ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institute of Theoretical Computer Science
Rasmus Kyng, Bernd Gärtner, David Steurer, Vera Traub

Algorithms, Probability, and Computing **Exercises KW43** **HS25**

General rules for solving exercises

- When handing in your solutions, please write your exercise group on the front sheet:

  Group A: Wed 14–16 CAB G 56

  Group B: Wed 14–16 CAB G 57

  Group C: Wed 16–18 CAB G 56

  Group D: Wed 16–18 CAB G 57

- This is a theory course, which means: if an exercise does not explicitly say "you do not need to prove your answer", then a formal proof is **always** required.

The following exercises will be discussed in the exercise class on October 22, 2024. These are "**in-class**" exercises, which means that we do not expect you to solve them before the exercise session. Instead, your teaching assistant will solve them with you in class.

Exercise 1

We are given a set P of n points in $\mathbf{R}^2$ and a point q which has distinct distances to all points in P. We add the points of P in random order (starting with the empty set), and observe the nearest neighbor of q in the set of points inserted so far. What is the expected number of distinct nearest neighbors that appear during the process?

Exercise 2

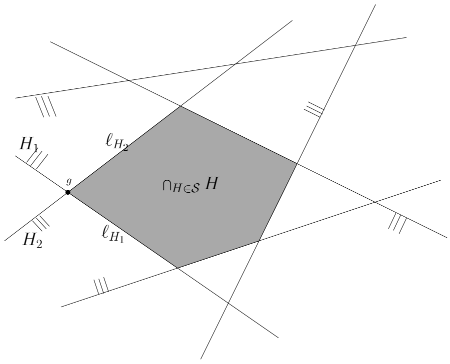
Show that every linear program can also be converted into the following *equational form*:

$$\text{maximize } c^T x \text{ subject to } Ax = b,\ x \geq 0.$$

What is the maximum increase in the number of variables and in the number of constraints in such a transformation?

Exercise 3

Suppose we are given a set $\mathcal{S}$ of n closed halfspaces in the plane. For each $H \in \mathcal{S}$, let $\ell_H \subset H$ denote its boundary line. We assume that the halfspaces are in general position such that no two boundary lines are parallel and no three boundary lines meet in a single point. Consider the input to be given in the form of linear inequalities, say.



In this task we are interested in a randomized algorithm to decide whether the intersection of the given halfspaces is non-empty, that is whether $R(\mathcal{S}) = \emptyset$ for $R(\mathcal{S}) := \bigcap_{H\in\mathcal{S}} H$, or not. If $\mathcal{S}$ has a non-empty intersection, we would also be interested in a *certificate point*, that is in a point $x \in \bigcap_{H\in\mathcal{S}} H$ to demonstrate non-emptiness. To make your calculations simpler, we want to make certificate points unique. To this end, we assume $|\mathcal{S}| \geq 2$ and fix, arbitrarily, two halfspaces $H_1, H_2, \in \mathcal{S}$. The region $R(\mathcal{S})$ is obviously contained in a wedge formed by the lines ℓ_{H_1} and ℓ_{H_2} (see figure). Before starting any algorithm, you may assume that the input is rotated[1] first in such a way that this wedge opens to the right and the intersection point $g \in \ell_{H_1} \cap \ell_{H_2}$ acts as a guard that no point in $R(\mathcal{S})$ can have a smaller x-coordinate than g (see figure). We then define for any $\mathcal{S}' \subseteq \mathcal{S}$ with $H_1, H_2 \in \mathcal{S}'$ the unique certificate point $c(\mathcal{S}')$ as the point in $R(\mathcal{S}')$ that has the smallest x-coordinate. You may assume that H_1 and H_2 are fixed before and known to all your algorithms below.

Following are your tasks:

(a) Let $|\mathcal{S}| \geq 3$ (with H_1 and H_2 as described above) and let $H \in \mathcal{S} \setminus \{H_1, H_2\}$ be an arbitrary one of the halfspaces. Prove: if $R(\mathcal{S}) \neq \emptyset$, then either $c(\mathcal{S}) = c(\mathcal{S} \setminus \{H\})$ or $c(\mathcal{S}) \in \ell_H$.

(b) Let $|\mathcal{S}| \geq 3$ (with H_1 and H_2 as described above) and let $H \in \mathcal{S} \setminus \{H_1, H_2\}$ be an arbitrary one of the halfspaces. Assume that $R(\mathcal{S} \setminus \{H\}) \neq \emptyset$. Write down a deterministic algorithm that runs in time linear in $n = |\mathcal{S}|$ and that on input $(\mathcal{S}, H, c(\mathcal{S} \setminus \{H\}))$ determines whether $R(\mathcal{S}) \neq \emptyset$ and if so outputs $c(\mathcal{S})$.

(c) Let again $|\mathcal{S}| \geq 3$ (with H_1 and H_2 as described above). Using (b), write down a randomized algorithm which, given $\mathcal{S}$, determines whether $R(\mathcal{S}) \neq \emptyset$ and if so outputs $c(\mathcal{S})$. Your algorithm should run in expected time linear in $n = |\mathcal{S}|$.

[1] this rotation can always be done such that we also do not have vertical or horizontal lines, which you may assume