modellen verteilen die Kunden ihre Nachfrage anteilig auf mehrere Einrichtungen.

1.3.2.1 Gravitions-Modelle (Das Modell von Huff)

Definition 1.3.2. Anziehungskraft der Einrichtung E auf den Kunden K

$$A(E, K) = \frac{w_E}{d(K, E)^r} \quad (1.6)$$

- w_E : Größe der Einrichtung E an diesem Standort
- $d(K, E)$: die Entfernung vom Kunden K zum Standort der Einrichtung E
- r : eine Potenz (Verfallskonstante) für die Entfernung

Die Wahrscheinlichkeit P , dass ein Kunde eine Einrichtung an einem bestimmten Standort besucht, berechnet sich nach diesem Modell folgendermaßen:

$$P(E, K) = \frac{A(E, K)}{\sum_{\text{alle } \bar{E}} A(\bar{E}, K)} \quad (1.7)$$

Definition 1.3.3. Indifferenzmege zweier Einrichtungen

die Menge aller Punkte, welche beide Einrichtungen mit derselben Wahrscheinlichkeit aufsuchen.

Der Anteil der Nachfrage eines Kunden, welcher auf eine bestimmte Einrichtung entfällt, ergibt sich aus dem Produkt der Gesamtnachfrage des Kunden mit der „Besuchs“-Wahrscheinlichkeit

$$D(E, K) = D(K) \cdot P(K, E) \quad (1.8)$$

- $D(K)$: Gesamtnachfrage des Kunden

Gesamtnachfrage einer Einrichtung = Summe aller Anteile von Kundennachfragen

$$D(E) = \sum_{\text{alle } K} D(E, K) \quad (1.9)$$

gesamter Marktanteil der Einrichtung:

$$MS(E) = \frac{D(E)}{\sum_{\text{alle } K} D(K)} \quad (1.10)$$

Kriterium für Standort einer neuen Einrichtung \bar{E} :

$$\underset{\bar{E}}{\operatorname{argmax}} \quad MS(\text{einige Firma}) = \frac{D(\bar{E}) + \sum_{\text{eigene, bereits exist. } E} D(E)}{\sum_{\text{alle } K} D(K)}$$

Zusammenfassung

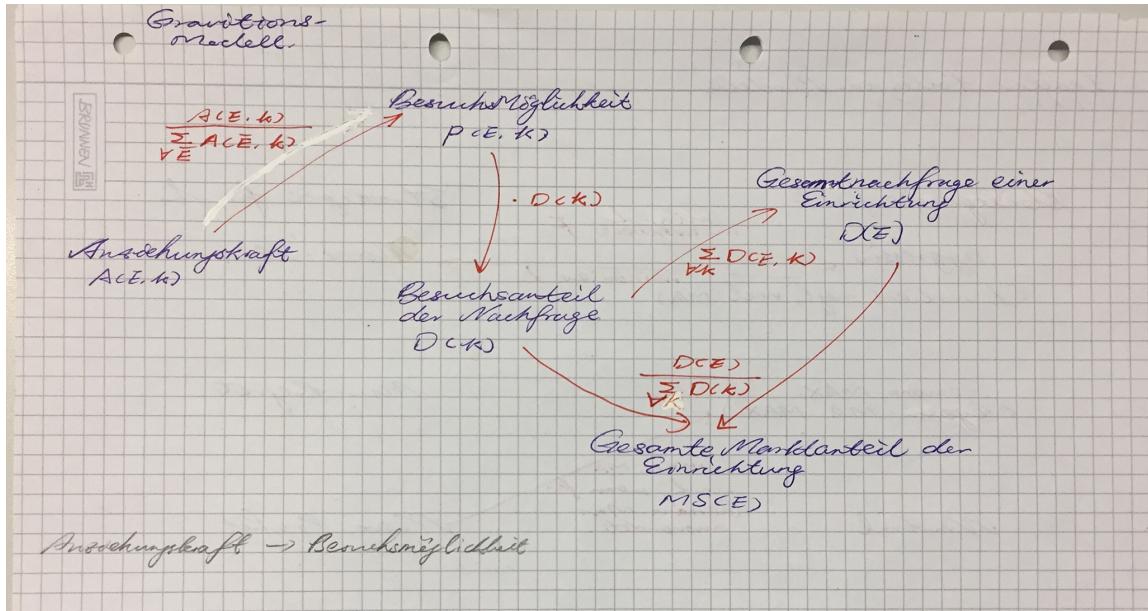


Abbildung 1.3: Gravitationsmodelle

Example 1.3.2. Aufgabe 4

1.3.3 Leader-Follower-Modelle

1.3.3.1 Regeln

- Bei diesen Modellen platzieren zwei Wettbewerber nacheinander neue Einrichtungen in einem (bisher unerschlossenen) Markt.
- Dabei wählt zuerst der Leader (Erstplatzierende) L Standorte für all seine neuen Einrichtungen und danach der Follower (Zweitplatzierende) F .
- Standortwahl von L und F beeinflussen sich gegenseitig.

1.3.3.2 Probleme / Herausforderungen

- **Follower F**

Braucht sich erst auf eine Platzierungsstrategie festlegen, wenn der Erstplatzierende L seine Standortwahl schon getroffen hat.

- **Leader L**

Hat das Problem, dass er die Platzierungsstrategie von F nicht kennt.

1.3.3.3 Typische Strategien für Follower F

- **Aggressiv**

Platziere neue Einrichtungen so, dass L möglichst viel Marktanteil verliert.

- **Gewinnmaximierend**

Platziere neue Einrichtungen so, dass der eigene Marktanteil maximal wird.

- **Neutral**

Platziere neue Einrichtungen nach anderen Kriterien.

1.3.3.4 Verschiedene Strategien für Leader L

- **Maxi-Min**

L platziert neue Einrichtungen so, dass sein Marktanteil maximal wird, wenn F eine aggressive Strategie anwendet, d. h. F das Ziel hat den Marktanteil von L zu minimieren.

Rechenweg:

1. Bestimme zu jedem L denjenigen Follower-Standort \tilde{f} , der dem Leader den geringsten Gewinn bringt, wenn der Leader bei l platziert.
2. Berechne den daraus resultierenden Marktanteil des Leaders
3. Wähle den Standort, für den der Marktanteil des Leaders maximal wird

- **Min-Regret (minimales Bedauern)**

L platziert neue Einrichtungen so, dass, egal welche Standortentscheidung F dann trifft, der Zugewinn an Marktanteil für L durch eine mögliche Andersplatzierung seiner Einrichtungen (nachdem F seine Wahl getroffen hat) minimal wird.

- **Max-Profit**

L platziert neue Einrichtungen so, dass ein Marktanteil maximal wird, wenn F ebenfalls eine gewinnmaximierende Strategie anwendet.

Rechenweg:

1. Bestimme zu jedem L denjenigen Follower-Standort \tilde{f} , der dem Follower den größten Gewinn bringt, wenn der Leader bei l platziert.
2. Berechne den daraus resultierenden Marktanteil des Leaders
3. Wähle den Standort, für den der Marktanteil des Leaders maximal wird

Zusammenfassung

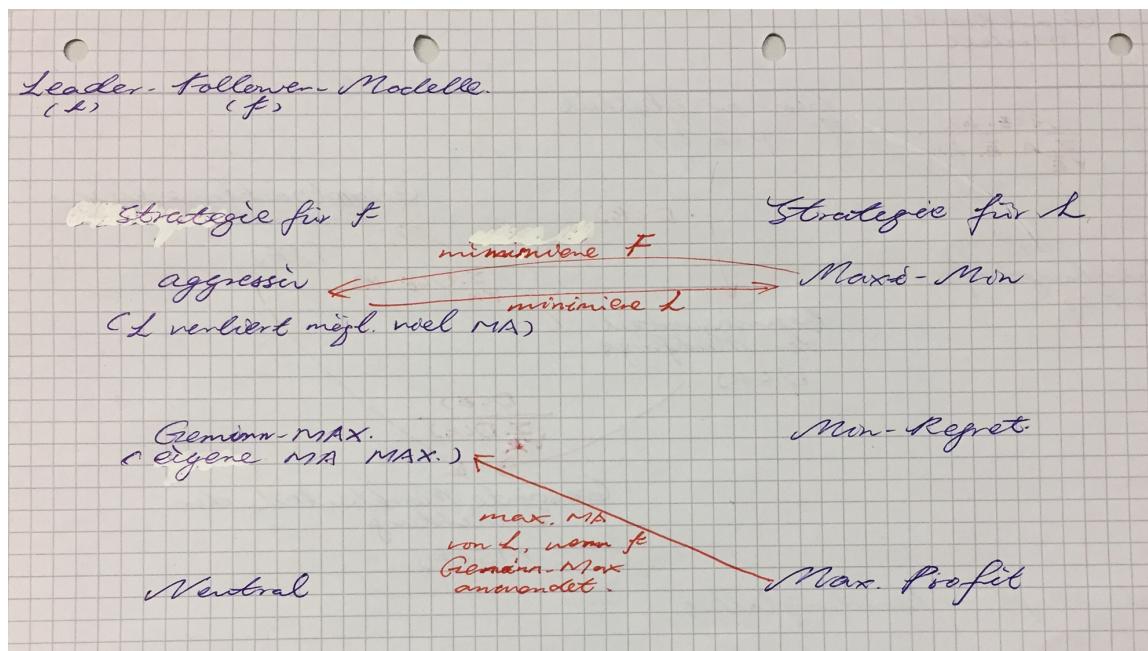


Abbildung 1.4: Leader-Follower-Modelle

Example 1.3.3. Aufgabe 6

Kapitel 2

Standortplanung in der Ebene

2.1 Theorie der Standortplanung

Grundlegende Aufgabenstellung

Platzierung ein oder mehrerer neuer Einrichtungen in Abhängigkeit bereits existierender Einrichtungen

Drei Klassen von Standortproblemen:

- **Standortprobleme in der Ebene** (Einfachste, allerdings auch unrealistischste Problemklasse)
- **Standortprobleme auf Netzwerken** (praxisbezug viel ausgeprägter als bei planaren Standortproblemen)
- **Diskrete Standortprobleme** (realistischste und flexibelste Klasse von Standortproblemen)

2.2 Begriffe und Notationen

Definition 2.2.1. Standorte als Punkte idealisiert

Definition 2.2.2. Neue Einrichtungen $X = x_1, \dots, x_p, p \geq 1$

- dürfen überall in der Ebene platziert werden
- ihre Standorte werden durch Koordinaten repräsentiert

$$x_i = (x_{i,1}, x_{i,2}) \in \mathbb{R}^2, \forall i = 1, \dots, p$$

Definition 2.2.3. Existierende Einrichtungen (Kunden) $A = a_1, \dots, a_n$

- werden durch eine Menge von Punkten in der Ebene repräsentiert

$$a_i = (a_{i,1}, a_{i,2}) \in \mathbb{R}^2, \forall i = 1, \dots, n$$

- Jedem Kunden $a_i \in A$ wird ein positives Gewicht $w_i > 0$ zugeordnet.

Definition 2.2.4. Kosten

Lediglich Transportkosten werden berücksichtigt. Sie sind proportional zur Menge und zur zurückgelegten Entfernung.

2.2.1 Distanzmessung

Definition 2.2.5. Metrik / Distanzfunktion $d : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$

welche für alle $x, y, z \in \mathbb{R}^n, n \geq 1$ folgende Eigenschaften erfüllt:

- Definitheit: $d(x, y) \geq 0$ und $d(x, y) = 0 \Leftrightarrow x = y$
- Symmetrie: $d(x, y) = d(y, x)$
- Dreiecks-Ungleichung: $d(x, y) \leq d(x, z) + d(z, y)$

2.2.2 l_p -Metrik

$$l_p(x, y) = \sqrt[p]{|x_1 - y_1|^p + |x_2 - y_2|^p}, p \geq 1 \quad (2.1)$$

Rechtwinklige Entfernung (l_1 – oder Manhatten-Metrik)

$$l_1(x, y) = |x_1 - y_1| + |x_2 - y_2| \quad (2.2)$$

Euklidische Entfernung (l_2 – oder Luftlinien-Metrik)

$$l_2(x, y) = \sqrt{|x_1 - y_1|^2 + |x_2 - y_2|^2} \quad (2.3)$$

Quadrierte euklidische Entfernung (l_2^2 – Metrik)

$$l_2(x, y) = |x_1 - y_1|^2 + |x_2 - y_2|^2 \quad (2.4)$$

Tchebychev Entfernung (l_∞ -Metrik)

$$l_\infty(x, y) = \max\{|x_1 - y_1|, |x_2 - y_2|\} \quad (2.5)$$

2.3 1-Medianprobleme

2.3.1 Einleitung

2.3.1.1 Aufgabe

Platziere eine neue Einrichtung so in der Ebene, dass die Summe der gewichteten Entfernungen von dem neuen Standort zu allen Kunden minimal wird.

$$\min_{x \in \mathbb{R}^2} f(x) := \sum_{i=1}^n w_i d(x, a_i) \quad (2.6)$$

- $A = a_1, \dots, a_n$: die n Kunden
- $w_i > 0, i = 1, \dots, n$: die damit assoziierten Gewichte
- x : der gesuchte Standort der neuen Einrichtung
- d : eine Metrik

Nenne einen Punkt $x^* \in \mathbb{R}^2$, welcher die Funktion $f(x)$ minimiert ($f(x^*) \leq f(x), \forall x \in \mathbb{R}^2$), **optimal**. Bezeichne die Menge aller optimalen Punkte der Funktion $f(\cdot)$ mit $\mathcal{X}^*(f)$.

2.3.1.2 Dominanzkriterium

Gilt für einen Kunden a_k mit zugehörigem Gewicht w_k , dass

$$w_k \geq \frac{1}{2} \sum_{i=1}^n w_i,$$

d. h. der Kunde vereinigt mindestens die Hälfte aller Gewichte auf sich, so ist der Standort des Kunden a_k eine optimale Lösung des Problems.

2.3.2 1-Medianprobleme mit l_1 -Metrik

Aufgabe 7

Zielfunktion

$$\begin{aligned}
 f(x) &= \sum_{i=1}^n w_i l_1(x, a_i) \\
 &= \sum_{i=1}^n w_i (|x_1 - a_{i,1}| + |x_2 - a_{i,2}|) \\
 &= \underbrace{\sum_{i=1}^n w_i (|x_1 - a_{i,1}|)}_{=:f_1(x_1)} + \underbrace{\sum_{i=1}^n w_i (|x_2 - a_{i,2}|)}_{=:f_2(x_2)}
 \end{aligned} \tag{2.7}$$

→ Die ursprüngliche Aufgabe reduziert sich auf das Lösen zweier 1-dimensionaler Probleme, sogenannter “Probleme auf der Linie”.

2.3.2.1 Das 1-Medianproblem mit l_1 -Metrik auf der Linie

Zielfunktion

$$\min_{x \in \mathbb{R}} f(x) := \sum_{i=1}^n w_i |x - a_i|$$

Algorithm 2 Lösungsverfahren für das 1-Medianproblem mit l_1 -Metrik auf der Linie

- 1: Berechne die Summe W aller Gewichte $W := \sum_{i=1}^n nw_i$
- 2: **if** (Es exist. ein $w_k \geq \frac{1}{2}W$) **then**
- 3: Der Kundenstandort a_k ist eine optimale Lösung, $x^* = a_k$
- 4: Stopp
- 5: **else**
- 6: Sortiere die Kundenstandorte a_1, \dots, a_n nach monoton wachsenden Koordinaten $a_{i_1} \leq a_{i_2} \leq \dots \leq a_{i_n}$
- 7: Bestimme unter Berücksichtigung einer zu den Kundenstandorten analogen Sortierung der Gewichte dasjenige h , für welches folgendes gilt:

$$\sum_{j=1}^{h-1} w_{i_j} < \frac{1}{2}W \text{ und } \sum_{j=1}^h w_{i_j} \geq \frac{1}{2}W$$

(die Gewichte der Reihe nach so lange aufsummiert, bis diese Summe gerade eben mehr als die Hälfte der gesamten Gewichte umfasst.)

- 8: $x^* = a_{i_h}$ ist die Koordinate eines optimalen Standorts, d.h. $\mathcal{X}^*(f) = \{a_{i_h}\}$
 - 9: **end if**
-

Example 2.3.1.

1-Medianprobleme mit l_1 -Metrik

Beispiel: 2-dimensionales Problem

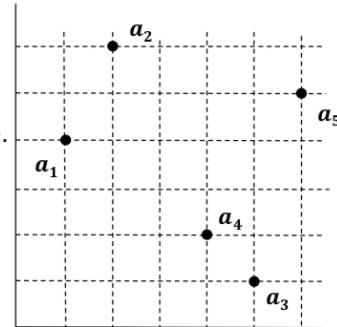
Sei $A = \{(1,4), (2,6), (5,1), (4,2), (6,5)\}$ und $w = \{2,3,1,1,2\}$.

Kundenstandorte für das erste Teilproblem:

$$a_{1,1} = 1, a_{2,1} = 2, a_{3,1} = 5, a_{4,1} = 4, a_{5,1} = 6$$

Verfahren

1. $W = 2 + 3 + 1 + 1 + 2 = 9$.
2. Dominanzkriterium ist für keinen der Kunden erfüllt.
3. Sortierung der Kundenstandorte liefert $a_{1,1} \leq a_{2,1} \leq a_{4,1} \leq a_{3,1} \leq a_{5,1}$
4. Aus 3. \Rightarrow Reihenfolge der aufzusummierenden Gewichte ist $\{w_1, w_2, w_4, w_3, w_5\} = \{2, 3, 1, 1, 2\}$.
Man erhält $w_1 = 2 < \frac{1}{2}W = 4.5$ und $w_1 + w_2 = 5 > \frac{1}{2}W$.
5. $x_1 = a_{2,1} = 2$ ist der optimale Standort für das erste Teilproblem.



1-Medianprobleme mit l_1 -Metrik

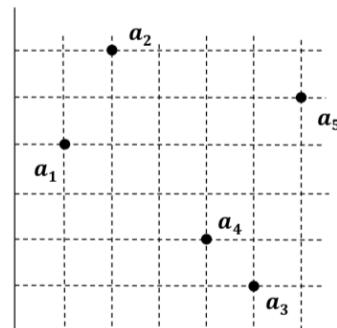
Kundenstandorte für das zweite Teilproblem:

$$a_{1,2} = 4, a_{2,2} = 6, a_{3,2} = 1, a_{4,2} = 2, a_{5,2} = 5$$

Schritte 1. und 2. gelten unverändert für das zweite Teilproblem.

Verfahren

3. Sortierung der Kundenstandorte liefert $a_{3,2} \leq a_{4,2} \leq a_{1,2} \leq a_{5,2} \leq a_{2,2}$
4. Aus 3. \Rightarrow Reihenfolge der aufzusummierenden Gewichte $\{w_3, w_4, w_1, w_5, w_2\} = \{1, 1, 2, 2, 3\}$
Man erhält $w_3 + w_4 + w_1 = 4 < \frac{1}{2}W$ und $w_3 + w_4 + w_1 + w_5 = 6 > \frac{1}{2}W$.
5. Damit ist $x_2 = a_{5,2} = 5$ der optimale Standort für das zweite Teilproblem.



Optimale Lösung des Problems $x^* = (2,5)$ mit Zielfunktionswert $f(x^*) = 27$

Abbildung 2.1: das 1 medianproblem mit metrik auf der linie

Restriktive Standortprobleme mit verbotenes Gebiet R

potentielle Standorte: Schnittpunkte der achsenparallelen Geraden durch Kunden mit verboten Gebiet

Example 2.3.2. Aufgabe 7(b)

2.3.3 1-Medianprobleme mit l_2 -Metrik

Zielfunktion

$$\begin{aligned}
f(x) &= \sum_{i=1}^n w_i l_2^2(x, a_i) \\
&= \sum_{i=1}^n w_i l_2^2((x_1 - a_{i,1})^2 + (x_2 - a_{i,2})^2) \\
&= \sum_{i=1}^n w_i l_2^2(x_1 - a_{i,1})^2 + \sum_{i=1}^n w_i l_2^2(x_2 - a_{i,2})^2 \\
&=: f_1(x_1) + f_2(x_2)
\end{aligned} \tag{2.8}$$

$f_1(x_1)$ und $f_2(x_2)$ sind differenzierbar.

Ableiten nach x_1 bzw. x_2 und Nullsetzen der jeweiligen Ableitung ergibt:

$$\underbrace{x^*}_{\text{Schwerpunkt}} = \left(\frac{\sum_{i=1}^n w_i a_{i,1}}{\sum_{i=1}^n w_i}, \frac{\sum_{i=1}^n w_i a_{i,2}}{\sum_{i=1}^n w_i} \right)$$

2.3.4 1-Medianprobleme mit l_2 -Metrik

Example 2.3.3. Aufgabe 8

Zielfunktion

$$f(x) = \sum_{i=1}^n w_i l_2(x, a_i) = \sum_{i=1}^n w_i \sqrt{(x_1 - a_{i,1})^2 + (x_2 - a_{i,2})^2} \tag{2.9}$$

Die Zielfunktion ist nicht zerlegbar!

Verschärftes Dominanzkriterium

Der Standort a_j eines Kunden ist optimal, falls

$$\gamma(a_j) := l_2\left(\sum_{i=1, i \neq j}^n w_i \frac{a_j - a_i}{l_2(a_j, a_i)}, \mathbf{0}\right) \leq w_j, \mathbf{0} = (0, 0) \tag{2.10}$$

Algorithm 3 Das Approximations-Verfahren von Weiszfeld

```

1: if (Verschärftes Dominanzkriterium  $\gamma(a_j) \leq w_j$  erfüllt) then
2:    $x^* = a_j$ , STOPP!
3: else
4:    $k := 0$ 
5:    $x^{(0)} := \frac{\sum_{i=1}^n w_i a_i}{\sum_{i=1}^n w_i}$  (Schwerpunkt)
6:   while  $(!(\delta^{(k+1)} := \frac{f(x^{(k)}) - f(x^{(k+1)})}{f(x^{(k)})} \leq \delta))$  do
7:      $x^{(k+1)} := \frac{\sum_{i=1}^n w_i \frac{a_i}{l_2(x^{(k)}, a_i)}}{\sum_{i=1}^n w_i \frac{1}{l_2(x^{(k)}, a_i)}}$ 
8:      $k := k + 1$ 
9:   end while
10: end if

```

2.4 1-Centerprobleme

2.4.1 Aufgabe

Platziere eine neue Einrichtung so in der Ebene, dass die maximale gewichtete Entfernung von dem neuen Standort zu den Kunden minimal wird.

2.4.2 formel

$$\min_{x \in \mathbb{R}^2} g(x) \text{ mit } g(x) := \max_{i=1,\dots,n} w_i d(x, a_i) \quad (2.11)$$

2.4.3 1-Centerprobleme mit l_1 -Metrik

Example 2.4.1. [Aufgabe 9](#)

Zielfunktion

$$\begin{aligned} g(x) &= \max_{i=1,\dots,n} w_i l_1(x, a_i) \\ &= \max_{i=1,\dots,n} w_i (|x_1 - a_{i,1}| + |x_2 - a_{i,2}|) \end{aligned} \quad (2.12)$$

Algorithm 4 Lösungsverfahren für das l_1 -Centerproblem

1: Transformiere alle Kundenstandorte mit

$$T = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$$

- 2: Finde eine optimale Lösung x_∞^* des l_∞ -Problems mit den Kundenstandorten A'
 3: Erhalte eine optimale Lösung x^* des l_1 -Problems:

$$x^* = x_\infty^* \cdot T^{-1}$$

2.4.3.1 1-Centerprobleme mit l_∞ -Metrik

Zielfunktion

$$\begin{aligned} g(x) &= \max_{i=1,\dots,n} w_i l_\infty(x, a_i) \\ &= \max_{i=1,\dots,n} w_i (|x_1 - a_{i,1}|, |x_2 - a_{i,2}|) \end{aligned} \quad (2.13)$$

2.4.3.1.1 Der ungewichtete Fall

Umformulierung des Problems:

$$\min_{x \in \mathbb{R}^2} \max_{i=1,\dots,n} l_\infty(x, a_i) \Leftrightarrow \min z, \text{ u.d.N } l_\infty(x, a_i) \leq z, \forall i = 1, \dots, n$$

\Rightarrow Minimiere z unter der Nebenbedingung, dass die Entfernung von einem Punkt x zu allen Kundenstandorten kleiner oder gleich z ist ($l_\infty(x, a_i) \leq z, \forall i = 1, \dots, n$).

Algorithm 5 Lösungsverfahren für ungewichtete 1-Centreprobleme mit l_∞ -Metrik

-
- 1: Berechne für die Kundenstandorte das umschreibende Rechteck R . R ist durch zwei gegenüberliegende Eckpunkte (ul, or) eindeutig bestimmt.
 - 2: **if** (R ist ein Quadrat) **then**
 - 3: $x^* =$ Mittelpunkt von R , **STOPP!**
 - 4: **else**
 - 5: Dehne das Rechteck R entlang der kürzeren Seiten in beide Richtungen jeweils zu einem Quadrat Q_1 und Q_2 aus.
 - 6: Die Verbindungsline zwischen den Mittelpunkten M_1 und M_2 der beiden Quadrate Q_1 und Q_2 ist die optimale Lösungsmenge

$$\mathcal{X}^*(g) = \overline{M_1 M_2} = \{x \in \mathbb{R}^2 : x = \lambda M_1 + (1 - \lambda) M_2, \forall \lambda \in [0, 1]\} \quad (2.14)$$

7: **end if**

Example 2.4.2.**Ungewichtete 1-Centerprobleme mit l_∞ -Metrik****Beispiel**

Sei $A = \{(1,4), (2,6), (5,1), (4,2), (8,5)\}$

Verfahren

1. Umschreibendes Rechteck $R = (ul, or)$

$$ul = (\min_{i=1,\dots,n} a_{i,1}, \min_{i=1,\dots,n} a_{i,2}) = (1,1)$$

$$or = (\max_{i=1,\dots,n} a_{i,1}, \max_{i=1,\dots,n} a_{i,2}) = (8,6)$$

2. Kein **Quadrat** $\Rightarrow 3.$

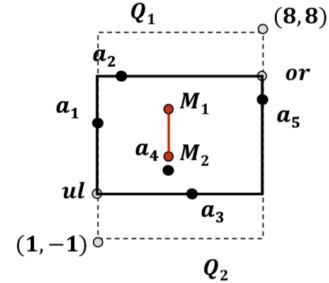
3. Dehne das **Rechteck** nach oben und unten jeweils zu einem **Quadrat** aus

$$Q_1 = (ul, (or_1, ul_2 + (or_1 - ul_1))) = ((1,1), (8,8)) \text{ und}$$

$$Q_2 = ((ul_1, or_2 - (or_1 - ul_1)), or) = ((1,-1), (8,6))$$

4. Mittelpunkte $M_1 = (4.5, 4.5)$ und $M_2 = (4.5, 2.5)$

$$\Rightarrow \mathcal{X}^*(g) = \{4.5\} \times [2.5, 4.5]$$

**2.4.3.1.2 Der allgemeine gewichtete Fall****Zielfunktion**

$$\begin{aligned}
 g(x) &= \max_{i=1,\dots,n} w_i \max\{|x_1 - a_{i,1}|, |x_2 - a_{i,2}|\} \\
 &= \underbrace{\max_{i=1,\dots,n} w_i |x_1 - a_{i,1}|}_{g_1(x_1)} \underbrace{\max_{i=1,\dots,n} w_i |x_2 - a_{i,2}|}_{g_2(x_2)}
 \end{aligned} \quad (2.15)$$

\Rightarrow Das l_∞ -Centerproblem lässt sich in zwei voneinander unabhängige Teilprobleme zerlegen.

Das 1-Centerproblem mit l_∞ -Metrik auf der Linie

$$\min_{x \in \mathbb{R}} g(x) \text{ mit } g(x) := \max_{i=1,\dots,n} w_i |x - a_i|$$

Algorithm 6 Lösungsverfahren für gewichtete 1-Centreprobleme mit l_∞ -Metrik auf der Linie

1: Berechne für jedes i und j mit $i, j \in 1, \dots, n$

$$\delta_{ij} = \frac{w_i w_j}{w_i + w_j} (a_j - a_i)$$

(Es gilt: $\delta_{ij} = -\delta_{ji} \Rightarrow$ Es reicht aus, δ_{ij} oder δ_{ji} zu berechnen; je nachdem ob $a_i \leq a_j$ oder $a_i > a_j$)

2: Ermittle

$$\delta_{pq} = \max\{\delta_{ij} : i, j \in \{1, \dots, n\}\}$$

3: Optimal Lösung:

$$z^* = \delta_{pq} \text{ und } x^* = \frac{w_p a_p + w_q a_q}{w_p + w_q}$$

Example 2.4.3.

1-Centerprobleme mit l_∞ -Metrik auf der Linie



Beispiel

Sei $A = \{1, 2, 5, 4, 6\}$ und $w = \{2, 3, 1, 1, 2\}$.

Verfahren

1.	i	j	$a_i \leq a_j?$	δ_{ij}
	1	2	$1 \leq 2$	$\delta_{12} = (a_2 - a_1) \cdot (w_1 \cdot w_2) / (w_1 + w_2) = (2 - 1) \cdot 2 \cdot 3 / (2 + 3) = 1.2$
		3	$1 \leq 5$	$\delta_{13} = (a_3 - a_1) \cdot (w_1 \cdot w_3) / (w_1 + w_3) = (5 - 1) \cdot 2 \cdot 1 / (2 + 1) = 2.66$
		4	$1 \leq 4$	$\delta_{14} = (a_4 - a_1) \cdot (w_1 \cdot w_4) / (w_1 + w_4) = (4 - 1) \cdot 2 \cdot 1 / (2 + 1) = 2.0$
		5	$1 \leq 6$	$\delta_{15} = (a_5 - a_1) \cdot (w_1 \cdot w_5) / (w_1 + w_5) = (6 - 1) \cdot 2 \cdot 2 / (2 + 2) = 5.0$
	2	3	$2 \leq 5$	$\delta_{23} = 2.25$
		4	$2 \leq 4$	$\delta_{24} = 1.5$
		5	$2 \leq 6$	$\delta_{25} = 4.8$
	3	4	$5 > 4$	$-\delta_{34} = \delta_{43} = 0.5$
		5	$5 \leq 6$	$\delta_{35} = 0.66$
	4	5	$4 \leq 6$	$\delta_{45} = 1.33$

2. $\delta_{pq} = \delta_{15} = 5$.

3. $z^* = \delta_{15} = 5$ und $x^* = (w_1 \cdot a_1 + w_5 \cdot a_5) / (w_1 + w_5) = (2 \cdot 1 + 2 \cdot 6) / (2 + 2) = 3.5$

Abbildung 2.2: Das 1-Centerproblem mit l_∞ -Metrik auf der Linie

Kombination der Lösungen der Einzelprobleme

Seien (x_1^*, z_1^*) und (x_2^*, z_2^*) optimale Lösungen der zwei Teilprobleme.

Dann ist $x^* := (x_1^*, x_2^*)$ eine optimale Lösungen mit $z^* = g(x^*) = \max\{g(x_1^*), g(x_2^*)\} = \max\{z_1^*, z_2^*\}$.

Menge aller optimalen Lösungen ist

$$\begin{aligned} \mathcal{X}^*(g) &= \mathcal{X}^*(g_1) \times \mathcal{X}^*(g_2) \\ &= [A_1^-(z^*), A_1^+(z^*)] \times [A_2^-(z^*), A_2^+(z^*)] \end{aligned}$$

- $A^-(z) := \max_{i=1, \dots, n} a_i - \frac{z}{a_i}$
- $A^+(z) := \max_{i=1, \dots, n} a_i + \frac{z}{a_i}$

2.4.4 1-Centerprobleme mit l_2 -Metrik

Example 2.4.4. Aufgabe 10

2.4.4.1 Zielfunktion

$$\begin{aligned} g(x) &= \max_{i=1,\dots,n} w_i l_2(x, a_i) \\ &= \max_{i=1,\dots,n} w_i \sqrt{(x_1 - a_{i,1})^2 + (x_2 - a_{i,2})^2} \end{aligned} \quad (2.16)$$

2.4.4.2 Der ungewichtete Fall: Das Kreis-Überdeckungsproblem

Umformulierung des Problems:

$$\min_{x \in \mathbb{R}^2} \max_{i=1,\dots,n} w_i l_2(x, a_i) \Leftrightarrow \min z, \text{ u.d.N. } l_2(x, a_i) \leq z, \forall i = 1, \dots, n$$

⇒ **Das minimale Kreis-Überdeckungsproblem:** Minimiere z unter der Nebenbedingung, dass die Entfernung von einem Punkt x zu allen Kundenstandorten kleiner oder gleich z ist.

Algorithm 7 Enumerations-Verfahren

- 1: Berechne alle minimalüberdeckenden Kreise von zwei oder drei Kundenstandorten a_i, a_j und $a_k \in A$
 - 2: Verwirf alle so gewonnenen minimalüberdeckenden Kreise, die nicht alle Kundenstandorten überdecken.
 - 3: Der verbleibende MÜK (Minimal Überdeckender Kreis) ist der MÜK für alle Kundenstandorten
-

Komplexität: $O(n^4) \rightarrow$ zu langsam!

Minimal-überdeckender Kreis (MÜK)

- von zwei Kundenstandorten a_i und a_j
 - Mittelpunkt: $x = (\frac{a_{i,1} + a_{j,1}}{2}, \frac{a_{i,2} + a_{j,2}}{2})$
 - Radius: $r = \frac{1}{2} l_2(a_i, a_j)$
- von drei Kunden $a_i, a_j, a_k \in A$
 - Fall 1: Das Dreieck a_i, a_j, a_k besitzt einen stumpfen Winkel ($\geq 90^\circ$) ⇒ Der MÜK ist durch die beiden Endpunkte der längsten Dreiecksseite definiert.
 - Fall 2: Der MÜK entspricht dem Umkreis des Dreiecks $a_i, a_j, a_k \Rightarrow$ Umkreis $UK(x, r)$
 - * Mittelpunkt x : Schnittpunkt zweier Mittelsenkrechten
 - Mittelsenkrechte zwischen zwei Punkten a_i und a_j gleichen Gewichts ist eine Gerade:

$$MS(a_i, a_j) := m_{ij} \cdot x_1 + b_{ij} := \frac{a_{i,1} - a_{j,1}}{a_{j,2} - a_{i,2}} \cdot x_1 + \frac{a_{j,1}^2 + a_{j,2}^2 - a_{i,1}^2 - a_{i,2}^2}{2(a_{j,2} - a_{i,2})}$$

- Schnittpunkt x zweier Mittelsenkrechten $MS(a_i, a_j) = m_{ij}x_1 + b_{ij}$ und $MS(a_i, a_k) = m_{ik}x_1 + b_{ik}$ ist der Punkt

$$x = \left(\frac{b_{ik} - b_{ij}}{m_{ij} - m_{ik}}, \frac{m_{ij} \cdot b_{ik} - b_{ij} \cdot m_{ik}}{m_{ij} - m_{ik}} \right)$$

- * Radius r : Entfernung von x zu einem der drei Eckpunkte des Dreiecks

$$r = l_2(x, a_i)$$

Algorithm 8 Verfahren von Elzinga & Hearn

- 1: Starte mit dem MÜK zweier beliebiger Kundenstandorten a und b
- 2: Werden alle Kunden von dem Kreis überdeckt \Rightarrow Schritt 9
- 3: Wähle einen Kunden c , der noch nicht überdeckt wird, und konstruiere den MÜK für diese drei Kunden.
- 4: Werden alle Kunden von dem Kreis überdeckt \Rightarrow Schritt 9.
- 5: Wird der MÜK von zwei Kunden a und b bestimmt \Rightarrow fahre mit a und b fort mit Schritt 3, \Rightarrow andernfalls fahre mit allen dreien fort mit Schritt 6.
- 6: Wähle einen weiteren Kunden d , der noch nicht überdeckt wird, und konstruiere (mit Hilfe des Enumerations-Verfahrens) den MÜK für diese vier Kunden.
- 7: Werden alle Kunden von dem Kreis überdeckt \Rightarrow Schritt 9
- 8: Wird der MÜK von zwei Kunden a und b bestimmt \Rightarrow fahre mit a und b fort mit Schritt 3, \Rightarrow andernfalls fahre mit den drei MÜK-definierenden Kunden fort mit Schritt 6.
- 9: Der momentane MÜK überdeckt alle Kundenstandorte und ist somit der optimale.

Example 2.4.5.

Ungewichtete 1-Centerprobleme mit l_2 -Metrik



Beispiel

Sei wieder $A = \{(1,1), (2,5), (3,3), (4,2)\}$ und $w = \{1,1,1,1\}$.

Verfahren

1. Starte mit den Kunden a_1 und a_3 . $MÜK = K((2, 2), 1.41)$.
2. a_2 und a_4 nicht überdeckt.
3. Wähle a_4 . $MÜK$ der Kunden a_1 , a_3 und $a_4 = K((2.5, 1.5), 1.58) = K(a_1, a_4)$.
4. a_2 nicht überdeckt.
5. **Der MÜK wird von a_1 und a_4 bestimmt.**
6. Wähle a_2 . $MÜK$ der Kunden a_1 , a_2 und $a_4 = K((2.05, 2.86), 2.14)$.
7. **Alle Kunden überdeckt!**
9. $K((2.05, 2.86), 2.14)$ ist der minimal überdeckende Kreis für alle Standorte.
 $\Rightarrow x = (2.05, 2.86)$ ist der **optimale Standort** für das l_2 -Centerproblem und $r = 2.14$ der **optimale Zielfunktionswert**.

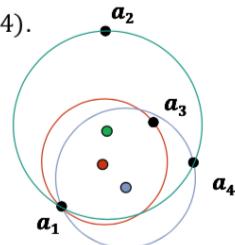


Abbildung 2.3: Verfahren von Elzinga & Hearn

2.5 Mehrstandortprobleme

Example 2.5.1. Aufgabe 10

Platziere p neue Einrichtungen $X = x_1, \dots, x_p, p \geq 2$ in der Ebene

- Modellen mit Interaktion
 - Neue Einrichtungen bieten jeweils unterschiedlichen Service (verschiedene Güter) an.

- Kunden haben Nachfrage nach unterschiedlichen Serviceleistungen (Gütern) von mehreren der neuen Einrichtungen.
- Interaktion, z. B. Austausch von Gütern, zwischen neuen Einrichtungen erlaubt.
- Zuordnungs-Modellen (Standort-Einzugsbereichs-Modellen
 - Neue Einrichtungen bieten alle gleichen Service (identische Güter) an.
 - Kunden befriedigen ihre Nachfrage von einem der neuen Standorte.

2.5.1 Modelle mit Interaktion

2.5.1.1 Annahmen

Nachfrage des Kunden i nach einem Service der neuen Einrichtung k : w_{ik}

Interaktion zwischen zwei neuen Einrichtungen k und l : s_{kl}

2.5.1.2 Zielfunktion

$$\min_{X \subseteq \mathbb{R}^2, |X|=p} f(X) := \sum_{i=1}^n \sum_{k=1}^p w_{ik} d(x_k, a_i) + \sum_{k=1}^{p-1} \sum_{l=k+1}^p s_{kl} d(x_k, x_l) \quad (2.17)$$

2.5.1.3 Notationen

x_1, \dots, x_p sind nun Punkte mit Koordinaten $x_{k,1}$ und $x_{k,2}, k = 1, \dots, p$

$X_1 := x_{1,1}, \dots, x_{p,1}$ und $X_2 := x_{1,2}, \dots, x_{p,2}$

2.5.1.4 Das Interaktions-Modell mit l_1 -Metrik

Zielfunktion

$$\begin{aligned} f(X) &= \sum_{i=1}^n \sum_{k=1}^p w_{ik} (|x_{k,1} - a_{i,1}| + |x_{k,2} - a_{i,2}|) + \sum_{k=1}^{p-1} \sum_{l=k+1}^p s_{kl} (|x_{k,1} - x_{l,1}| + |x_{k,2} - x_{l,2}|) \\ &= \underbrace{\sum_{i=1}^n \sum_{k=1}^p w_{ik} |x_{k,1} - a_{i,1}|}_{=: f_1(X_1)} + \underbrace{\sum_{k=1}^{p-1} \sum_{l=k+1}^p s_{kl} |x_{k,1} - x_{l,1}|}_{=: f_2(X_2)} + \underbrace{\sum_{i=1}^n \sum_{k=1}^p w_{ik} |x_{k,2} - a_{i,2}|}_{=: f_1(X_1)} + \underbrace{\sum_{k=1}^{p-1} \sum_{l=k+1}^p s_{kl} |x_{k,2} - x_{l,2}|}_{=: f_2(X_2)} \end{aligned} \quad (2.18)$$

→ Zielfunktion zerfällt in zwei voneinander unabhängige Funktionen.

Das Interaktions-Modell mit l_1 -Metrik auf der Linie

Zielfunktion

$$\sum_{i=1}^n \sum_{k=1}^p w_{ik} |x_k - a_i| + \sum_{k=1}^{p-1} \sum_{l=k+1}^p s_{kl} |x_k - x_l| \quad (2.19)$$

Umformulierung als lineares Programm

Seien $a, b \in \mathbb{R}$ und $c, d \in \mathbb{R}_+$

Gilt:

$$c \cdot d = 0 \text{ und } a - b = c - d$$

Dann:

$$|a - b| = c + d$$

Ersetze:

$$|x_k - a_i| = \alpha_{ik} + \beta_{ik} \in \mathbb{R}_+, \text{ für } \alpha_{ik}, \beta_{ik} \in \mathbb{R}_+$$

und

$$|x_k - x_l| = \gamma_{kl} + \delta_{kl} \in \mathbb{R}_+, \text{ für } \gamma_{kl}, \delta_{kl} \in \mathbb{R}_+$$

Dann:

$$\begin{aligned} \min \quad & \sum_{i=1}^n \sum_{k=1}^p w_{ik} (\alpha_{ik} + \beta_{ik}) + \sum_{k=1}^{p-1} \sum_{l=k+1}^p s_{kl} (\gamma_{kl} + \delta_{kl}) \\ \text{u.d.N.} \quad & x_k - \alpha_{ik} + \beta_{ik} = a_i, \quad \forall i = 1, \dots, n, k = 1, \dots, p \\ & \alpha_{ik} \cdot \beta_{ik} = 0, \quad \forall i = 1, \dots, n, k = 1, \dots, p \\ & x_k - x_l - \gamma_{kl} + \delta_{kl} = 0, \quad \forall k, l = 1, \dots, p, k < l \\ & \gamma_{kl} \cdot \delta_{kl} = 0, \quad \forall k, l = 1, \dots, p, k < l \\ & \alpha_{ik}, \beta_{ik}, \gamma_{kl}, \delta_{kl} \geq 0, \quad \forall i, k, l \text{ und } x_k, x_l \in \mathbb{R}, \forall k, l \end{aligned} \tag{2.20}$$

Example 2.5.2.

Mehrstandortprobleme mit Interaktion



Beispiel für ein 2-dimensionales Problem

Sei $A = \{(10, 15), (20, 25), (40, 5)\}$, $w_1 = \{2, 4\}$, $w_2 = \{1, 0\}$, $w_3 = \{0, 5\}$, $s_{k\ell} = 2$.

Teilproblem 1: Kundenstandorte $a_{1,1} = 10$, $a_{2,1} = 20$, $a_{3,1} = 40$

$$\begin{aligned} \min \quad & 2(\alpha_{11} + \beta_{11}) + 4(\alpha_{12} + \beta_{12}) + 1(\alpha_{21} + \beta_{21}) + 5(\alpha_{32} + \beta_{32}) + 2(\gamma_{12} + \delta_{12}) \\ \text{u.d.N.} \quad & x_1 \\ & -\alpha_{11} + \beta_{11} = 10 \\ & x_2 \\ & -\alpha_{12} + \beta_{12} = 10 \\ & x_1 \\ & -\alpha_{21} + \beta_{21} = 20 \\ & x_2 \\ & -\alpha_{32} + \beta_{32} = 40 \\ & x_1 - x_2 - \gamma_{12} + \delta_{12} = 0 \end{aligned}$$

Optimale Lösung

$$x_{1,1}^*, x_{2,1}^* \in [10, 20], f_1(X_1^*) = f_1(\{x_{1,1}^*, x_{2,1}^*\}) = 160$$

Abbildung 2.4: Mehrstandorten Problem mit Interaktion Umformulierung Bsp

2.5.1.5 Das Interaktions-Modell mit l_2 -Metrik

2.6 Zuordnungs-Modelle

- Neue Einrichtungen bieten alle gleichen Service (identische Güter) an.

- Kunden befriedigen ihre Nachfrage von einem der neuen Standorte.

Präferenzfunktion

Definiere eine Präferenzfunktion, welche jeden Kunden einer neuen Einrichtung zuordnet:

$$\text{Präf} : A \rightarrow X \quad \text{Präf}(a_i) = x_k \in X \text{ mit } d(x_k, a_i) = \min_{l=1, \dots, p} d(x_l, a_i)$$

(a_i liegt näher bei x_k als bei allen anderen Lösungspunkten; die Nachfrage von a_i wird von der neuen Einrichtung x_k befriedigt.)

Menge aller Kunden, welche einem Standort x_l zugeordnet sind

$$A_k := \{a_i \in A \mid \text{Präf}(a_i) = x_k\} = \{a_i \in A \mid d(x_k, a_i) = d(X, a_i)\}$$

2.6.1 Das p-Median Zuordnungsproblem

Zielfunktion

$$f(X) = f(\{x_1, \dots, x_p\}) = \sum_{i=1, \dots, n} w_i d(X, a_i) \quad (2.21)$$

2.6.1.1 Das p-Median Zuordnungsproblem mit l_1 -Metrik

Zielfunktion

$$f(X) = \sum_{i=1, \dots, n} w_i \min_{k=1, \dots, p} \{|x_{k,1} - a_{i,1}| + |x_{k,2} - a_{i,2}|\}$$

Definiere

$$y_k := \begin{cases} 1 & \text{falls der Schnittpunkte } s_k \text{ Teil der optimalen Lösung ist, } k = 1, \dots, |S| \\ 0 & \text{sonst} \end{cases}$$

$$x_{ik} := \begin{cases} 1 & \text{falls Kunde } i \text{ dem Schnittpunkte } s_k \text{ zugeordnet ist, } i = 1, \dots, n, k = 1, \dots, |S| \\ 0 & \text{sonst} \end{cases}$$

$$c_{ik} := w_i l_1(a_i, s_k), i = 1, \dots, n, k = 1, \dots, |S|$$

Modell

$$\begin{aligned} \min \quad & \sum_{i=1, \dots, n} \sum_{k=1, \dots, |S|} c_{ik} x_{ik} \\ \text{u.d.N} \quad & \sum_{k=1, \dots, p} x_{ik} = 1, \forall i \quad \text{Jeder Kunde } i \text{ wird einem Schnittpunkte zugeordnet} \\ & \sum_{k=1, \dots, |S|} y_k = p \quad (\text{p Schnittpunkte werden ausgewählt}) \\ & x_{ik} \leq y_k, \forall i, \forall k \quad (\text{Kunde } i \text{ darf nur eineme ausgewählten Schnittpunkt } k \text{ zugeordnet werden}) \\ & x_{ik}, y_k \in \{0, 1\} \end{aligned} \quad (2.22)$$

Ganzzahliges lineares Programm

→ Lässt sich mit Branch & Bound-Verfahren lösen.

2.6.1.2 Das Zuordnungsproblem mit l_2 -Metrik

Zielfunktion

$$f(X) = \sum_{i=1, \dots, n} w_i \min_{k=1, \dots, p} l_2(x_k, a_i) \quad (2.23)$$

Algorithm 9 Das Verfahren von Cooper

- 1: Wähle eine Startlösung $X^0 = x_1^{(0)}, \dots, x_p^{(0)}$ setze $l := 0$
 - 2: **while** ($\delta^{(l+1)} := \frac{f(X^l) - f(X^{l+1})}{f(X^l)} \leq \delta$ für ein $\delta > 0$) **do**
 - 3: **Zuordnungsschritt (allocation)**: Berechne für $X^{(l)}$ die Partition $Pa^{(l)} = \{A_1^{(l)}, \dots, A_p^{(l)}\}$ der Kundenstandorte
 - 4: **Platzierungsschritt (location)**: Bestimme die optimale Lösung $X_k^{(l+1)}$ des 1-Standortproblems mit Kunden $A_k^{(l)}$. Setze $X^{(l+1)} := \{x_1^{(l+1)}, \dots, x_p^{(l+1)}\}$
 - 5: $l := l + 1$
 - 6: **end while**
-

Kapitel 3

Standortplanung auf Netzwerken

3.1 Graphentheorie

Gewichteter ungerichteteter Graph $G = (V, E)$

- $V = v_1, \dots, v_n$: eine nichtleere Menge von Knoten
- $E = e_1, \dots, e_m$: Menge von Kanten

Ungerichtetes Netzwerk $N = (G, l)$

- $G = (V, E)$: ein gewichteter ungerichteter Graphen
- $l : E \rightarrow \mathbb{R}_+$: $l(e) = [v_i, v_j]$: Kantenbewertung (Länge der Kante)

Wege

$P(v_i, v_j) = (v_i = v_{i_1}, e_{j_1}, v_{i_2}, e_{j_2}, \dots, v_{i_{t-1}}, e_{j_{t-1}}, v_{i_t} = v_j)$

Kurz: $P(v_i, v_j) = (v_1, v_2, \dots, v_{i_{t-1}}, v_j)$

Gilt $v_i = v_j \Rightarrow$ Zyklus Z

Länge eines Weges:

$$l(p) := \sum_{s=1, \dots, t-1} l(e_{j_s})$$

Zusammenhang

Ein Graph heißt zusammenhängend, falls je zwei Knoten durch einen Weg miteinander verbunden sind.

Baum

Ein Baum $T = (V, E)$ ist ein zyklusfreier, zusammenhängender Graph, d. h. jedes Paar Knoten ist durch genau einen Weg miteinander verbunden.

Die Endknoten eines Baums nennt man Blatt-Knoten oder kurz Blätter. Ein Blatt ist mit genau einem anderen Knoten des Baums adjazent.

Teilgraph

$G' = (V', E')$ ist ein Teilgraph von $G = (V, E)$, falls $V' \subseteq V$ und $E' \subset E$ und für alle Kanten $e = [v_i, v_j] \in E'$ gilt, dass $v_i, v_j \in V'$

G' heißt **spanneder Teilgraph**, falls $V' = V$

Distanz zwischen zwei Knoten

Die Distanz $d(v_i, v_j)$ zwischen zwei Knoten v_i und $v_j \in V$ entspricht der Länge eines kürzesten Weges $P'(v_i, v_j)$ zwischen v_i und v_j

$$d(v_i, v_j) := l(P'(v_i, v_j)) := \min\{l(P(v_i, v_j)) \mid P(v_i, v_j) \text{ ein Weg von } v_i \text{ nach } v_j\}$$

- symmetrisch

- nicht-negativ
- $d(v_i, v_i) = 0$ für alle $v_i \in V$

Punkte des Netzwerks

Ein Punkt x auf einer Kante $e = [v_i, v_j]$ ist definiert als Paar $x = ([v_i, v_j], t)$, $0 \leq t \leq 1$ mit $d(x, v_i) = t \cdot l(e)$ und $d(x, v_j) = (1 - t) \cdot l(e)$

- $([v_i, v_j], 0) = v_i$
- $([v_i, v_j], 1) = v_j$

$P(G)$: die Menge aller Punkte $\Rightarrow V \subseteq P(G)$

Distanz zwischen Punkt und Knoten

Sei $x = ([v_i, v_j], t) \in P(G)$ und $v_k \in V$

Distanz zwischen x und v_k := Länge eines kürzesten Weges von x nach v_k

Bottleneck-Punkte

Ein Punkt x auf Kante $e = [v_i, v_j]$ heißt **Bottleneck-Punkt (BNP) des Knoten v_k** , falls

$$d(x, v_i) + d(v_i, v_k) = d(x, v_j) + d(v_j, v_k)$$

d.h. die kürzesten Wege von x nach v_k über v_i bzw. v_j haben identische Länge.

Bezeichnung: BN_{ij}^k Bottleneck-Punkt des Knotens v_k auf der Kante $[v_i, v_j]$

Für einen Bottleneck-Punkt BN_{ij}^k des Knotens v_k auf der Kante $[v_i, v_j]$ gilt

$$\begin{aligned} d(x, v_i) + d(v_i, v_k) &= d(x, v_j) + d(v_j, v_k) \\ \Leftrightarrow tl_{ij} + d_{ik} &= (1 - t)l_{ij} + d_{jk} \\ \Leftrightarrow 2tl_{ij} &= l_{ij} + d_{ik} - d_{jk} \\ \Leftrightarrow t &= \frac{l_{ij} + d_{jk} - d_{ik}}{2l_{ij}} =: t_{ij}^k \end{aligned}$$

Bestimmung aller Bottleneck-Punkte einer Kante $[v_i, v_j]$: Berechne t_{ij}^k für alle Knoten $v_k \in V$. Gilt $0 < t_{ij}^k < 1$, so liegt ein BNP vor, sonst nicht.

3.2 1-Medianprobleme

Platziere eine neue Einrichtung so auf dem Netzwerk, dass die Summe der gewichteten Entfernungen von dem neuen Standort zu den Knoten minimal wird.

Zielfunktion

$$f(x) := \sum_{i=1}^n w_i d(x, v_i), x \in P(G)$$

- Knotenbeschränktes: $\min_{v \in V} f(v)$
- Absolutes: $\min_{x \in P(G)} f(v)$

Dominanzkriterium

Gilt für einen Knoten v_k mit zugehörigem Gewicht w_k , dass

$$w_k \geq \frac{1}{2} \sum_{i=1, \dots, n} w_i$$

3.2.1 1-Medianprobleme auf allgemeinen Graphen

3.2.1.1 Knotenbeschränktes 1-Medianproblem

1. Berechne die gewichtete Distanzmatrix

$$\text{diag}(w_1, \dots, w_n) \cdot D = \begin{pmatrix} w_1 d(v_1, v_1) & \cdots & w_1 d(v_n, v_1) \\ \vdots & \ddots & \vdots \\ w_1 d(v_1, v_n) & \cdots & w_1 d(v_n, v_n) \end{pmatrix}$$

(Zeile i enthält die gewichteten Entfernungen von v_i zu allen anderen Knoten v_j : $w_i d(v_j, v_i)$)

2. Die Spalte mit der **kleinsten Spaltensumme** liefert einen Median.

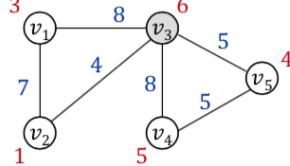
Example 3.2.1.

Knotenbeschränkte 1-Medianprobleme



Beispiel

Folgendes Netzwerk



Berechne die gewichtete Distanzmatrix

$$\begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{pmatrix} 3 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 6 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 4 \end{pmatrix} \cdot \begin{pmatrix} 0 & 7 & 8 & 16 & 13 \\ 7 & 0 & 4 & 12 & 9 \\ 8 & 4 & 0 & 8 & 5 \\ 16 & 12 & 8 & 0 & 5 \\ 13 & 9 & 5 & 5 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 21 & 24 & 48 & 39 \\ 7 & 0 & 4 & 12 & 9 \\ 48 & 24 & 0 & 48 & 30 \\ 80 & 60 & 40 & 0 & 25 \\ 52 & 36 & 20 & 20 & 0 \end{pmatrix}$$

Spaltensumme: 187 141 88 128 103

$\Rightarrow v^* = v_3$ ist der **Median des knotenbeschränkten Problems**.

Optimaler Zielfunktionswert $f(v^*) = 88$. Es ist: $\chi^*(f(V)) = \{v_3\}$

Bemerkung: Es kann **mehr als einen Median** geben.

Abbildung 3.1: Knotenbeschränktes 1-Medianproblem Bsp

3.2.1.2 Absolutes 1-Medianproblem

Zielfunktion

$$f(x) := \sum_{i=1}^n w_i d(x, v_i), x = (e, t), 0 \leq t \leq 1$$

Hakimi-Knotendominanzkriterium Man findet unter den Knoten bereits einen absoluten Median:

$$\chi^*(f(G)) \cap V \neq \emptyset$$

Lösungsverfahren: Verwende das Verfahren für das knotenbeschränkte Problem zur Bestimmung eines absoluten Medians.

- Ist die Summe aller Gewichte ungerade, d. h. $\sum_{i=1, \dots, n} w_i$, so gilt: $\chi^*(f(G)) \subseteq V$

- Sind zwei adjazente Knoten optimal und ein beliebiger Punkt auf der Kante dazwischen ebenfalls, so ist die gesamte Kante optimal.

Example 3.2.2.

Beispiel

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix} \cdot \begin{pmatrix} 0 & 8 & 3 & 7 \\ 8 & 0 & 7 & 3 \\ 3 & 7 & 0 & 4 \\ 7 & 3 & 4 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 8 & 3 & 7 \\ 8 & 0 & 7 & 3 \\ 6 & 14 & 0 & 8 \\ 14 & 6 & 8 & 0 \end{pmatrix}$$

Spaltensummen: 28 28 18 18

$\Rightarrow v_3$ und v_4 sind optimal. Zudem gilt $f\left(\left([v_3, v_4], \frac{1}{2}\right)\right) = 18$

$\Rightarrow [v_3, v_4]$ ist optimal. Es gilt: $\chi^*(f(G)) = \{[v_3, v_4]\}$

Abbildung 3.2: Absolut 1 Medianproblem Bsp

3.2.2 1-Medianprobleme auf Bäume

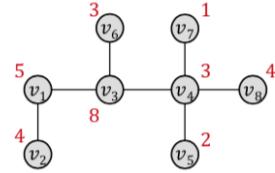
Algorithm 10 Einklapp Verfahren von Goldmann

1. Berechne $W = w(V)$
 2. Gilt $T = \{v_i\}$, so $v^* := v_i \Rightarrow$ Stopp!
 - Sonst wähle eine Blatt-Kante $e = [v_i, v_j]$ von T mit Blatt v_i
Falls $w_i \geq \frac{W}{2}$, so $v^* := v_i \rightarrow$ Stopp! (Gilt $w_i = \frac{W}{2}$, so ist die komplette Kante $[v_i, v_j]$ optimal.)
 3. Setze $w_j := w_j + w_i$ und lösche v_i und $[v_i, v_j]$ aus T .
Gehe zu Schritt 2.
-

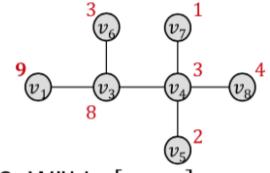
Example 3.2.3.

1-Medianprobleme auf Bäumen

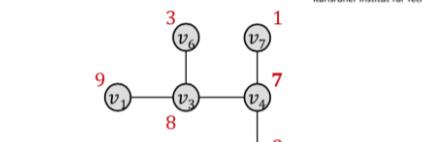
Beispiel



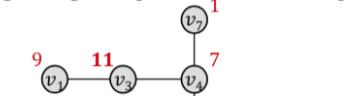
1. $W = w(V) = 30$. $W/2 = 15$.
2. Wähle $[v_2, v_1]$. $w_2 = 4 < 15$.
3. $w_1 = w_1 + w_2 = 9$. Lösche v_2 .



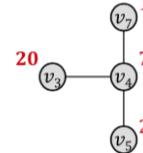
2. Wähle $[v_8, v_4]$. $w_8 = 4 < 15$.
3. $w_4 = w_4 + w_8 = 7$. Lösche v_8 .



2. Wähle $[v_6, v_3]$. $w_6 = 3 < 15$.
3. $w_3 = w_3 + w_6 = 11$. Lösche v_6 .



2. Wähle $[v_1, v_3]$. $w_1 = 9 < 15$.
3. $w_3 = w_3 + w_1 = 20$. Lösche v_1 .



2. Wähle $[v_3, v_4]$. $w_3 = 20 \geq 15$
 $\Rightarrow v^* = v_3$ ist der Median

Abbildung 3.3: 1 Medianproblem auf Bäume

3.3 1-Centerprobleme

Platziere eine neue Einrichtung so auf dem Netzwerk, dass die maximale gewichtete Entfernung von dem neuen Standort zu den Knoten minimal wird.

Zielfunktion

$$g(x) := \max_{i=1,\dots,n} w_i \cdot d(x, v_i), x \in P(G)$$

3.3.1 1-Centerprobleme auf allgemeinen Graphen

Zielfunktion: $\min_{v \in V} g(v)$

Lösungsverfahren

1. Berechne die gewichtete Distanzmatrix

$$\text{diag}(w_1, \dots, w_n) \cdot D = \begin{pmatrix} w_1 d(v_1, v_1) & \cdots & w_1 d(v_n, v_1) \\ \vdots & \ddots & \vdots \\ w_1 d(v_1, v_n) & \cdots & w_1 d(v_n, v_n) \end{pmatrix}$$

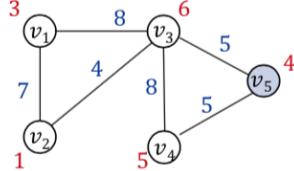
2. Bestimme für jede Spalte den **maximalen Eintrag** in der Spalte. Die Spalte mit dem **kleinsten maximalen Eintrag** liefert einen Center.

Example 3.3.1.

Knotenbeschränkte 1-Centerprobleme

Beispiel

Folgendes Netzwerk



Berechne die gewichtete Distanzmatrix

$$\begin{array}{l} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{pmatrix} 3 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 6 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 4 \end{pmatrix} \cdot \begin{pmatrix} 0 & 7 & 8 & 16 & 13 \\ 7 & 0 & 4 & 12 & 9 \\ 8 & 4 & 0 & 8 & 5 \\ 16 & 12 & 8 & 0 & 5 \\ 13 & 9 & 5 & 5 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 21 & 24 & 48 & 39 \\ 7 & 0 & 4 & 12 & 9 \\ 48 & 24 & 0 & 48 & 30 \\ 80 & 60 & 40 & 0 & 25 \\ 52 & 36 & 20 & 20 & 0 \end{pmatrix}$$

maximale Einträge: 80 60 40 48 39

⇒ $v^* = v_5$ ist der Center des knotenbeschränkten Problems.

Optimaler Zielfunktionswert: $g(v^*) = 39$.

Bemerkung

Es kann mehr als einen Center geben.

Abbildung 3.4: Knotenbeschraenktes 1-Centerproblem Bsp

3.3.2 1-Centerprobleme auf Bäume

Sowohl das absolute als auch das knotenbeschränkte Problem auf Bäumen mit nicht-identischen Knotengewichten wird mit den Verfahren der vorherigen Abschnitte für allgemeine Graphen gelöst.

Für ungewichtete Probleme gibt es effizientere Verfahren.

3.3.2.1 Das ungewichtete 1-Centerproblem

Zielfunktion

$$g(x) := \max_{i=1,\dots,n} d(x, v_i), x \in P(T)$$

Der absolute, ungewichtete Center eines Baums ist der Mittelpunkt eines längsten Weges zwischen zwei Knoten im Baum.

Algorithm 11 Verfahren für absolute 1-Centerprobleme auf Bäume

- 1: Wähle einen beliebigen Punkt $x \in P(T), T = (V, E)$
- 2: Bestimme den am weitesten von x entfernt liegenden Knoten v in T .

$$d(v, x) := \max_{i=1,\dots,n} d(v_i, x)$$

- 3: Bestimme den am weitesten von v entfernt liegenden Knoten v' in T

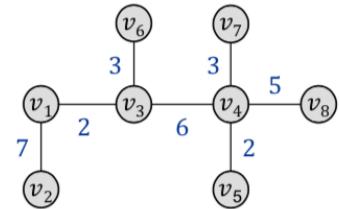
$$d(v', v) := \max_{i=1,\dots,n} d(v_i, v)$$

- 4: $x^* = \text{Mittelpunkt von } P(v, v') : d(x^*, v) = d(x^*, v') = \frac{1}{2}d(v, v') = g(x^*)$
-

Example 3.3.2.

Beispiel

Betrachte den folgenden **Baum** $T = (V, E)$.



Verfahren

1. Wähle $x = v_6$.
2. $d(v, x) = \max\{5, 12, 3, 9, 11, 0, 12, 14\} = d(v_8, x) \rightarrow v = v_8$
3. $d(v', v) = \max\{13, 20, 11, 5, 7, 14, 8, 0\} = d(v_2, v) \rightarrow v' = v_2$
4. $\frac{1}{2}d(v, v') = 10 \Rightarrow x^* = ([v_3, v_4], 0.16)$.

Abbildung 3.5: Absolute 1-Centerproblem auf ungewichtete Baeume Bsp

Das knotenbeschränkte Problem

Sei $x^* = ([v_i, v_j], t)$ ein absoluter Center auf der Kante $e = [v_i, v_j]$

Dann ist einer der Endknoten der Kante $[v_i, v_j]$ ein Center:

$$g(x^*) := \min\{g(v_i), g(v_j)\}$$

Example 3.3.3.

Beispiel (Fort.):

$x^* = ([v_3, v_4], 0.16)$ ist ein **absoluter Center**.

Dann:

$$g(x^*) := \min\{g(v_3), g(v_4)\} = \min\{11, 15\} = 11$$

$\Rightarrow x^* = v_3$ ist ein **knotenbeschränkter Center**

Abbildung 3.6: Knotenbeschränktes 1-Centerproblem auf ungewichtete Baeume Bsp

3.4 Mehrstandortprobleme

Kapitel 4

Diskrete Standortplanung

4.1 Klassifikation diskreter Standortprobleme

Planungshorizont

Anzahl der Zeitperioden (z. B. Monate, Jahre) im Planungszeitraum.

- Statische (Ein-Perioden) Modelle
Standortentscheidungen werden zu Beginn des Planungshorizontes getroffen
- Dynamische (Mehr-Perioden) Modelle
Entscheide wo und wann, d. h. in welcher Periode, Einrichtungen platziert werden.

Einrichtungstypen

Unterscheide verschiedene Ausprägungen von Einrichtungen,

Produkttaggregation

Ist nur ein Produkt oder sind mehrere Produkte mit unterschiedlichen Charakteristiken zu planen?

Stufigkeit

Anzahl der Stufen des Distributionsnetzwerks.

- Einstufige Modelle
Güterverteilung über eine Transportstufe (z. B. Lager→Händler). Standortentscheidung nur auf einer Ebene (z. B. Lager).
- Mehrstufige (hierarchische) Modelle
Güterverteilung über mehrere Transportstufen (z. B. Werk→Lager→Händler).

Interaktion

Interaktion zwischen Einrichtungen desselben Typs möglich, z. B. Transporte zwischen Warenhäusern.

Unsicherheit

- Deterministische Modelle
alle Daten und Parameter sind vollständig bekannt
- Probabilistische (stochastische) Modelle
manche Daten und Parameter sind unsicher

Flussrichtung

- Distributionslogistik
Waren fließen von Werken/Lieferanten zu den Lagern/Kunden
- Entsorgungslogistik (Reverse Logistics)
Waren fließen von Kunden zu Recyclinganlagen, Deponien

Kapazitätsrestriktionen

Beschränken die Mengen, welche produziert, transportiert, umgeschlagen etc. werden können.

Transportmodus

- Touren-Belieferungen
Auslieferungen von einer übergeordneten Einrichtung zu untergeordneten Einrichtungen (oder umgekehrt) werden zu Touren gebündelt.
- Direkt-Belieferungen
Auslieferungen von übergeordneten zu untergeordneten Ebenen geschehen direkt und ohne Umwege.

4.2 Das Warehouse Location Problem (WLP)

Auch als **Uncapacitated Facility Location Problem (UFLP)** oder **Simple Plant Location Problem (SPLP)** bekannt.

Modell-Annahmen

- statisch, einstufig, unkapazitiert, deterministisch
- ein Produkt, Direkt-Belieferung, ein Einrichtungstyp, keine Interaktion

Gegeben

- Menge von Kunden $I = \{1, \dots, n\}$ mit Bedarfen $b_i, i = 1, \dots, n$
- Menge potentieller Standorte $J = \{1, \dots, m\}$ für neue Einrichtungen
- Kosten t_{ij} für den Transport einer Mengeneinheit vom potentiellen Standort $j \in J$ zu einem Kunden $i \in I$. → Gesamtkosten zur Belieferung des Kunden i vom Standort j : $c_{ij} = b_i \cdot t_{ij}$
→ Kostenmatrix $C = (c_{ij})_{i=1, \dots, n, j=1, \dots, m}$
- Fixkosten f_j für die Errichtung einer neuen Einrichtung am Standort j .

Entscheidungen

- Anzahl neuer Einrichtungen
- Standorte neuer Einrichtungen
- Zuordnung von Kunden zu neuen Einrichtungen

4.2.1 Modellierung

Standortentscheidung

$$y_j = \begin{cases} 1 & \text{falls am potentiellen Standort } j \text{ eine neue Einrichtung platziert wird, } j \in J \\ 0 & \text{sonst} \end{cases}$$

Zuordnungsentscheidung

$$x_{ij} = \begin{cases} 1 & \text{falls Kunde } i \text{ einer neuen Einrichtung am Standort } j \text{ zugeordnet wird, } i \in I, j \in J \\ 0 & \text{sonst} \end{cases}$$

Mathematisches Modell

$$\begin{aligned}
\min \quad & \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} + \sum_{j=1}^m f_k y_j \\
\text{u.d.N} \quad & \sum_{j=1}^m x_{ij} = 1, \forall i \quad (\text{Jeder Kunde } i \text{ wird genau einer neuen Einrichtung an einem Standort } j \text{ zugeordnet}) \\
& x_{ij} \leq y_j, \forall i, \forall j \quad (\text{Kunde } i \text{ darf nur dann Standort } j \text{ zugeordnet werden, wenn an } j \text{ eine Einrichtung platziert}) \\
& x_{ij} \geq 0, y_j \in \{0, 1\}, \forall i, \forall j
\end{aligned} \tag{4.1}$$

4.2.2 Heuristiken

Methoden zur Bestimmung einer heuristischen Lösung X_H für ein Problem.

(Finden nicht notwendigerweise eine optimale Lösung.)

Güte einer heuristischen Lösung X_H : Relative Abweichung in Prozent

$$RPD(X_H, X^*) = \frac{f(X_H) - f(X^*)}{f(X^*)} * 100\% \leq \frac{f(X_H) - LB}{LB}$$

Es gibt zwei Klassen von Standort-Zuordnungs-Heuristiken:

- Konstruktionsheuristiken: Bestimmen „aus dem Nichts heraus“ eine Lösung des Problems.
 - Greedy (ADD-Heuristik)
 - Stingy (DROP-Heuristik)
- Verbesserungsheuristiken: Versuchen eine bereits vorhandene Lösung zu verbessern.
 - Interchange (ADD&DROP-Heuristik)
 - Metaheuristiken: Variable Neighborhood/Tabu Search, Genetic Algorithms

4.2.2.1 Greedy-Heuristik

Algorithm 12 Greedy-Heuristik

```

1: Init:  $X^0 := \emptyset, J' := J, l := 1$ . Berechne  $u_i^0 = \max_{j=1,\dots,m} c_{ij}, F(X^0) = \sum_{i=1}^n u_i^0$ 
2: while ( $J \neq \emptyset$ ) do
3:   Berechne  $\Delta_j^l = \sum_{i=1}^n \max\{0, u_i^{l-1} - c_{ij}\}$  und  $\omega_j^l = \Delta_j^l - f_j = \sum_{i=1}^n \max\{0, u_i^{l-1} - c_{ij}\} - f_j, \forall j \in J'$ 
4:   Bestimme potentiellen Standort  $j \in J'$  mit
      
$$\tilde{\omega}_j^l = \max_{k \in J'} \{\omega_k^l : \omega_k^l > 0\}$$

5:    $X^l := X^{l-1} \cup \{j\}, J' := J' \setminus \{j\}, F(x^l) = F(X^{l-1}) - \tilde{\omega}_j^l$ 
6:   if ( $\omega_h^l \leq 0, h \in J'$ ) then
7:      $J := J' \setminus \{h\}$ 
8:   end if
9:   Aktualisiere geringste Kosten für alle Kunden  $i$ :  $u_i^l = \min\{c_{ij} | j \in X^l\}$ 
10:   $l := l + 1$ 
11: end while

```

Output: eine heuristische Lösung des Problems X^{l-1}

Example 4.2.1.

Greedy-Heuristik: Beispiel

Beispiel

- Folg. Transport- und Errichtungskosten
- Iterationsschritt $\ell = 1, J' = \{1,2,3,4\}$

				u_i^0
8	10	19	13	19
7	14	24	21	24
18	12	5	12	18
16	14	7	13	16
15	15	8	18	18
$(19 - 8) + (24 - 7) + (18 - 18) + (16 - 16) + (18 - 15) = 31$				f_j
				21 17 20 22
				95 = $F(X^0)$
				$\Delta_j^1 = 31$
				$\omega_j^1 = 10 \quad 13 \quad 12 \quad -4$

$$C = \begin{pmatrix} 8 & 10 & 19 & 13 \\ 7 & 14 & 24 & 21 \\ 18 & 12 & 5 & 12 \\ 16 & 14 & 7 & 13 \\ 15 & 15 & 8 & 18 \end{pmatrix} \quad F = \begin{pmatrix} 21 \\ 17 \\ 20 \\ 22 \end{pmatrix}$$

$$\boxed{\begin{aligned} u_i^0 &= \max\{c_{ij} \mid j = 1, \dots, m\} \\ w_j^\ell &= \sum_{i=1}^n \max\{0, u_i^{\ell-1} - c_{ij}\} - f_j \\ &= \Delta_j^\ell - f_j \end{aligned}}$$

- Größte Kostenersparnis mit Standort 2: $\tilde{\omega}_j^l = \omega_2^1 = 13$
 $\Rightarrow X^1 = X^0 \cup \{2\}, J' = \{1,3,4\}$ und $F(X^1) = F(X^0) - \omega_2^1 = 95 - 13 = 82$.
 Da $\omega_4^1 < 0$, lösche Standort 4 aus J' $\Rightarrow J' = \{1,3\}$.

Greedy-Heuristik: Beispiel

- Iterationsschritt $\ell = 2, J' = \{1,3\}$ und $X^1 = \{2\}$.
- | | | | | $u_i^0 \quad u_i^1$ |
|--------------|-----|----|----|---------------------|
| $x_1 = 2$ | 8 | 10 | 19 | 13 19 10 |
| | 7 | 14 | 24 | 21 24 14 |
| | 18 | 12 | 5 | 12 18 12 |
| | 16 | 14 | 7 | 13 16 14 |
| | 15 | 15 | 8 | 18 15 |
| f_j | 21 | 17 | 20 | 22 95 |
| Δ_j^1 | 31 | 30 | 32 | 18 |
| ω_j^1 | 10 | 13 | 12 | -4 |
| Δ_j^2 | 9 | - | 21 | - 82 |
| ω_j^2 | -12 | - | 1 | - |
- Einige Kostenersparnis mit Standort 3: $\tilde{\omega}_j^l = \omega_3^2 = 1$.
 - $\Rightarrow X^2 = X^1 \cup \{3\}, J' = \{1\}$ und $F(X^2) = 82 - 1 = 81$.
 - Da $\omega_1^2 < 0$, lösche Standort 1 aus J' $\Rightarrow J' = \emptyset \rightarrow$ Stopp
 - Heuristische Lösung: $X_H = X^2 = \{2,3\}$ und $F(X_H) = 81$.
 - Wert der LP-Relaxation: $76 \Rightarrow X_H$ max. 6.6% schlechter als die optimale Lösung.

Abbildung 4.1: Greedy Heuristik Bsp

Example 4.2.2. Aufgabe 14

4.2.2.2 Interchange-Heuristik

4.2.3 Das DUALOC-Verfahren

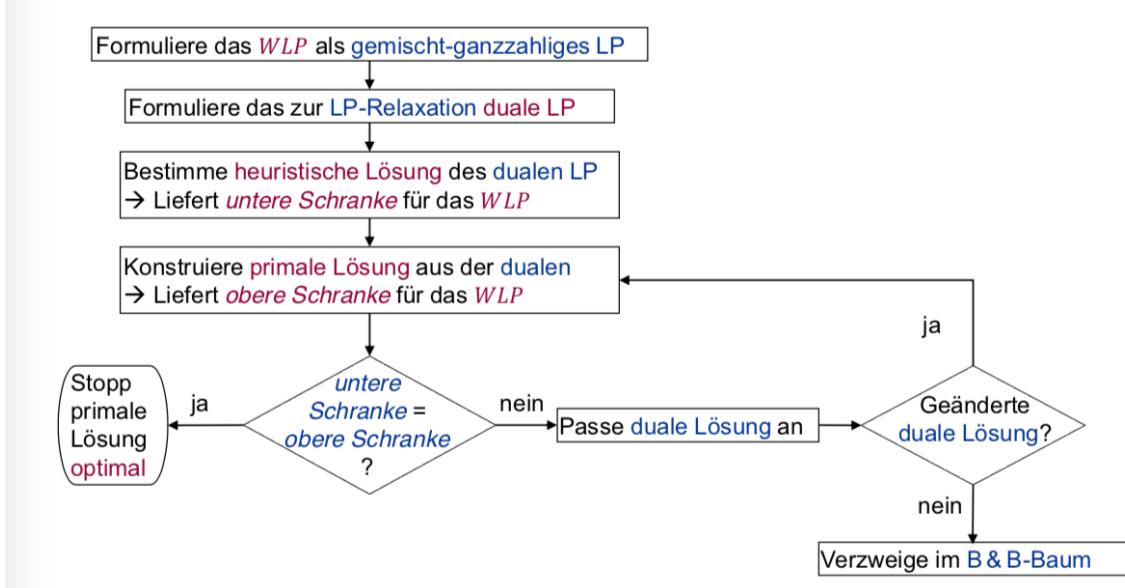


Abbildung 4.2: DUALOC-Verfahren

Example 4.2.3. Aufgabe 15, 16

1. WLP als gemischt-ganzzahliges lineares Programm

$$\begin{aligned}
 \min \quad & \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} + \sum_{j=1}^m f_j y_j \\
 \text{u.d.N} \quad & \sum_{j=1}^m x_{ij} = 1, \forall i \\
 & x_{ij} \leq y_j, \forall i, \forall j \\
 & x_{ij} \geq 0, y_j \in \{0, 1\}, \forall i, \forall j
 \end{aligned} \tag{4.2}$$

LP-Relaxation (LP) des gemischt-ganzzahligen linearen Programms für das WLP

$$\begin{aligned}
 \min \quad & \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} + \sum_{j=1}^m f_j y_j \\
 \text{u.d.N} \quad & \sum_{j=1}^m x_{ij} = 1, \forall i \\
 & x_{ij} \leq y_j, \forall i, \forall j \\
 & x_{ij} \geq 0, y_j \geq 0, \forall i, \forall j
 \end{aligned} \tag{4.3}$$

2. duale lineare Programm (DP) mit den Dual-Variablen v_i und w_{ij}

$$\begin{aligned}
 D(v, w) = \max & \quad \sum_{i=1}^n v_i \\
 \text{u.d.N} & \quad \sum_{i=1}^n w_{ij} \leq f_j, \forall j \\
 & \quad v_i - w_{ij} \leq c_{ij}, \forall i, \forall j \\
 & \quad v_i \leq 0, \forall i \\
 & \quad w_{ij} \geq 0, \forall i, \forall j
 \end{aligned} \tag{4.4}$$

Reduziertes duales LP (RDP)

$$\begin{aligned}
 D(v, w) = \max & \quad \sum_{i=1}^n v_i \\
 \text{u.d.N} & \quad \sum_{i=1}^n \max\{0, v_i - c_{ij}\} \leq f_j, \forall j \\
 & \quad v_i \leq 0, y_j \geq 0, \forall i
 \end{aligned} \tag{4.5}$$

3. Löse das (RDP) mit dem **Dual Ascent-Verfahren** und erhalte eine heuristische Lösung von (DP). → liefert die untere Schranke für das (WLP).

4.2.4 Dual Ascent-Verfahren

Idee/Vorgehensweise

Erhöhe, ausgehend von einer Startlösung v , reihum jedes v_i solange um einen kleinen Wert, bis keines mehr erhöht werden kann.

Definiere:

- $J_i := \{j \in J | v_i \geq c_{ij}\}$
- $s_j := f_j - \sum_{i=1}^n \max\{0, v_i - c_{ij}\}, \forall j \in J$
- $k(i) := \min\{k | c_i^k \geq v_i\}$

Algorithm 13 Dual Ascent-Verfahren

```

1: Init:
2: Sortiere die Kostenelemente  $c_{ij}$  jedes Kunden  $i \in I$  nach monoton zunehmenden Werten und bezeichne
   sie mit  $c_i^k: c_i^1 \leq c_i^2 \leq \dots \leq c_i^m \leq c_i^{m+1} := \infty$ 
3:  $I' := I, v_i := c_i^1$  für  $\forall i \in I$ 
4: Berechne  $J_i := \{j \in J | v_i \geq c_{ij}\}, \forall i \in I$  und  $s_j := f_j - \sum_{i=1}^n \max\{0, v_i - c_{ij}\}, \forall j \in J$ 
5: Bestimme  $k(i) := \min\{k | c_i^k \geq v_i\}, \forall i \in I$ 
6: if ( $v_i = c_i^{k(i)}$ ) then
7:    $k(i) := k(i) + 1$ 
8: end if
9: while ( $I' \neq \emptyset$ ) do
10:   Für  $\forall i \in I'$ : Bestimme  $\Delta_i := \min\{s_j | j \in J_i\}$ 
11:   if ( $\Delta_i = 0$ ) then
12:      $I' := I' \setminus \{i\}$ 
13:     Wähle das nächste  $i \in I'$ 
14:   else
15:     Setze  $\Delta := \min\{\Delta_i, c_i^{k(i)} - v_i\}$ 
16:     if ( $\Delta = \Delta_i$ ) then
17:        $I' := I' \setminus \{i\}$ 
18:     else
19:        $k(i) := k(i) + 1$ 
20:     end if
21:     Setze  $v_i := v_i + \Delta$ 
22:     Für  $\forall j \in J_i$ , setze  $s_j := s_j - \Delta$ 
23:     Aktualisiere  $J_i$ 
24:   end if
25: end while

```

Output: eine heuristische Lösung von (RDP) (Untere Schranke für das (WLP))

Example 4.2.4.

Dual Ascent-Verfahren: Beispiel

Beispiel

- Folg. Transport- und Errichtungskosten d. h. 6 Kunden und 5 pot. Standorte.

$$C = \begin{pmatrix} 0 & 4 & 6 & 12 & 3 \\ 2 & 6 & 3 & 4 & 0 \\ 4 & 9 & 10 & 12 & 8 \\ 6 & 0 & 12 & 3 & 7 \\ 9 & 12 & 0 & 3 & 8 \\ 12 & 8 & 4 & 0 & 6 \end{pmatrix} \quad F = \begin{pmatrix} 12 \\ 6 \\ 9 \\ 9 \\ 12 \end{pmatrix}$$

1. Initialisierung

- Sortierung der Kostenelemente: z. B. für $i = 1$
- $c_1^1 = 0 \leq c_1^2 = 3 \leq c_1^3 = 4 \leq c_1^4 = 6 \leq c_1^5 = 12 \leq c_1^6 = \infty$ usw.
- $I' := \{1, \dots, 6\}$ und $v_1 = v_2 = v_4 = v_5 = v_6 = 0, v_3 = 4$
- $J_1 = \{1\}, J_2 = \{5\}, J_3 = \{1\}, J_4 = \{2\}, J_5 = \{3\}, J_6 = \{4\}$.
- $s_1 = 12, s_2 = 6, s_3 = 9, s_4 = 9, s_5 = 12$.
- $k(1) = k(2) = k(3) = k(4) = k(5) = k(6) = 2$.
- Sammle all diese Daten in einem Tableau.

$$\begin{aligned} J_l &= \{j \in J \mid v_l \geq c_{ij}\} \\ s_j &:= f_j - \sum_{i=1}^n \max\{0, v_i - c_{ij}\} \\ k(i) &:= \min\{k \mid c_i^k \geq v_i\} \end{aligned}$$

Abbildung 4.3: Dual Ascent-Verfahren Bsp

Dual Ascent-Verfahren: Beispiel

Die Sortierung der Kostenelemente wird der Einfachheit halber nicht explizit im Tableau durchgeführt.

Die Kostenelemente $c_i^{k(i)}$ werden durch Unterstriche kenntlich gemacht.

$i \setminus j$	1	2	3	4	5	v_i	J_i
1	0	4	6	12	<u>3</u>	0	{1}
2	<u>2</u>	6	3	4	0	0	{5}
3	4	9	10	12	<u>8</u>	4	{1}
4	6	0	12	<u>3</u>	7	0	{2}
5	9	12	0	<u>3</u>	8	0	{3}
6	12	8	<u>4</u>	0	6	0	{4}
s_j	12	6	9	9	12		

2. Iteration: Starte mit $i = 1$.

$$\Delta_1 = \min\{s_j \mid j \in J_1\} = s_1 = 12.$$

$$\Delta = \min\{\Delta_1, c_1^2 - v_1 = 3 - 0\} = 3.$$

$$\text{Da } \Delta = c_1^2 - v_1,$$

setze $k(1) := k(1) + 1 = 2 + 1 = 3$.

Setze $v_1 := v_1 + \Delta = 0 + 3 = 3$ und für alle $j \in J_1 = \{1\}$ setze $s_j := s_j - \Delta$, d. h. $s_1 - \Delta = 12 - 3 = 9$.

Aktualisiere $J_1: J_1 = \{1, 5\}$.

$i \setminus j$	1	2	3	4	5	v_i	J_i	v_i	J_i
1	0	<u>4</u>	6	12	<u>3</u>	0	1	<u>3</u>	{1, 5}
2	<u>2</u>	6	3	4	0	0	5	0	{5}
3	4	9	10	12	<u>8</u>	4	1	4	{1}
4	6	0	12	<u>3</u>	7	0	2	0	{2}
5	9	12	0	<u>3</u>	8	0	3	0	{3}
6	12	8	<u>4</u>	0	6	0	4	0	{4}
s_j	12	6	9	9	12				
s_j	<u>9</u>	6	9	9	12				

Abbildung 4.4: Dual Ascent-Verfahren Bsp

Dual Ascent-Verfahren: Beispiel

$i = 2$:

- $\Delta_2 = \min\{s_j \mid j \in J_2\} = s_5 = 12$.
- $\Delta = \min\{\Delta_i, c_2^2 - v_2 = 2 - 0\} = 2$.
- Da $\Delta = c_2^2 - v_2$, setze $k(2) := k(2) + 1 = 3$.
- Setze $v_2 := v_2 + \Delta = 2$
und für alle $j \in J_2 = \{5\}$
setze $s_j := s_j - \Delta$, d.h. $s_5 - \Delta = 12 - 2 = 10$.
- Aktualisiere J_2 : $J_2 = \{1, 5\}$.

$i \setminus j$	1	2	3	4	5	v_i	J_i	v_i	J_i
1	0	4	6	12	3	0	1	3	{1, 5}
2	2	6	3	4	0	0	5	2	{1, 5}
3	4	9	10	12	8	4	1	8	{1, 5}
4	6	0	12	3	7	0	2	3	{2, 4}
5	9	12	0	3	8	0	3	3	{3, 4}
6	12	8	4	0	6	0	4	4	{3, 4}
s_j	12	6	9	9	12				
s_j	9	6	9	9	10				

s_j | 12 6 9 9 12 |

s_j | 9 6 9 9 10 |

■ Nach $i = 3, \dots, 6$

$i \setminus j$	1	2	3	4	5	v_i	J_i	v_i	J_i
1	0	4	6	12	3	0	1	3	{1, 5}
2	2	6	3	4	0	0	5	2	{1, 5}
3	4	9	10	12	8	4	1	8	{1, 5}
4	6	0	12	3	7	0	2	3	{2, 4}
5	9	12	0	3	8	0	3	3	{3, 4}
6	12	8	4	0	6	0	4	4	{3, 4}
s_j	12	6	9	9	12				
s_j	5	3	6	5	10				
s_j	2	0	4	0	7				

Abbildung 4.5: Dual Ascent-Verfahren Bsp

Dual Ascent-Verfahren: Beispiel

- $I' \neq \emptyset \Rightarrow$ starte neue Iteration. Tableau nach $i = 1, \dots, 6$

$i \setminus j$	1	2	3	4	5	v_i	J_i	v_i	J_i	v_i	J_i
1	0	4	6	12	3	0	1	3	{1, 5}	4	{1, 2, 5}
2	2	6	3	4	0	0	5	2	{1, 5}	3	{1, 3, 5}
3	4	9	10	12	8	4	1	8	{1, 5}	9	{1, 2, 5}
4	6	0	12	3	7	0	2	3	{2, 4}	6	{1, 2, 4}
5	9	12	0	3	8	0	3	3	{3, 4}	5	{3, 4}
6	12	8	4	0	6	0	4	4	{3, 4}	4	{3, 4}
s_j	12	6	9	9	12						
s_j	5	3	6	5	10						
s_j	2	0	4	0	7						

Bemerkungen

- v_4 wird zwar auf c_{41} erhöht, trotzdem ist $\Delta_4 = \Delta$.
- v_5 konnte nicht auf den drittgrößten Kostenwert 8 in Zeile 5 erhöht werden, da ansonsten s_4 negativ würde. Also $\Delta_5 = \Delta$.
- Für v_6 sogar $\Delta_6 = 0$.
⇒ Änderung: $I' = \{1, \dots, 3\}$

Abbildung 4.6: Dual Ascent-Verfahren Bsp

Dual Ascent-Verfahren: Beispiel

- $I' \neq \emptyset \Rightarrow$ starte neue Iteration. Tableau nach $i = 1, \dots, 3$

$i \setminus j$	1	2	3	4	5	v_i	J_i	v_i	J_i	v_i	J_i	v_i	J_i
1	0	4	6	12	3	0	1	3	{1, 5}	4	{1, 2, 5}	4	{1, 2, 5}
2	2	6	3	4	0	0	5	2	{1, 5}	3	{1, 3, 5}	4	{1, 3, 4, 5}
3	4	9	10	12	8	4	1	8	{1, 5}	9	{1, 2, 5}	9	{1, 2, 5}
4	6	0	12	3	7	0	2	3	{2, 4}	6	{1, 2, 4}	6	{1, 2, 4}
5	9	12	0	3	8	0	3	3	{3, 4}	5	{3, 4}	5	{3, 4}
6	12	8	4	0	6	0	4	4	{3, 4}	4	{3, 4}	4	{3, 4}
s_j	12	6	9	9	12								
s_j	5	3	6	5	10								
s_j	2	0	4	0	7								
s_j	1	0	3	0	6								

- Lediglich v_2 konnte auf den viertgrößten Kostenwert 4 erhöht werden.
- Änderung: $I' = \{2\}$
- $I' \neq \emptyset \Rightarrow$ starte neue Iteration.
- Allerdings kann v_2 nicht weiter erhöht werden ($s_2 = 0$) \Rightarrow Stopp
- Lösung: $v = (4, 4, 9, 6, 5, 4)$ mit $RD(v) = 32$.

Abbildung 4.7: Dual Ascent-Verfahren Bsp

4. Konstruiere primär Lösung ausgehend von einer zulässige Lösung v von (RDP) mit **Konstruktionsheuristik** \rightarrow liefert die obere Schranke für das (WLP)

4.2.5 Konstruktionsheuristik

Algorithm 14 Konstruktionsheuristik

- 1: **Init:** setze $X = \emptyset$, berechne $J^+ = \{j \in J \mid s_j = 0\}$
 - 2: **Selektion von Standorten**
 - 3: Für $\forall i \in I$:
 - 4: **if** (Es gibt für Kunde i genau einen Standort $j \in J^+$ mit $c_{ij} \leq v_i$ und $j \notin X$) **then**
 - 5: Setze $X = X \cup \{j\}$, $y_j = 1$
 - 6: **else if** (Es gibt für Kunde i mehrere Standorte $j \in J^+$ mit $c_{ij} \leq v_i$ aber keines dieser $j \in X$) **then**
 - 7: setze $X = X \cup \{j^*\}$, $y_{j^*} = 1$ für das j^* mit $c_{ij^*} = \min\{c_{ij} \mid c_{ij} \leq v_i\}$
 - 8: **end if**
 - 9: **Zuordnung**
 - 10: Ordne jeden Kunden i dem kostengünstigsten Standort $j \in X$ zu: $x_{ij^*} = 1$ für das j^* mit $c_{ij^*} = \min\{c_{ij} \mid c_{ij} \leq v_i, j \in X\}$
- Output:primäre Lösung (obere Schranke für das (WLP))
-

Example 4.2.5.

Konstruktionsheuristik

Beispiel

Zulässige duale Lösung $v = (4, 4, 9, 6, 5, 4)$ des Dual Ascent-Verfahrens, mit $s = (1, 0, 3, 0, 6)$ und $RD(v) = 32$.

1. Initialisierung

$$X = \emptyset \text{ und } J^+ = \{j \in J \mid s_j = 0\} = \{2, 4\}.$$

2. Selektion von Standorten

Genau ein Standort $j \in J^+$ mit $c_{ij} \leq v_i$ für Kunde i ?

$$i = 1: \text{ lediglich } c_{12} \leq v_1 \Rightarrow X := \{2\},$$

$$i = 2: \text{ lediglich } c_{24} \leq v_2 \Rightarrow X := \{2, 4\} = J^+.$$

Können schon stoppen, da alle erlaubten Standorte bereits ausgewählt.

$$\Rightarrow y_2 = y_4 = 1 \text{ und } y_1 = y_3 = y_5 = 0.$$

3. Zuordnung

$$x_{12} = x_{24} = x_{32} = x_{42} = x_{54} = x_{64} = 1. \text{ Alle anderen } x_{ij} = 0.$$

$i \setminus j$	1	2	3	4	5	v_i
1	0	4	6	12	3	4
2	2	6	3	4	0	4
3	4	9	10	12	8	9
4	6	0	12	3	7	6
5	9	12	0	3	8	5
6	12	8	4	0	6	4
s_j	1	0	3	0	6	32

Abbildung 4.8: Konstruktionsheuristik Bsp

5. Optimalitätsbedingungen erfüllt?

- $s_j \geq 0 \Rightarrow y_j = 0; s_j = 0 \Rightarrow y_j > 0$
- $v_i < c_{ij} \Rightarrow x_{ij} = 0; v_i \geq c_{ij} \Rightarrow x_{ij} \geq 0$
- $v_i > c_{ij} \Rightarrow x_{ij} = y_j$

Falls die drei Bedingungen erfüllt sind \rightarrow Die beiden erhaltenen Lösung sind optimal

Sonst \rightarrow **Dual Adjustment Verfahren**, dann wieder Konstruktionsheuristik durchführen, bis die Optimalitätsbedingungen erfüllt werden.

4.2.5.1 Dual Adjustment-Verfahren

Voraussetzung

zulässige Lösungen v von (RDP) und (x, y) von (WLP).

Definiere:

- $J_i^x := \{j \in X \mid v_i > c_{ij}\}$
- $c_{ij^*} = \min\{c_{ij} \mid j \in J_i^x\}$
- $c_{ij^{**}} = \min\{c_{ij} \mid j \in J_i^x, j \neq j^*\}$
- $I_j^x = \{i \in I \mid j \in X \text{ ist der einzige Standort mit } v_i \geq c_{ij}\}, j \in X$

Algorithm 15 Dual Adjustment-Verfahren

Input: zulässige Lösungen v von (RDP) und (x, y) von (WLP). (Nur für die Spalten mit $s_j = 0$ durchführen)

1: **Iteration**2: Für $\forall i \in I$:3: Gilt $|J_i^x| \leq 1$ oder $I_{j^*}^x \cup I_{j^{**}}^x = \emptyset$, so fahre mit dem nächsten i fort.4: Bestimme das nächstkleinere $c_{ij} : \Delta := \max\{c_{ij} | c_{ij} < v_i, j \in J\}$ 5: Setze $s_j := s_j + (v_i - \Delta)$ für $\forall j \in J_i^x$ 6: Setze $v_i := \Delta$

7: Führe das Dual Ascent-Verfahren nacheinander aus mit

- $I' := I_{j^*}^x \cup I_{j^{**}}^x$
- $I' := I' \cup \{i\}$
- $I' := I$

Example 4.2.6.**Dual Adjustment-Verfahren****Beispiel**

Zulässige duale Lösung $v = (4, 4, 9, 6, 5, 4)$
mit $s = (1, 0, 3, 0, 6)$ und $RD(v) = 32$.

Zulässige Lösung (x, y) von (WLP):

- $y_2 = y_4 = 1$ und $y_1 = y_3 = y_5 = 0$
- $x_{12} = x_{24} = x_{32} = x_{42} = x_{54} = x_{64} = 1$.
Alle anderen $x_{ij} = 0$.

$\Rightarrow X = \{2, 4\}$, $F(x, y) = 35$.

Iteration

- Für $i = 1, \dots, 3$ gilt $|J_i^x| \leq 1$.
- $i = 4$: $|J_4^x| = 2$ ($J_4^x = \{2, 4\}$) und $I_2^x \cup I_4^x = \{1, 3\} \cup \{2, 5, 6\} \neq \emptyset$.
- **Nächstkleineres** c_{4j} :
 $\Delta := \max\{c_{4j} | c_{4j} < v_4, j \in J\} = c_{44} = 3$.
- $s_2 := s_2 + (v_4 - \Delta) = 0 + 6 - 3 = 3$, $s_4 = 3$, $v_4 = 3$.

$i \setminus j$	1	2	3	4	5	v_i
1	0	4	6	12	3	4
2	2	6	3	4	0	4
3	4	9	10	12	8	9
4	6	0	12	3	7	6
5	9	12	0	3	8	5
6	12	8	4	0	6	4
s_j	1	0	3	0	6	32

$J_i^x = \{j \in X | v_i > c_{ij}\}$
 $I_j^x = \left\{ \begin{array}{l} i \in I | j \in X \text{ ist der einzige} \\ \text{Standort mit } v_i \geq c_{ij} \end{array} \right\}$

Abbildung 4.9: Dual Adjustment-Verfahren Bsp

Dual Adjustment-Verfahren

Beispiel

Führe das Dual Ascent-Verfahren mit $I' = \{1, 2, 3, 5, 6\}$ basierend auf folgendem Tableau aus.

$i \setminus j$	1	2	3	4	5	v_i
1	0	4	6	12	3	4
2	2	6	3	4	0	4
3	4	9	10	12	8	9
4	6	0	12	3	7	3
5	9	12	0	3	8	5
6	12	8	4	0	6	4
s_j	1	3	3	3	6	29

Man erhält:

$i \setminus j$	1	2	3	4	5	v_i
1	0	4	6	12	3	5
2	2	6	3	4	0	4
3	4	9	10	12	8	9
4	6	0	12	3	7	3
5	9	12	0	3	8	8
6	12	8	4	0	6	4
s_j	0	2	0	0	5	33

Mit einer neuen dualen Lösung $v = (5, 4, 9, 3, 8, 4)$ und verbessertem Zielfunktionswert 33.

Weitere Schritte bringen keine Verbesserung.

Abbildung 4.10: Dual Adjustment-Verfahren Bsp

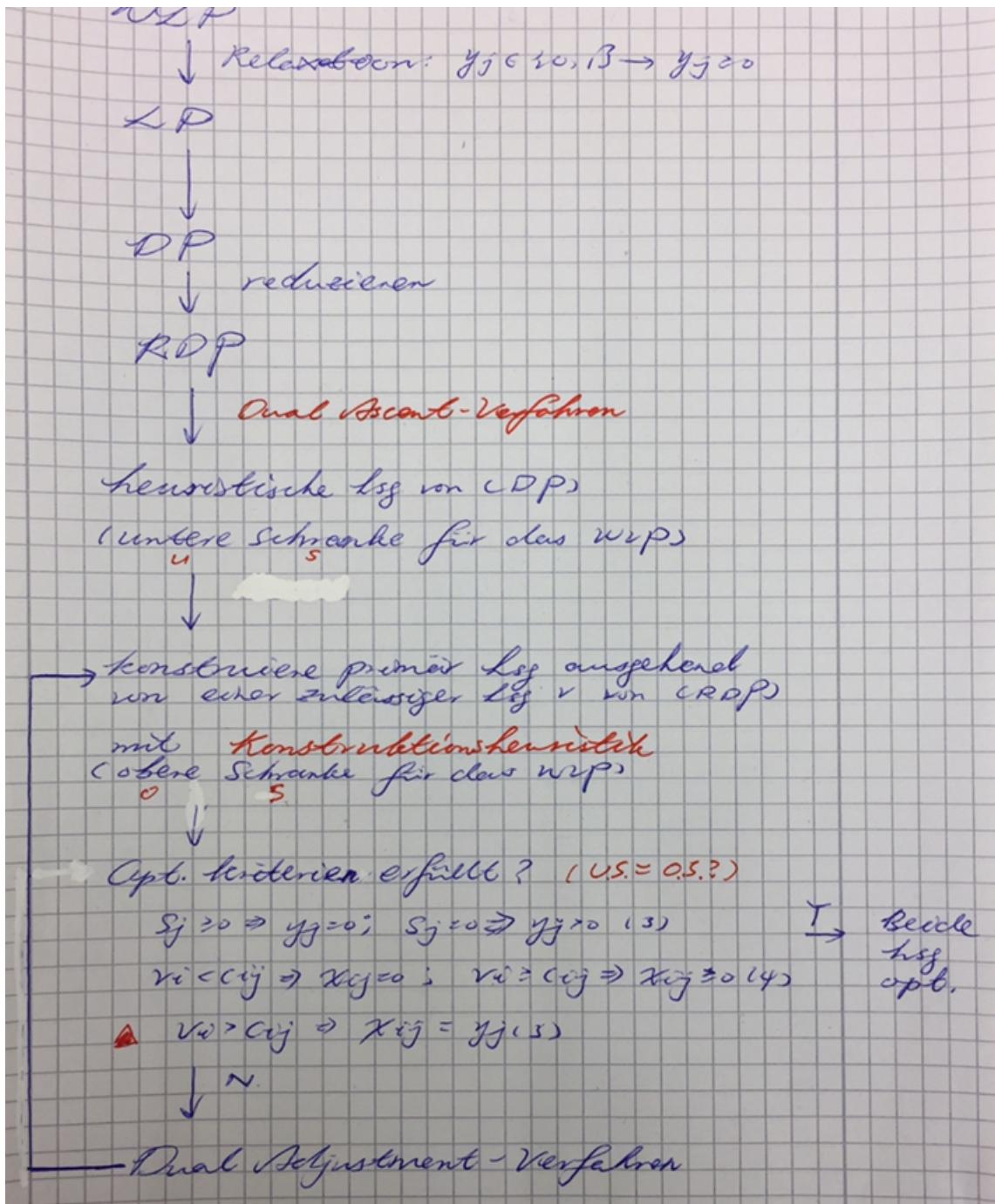


Abbildung 4.11: WLP

Note 2.

4.3 Hub-Location-Probleme

Kapitel 5

Gebietsplanung

Ziel: kleine geographische Einheiten (sogenannte Basisgebiete) zu über- geordneten Gebieten (häufig als Bezirke, Cluster oder Territorien bezeichnet) unter der Berücksichtigung verschiedener relevanter Planungskriterien zusammenzufassen.

5.1 Basismodell

5.1.1 Definitionen und Notationen

Ein Gebietsplanungsproblem umfasst eine Menge $V = \{1, \dots, M\}$ von Basisgebieten.

Ein **Basisgebiet** $i \in V$ ist durch seinen Mittelpunkt $b_i = (x_i, y_i)$ bestimmt.

Für jedes Basisgebiet $i \in V$ ist eine einzelne quantifizierbare Eigenschaft, das sogenannte **Aktivitätsmaß** w_i , gegeben.

Gebiet:

- Gebiete D_1, \dots, D_p sind disjunkte Teilmengen der Basismenge, so dass jedes Basisgebiet in genau einem Gebiet enthalten ist:

$$D_1 \cup \dots \cup D_p = V \quad \text{und} \quad D_i \cap D_j = \emptyset, i \neq j, b < p$$

- Aktivitätsmaß oder die Größe eines Gebiets: die Summe der Aktivitätsmaße seiner Basisgebiete

$$w(D_j) = \sum_{i \in D_j} w_i$$

- **Zentrum** des Gebiets j : c_j (Im Allgemeinen entspricht c_j einem der Mittelpunkte der zum Gebiet j gehörenden Basisgebiete.)

Zusammenhang:

Basisgebiet $b_i = (x_i, y_i) \in$ Menge der Basisgebiete $V \supseteq$ Gebiet D_j

5.1.2 Modell-Kriterien

Balance

Alle Gebiete sollen balanciert sein, also möglichst gleich groß bzw. stark bezüglich der Aktivitätsmaße der Gebiete.

- **perfekt balanciert** \Leftrightarrow sein Aktivitätsmaß entspricht dem durchschnittlichen Aktivitätsmaß aller Gebiete μ

$$w(D_j) = \mu = \frac{w(V)}{p} \quad (5.1)$$

Auf Grund der diskreten Struktur des Problems können perfekt balancierte Vertriebsgebiete im Allgemeinen NICHT erzielt werden.

- **relative Abweichung** des Aktivitätsmaßes eines Gebiet

$$bal(D_j) = \frac{|w(D_j) - \mu|}{\mu}$$

Kontiguität (Contiguity)

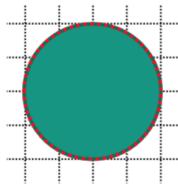
Zwei Basisgebiete werden als benachbart bezeichnet, wenn ihre geographischen Anordnungen nichtleere Schnittmengen besitzen.

Kompaktheit (Compactness)

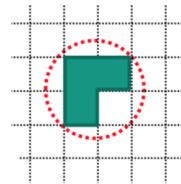
- Reock-Test: Bilde das Verhältnis der Fläche des Gebiets zur Fläche des kleinsten, das Gebiet umschließenden Kreises.

$$cp(D_j) = \frac{A_{D_j}}{A_{uK}} \leq 1$$

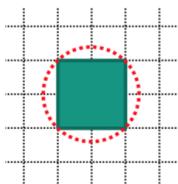
Beispiele – Reock-Test:



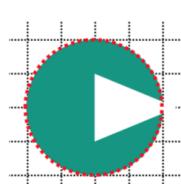
$$\begin{aligned} A_{D_j} &= 4 \cdot \pi \\ r_{uK} &= 2 \\ A_{uK} &= 4 \cdot \pi \\ cp(D_j) &= \frac{4 \cdot \pi}{4 \cdot \pi} = 1 \end{aligned}$$



$$\begin{aligned} A_{D_j} &= 3 \\ r_{uK} &= \sqrt{2} \\ A_{uK} &= 2 \cdot \pi \\ cp(D_j) &= \frac{3}{2 \cdot \pi} = 0,48 \end{aligned}$$



$$\begin{aligned} A_{D_j} &= 4 \\ r_{uK} &= \sqrt{2} \\ A_{uK} &= 2 \cdot \pi \\ cp(D_j) &= \frac{4}{2 \cdot \pi} = 0,64 \end{aligned}$$



$$\begin{aligned} A_{D_j} &= 4 \cdot \pi - 2 \\ r_{uK} &= 2 \\ A_{uK} &= 4 \cdot \pi \\ cp(D_j) &= \frac{4 \cdot \pi - 2}{4 \cdot \pi} = 0,84 \end{aligned}$$

Abbildung 5.1: Reock Test Bsp

- Schwartzberg-Test: Bilde das Verhältnis zwischen dem Umfang eines Kreises, der dadurch festgelegt ist, dass er den gleichen Flächeninhalt wie das Gebiet hat, und dem Umfangs des Gebiets.

$$cp(D_j) = \frac{2 \cdot \sqrt{\pi \cdot A_{D_j}}}{U_{D_j}} \leq 1$$

Beispiele – Schwartzberg-Test:

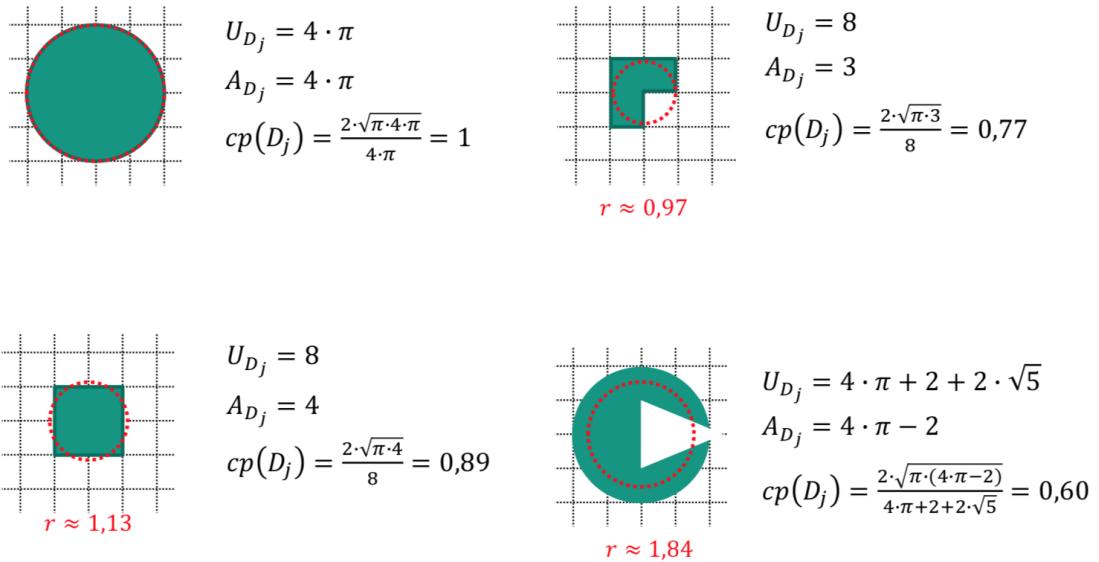


Abbildung 5.2: Schwartzberg-Test Bsp

Je näher $cp(D_j)$ an 1, umso kompakter ist das Gebiet.

5.1.3 Ziel der Gebietsplanung

Untergliedere alle Basisgebiete V in p Gebiete, welche die Planungskriterien der Balance, Kompaktheit, Kontiguität und Disjunktheit erfüllen.

5.2 Vorgehensweisen zur Gebietsplanung

5.2.1 Notation

- V : Menge der Basisgebiete
- w_u : Aktivitätsmaß des Basisgebiets u
- p : Anzahl der Gebiete
- d_{uv} : Distanzen
- μ : Durchschnittsgröße bzgl. w ((5.1))
- τ : Toleranz der Balance
- $x_{uv} = \begin{cases} 1 & \text{falls Basisgebiet } u \text{ einem Gebiet mit dem Zentrum } v \text{ zugeordnet wird.} \\ 0 & \text{sonst} \end{cases}$

5.2.2 LP-Formulierung

$$\begin{aligned}
 & \min_{u,v \in V} d_{uv}^2 w_u x_{uv} \\
 \text{s.t.} \quad & \sum_{v \in V} x_{uv} = 1 && (u \in V) \quad (\text{Vollst. Zuordnung}) \\
 & \mu(1 - \tau)x_{vv} \leq \sum_{u \in V} w_u x_{uv} \leq \mu(1 + \tau)x_{vv} && (v \in V) \quad (\text{Balance}) \\
 & \sum_{v \in V} x_{vv} = p && (p \text{ Gebiete}) \\
 & x_{uv} \in \{0, 1\} && (u, v \in V)
 \end{aligned}$$

5.3 Recursive-Partitioning-Algorithmus

(Bsp: Aufgabe 19)

Grundidee:

- Unterteile das Problem rekursiv auf geometrische Weise durch Linien in immer kleinere Teilprobleme.
- Wiederhole dies solange, bis eine elementare Größe erreicht ist, in welcher das Gebietsplanungsproblem in effizienter Zeit gelöst wird.

5.3.1 Definitionen

Partitionsproblem

$PP = (B, q)$ wird als Partitionsproblem bezeichnet, falls $B \subseteq V$ und $1 \leq q \leq p$ gilt.

Linienpartition

$LP = (B_l, B_r, q_l, q_r)$ wird als Linienpartition bezeichnet, falls:

1. $B_l \cup B_r = B$ und $B_l \cap B_r = \emptyset$
2. \exists Line $L : B_l = B \cap H^{\leq}(L)$ und $B_r = B \cap H^{>}(L)$
3. $q \leq q_l, q_r \leq q$ und $q_l + q_r = q$

5.3.2 Recursive-Partitioning

Gleichmäßige Aufteilung:

- $q_l = q_r = \frac{q}{2}$, falls q gerade
- $q_l = \frac{q-1}{2}, q_r = \frac{q+1}{2}$ und $q_l = \frac{q+1}{2}, q_r = \frac{q-1}{2}$, falls q ungerade

Balance:

$$bal(LP) = \max\left\{\frac{\left|\frac{w(B_L)}{q_l} - \mu\right|}{\mu}, \frac{\left|\frac{w(B_r)}{q_r} - \mu\right|}{\mu}\right\}$$

Partitionsposition:

Bestimme k' , so dass $\frac{w(B_l^{k'})}{q_l} < \frac{w(B)}{q}$ und $\frac{w(B_l^{k'+1})}{q_l} \geq \frac{w(B)}{q}$

$$k^* = \begin{cases} k' & \text{falls } \frac{w(B)}{q} - \frac{w(B_l^{k'})}{q_l} \leq \frac{w(B_l^{k'+1})}{q_l} - \frac{w(B)}{q} \\ k' + 1 & \text{sonst} \end{cases}$$

Kompaktheit:

$$cp(LP) = d(c_1 \cdot c_2) = l_2(c_1, c_2)$$

5.3.3 Algorithmus

Definiere:

$$k' : \frac{w(B_l^{k'})}{q_l} < \frac{w(B)}{q} \wedge \frac{w(B_l^{k'+1})}{q_l} \geq \frac{w(B)}{q}$$

$$k^* := \begin{cases} k', & \text{falls } \frac{w(B)}{q} - \frac{w(B_l^{k'})}{q_l} \leq \frac{w(B_l^{k+1})}{q_l} - \frac{w(B)}{q} \\ k' + 1, & \text{sonst} \end{cases}$$

Algorithm 16 Recursive-Partitioning-Algorithmus

Input: Anzahl Suchrichtungen K, β, V, p

- 1: Markiere Partitionsproblem $PP = (V, p)$ als ungelöst
- 2: **while** es gibt ungelöste Probleme **do**:
- 3: Wähle ein ungelöstes Problem $PP = (B, q)$
- 4: **if** $q = 1$ **then**
- 5: füge B der Lösungsmenge DL hinzu und markiere PP als gelöst
- 6: **end if**
- 7: **if** $q > 1$ **then**
- 8: Bestimme für jede Suchrichtung die Liniенpartition $LP(k^*)$ mit der besten Balance

$$bal(LP) = \max\left\{\frac{\left|\frac{w(B_l)}{q_l} - \mu\right|}{\mu}, \frac{\left|\frac{w(B_r)}{q_r} - \mu\right|}{\mu}\right\}$$

und füge sie einer Menge FLP hinzu.

- 9: Bewerte alle Liniенpartitionen LP der Menge FLP durch:

$$rk(LP) = \beta \frac{bal(LP)}{bal^{max}} + (1 - \beta) \frac{cp(LP)}{cp^{max}}$$

mit

$$cp(LP) = d(c_1, c_2) = l_2(c_1, c_2)$$

- 10: Wähle $LP^* = (B_l^*, B_r^*, q_l^*, q_r^*) = \min_{LP \in FLP} rk(FLP)$
- 11: Erstelle Partitionsprobleme $PP_l = (B_l^*, q_l^*)$ und $PP_r = (B_r^*, q_r^*)$ und markiere sie als ungelöst, markiere PP als gelöst.
- 12: **end if**
- 13: **end while**

Output: Gebietslayout DL
