

## Praktische Optimierung mit Modellierungssprachen Aufgabenblatt 5

Abgabe: bis spätestens Montag, den 04.07.2016 um 23:59 Uhr,  
über das Online-Abgabesystem (für eine 1,0)

### Aufgabe 5 (Frequenzzuweisung für Antennen)

Im Mobilfunk senden die Antennen vieler Basisstationen auf jeweils einer Frequenz. Da es zu Interferenzen kommen kann, wenn diese zu nah beieinander stehen, werden mehrere Frequenzen benötigt. Die Lizenzen hierfür sind allerdings sehr teuer, sodass man daran interessiert ist, die Zuteilung zu den Antennen so zu arrangieren, dass insgesamt möglichst wenig Frequenzen benötigt werden.

Gegeben sind  $n$  Antennen mit ihren jeweiligen Positionen  $(x_1, y_1), \dots, (x_n, y_n) \in \mathbb{R}^2$  sowie ein kritischer Abstand  $R \in \mathbb{R}$ .

Gesucht ist eine Zuweisung einer minimalen Anzahl von Frequenzen  $1, 2, \dots$  zu den Antennen, sodass jede Antenne mindestens eine Frequenz erhält. Ferner dürfen keine zwei Antennen, deren euklidische Abstände kleiner oder gleich  $R$  sind, die gleiche Frequenz bekommen.

Entwickelt zunächst ein *IP*, welches das beschriebene Problem so modelliert, dass Lösungen, die durch das Vertauschen von Frequenzen auseinander hervorgehen, zum selben Lösungsvektor korrespondieren (diese Symmetrie also “gebrochen” wurde). Verwendet dafür eine Variable für jede Teilmenge aus Antennen, bei denen es zu keinen Interferenzen kommt. Wie ihr die Variable benennt ist euch überlassen. Im Folgenden wollen wir nur die *LP-Relaxation* dieses *IP*s mittels *column generation* lösen.

Entwickelt python-Code für die solve-Funktion der vorgegebenen Datei `freqAssignColGen.py`, welcher euer LP als gurobi-Modell erzeugt und dieses mittels *column generation* löst. Das Pricing-Problem soll ebenfalls mit gurobi gelöst werden. Hochzuladen ist Code, dessen solve-Funktion geeignete gurobi-Modelle erzeugt und damit die LP-Relaxation löst. Die solve-Funktion soll beide Modelle zurückgeben, also das Modell der LP-Relaxation des *Restricted Master Problems* und das Modell des Pricing-Problems (im Zustand nach letztem Pricing).

Die Instanzdateien sind bereits so vorbereitet, dass sie die `solve`-Funktion in der Datei `freqAssignColGen.py` aufrufen. Die hochgeladene Datei muss folgenden Kriterien genügen:

- Sie muss `freqAssignColGen.py` heißen und eine Funktion `solve(radius, positions)` enthalten.
- Dabei ist `radius` der kritische Radius  $R$  und `positions` eine Liste von Koordinatenpaaren  $(x_i, y_i)$ .
- Die Variable des Pricing-Modells, die zu der Antenne  $i$  gehört, soll den Namen `x_i`,  $i \in \{1, \dots, n\}$  haben.

- Die `solve`-Funktion muss ein Paar (`rmp`, `pp`) von Modellen **zurückgeben**! Dabei ist `rmp` das gurobi-Modellobjekt für das *Restricted Master Problem* und `pp` das gurobi-Modellobjekt zum Pricing-Problem.

**Hinweis:** Schaut in die Unterlagen und den Code der Veranstaltung vom 13.6.

Für eine 1,0 ladet ihr bitte eure Datei über das Online-Abgabesystem bis spätestens Montag, den 04.07.2016 um 23:59 Uhr hoch.