Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institute of Theoretical Computer Science
Bernd Gärtner, Rasmus Kyng, Angelika Steger, David Steurer, Emo Welzl

Algorithms, Probability, and Computing **Special Assignment 2** **HS23**

- Write your solutions using a computer, where we strongly recommend to use LaTeX. **We do not grade hand-written solutions.**
- The solution is due on **December 12th, 2023** by **2 pm**. Please submit one file per exercise on Moodle.
- For geometric drawings that can easily be integrated into LaTeX documents, we recommend the drawing editor IPE, retrievable at `http://ipe.otfried.org` or through various package managers.
- Write short, simple, and precise sentences.
- This is a theory course, which means: if an exercise does not explicitly say "you do not need to prove your answer" or "justify intuitively", then a formal proof is **always** required. You can of course refer in your solutions to the lecture notes and to the exercises, if a result you need has already been proved there.
- We would like to stress that the ETH Disciplinary Code applies to this special assignment as it constitutes part of your final grade. The only exception we make to the Code is that we encourage you to verbally discuss the tasks with your colleagues. However, you need to include a list of all of your collaborators in each of your submissions. It is strictly prohibited to share any (hand)written or electronic (partial) solutions with any of your colleagues. We are obligated to inform the Rector of any violations of the Code.
- There will be two special assignments this semester. Both of them will be graded and the average grade will contribute 20% to your final grade.
- As with all exercises, the material of the special assignments is relevant for the (midterm and final) exams.

Exercise 1

10 **points**

(*Flow in bipartite graph*)
Let $G = (V, E)$ be bipartite graph where $V = S \cup T$, $|S| = n$, and $|T| = m$. Furthermore let $s : S \mapsto \mathbb{R}_{\geq 0}$, $d : T \mapsto \mathbb{R}_{\geq 0}$ be positive functions.

All the edges $e = (u, v)$ in G have one endpoint in S and one endpoint in T and G represents a network that connect nodes $i \in S$ with the nodes $j \in T$. Suppose that each node $i \in S$ comes with a quantity $s(i)$ of available resources and each node $j \in T$ comes with a demand of resources $d(j)$. We want to figure out whether there is a way to route the resources from the nodes in S to the nodes in T that the satisfies the demand. For simplicity, we make the assumption that $\sum_i s(i) = \sum_j d(j)$, i.e. the total quantity of resources available matches the total demand.

Create variable x_{ij} for each edge $(i, j) \in E$ and consider the following linear program.

$$\max \sum_{e \in E} 0 \, x_e$$

such that:

$$\sum_j x_{ij} \leq s(i), \quad \forall i \in S;$$

$$\sum_i x_{ij} \geq d(j), \quad \forall j \in T;$$

$$x_{i,j} \geq 0$$

(a) Prove that there is a way to match the resources to the demands if and only if the linear program is feasible.

(b) Write the dual of the linear program above.

(c) Prove that the first linear program has a solution if and only if the following condition holds. For each $Q \subseteq S$,

$$\sum_{i \in Q} s(i) \leq \sum_{j : \exists (i,j) \in E, i \in Q} d(j)$$

and for every $Q \subseteq T$,

$$\sum_{j \in Q} d(j) \leq \sum_{i : \exists (i,j) \in E, j \in Q} s(i)$$

.

Hint: look at Theorem 4.6 from the lecture notes.

Exercise 2 — 25 points

(*Grading scheduling*)

The APC exam consists of n exercises, and the course has m teaching assistants that partake in grading. Based on their skills, each TA i takes $t_{ij} > 0$ hours to grade exercise number j. We aim to grade all the exercises in the least time possible such that feedback can be provided in a timely manner[1]. In other words, let x_{ij} an indicator variable that attains value 1 if exercise j get assigned to TA i and zero otherwise. Then we want to minimize $\max_i \sum_j x_{ij} t_{ij}$. Consider the following integer program[2].

$$\min T$$

$$\sum_i x_{ij} \geq 1, \quad \forall j$$

$$T \geq \sum_j x_{ij} t_{ij}, \quad \forall i$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j$$

(a) Argue that the integer program finds the optimal assignment of exercises to TAs.

(b) Relax the program above to an LP (substitute $x_{ij} \in \{0, 1\}$ with $x_{ij} \in [0, 1]$) and prove that the integrality gap of the program above is at least m. In other words, find an example of $t_{i,j}$ so that the ratio between the best solution of the integer program above and the relaxed program is at least m.

We now modify the algorithm to get a much better integral solution via an intricate rounding scheme. Let $\tilde{t}$ be a fixed value and consider the relaxation of the LP as in (b) above with the extra constraint that

$$x_{ij} = 0, \quad \text{if } t_{ij} > \tilde{t}$$

for some threshold value $\tilde{t}$. When $\tilde{t}$ is chosen appropriately, this captures the intuition that questions should not be allocated to TAs that know very little about the topic.

(c) Show that the LP is feasible for some value $\tilde{t} = t_{ij}$.

Now, suppose that this new problem is feasible for some $\tilde{t}$ and and let x be an optimal but fractional solution x with cost $T(\tilde{t})$. We want to show that it is possible to round x into an integral solution of cost at most $T(\tilde{t}) + \tilde{t}$.

We let $q_i = \lceil \sum_j x_{ij} \rceil$. This intuitively corresponds to some educated guess of the number questions TA i is supposed to grade. To allocate questions to TAs, we build a bipartite graph with TAs on the left side, and questions on the right side. TA i appears q_i times on the left side, which encodes that they can be given up to q_i questions to grade. We refer to the copies of TA i as $i_1, \ldots i_{q_i}$. Each question just appears once and we will refer to its vertex as j.

[1] For simplicity, we assume that each exercise is only graded by one TA.

[2] A integer program is a linear program with the additional constraint that the solution has to be integral. This turns out to be a very powerful modification, that allows encoding NP hard problems, too.

Next, we define a tricky set of indices. We let $\pi_i(j)$ denote the index of the question TA i is j-th slowest at grading. To illustrate this definition, consider the following example with 3 questions where TA i takes two hours to grade question 1 ($p_{i1} = 2$), three hours for question 2 ($p_{i2} = 3$) and just one hour for question 3 ($p_{i3} = 1$). Then $\pi_i(1) = 2$, $\pi_i(2) = 1$ and $\pi_i(3) = 3$.

We then define a bipartite graph by adding the following edges for each TA i.

1. Initialize $c \leftarrow 1$, $j \leftarrow 1$, $r \leftarrow 1$ and $w \leftarrow x_{i\pi_i(1)}$.
2. While ($j \leq n$):
   - If $w \leq r$, add an edge of weight $x_{i\pi_i(j)}$ from exercise $\pi_i(j)$ to TA i_c if $x_{i\pi_i(j)} \neq 0$. Update $r \leftarrow r - w$, $j \leftarrow j + 1$ and $w \leftarrow x_{i\pi_i(j)}$ (with the newly updated j).
   - Else, add an edge of weight $x_{i\pi_i(j)}$ from exercise $\pi_i(j)$ to TA i_c if $x_{i\pi_i(j)} \neq 0$. Update $w \leftarrow w - r$, $r \leftarrow 1$ and $c \leftarrow c + 1$.

Finally, we construct an unweighted bipartite graph by taking all edges of the weighted bipartite graph we built.

(d) Prove that c is bounded by q_i when running the above algorithm for TA i, i.e. we don't attempt to add an edge to a copy of a TA that doesn't exist.

(e) Show that the unweighted bipartite graph built contains a matching with n edges.

Take a matching with n edges in the unweighted bipartite graph we built.

(f) Prove that the matching returned corresponds to an integral solution with value at most $T(\tilde{t}) + \tilde{t}$ where $T(\tilde{t})$ is the fractional solution obtained with threshold $\tilde{t}$.

(g) Let OPT be the cost of the optimal solution of the integer program, i.e. the time required for the grading with the optimal assignment. Describe a polynomial time algorithm that finds an assignment of the exercise to TAs such that the exam can be graded in time at most 2OPT.
Hint: How do you choose $\tilde{t}$?

Some useful facts.

Definition 1. *Given a graph $G = (V, E)$, a* fractional matching *of G is a function $z : E \mapsto [0, 1]$ such that*

$$\sum_{e \ni i} z(e) \leq 1$$

for all $i \in V$. The size of the fractional matching is $\sum_{e \in E} z(e)$.

Fact 2. *Every bipartite graph is a stable graph, i.e. the size of the biggest fractional matching in a bipartite graph is an integer and equals the size of the biggest matching in the graph.*

Exercise 3

20 **points**

(*Random orientation*)
Let $G = (V, E)$ be a graph and let $\overrightarrow{G}$ be a graph obtained by orienting the edges of G independently with probability $1/2$ in either direction. Given $\overrightarrow{G}$, define the skew symmetric matrix A_s such that

$$A_s(i,j) = \begin{cases} +1, & (i,j) \in E \text{ with orientation from } i \text{ to } j, \\ -1, & (i,j) \in E \text{ with orientation from } j \text{ to } i, \\ 0, & \text{otherwise.} \end{cases}$$

(a) Let $\mathrm{pm}(G)$ denotes the number of *perfect matchings* in G, prove that

$$\mathbf{E}\left[\det A_s(\overrightarrow{G})\right] = \mathrm{pm}(G).$$

Hint: Use the definition of determinant and the linearity of expectation.

(b) Given two perfect matchings M_1, M_2 of G, denote with $a(M_1, M_2)$ the number of cycles in G made of (alternating) edges contained in M_1 and M_2 (or equivalently the cycles in M_1 xor M_2), and denote with $\mathcal{M}$ the set of all perfect matchings in G. Prove that

$$\mathbf{E}\left[\left(\det A_s(\overrightarrow{G})\right)^2\right] = \sum_{M_1 \in \mathcal{M}} \sum_{M_2 \in \mathcal{M}} 3^{a(M_1,M_2)}.$$

Hint: You may find inspiration in the proof of Theorem 5.3. First deal with permutations whose directed graph uses edges that do not appear in G. Next argue about the contribution from permutations where this graph contains at least one cycle of odd length, and finally the rest.

(c) Let

$$\mathrm{var}\left[\det A_s(\overrightarrow{G_i})\right] = \mathbf{E}\left[\left(\det A_s(\overrightarrow{G})\right)^2\right] - \left(\mathbf{E}\left[\det A_s(\overrightarrow{G})\right]\right)^2.$$

Design a randomized algorithm that computes the determinant of at most $N = 2\,\mathrm{var}\left[\det A_s(\overrightarrow{G_i})\right]$ matrices, runs in $O(N \cdot \mathrm{poly}(|V|))$ time, and returns x such that

$$\mathbf{Pr}\left[|\mathrm{pm}(G) - x| \geq 1\right] \leq \frac{1}{2}.$$

(d) Find a collection of graphs G such that $\mathrm{var}\left[\det A_s(\overrightarrow{G})\right]$, where $\overrightarrow{G}$ is a random orientation of G, is *not* polynomially bounded in the size of G, i.e. there is no polynomial p such that $\mathrm{var}\left[\det A_s(\overrightarrow{G})\right] \leq p(|V(G)| + |E(G)|)$. (This means that the algorithm we just developed does not run in polynomial time for all the input graphs.)
Hint: Consider multiple copies of a graph containing only one cycle.

A useful fact.

Theorem 3 (Chebyshev's Inequality). *Let X be a random variable with finite expected value μ and finite non-zero variance σ^2. Then for any real number $k > 0$,*

$$\mathbf{Pr}\left[|X - \mu| \geq k\sigma)\right] \leq \frac{1}{k^2}.$$