MAX PLANCK INSTITUTE FOR INFORMATICS

SIC Saarland Informatics Campus

Lecturers: **Evangelos Kipouridis and Tomasz Kociumaka**
Tutors: **Anouk Duyster and Yonggang Jiang**

Winter 2025/26

Randomized and Approximation Algorithms, Exercise Sheet 13

https://www.mpi-inf.mpg.de/departments/algorithms-complexity/teaching/winter-2025/26/randomized-and-approximation-algorithms

Total Points: **100** + 90 bonus points Due: **14:00** on Thursday, **February 5**, 2026

Writing LPs in LaTeX is not the easiest thing in the world. Personally I use this code:

```
\[
\begin{array}{ll}
\min & \sum_{j=1}^{n} c_j x_j \\
\text{s.t.} & \sum_{j=1}^{n} a_{ij} x_j \ge b_i \quad (i=1,\dots,m) \\
            & x_j \ge 0 \quad (j=1,\dots,n)
\end{array}
\]
```

Feel free to re-use it, or ask some LLM for suggestions.

Exercise 1 — $6 + (-1)$ bonus points

a Report how many work-hours were required, from the moment you opened this file to the moment you submitted your solutions.

b Give feedback on anything you want. Is the pace too slow/fast for you? Is the workload too little/much for you? You can clearly skip this part (which anyway gives negative points, to make not-replying even safer), or send me an anonymous email.

Exercise 2 — **15 + 5 + 20 + 10** points

In a previous lecture we discussed a 4-approximation algorithm for Correlation Clustering. This algorithm rounds the natural LP:

$$\begin{array}{ll} \min & \sum_{\{u,v\}\in E} d_{uv} + \sum_{\{u,v\}\notin E}(1-d_{uv}) \\ \text{s.t.} & d_{uv} \le d_{uw} + d_{vw} \quad (u,v,w \in V) \\ & d_{uv} \le 1 \quad (u,v \in V) \\ & d_{uv} \ge 0 \quad (u,v \in V) \end{array}$$

This algorithm selected an arbitrary unclustered node u, and based on some condition either created a singleton cluster containing only u, or created a cluster containing u and all nodes v with[1] $d_{uv} < 0.5$.

In the previous exercise sheet you were asked to bound the cost of our algorithm for non-edges within clusters. In particular, you had to prove that this cost is at most 4 times the LP cost. In this exercise, we prove that in fact this cost is upper bounded by the LP cost (with the factor 4).

a Derive the dual LP.

b Argue that the dual LP is feasible.

c Let d^*_{uv} be an optimal solution to the primal LP. Prove that the optimal objective value for the dual is at least as large as the number of pairs $\{u,v\} \notin E$ for which $d^*_{uv} < 1$. *Hint:* You can use the following fact from the upcoming lecture (Lecture 14). It is called *complementary slackness*: take any optimal solution to the primal LP, and any optimal solution to the dual LP. If a variable in the primal is larger than 0, then the corresponding dual constraint is tight (the inequality is an equality).

[1] In fact, in the lecture we clustered all nodes v with $d_{uv} \le 0.5$, but the analysis stays exactly the same under the new criterion which excludes nodes v with $d_{uv} = 0.5$ from the cluster.

d Conclude that the cost of the algorithm for the non-edges within clusters is at most equal to the LP cost.

Exercise 3 — 10 + 20 + 20 points + 10 + 15 bonus points

Fon online bipartite matching, we designed a simple online algorithm that is $\frac{1}{2}$ competitive. The idea is that whenever a node $r \in R$ appears, we connected it to *any* unmatched neighbor.

a (*) Prove that $\frac{1}{2}$ is the optimal (largest) competitive ratio we can achieve with a deterministic algorithm.

We now design a $(1 - \frac{1}{e})$ competitive algorithm. First, each node $l \in L$ draws a random number Y_l between 0 and 1 (let's assume that we can work with real numbers in this problem). Then, when a node $r \in R$ appears, we connect it to the unmatched neighbor $l \in L$ (if any) with the smallest drawn number Y_l.

Regarding the analysis, we proceed with a primal-dual approach. Let F be a positive number and $g(x)$ be an increasing function (in the end we will choose the $F, g(x)$ that optimize our analysis, which turn out to be $F = (1 - \frac{1}{e})$ and $g(x) = e^{-x}$). Every unmatched node $l \in L$ has $\alpha_l = 0$, and similarly every unmatched node $r \in R$ has $\beta_R = 0$. When we match nodes $l \in L$ with $r \in R$, we let

$$\alpha_l = \frac{g(Y_l)}{F}, \beta_r = \frac{1 - g(Y_r)}{F}$$

Let $(i, j) \in E$ be any edge in the graph with $i \in L, j \in R$. Consider an instance of the algorithm on $G \setminus \{i\}$, with the same choice of $Y_{i'}$ for all other $i' \in L$. Let y^c be the value of $Y_{i'}$ for the i' that is matched to j ($y^c = 1$ if j is unmatched). Let β_j^c be the value of β_j in this run. We further impose that $g(1) = 1$, which implies $\beta_j^c = \frac{1-g(y^c)}{F}$.

a Suppose you are given $Y_{i'}$ for all $i' \in L \setminus \{i\}$. Prove that if $Y_i < y_c$ then i gets matched.

b Suppose you are given $Y_{i'}$ for all $i' \in L \setminus \{i\}$. Prove that for all choices of Y_i we have $\beta_j \geq \beta_j^c$.

c Suppose g and F are such that $\forall \theta \in [0, 1]$ it holds that $(\int_0^\theta g(z)dz) + 1 - g(\theta) \geq F$. Prove that the duals constructed are feasible in expectation, that is $\mathbb{E}[\alpha_l + \beta_r] \geq 1$ for every edge (l, r).

d (*) It can be shown that the combination of g, F maximizing F and satisfying the above integral inequality is $g(x) = e^{-x}, F = (1 - 1/e)$. Using this fact, prove that our algorithm is $(1 - 1/e)$ competitive. *Hint:* The only reason this is not completely trivial, is that feasibility only holds in expectation, not always.

Exercise 4 — 10 + 20 + 10 + 20 bonus points

The small social network startup from the 8th assignment realized that deleting users is not good for business. So they now decide to drop friendships instead. The goal is once again the following: for any three people (say Alice, Bob, and Carol), whenever Alice is friends with Bob and Bob is friends with Carol, then Alice *must also* be friends with Carol. In graph terms, the friendship graph should be a disjoint union of cliques.

Unfortunately, the current network is messy. The startup asks you to delete as few friendships (edges) as possible so that the remaining friendship graph satisfies this policy.

Formally, you are given a simple graph $G = (V, E)$ and may remove edges. In the resulting graph $G' = (V, E')$, for every three distinct nodes $u, v, w \in V$ with $\{u, v\} \in E'$ and $\{v, w\} \in E'$, it must hold that $\{u, w\} \in E'$. Equivalently, each connected component of G' must be a clique. The goal is to minimize $|V| - |V'|$, the number of deleted nodes.

Task. We want to show that approximating this problem within a factor 1.99 in polynomial time is unlikely (impossible under the Unique Games Conjecture).

It suffices to show that an approximation factor 1.99 in polynomial time would imply an 1.999 approximation to the Minimum Vertex Cover, which is impossible under the Unique Games Conjecture. You can assume that n is sufficiently large (larger than any constant you want, as for small instances we can anyways brute-force). To do that, starting from a Minimum Vertex Cover instance $G_{VC} = (V_{VC}, E_{VC})$, we define the graph $G = (V, E)$ for our problem, where $V = V_{VC} \cup V'$, $|V'| = n^3$, and E contains all pairs with at least one endpoint in V' and all the pairs that do NOT share an edge in G_{VC}. We then solve our problem within an 1.99 approximation, and acquire a partition of V such that every connected component is a clique.

a (*) Prove that the optimal cost is less than $n^4 + n^2$.

b (*) Prove that there exists a connected component in the output whose size is at least $0.99n^3$.

c (*) Prove that the nodes in V_{VC} that are not contained in the big connected component are actually a vertex cover in G_{VC}.

d (*) Prove that the number of nodes in V_{VC} that are not contained in the big connected component is at most 1.999 times the Minimum Vertex Cover in G_{VC}.