Freie Universität Berlin
FB Mathematik und Informatik

Ralf Borndörfer, Fabio d'Andreagiovanni
Jonad Pulaj

Network Design and Operation (WS 2015)

Excercise Sheet 7

Submission: Mo, 7. December 2015, tutorial session

Exercise 1. **10 Points**

Consider the uncapacitated facility location problem

$$\begin{array}{llrcll}
\text{(UFL)} & \min & \sum_{i\in I} f_i y_i + \sum_{ij\in A} d_{ij}x_{ij} & & & \\
\text{(i)} & & \sum_{i\in I} x_{ij} & \geq & 1 & \forall j \in J \\
\text{(ii)} & & y_i & \geq & x_{ij} & \forall ij \in A \\
\text{(iii)} & & y_i & \in & \mathbb{Z}_+ & \forall i \in I \\
\text{(iv)} & & x_{ij} & \in & \mathbb{Z}_+ & \forall ij \in A,
\end{array}$$

where $I = \{1,\ldots,m\}$, $J = \{1,\ldots,n\}$, $A \subseteq I \times J$, $f \in \mathbb{R}^I_+$, and $d \in \mathbb{R}^A_+$. Prove:

a) (UFL) has an optimal 0/1-solution.
b) (UFL) $\iff$ (UFL) (i), (ii), (iii), $x_{ij} \geq 0\ \forall ij \in A$.

Exercise 2. **10 Points**

Consider the following set covering problem associated with model (UFL) of exercise 1:

$$\begin{array}{llrcll}
\text{(SCP)} & \min & \sum_{(i,J')\in\mathcal{J}} c_{(i,J')} z_{(i,J')} & & & \\
\text{(i)} & & \sum_{J'\ni j} z_{(i,J')} & \geq & 1 & \forall j \in J \\
\text{(ii)} & & z_{(i,J')} & \in & \{0,1\} & \forall (i,J') \in \mathcal{J}.
\end{array}$$

Here, $J(i) := \{j \in J : ij \in A\ \forall i \in I\}$, $\mathcal{J} := \{(i,J') : i \in I, \emptyset \subsetneq J' \subseteq J(i)\}$, and $c(i,J') := f_i + \sum_{j\in J'} d_{ij}\ \forall (i,J') \in \mathcal{J}$. Prove:

a) There is a one-to-one correspondence between optimal 0/1-solutions of (UFL) and (SCP).
b) UFL is APX-hard. **Hint:** SCP is APX-hard.

Exercise 3. **10 Points**

In the *(Metric) Capacitated Facility Location Problem* (MCFL), we are given a natural number u_i for each facility i, and a facility i can serve at most u_i clients.

Adjust the IP formulation for the *(Metric) Uncapacitated Facility Location Problem* (MUFL) to this situation and show that the integrality gap between the LP and the IP optimum is unbounded.

Exercise 4. **10 Points**

Modify the LP rounding Algorithm for the MUFL as follows:

i) Change the definition of x' in the filtering step to

$$x'_{ij} \leftarrow \begin{cases} 0, & \text{if } i \notin N_j(\beta) \\ \alpha \bar{x}_{ij}, & \text{else} \end{cases}$$

for an appropriate α of your choice.

ii) Replace in the LP rounding step N_j by $N_j(\beta) := \{i \in I : x_{ij} > 0 \wedge d_{ij} \leq \beta D_j\}$.

a) What is the resulting approximation ratio depending on β?
b) What is the best approximation ratio that can be obtained by modifying β?

Exercise 5. **Tutorial Session**

Consider a complete graph $G = (V, E)$ on an even number of nodes V in $\mathbb{R}^n$ with Euclidean distances as edge weights. The *Euclidean Perfect Matching Problem* (EPMP) is to find a minimum weight set of edges such that every node is contained in exactly one edge. Edmonds [1965] showed that the EPMP (and, in fact, any perfect matching problem) can be solved using the linear program

$$\begin{array}{llcll} \text{(EPMP)} & \min \sum_{uv \in V^2} c_{uv} x_{uv} & & & \\ \text{(i)} & x(\delta(v)) & = & 1 & \forall v \in V \\ \text{(ii)} & x(\delta(W)) & \geq & 1 & \forall W \subseteq V, |W| \text{odd} \\ \text{(iii)} & x_{uv} & \geq & 0 & \forall uv \in V^2; \end{array}$$

constraints (EPMP) (i) and (iii) are the *degree* and *odd set* or *odd cut* constraints, respectively. Consider the following primal-dual algorithm for the EPMP.

Algorithm 1: Primal-dual Euclidean perfect matching algorithm.

Input : complete graph $G = (V, E)$, $V \subseteq \mathbb{R}^n$, $c_{uv} := ||u - v||_2 \forall uv \in V^2$
Output: forest $F \subseteq E$

1 $y \leftarrow 0$, $k \leftarrow 1$, $F^k \leftarrow \varnothing$;
2 If there is no odd tree (with an odd number of nodes) in F^k, output $F \leftarrow F^k$, stop;
3 $uv \leftarrow \text{argmin}_{u \in V(S), v \in V(T), S,T \text{ different trees in } F^k} \epsilon_{uv} := \dfrac{c_{uv} - y_u - y_v - \sum_{uv \in \delta(W), W \subseteq V} y_W}{|V(S)\%2 + |V(T)|\%2};$
4 $e^k \leftarrow uv$, $\epsilon^k \leftarrow \epsilon_{uv}$;
5 $y_v \leftarrow y_v + \epsilon^k \forall v \in V$, $y_W \leftarrow y_W + \epsilon^k \forall W = V(T), T$ odd tree in $F^k, |W| > 1$;
6 $F^{k+1} \leftarrow F^k \cup \{e^k\}$, $k \leftarrow k + 1$, goto 2;

---

a) Set up the dual (EPMD) of (EPMP), associating variables y_v and y_W with the degree and odd cut constraints.
b) At least one of the trees S and T in step 3 is odd.
c) Algorithm 1 terminates in $|V| - 1$ iterations with a spanning forest F of even trees and a dual feasible solution y, i.e., y is feasible for EPMD.

d) A spanning forest of even trees can be reduced to a perfect matching of no greater weight by (i) deleting *even edges* (edges whose deletion subdivides an even tree into even trees) (ii) finding in some even tree with at least 4 nodes and all odd edges two leaves w and w' adjacent to a common third node v, deleting edges wv and $w'v$, and joining w and w'.

e) $L := \sum_{v \in V} y_v + \sum_{W \subseteq V} y_W$ is a lower bound for the weight of any perfect matching.

f) Consider an edge $uv \in F$. In iteration k, let S^K and T^k be the trees in F containing u and v, respectively, and let $c^k_{uv} := \epsilon^k(|S^k|\%1 + |T^k|\%1)$, if $S^k \neq T^k$, and 0 otherwise. Then $c^k_{uv} = 0$ for $uv \in F^k$ and $c_{uv} = \sum_{k=1}^{t} c^k_{uv}$, if Algorithm 1 terminates after t iterations.

g) If in each iteration $\sum_{uv \in F} c^k_{uv} \leq 2\epsilon^k \cdot |\{\text{odd trees in } F^k\}|$, then $\sum_{uv \in F} c_{uv} \leq 2L$, i.e., Algorithm 1 is 2-approximative.

h) Consider for some arbitrary, but fixed iteration k the forest $\bar{F}$ obtained from F by shrinking every tree of F^k into a single node. Partition the nodes of $\bar{F}$ into two sets $\bar{V}'$ and $\bar{V}''$ corresponding to odd and even trees, respectively. Then $\bar{F}$ has no leaves in $\bar{V}''$, $\sum_{v \in \bar{V}'} \deg_{\bar{F}}(v) \leq 2|\bar{V}'|$, and g) holds.

i) The values of the dual variables y in Algorithm 1 can be interpreted as radii of circles and “moats” around nodes and odd sets. Solve the two problems in Figure 1 graphically using Algorithm 1.

j) Implement the linear programming formulation EPMP for the Euclidean perfect matching problem. Solve it, and verify the optimality of the solution graphically.

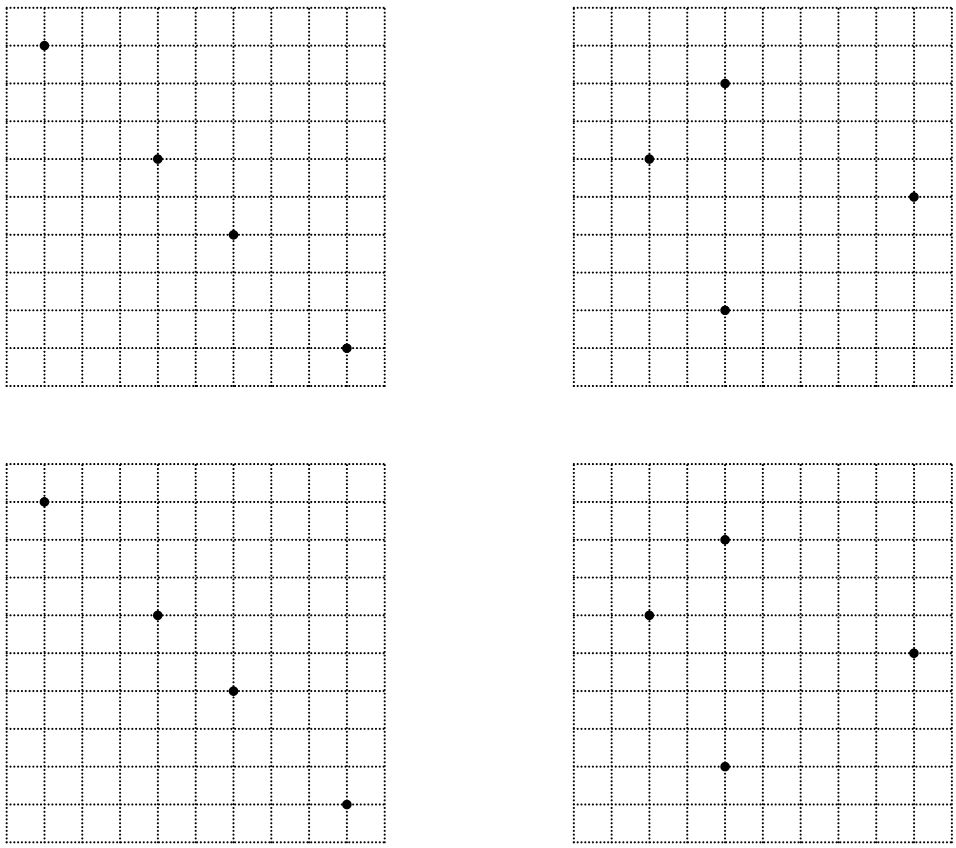

Figure 1: Euclidean perfect matching problem.