ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institute of Theoretical Computer Science
Bernd Gärtner, Mohsen Ghaffari, Angelika Steger, David Steurer

Algorithms, Probability, and Computing **Special Assignment 2** **HS19**

- Write an exposition of your solution using a computer, where we strongly recommend to use LaTeX. **We do not grade hand-written solutions**.
- The solution is due on **Tuesday, December 3, 2019** by **2:15 pm**. Please bring a print-out of your solution with you to the lecture. If you cannot attend the lecture on time (and please only then), you may alternatively email your solution as a PDF to "chih-hung.liu@inf.ethz.ch" with subject "**[APC19] SA2 (Last-name_First-name)**", likewise **until 2:15 pm**. We will send out a confirmation that we have received your file. Make sure you receive this confirmation within the day of the due date, otherwise complain timely.
- If you attend an exercise group, write down your group, so that we can give your assignment back to you in the exercise class.
- For geometric drawings that can easily be integrated into LaTeX documents, we recommend the drawing editor IPE, retrievable at `http://ipe.otfried.org` in source code and as an executable for Windows.
- 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. 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

20 **points**

Let A be an $n \times n$ matrix whose entries are in $\{0, 1\}$. For every $i, j \in \{1, \ldots, n\}$, let $\epsilon_{i,j}$ be a number taken uniformly at random from $\{-1, 1\}$, and let B be the random matrix with $b_{i,j} = a_{i,j}\epsilon_{i,j}$.

(a) Prove that for any natural number k, any $2k + 1$ permutations $\pi_1, \ldots, \pi_{2k+1} \in S_n$, and any $i \in \{1, \ldots, n\}$ it holds that:

$$\mathbb{E}\left[\epsilon_{i,\pi_1(i)}\epsilon_{i,\pi_2(i)} \cdots \epsilon_{i,\pi_{2k+1}(i)}\right] = 0.$$

(b) By using the result of the previous task, prove that for any natural number k it holds that:

$$\mathbb{E}\left[\left(\det(B)\right)^{2k+1}\right] = 0.$$

Exercise 2

20 **points**

This exercise illustrates that linear programming is a tool for finding approximate solutions.

We have n points lying on a circle, numbered from 1 to n. For convenience, we denote by arc i the arc between vertices i and $i+1$, where $n+1 := 1$. Let P be a set of pairs. For each pair (i,j), we want to send a flow from point i to point j, clockwise or counterclockwise. The width w_i of the arc i is defined as the total number of flows through arc i. The goal is to minimize the circle width, which is defined as $\max_i w_i$. Fig. 1 shows an example.

(a) Denote by the variable w the circle width. Further, for each pair $(i,j) \in P$, we define the variable $x_{ij} \in \{0,1\}$, where $x_{ij} = 0$ corresponds to a clockwise flow from i to j, and $x_{ij} = 1$ corresponds to a counterclockwise flow from i to j. Design an integer linear program IP to model the problem with the variables above.

In order to show that IP models the problem, you should show that an optimal solution to IP corresponds to an optimal solution to the problem and vice versa.

Hint: Create a parameter a_{ijk}, where $a_{ijk} = 1$ if the counterclockwise flow from i to j passes through arc k and $a_{ijk} = 0$ otherwise.

(b) Denote by OPT the optimal value of IP. Note that IP is not a linear program because the constraint $x_{ij} \in \{0,1\}$ is not a linear constraint. Design an algorithm that involves solving a linear program such that the solution returned by the algorithm gives a circle width of at most 2OPT.

(You can assume that the linear program has an optimal solution without proving it.)

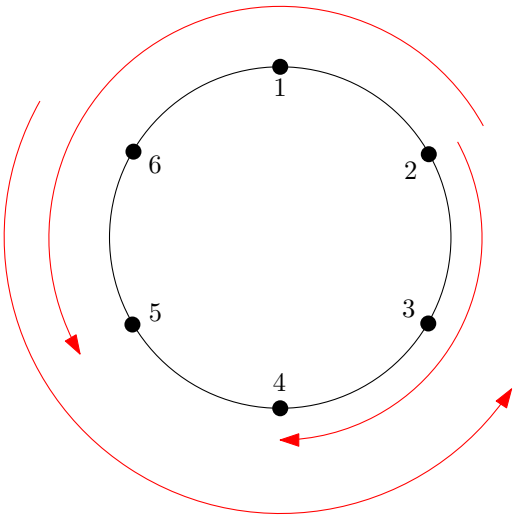

Figure 1: An example of flows for $P = \{(2,4),(2,5),(6,3)\}$ that minimize the circle width. The flow for the pair (2,4) is clockwise and those for (2,5) and (6,3) are counterclockwise. The minimum circle width is 2.

Exercise 3

20 **points**

Consider the following linear program (formulated as a geometric problem):

> Given a set H^* of n half-spaces in $\mathbb{R}^d$, compute the **lowest** vertex of the convex polyhedron that is the intersection among all half-spaces in H^*.

Assume that all the half-spaces are **upper**, and assume that no boundary hyperplane is vertical, the common intersection among more than d boundary hyperplanes is empty, and for $2 \leq j \leq d$, the intersection of any j boundary hyperplanes is $(d-j)$-dimensional. The notions of "lowest" and "upper" in this exercise are defined with respect to the x_d-axis. For the existence of the lowest vertex, we add one more half-space $h_0 : x_d \geq 0$. There may exist more than one lowest vertex, but we only need to compute one.

In the midterm exam, we have developed a randomized algorithm for $d = 2$ with expected $\mathcal{O}(n)$ running time. Denote this algorithm by *Random-2D-LP*. Now we will develop a randomized algorithm for $d > 2$ using Random-2D-LP as follows:

Algorithm 1 Random-LP(H^*, n, d)

```
if d = 2 then
  Return Random-2D-LP(H*)
end if
Generate a random permutation (h*_1, h*_2, ..., h*_n) of half-spaces in H*
Let h_i denote the boundary hyperplane of h*_i for 0 ≤ i ≤ n
Assign v to be the intersection point among the d hyperplanes, h_0, h_1, ..., h_{d-1}
for d ≤ i ≤ n do
  if v ∉ h*_i then
    Let H*_i be {h*_j ∩ h_i | 1 ≤ j < i}
    Update v to be Random-LP(H*_i, i − 1, d − 1)
  end if
end for
Return v
```

One may be aware that although each element in H_i^* is a $(d-1)$-dimensional half-space, it lies in $\mathbb{R}^d$. For simplicity, we assume the existence of two black-box tools: one tool converts the $i-1$ elements in H_i^* into half-spaces in $\mathbb{R}^{d-1}$ such that one lowest vertex in the primal space is still one lowest vertex in the dual space, and the other tool convert a vertex in the dual space into a vertex in the primal space. You do not need to worry about the conversion.

Complete the following three tasks:

(a) Prove that if $v \notin h_i^*$, then one lowest vertex of the polyhedron $\bigcap_{0 \leq j \leq i} h_j^*$ lies in h_i.

(b) Prove that if we only consider the running time used for the executions of Random-2D-LP, the expected running time of Random-LP is $\mathcal{O}(d! \cdot n)$.

(c) Give an asymptotically maximum value for d such that the expected running time is $\mathcal{O}(n^2)$. (*Hint*: $(\frac{d}{2})^{\frac{d}{2}} \leq d! \leq d^d$)