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.

b) Wieviele Subtour Elimination Constraints gibt es?

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).

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

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?

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.)