Assignment 3

Please type down your answers using Latex. Please hand in your assignment (by email or by hard copy) before the class of November 4, 2020. You are welcome to discuss among yourselves, but everyone must write down his/her own answers. Please feel free to contact me if you have questions.

MAX SAT

Recall that in MAX SAT problem, we are given clauses $C_1, C_2, \cdots, C_m$, where each clause C_j contains disjuctive literals, either in the positive form x_i or in the negative form $\overline{x}_i$. Each clause C_j also comes with a weight w_j. The goal is to assign the values "true/false" to the literals so that that total weight of clauses satisfied is maximised.

Let P_j denote the set of literals in clause C_j that take positive form and N_j those with negative form. We can write the MAX SAT as the following linear program.

$$\begin{aligned} \max \sum_{j=1}^{m} w_j z_j & \\ \sum_{i \in P_j} y_i + \sum_{i \in N_j} 1 - y_i \geq z_j & \quad \forall \text{ clause } C_j \\ 0 \leq y_i \leq 1 & \quad \forall \text{ literal } x_i \\ 0 \leq z_j \leq 1 & \quad \forall \text{ clause } C_j \end{aligned}$$

Let y^* and z^* be the optimal fractional solution of this program. Now for each literal x_i, we set x_i to be true with probability $\frac{y_i^*}{2} + \frac{1}{4}$. Prove that this rounding strategy gives a 3/4-approximation, in expectation.

The following hint gives away almost everything: try to prove that each clause C_j has probability at least $\frac{3z^*}{4}$ of being true.

Machine Scheduling

Suppose that we are given a set of machines $\mathcal{M} = \{M_1, \cdots, M_m\}$ and a set of jobs J_1, $\cdots$, J_n. Each job J_i has a weight $w(J_i) \in \{1, 2\}$ and it can only be assigned to a subset $A(J_i) \subseteq \mathcal{M}$ of machines.

An *assignment* is a mapping ϕ of jobs to machines, under the restriction that a job J_i must be assigned to one of the machines allowed. In other words, $\phi(J_i) \in A(J_i)$. The *makespan* of an assignment ϕ is the largest total weight of jobs assigned to a machine, in other words,

$$\max_{i=1}^{m} \sum_{j:\phi(J_i)=M_i} w(J_i).$$

The objective is to find an assignment that minimises the makespan. This problem in fact is NP-hard. Design a 2-approximation algorithm for it.