Übungsblatt 3

Adrian Schollmeyer

Aufgabe 1

Das Netz ist jetzt vollständig dimensioniert. Entsprechend gibt es keinen variablen Anteil bei den Netzkapazitäten mehr.

$$F = r \tag{1}$$
$$\forall e = 1 \ldots, E : \sum_{d=1}^{D} \sum_{p=1}^{P_d} \delta_{edp} x_{dp} \leq c_e \cdot r \tag{2}$$
$$\forall d = 1 \ldots, D : \sum_{p=1}^{P_d} x_{dp} = h_d \tag{3}$$

Es kommt $r \approx 1,06$ heraus. Das bedeutet, dass das Netz mehr Kapazität benötigt als eigentlich vorhanden ist, um die Demands zu erfüllen. Dafür werden noch Kandidatenpfade $12-10-7-5$ und $2-7-10,9$ hinzugefügt, die dafür sorgen, dass die Kapazitäten nicht überschritten werden.

Aufgabe 2

(a)

Definiere:

- y_e, k... Anzahl Module vom Typ k auf Link e
- M_k... Größe eines Moduls vom Typ k (Konstante!)
- $\xi_{e,k}$ Kosten für die Installation eines Modules vom Typ k auf Link e

$$\min \sum_{e=1}^{E} \xi_{e,1} y_{e,1} \tag{4}$$

sodass

$$\sum_{d=1}^{D} \sum_{p=1}^{P_d} x_{dp} \delta_{edp} \leq M_1 y_{e,1} + c_e \qquad \forall e = 1, \ldots, E \tag{5}$$
$$\sum_{p=1}^{P} x_{dp} = h_d \qquad \forall d = 1, \ldots, D \tag{6}$$

Dadurch ergeben sich größere Linkkapazitäten, da nur in 300 $kbps$-Schritten Kapazität gekauft werden kann. An manchen Stellen wird dadurch Restkapazität von Links für andere Demands mitbenutzt, um die Anzahl zu kaufender Module zu reduzieren.

(b)

$$\min \sum_{e=1}^{E} \sum_{k=1}^{2} \xi_{e,k} y_{e,k} \tag{7}$$

sodass

$$\sum_{d=1}^{D} \sum_{p=1}^{P_d} \sum_{k=1}^{2} x_{dp} \delta_{edp} \leq M_k y_{e,k} + c_e \qquad \forall e = 1, \ldots, E \tag{8}$$

$$\sum_{p=1}^{P} x_{dp} = h_d \qquad \forall d = 1, \ldots, D \tag{9}$$

(c)

Für die Modellierung eines Budgets kann wieder ein Faktor r eingeführt und die Kosten nach oben beschränkt werden. Dann wird wieder r maximiert statt die Kosten minimiert. Daraus ergibt sich $r \approx 93\,\%$.

Aufgabe 3

(a)

$$f_1(x) = \frac{f(250)}{250} x \tag{10}$$

$$f_2(x) = a_2 x + b_2 \tag{11}$$

$$a_2 = \frac{f(500) - f(250)}{250} x \tag{12}$$

$$b_2 = f(250) - 250 a_k = -125\,000 \tag{13}$$

$$f_3(x) = a_3 x + b_3 \tag{14}$$

$$f_4(x) = a_4 x + b_4 \tag{15}$$

allgemein:

$$f_k(x) = a_k x + b_k \tag{16}$$

$$m = 250 \tag{17}$$

$$a_k = \frac{f(k \cdot m) - f((k-1) \cdot m)}{m} \tag{18}$$

$$b_k = f((k-1) \cdot m) - a_k \cdot (k-1) \cdot m \tag{19}$$

(b)

Um jetzt mit einer Approxmation zu arbeiten, muss man den Optimierer zwingen, im richtigen Intervall die richtige Funktion zu wählen. Dafür betrachten wir $f'_k(x) = a_k \cdot x$ und optimieren die Summe $\sum_k f'_k$ (also ohne Berücksichtigung der b_k). Durch Beschränkung, wie groß x_k werden darf, kann man den Optimierer zwingen, bei Überschreitung von m die nächstteurere Funktion zu wählen.

Dies funktioniert nur bei einer Minimierung, nicht bei einer Maximierung!

$$\min \sum_k f'_k(x_k) \tag{20}$$

$$\sum_k x_k = x \tag{21}$$

$$0 \leq x_k \leq m \tag{22}$$

Hier:

$$\min f = 250x_1 + 750x_2 + 1250x_3 + 1750x_4 \tag{23}$$

sodass

$$\sum_{k=1}^{4} x_k = 800 \tag{24}$$

$$\forall k = 1, \dots, 4 : 0 \leq x_k \leq 250 \tag{25}$$

Das Ergebnis ist 650 000. Das liegt daran, dass wir nicht genau an einer Stützstelle liegen und die Approximation die Originalfunktion überschätzt.

$$\min \sum_{e}^{E} \xi_e f_e \tag{26}$$

$$\forall e = 1, \dots, E : f_e = 250z_{e,1} + 750z_{e,2} + 1250z_{e,3} + 1750z_{e,4} \tag{27}$$

$$\forall e = 1, \dots, E : y_e = \sum_{i=1}^{4} z_{e,i} \tag{28}$$

$$\forall e = 1, \dots, E \forall i = 1, \dots, 4 : 0 \leq z_{e,i} \leq 250 \tag{29}$$

(die üblichen Nebenbedingungen sind hier nicht aufgeführt)

Bemerkung Wer mal mit nem richtigen Optimierer spielen möchte, kann sich mal Gurobi anschauen.