Algorithmic game theory – Tutorial 1*

1 Linear programmming bootcamp

Many practical and purely combinatorial problems can be formulated as an instance of linear programming (LP). We can then use known methods on such an LP instance to solve it. Every LP instance can be written in the *canonical form* given by a matrix $A \in \mathbb{R}^{n \times m}$ and vectors $\mathbf{b} \in \mathbb{R}^n$ and $\mathbf{c} \in \mathbb{R}^m$:

$$\begin{aligned} \max \mathbf{c}^\top \mathbf{x} \\ \text{for } \mathbf{x} \in \mathbb{R}^m, \mathbf{x} \geq \mathbf{0} \\ \text{under constraints } A\mathbf{x} \leq \mathbf{b}. \end{aligned}$$

Exercise 1. *A bakery sells bread, rolls, baguettes, and donuts.*

- *To bake a loaf of bread, we need half a kilo of flour, 10 eggs, and 50 g of salt.*
- *For a single roll, we need 150 g of flour, 2 eggs and 10 g of salt.*
- *For a single baguette, we need 230 g of flour, 7 eggs, and 15 g of salt.*
- *To bake a donut, we need 100 g of flour and 1 egg.*

The bakery has 5 kilos of flour, 125 eggs, and half a kilo of salt. The price for a piece of bread is 20 crowns, 2 crowns for a roll, 10 for a baguette, and 7 crowns for a donut.

The baker wants to make as much money as possible. How many pieces of each pastry should he make? Formulate the problem as an LP instance.

Exercise 2. *Show how to:*

1. *Reformulate a maximizing LP instance as a minimizing one.*
2. *Reformulate an LP instance with variables $\mathbf{x} \geq \mathbf{0}$ as an instance with variables $\mathbf{x}' \in \mathbb{R}^m$ and vice versa.*
3. *Formulate an LP instance with constraints given by inequalities as an LP instance whose contraints are equalities and vice versa.*

If we can use integer variables, then even NP-hard problems can be formulated as integer LPs. Without integer variables, each LP can be solved in polynomial time. In practice, the so-called *simplex method* is used, as it is usually fast enough, although there are some rare instances where it can run for exponentially many steps.

Exercise 3. *Formulate the* Knapsack problem *using linear programming. That is, given n items, where the ith one has a weight v_i and price c_i, we have a knapsack with capacity V and we want to fill it with items so that the overall price of items in the knapsack is maximized while the capacity is not exceeded.*

2 Duality

Consider the following linear program P with m variables and n constraints:

$$\max \mathbf{c}^\top \mathbf{x} \text{ under constraints } A\mathbf{x} \leq \mathbf{b} \text{ a } \mathbf{x} \geq \mathbf{0}. \tag{P}$$

We call P the *primal linear program* (or simply *primal*). Its *dual linear program* (or simply *dual*) is the following linear program D with n variables and m constraints:

$$\min \mathbf{b}^\top \mathbf{y} \text{ under constraints } A^\top \mathbf{y} \geq \mathbf{c} \text{ a } \mathbf{y} \geq \mathbf{0}. \tag{D}$$

Explanation: when solving P, we are trying to find a linear combination of n constraints of the system $A\mathbf{x} \leq \mathbf{b}$ with some coefficients $y_1, \ldots, y_n \geq 0$ such that the resulting inequality has the jth coefficient at least c_j for each $j \in \{1, \ldots, m\}$ and the right-hand side is as small as possible.

*Information about the course can be found at http://kam.mff.cuni.cz/~balko/

Exercise 4. *Write a dual linear program D for the following primal linear program P:*

$$\begin{aligned} \max 6x_1 + 4x_2 + 2x_3 \\ 5x_1 + 2x_2 + x_3 \leq 5 \\ x_1 + x_2 \leq 2 \\ x_2 + x_3 \leq 2 \\ x_1, x_2, x_3 \geq 0 \end{aligned}$$

The following result is perhaps the most important theoretical result about linear programs.

Theorem 1 (The Duality theorem)**.** *Given the linear programs P and D, one of the following fours situations happens:*

(a) Neither P nor D has a feasible solution.

(b) The program P is unbounded and D does not have a feasible solution.

(c) The program P does not have a feasible solution and D is unbounded.

(d) Both programs P i D have a feasible solution. Then they optimal solutions $\mathbf{x}^*$ *and* $\mathbf{y}^*$ *and we have* $\mathbf{c}^\top \mathbf{x}^* = \mathbf{b}^\top \mathbf{y}^*$.

Rewriting general linear programs using duality can be done using the following table:

	Primal	Dual
Variables	$\mathbf{x} = (x_1, \ldots, x_m)$	$\mathbf{y} = (y_1, \ldots, y_n)$
Matrices	$A \in \mathbb{R}^{n \times m}$	$A^\top \in \mathbb{R}^{m \times n}$
Right-hand side	$\mathbf{b} \in \mathbb{R}^n$	$\mathbf{c} \in \mathbb{R}^m$
Objective function	$\max \mathbf{c}^\top \mathbf{x}$	$\min \mathbf{b}^\top \mathbf{y}$
Constraints	ith constraint has $\leq$	$y_i \geq 0$
	$\geq$	$y_i \leq 0$
	$=$	$y_i \in \mathbb{R}$
	$x_j \geq 0$	jth constraint has $\geq$
	$x_j \leq 0$	$\leq$
	$x_j \in \mathbb{R}$	$=$

Exercise 5. *Write a dual linear program D for the following primal linear program P:*

$$\begin{aligned} \max x_1 - 2x_2 + 3x_4 \\ x_2 - 6x_3 + x_4 \leq 4 \\ -x_1 + 3x_2 - 3x_3 = 0 \\ 6x_1 - 2x_2 + 2x_3 - 4x_4 \geq 5 \\ x_2 \leq 0 \\ x_4 \geq 0 \end{aligned}$$