OPERATIONS RESEARCH
KACKERTSTR. 7
52072 AACHEN

Prof. Dr. Marco Lübbecke
Jonas Witt, M.Sc.

Operations Research 1
9. Übungsblatt

Diskussion: Am 08.01. in der Übung.

Alle Antworten sind jeweils kurz zu begründen!

Aufgabe 28 (TSP: Separieren von SECs) *(Trainingsaufgabe)*

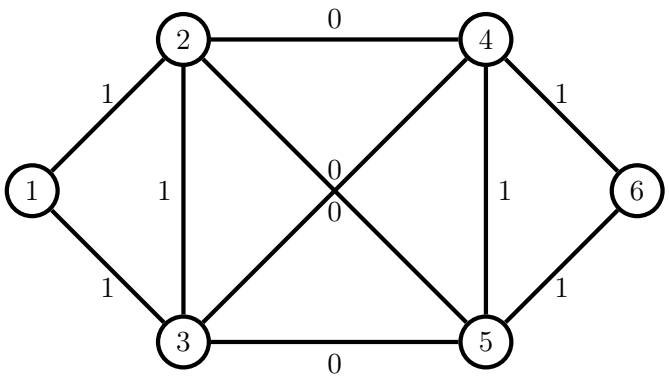
Gegeben sei eine TSP-Instanz $G = (V, E)$ mit Kosten c_{ij} für Kante $ij \in E$. Wir betrachten folgende Formulierung des TSP als ganzzahliges lineares Programm:

$$\begin{aligned}
\min \quad & \sum_{ij \in E} c_{ij} x_{ij} & & \\
\text{s. t.} \quad & \sum_{ij \in \delta(i)} x_{ij} = 2 & & \forall i \in V \\
& \sum_{ij \in \delta(S)} x_{ij} \geq 2 & & \forall S \subsetneq V, S \neq \emptyset \qquad (1) \\
& x_{ij} \in \{0,1\} & & \forall ij \in E \ .
\end{aligned}$$

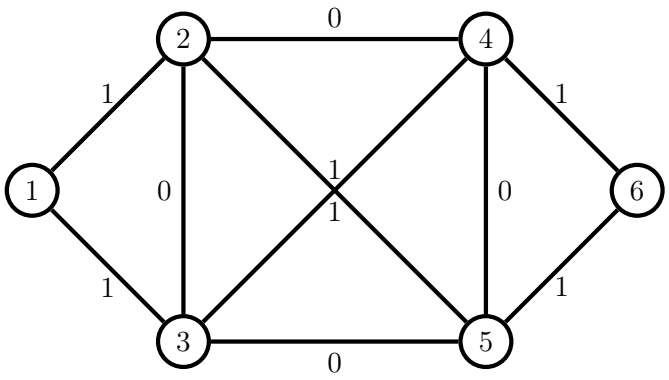
Sei nun $\bar{x} \in [0,1]^{|E|}$ eine Lösung der LP-Relaxation dieses ganzzahligen linearen Programms ohne Subtour-Eliminations-Constraints (1).

Die TSP-Instanzen sind im folgenden skizziert und die Werte auf den Kanten geben die Werte $\bar{x}_{ij}$ der x_{ij}-Variablen für $ij \in E$ an. Geben Sie jeweils einen verletzten Subtour-Eliminations-Constraint an oder begründen Sie, warum kein verletzter Subtour-Eliminations-Constraint existiert.

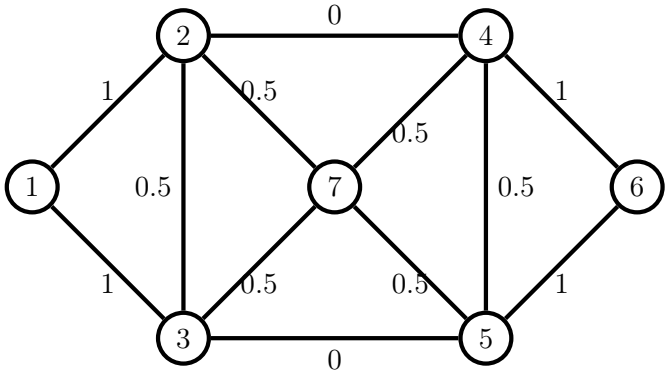
a)



b)



c)



Aufgabe 29 (Dynamische Programmierung) *(Trainingsaufgabe)*

Betrachten Sie das folgende Rucksackproblem bestehend aus 6 Elementen, deren Gewicht und Profit in der folgenden Tabelle angegeben sind:

i	1	2	3	4	5	6
a_i	1	4	1	3	2	5
p_i	4	1	5	2	2	7

Berechnen Sie eine optimale Lösung für eine Kapazität von 10 Einheiten mit Hilfe des Dynamischen Programms aus der Vorlesung.

Aufgabe 30 (GAMS: Santa's Stolen Sleigh) *(Bewertungsaufgabe: 20 Punkte)*

Da die Kaggle-Challenge (`https://www.kaggle.com/c/santas-stolen-sleigh`) immer noch nicht zur Zufriedenheit des Weihnachtsmannes gelöst wurde, beschäftigt sich der Weihnachtsmann nun selber mit der Lösung des Problems und hat dazu das Problem in Teilprobleme folgender Art aufgeteilt, für dessen Lösung er Ihre Hilfe benötigt:

Gegeben sei eine Menge $I = \{1, \ldots, n\}$ von Orten, die der Weihnachtsmann startend vom Nordpol aus mit seinem Ersatz-Schlitten besuchen möchte, um an jedem Ort genau ein Geschenk zu verteilen. Hierbei entspricht Ort 1 stets dem Nordpol. Für jeden Ort $i \in I$ seien Breitengrad lat_i und Längengrad long_i angegeben als Dezimalgrad sowie das Gewicht w_i des Geschenkes mit Ziel i gegeben. Hierbei gilt $w_1 = 0$ und der Schlitten hat ein zusätzliches Basisgewicht in Höhe von 10. Sie können davon ausgehen, dass die gegebenen Geschenke alle gleichzeitig auf dem Ersatz-Schlitten transportiert werden können. Breitengrad und Längengrad können von Dezimalgrad in Radiant umgerechnet werden, indem sie mit $\frac{\pi}{180}$ multipliziert werden. Die Distanz d_{ij} zwischen Ort $i \in I$ und Ort $j \in I$ sei gegeben durch die Haversine-Formel

$$d_{ij} = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\text{lat}_j - \text{lat}_i}{2}\right) + \cos(\text{lat}_i)\cos(\text{lat}_j)\sin^2\left(\frac{\text{long}_j - \text{long}_i}{2}\right)}\right) \ ,$$

wobei $r = 6378.388$ der Erdradius (Äquatorradius) angegeben in km ist.

Gesucht ist nun eine Rundtour durch alle Orte der Menge I, wobei die Tour im Nordpol starten und enden soll. Um die Belastung der Rentiere möglichst gering zu halten, soll dabei die *Gewichtete Rentier-Erschöpfung* minimiert werden. Diese berechnet sich für eine gegebene Besuchtsreihenfolge $i_1, \ldots, i_n$ der Orte in I (hierbei gilt immer $i_1 = 1$) zusammen mit der Definition $i_{n+1} = 1$ und unter der Annahme, dass der Schlitten ein Basisgewicht von 10 hat, als:

$$\sum_{p=1}^{n}\left(10 + \sum_{k=1}^{n} w_{i_k} - \sum_{l=1}^{p} w_{i_l}\right) d_{i_p i_{p+1}} \ .$$

Sie entspricht somit der zurückgelegten Distanz multipliziert mit dem Gewicht des gesamten Schlittes (inklusive Basisgewicht und Gewicht der Geschenke) für jeden Abschnitt.

Im Lernraum finden Sie die Datei `santa.gms` sowie `santa-data1.gms`, `santa-data2.gms` etc. Ergänzen Sie die Datei `santa.gms`, so dass das eben beschriebene Problem gelöst wird. Einige Parameter sind Ihnen in den Dateien `santa-data1.gms` etc. gegeben. Verwenden Sie diese als Input, wie schon in der Datei `santa.gms` vorvermerkt. Außerdem sind die Werte von π und r schon in der Datei `santa.gms` definiert. Sie können die anderen Instanzen testen, indem Sie den include-Befehl ändern. Die optimalen Zielfunktionswerte lauten 1085234.212738 für `santa-data1.gms`, 400765.937651 für `santa-data2.gms`, 2342305.374989 für `santa-data3.gms`, 2391289.813540 für `santa-data4.gms` und 2.237090 für `santa-data5.gms`.

Falls Sie auf Funktionen oder Begriffe stoßen, bei denen Sie nicht wissen, wie Sie diese in GAMS verwenden, werfen Sie bitte einen Blick in die GAMS-Dokumentation (z.B. für die Benutzung der Trigonometrischen Funktionen). Beachten Sie dabei insbesondere, ob die Trigonometrischen Funktionen in GAMS eine Eingabe in Dezimalgrad oder Radiant erwarten.

Falls Sie ein *Big M* verwenden, wählen Sie einen geeigneten Wert für M in Abhängigkeit von den Daten.

Laden Sie Ihre GAMS-Datei bis 08.01.2016 um 14:15 Uhr unter

`https://orb.or.rwth-aachen.de/ws15_or1/`

hoch.

Achten Sie dabei darauf, dass die Daten vom Modell getrennt sind, d.h. die Daten werden weiterhin über den `$include`-Befehl eingelesen (und nicht in die Datei kopiert!) und das Problem wird so modelliert, dass die Daten ausgetauscht werden könnten.

Hinweis: Versuchen Sie das Problem mit ähnlichen Variablen und Nebenbedingungen wie beim TSP mit Zeitfenstern zu modellieren.

Bemerkung: Die Schwierigkeit dieser Aufgabe liegt in der korrekten Berechnung der Distanzen, der Modellierung der Zielfunktion sowie der Verwendung eines geeigneten Wertes für *Big M* (in Abhängigkeit von den Daten).