OPERATIONS RESEARCH
KACKERTSTR. 7
52072 AACHEN

Prof. Dr. Marco Lübbecke

Michael Bastubbe

Praktische Optimierung mit Modellierungssprachen
Aufgabenblatt 2

Abgabe: bis spätestens Montag , den 06.06.2016 um 23:59 Uhr,
über das Online-Abgabesystem (für eine 1,0)

Aufgabe 2 (Standordplanung von Feuerwachen)

Ein Landkreis hat beschlossen die Kosten für den Brandschutz seiner n Gemeinden zu prüfen. Gegebenenfalls können alte Feuerwachen $\{s1, \ldots, sm\}$ abgerissen und neue $\{t1, \ldots, tk\}$ gebaut werden. Die Gemeinden, sowie die alten und potentiellen neuen Standorte für Feuerwachen, sind durch x- und y-Koordinaten gegeben.

Jede Gemeinde ($\{b1, \ldots, bn\}$) soll von genau einer Feuerwache beschützt werden. Die Bewachungskosten, die dabei entstehen sind der euklidische Abstand zwischen Feuerwache und Gemeinde multipliziert mit einem gegebenen Kostenkoeffizient c (`cost_coef`).

Die Errichtungskosten einer potentiellen neuen Feuerwache t_i sind durch f_i (`construction_cost`), $i = 1, \ldots, k$ gegeben. Weiterhin sind die Abbrisskosten einer alten Feuerwache si durch g_i (`destrcution_cost`)), $i = 1, \ldots, m$ definiert.

Jede Feuerwache kann maximal b (`capacity`) Gemeinden beschützen.

Zur Zeit ist der Brandschutz jeder Gemeinde durch eine alte Feuerwache sichergestellt. Für jede Gemeinde ist gegeben (`currently_protected_by`)) welche Feuerwache die Gemeinde derzeit beschützt. Wird eine vorhandene Station nicht abgerissen, ändert sich die Zuordnung der von ihr beschützten Gemeinden nicht; insbesondere können ihr keine neuen Gemeinden zugeordnet werden.

Gesucht ist eine Auswahl von alten Feuerwachen, die abgerissen, eine Auswahl von potentiell neuen Feuerwachen, die errichtet werden und eine Zuordnung der Gemeinden auf dann existierende Feuerwachen, sodass jede Gemeinde von genau einer Feuerwache beschützt wird und die Summe aus Bewachungs-, Errichtungs- und Abbrisskosten minimal ist.

Im L^2P findet ihr die beiden Instanzen **firedata1.csv** und **firedata2.csv** zu einem Landkreis, der in Abbildung 1 schematisch dargestellt ist. Die erste Zeile beschreibt das Format der Dateien.

Entwickelt ein IP, welches das beschriebene Problem modelliert. Entwickelt python-Code, der die Instanzen einlesen und euer IP als gurobi-Modell baut und die gefundene optimale Lösung visualisiert. Hochzuladen ist 1) Code, der Instanzdateien einlesen kann und ein gurobi-Modell baut und zurückgibt mit dem ihr das Problem gelöst habt (Details unten) und 2) Visualisierungen eurer gefundenen Optimallösungen für die beiden Instanzen.

Genauer gesagt:

1) Ein Zip-archiv, das (mindestens) eine Datei runfirehouses.py enthält, in der eine Methode `solve(full_path_instance)` implementiert ist, die euer gurobi-Modellobjekt **zurück-**

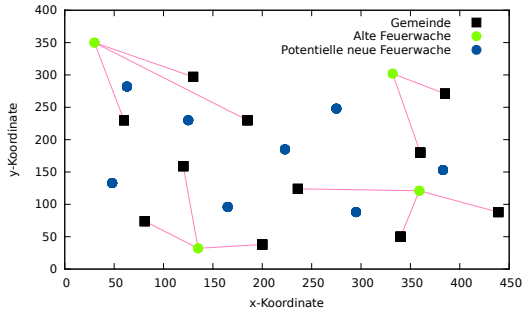

Abbildung 1: Landkreis mit Gemeinden

gibt!, wobei `full_path_instance` der Pfad zur Instanzdatei (als String) ist. Dabei ist weiterhin zu beachten:

- das zip-Archiv soll keinen Visualisierungscode beinhalten (ihr erspart euch also Arbeit, wenn ihr bereits zu Beginn die Visualisierung vom Modell trennt)
- das Modell soll mindestens die folgenden Variablen beinhalten, die exakt so heißen: "x_i_j" wobei $i \in \{s1, \ldots, sm, t1, \ldots, tk\}$ (`loc_id`s der Feuerwachen) und $j \in \{b1, \ldots, bn\}$ (`loc_id`s der Bezirke) sind, dabei soll $x\_i\_j = 1$ sein, genau dann wenn Feuerwache i Bezirk j beschützt (sonst $x\_i\_j = 0$). Beispiel: für **firedata1.csv** soll es also u.A. eine Variable mit Namen "x_s1_b1" geben.
- stellt sicher, dass die Dateien im Archiv ausreichen um durch Aufruf der Methode `solve(full_path_instance)` die jeweilige Instanz einzulesen und euer Modell zu returnen

2) Jeweils als pdf-Dateien, ihr könnt euch an den Bildern in Abbildung 2 orientieren. **Wichtig:** euer Gruppenname muss auf dem Bild auftauchen (z.B. bei der Achsenbeschriftung)

Die Lösungen zu den beiden Instanzen sollten ähnlich zu denen in Abbildung 2 sein.

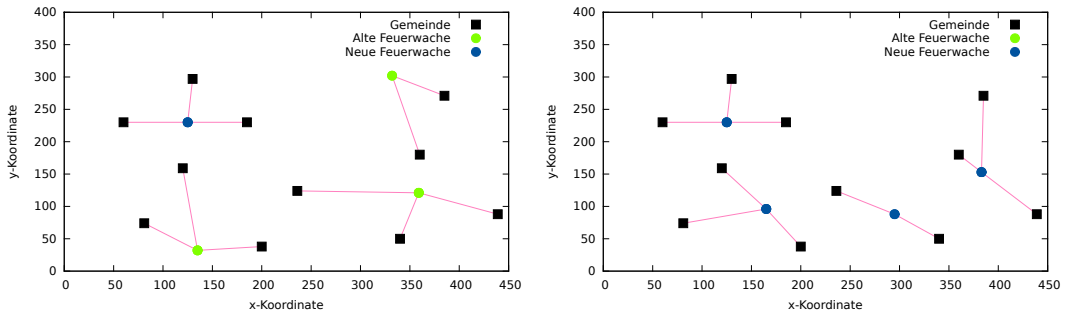

Abbildung 2: Optimale Lösungen zu fireData1.gms und fireData2.gms

Für eine 1,0 ladet ihr bitte eure Dateien über das Online-Abgabesystem bis spätestens Montag, den 06.06.2016 um 23:59 Uhr hoch.