P-Median Problem

Michael Enser

14.11.2011

Inhaltsverzeichnis

1 Allgemeines p-Median-Problem

Das p-Median-Problem lässt sich als binäres lineares Optimierungsproblem formulieren. Dazu führen wir zuerst einige Definitionen ein.

- p Anzahl der ausgewählten Standorte oder Mediane
- I Menge der möglichen Standorte, $I = \{1, \ldots, l\}$
- J Menge der möglichen Nachfolger, $J = \{1, \ldots, m\}$
- M eine genügend große Zahl $> l * m$
- d_{ij} Entfernung $d[i, j]$ vom Standort $i \in I$ zum Nachfrager $j \in J$
- b_j Bewertung des Nachfrage-Knotens $j \in J$
- x_{ij} 0/1-Entscheidungsvariable, die angibt, ob Nachfrager j vom Standort i bedient wird ($x_{ij} = 1$) oder nicht ($x_{ij} = 0$)
- y_i 0/1-Entscheidungsvariable, die angibt, ob Standort i ausgewählt wird ($y_i = 1$) oder nicht ($y_i = 0$)

Nun können wir das Optimierungsproblem definieren.

1.1 Allgemeine Defintion

Das p-Median-Problem lässt sich mit den obigen Definitionen als folgende Optimierungsproblem schreiben

$$\min z = \sum_{i \in I} \sum_{j \in J} b_j * d_{ij} * x_{ij} \tag{1}$$

und folgende Nebenbedingungen

$$\sum_{i \in I} x_{ij} = 1 \qquad \text{für alle } j \in J \qquad \text{(NB. 1)}$$

$$\sum_{j \in J} x_{ij} \leq M * y_i \qquad \text{für alle } i \in I \qquad \text{(NB. 2)}$$

$$\sum_{i \in I} y_i = p \qquad \text{(NB. 3)}$$

$$x_{ij}, y_i \in \{0, 1\} \qquad \text{für alle } i \in I, j \in J \qquad \text{(NB. 4)}$$

In der Nebenbedingung (NB. 1) wird geprüft, ob jeder Nachfrager j genau einem Standort i zugeordnet wird. In der Nebenbedingung (NB. 2) stellt man sicher, dass ein Standort i auch vorhanden ist, wenn ihm ein Nachfolger zugeordnet wird. Die Nebenbedingung (NB. 3) legt die Anzahl der Standorte, also die Anzahl der Mediane, fest. Allerdings ist dieses Optimierungsproblem grundsätzlich np-schwer lösbar.

1.2 Ein kleines Beispiel

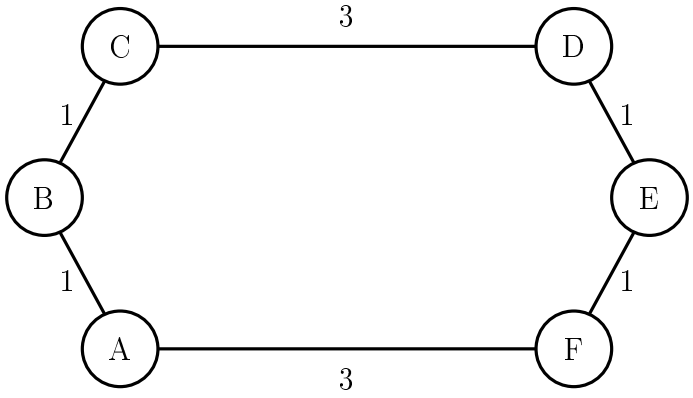

Abbildung 1: simples Beispiel

In dieser obigen Graphik sind die Knoten {A, B, C, D, E, F} Städte in denen Firmen liegen, welche mit bestimmten Materialien beliefert werden sollen. Um das Beispiel

sehr einfach zu halten habe ich hier den benötigten Bedarf für jede Firma gleich hoch angesetzt. Nun darf man 2 Standorte für Logistikzentren wählen, aus denen alle Firmen beliefert werden sollen. Da es sich hier um ein simples Beispiel handelt kann man schnell erkennen, dass man, wenn man die Logistikzentren in den Städten {B, E} wählt, eine ideale Lösung für das Optimierungsproblem. Somit würden die Firmen in den Städten {A, B, C} von dem Logistikzentrum in der Stadt B beliefert werden. Die Firmen in den Städten {D, E, F} von dem Logistikzentrum in der Stadt E beliefert werden.

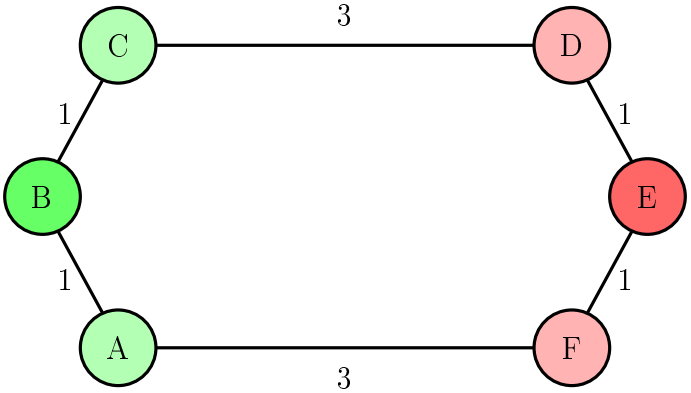

Abbildung 2: simples Beispiel

2 1-median Problem in einem Baum

Wie schon vorher angeführt, ist das allgemeine p-median Problem np-schwer lösbar. Betrachtet man allerdings einen Baum mit einem einzigen Median, so lässt sich ein effizienter Algorithmus zum Lösen des Optimierungsproblems finden. Dazu betrachten wir zuerst einen Sonderfall des 1-Median Problems.

Lemma 1. *Sei $G = (V, E)$ ein Graph. Weiters sei $J = \{1, \ldots, m\}$ die Menge der möglichen Nachfrager mit dem Bedarf b_j, $j \in J$. Sei $B = \sum_{j \in J} b_j$ und es existiere ein $x \in J$ mit $x > B/2$. Dann wähle man x als Median, um die ideale Lösung des Optimierungsproblems zu erhalten.*

Proof. Der Bedarf von x sei b_x mit $b_x > B/2$. Nun wähle man irgendeinen anderen Punkt als Standort, entweder einen anderen Knoten oder einen Punkt auf den Verbindungswegen zwischen den Knoten. Nun bewege man den Standort um eine Distanz von δ Richtung Knoten x. Daraus folgt

$$\text{Veränderung der Objektfunktion} = (B_1 - B_2) * \delta \tag{2}$$

Mit B_1 gleich der $\sum_{i \in A} b_i$, mit A gleich der Summe der Knoten, deren Distanz zum Standort sich um δ vergrößert, und B_2 gleich der $\sum_{j \in B} b_j$ mit B gleich der Summe der Knoten, deren Distanz zum Standort sich um δ verringert.
Da wir den Standort aber näher zu unserem Knoten x bewegen, kommt er auch mehr als der Hälfte des Bedarfs näher. Somit wird sich die Objektfunktion um zumindest $\delta(2 * b_x - B) \geq 0$ verringern.
Also haben wir gezeigt, dass umso näher wir dem Knoten x kommen, desto besser wird die Lösung der Optimierungsaufgabe. Daraus folgt natürlich, dass, wenn man den Standort gleich x wählt, die optimale Lösung erhält.

□

Diese Lösung funktioniert in jedem Graphen. Nun behandeln wir den Fall, dass der Graph ein Baum ist.

Natürlich gibt es auch noch den Fall, dass es keinen Knoten mit mehr als der Hälfte des Bedarfs gibt. Dazu beginnen wir wieder mit einem Beispiel.

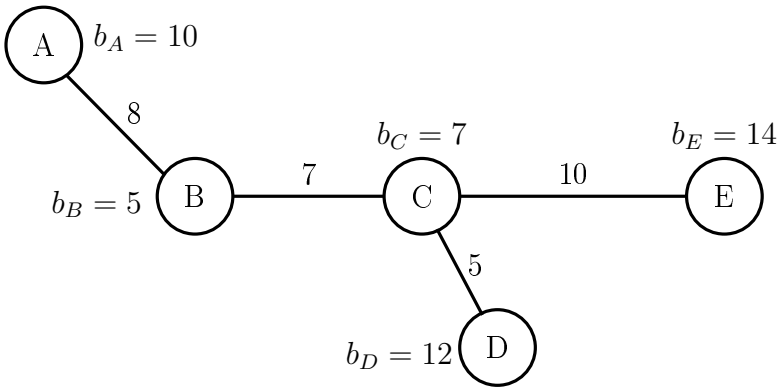

Abbildung 3: Beispiel mit Baum als Graphen

Nun suchen wir eine Lösung des 1-Median Problems für den obigen Graphen. Zuerst setzten wir den Standort des Medians zwischen den Knoten B und C fest. Nun sehen wir, dass sich ein 15-facher Bedarf $(b_A + b_B)$ links und ein 33-facher Bedarf $(b_C + b_D + b_E)$ rechts vom Standort befinden. Wenn wir Standort um δ näher zum Knoten C bewegen, würde das bedeuten, dass sich die Objektfunktion um 18 δ $[(33 - 15)\delta = 18\delta]$ verringert. Somit ist die beste Lösung für einen Punkt zwischen C und D genau dann gefunden, wenn man Knoten C wählt. Wenn wir jetzt von C weg um δ in Richtung Knoten D bewegen, wird die Objektfunktion um 24δ vergrößern. Als nächstes beobachten wir eine Verschiebung des Standorts vom Knoten C in Richtung Knoten E. Dadurch wird die Objektfunktion um 20 δ vergrößert.
Somit haben wir für dieses Beispiel wieder die ideale Lösung gefunden, indem wir C als Median wählen.

Aus diesem Beispiel könnten wir möglicherweise einen Algorithmus finden, mit dem man für jeden Baum eine ideale Lösung effizient berechnen kann. Wir haben also einen Graphen G=(V,E), welcher ein Baum ist. Zuerst betrachten wir die Blätter des Baumes. Davon muss es in jedem Baum mindestens 2 geben. O. B. d. A. nennen wir ein Blatt v_1. Jetzt rechnen wir den Bedarf von v_1 zu dessen Nachbarn v_2 dazu. Jetzt nennen wir G, ohne v_1 und ohne Kante (v_1, v_2), G_1. Der neue Graph G_1 ist natürlich wieder ein Baum. Dieses 'Knotenlöschen' wiederholt man sooft, bis ein Knoten mehr als die Hälfte des Gesamtbedarfs hat. Und somit hat man die optimale Lösung für den 1-Median gefunden.

Probieren wir das ganze an unserem Beispiel noch einmal aus.

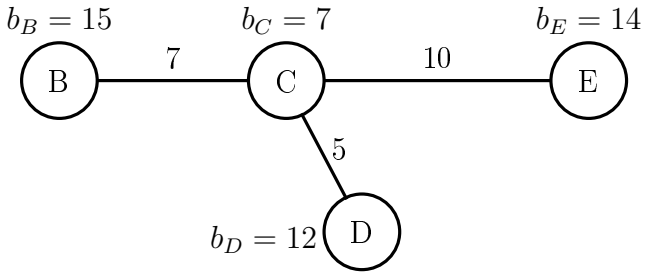

Abbildung 4: Nach der ersten Knotenlöschung

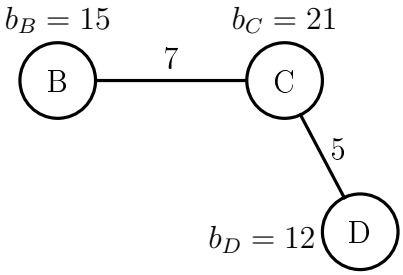

Abbildung 5: Nach der ersten Knotenlöschung

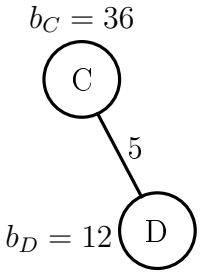

Abbildung 6: Nach der ersten Knotenlöschung

In Abbildung 4 wurde Bedarf von Knoten A zum Bedarf von Knoten B addiert und danach Knoten A gelöscht. Weiters wurde in Abbildung 5 der Bedarf von Knoten E zu C addiert und gelöscht. In Abbildung 6 wurde der aktuelle Bedarf von Knoten B, also der ursprüngliche Bedarf von Knoten A und B, zu C addiert und danach gelöscht. Und erst jetzt erhält man einen Knoten, nämlich C, der mehr als die Hälfte des Gesamtbedarfs hat. Und somit ist C unser 1-Median. Dieser Algorithmus hat eine Laufzeit von $O(n)$.

3 Quellennachweis

- http://www.fernuni.de/BWLQUAM/assets/courses/k00852-5.pdf

- Daskin [6], S. 198–208