ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institute of Theoretical Computer Science
Mohsen Ghaffari, Angelika Steger, David Steurer, Emo Welzl, Peter Widmayer

Solution for the in-class exercise

As in the lecture notes we use slack variables to get system (1) from the exercise description into form (S1):

$$\begin{array}{lrcl} \text{maximize} & c^T(x_1 - x_2) & & \\ \text{subject to} & A(x_1 - x_2) & \leq & b \\ & x_1, x_2 & \geq & 0 \end{array} \tag{S1}$$

This is the same as

$$\begin{array}{lrcl} \text{maximize} & \begin{pmatrix} c \\ -c \end{pmatrix}^T \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} & & \\ \text{subject to} & (A, -A) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} & \leq & b \\ & \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} & \geq & 0. \end{array} \tag{S2}$$

Using the definition of the dual we get that the dual of (S2) is

$$\begin{array}{lrcl} \text{minimize} & b^T y & & \\ \text{subject to} & (A, -A)^T y & \geq & \begin{pmatrix} c \\ -c \end{pmatrix} \\ & y & \geq & 0. \end{array} \tag{S3}$$

The claim follows since

$$(A, -A)^T y \geq \begin{pmatrix} c \\ -c \end{pmatrix}$$

is equivalent to $A^T y = c$.

Solution 1: Solving Linear Programs via Binary Search

Suppose we are given a linear program L that asks us to maximize the objective function $\mathbf{c}^T x$ where $\mathbf{c} \neq \mathbf{0}$. Minimization problems can be dealt with analogously, and problems with $\mathbf{c} = \mathbf{0}$ can be solved with one single call to an algorithm that solves the feasibility problem (because any feasible solution is also an optimal solution).

As we will see, with binary search we will not manage to find an optimal solution, but we can get arbitrarily close. So, suppose x^* is an optimal solution to the given linear program L with objective value $\mathrm{OPT} := \mathbf{c}^T x^*$. Our goal is to find an approximate solution $\tilde{x}$ that satisfies $\mathrm{OPT} - \mathbf{c}^T \tilde{x} \leq \epsilon$, where $\epsilon > 0$ is an arbitrary but fixed error term.

In order to perform binary search, we need to initialize and maintain upper and lower bounds on the optimum objective value OPT. For this we will need a stronger version of Theorem 6.2, which says that there exists an optimal solution, let us call it x^* without loss of generality, that is contained in the cube $[-K, K]^n$ with $K \leq 2^{O(\langle L \rangle)}$. We note that the proof in the lecture notes already implies this stronger statement.

It is an easy task to find the vertex x_{max} (resp., x_{min}) of the cube $[-K, +K]^n$ which maximizes $\mathbf{c}^T x_{max}$ (resp., minimizes $\mathbf{c}^T x_{min}$). Indeed, the sign of any coordinate of $\mathbf{c}$ corresponds to the sign of the corresponding coordinate of x_{max}. Also, clearly, $x_{min} = -x_{max}$. Since, as we said earlier, x^* is contained in $[-K, K]^n$ we get that $\alpha := \mathbf{c}^T x_{max}$ and $\beta := \mathbf{c}^T x_{min}$ are upper and lower bounds, respectively, for OPT.

Now we can perform binary search for OPT. That is, we let $\gamma := \frac{1}{2}(\alpha + \beta)$ and we add the constraint $\mathbf{c}^T x \geq \gamma$ to L. We check whether the new program is still feasible. If it is, then we update the lower bound $\beta := \gamma$. If it is not, then we remove the new constraint again and we update the upper bound $\alpha := \gamma$. In any case, the size of the interval $[\beta, \alpha]$ that contains OPT halves in every step of the search. Therefore, the number of steps until we reach $\alpha - \beta \leq \epsilon$ (and therefore also our goal $\mathrm{OPT} - \beta \leq \epsilon$) is at most

$$\log_2 \frac{\mathbf{c}^T x_{max} - \mathbf{c}^T x_{min}}{\epsilon} = \log_2(2^{O(\langle L \rangle)}) - \log_2(\epsilon) = O(\langle L \rangle),$$

for any fixed $\epsilon > 0$.

Solution 2: Equivalence of the Three Farkas Lemmas

(a) This is almost the exact same argument as in the previous exercise. Suppose there is $\mathbf{y}$ with $A^\mathsf{T}\mathbf{y} \geq \mathbf{0}$ and $\mathbf{b}^\mathsf{T}\mathbf{y} < 0$. Furthermore, towards a contradiction, suppose there is an $\mathbf{x} \geq \mathbf{0}$ with $A\mathbf{x} = \mathbf{b}$ (or, equivalently, $\mathbf{x}^\mathsf{T}A^\mathsf{T} = \mathbf{b}^\mathsf{T}$). We arrive at a contradiction (and hence conclude that $A\mathbf{x} = \mathbf{b}$ has no non-negative solution) by observing that

$$0 \leq \mathbf{x}^\mathsf{T}A^\mathsf{T}\mathbf{y} = \mathbf{b}^\mathsf{T}\mathbf{y} < 0,$$

where the first inequality is justified because both $\mathbf{x}^\mathsf{T}$ and $A^\mathsf{T}\mathbf{y}$ are non-negative.

Now suppose instead that there is $\mathbf{y} \geq \mathbf{0}$ with $A^\mathsf{T}\mathbf{y} \geq \mathbf{0}$ and $\mathbf{b}^\mathsf{T}\mathbf{y} < 0$. Furthermore, towards a contradiction, suppose there is an $\mathbf{x} \geq \mathbf{0}$ with $A\mathbf{x} \leq \mathbf{b}$ (or, equivalently, $\mathbf{x}^\mathsf{T}A^\mathsf{T} \leq \mathbf{b}^\mathsf{T}$). We arrive at a contradiction (and hence conclude that $A\mathbf{x} \leq \mathbf{b}$ has no non-negative solution) by observing that

$$0 \leq \mathbf{x}^\mathsf{T}A^\mathsf{T}\mathbf{y} \leq \mathbf{b}^\mathsf{T}\mathbf{y} < 0,$$

where the second inequality is justified because of our assumption $A\mathbf{x} \leq \mathbf{b}$ and because $\mathbf{y}$ is non-negative.

(b) We only prove the implication I $\Rightarrow$ II. The other implications can be proved in a very similar fashion.

We note that $A\mathbf{x} = \mathbf{b}$ has no non-negative solution $\mathbf{x}$ if and only if the system $\mathcal{A}\mathbf{x} \leq \mathcal{B}$ is unsolvable, where

$$\mathcal{A} := \begin{pmatrix} A \\ -A \\ -\mathbb{1} \end{pmatrix}, \qquad \mathcal{B} := \begin{pmatrix} \mathbf{b} \\ -\mathbf{b} \\ \mathbf{0} \end{pmatrix},$$

and where $\mathbb{1}$ is the identity matrix of appropriate dimension. Indeed, the system $\mathcal{A}\mathbf{x} \leq \mathcal{B}$ just encodes the constraints $A\mathbf{x} \leq \mathbf{b}$, $A\mathbf{x} \geq \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$. We now use Farkas lemma I and see that $\mathcal{A}\mathbf{x} \leq \mathcal{B}$ is unsolvable if and only if there is a vector $\mathcal{Y} \geq \mathbf{0}$ with $\mathcal{A}^\mathsf{T}\mathcal{Y} = \mathbf{0}$ and $\mathcal{B}^\mathsf{T}\mathcal{Y} < 0$. We write the vector $\mathcal{Y}$ as

$$\mathcal{Y} = \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_3 \end{pmatrix}$$

so that $\mathcal{A}^\mathsf{T}\mathcal{Y} = A^\mathsf{T}\mathbf{y}_1 - A^\mathsf{T}\mathbf{y}_2 - \mathbf{y}_3$ and $\mathcal{B}^\mathsf{T}\mathcal{Y} = \mathbf{b}^\mathsf{T}\mathbf{y}_1 - \mathbf{b}^\mathsf{T}\mathbf{y}_2$. Finally, we note that there exists $\mathcal{Y} \geq \mathbf{0}$ with $\mathcal{A}^\mathsf{T}\mathcal{Y} = \mathbf{0}$ and $\mathcal{B}^\mathsf{T}\mathcal{Y} < 0$ if and only if there is $\mathbf{y}$ with $A^\mathsf{T}\mathbf{y} \geq 0$ and $\mathbf{b}^\mathsf{T}\mathbf{y} < 0$, which concludes the proof. Indeed, for the "only if" we define $\mathbf{y} := \mathbf{y}_1 - \mathbf{y}_2$ and see that

$$A^\mathsf{T}\mathbf{y} = A^\mathsf{T}\mathbf{y}_1 - A^\mathsf{T}\mathbf{y}_2 \geq A^\mathsf{T}\mathbf{y}_1 - A^\mathsf{T}\mathbf{y}_2 - \mathbf{y}_3 = \mathcal{A}^\mathsf{T}\mathcal{Y} = \mathbf{0},$$

where the inequality is justified because $\mathcal{Y}$ (and thus also $\mathbf{y}_3$) is non-negative, and we also see that

$$\mathbf{b}^\mathsf{T}\mathbf{y} = \mathbf{b}^\mathsf{T}\mathbf{y}_1 - \mathbf{b}^\mathsf{T}\mathbf{y}_2 = \mathcal{B}^\mathsf{T}\mathcal{Y} < \mathbf{0}.$$

For the "if" we can always choose $\mathbf{y}_1$ and $\mathbf{y}_2$ in such a way that both are non-negative and such that $\mathbf{y}_1 - \mathbf{y}_2 = \mathbf{y}$. Since we know that $A^\mathsf{T}\mathbf{y} = A^\mathsf{T}\mathbf{y}_1 - A^\mathsf{T}\mathbf{y}_2 \geq \mathbf{0}$ we can also choose a non-negative $\mathbf{y}_3$ with $A^\mathsf{T}\mathbf{y} - A^\mathsf{T}\mathbf{y}_2 - \mathbf{y}_3 = \mathbf{0}$.

Solution 3: Deciding Feasibility vs. Finding Feasible Solutions

First, we check with one call to the oracle whether the given system of linear inequalities has a solution. If the answer is `NO` then we stop and also output `NO`. If the answer is `YES` then we proceed as follows.

(a) If there are only equations in the system, then this just means that we have a system $A\mathsf{x} = \mathbf{b}$ of linear equations, for which we can find a solution in polynomial time by Gauss elimination.[1] So, in this case we need no additional calls to the oracle.

(b) If there is at least one inequality, say $\mathsf{ax} \leq b$, then we replace it by $\mathsf{ax} = b$. If the new, more constrained, system still has a solution (which can be checked with one additional call to the oracle), then we can recursively find a solution to the original system by finding a solution to the more constrained system. If the new, more constrained, system turns out to have no solution, then we drop the constraint $\mathsf{ax} \leq b$ completely to obtain a smaller system, which again can be solved recursively. This is a sound strategy because replacing a constraint $\mathsf{ax} \leq b$ with $\mathsf{ax} = b$ can turn a feasible problem into an infeasible one if and only if the hyperplane corresponding to $\mathsf{ax} \leq b$ is not part of the boundary of the feasible region (in other words, it is redundant).

Since we need one initial call to the oracle, and exactly one call per inequality that we get rid of (either by replacing it with an equality or by dropping it completely), the total number of calls to the oracle will be $m+1$, where m is the number of inequalities in the original system.

[1] When the computation has to be done exactly, naive implementations of Gauss elimination can lead to an exponential blow-up of the encoding size of intermediate results. However, there are more clever implementations which do not have this problem and which do run in polynomial time.