Technische Universität München
Zentrum Mathematik

Diskrete Optimierung: Fallstudien aus der Praxis

Barbara Wilhelm | Michael Ritter

Station 2: Separation von Subtour Elimination Constraints

Diskutieren Sie folgende Fragen in der Gruppe und tragen Sie Ihre Antworten auf dem Arbeitsblatt ein. Aufgaben a) – d) sind Pflichtaufgaben, alle anderen sind optional (wenn Sie noch Zeit haben).

1 Problembeschreibung

Wir betrachten das *Traveling Salesman-Problem*.

Problem: Traveling Salesman (TSP)
Input: Ein Graph $G = (V, E)$ mit $n \in \mathbb{N}$ Knoten und $m \in \mathbb{N}$ Kanten, eine Gewichtsfunktion $c : E \to \mathbb{N}_0$.

Aufgabe: Bestimme einen (bezüglich c) kürzesten Hamilton-Kreis in G oder stelle fest, dass G keine Hamilton-Kreis enthält. Ein Hamilton-Kreis ist ein Kreis in G, der jeden Knoten aus V genau einmal durchläuft. Seine Länge ist die Summe aller Kantenlängen der im Kreis enthaltenen Kanten.

2 TSP-Polytop

Für eine Instanz $(G = (V, E), c)$ des TSP sei $\mathcal{T}$ die Menge aller Hamilton-Kreise („Touren“) im Graphen G. Zu einer Tour $\tau \in \mathcal{T}$ definieren wir einen charakteristischen Vektor $x^{(\tau)} \in \{0, 1\}^m$ wie folgt:

$$x_{ij}^{(\tau)} = \begin{cases} 1, & \text{falls } \tau \text{ die Kante } \{i, j\} \text{ benutzt,} \\ 0, & \text{sonst.} \end{cases}$$

Das *TSP-Polytop* P_{TSP} ist dann die konvexe Hülle aller zu Touren gehörenden charakteristischen Vektoren:

$$P_{TSP} := \operatorname{conv}\left\{x^{(\tau)} : \tau \in \mathcal{T}\right\} \subseteq [0, 1]^m$$

3 ILP-Formulierung

Für eine ILP-Formulierung des TSP verwenden wir binäre Variablen $x_{ij} \in \{0, 1\}$, die für jede Kante $\{i, j\}$ angeben, ob die Kante in der Lösung enthalten ist oder nicht.

$$x_{ij} = \begin{cases} 1, & \text{falls die Kante } \{i, j\} \text{ in der Lösung benutzt wird,} \\ 0, & \text{sonst.} \end{cases}$$

Diese Formulierung ist analog zu den charakteristischen Vektoren $x^{(\tau)}$ von oben, wir müssen allerdings die Bedingung „$x \in \mathcal{T}$" in das ILP hineincodieren. Zunächst einmal muss eine Tour jeden Knoten genau einmal besuchen, es gibt also für jeden Knoten genau zwei mit diesem Knoten inzidente Tourkanten (eine Kante „in den Knoten hinein" und eine Kante „aus dem Knoten heraus"). Das führt zur sogenannten *Gradbedingung*

$$\sum_{i \in V : \{i,v\} \in E} x_{iv} = 2 \quad \forall\, v \in V.$$

Diese Bedingung schließt aber nicht aus, dass die Lösung in mehrere disjunkte Kreise zerfällt, sogenannte Subtouren. Um das zu vermeiden, setzt man die sogenannten *Subtour Elimination Constraints* ein:

$$\sum_{e \in \delta(S)} x_e \geq 2 \quad \forall\, S \subseteq V, 1 \leq |S| \leq n-1,$$

wobei $\delta(S)$ die Menge aller Kanten ist, die einen Knoten in S mit einem Knoten außerhalb verbinden, also

$$\delta(S) := \{e \in E : |e \cap S| = 1\}.$$

Anmerkung: Vielleicht kennen Sie die Subtour Elimination Constraints in der Form $\sum_{e \in E(S)} x_e \leq |S| - 1$. *Die oben verwendete Formulierung ist äquivalent dazu.*

a) Begründen Sie anschaulich, warum die Subtour Elimination Constraints für P_{TSP} gültige Ungleichungen sind.

Die Formulierung $\sum_{e \in E(S)} x_e \leq |S| - 1$ lässt sich anschaulich so verstehen, dass die Lösung keine Subtour enthalten darf, also keinen Kreis, der weniger als n Knoten besucht. Enthielte die Lösung zwischen den Knoten einer echten Teilmenge $S \subsetneq V$ (also $|S| \leq n-1$) mindestens $|S|$ viele Kanten, so ergäbe dies zwangsläufig einen Kreis innerhalb von S und damit auf $< n$ Knoten. Da wir dies für jede echte Teilmenge von V ausschließen, kann die Lösung keine Subtouren enthalten.

Die Formulierung $\sum_{e \in \delta(S)} x_e \geq 2$ kann man auch direkt interpretieren: Für jede echte Untermenge $S \subsetneq V$ der Knotenmenge muss es jeweils eine Kante geben, auf der die Tour in diese Menge hineinläuft, und eine weitere Kante, auf der die Tour die Menge wieder verlässt (es kann natürlich auch mehrere solcher Kanten geben, die Tour muss die Knoten in S ja nicht hintereinander durchlaufen). Damit wird sichergestellt, dass S an die Tour „angebunden" ist, also keine unabhängige Subtour darstellen kann.

---

b) Wieviele Subtour Elimination Constraints gibt es?

Es gibt $2^n - 2$ Subtour Elimination Constraints, also für jede Teilmenge S von V eine, nur die Teilmengen $\emptyset$ und V sind dabei per Definition ausgeschlossen. Natürlich kann man diese Menge noch etwas weiter reduzieren, so führen z. B. eine Menge und ihr Komplement jeweils zur identischen Subtour Elimination Constraint. Entscheidend ist aber, dass es exponentiell viele Bedingungen sind, und das ändert sich auch mit solchen Überlegungen leider nicht.

Insgesamt lässt sich das Traveling Salesman-Problem folgendermaßen als ILP schreiben:

$$\begin{aligned}
\min \sum_{\{i,j\}\in E} & c_{ij}x_{ij} \\
\sum_{i\in V:\{i,v\}\in E} & x_{iv} = 2 \quad \forall\, v \in V \\
\sum_{e\in\delta(S)} & x_e \geq 2 \quad \forall\, S \subseteq V, 1 \leq |S| \leq n-1 \\
& x \in \{0,1\}^m
\end{aligned}$$

Da es sehr viele Subtour Elimination Constraints gibt, wird das ILP für mittlere und große TSP-Instanzen viel zu groß (es wäre selbst auf Hochleistungsrechnern nicht mehr speicher-, geschweige denn rechenbar). Da man eigentlich nur an einer optimalen Ecke von P_{TSP} interessiert ist, benötigt man üblicherweise nicht alle Subtour Elimination Constraints (zur Bestimmung einer optimalen Ecke von P_{TSP} genügen ja $\dim(P_{TSP})$ viele aktive Nebenbedingungen). Aus diesem Grund geht man bei der Lösung von TSP oft so vor: Man beginnt mit der Relaxation

$$\begin{aligned}
\min \sum_{\{i,j\}\in E} & c_{ij}x_{ij} \\
\sum_{i\in V:\{i,v\}\in E} & x_{iv} = 2 \quad \forall\, v \in V \\
& x \leq \mathbb{1} \\
& x \geq 0.
\end{aligned}$$

Ist die Optimallösung dieser Relaxation nicht ganzzahlig oder enthält sie Subtouren, so sucht man eine verletzte Subtour Elimination Constraint und fügt sie als Schnittebene hinzu. Damit fährt man fort, bis entweder eine ganzzahlige Optimallösung gefunden wird, oder bis es keine verletzte Subtour Elimination Constraint mehr gibt.

4 Separation

An dieser Station wollen wir uns mit dem Fall beschäftigen, dass die Lösung (ganzzahlig oder nicht) eine Subtour Elimination Constraint verletzt. Wie findet man in diesem Fall eine Menge S, so dass die zugehörige Bedingung eine verletzte Schnittebene darstellt und somit die aktuelle Lösung abschneidet?

c) Sei x^* eine Lösung der obigen Relaxation. Formulieren Sie folgendes *Separationsproblem für Subtour Elimination Constraints* als ILP: Finde eine Menge $S \subseteq V$, so dass x^* die zu S gehörige Subtour Elimination Constraint verletzt (oder melde, dass es keine verletzte Subtour Elimination Constraint gibt).

Zu Beginn ein Hinweis: Die Lösung x^* der Relaxation muss nicht unbedingt ganzzahlig sein! Man kann die Komponenten von x^* also nicht als „Kante wird verwendet“ oder „Kante wird nicht verwendet“ interpretieren, wie das einige versucht haben. Ist x^* zufällig ganzzahlig, so kann man natürlich auch einfacher nach einer passenden Menge S suchen, der hier beschriebene Ansatz funktioniert aber auch für fraktionelle Lösungen (und die sind leider eher die Regel als die Ausnahme).

Gesucht ist eine Menge $S \subseteq V$, so dass $\sum_{e \in \delta(S)} x_e^* \geq 2$, oder eine Bestätigung, dass keine solche Menge existiert. Um das Problem als ILP zu formulieren, führen wir Indikatorvariablen $s_i \in \{0, 1\}$, $i \in \{1, \dots, n\}$ mit $n = |V|$, ein. Dabei bedeutet $s_i = 1$, dass der Knoten i in S liegt, $s_i = 0$ entsprechend, dass $i \notin S$.

Wir müssen mit Hilfe dieser Variablen jetzt die Summe $\sum_{e \in \delta(S)} x_e^*$ formulieren, also die Kanten in $\delta(S)$ „herausfiltern". Dazu führen wir einen zweiten Satz von Indikatorvariablen $d_e \in \{0, 1\}$ für alle $e \in E$ ein. Es soll genau dann $d_e = 1$ gelten, wenn $e \in \delta(S)$ für die durch die s_i-Variablen beschriebene Menge S. Die Summe lässt sich dann als

$$\sum_{e \in E} x_e^* d_e$$

schreiben (das ist linear, weil x^* ein konstanter Wert ist).

Natürlich müssen wir die Variablen s und d noch miteinander verknüpfen. Für jede Kante $e = \{i, j\} \in E$ soll gelten: $d_e = 1 \Leftrightarrow s_i \neq s_j$. Das kann man in Ungleichungsform auch als $d_e \geq |s_i - s_j|$ schreiben oder linear in der Form

$$\begin{aligned} d_{ij} &\geq s_i - s_j, \\ d_{ij} &\geq s_j - s_i. \end{aligned}$$

Schließlich muss man noch sicherstellen, dass $S \neq \emptyset$ gilt. Dazu setzen wir einfach einen Knoten fest, der in S liegt – ob wir die Menge S selbst oder ihr Komplement beschreiben, spielt für die Ungleichung ja keine Rolle, es kommt nur auf $\delta(S)$ an. Also fügen wir z. B. $s_1 = 1$ als Gleichung hinzu. Jetzt besteht natürlich die Gefahr, dass alle s_i auf 1 gesetzt werden (dann wäre $d = 0$ möglich). Also wählen wir einen Knoten $t \in V$ mit $t \neq 1$ aus und setzen $s_t = 0$. Das führt wiederum dazu, dass wir die richtige Menge S mit unseren Festlegungen nicht finden, falls sie sowohl den Knoten 1 als auch den Knoten t enthält. Dem lässt sich aber einfach dadurch abhelfen, dass wir mit $t = 2$ beginnend alle möglichen Werte für t probieren, bis wir entweder das gesuchte S gefunden oder gezeigt haben, dass sich für keinen Knoten t eine passende Menge ergibt.

Die Summe aus der Schnittungleichung verwenden wir als Zielfunktion. So ist sichergestellt, dass immer eine zulässige Lösung des ILP existiert. Am Zielfunktionswert können wir dann ablesen, ob das ILP eine Menge S gefunden hat, deren Schnittungleichung verletzt ist (bzw. wir haben den Beweis, dass es keine passende Menge S mehr gibt).

Das ILP lautet dann

$$
\begin{aligned}
\min \sum_{e \in E} x_e^* d_e & \\
d_{ij} &\geq s_i - s_j, \\
d_{ij} &\geq s_j - s_i. \qquad\qquad s_1 = 1 \\
s_t &= 0 \\
s &\in \{0,1\}^{|V|} \\
d &\in \{0,1\}^{|E|}.
\end{aligned}
$$

Eine verletzte Schnittungleichung haben wir genau dann gefunden, wenn der Zielfunktionswert strikt kleiner als 2 ist (2 war die rechte Seite der Schnittungleichung).

---

d) Erkennen Sie das Separationsproblem wieder? Zu welcher Problemklasse gehört es? Was wissen Sie darüber?

Das oben hergeleitete ILP ist ein Minimum Cut-Problem (vgl. Termin 18A), das Duale eines Maximum Flow-Problems. Sie wissen bereits, dass Minimum Cut genau dann lösbar ist, wenn das zugehörige Maximum Flow-Problem eine Lösung besitzt, und dass die beiden Zielfunktionswerte dann übereinstimmen. Statt dem Minimum Cut-Problem können wir also auch das zugehörige Maximum Flow-Problem lösen. Für Flussprobleme kennen Sie vielleicht (aus einer anderen Vorlesung) schon einen Algorithmus. Falls nicht: Uns kommt es hier vor allem darauf an, dass man solche Probleme in polynomieller Zeit (sogar sehr effizient) lösen kann.

---

e) Angenommen, Sie hätten einen Algorithmus, der das Separationsproblem für Subtour Elimination Constraints effizient löst. Wie würden Sie daraus dann einen Schnittebenen-Algorithmus für TSP machen? Warum können Sie TSP damit nicht effizient lösen?

Naheliegend ist folgendes Vorgehen: Löse zuerst die Relaxation des TSP-ILP ohne Subtour Elimination Constraints. Verwende dann den Separationsalgorithmus, um eine verletzte Subtour Elimination Constraint zu bestimmen, und füge sie zur Relaxation hinzu. Löse diese Relaxation erneut und fahre fort, bis keine Subtour mehr gefunden wird.

Dieses Vorgehen hat leider zwei „wunde Punkte“: Zunächst einmal gibt es exponentiell viele Subtour Elimination Constraints. Wenn wir also der Reihe nach all diese Ungleichungen separieren und hinzufügen, hätte unser Algorithmus keine polynomielle Laufzeit mehr. Für diese Problem gibt es allerdings einen Ausweg: Ein berühmtes Resultat aus der polyedrischen Kombinatorik besagt, dass man mit einem polynomiellen Separationsalgorithmus auch polynomiell über das von den Schnittungleichungen beschriebene Polytop optimieren kann. Ganz grob gesprochen verwendet man dazu so etwas wie ein Analogon zur binären Suche, den Ellipsoid-Algorithmus, um nicht alle möglichen Schnitte tatsächlich separieren zu müssen. Details dazu würden hier leider viel zu weit führen.

Es bleibt aber das zweite Problem: Möglicherweise ist die Lösung der Relaxation selbst nach Einfügen aller Subtour Elimination Contraints noch nicht ganzzahlig. Das liegt daran, dass P_{TSP} mit den Gleichungen der Relaxation und den Subtour Elimination Constraints noch nicht vollständig beschrieben ist. Das Polytop hat noch eine ganze Reihe von Facetten, auf die wir bisher nicht eingangen sind (bestimmte Kammungleichungen z. B. induzieren Facetten) und auch nicht eingehen werden: Eine vollständige Beschreibung des TSP-Polytops kennt man bis heute nicht. Es kommt sogar noch schlimmer – ein Resultat von Karp und Papadimitriou besagt Folgendes: Falls das Problem „Gegeben die Dimension n und eine Ungleichung $a^T x \leq \beta$, definiert die Ungleichung eine Facette des TSP-Polytops?“ in $\mathcal{NP}$ liegt, dann ist $\mathcal{NP} = co - \mathcal{NP}$. Die letzte Aussage ist wieder eine der „großen ungelösten Fragen“ der Komplexitätstheorie, ähnlich wie „Ist $\mathcal{P} = \mathcal{NP}$?“. Beachten Sie, dass man nur zeigen müsste, dass das Problem in $\mathcal{NP}$ liegt, $\mathcal{NP}$-Vollständigkeit ist gar nicht verlangt!

---

f) Kennen Sie vielleicht einen Algorithmus, um das Separationsproblem effizient zu lösen? (Aus Optimierung 2 oder aus dieser Veranstaltung kennen Sie den nicht, aber vielleicht haben Sie in einer anderen Veranstaltung schon etwas dazu gehört.)

Wenn Sie schon davon gehört haben, kennen Sie wahrscheinlich den Algorithmus von Ford-Fulkerson oder den Algorithmus von Dinic, vielleicht auch den von Goldberg-Tarjan. Wenn nicht (und wenn Sie neugierig sind), finden Sie mehr zu diesen Algorithmen in [PS98] oder in [KV07].

Referenzen

[KV07] Bernhard Korte und Jens Vygen. *Combinatorial Optimization*. 4. Aufl. Springer, 2007.

[PS98] Christos H. Papadimitriou und Kenneth Steiglitz. *Combinatorial optimization: Algorithms and complexity*. Corrected reprint of the 1982 original. Dover Publications, Inc., Mineola, NY, 1998.