ETH

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** **HS22**

- 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 **Tuesday, December 6th, 2022** by **2:15 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 — 20 points

(*Minimum-Cost Distribution Networks Using Linear Programming*)
We consider the problem of finding a *minimum-cost distribution network*: We are given a weighted directed graph $G = (V, A)$, with cost function $c : A \mapsto \mathbb{N}$. Furthermore, we are given a special vertex $d \in V$, our *distribution base*. A *distribution network* is a subgraph $G' = (V, A')$ of G, such that for every vertex $v \in V \setminus \{d\}$, there exists a directed path from d to v in G'.

Consider the following integer program IP, where each arc $(u, v) \in A$ is represented by one variable $x_{(u,v)}$. We use the shorthand $c_{u,v} := c((u, v))$.

$$\text{minimize} \sum_{\{u,v\} \in A} c_{u,v} x_{(u,v)} \quad \text{subject to}$$

$$\forall C \subset V \text{ with } d \in C : \sum_{u \in C, v \in V \setminus C, (u,v) \in A} x_{(u,v)} \geq 1,$$

$$\forall (u, v) \in A : x_{(u,v)} \in \{0, 1\}.$$

(a) (8 points) Prove that the set of optimal solutions of IP is equal to the set of vectors corresponding to minimum-cost distribution networks.

(b) (12 points) It is known that relaxing all constraints $x_{(u,v)} \in \{0, 1\}$ of IP to the constraints $0 \leq x_{(u,v)} \leq 1$ yields a linear program, where the set of basic feasible solutions is equal to the set of feasible solutions of IP. Using this fact, give an algorithm to compute a minimum-cost distribution network in polynomial time.
Hint: You may use that a maximum flow can be found in polynomial time without proof.

Exercise 2

30 points

(*Matchings in Bipartite Graphs Using Linear Programming*)
We consider an undirected bipartite graph $G = (V, E)$.

(a) (6 points) Consider the following linear program:

$$\text{maximize } \mathbf{1}^T x \text{ subject to } \sum_{e \in \delta(v)} x_e \leq 1 \ \forall v \in V, \text{ and } x_e \geq 0 \ \forall e \in E,$$

where $\delta(v)$ is the set of edges incident to v.

Prove that any integral feasible solution of this LP corresponds to a matching in G, and any optimal integral solution corresponds to a maximum matching in G.

In the following, let x be a feasible solution x of the LP above. Consider the set of edges $F \subseteq E$ for which x_e is fractional (meaning it is not integral, i.e., $0 < x_e < 1$).

(b) (12 points) Prove that if F contains a cycle, x can not be a basic feasible solution.

(c) (12 points) Prove that if F is a non-empty forest, x can not be a basic feasible solution.

For both (b) and (c) you may use the following fact without proving it:
For any basic feasible solution x, there do not exist two distinct feasible solutions x_1, x_2 such that x is their midpoint, i.e., that $\frac{x_1 + x_2}{2} = x$.

Note that in this exercise, we have proven that the integrality gap of this linear program for bipartite maximum matching is 1.

Exercise 3 **20 points**

(*Pfaffian Orientations*)
Recall that an orientation of a graph G is a *Pfaffian orientation*, if every nice cycle is oddly oriented. As defined in the lecture notes, a *nice cycle* is an even cycle such that removing the vertices (and their incident edges) of the cycle from G yields a graph that contains a perfect matching. Furthermore, an even cycle is *oddly oriented* if when walking along the cycle in any of the two direction, we encounter an odd number of edges oriented against our walking direction.

For a graph G, we now consider a random orientation: Each edge is independently oriented with probability 1/2 in each of the two possible directions. We write G_{ro} for the resulting random orientation.

(a) (3 points) Give a graph G with at least one perfect matching, such that G_{ro} is a Pfaffian orientation with probability 1.

(b) (4 points) Characterize exactly (give a condition that is both necessary and sufficient) the graphs G for which G_{ro} is a Pfaffian orientation with probability 1.[1]

(c) (4 points) Show that there exists no graph G for which $1/2 < P[G_{ro} \text{ is a Pfaffian orientation}] < 1$.

(d) (3 points) For every positive integer k, give a graph $G^{(k)}$ such that $G^{(k)}_{ro}$ is a Pfaffian orientation with probability $1/2^k$.

Furthermore, prove the following two facts about Pfaffian orientations:

(e) (3 points) Let e be an edge of G such that e is not contained in any perfect matching of G. Then, G has a Pfaffian orientation if and only if $G \setminus \{e\}$ (the graph obtained by removing e from G) has a Pfaffian orientation.

(f) (3 points) Let e be an edge of G such that e is contained in exactly one even cycle of G. Furthermore, assume that every nice cycle in G that does not contain e is also a nice cycle in $G \setminus \{e\}$. Then, G has a Pfaffian orientation if and only if $G \setminus \{e\}$ has a Pfaffian orientation.

[1] Of course, the condition "All orientations of G are Pfaffian." or any similar trivial condition is not a solution.

Exercise 4 25 points

(*Unique Perfect Matchings*)
Devise randomized algorithms to solve the following problems. Your algorithms should run in $O(M(n))$ time (where $M(n)$ is the time needed to multiply two n by n matrices, and you can assume $M(n)$ is at least $\Omega(n^2)$) and succeed with probability at least $1/2$.

(a) (10 points) Let G be a bipartite graph, and M a given perfect matching in G. Decide whether there exists another perfect matching $M' \neq M$ in G.

(b) (5 points) Let G be a general graph, and M a given perfect matching in G. Decide whether there exists another perfect matching $M' \neq M$ in G.
Note that an algorithm that can solve (b) can also solve (a).

(c) (10 points) Let G be a general graph, and M_1 and M_2 be two distinct given perfect matchings in G. Decide whether there exists another perfect matching $M' \neq M_1, M' \neq M_2$ in G.

As always, you should prove the correctness of your algorithms, and analyze their runtime and success probability.