Das Briefträgerproblem

Paul Tabatabai

30. Dezember 2011

Inhaltsverzeichnis

1 Problemstellung und Modellierung

1.1 Problem

Das Problem wurde erstmals 1962 von dem chinesischen Mathematiker Mei Ko Kwan untersucht, der ihm den Namen *Chinese Postman Problem* gab.
Ein (chinesischer) Briefträger soll über den kurzmöglichsten Weg alle Straßen einer Stadt besuchen, und am Ende wieder an seinem Startpunkt angelangen.

1.2 Modellierung

Für die Modellierung des Problems betrachtet man einen zusammenhängenden ungerichteten Graphen $G = (V, E)$, wobei die Kantenmenge $E(G)$ die Straßen der Stadt sind und die Knotenmenge $V(G)$ die Kreuzungen der Straßen darstellen.
Die Längen der Straßen werden durch eine Gewichtsfunktion der Kanten modelliert: $w : E \to \mathbb{R}_{\geq 0}$.
Gesucht ist ein minimaler (minimal bezüglich der Kanten-Gewichte) Zyklus, der jede Kante mindestens einmal enthält.

2 Lösung

Die Lösung des Problems ist von dem modellierten Graphen abhängig. Für eulersche Graphen und Bäume gibt es eine triviale Lösung. Für beliebige Graphen stellen wir ein Lösungsverfahren vor und beweisen dessen Richtigkeit.

2.1 Lösung für eulersche Graphen

Ein Graph heißt *eulersch*, wenn er eine geschlossene Tour enthält, die jede Kante genau einmal enthält. Eine Tour heißt *geschlossen*, wenn sie wieder an ihrem Startpunkt endet (so wie es auch bei unserem Problem verlangt wird). In eulerschen Graphen ist der gesuchte Zyklus genau die Euler-Tour des Graphen. Gefordert ist ein Zyklus, der jede Kante *mindestens* einmal enthält, also ist die Euler-Tour, die jede Kante *genau* einmal enthält, auf jeden Fall minimal bezüglich der Kantengewichte.

2.2 Lösung für Bäume

Mit Induktion lässt sich schnell zeigen, dass in Bäumen jede Kante genau zwei mal in dem gesuchten Zyklus vorkommt.

2.3 Generelle Lösung (nach [8])

Zunächst betrachten wir die theoretische Lösung des Briefträgerproblems. Dann stellen wir einen konkreten Algorithmus vor, mit dem diese Lösung erzielt werden kann und beweisen dessen Korrektheit.

2.3.1 Theoretische Lösung

Sei X die Menge der Knoten aus G mit ungeradem Grad.
Wir fügen eine neue Menge von Kanten E' zu dem Graphen hinzu, so dass folgende drei Bedingungen erfüllt sind:

a) Jede Kante $e' \in E'$ ist parallel zu einer Kante $e \in E$. (Das heißt, die Kante e' verläuft zwischen den gleichen Knoten wie e und $w(e') = w(e)$)

b) In $G(V, E')$ haben genau die Knoten aus X ungeraden Grad.

c) $w(E')$ ist minimal; Es gilt $w(E') \leq w(E'')$ für alle Kantenmengen E'', die 1. und 2. erfüllen.

Der so konstruierte Graph $G(V, E \cup E')$ ist ein *eulerscher* Multigraph (d.h. ein eulerscher Graph, in dem Mehrfachkanten erlaubt sind). Jeder Euler-Tour dieses Graphen hat die *minimale* Länge $w(E)+w(E')$. Wenn man sich vor Augen führt, dass das Konstruieren eines solchen eulerschen Multigraphen äquivalent zur Frage ist, welche Kanten (Straßen) mehrfach durchlaufen werden müssen, ist klar, dass sich jede Lösung des Problems so darstellen lässt.

2.3.2 Algorithmus

Wir stellen jetzt den Algorithmus für die Lösung vor. Sei $G(V, E)$ ein zusammenhängender Graph mit Gewichtsfunktion $w : E \to \mathbb{R}_{\geq 0}$. Sei weiters $d_G : V \times V \to \mathbb{R}_{\geq 0}$ die Funktion, die jedes 2-Tupel von Knoten auf die Länge des kürzesten Pfades in G zwischen den Knoten abbildet.

---

1. Bestimme $X := \{v \in V : deg(v) \text{ ungerade}\}$.
2. Bestimme $d_G(x, y)$ $\forall x, y \in V$.
3. Sei H der vollständige Graph über X. Bestimme ein perfektes Matching M in H mit minimalem Gewicht bezüglich d_G.
4. Bestimme für alle $(x, y) \in M$ einen kürzesten Weg W_{xy} zwischen x und y in G. Füge dann für jede Kante in W_{xy} eine neue Kante mit selbem Gewicht zu G hinzu. Sei G' der so definierte Multigraph.
5. Bestimme eine Euler-Tour C' in G'. Ersetze jede Kante aus C', die nicht in G enthalten ist, durch die dazugehörige parallele Kante in G mit selbem Gewicht. Sei C die so konstruierte Euler-Tour in G.

Genauere Erklärungen der Teilschritte folgen im nächsten Abschnitt, zunächst zeigen wir die Korrektheit des Algorithmus.

2.3.3 Korrektheit des Algorithmus

In Schritt 4. wird für jedes Matching M von H (der vollständige Graph über X mit Gewichtsfunktion d_G) eine Knotenmenge E' zu G hinzugefügt, so dass die Bedingungen a) und b) aus der theoretischen Lösung erfüllt sind.
Der so entstehende Zyklus hat die Länge $w(E) + d_G(M)$; deshalb ist es klar,

dass das Matching M aus Schritt 3. minimal sein soll. Um zu zeigen, dass die Minimalitätsbedingung c) aus der Theoretischen Lösung erfüllt ist, muss allerdings noch gezeigt werden, dass sich keine andere Kantenmenge E' finden lässt, mit der sich ein noch kleinerer Zyklus konstruieren lässt. Dafür brauchen wir das folgende Lemma und den folgenden Satz (bewiesen von D. Jungnickel in [1]):

Lemma 1. *Sei $G(V, E)$ ein zusammenhängender Graph mit Gewichtsfunktion $w : E \to \mathbb{R}_{\geq 0}$. Sei $X \subseteq V$ mit $|X|$ gerade und H der vollständige Graph über X. Die Kanten von H haben die Gewichte $d_G(x, y)$.*
Für jedes minimale, perfekte Matching M von H und für jedes E_0 mit $d_{(V,E)}(a, b) = d_{(V,E_0)}(a, b)$ $(\forall a, b \in X :)$ gilt:

$$d_G(M) \leq w(E_0)$$

Beweis. Sei $M = \{x_1y_1, \ldots, x_2y_2\}$ ein minimales perfektes Matching von H. Dann ist $d_G(M) = d_G(x_1, y_1)+\cdots+d_G(x_n, y_n)$. Sei weiters P_i ein kürzester Pfad von x_i zu y_i in (V, E_0). Dann gilt aufgrund unserer Forderung an E_0: $w(P_i) = d_G(x, y)$. Wir wollen nun zeigen, dass keine Kante e in mehr als einem der Pfade P_i enthalten sein kann. Daraus folgt dann die Aussage unseres Lemmas.
Annahme: Es existiert eine Kante e, die in zwei Pfaden (P_1 und P_2) enthalten ist. P_1 und P_2 haben dann folgende Form:

$$P_1 = (x_1, P_1', u, e, v, P_1'', y_1)$$
$$P_2 = (x_2, P_2', u, e, v, P_2'', y_2)$$

Es folgt:

$$\begin{aligned}
&d_G(x_1, y_1) + d_G(x_2, y_2) = \\
&d_G(x_1, u) + w(e) + d_G(v, y_1) + d_G(x_2, u) + w(e) + d_G(v, y_2) \ > \\
&d_G(x_1, u) + d_G(x_2, u) + d_G(v, y_1) + d_G(v, y_2) \ \geq \\
&d_G(x_1, x_2) + d(y_1, y_2)
\end{aligned}$$

Das ist aber ein Widerspruch zu der Annahme, dass unser Matching M minimal ist, denn würde man in M $\{x_1y_1, y_2y_2\}$ durch $\{x_1x_2, y_1y_2\}$ ersetzen, erhielte man ein Matching von kleinerem Gewicht.

□

Satz. *Der Algorithmus zum Lösen des Briefträgerproblems ist korrekt.*

Beweis. Sei M ein minimales perfektes Matching von (H, d_G), wobei H der vollständige Graph über der Menge von Knoten auf G mit ungeradem Grad ist. Dann liefert unser Algorithmus einen Zyklus der Länge $w(E) + d_G(M)$. Sei E' eine Menge von Kanten, die die Bedingungen a), b) und c) unserer theoretischen Lösung erfüllt. Der so konstruierbare Zyklus hat dann die Länge $w(E) + w(E')$. Wir müssen zeigen, dass $w(E') \geq d_G(M)$ gilt.
Sei Z eine Zusammenhangskomponente von (V, E') mit mindestens zwei Knoten und X wie üblich die Menge der Knoten aus G mit ungeradem Grad. Der Schnitt $Z \cap X$ kann dann nicht leer sein, denn sonst könnten wir alle Kanten aus E', die in Z enthalten sind einfach weglassen, und erhielten so eine kleinere

Kantenmenge, die immer noch a) und b) erfüllt.
Da X die Menge von Knoten von (V, E') mit ungeradem Grad ist, ist $X \cap Z$ ebenfalls eine Menge von Knoten mit ungeradem Grad. Nach dem *Handschlagslemma* ist also $|Z \cap X|$ gerade.
Das bedeutet, dass die Zusammenhangskomponenten $(Z_1, \ldots, Z_k)$ von (V, E') eine Zerlegung $(X_1, \ldots, X_k)$ von X induzieren, so dass für alle $i \in \{1, \ldots, k\}$ gilt: $|X_i|$ ist gerade und zwei beliebige Knoten von X_i sind immer durch einen Pfad in (V, E') verbunden.

Seien $x, y \in X_i$ und sei P_{xy} der Pfad von x bis y in E'. Dann ist P_{xy} trivialerweise ein kürzester Pfad zwischen x in y in (V, E'). P_{xy} ist aber auch ein kürzester Pfad zwischen x in y in G; Angenommen, es gäbe einen kürzeren Pfad Pxy' zwischen den zwei Knoten. Dann würde eine Kantenmenge E'', in der alle Kanten aus Pxy durch die Kanten aus Pxy' ersetzt wurden immer noch a) und b) erfüllen und wäre von kleinerem Gewicht.

Sei E'_i die Menge von Kanten aus E', so dass beide Endknoten der Kanten in Z_i (Zusammenhangskomponente, die X_i induziert) enthalten sind. Sei H_i nun der vollständige Graph über Z_i mit Gewichtsfunktion $d_{(Z_i, E'_i)}(x, y)$ und sei M_i ein minimales perfektes Matching von H_i. Nach unserem Lemma gilt $d_{(Z_i, E'_i)}(M_i) \leq w(E'_i)$. Die Vereinigung aller Matchings $M_1 \cup M_2 \cup \cdots \cup M_k$ ist ein perfektes Matching von H und $E' = E'_1 \cup E'_2 \cup \cdots \cup E'_k$. Daraus folgt:

$$w(E') = w(E'_1) + \cdots + w(E'_k) \geq d_G(M_1) + \cdots + d_G(M_k) \geq d_G(M)$$

□

3 Verwendete Algorithmen

3.1 Kürzeste Pfade: Algorithmus von Floyd-Warshall

Der *Algorithmus von Floyd-Warshall* bestimmt die Länge eines kürzesten Pfades zwischen allen zwei Knoten x, y eines Graphen mit positiven Kantengewichten w_{ij} (Wenn ij keine Kante des Graphen ist, setze $w_{ij} = \infty$). Sei $V = \{1, ..., n\}$.

Algorithm 1 Floyd-Warshall

```
for i = 1 to n do
  for j = 1 to n do
    if i ≠ j then
      d(i, j) ← w_ij
    else
      d(i, j) ← ∞
    end if
  end for
end for

for k = 1 to n do
  for i = 1 to n do
    for j = 1 to n do
      if d(i, k) + d(j, k) < d(i, j) then
        d(i, j) ← d(i, k) + d(j, k))
      end if
    end for
  end for
end for
```

Der Algorithmus kann so modifiziert werden, dass er neben der Längen der kürzesten Pfade ebenfalls die Pfade selbst liefert. Man muss sich dazu lediglich für alle Knotenpaare x, y den Knoten mit höchstem Index merken, den man durchlaufen muss, wenn man von x über einen kürzesten Pfad y erreichen möchte. Diesen Knoten nennen wir B_{xy}. Der Algorithmus sieht dann so aus (Die Initialsierungsschleife wird hier nicht wieder angeführt):

```
for k = 1 to n do
  for i = 1 to n do
    for j = 1 to n do
      if d(i, k) + d(j, k) < d(i, j) then
        d(i, j) ← d(i, k) + d(j, k))
        B_ij ← k
      end if
    end for
  end for
end for
```

Um dann den Pfad P_{ij} zu konstruieren, benutzt man folgendes rekursive Verfahren:

Verfahren 2 getPath(i,j)

if $d(i,j) = \infty$ **then**
  return "kein Pfad zwischen i und j."
end if
$B \leftarrow B_{ij}$
if $B =$ null **then**
  return " " /* $\{i,j\}$ ist eine Kante.
else
  return GetPath$(i,B) + B +$ GetPath(B,j)
end if

3.2 Minimales perfektes Matching: Edmonds Matching Algorithmus

Edmonds' Algorithmus liefert ein minimales perfektes Matching für vollständige Graphen mit gerader Knotenzahl. Da der Algorithmus fortgeschrittene Kenntnisse aus der Optimierung benötigt, sei hier nur auf [8] verwiesen.

3.3 Euler-Tour: Algorithmus von Hierholzer

Mit dem *Algorithmus von Hierholzer* lässt sich eine Euler-Tour in einem eulerschen Graphen $G(V,E)$ bestimmen. Der Algorithmus benutzt die Tatsache, dass sich eulersche Graphen in paarweise Kantendisjunkte Zyklen zerlegen lassen.

Algorithm 3 Hierholzer

1. Konstruiere ausgehend von einem beliebigen Knoten v einen Kreis K in G, der jede Kante höchstens einmal enthält.
2. Wenn K ein Euler-Kreis breche ab, sonst:
3. Vernachlässige alle Kanten aus K.
4. Suche in in den Knoten aus K einen Knoten, dessen Grad größer als 0 ist. Starte von diesem Knoten aus einen neuen Kreis K', der keine Knoten aus K enthält und jede Kante aus G höchstens einmal enthält.
5. Füge den Kreis K' in den Kreis K ein: Ersetze den Startpunkt von K' in K durch alle Punkte von K' (in korrekter Reihenfolge). Nenne den so konstruierten Kreis wieder K und gehe zu Schritt 2.

4 Beispiele

4.1 Beispiel 1

Wir betrachten den Graphen G aus Abbildung 1.

Die Knoten aus G mit ungeradem Grad sind $\{a,b,c,d,h,i\}$. Sei H der vollständige Graph über diesen Knoten mit Gewichtsfunktion d_G. Wir suchen jetzt ein minimales perfektes Matching von H.
Man sieht schnell, dass unser gesuchtes Matching jenes aus Abbildung 2 ist.

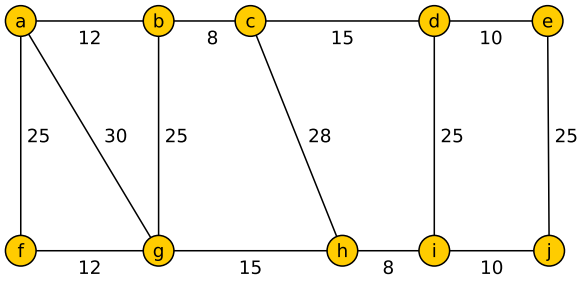

Abbildung 1: Ungerichteter Graph mit positiven Kantengewichten

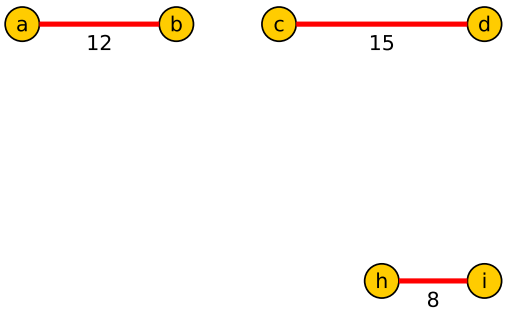

Abbildung 2: $M = \{ab, cd, hi\}$

Für jedes $xy \in M$ fügen wir jetzt für jede Kante aus dem kürzesten Pfad P_{xy} (In unserem Fall ist das genau eine Kante) eine parallele Kante zu G hinzu und erhalten so den Graphen G' in Abbildung 3.

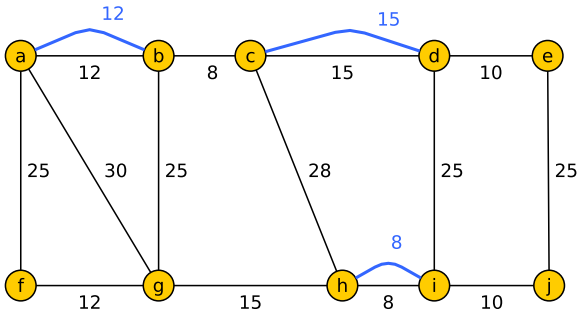

Abbildung 3: Eulerscher Multigraph G'

In diesem Multi-Graphen G' suchen wir jetzt eine Euler-Tour C' und ersetzen darin jede Kante aus C', die nicht in G ist, durch die dazugehörige parallele Kante in G. So erhalten wir den gewünschten Euler-Zyklus.

Informell kann man auch sagen, die Mehrfachkanten geben an, welche Straßen der Briefträger zwei mal durchlaufen muss.

4.2 Beispiel 2

Wir betrachten den Graphen G aus Abbildung 4.

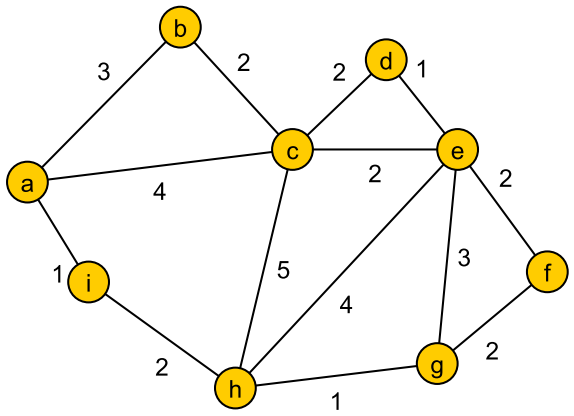

Abbildung 4: Ungerichteter Graph mit positiven Kantengewichten

Die Knoten aus G mit ungeradem Grad sind $\{a, c, e, g\}$. Sei H wieder der vollständige Graph über diesen Knoten mit Gewichtsfunktion d_G. Das Matching M in Abbildung 5 ist ein minimales perfektes Matching von H.

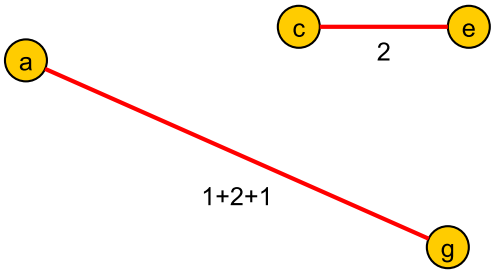

Abbildung 5: $M = \{ag, ce\}$

Für jedes $xy \in M$ fügen wir jetzt für jede Kante aus dem kürzesten Pfad P_{xy} die entsprechende parallele Kante zu G hinzu und erhalten so den Multigraphen G' in Abbildung 6.

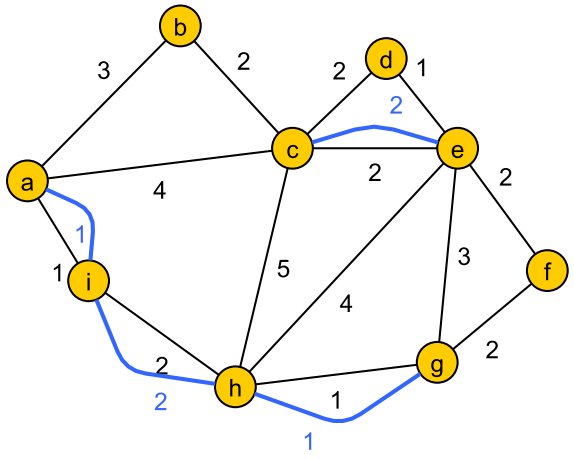

Abbildung 6: Eulerscher Multigraph G'

Aus G' können wir wie im vorherigen Beispiel den gesuchten Zyklus konstruieren.

Literatur

[1] Dieter Jungnickel: Graphs, Networks and Algorithms, Springer 2003

[2] http://de.wikipedia.org/wiki/Briefträgerproblem

[3] http://en.wikipedia.org/wiki/Floyd-Warshall_algorithm

[4] http://en.wikipedia.org/wiki/Eulerian_path

[5] http://web.mit.edu/urban_or_book/www/book/chapter6/6.4.4.html

[6] http://de.wikipedia.org/wiki/Algorithmus_von_Hierholzer

[7] http://de.wikipedia.org/wiki/Eulerkreisproblem

[8] B. Korte und J. Vygen: Kombinatorische Optimierung: Theorie und Algorithmen, Springer 2008