Abteilung Algorithmik
Institut für Betriebssysteme und Rechnerverbund **WS 08/09**
TU Braunschweig

Prof. Dr. Sándor Fekete
Nils Schweer

Mathematische Methoden der Algorithmik
Übung 5 vom 21.01.2009

Schriftliche Abgabe bis zum 04.02.09 in den Schrank vor der Abteilung *Algorithmik*.

Aufgabe 1 (Traveling-Salesman-Problem):

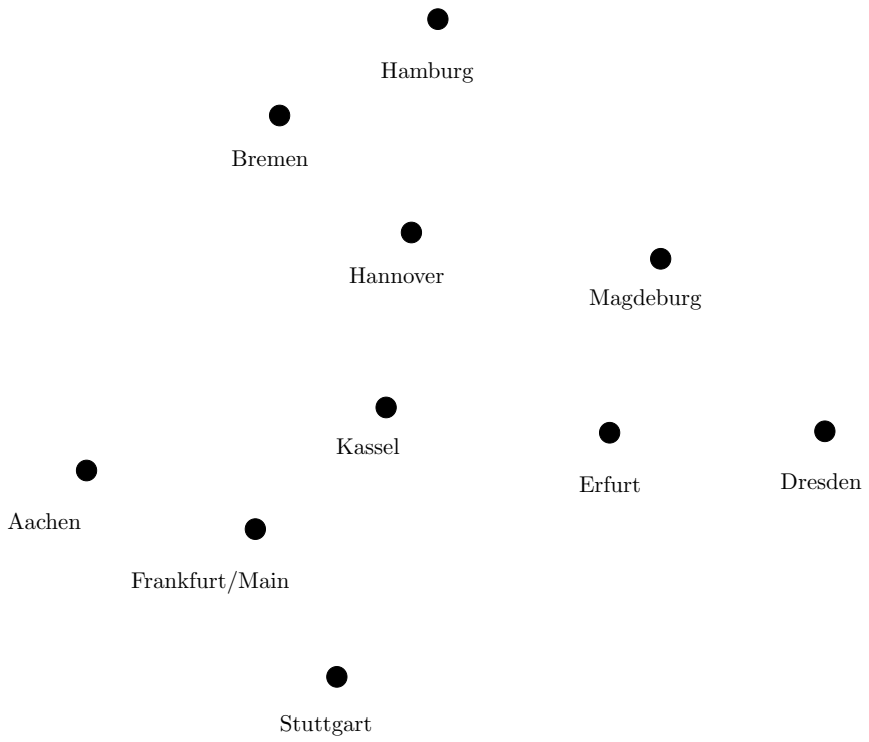

Abbildung 1: Zehn Städte in Deutschland

Zwischen je zwei dieser Städte gibt es eine direkte Straßenverbindung. Die Werte (auf Vielfache von 10 Kilometern gerundet) sind die folgenden:

		2	3	4	5	6	7	8	9	10
1	Aachen	37	65	45	24	48	35	31	50	41
2	Bremen		48	35	45	11	12	28	25	65
3	Dresden			22	49	49	39	40	23	53
4	Erfurt				27	38	29	14	21	44
5	Frankfurt/Main					51	36	19	45	20
6	Hamburg						15	31	27	66
7	Hannover							24	14	53
8	Kassel								25	36
9	Magdeburg									57
10	Stuttgart									

Wir betrachten obige Instanz des Traveling Salesman Problem (TSP).
Erzeuge eine LP-Datei, die das LP (ohne Subtour-Elimination-Constraints) zur obigen Instanz des TSP Problems enthält und löse es mit CPLEX. Dannach ergeben sich drei sinnvolle Möglickeiten, um eine Subtour-Elimination-Constraint einzufügen (Welche?). Füge diese jeweils einzeln hinzu und löse das modifizierte LP erneut. Welche Lösungen ergeben sich? (Zeichnung!)
Füge nun hinreichend viele Subtour-Elimination-Constraints ein, um einen zusammenhängenden Trägergraphen zu bekommen. Findest Du damit eine Optimallösung, die eine Rundreise beschreibt? Gib eine optimale Rundreise an.

Aufgabe 2 (Modellierung):

Modelliere die folgenden Probleme jeweils als IP. Beschreibe dabei jede Restriktion kurz mit eigenen Worten.

a) Gegeben sei ein Behälter der Größe W sowie n Objekte mit Größen w_i und Profiten p_i , $i = 1, \ldots, n$. Ziel ist es, den Behälter so zu befüllen, dass der Profit maximal ist.

b) Gegeben seien unendlich viele Behälter der Größe 1 sowie n Objekte mit Größen w_i, $i = 1, \ldots n$. Ziel ist es, alle Objekte in möglichst wenige Behälter zu packen.

(Hinweis: Eine Menge von Objekten kann in einen Behälter gepackt werden, wenn die Summe der Größen kleiner als die Behältergröße ist.)

c) Gegeben seien m (identische) Maschinen und n Jobs, wobei der i-te Job $p_i \in \mathbb{N}$ Zeiteinheiten ($i = 1, \ldots, n$) auf einer Maschine bearbeitet werden muss. Für jeden Job gibt es außerdem eine *release time*, d.h. einen Zeitpunkt zu dem der Job frühestens gestartet werden darf. Ziel ist es, die Zeit zu minimieren, bis der letzte Job bearbeitet ist.

(Hinweis: Ein Job darf nur auf einer Maschine bearbeitet werden und seine Bearbeitung darf nicht unterbrochen werden. Eine Maschine darf natürlich (nacheinander) mehrere Jobs bearbeiten.)

(8+10+12 Punkte)